ÉCOLE POLYTECHNIQUE DE BRUXELLES
UNIVERSITÉ LIBRE DE BRUXELLES

# Impact of real-world annotations on the training and evaluation of deep learning algorithms in digital pathology

**Thesis presented by Adrien FOUCART**
with a view to obtaining the PhD Degree in biomedical engineering (“Docteur en ingénierie biomédicale”)
Academic year 2022-2023

Supervisor: Professor Christine DECAESTECKER

Co-supervisor: Professor Olivier DEBEIR

Laboratory of Image Synthesis and Analysis

**Thesis jury:**

Hughes BERSINI (Université Libre de Bruxelles, Chair)
Gauthier LAFRUIT (Université Libre de Bruxelles, Secretary)
Geert LITJENS (Computational Pathology Group, Radboud University Medical Center)
Raphaël MARÉE (Montefiore Institute, Université de Liège)
Isabelle SALMON (Hôpital Érasme)
Gianluca BONTEMPI (Université Libre de Bruxelles)
Olivier DEBEIR (Université Libre de Bruxelles)
Christine DECAESTECKER (Université Libre de Bruxelles)

# Acknowledgments

This thesis is the product of seven years working in the LISA laboratory, as a researcher, a PhD student, and a teaching assistant. Many people have contributed, in one way of another, to the realisation of this work, and I would like to thank them here:

My supervisors, Pr. Christine Decaestecker and Pr. Olivier Debeir, for their support and their valuable feedback throughout the thesis, and for giving me the opportunity to start this thesis in the first place.

The past and present researchers, professors, and members of the lab that I had the pleasure to interact with during these years, for their ideas and inputs, and for the shared discussions, coffees, drinks, burgers, and all the things that make the work environment a better place. I particularly would like to thank Arlette Grave for her support through the various administrative processes of the university. My thanks also go to Isabelle Salmon of DIAPath, and the staff at Erasme Hospital and at the CMMI that contributed materials and feedback.

Being a teaching assistant was extremely stimulating and provided countless opportunities for procrastination, so I need to thank all students who kept me busy during these years. In particular, I would like to thank Élisabeth Gruwé and Alain Zheng for their contribution to this research through their excellent work during their master thesis.

I also want to thank Seyed Alireza Fatemi Jahromi and Amirezza Mahbod for sharing valuable data from their own research, and the NVIDIA Corporation for providing us with a Titan X GPU, which was used in most of the experimental work of this thesis.

I would not be where I am today without my parents Lucy Dever and Dominique Foucart, and the undoubtedly very effective education and support that they have provided to their children. My brothers François and Renaud must also get some thanks (and/or blame) for their contributions, as does my uncle Thierry Dever, without whom I probably wouldn't have gotten as comfortable with computers.

Finally, for her support (even during the last months of this thesis), I thank my partner Céline Mathieu. And last (but certainly not least) my daughter Margot, for making sure that I didn't stay *too* focused.

# Table of Contents

# List of abbreviations

We provide here, in alphabetical order, a list of abbreviations used in this thesis.

| | |
|---|---|
| ACC | Accuracy |
| AE | Auto-Encoder |
| AI | Artificial Intelligence |
| AP | Average Precision |
| ASSD | Average Symmetric Surface Distance |
| AUPRC | Area Under the PR Curve |
| AUROC | Area Under the ROC curve |
| BCC | Basal Cell Carcinoma |
| BP | Backpropagation |
| CE | Cross-Entropy |
| CM | Confusion Matrix |
| CNN | Convolutional Neural Network |
| CUDA | Compute Unified Device Architecture (software layer for GPU programming) |
| CYDAC | Cytophotometric Data Conversion System |
| DCNN | Deep Convolutional Neural Network |
| DA | Data Augmentation |
| DCNN | Deep Convolutional Neural Network |
| DL | Deep Learning |
| DNN | Deep Neural Network |
| DP | Digital Pathology |
| DQ | Detection Quality |
| DSC | Dice Similarity Coefficient |
| ER+ | Oestrogen receptor positive |
| FCN | Fully Convolutional Network |
| FN | False Negative |
| FNN | Feedforward Neural Network |
| FP | False Positive |
| FPR | False Positive Rate |

| | |
|---|---|
| GAN | Generative Adversarial Network |
| GlaS | Gland Segmentation challenge |
| GM | Geometric Mean |
| GMDH | Group Method of Data Handling |
| GPU | Graphical Processing Unit |
| H&E | Haematoxylin and Eosin |
| HD | Hausdorff's Distance |
| HER2 | Human epidermal growth factor receptor 2 (also the name of a 2016 competition) |
| HSV | Hue-Saturation-Value colour space |
| ICPR | International Conference on Pattern Recognition |
| IDC | Invasive Ductal Carcinoma |
| IHC | Immunohistochemistry |
| ILSVRC | ImageNet Large Scale Visual Recognition Challenge |
| IoU | Intersection over Union |
| ISUP | International Society of Urological Pathology |
| ISBI | IEEE International Symposium on Biomedical Imaging |
| $\kappa, \kappa_U, \kappa_L, \kappa_Q$ | Cohen's kappa (in general); Unweighted, Linear and Quadratic Cohen's kappa. |
| LSTM | Long Short-Term Memory |
| mAP | Mean Average Precision |
| MCC | Matthews Correlation Coefficient |
| MDS | Multi-Dimensional Scaling |
| MICCAI | Medical Image Computing and Computer Assisted Interventions (scientific society and conference) |
| MIL | Multiple Instance Learning |
| ML | Machine Learning |
| MLP | Multi-Layer Perceptron |
| MNIST | Modified National Institute of Standards and Technology (handwritten digits dataset) |
| MoNuSAC | Multi-organ Nuclei Segmentation and Classification challenge |
| MP | Max Pooling |
| MRI | Magnetic Resonance Imaging |

| | |
|---|---|
| MSE | Mean-Squared Error |
| NCM | Normalized Confusion Matrix |
| NPV | Negative Predictive Value |
| NSD | Normalized Surface Distance |
| PCA | Principal Component Analysis |
| PPV | Positive Predictive Value |
| PQ | Panoptic Quality |
| PR curve | Precision-Recall curve |
| PRE | Precision |
| PR in HIMA | Pattern Recognition in Histological Image Analysis (2010 challenge) |
| R | Rate of agreement |
| REC | Recall |
| ReLU | Rectified Linear Units |
| ResNet | Network architecture based on “residual units” |
| RGB | Red-Green-Blue colour space |
| ROC | Receiver Operating Characteristic |
| SEN | Sensitivity |
| SGD | Stochastic Gradient Descent |
| SNOW | Semi-Supervised, NOisy and/or Weak |
| SPE | Specificity |
| SQ | Segmentation Quality |
| SSL | Semi-Supervised Learning |
| STAPLE | Simultaneous Truth and Performance Level Estimation |
| SVM | Support Vector Machine |
| TCGA | The Cancer Genome Atlas |
| TCIA | The Cancer Imaging Archive |
| TMA | Tissue Microarray |
| TN | True Negative |
| TNM | Tumour-Node-Metastasis (cancer staging system) |
| TNR | True Negative Rate |
| TP | True Positive |

| | |
|---|---|
| TPR | True Positive Rate |
| UL | Unsupervised Learning |
| U-Net | Network architecture based on “long-skip” connections |
| WSI | Whole-Slide Imaging / Whole-Slide Image |
| WSL | Weakly Supervised Learning |
| XML | Extensible Markup Language |

# Notations sheet

While ad-hoc mathematical conventions are sometimes adopted in the manuscript depending on the context (and are thus explained in the text), we reference here the main notations that we use through the thesis.

| | |
|---|---|
| $x, y, t$ | Input, output, target output of a system (algorithm, neuron, function…)<br>**Bold letters** generally denote vectors |
| $w$ | Parameter (weight) of a model |
| $L(y, t)$ | Loss function |
| $T = \{T_i\}$,<br>$P = \{P_i\}$ | Sets of target and predicted items |
| $\lvert \cdot \rvert$ | Cardinality of a set |
| $CM$ | Confusion matrix, with $CM_{ij}$ the confusion between target class $i$ and predicted class $j$ |
| $m, c$ | Number of classes/categories in a dataset, class index. |
| $N, i$ | Number of samples in a dataset, sample index |
| $\lambda_c$ | Class imbalance parameter |
| $\pi_c$ | Class proportionality parameter |
| $\delta$ | Distance parameter |
| $Normal(\mu, \sigma)$ | Normal distribution for random sampling with mean $\mu$ and standard deviation $\sigma$ |
| $Uniform(a, b)$ | Uniform distribution for random sampling with bounds $[a, b]$ |

# Introduction

This dissertation studies how the reality of digital pathology annotations affects modern image analysis algorithms, as well as the evaluation processes that we use to determine which algorithms are better. In the ideal supervised learning scenario, we have access to a "**ground truth**": the output that we want from the algorithms. This "ground truth" is assumed to be unique, and the evaluation of algorithms is typically based on comparing it to the actual output of the algorithm. In the world of biomedical imaging, and more specifically in digital pathology, the reality is very different from this ideal scenario. Image analysis tasks in digital pathology are trying to replicate assessments made by highly trained experts, and these assessments can be complex and difficult, and therefore come with different levels of subjectivity. As a result, the annotations provided by these experts (and typically considered as "ground truth" in the training and evaluation of deep learning algorithms) are necessarily associated with some uncertainty. Our work focuses on different aspects of the impact of this uncertainty, as detailed below.

**Table 0.1. Publications associated with this thesis**

| Year | Title and journal/conference | Reference |
|---|---|---|
| 2018 | Artifact identification in digital pathology from weak and noisy supervision with deep residual networks. *4th International Conference on Cloud Computing Technologies and Applications (CloudTech)*. | [1] |
| 2019 | SNOW: Semi-supervised, Noisy and/or weak data for deep learning in digital pathology. *16th International Symposium on Biomedical Imaging (ISBI).* | [2] |
| 2019 | Strategies to Reduce the Expert Supervision Required for Deep Learning-Based Segmentation of Histopathological Images. *Frontiers in Medicine* (as second author). | [3] |
| 2020 | SNOW supervision in digital pathology: managing imperfect annotations for segmentation in deep learning. *Preprint on ResearchSquare*. | [4] |
| 2021 | Processing multi-expert annotations in digital pathology: a study of the Gleason2019 challenge. *17th International Symposium on Medical Information Processing and Analysis (SIPAIM)*. | [5] |
| 2022 | Comments on "Monusac2020: A Multi-Organ Nuclei Segmentation and Classification Challenge". *IEEE Transactions on Medical Imaging.* | [6] |
| 2022 | Evaluating participating methods in image analysis challenges: lessons from MoNuSAC 2020. *Preprint on ResearchGate (submitted for publication, in revision)*. | [7] |
| 2022 | Shortcomings and areas for improvement in digital pathology image segmentation challenges. *Preprint on ResearchGate (submitted for publication, in revision)*. | [8] |

In the rest of this introduction, we summarize the context of digital pathology image analysis, we explain how our publications (listed in Table 0.1) contribute to the domain, and finally we explain the structure of the dissertation and the main contributions of our work.

In the 1960s, the Cytophotometric Data Conversion (CYDAC) system was developed to acquire microscopic images of cells onto magnetic tapes. Based on those images, Prewitt and Mendelsohn explored the possibility of developing computer vision algorithms to automatically differentiate four cellular types. Their proposed 1966 method [9], illustrated in Figure 0.1, relied on the extraction of optical density features from the images. Samples from the four cell types could then be projected onto a two-dimensional feature space, which could be separated in four corresponding quadrants. Based on their results, Prewitt and Mendelsohn appeared cautiously optimistic on the prospects of image analysis for contributing to automated microscopic diagnosis, writing [9, p. 1052]:

> *This preliminary success encourages us to try the method on larger and more inclusive populations, but future research alone will tell whether this approach will yield the discriminatory power to master the normal blood smear and to go beyond it.*

Technology steadily improved in the following years: better acquisition devices, better storage mechanisms, improved computer vision algorithms. Yet the basic premise of Prewitt and Mendelsohn's approach remained relevant well into the beginning of the 21st century. A 2007 method by Doyle et al. [10] for the grading of prostate cancer, for instance, follows a similar overall structure, as shown in Figure 0.2. The images are larger, in colour, and with a better quality. Instead of two features, they have more than 100. Finally, instead of a manual separation of the feature space in four quadrants, they use a Support Vector Machines (SVM) classifier to discriminate between four cancer types.

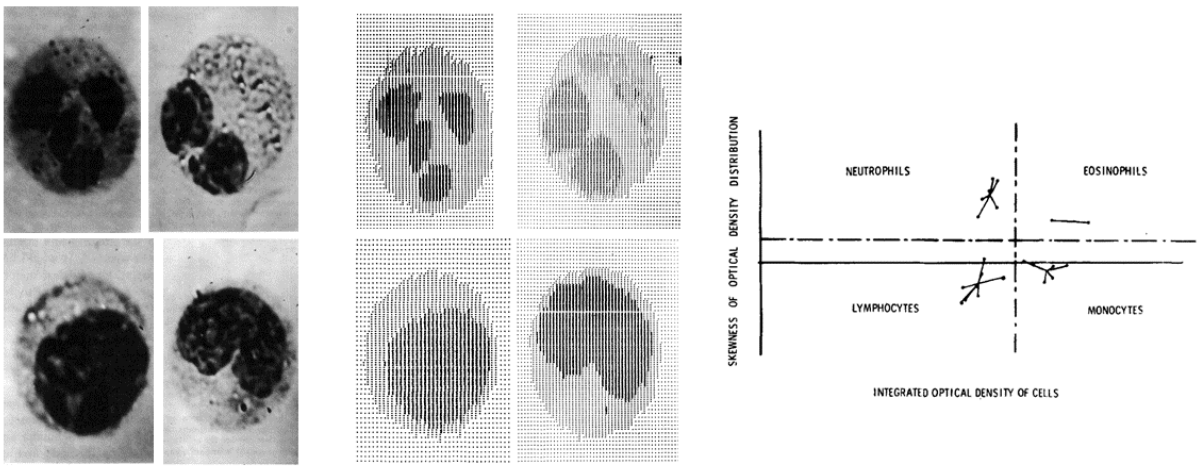



**Figure 0.1. Illustration of a pipeline for classifying four cell types (neutrophils, eosinophils, lymphocytes, and monocytes) based on images acquired from the CYDAC system. On the left are reference photographs of one cell per type. In the middle are printouts from the "boundary determination" step, separating the cells in regions of similar optical density. On the right, several samples per cell types are placed in a two-dimensional feature space based on the distribution of optical densities, with a proposed decision function (dashed line) to separate the four classes. All images are reproduced from Prewitt and Mendelsohn** [9]**.**

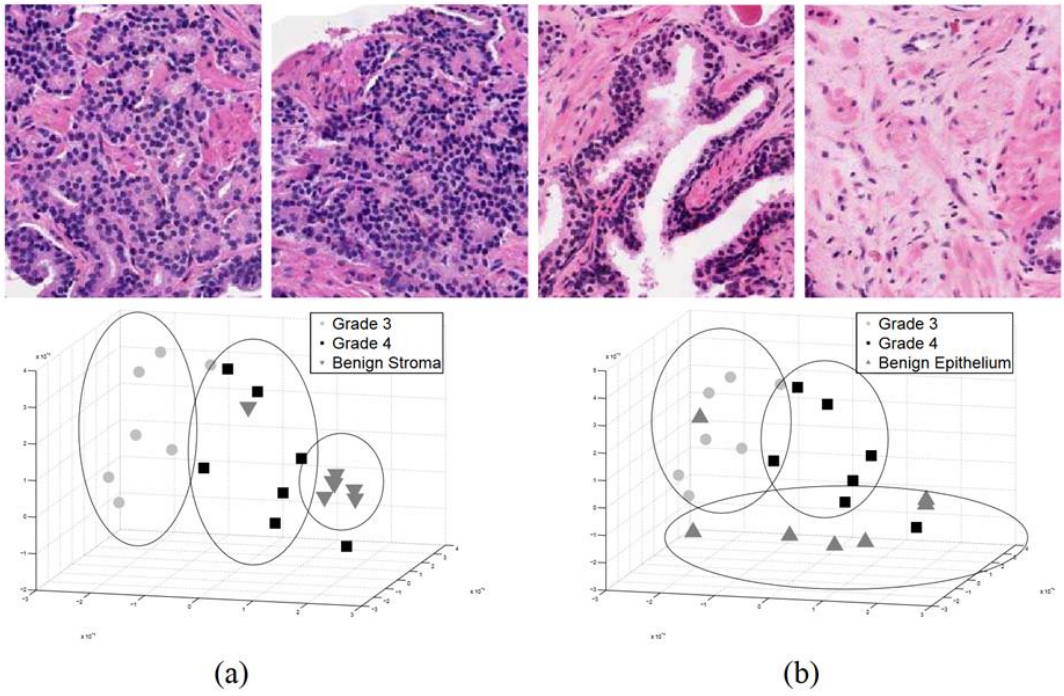



**Figure 0.2. Illustration of a pipeline for automatic grading of prostate cancer from H&E-stained prostate biopsy image. Top: tissue patches corresponding to, from left to right, Gleason grade 3 adenocarcinoma, Gleason grade 4 adenocarcinoma, benign epithelium and benign stroma. Bottom: scatter plot of the tissue regions reduced to a 3D space feature embedding, with ellipses highlighting "clusters of similar types". All images reproduced from Doyle et al.** [10]**.**

The 2009 review of "histopathological image analysis" by Gurcan et al. [11] came at this critical time in the history of computer vision, at the peak of what is now often considered the "traditional approach". Typically: pre-processing, extraction of "handcrafted features", dimensionality reduction or feature selection, and finally classification with for instance SVMs, or decision trees. In just a few years, this "traditional approach" would be largely replaced by the "deep learning" approach. Where the traditional approach can generally be thought of as "expert driven", deep learning is primarily "raw-data driven". Instead of explicitly defining the processing steps and the features to be extracted, let an algorithm derive those steps from the data themselves. To make a "raw-data driven" approach work, however, unsurprisingly requires a large amount of data. The transition from traditional computer vision to deep learning is therefore intimately linked to another transition: from "traditional" to "digital" pathology [12], [13].

Data acquisition in "traditional" pathology required manually taking pictures of regions of interest, a time-consuming process. The lack of data storage (and bandwidth for transmission) also severely limited how many images could reasonably be used for image analysis research. Whole-slide scanners, able to digitize high-resolution microscopy images relatively easily, were initially developed in the late 1990s and started to really become widely adopted in the late 2000s [14]. Combined with lower prices for data storage and increased availability of high bandwidth internet access, they created the opportunity for raw-data driven approaches to become viable. Alongside this increased access to data came an increased access to computing power, and particularly to *parallel* computing power, in the form of "General Purpose" Graphical Processing Units (GPGPUs). The CUDA library was launched in 2007 [15] by the NVIDIA company, and GPUs

have since largely moved from being "just" about graphics to being essential tools for parallel computing, and particularly adapted to image processing.

The early 2010s where when it all came together, with widespread access to GPUs, large quantities of data from digital pathology, and the release of easy-to-use software for the development of deep neural networks (such as Theano in 2007, Caffe in 2013, Keras and TensorFlow in 2015 [16]).

When the work on this thesis started in 2015, "deep learning" was in a transitional phase. It was no longer a niche studied by a handful of specialized research teams, and the "deep learning revolution" was already well underway, but it did not yet appear to be the *only* widely used approach, as it is now. Deep neural networks had been used in digital pathology tasks with very promising results by then [17]–[20], yet despite the apparent success of such methods, it still seemed that their adoption in clinical practice was a long way ahead. Seven years later, it is still unclear what place exactly deep learning methods will be able to take in the pathology workflow. To quote a 2021 review by van der Laak et al. [21, p. 780]:

> *Even though promising results for deep learning [computational pathology] algorithms have been shown in many studies, it is still too early to distinguish the hope from the hype. (…)*
>
> *What could be achieved relatively soon is AI algorithms that work in conjunction with pathologists, rather than as stand-alone solutions, to remove the need for tedious, repetitive work, such as identifying lymph node metastases, or to increase the quality of diagnostic grading.*

As we started working on potential applications for deep learning in digital pathology, such as the detection of artefacts [1], a key characteristic of pathology datasets attracted our focus: the frequent absence of genuine "**ground truth**".

The rise of popularity of deep learning came in large part through image analysis challenges, with the most famous example being the "ImageNet Large Scale Image Recognition Challenge" [22]. In ImageNet, the target classes are relatively straightforward. In total, more than one million images spanning 1000 classes were used in the challenge[1]. These are ideal conditions for deep learning algorithms: a large quantity of data, with a reliable ground truth. If the supervision says that an image shows a bicycle, a sock, a cucumber, or a triceratops, it can be safely assumed that this label is correct.

In digital pathology, there is no such abundance of *annotated* data. With whole-slide scanners, and the efforts of organisations, such as the US National Cancer Institute, to aggregate images acquired from multiple hospitals and make them available to the public[2], it has become easier to find large quantities of *images*. But knowing the exact content of these images is a very different matter.

The objects of interest in digital pathology image analysis are, in general, very loosely defined. The boundaries of the objects (which can be very fuzzy) and their exact nature are not necessarily straightforward to determine. The task of automatically determining a diagnostic based on the images is even more complex. Digital pathology datasets are therefore necessarily smaller, and the available "expert annotations" often cannot be considered as the "ground truth", but rather an

[1] https://www.image-net.org/download.php
[2] https://portal.gdc.cancer.gov/

expert's opinion. The question of how the inevitable imperfections of digital pathology datasets influence the results of deep learning algorithms thus became a large point of interest in our work [2]–[4].

A less studied aspect of imperfect annotations is their impact on the **evaluation** of image analysis algorithms. In challenges and other publications, evaluation metrics are computed as if the annotations in the test set are certain. Yet the same imperfections that make the learning process more difficult also make the evaluation process less reliable. The question of interobserver variability is particularly relevant here: when experts disagree, how do we really measure the relative performance of algorithms? This question led us to our study of the Gleason 2019 challenge [5]. Also important in the evaluation process of deep learning algorithms is the variability due to the randomized nature of the algorithms themselves: from the initial conditions of the algorithms to the techniques of data augmentation, and the randomness in the optimization of the models. The importance of using proper statistical tools in the assessment of the algorithms, and of recognizing the limits of the conclusions we can draw from those results, is something that we examined in the evaluation of our experiments [2], [4], and that is further developed in this work.

As we reviewed the results of different **digital pathology challenges** [8] and explored the choices made by challenge organisers, we found some interesting points of attention. First, we noticed that several challenges suffered from a lack of quality control in the published dataset and/or in the evaluation process. We had already noticed some problems in the Gleason 2019 challenge [5], and we further found issues with the MoNuSAC 2020 challenge, which were reported in a comment article [6] that led to a correction [23] of the previously published results [24]. Second, it became clear that the choice of an evaluation metric that fits a particular digital pathology task is far from trivial. We used the MoNuSAC challenge results to study how the choice of metric can hide or reveal important insights on the strengths and weaknesses of the participating teams' results [7].

This is therefore the central question to be explored in this work: **what is the impact of real-world annotations in digital and computational pathology**? This dissertation is structured as follows. In **Chapter 1** and **Chapter 2**, we provide context on what "deep learning" and "digital pathology" are, on their definitions and their history. In **Chapter 3**, we review the state of the art of deep learning in digital pathology, with a particular focus on the various competitions organized since 2010. **Chapter 4** examines in detail evaluation metrics and processes, their use in digital pathology competitions, and through several experiments and original analysis we outline their limitations and biases and provide recommendations for future research. The impact of incomplete, imprecise, and noisy annotations on the learning and evaluation processes is explored in **Chapter 5**. In **Chapter 6**, we look at a practical case with our work on artefact detection and segmentation. The question of interobserver variability is the focus of **Chapter 7**. Our findings and subsequent recommendations on quality control problems in competitions are explained in **Chapter 8**. A general discussion of our findings and a conclusion can be found in **Chapter 9**. In addition to these chapters, a description of some of the main datasets used through this work can be found in **Annex A**. In **Annex B**, we describe the deep learning models used in our experimental work, as well as some from the state-of-the-art that are commonly used in digital pathology. Code for reproducing the experimental results and figures is available as supplementary materials on GitHub (https://github.com/adfoucart/thesis-code-suppl).

Our main contributions to the state-of-the-art are the following. First, we studied the effects of imperfect annotations on deep learning algorithms and proposed adapted learning strategies to counteract adverse effects. Second, we analysed the behaviour of evaluation metrics and proposed guidelines to improve the choices made in the evaluation processes. Third, we demonstrated how the integration of interobserver variability into the evaluation process can provide better insights into the results of image analysis algorithms, and better leverage the annotations from multiple experts. Finally, we reviewed digital pathology challenges and found important shortcomings in their design choices and in their quality control and demonstrated the need for increased transparency.

The past decade has seen lots of changes in the field of computer vision, and in the field of pathology. The prospect of being able to include automated methods to the pathology pipeline, to help in the diagnostic process or in the search for reliable biomarkers that would make these diagnoses easier to obtain, appears increasingly likely. The adoption of such methods in clinical practice, however, requires a large amount of trust. Trust in the capacity of deep learning methods to navigate through the biases and limitations of manmade datasets, and trust that the apparent results of these methods reflect their true abilities when confronted with new examples in real-world settings. A strong movement to improve digital pathology competitions and benchmarks has been underway in the past few years. With this work, we hope to contribute to that movement, and to help bridge the gap between the clean setting of machine learning benchmarks, and the much messier setting of the clinical world.

# 1 Deep learning in computer vision

Deep learning has often been described as a **revolution**: in academic writing [25], [26] in books [27], [28] and in the media[3]. The state of the art of computer vision is filled with deep learning algorithms, and this revolution has certainly swept by the fields of medical imaging [29] and pathology [30]. It may therefore be surprising that there is hardly a good, clear, agreed upon definition of what "deep learning" actually is.

In the introduction to the "Deep Learning" book by Goodfellow, Bengio and Courville [31], the authors first propose deep learning as a "solution" to "solving the tasks that are easy for people to perform but hard for people to describe formally (...), like recognizing spoken words or faces in images. (...) This solution is to allow computers to learn from experience and understand the world in terms of a hierarchy of concepts, with each concept defined in terms of its relation to simpler concepts. (...) If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, we call this approach to AI **deep learning**."

Further in the introduction, they propose a more straightforward definition:

> *"Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones."* (Goodfellow et al [31], p8)

In these definitions, we see deep learning defined by the *kind of tasks* that it solves (intuitive but hard to formalize), by *the structure of its models* (as a graph with many layers), and by a more *semantic description* of these layers as representing *degrees of abstraction*.

In LeCun, Bengio & Hinton's Nature review of Deep Learning [25], it is defined as "representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level." This definition adds the notion that deep learning is a subset of *representation learning*. In representation learning, features are learned automatically from the raw data. Deep learning is therefore in this definition representation learning where the features are built on top of each other in increasing levels of abstraction.

In his own review for *Neural Networks*, Schmidhuber [32] inches towards a more formal definition based on the idea of "Credit Assignment Paths", which are "chains of possibly causal links" between the elements of the model, "e.g. from input through hidden to output layers in [Feedforward Neural Networks (FNNs)]". His conclusion on what is deep learning based on that definition, however, shows how arbitrary the distinction may be: "At which problem depth does Shallow Learning end, and Deep Learning begin? Discussions with DL experts have not yet yielded

[3]Forbes: https://www.forbes.com/sites/allbusiness/2018/10/20/machine-learning-artificial-intelligence-could-transform-business/
TechRepublic: https://www.techrepublic.com/article/the-deep-learning-revolution-how-understanding-the-brain-will-let-us-supercharge-ai/

a conclusive response to this question. Instead of committing myself to a precise answer, let me just define for the purposes of this overview: problems of depth > 10 require Very Deep Learning."

A revolution implies a sudden, fundamental paradigm shift. This begs the question: what, if anything, about deep learning can be said to constitute a revolution, and when did such a revolution happen?

This chapter aims to answer this question and is organized as follows:

First, we will review the historical roots of modern deep learning algorithms, from the early days of computer science and artificial intelligence.

From there, we will lay out the definitions that we will use for "deep learning" and its related nomenclature in the rest of this thesis.

Third, we will examine the main types of tasks that are commonly found in computer vision, as well as how the classical computer vision pipeline has evolved with the adoption of deep learning methods.

After the tasks, we will introduce the elements common to most deep learning solutions for solving them: the building blocks of deep learning.

Finally, we will look at how "deep learning models" are trained, and the process through which they are evaluated.

## 1.1 The roots of deep learning

It is probably not a controversial statement to say that "Deep Learning" is a concept that stands out in the overall field of artificial intelligence and machine learning, and became largely popular around the year 2012, particularly in computer vision. This is when the work of the Stanford-Google team of Quoc Le, Andrew Ng and their colleagues on unsupervised learning [33] drew a lot of media attention for creating a neural network that "taught itself to recognize cats." [4] It is also when the IDSIA team of Dan Cireșan, Ueili Meier and Jurgen Schmidhuber used an ensemble of convolutional networks to beat the state of the art on several major computer vision benchmarks [34], while the University of Toronto's team of Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton used their own network to win the prestigious ImageNet "Large Scale Visual Recognition Challenge" by a comfortable margin [35].

In an ImageNet post-challenge publication by Russakovsky et al., the organisers write [36]:

> *ILSVRC2012 was a turning point for large-scale object recognition, when large-scale deep neural networks entered the scene. [section 5.1 p.227]*
>
> *Following the success of the deep learning-based method in 2012, the vast majority of entries in 2013 used deep convolutional neural networks in their submission. [section 5.1 p232]*

[4] J. Markoff, New York Times, June 26th, 2012: https://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html

> *With the availability of so much training data (along with an efficient algorithmic implementation and GPU computing resources) it became possible to learn neural networks directly from the image data. [section 5.2 p233]*

If 2012 stands out as a clear mark for deep learning's popularity, the question of when, and who, started the deep learning revolution, is a lot more controversial. When the "Deep Learning" page was started on Wikipedia in 2011, it named "one of the earliest successful implementations"[5] as a 2006 publication by Hinton, Osindero and Teh introducing Deep Belief Networks [37]. As deep learning came under more scrutiny, however, some argued that deep learning dated from at least as early as 1980 and Fukushima's Neocognitron [38], or that it was simply a rebranding of neural networks.[6]

**Table 1.1. A timeline of milestones in the history of deep learning for computer vision.**

| Year | Milestone | Reference(s) |
|---|---|---|
| 1943 | Computational model of the neuron, simple units interconnected into a complex network. | McCulloch & Pitts, 1943 [39] |
| 1949 | Learning by changing the weights of the connections. | Hebb, 1949 [40] |
| 1957 | Definition of the neuron's main function based on a weighted sum of the inputs. | Rosenblatt, 1957 [41] |
| 1962 | Multiple layers of neurons for added complexity. Learning based on "error-correction" on supervised samples. | Rosenblatt, 1962 [42] |
| 1971 | Layer-by-layer training of multilayers perceptron-like networks | Ivakhnenko, 1971 [43] |
| 1980 | Local receptive fields, convolutional networks, shared weights, down-sampling. | Fukushima, 1980 [38] |
| 1974-1989 | Gradient descent with backpropagation for multilayers neural networks. | Werbos, 1974 & 1981 [44], [45]; Rumelhart, 1986 [46]; LeCun, 1989 [47] |
| 1991-1997 | Formalization of the "vanishing gradient" problem. Long Short-Term Memory for Recurrent Neural Networks. | Hochreiter, 1997 [48] |
| 2006 | Deep Belief Network, training layer-by-layer with an unsupervised pre-training. | Hinton, 2006 [37] |
| 2009 | Benefits of training deep neural networks on GPUs. | Raina, 2009 [49] |
| 2012 | Major wins in computer vision competitions. | Le, 2012 [33]; Cireşan, 2012 [34]; Krizhevsky, 2012 [35] |

[5] "Deep Learning" on Wikipedia, July 20th, 2011: https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=440437784
[6] "Deep Learning" on Wikipedia, December 23rd, 2015: https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=696509140

To get a better sense of what deep learning is and how it came to be, we have to go back to the start of artificial intelligence itself. A summarized timeline of the major milestones of deep learning's history is presented in Table 1.1, and we will expand on this timeline in the rest of this section.

### 1.1.1 Computational model of the neuron

In 1937, Alan Turing published "On computable numbers, with an application to the *Entscheidungsproblem*" [50]. In the history of computer science, this is mostly known for the introduction of what would later be known as the *universal Turing machine*, but it is first and foremost a mathematical treatise aiming at formally defining the mathematical idea of *computation*. This formalism would greatly influence the works of Warren McCulloch and Walter Pitts [51], who in 1943 created the first **computational model of the neuron** [39].

This model works thusly:

a) Neurons are connected through *excitatory* or *inhibitory* links.
b) A neuron is either *active* or *inactivate* (binary output)
c) If any of its *inhibitory inputs* is active, a neuron becomes inactive.
d) If only *excitatory inputs* are active, the neuron becomes active if the number of active connections is above a certain pre-determined threshold.

Following these simple rules, McCulloch & Pitts are able to form logic functions based on different configurations of neurons. They further show that such a network is equivalent to a Turing machine. These conclusions opened a wide range of possibilities, which would lead to the rise of *connectionism*, the study of systems whose complexity come from the interconnection of its comparatively simple elements.

In 1948, in a report[7] entitled "Intelligent Machinery" that would not be published until long after his death, Turing independently proposed another form of "**neural network**", which he called the "B-Type Unorganized Machine". In Turing's model, each neuron performs the NAND logic operation on two inputs (based on which any Boolean expression can be expressed [52]). From this very simple base, he combines neurons into *connection-modifier* elements which take a single input and can be set into an *interchange* mode (where it switches the input) or an *interrupt* mode (where it overrides the signal and always outputs "1"). The main novelty of this approach is that it provides a way to introduce a **learning mechanism**: Turing imagines a network that is initially randomly organised but is trained by switching the behaviour of those *connection-modifiers* so that some paths in the network are activated or inhibited. As the report went unpublished, however, it did not have the opportunity to have an impact on connectionist research.

### 1.1.2 The perceptron and connectionism

The next major step in the story comes in the late 1950s and early 1960s, with Frank Rosenblatt's publications on the "***perceptron***" [41], [42], [53]. The perceptron was "a hypothetical nervous system (…) designed to illustrate some of the fundamental properties of intelligent systems in general" [53]. It should be noted that, while the term "perceptron" is sometimes used today to refer to a single artificial neuron [54], the "perceptron" in Rosenblatt's publications is the entire

[7] Available online: https://www.npl.co.uk/getattachment/about-us/History/Famous-faces/Alan-Turing/80916595-Intelligent-Machinery.pdf?lang=en-GB (National Physical Laboratory)

network. Contrary to the very theoretical nature of McCulloch & Pitt's work, Rosenblatt's perceptron is also a physical machine. A 1957 technical report [41] describes the main components of the first version of the network: a *sensory system* (a set of photocells), a *response system* (lights which are lit up when their corresponding class is recognized), and an *association system* in between. Units in the sensory system have positive or negative connections to the association system. Each unit in the association system has a "fixed parameter" *T*, "the threshold value which corresponds to the algebraic sum of input pulses necessary to evoke an output" [41]. This output is binary: the association units therefore perform a thresholding operation based on the input signal (see Figure 1.1). These units are then further connected to units in the *response system*, which "are activated when the mean or net value of the signals received exceeds a critical level." The first perceptron had a single "association" layer, but Rosenblatt's 1962 book [42] contains experiments on "**multi-layer perceptrons**", containing several sequential association units. Different **learning techniques** are also discussed, including an "**error-corrective reinforcement system**" which changes the weights of the associations proportionnaly to the error made by the response unit, reproducing notions of learning introduced by Donald Hebb a few years earlier [40]. The conclusions from Rosenblatt's work were very optimistic on the capabilities of the perceptron. While "single-layer" perceptrons (with one association layer) showed poor generalization capabilities, he noted that "the addition of a [second association] layer (…) permits the solution of generalization problems" [42].

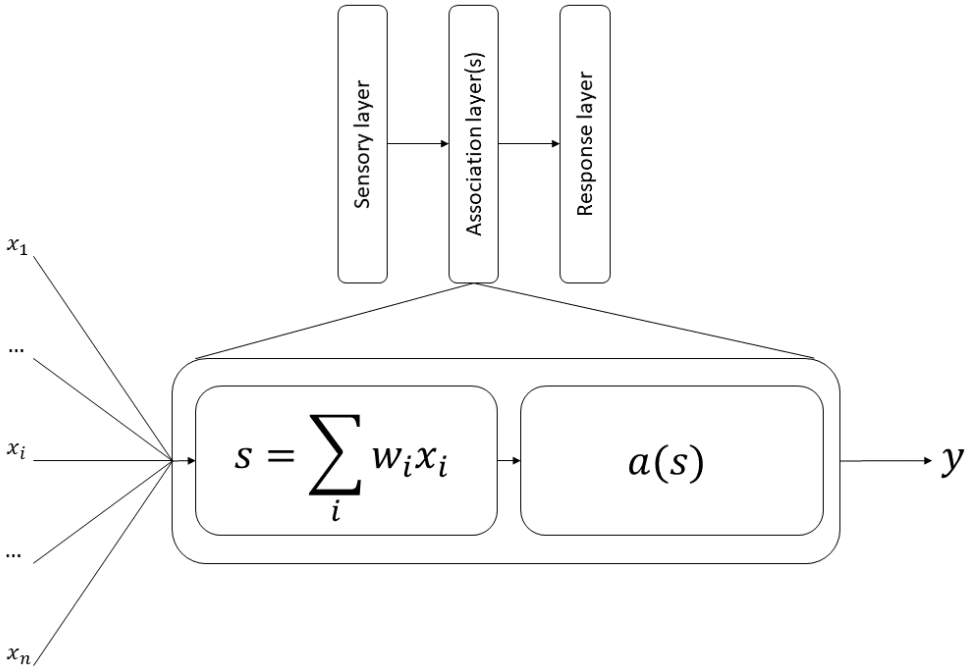



**Figure 1.1. Perceptron-like architecture. A sensory layer feeds into association layers, then into a response layer. In the association layers, each neuron performs a weighted sum of its inputs. This sum is then transformed by an *activation function*. In the original perceptron, the activation function was a step function, and the weights were either +1 (excitatory connection) or -1 (inhibitory connection), but later neural networks would relax these restrictions.**

In the late sixties, a big controversy arose around the perceptrons. The context of that controversy is well recapitulated in a *Social Studies of Science* paper by Mikel Olazaran [55]. The controversy opposed diverging views of Artificial Intelligence. The first one, exemplified by Rosenblatt, was trying to "build computational architectures bearing some resemblance to the brain's nets of neurons". Against this "**connectionist**" view were the proponents of "**symbolic AI**", where the decision rules were encoded in explicit symbolic rules and algorithms. One of the leading researchers in that school of thought was Marvin Minsky. In 1969, Minsky and Seymour Papert published *Perceptrons: An Introduction to Computational Geometry* [56]. They focused on those single-layer perceptrons only, and set out to demonstrate that the limitations of perceptrons could not be overcome, and that further research in the domain was doomed to fail. The controversy lays mostly in the fairness of Minsky and Papert's argument. As Olazaran notes, "strictly speaking, Minsky and Papert showed that single-layer nets, defined in a certain way, had some important limitations", yet their stated conclusions went a lot further than that. Their version of the perceptron also had further constraints, for instance on the limit of incoming connections to a neuron, that did not exist in Rosenblatt's work. Minsky and Papert's book is often considered as the main cause of a **rejection of neural networks** in the late sixties, and the emergence of symbolic AI as "the only AI paradigm" [55]… for a time, and mostly in the United States and western Europe.

Minsky and Papert's claim that multilayers perceptrons were essentially untrainable is disputed. Schmidhuber notes [32] that training mechanisms for multilayers perceptrons existed since the mid-sixties, and that "it seems surprising in hindsight that a book on the limitations of simple linear perceptrons with a single layer discouraged some researchers from further studying NNs". Perhaps the most important contributor to neural networks during that time was Alexey Ivakhnenko. While he mostly published in Soviet journal "Avtomatika"[8], Ivakhnenko also summarized some of his work in English, as in a 1971 publication [43] where he presents an eight-layers perceptron trained with his "**Group Method of Data Handling**" (GMDH). GMDH networks are trained layer by layer, and use a validation set to "regularize" the network and remove unnecessary ("harmful") units. Ivakhnenko also replaced the thresholding function used in the perceptron by a polynomial function.

### 1.1.3 Convolutional networks

A big limitation of multilayers perceptrons is that the number of trainable parameters can get very large very fast, as the increase the number of neurons and thus of potential connections. A key development, particularly for computer vision, comes in 1980 with Kunihiko Fukushima's *Neocognitron* [38]. The key ideas of the *Neocognitron* is to limit the connections in the network to a "**receptive field**", which is a locally constrained subset of the neurons from the previous layer, and to share the weights of the connections between neurons of the same layer. While Fukushima does not use that particular vocabulary, this is essentially what is known as a ***convolutional network***. A classical perceptron-like layer is fully-connected, or "dense". This means that each neuron of layer *n* is connected to each neuron of layer *n-1* with an independent weight. If $N_n$ is the number of neurons in layer $n$, a perceptron-like layer will therefore have $N_{n-1} \times N_n$ learnable parameters. In a convolutional layer, the neurons will be organized so that their outputs form ***feature maps***. Each feature map will be the result of the convolution between the input and a kernel (see Figure 1.2). If there are $k$ feature maps in layer $n-1$ and $l$ feature maps in layer $n$,

[8] See for instance a "bibliography of perceptron literature" by Rosenblatt (https://apps.dtic.mil/sti/pdfs/AD0420696.pdf#page=196, last accessed September 7th, 2021) in 1963.

and each convolutional kernel has a size of $w \times h$, there will therefore be $w \times h \times k \times l$ learnable parameters in the layer. The region in the input image which influences the value of a pixel of the feature maps of layer $n$ is called its **receptive field**. Each subsequent convolutional layer will therefore have a larger receptive field.

The other major contribution of Fukushima's network is the idea of combining these "convolutional" layers with **down-sampling layers**, which reduce the size of the feature maps and, as a consequence, further increases the receptive field relative to the input image (see Figure 1.3). This idea is directly related to models of visual cortex and introduce an invariance to translation which is completely absent in perceptron-like architectures. A 32x32 pixels greyscale image connected to a 16x16 feature map using dense connections would take 262.144 parameters, while a convolutional layer with a 5x5 kernel would use only 25 parameters. Despite this enormous reduction in the complexity of the model, however, the performances of convolutional networks are excellent, as a "dense" network would need to essentially learn all patterns in all positions of the image independently, while a convolutional layer only has to learn them once.

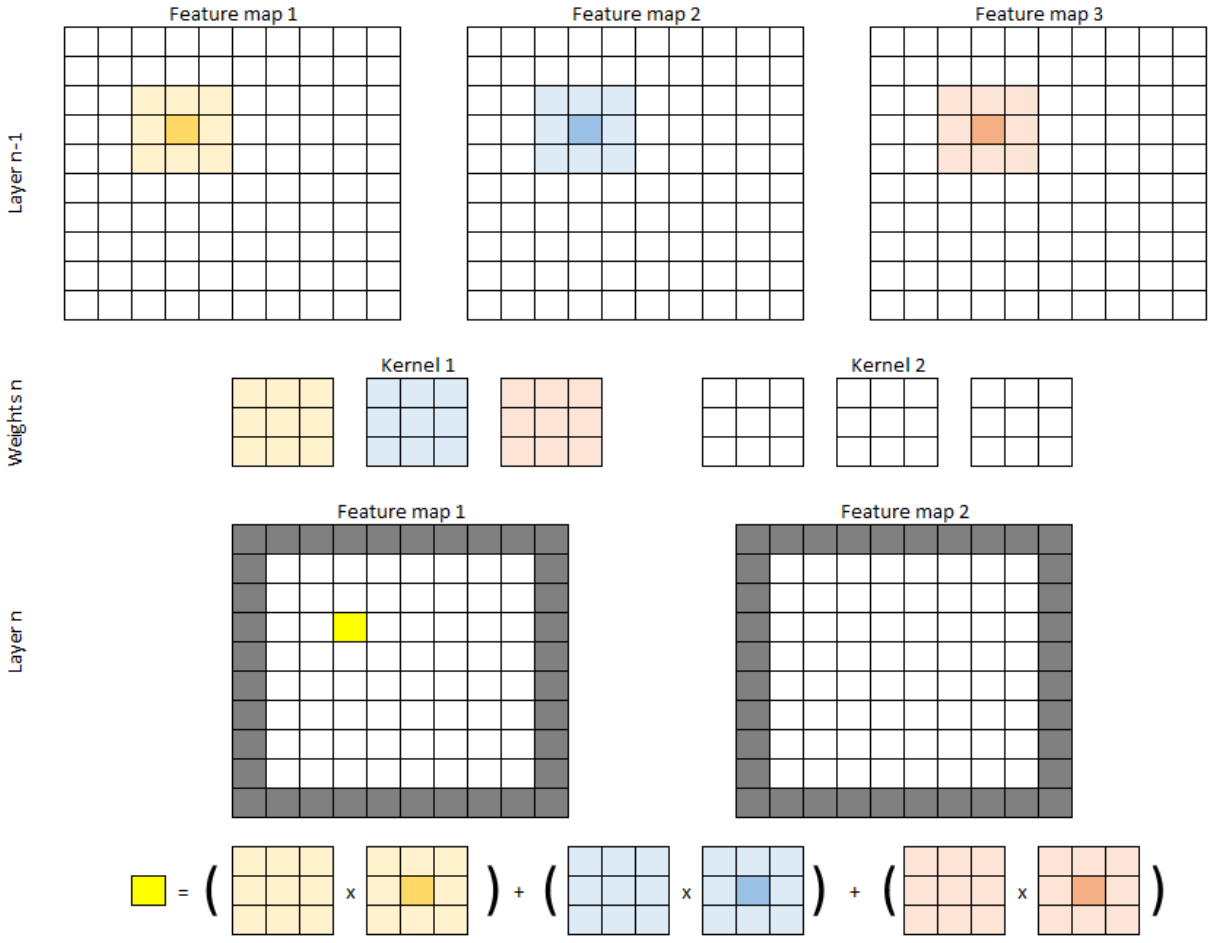



**Figure 1.2. Illustration of the convolution operation. The output of a feature map in layer $n$ is given by the convolution of the $l$ feature maps of layer $n-1$ and a kernel of size $w \times h \times l$, where in this example $w = h = l = 3$.**

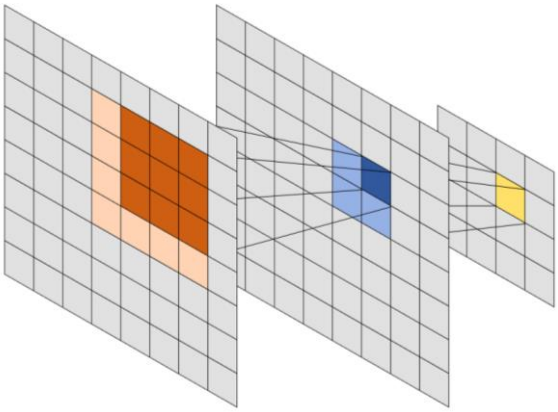


**Figure 1.3. Receptive field in a network with convolutions and down-sampling. The darker "neurons" (which, in a computer vision problem, would be directly related to the pixels layout of the input image) in the first and second feature maps are connected through the convolution operation (here, with a 3x3 kernel), while the last feature map is the result of a 2x2 down-sampling. After the 3x3 convolution and the 2x2 down-sampling, the "receptive field" of the single yellow neuron in the last layer is the 4x4 coloured region in the first.**

Convolutional networks are thus particularly well-adapted to signals where there is a meaningful spatial relationship between the input elements, such as images. The "feature maps" produced by convolutional layers encode this spatial relationship in their structure. A layer $n$ that has 10 feature maps with dimensions of 25x25px thus encodes the response to 10 different kernels and the localisation of these responses.

### 1.1.4 Backpropagation and stochastic gradient descent

With the *Neocognitron*, we are getting a lot closer to something that would be recognized as a "deep neural network" today. If the "architecture" of the network is recognizable, the learning mechanism did not yet include what would become the standard for neural networks: **backpropagation**. Backpropagation appears to have been "discovered" (or at least applied to neural networks) independently by several researchers through the seventies and eighties, with the most often cited sources nowadays being Paul Werbos' 1974 thesis [57] and David Rumelhart's 1986 *Nature* publication alongside Hinton & Williams [58]. The history of the "invention" of backpropagation is subject to some amount of controversy. Jürgen Schmidhuber argued repeatedly, in his review [59] and in his personal online publications[9], that the credit for efficient backpropagation should go to the 1970 thesis of Linnainmaa, and that LeCun, Hinton and Bengio repeatedly failed to accurately reference his work, or the importance of Werbos' work, in a bid to claim credit for themselves. Whether or not there are deontological problems with Rumelhart, Hinton & Williams' work, it is clear that they wildly popularized backpropagation for neural networks. It should also be noted that neither Werbos' work, nor especially Linnainmaa's (who published his thesis in Finnish) had much visibility at the time, and that it is entirely plausible that Rumelhart and Hinton had not read any of their contributions. The ongoing feud between Schmidhuber and Hinton, spanning their academic works, social network publications[10]

[9] See https://people.idsia.ch/~juergen/deep-learning-conspiracy.html or https://people.idsia.ch/~juergen/critique-turing-award-bengio-hinton-lecun.html

[10] https://www.reddit.com/r/MachineLearning/comments/g5ali0/d_schmidhuber_critique_of_honda_prize_for_dr/fo8rew9/

and blog post responses[11] demonstrates how difficult it is to get an accurate, unbiased historical record.

Backpropagation is a way to implement **gradient descent** to optimize the weights of a multilayers neural network in order to **minimize** an error function (usually called the **loss function** in modern terminology), as illustrated in Figure 1.4.

Let $\mathbf{t}$ be the *target output* of a network, $\mathbf{x}$ an input vector, and $\mathbf{y} = f(\mathbf{x}, \mathbf{w})$ the output of the network with $\mathbf{w}$ representing all the connection weights within the network. The loss, in general, is a function of $\mathbf{t}$ and $\mathbf{y}$ (which could be vectors or scalars depending on the problem):

$$Loss = L(\mathbf{t}, \mathbf{y})$$

With gradient descent, the goal is to compute the gradient of the loss function with regards to the weights:

$$\frac{\partial L}{\partial \mathbf{w}}$$

So that the weights can be updated using a certain **learning rate** $\eta$:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial L}{\partial \mathbf{w}}$$

The problem that backpropagation aims to solve is that $\frac{\partial L}{\partial \mathbf{w}}$ cannot be computed directly, and so gradient descent alone cannot be used to update all the weights of the network. Backpropagation starts from the last layer of the network:

$$y_{N,i} = a(s_{N,i}) = a(\sum_j w_{N,ij}\, y_{N-1,j})$$

Where $N$ is the number of layers in the network, $y_{n,i}$ is the output of the $i$-th neuron of the $n$-th layer, and $w_{n,ij}$ is the weight of the connection between the $j$-th neuron of the $(n-1)$-th layer and the $i$-th neuron of the $n$-th layer. It should be noted that in many formulations of the operations happening at the neuron level, there will be an added *bias* term to the summation. However, the complete formulation $\sum_j w_{N,ij}\, y_{N-1,j} + b_{N,i}$ is equivalent to the formulation presented above if we simply consider that the bias is associated with a constant input of 1.

The first step is to compute the derivative of $L$ with respect to $s_{N,i}$, which using the chain-rule of partial derivatives gives us:

$$\frac{\partial L}{\partial s_{N,i}} = \frac{\partial L}{\partial y_{N,i}} \frac{\partial y_{N,i}}{\partial s_{N,i}} = \frac{\partial L}{\partial y_{N,i}} a'(s_{N,i})$$

With $a'$ the total derivative of the activation function.

If the loss function was chosen to be derivable with respect to $y_{N,i}$, all the terms of this equation can be directly computed. For instance, if the loss function is the mean squared error:

[11] https://people.idsia.ch//~juergen/critique-honda-prize-hinton.html#reply

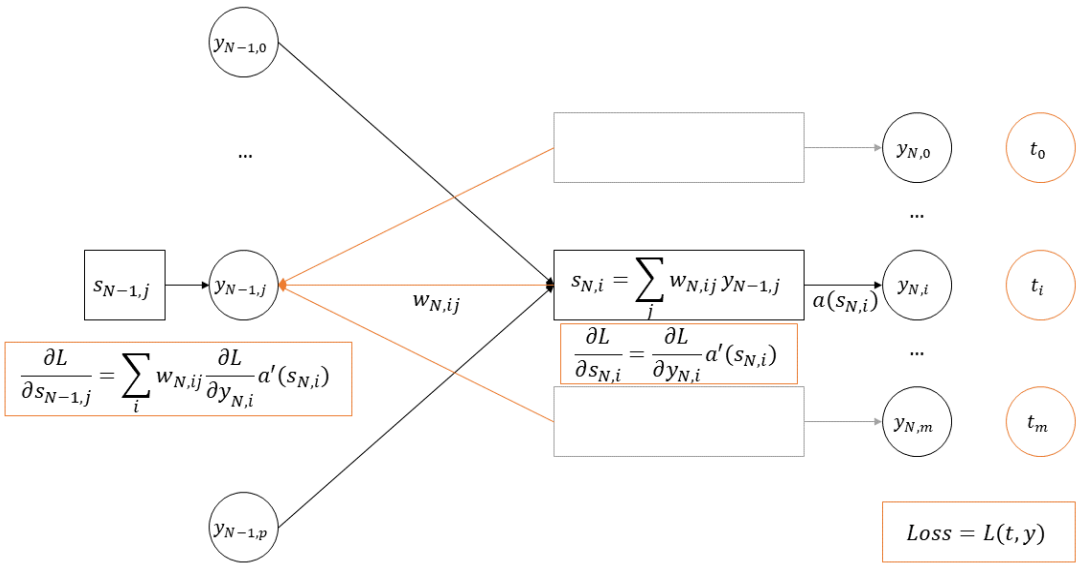



**Figure 1.4. Principle of the backpropagation of errors for gradient descent in a neural network. The loss $L$ is computed from the outputs $y_{N,i}$ and the targets $t_i$. The gradient in the output layers $\frac{\partial L}{\partial s_{N,i}}$ are then computed, then propagated to the previous layer proportionally to the weight of the connections and to the derivative of the activation function.**

$$L = \frac{1}{m}\sum_i (y_{N,i} - t_i)^2$$

With $m$ the number of output neurons of the network, then $\frac{\partial L}{\partial y_{N,i}} = \frac{2}{m}\sum_i (y_{N,i} - t_i)$.

The next step is to propagate the error to the previous layers, using the rule:

$$\frac{\partial L}{\partial s_{n,j}} = \sum_i w_{n+1,ij} \frac{\partial L}{\partial s_{n+1,i}} = \sum_i w_{n+1,ij} \frac{\partial L}{\partial y_{n+1,i}} a'(s_{n+1,i})$$

This allows for the computation of $\frac{\partial L}{\partial s_{n,i}}$ in every neuron of the network. From there, the gradient relative to the weights can also be computed:

$$\frac{\partial L}{\partial w_{n,ij}} = \frac{\partial L}{\partial s_{n,j}} \frac{\partial s_{n,j}}{\partial w_{n,ij}} = \frac{\partial L}{\partial s_{n,j}} y_{n-1,i}$$

Each weight can then be updated with:

$$w_{n,ij} \leftarrow w_{n,ij} - \frac{\partial L}{\partial w_{n,ij}}$$

While the error can be computed on the entire training set before updating the weights, a **stochastic gradient descent** method was quickly preferred, updating the weights based on a single example at a time (or a small "mini-batch"), for better convergence speed. This was notably discussed by Yann LeCun and several colleagues in 1989, presenting a convolutional network using backpropagation to recognize handwritten zip codes for the U.S. Postal Service [47].

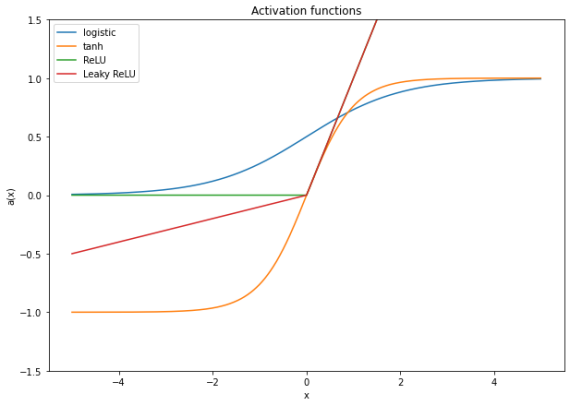



**Figure 1.5. Example of four different activation functions: two "sigmoid" functions (logistic and tanh), with the more recent ReLU and Leaky ReLU functions.**

LeCun's publications on handwritten digits and, later, document recognition [47], [60], [61] may be a turning point in neural networks, not so much because of the novelty of any specific part of his methods, but because of its very *practical* aspect. They showed that convolutional neural networks, trained with stochastic gradient descent and backpropagation, using sigmoid activation functions, could be trained to solve practical, real-world computer vision problems. LeCun's 1998 "LeNet-5" has all the characteristics of a standard classification architecture, as seen in modern deep learning models: a succession of convolution and down-sampling (with max-pooling) layers which increase the number of feature maps while decreasing their size, followed by a few "dense", perceptron-like layers with decreasing number of neurons until the last output layer, which contains as many neurons as there are classes to discriminate.

### 1.1.5 The vanishing gradient and "deep" improvements

The main issue that neural networks faced in order to go "deeper" was the "**vanishing gradient**" problem, formally described by Sepp Hochreiter in his 1991 Ph.D. thesis in German and published by Hochreiter and Schmidhuber in 1997 [48]. The vanishing gradient problem is a direct consequence of the backpropagation algorithm. As the "gradients" $\frac{\partial L}{\partial w}$ are propagated through the network, the derivatives of each layer's activation functions are multiplied together. The most common activation functions at the time were sigmoid functions such as the logistic function or the hyperbolic tangent (shown in Figure 1.5). The derivatives of these functions have a small peak in the centre, and quickly fall close to zero on both sides. This has a catastrophic effect on learning, as the multiplication of those very small gradients lead to some weights in the network being essentially incapable of being updated.

Some improvements were proposed to address this problem over the next years. Hochreiter and Schmidhuber's Long Short-Term Memory for recurrent neural networks was explicitly designed to avoid vanishing gradients [48]. Changing the activation function to the Rectified Linear Units (ReLU) [62] and its variations such as Leaky ReLU [63] (see Figure 1.5), which have a much larger "active" region (in the sense of a range of input with a significantly larger than zero output), also proved to be very effective. The "old" idea of training a network layer-by-layer was also one of the key ideas of Hinton's 2006 "deep belief" article [37], which also used an "unsupervised" pre-training step. As we have mentioned at the beginning of this section, this publication is when the terms "deep learning" and "deep neural network" start to gain traction. "Short-skip" and "long-skip" connections, such as in Highway Networks [64], ResNet [65] and U-Net [66], which create

bypasses through which the gradients can flow more easily to early layers also helped to overcome the limitations of deeper neural networks.

### 1.1.6 The Deep Learning revolution

As we mentioned at the beginning of this story, a key year for the deep learning revolution was 2012. Yet none of the networks or learning methods that cemented themselves as the state of the art of computer vision at that time were very different from anything that existed before. In fact, as we have seen through this historical review, all the major elements of "deep learning" publications slowly evolved in incremental steps, building on top of each other.

Is "deep learning" therefore really a buzzword, a rebranding of neural networks? There is no doubt that some form of revolution happened, but it was obviously not a revolution in the core theory of neural networks. Two major contributing events, however, can be traced to the same period. The first is the start of the "Big Data" era [67]. All these 2012 works relied on datasets of sizes and diversities which were simply unimaginable a few years earlier. ImageNet and Google's databases in particular use millions of diverse colour images. MNIST, LeCun's 1998 database of handwritten digits, had by comparison 60.000 small, aligned, black-and-white images, which were resized to 28x28px (with interpolated values, so that the final images are greyscale [47]).

The second event was the massive development of general computing on graphical processing units (GPU). In 2006, the release of CUDA by NVIDIA made parallel processing on GPUs a lot more accessible by allowing developers to use C to program GPUs [68]. As neural networks are massively parallelizable, it was quickly shown that training neural networks on GPUs causes massive speedups in training time, thus allowing larger networks to be trained [49].

The deep learning revolution therefore appears to be more of a practical than theoretical one. As "Big Data" provided the amount of training samples required for training networks with large number of parameters without massive overfitting and GPUs made the training times more reasonable, large neural networks suddenly became something that could realistically be done outside of hyper specialized research groups with access to supercomputers.

This, in turn, led to the release of software tools designed to make the building and training of large neural networks easier and even more accessible. Starting from 2009, Theano was developed at the University of Montréal[12], providing a Python framework for fast tensor operations on GPUs [69]. Other frameworks such as Caffe, developed by Berkeley AI Research and released in 2014 [70]; TensorFlow, released by Google in 2015 [71]; or Facebook's PyTorch, presented at NeurIPS 2019 [72]; contributed to the widespread adoption of deep neural networks outside of pure machine learning research and into practical applications.

This practicality also shows that the study of neural network did not only evolve in its theory, but also in the objectives of its actors. Early connectionists were overtly interested in using *machine intelligence* as a proxy for better understanding *human intelligence*, and the *process of learning*. We then see a movement towards studying *machine intelligence* by itself: artificial neural networks did not necessarily have to resemble animal neural networks anymore, the main concern was that they were capable of learning ever more complex functions. Finally, we get to the modern vision of task-oriented deep learning, exemplified by the importance that "Grand Challenges", benchmarks and competitions have taken in the deep learning era.

[12] https://github.com/Theano/Theano/blob/master/HISTORY.txt

## 1.2 Defining deep learning

From this historical perspective, it is clear that the concepts of artificial intelligence and its subdivisions have fluctuated over time. Definitions of "artificial intelligence", "machine learning", "deep learning" and all related concepts can vary in the scope of what they include. In this section, we will provide and justify the definitions that will be used in the rest of this thesis.

> **Artificial intelligence** is the study of systems which take actions or make decisions based on their perception of their environment.

This is a very broad definition. A key part is that artificial intelligence systems do not exist in a vacuum: they perceive and act on the "real world", and both the perception mechanism and the action/decision mechanism are a part of that whole system. In computer vision, the "perception" would typically be some form of image acquisition, and the "action" could be, for instance, the generation of semantic information extracted from the image.

> **Machine learning** is the study of artificial intelligence systems that are capable of improving their performance based on their experience.

This implies two very important aspects of a machine learning agents: they must include parameters that can be changed as a response to the observations, and they must include a metric for measuring its performance, as well as a mechanism for updating its parameters. These are determined by the machine learning model and the machine learning algorithm:

> A **machine learning model** describes the relationship between the input of the system (the "perception") and its output (the "action"), as well as the parameters of that relationship. When the parameters are set, the model is said to be **trained**.
>
> A **machine learning algorithm** is a method used to find the best parameters of the model according to certain criteria.

Machine learning problems are often separated into "supervised" and "unsupervised" problems:

> A **supervised** problem is one where each sample in the training set is associated with a known target value (the "label").
>
> In an **unsupervised** problem, the labels in the training set are unknown, and the model has to learn from the distribution of the data.

A useful distinction can also be made between the "parameters" and the "hyper-parameters" of a model.

> The **parameters** are internal to the model and are set through the learning algorithm.
>
> The **hyper-parameters** are external to the model and are often set through the validation process. They are typically related to design choices in the model, like the maximum depth of a decision tree, or the size of the layers in a convolutional neural network.

It is often the case that a machine learning model can be split into two parts: feature extraction and decision function.

> **Features extraction** is a processing of the raw input of the system to project it into a new space of variables (the "features space") where the problem is easier to solve.

> The **decision function** is a mathematical relationship between the input in features space and the output of the system.

A further distinction is made in the features extraction step depending on how those features are determined:

> **Handcrafted features** are chosen and defined by a human expert based on their experience and their observation of the data.
>
> **Learned features** are derived from the data by a machine learning algorithm. The subset of machine learning that include learned features is called **representation learning**.

We can now finally arrive at the elements that separate so-called "deep" learning from the rest of machine learning:

> **Deep learning** is the study of machine learning models where features are represented in a layered structure of increasing abstraction, and of machine learning algorithms that learn the features representation and the decision function as a single step.

From this definition, we get the key aspects of what would generally be described as a "deep learning" system today.

a) **Layers of abstraction**: low-level features are extracted from the raw input of the system, and higher-level features are learned by combining these low-level features.
b) **Learning features and decision in one step**: the main difficulties that prevented multi-layered neural networks to learn efficiently in the past were generally related to the ability to efficiently learn the layers further from the output. Solutions to that problem typically involved learning layer-by-layer, but that effectively means learning the features representation separately from the decision function, which falls into the broad category of "representation learning" without being really specific to deep learning.

This definition also implies that the answer to the question of whether, for instance, perceptrons of the past belong to the "deep learning" category actually depends on how they were used and trained. The original, single-layer perceptron of Rosenblatt was trained directly on the raw input, and learned "features" and "decision" in one step, but it clearly did not include layers of abstraction. A multilayers perceptron could be included in the "deep learning" category if applied directly on the raw data, but not if it relied on handcrafted features. The perceptron-like networks of Ivakhnenko, for instance, were trained layer-by-layer and relied on handcrafted features in the examples he used [43], and would not be included either. Fukushima's Neocognitron [38] is much closer to the definition, as it clearly demonstrates layers of abstraction and learn directly from pixel data, yet the images used are synthetic, binary representations of numbers and letters, and the network is trained in an unsupervised way and does not include any "action" step. The letters or numbers are not classified: it is just observed that the network has learned different responses to different types of patterns. LeCun's handwritten digits recognition [60] includes a big pre-processing step to isolate the digits and standardize their presentation, yet its inputs are still close enough to the raw data that it is probably reasonable to include it into the "deep learning" definition. As a direct predecessor to the highly publicized AlexNet [35], it is also unsurprising that it represents one of the common archetypes of modern deep neural networks for computer vision: the "image classification" network.

This definition of deep learning does not necessarily require the model to be any type of neural network. However, in practice, most if not all model used in modern deep learning are neural networks, making "deep learning model" and "deep neural networks" functionally synonyms. It is therefore useful here to also introduce a few definitions relative to neural networks.

> An **artificial neural network** (generally just shortened to "neural network" in the context of machine learning) is a model that can be decomposed into a graph of interconnected units (the "neurons") which perform a weighted sum of their inputs, with an activation function that may introduce non-linearities in the overall function. The structure of the graph is often referred to as the **architecture** of the network.

While it is in theory possible to explicitly model every single connection and every single neuron in a network, it is generally more practical to use an organization in layers:

> A **layer** in a neural network is a group of neurons that share a common set of inputs and outputs and no "internal" connections (as in: no neuron in the layer is connected to another neuron in the same layer). Neurons in a layer will also typically share the same activation function.
>
> A **connection** exists between layers if the output of some neurons of one layer are used as input by some neurons of the other.

Finally, it is sometimes useful to be able to refer to multiple functionally related layers at the same time. We will refer to those groups of layers as blocks:

> A **block** is a group of layers in a neural network that perform a particular function together within the network.

## 1.3 Computer vision tasks

A complete taxonomy of computer vision tasks is outside of the scope of this thesis, but we will in this section propose a useful framework for categorizing the different types of tasks that will be relevant to the rest of this work.

First, what do we mean by a "Task"? To quote Goodfellow et al: "Machine learning tasks are usually described in terms of how the machine learning system should process an *example*."[31] In other words, a Task is defined by the type of *output* that is expected, and the kind of *input* that is being processed. In computer vision, the *input* will typically be an *image*, most likely represented as a matrix containing the pixel values. It should be noted that with this definition we are not considering differences in the *learning process* (for instance: supervised or unsupervised learning), which can lead to completely different categorizations of machine learning algorithms.

One big distinction that can be made relates to the nature of the output value. In ***classification*** tasks, the desired output comes from a discrete, finite set of possibilities. In ***regression*** tasks, the output consists in a set of continuous variables [73].

For most of the tasks that we are concerned with in this work, however, a more useful categorization is related to the nature of the *target*: the images, objects, or pixels.

> In a **classification task**, the goal is to assign a *class* to an entire *image*. The output of such a task will typically be a single class, often supported by a vector of class probabilities. In some cases, the categories can be *ordered*.

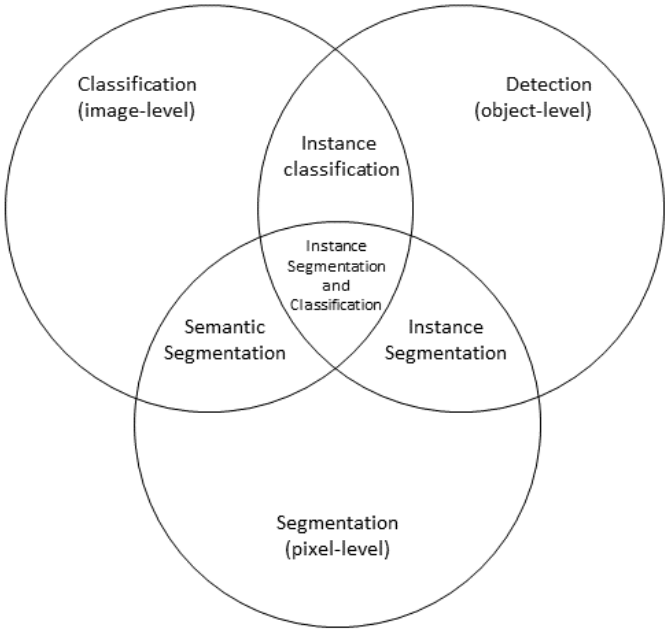



**Figure 1.6. Proposed categorization of computer vision tasks.**

In a **detection task**, the goal is to *detect the presence* of instances of a target object. The output of such a task will typically take the form of a set of bounding boxes within which the instances have been found.

In a **segmentation task**, the goal is to separate the "foreground" pixels (which are parts of "objects of interest") from the "background" pixels (everything else). The output will take the form of a binary mask, often supported by a pixel probability "heat map".

These three basic tasks can be combined with each other to form more complex tasks:

An **instance classification task** combines object detection with classification (assign a class to each of the detected objects). The output will typically be a set of bounding boxes with corresponding class values (supported by vectors of class probabilities).

An **instance segmentation task** combines object detection with segmentation (determine which pixels are part of which object instance). The output will be a pixel "label map", where each pixel is assigned to a label representing the object that it belongs to.

A **semantic segmentation task** combines pixel segmentation with classification (assign a class to each of the pixels). The output will be a pixel "class map", often supported by a "class probabilities" map where each pixel is associated to a vector of class probabilities.

An **instance segmentation and classification** task combines all three basic tasks, so that each object is detected, assigned to a class, and associated to a pixel mask. The output will be a class map alongside a label map, or a label map with an associated "per-label" class prediction.

It should be noted that, while regression tasks are not explicitly represented in that categorization, they will generally be very close to one of these (except that class values will be replaced by continuous values). For instance, a counterpart to "semantic segmentation" could be "image generation", where for each pixel a colour value is predicted. This categorization is also not

exhaustive. Image registration, for instance, does not really fit into this taxonomy. These however cover the tasks where deep learning methods have been most relevant in digital pathology and will be where we will focus our attention in this work. For the same reason, we will generally limit ourselves to supervised methods, and only consider "unsupervised learning" as a pre-training step (as in representation learning) in the context of a supervised task.

## 1.4 The deep learning pipeline

While the nature of the different tasks will certainly also influence the design of the learning process, most machine learning solutions to computer vision problems will follow a relatively similar pipeline, illustrated in Figure 1.7. Deep learning solutions follow the same process. Each of the steps in this pipeline come with their own challenges and are largely application dependent. In this section, we will briefly explain the elements that are commonly found in computer vision pipelines. A summary of the hyper-parameters that are related to these different steps is presented in Table 1.2.

### 1.4.1 Data acquisition

The data acquisition step will obviously be largely concerned with the **technical aspects** of the acquisition. For instance, in digital pathology, the variety of protocols that were applied for processing and staining the tissue samples and the scanners used to acquire the images will influence the final result. It is also concerned with the **selection** of the samples. A machine learning algorithm will learn to use the data that is provided in the training set. If this data is not representative of the domain of applicability, the algorithm will most certainly fail.

Another closely related question is the constitution of the *training set* and of the *test set* (and/or the *validation set*). The training set is the data that is used to determine the learnable parameters of the model. The validation set is used to determine the hyper-parameters of the model and of the algorithm. The test set is used to evaluate the model once all parameters and hyper-parameters have been fixed. Improper handling of the data can lead to bad interpretation of the results. As a general rule, the test set should be as "independent" from the training and validation set as possible: different acquisition hardware, different sample source (e.g., individual patients in a medical application), etc. More independence means that the test set evaluation will more accurately represent the generalization capabilities of the algorithm.

Equally important to the supervised learning process as the image acquisition is the **annotation** of the data. The ideal situation for a deep learning algorithm is to have a perfectly supervised dataset, meaning that to each data sample is attached a "ground truth", which is the expected output of the trained algorithm for that sample. This optimal situation is rarely found, particularly in digital pathology datasets. This will be discussed at length in the rest of this thesis.


**Figure 1.7. Machine learning pipeline.**

**Table 1.2. Main hyper-parameters of a deep learning pipeline, excluding the hyper-parameters related to the model itself.**

| Hyper-parameter | Choices and possibilities |
|---|---|
| Pre-processing steps | Resizing, tiling, colour space change, de-noising… |
| Data augmentation steps | Affine/elastic transforms, noise, blur, colour transforms, GANs… |
| Mini-batch size | Trade-off between memory availability, training speed (larger size = quicker) and end result quality (smaller size may be better). |
| Initializer | Xavier [74], He [75]… |
| Optimizer | Adam [76], RMSProp [77], ADADELTA [78]… |
| Loss function | Cross-entropy, mean-squared error, customized losses… |
| Stopping criterion | Fixed number of epochs or based on improvement on a validation set. |
| Post-processing steps | Image reconstruction, resizing, de-noising, labelling… |
| Evaluation process | Metric, aggregation method, statistical tests… |

### 1.4.2 Data preparation

Even "deep learning" algorithms can usually not simply take as input the whole training set. Some pre-processing will typically be necessary. The pre-processing that needs to be applied will again depend on the application. Two distinct types of pre-processing are very common in computer vision application: *normative pre-processing* and *data augmentation*.

**Normative pre-processing** includes steps designed to restrict the domain of the input space. For instance, many network architectures require a fixed-size input. Pre-processing steps to achieve that include resizing or tiling of the input images. It may also be interesting to change the colour space (for instance, moving from RGB to HSV, or to a colour space more relevant to the application). A normalization of the dataset can also be done at this stage, or a rescaling of the input values to a predetermined range (for instance, -1 to +1).

**Data augmentation** has a very different role. It seeks to increase the size and diversity of the dataset by generating artificial examples based on the available samples. Instead of restricting the domain of the input space, the goal is in this case to fill this domain with as many (realistic) examples as possible. Common data augmentation steps include affine transforms, blur, noise, colour transforms, elastic transforms, etc. GANs can also be used to generate new examples (although this solution sometimes just deports the problem of finding real examples to the training of the GAN itself).

### 1.4.3 Training of a deep learning model

While training the model, data samples are generally presented in **batches** (often referred to as "mini-batches"), with the parameters of the network being updated after each batch. The **batch size** is a design choice that is often constrained by the available hardware, as using larger batches require more memory usage in the GPU. At the extreme end of the spectrum, training can be done

with batches consisting of a single example. This is for instance the case in stochastic gradient descent. The batch size can influence both the speed of the training process, and the quality of the results. As a general rule, larger mini-batches tend to train faster, but lead to worse generalization [79]–[81].

When training neural networks from scratch, another important aspect is the **initialization** of the parameters. The initial parameters of the connections are chosen randomly, but the choice of distribution from which these random weights are drawn is important, as bad initialization can lead to a higher risk of vanishing or exploding gradients [82]. Usual choices are to use either a normal or a uniform distribution, with a mean of 0 and a variance that depends on the number of neurons in the layers on both side of the connection. The two most common variants are Xavier initialization [74] and He initialization [75]. Xavier initialization was initially designed for networks using the *tanh* activation function, while He initialization was designed to adapt it to networks using the "rectified linear units" (ReLU) functions.

Another important aspect of the learning algorithm is the choice of the **optimizer**. All commonly used optimization techniques for deep neural networks are based on gradient descent and backpropagation. The main steps of this optimization process are:

a) Computing a *loss* $L$ on a mini-batch, which measures the error that the model currently makes on the training data.
b) Use the rules of *backpropagation* to compute the contributions of each parameter $w$ of the model to that loss: $\frac{\partial L}{\partial w}$.
c) Update the parameters depending on their contribution to the loss and to a learning rate $\eta$: $w \leftarrow w - \eta \frac{\partial L}{\partial w}$.

The learning rate can be fixed through the learning process or can be adaptive. More than a hundred different optimization methods have been proposed for training deep neural networks, but no clear rule has emerged to determine which one should be used for a particular problem, with some research suggesting that the difference between methods is often less important than the difference that can be observer within the same method with different hyper-parameters or even with different random seeds [83].

The result of those optimizers will also obviously depend on the **loss function** being used. The default, general purpose loss functions are the *cross-entropy* ($CE$) for classification problems and the *mean-squared error* ($MSE$) for regression problems, defined as:

$$L_{CE}(\mathbf{y}, \mathbf{t}) = -\frac{1}{\mathrm{n}} \sum_{i=1}^{n} \sum_{\mathrm{c}=1}^{\mathrm{m}} t_{ic} \log\left(y_{ic}\right)$$

$$L_{MSE}(\mathbf{y}, \mathbf{t}) = \frac{1}{n} \sum_{i=1}^{n} (t_i - y_i)^2$$

With $n$ the number of samples, $m$ the number of classes, $t_{ic}$ the binary target value of sample $i$ for class $c$ (for classification problems), $t_i$ the scalar target value of sample $i$ (for regression problems), and with $y_{ic}$ and $y_i$ similarly defined for the output of the model.

More complex loss functions can be used to address problems such as class imbalance [84] or noisy labels [85].

Finally, another choice that needs to be made is the **stopping criterion** for the optimization process. The simplest rule is to run the optimizer for a fixed number of epochs (one epoch meaning that all training samples have been processed once), but this can easily lead to over-fitting. Early stopping is a common choice to avoid this problem, with a stopping rule that uses the validation set to check if the learning has failed to improve by more than a certain threshold value for more than a certain number of epochs (the "*patience*").

### 1.4.4 Post-processing

After the optimization process is complete, the output of the model may not be completely usable *as such* yet. For instance, in a segmentation problem, if the pre-processing step included a tiling of the image, the output of the model will also be tiled, whereas the true target output for the whole pipeline is a segmentation of the full image. **Image reconstruction** from the segmented tiles may therefore be necessary as a post-processing step. If the tiling was done with some overlap, the reconstruction rule is not trivial, as individual pixels will have different predicted values when they are included in multiple tiles. Their final output value may be determined as the *average* of the predicted probability values, but other rules may also be applied such as the *maximum* value, or a *weighted average* with the weight depending on the proximity to the centre of the tile (as predictions will tend to be better further from the borders).

Other **post-processing** steps may include some de-noising (for instance, using morphological operations to remove isolated predicted pixels and to smooth the borders of segmented objects), or labelling (in the case of instance segmentation).

### 1.4.5 Evaluation

The evaluation of a deep learning algorithm is a very important aspect of the process, as it is used not only to determine which model and which set of hyper-parameters are best for solving a particular task, but also to validate if the end results are adequate for a certain purpose (such as a clinical application).

We have previously mentioned the selection of the *test set* as an important aspect of the data acquisition process. It will of course also influence the quality of the evaluation. The most obvious aspect of the evaluation is the **metric** used. The choice of a good metric is important as different metrics will be associated with different biases.

Different **metric aggregation** methods can be used. The metric can be computed on every image of the test set, then averaged using the mean or the median. It may also be possible (for instance in segmentation problems) to compute the metric on the whole test set at once (giving more impact to larger images), or to first aggregate according to some sub-groups (for instance: per-class in a multi-class problem, per-patient, etc.)

In addition to this "averaged" value, it is often useful to look at the **distribution** of the metric over the test set. This allows for a more robust analysis, using for instance statistical tests to assess the significance of the results when comparing different algorithms, or different sets of hyper-parameters.

## 1.5 Deep model architectures

> *If your task is similar to another task that has been studied extensively, you will probably do well by first copying the model and algorithm that is already known to perform best on the previously studied task.*
> *-- Goodfellow, Bengio & Courville; Deep Learning, 2016* [31]

There is an infinite number of possibilities when defining the architecture of a deep neural network. Many practically used networks, however, share a lot of common characteristics and are made from the same building blocks. In this section, we will look at these building blocks, and how they are integrated into the architectural "archetypes" used in many computer vision tasks. We will not look outside of computer vision, as these are the type of applications most relevant to digital pathology. We will therefore not cover recurrent neural networks or LSTMs [48], commonly found in domains with sequential signals such as Natural Language Processing.

We can look at deep learning architectures at different levels. At the "macro" level, we have the overall "shape" of the network, which is mostly dependent on the inputs and outputs of the system (in other words: on the *task*). At the "micro" level, we have smaller design choices in which specific layers are used, and how exactly they are connected.

We will first look at the most common types of **layers** used in deep neural networks, then we will move on to the **macro-architectures** in relation to the tasks defined previously, and finally look at some of the **micro-architectural choices** and how they can affect the learning process and the performances of the model.

### 1.5.1 Layers

**Dense** (or "fully-connected") layers are the simplest layers from a conceptual point of view. The *n-th* layer of a network is dense if every neuron in layer *n-1* is connected to every neuron in layer *n*. Mathematically, dense layers are very easy to model using matrix notations:

$$L_n = A(W_n L_{n-1})$$

Where $L_n$ is a vector containing all the outputs of the *n-th* layer, $W_n$ is a matrix containing all the weights between each neuron of the *n-th* layer and each neuron of the previous layer, and A(X) is the activation function of the *n-th* layer. The main role of dense layers in modern convolutional networks is to perform the discrimination step based on the learned features, in a classification task. The number of learnable parameters in a dense layer depends on the number of neurons of the layer and of the previous one. If dense layer *n* has *N* neuron, and layer *n-1* has *M* neurons, there will be $M \times N$ learnable parameters in the layer.

**Convolutional** layers are very common in computer vision applications. One of their key characteristics is that they keep a sense of spatial relationship between the neurons. Neurons in convolutional layers are organized so that their outputs form "feature maps", which are essentially images that are the result of a convolution between the feature maps of the previous layers (or the input image) and some convolutional kernels. These kernels are small, usually square matrices (see Figure 1.2) whose values are the learnable parameters of the layer. The number of learnable parameters in such a layer therefore depends on the number of feature maps and the size of the convolutional kernels. If layer $n$ has $N$ feature maps, layer $n-1$ has $M$ feature maps, and the convolutional kernels are $K \times K$ squares, there will be $M \times N \times K \times K$ learnable parameters.

**Pooling** layers, as mentioned above, are very often found in combination with convolutional layers in networks designed for computer vision. Pooling layers do not have any learnable parameters. Like convolutional layers, they depend on the spatial relationship of the neurons in the previous layer, and they usually compute a simple statistic (often the *maximum,* but sometimes the *mean* or the *median*) on neighbourhoods. While "max-pooling" and "average-pooling" are the more common options, other variants have been proposed (such as probabilistic approaches which introduce some randomness in the pooling) [86]. They serve as a *down-sampling* step so as to increase the receptive field of the next layers, and, when using max-pooling, as a way to focus the attention of the network on the neurons that are active for a particular input (see Figure 1.3).

**Up-sampling** layers are used to increase the resolution, as a sort of "inverse" step from the pooling layers. Different techniques can be used for the up-sampling, including interpolation [66], transposed convolution [87], and methods using the pooling indices of the down-sampling layers to take into account the position of the feature map's pixels in the original image [88].

**Regularization** layers can be used to reduce the risk of overfitting and to increase the convergence speed of the learning process. The two most common options are "dropout" [89] and "batch normalization" [90]. Dropout works by randomly filtering out the signal between two layers during training (setting the weights of a fraction of the connexions to 0). This will force the network to learn redundant pathways for encoding distinct features, encouraging a better generalization capability. Batch normalization consists in re-scaling and re-centring the inputs of a layer based on the distribution of a mini-batch [90]. During the training phase, a running average of the mean and standard deviation on the whole training set are also computed and can be used for "normalizing" new samples on the trained network.

### 1.5.2 Macro-architectures: how networks adapt to tasks

A common way of looking at deep neural networks is to separate them into two parts: a "general purpose" *feature extraction* part, sometimes called the **encoder**, and a "task specific" part that will combine the features to arrive at the desired output. This, of course, makes it easy to relate deep learning methods to "classical" machine learning, where the "feature extraction" part would use handcrafted features.

The distinction, in a deep neural network, is however somewhat arbitrary, as all layers are typically trained together to both extract features and perform the task. The distinction, however, does make sense from a macro-architectural standpoint, as almost all computer vision models start with a relatively similar encoder.

The **encoder** is mostly made of a succession of convolutional layers and of pooling layers. At the end of the encoder, we will generally find a large amount of "feature maps" with a very low resolution. What comes after the encoder largely depends on the type of task that the model is designed to solve.

**Classification** tasks require an image-level class probability vector as the output. The associated macro-architecture will be to follow the encoder with a "**discriminator**", which is made of several successive dense layers like in a standard multilayer perceptron.

**Detection** tasks where the target outputs are bounding boxes are equivalent to binary classification tasks on the sub-image contained within those bounding boxes. Detection models therefore typically combine a "region proposal" part with a classification network. The encoder

will often be trained on the full images, then another process (which can be another neural network) will produce candidate regions of interest, and a discriminator will be trained on those regions of interest to identify those that include the target object.

**Segmentation** tasks require a per-pixel prediction. The most common way of achieving that is to follow the "encoder" with a symmetrical "**decoder**". A decoder is made of a succession of convolutional layers and up-sampling layers. Starting from the low-resolution feature maps at the end of the encoder, it will produce a high-resolution output.

Some of the extensions to the more "complex" tasks are straightforward. For instance, the difference between "simple" binary segmentation and "**semantic segmentation**" can simply be the number of output channels in the decoder. Similarly, **instance classification** just implies having a multi-class discriminator in the detection network.

The case of **instance segmentation** is more complex, with two distinct approaches commonly used. The first one is based on the "detection" networks and adds a branch to the network which is trained as a decoder to produce a segmentation on a detected region of interest. In this case, "instance segmentation" is therefore seen as "instance detection, with segmentation". The other approach starts from the segmentation, and then tries to separate the potentially overlapping instances. These networks will therefore generally have an encoder, followed by one or several decoders (for instance, one trained to segment the objects themselves, and the other specifically trained to segment the "borders"), and a post-processing step (usually outside of the neural network) that will take the outputs of the decoders and compute the label map. **Instance segmentation and classification** can be achieved by extending any of these approaches to a multi-class case.

Two others commonly found architectures that may be useful to introduce here are the auto-encoder architectures (which can be used for instance for unsupervised feature extraction) and the Generative Adversarial Networks (GAN), which can be used for image generation problems such as super-resolution [91].

**Auto-encoders** follow the same basic architecture as the segmentation network, with an encoder and a decoder block. The key difference is the output: instead of class probabilities, the output will be pixel values, and the auto-encoder will be trained to *reconstruct* the input image (so the *target output* is the same as the *input*).

An auto-encoder will therefore learn sets of features that best characterize the images present in the dataset in a way that is independent of any further usage of the data. On its own, such a network can be useful for applications such as image compression [92], noise filtering [93], anomaly detection [94] or super-resolution [95]. It can also serve as a first step in a semi-supervised learning strategy [96], learning the feature representation with an auto-encoder and then using those features as a pre-training on a supervised dataset.

**Generative Adversarial Networks** (GAN) combine two different networks with separate goals: a "generator", which learns to produce realistic data, and a "discriminator", which learns to detect if a data sample is "true" (meaning that it comes from the real dataset) or "fabricated" by the generator [97].

**Table 1.3. Common tasks and their associated macro-architectures, with references to existing example networks.**

| Task | Macro-architecture(s) | Example network(s) |
|---|---|---|
| Classification | Encoder + Discriminator | AlexNet [35], DanNet [98] |
| Detection | Encoder + Region Proposal + Discriminator | Fast R-CNN [99], YOLO [100] |
| Segmentation | Encoder + Decoder | U-Net [66], SegNet [88] |
| Instance segmentation (and classification) | Encoder + Region Proposal + Discriminator + Decoder | Mask R-CNN [101] |
| | Encoder + Decoder(s) + Post-processing | DCAN [102], HoVer-Net [103] |
| Others | Auto-Encoder, GANs | MCAE [104], DCGAN [105] |

The generator will typically take the same architecture as a "segmentation" network, with an encoder block and a decoder block. The key difference is that the input of the generator will be random noise, and its output will eventually be a realistic image. The discriminator, on the other hand, will be a classification network.

A summary of these task-adapted macro-architectures is presented in Table 1.3.

### 1.5.3 Micro-architectures: infinite choices

While most deep neural network architectures have very similar macro-architectures, infinite variations appear as soon as we look into the finer details of the model. These variations are where most of the "hyper-parameters" of the network itself can be found. In this section, we will briefly explain the most important choices that need to be made in the design of a network at the micro-architectural level. A summary is presented in Table 1.4.

a) Network **depth**: the depth of the network is controlled by the number of layers between the input and the output. For encoders, decoders, and discriminators, there could be any number of convolutional layers, dense layers, pooling and un-pooling layers. In general, deeper networks have a higher capacity for "abstraction", but require more resources to be trained and have a higher risk of overfitting.
b) Number of **kernels** (and associated **feature maps**) for each convolutional layer. This is often referred to as the **width** of the layers. A common trend is to *increase* the number of feature maps as we progress in the encoder (and reduce the size through pooling), and to *decrease* the number of feature maps through the decoder. The same choice needs to be made with the **number of neurons** in dense layers, which will also typically start large and then be reduced to eventually match the number of classes at the output.
c) **Kernel size** and **pooling size**. In convolutional layers, the size of the kernels will determine the number of trainable parameters of the model. The choice in general is to either have fewer convolutional layers with larger kernel sizes, or to stack more convolutional layers with smaller kernels (with 3x3 kernels a very common choice). For pooling, larger sizes mean that the receptive field of the feature maps increase faster, but more information is lost in the process. The effect of these choices can be very different depending on the network, application, dataset and training algorithm, and there are no hard rules on what the "best" choices are.
d) **Activation functions**. Convolutional and dense layers typically involve a two-steps computation, with first a linear operation on their input (the weighted sum) followed by

an activation function. In "older" neural networks, sigmoid functions such as the logistic function or the hyperbolic tangent were often used, but since the introduction of "Rectified Linear Units" [62], ReLU and its variants such as "Leaky ReLU" [63] have been the most widely used (see Figure 1.5).

e) **Residual units** or **residual blocks** are the most common example of "short-skip" connections. The principle of a residual block is to create a branching in the network, such that one branch goes through a series of convolutional layers, and the other branch takes a shortcut and is fed back into the result of these convolutions (see Figure 1.8). In the original ResNet [106], the "convolutional" path includes two convolutional layers, while the "shortcut" path is a simple identity mapping, and the two branches are merged with an addition. Many variations on this idea exist [107]. The number of convolutions in the convolutional path may vary, and there can be more than two branches. Using the addition operation at the end requires that the shape after the last convolution is exactly the same as the shape of the input. This means either having the same number of feature maps in the last convolution as in the input or adding a single convolution in the "shortcut" path. Alternatively, the addition can be replaced with a "concatenation" so that the two paths may have a different number of feature maps. Residual units largely mitigate the vanishing gradient problem by allowing the gradients to flow through the "shortcut" during backpropagation, thus making the training of deeper networks possible.

f) **Regularization parameters.** Whenever regularization layers are used, they come with their own set of choices and parameters. The position of the dropout or batch normalization layers in the overall architecture, and the number of such layers, can be changed. In a dropout layer, the percentage of dropped connections during the training phase also has to be set.

g) **Long-skip connections** are another type of shortcuts in the network architecture. Their main goal is to re-introduce high resolution features to layers in later parts of the network, particularly in the decoder of segmentation networks, which without the skip connection would only see heavily down-sampled feature maps. Where exactly those skip connections should be placed is another micro-architectural choice that needs to be made.

h) **Order of the layers**. While the overall shape of the architecture is dictated by the macro-architecture and generally related to the task definition, there are still smaller-scale choices to be made in the exact ordering of the layers: how many convolutional layers should be placed in between pooling layers? How many pooling layers should there be?

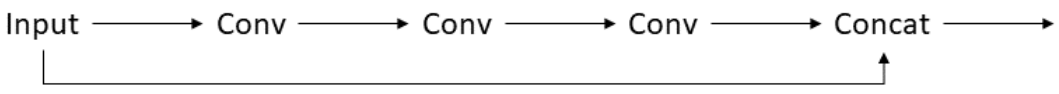



**Figure 1.8. Residual unit with two paths: one with multiple convolutions, and a short-skip connection (which may include a single convolution).**

## 1.6 Lost in a hyper-parametric world

Beyond the constraints of the task to be solved, the choices to be made while designing a deep learning solution to a given problem create an essentially infinite hyper-parameters space (and it could be argued that the *definition of the task* itself is a hyper-parameter of its own, as the translation from the "real-world" objectives to a formal definition is not trivial).

The question on how to best optimize all these hyper-parameters is still widely open. Automated methods have been proposed, using Bayesian optimisation [108] or genetic algorithms [109], [110], but as Zela et al. point out [108]:

> *With an abundance of choices in designing the architecture of deep neural networks, manual feature engineering has nowadays to a certain degree been replaced by manual tuning of architectures.*

Automated methods for hyper-parameters tuning also come at a computationally prohibitive cost and are generally unrealistic for a practical usage for anyone who does not have access to extremely large resources.

**Table 1.4. Main hyper-parameters related to the micro-architecture of a deep neural network.**

| Hyper-parameter | Choices and possibilities |
|---|---|
| Depth (number of layers) | Trade-off between the capacity for higher levels of abstraction and the risks of overfitting and difficulties in learning. |
| Width (number of kernels / feature maps) | Trade-off between capacity for learning more features and the risks of overfitting and longer learning time. |
| Size of convolutional kernels | Variable effect on training time and performance. |
| Size of pooling (and un-pooling) layers | Larger pooling increases the receptive field faster through the layers but loses more information along the way. |
| Activation functions | Most recent models use variations of the ReLU function [62]. |
| Residual units' usage and structure | If "short-skip" connections are used, number of convolutional layers skipped and concatenation method. |
| Dropout parameters | Dropout layers may be put at different parts of the architecture for regularization, with as a hyper-parameter the percentage of connections to randomly set to zero during training. |
| Use of Batch Normalization | Should "batch normalization" layers be introduced in the network for regularization? |
| Long-skip connections | Choice of which low-level features to re-introduce in which later stage of the network. |
| Order of the layers | Number of convolutional layers, of pooling layers, position of the regularization layers, etc. |

One of the difficulties of this search in the hyper-parameters space is that it can be very difficult to extricate the effects of a change in hyper-parameters from the effects of randomisation. Many parts of the deep learning pipeline include some randomisation: weight initialisation, data augmentation, data shuffling, etc. To be sure that a set of hyper-parameters is truly "better" than another, it would have to show its improvement over multiple random seeds, which add yet another multiplicative factor to the computational cost.

By necessity, hyper-parameters tuning needs to be limited, with most of them being fixed by the algorithm's designer based on experience and prior knowledge of working solutions. Micro-architectural choices, for instance, are often done based on very limited testing on small datasets to ensure that the resulting model is capable of learning useful features. Similarly, for weights initializers or optimizers' parameters, only a limited set of values are typically tested.

It is however something that should always be kept in mind when analysing results that compare different network architectures, or choices in the pipeline: there is always a possibility that a better tuning of the neglected hyper-parameters could change the results one way or another. This does not mean that those results are worthless, but that results from a single experiment should always be taken carefully.

## 1.7 Conclusions

While deep learning has been described as a revolution, our overview of the history of modern deep neural network would rather characterize it as a natural progression of concepts that were laid down from the very beginning of research into machine intelligence. If there has been a revolution, it is probably fair to say that it happened more in *hardware* development than in the theory of neural networks. Modern deep neural networks share common characteristics and building blocks with networks from the 1960s-1980s, and it is doubtful that modern improvements in optimization techniques or in the design of network architectures would have had a significant impact without the rapid improvement in parallel processing available to the research community.

Deep learning algorithms are always integrated into a larger pipeline, where the design of the model itself is only one of many hyper-parameters that need to be set. It is easy to get lost in all of the choices that come with the design of a deep learning solution. If deep learning, as many like to say, is a "black box"[13], it is certainly one with many levers and buttons attached to it.

[13] https://www.nature.com/news/can-we-open-the-black-box-of-ai-1.20731

# 2 Digital pathology and computer vision

As we have seen previously, it is difficult to come to an entirely satisfying definition of deep learning. While the term "digital pathology" is not quite as fuzzy, it is still not entirely straightforward.

The 2014 Springer book "Digital Pathology" [111] defines it thusly:

> *"Digital pathology, in its simplest form, is the conversion of the optical image of a classic pathologic slide into a digital image that can then be uploaded onto a computer for viewing."*

They note, however, that "the concept and functionality of digital pathology has grown exponentially, so that a more accurate current definition would be that it is the field of anatomic and microanatomic pathology information systems", including the "real-time evaluation, comparison, two- to three-dimensional reconstruction, archiving, dissemination for widespread viewing and consultation, compilation with other patient data, data mining, and use for education, clinical diagnosis and patient management, research, and the development of artificial intelligence tools [of specimens in digital form], and this may still only be scratching the surface."

A 2010 Scientific American editorial [112] makes a graphical comparison between the pathway of "traditional pathology" and that of "digital pathology". In traditional pathology, the slide is sent to the primary pathologist for "subjective analysis", then "may be sent, in series by mail, to one or more consultants". By contrast, in digital pathology, a slide is scanned and "screened by computer (objective analysis)", with the results attached to the patient electronic record so that "multiple reviewers can simultaneously see and discuss digitized slides and supporting documents."

The Digital Pathology Association has the following definition:

> *"Digital pathology is a dynamic, image-based environment that enables the acquisition, management and interpretation of pathology information generated from a digitized glass slide."*[14]

As can be seen from these definitions, digital pathology refers at the same time to the **acquisition process** of the optical information of the slide into a digital format, and to the **different uses** that can be made of the "virtual slide". The main components of digital pathology can therefore be summed up as:

a) The **acquisition** hardware (slide scanner).
b) The **visualisation** software.
c) The "**information system**", including the tools for sharing, distributing, and annotating the slides, and to link them to the rest of the patient records.
d) The **image analysis software** to automatically evaluate the content of the slide.

In this section, we will first briefly look at the history of computer-assisted pathology, from early attempts at automated pap smears analysis in the 1950s to the current era of deep learning and "grand challenges". We will then look at the digital pathology workflow, and the needs that pathologists have for automated computer vision methods for research or clinical practice. Finally, we will take a snapshot of the digital pathology world in 2010, at the time of the first

[14] https://digitalpathologyassociation.org/about-digital-pathology, last retrieved 15/03/2021

computer vision "challenge" in digital pathology, held at the ICPR conference, right before the Deep Learning "invasion".

## 2.1 History of computer-assisted pathology

As digital pathology is an extension of traditional pathology, it is interesting to briefly look back at the history of computer-assisted pathology. The idea of using computers to automatically assess pathology samples is almost as old as computers themselves. If we want to look at the meeting of computer science and pathology, the best starting point might be 1955, with the presentation of the Cytoanalyzer by Walter Tolles [113] (shown in Figure 2.1).

The system is described as a scanner which converts "the density field of the slide into a serial electric current which is then used to analyse these several cell properties", linked to a computer which function is to "accept the signal from the scanner, to apply certain rules of admission or rejection to the signals, to measure the desired cell characteristics and to distinguish between signals arising from abnormal cells and signals due to normal cells". The results of clinical trials indicated that it was "inadequate for practical applications" [114], but the idea was interesting enough to lead to the development of similar instruments in the following years [115].

These early attempts at automated cytology have very limited "intelligence" and rely on very simple signal thresholding rules. This is understandable, given that they came before the founding of "artificial intelligence" as a field of computer science. Only a decade later, however, computer vision techniques were starting to take shape in cytology, with the development of the CYDAC to scan microscopic images, digitize them and convert them to magnetic tapes [116]. Automated computer vision methods to extract morphological information and determine cell types were developed by Mortimer Mendelsohn and Judith Prewitt in the mid-1960s [9], [117].

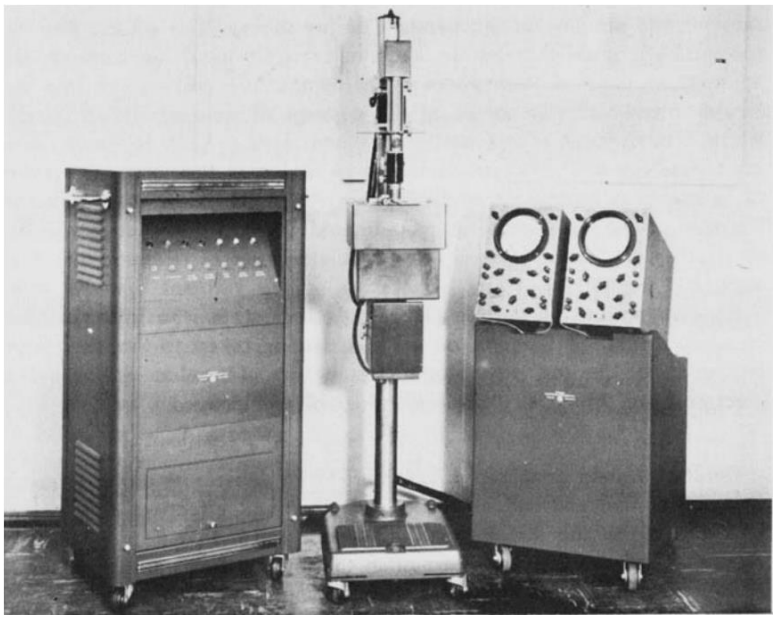


**Figure 2.1. The Cytoanalyzer in 1955** [113]**. On the left is the "power supply and computer", in the centre is the scanner, and on the right are oscilloscopes "for data-monitoring and presentation".**


**Figure 2.2. Demonstration of "satellite-enabled robotic-dynamic tele-pathology" in 1986. (a) Press briefing by Dr Weinstein in Texas. (b) Pathologist operating the light microscope from Washington, DC.** [118]

Around the same time, we find the first experiments in tele-pathology, with a Princeton laboratory mounting a black-and-white television camera on a light microscope [119]. The first clinical use of this technology would wait until 1968. A clinic at Logan International Airport in Boston was connected to the Massachusetts General Hospital in a very experimental telemedicine system which allowed pathologists in the hospital to remotely examine "black-and-white images of blood smears and urine samples using remote analogue video microscopy" [119]. The whole system was impressive in its scope. In addition to the video microscopy, it was equipped with "a range of cameras for long shots and close-ups to aid physical examination", and other cameras for transmitting X-rays and electrocardiograms [120].

One of the pathology residents at the Massachusetts General Hospital at the time, Ronald Weinstein, would end up becoming one of the pioneers of modern tele-pathology in the 1980s, and was driven in part by the large interobserver variability among experts in clinical studies [118]. A big innovation that made tele-pathology more reliable was the introduction of "dynamic-robotic tele-pathology" (see Figure 2.2), allowing the remote observer to choose the field of view and focus of the microscope.

The dynamic-robotic tele-pathology of the 1980s required a pathologist "on hand" to remotely operate the microscope. When digital cameras became more readily available in the 1990s, systems with cameras mounted on a microscope taking static images and sharing them on a network of pathologist workstation became possible. However, these were unable to capture the entire slide. In 1999, however, Wetzel and Gilbertson introduced an automated whole-slide imaging system with 20x magnification and 0.33 μm/pixel resolution capable of imaging a slide in 5 to 10 minutes [121]. This did not immediately revolutionize the practice of pathology. In 2011, Pantanowitz et al [122] wrote that "there appear to be several technical and logistical barriers to be overcome before [Whole-Slide Imaging] becomes a widely accepted modality in the practice of Pathology," citing the problem of artefacts such as tissue folds, bubbles and poor staining making the introduction of Whole-Slide Imaging (WSI) systems being shown "to stress the system in terms of reliability and throughput."

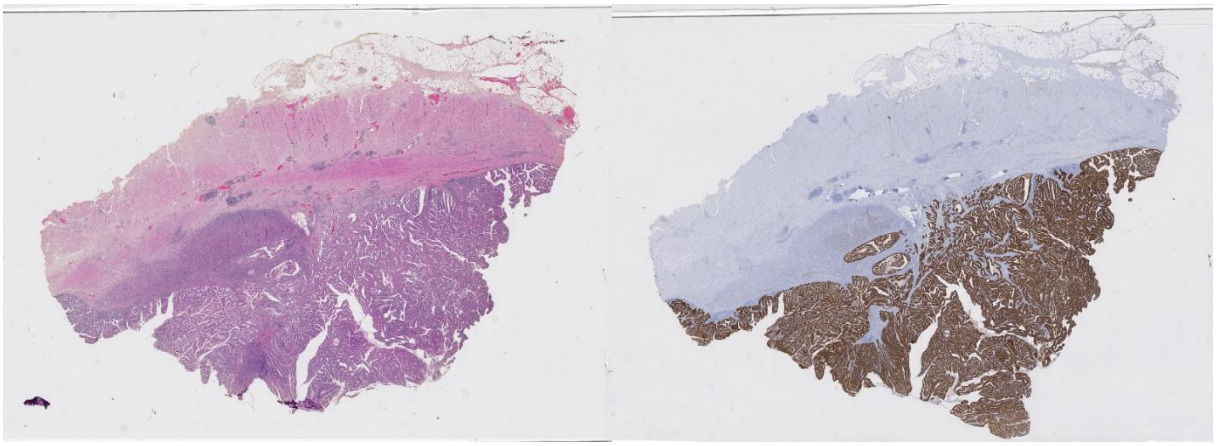


**Figure 2.3. Example of two whole-slide images (WSI) extracted from the same tissue block from a colorectal tumour. Stained with (left) H&E and (right) anti-pan-cytokeratin IHC with haematoxylin counterstain.**

As standardization of the process evolved and validation studies confirmed that digital pathology gave similar results to the "classical" methods, the regulatory framework evolved so that digital pathology started getting more readily adopted in clinical, research and educational practice [123].

The adoption of WSI had a huge impact on image analysis for pathology, with entire slides being now available for analysis instead of limited fields of view [124]. Larger datasets could now be gathered and provided to the image analysis and artificial intelligence community, leading to the development of algorithms that promised to "improve accuracy, reliability, specificity and productivity" of digital pathology [12]. As with deep learning, the development of new software tools such as the "vendor-neutral" OpenSlide library [125] has made the programmatic use of WSIs easier, allowing more researchers to design image analysis pipelines.

## 2.2 The digital pathology workflow

The digital pathology workflow is a long, multi-steps process that requires trained specialists and specialized hardware. In this section, we will look at those steps, from the extraction of the tissue samples from the patient to several uses of the digitized images.

### 2.2.1 Sample acquisition: from the body to the scanner

Tissue samples can be acquired from the patient during surgeries or biopsies. After extraction, a sample will go to a series of **tissue processing** steps [126]: fixation, dehydration, clearing, paraffin infiltration and embedding. The goal of these steps is to stabilize the tissue and to create a "block" that can be sectioned with minimal damage. The block can then be cut into thin slices (around 5 µm) using a microtome, which are then mounted onto glass slides.

The next important step is the **staining** of the tissue. Staining techniques were developed as early as the mid-nineteenth century [127]. Staining agents will selectively stain certain tissue components, creating contrasts that allow the pathologist to better see the structure of the tissue. The mechanisms through which certain stains will favour certain tissue components are complex, relying on different biochemical processes [128]. The most widely used general-purpose stains combination is haematoxylin combined with eosin (H&E). Essentially, haematoxylin stains nuclei in blue, eosin stains cytoplasm and extracellular matrix in pink. More recently,

immunohistochemical (IHC) techniques have been developed to selectively reveal the expression of target proteins in the tissue, using antigen-antibody interactions. Figure 2.3 shows two whole-slide images from the same tissue block, stained with H&E and using an IHC marker, respectively.

Once the slide is ready, it can go through the **scanning** process to be digitized. Slide scanners are essentially optical microscopes combined with a digital image capture device, and a mechanical system to change the acquired region. WSIs may be obtained using a line scanning process (with the camera capturing strips) or a tile scanning process (with the camera capturing squares) [129]. The highest resolution available is generally 'x20 equivalent' (around 0.5µm per pixel) or 'x40 equivalent' (0.25µm per pixel). The main challenges of the scanning process are the stitching of the lines or tiles, and the management of the focus over the whole slide, as the focal plane will vary with the topography of the tissue section.

As an alternative to the WSIs produced by this workflow, it is also possible to put multiple small tissue samples extracted from different patients on the same block to produce "tissue microarrays" (TMA). While WSIs have the advantage of providing more context by showing large regions of tissue, TMAs make it possible to process more samples more quickly and homogeneously.

### 2.2.2 Applications

The first, and probably most common application of digital pathology is telepathology. Digitized "virtual" slides can be sent to pathologists outside of the facility where the samples were acquired. This allows for a greater specialization of the pathologists and makes it easier to get second opinions for difficult cases [111].

More interesting to us in the context of this thesis are the applications related to image analysis. It is particularly important to understand what pathologists hope to get from computer vision algorithms, as there can sometimes be a disconnect between the needs of pathologists and the products of the machine learning and computer vision communities [130].

Hartman et al. list some "key areas where image analysis has had a role to play in pathology and tissue-based research" in 2014 [131]:

- Quantitative evaluation of nuclei (morphology, DNA content…)
- Measures related to tissue architecture (e.g. cellular organisation)
- Quantitative immunohistochemistry and biomarker discovery
- Tissue microarray analysis
- Measure of tumour heterogeneity
- Measures of fluorescence properties
- Automated tumour detection

In general, the main expected improvement of using automated methods is to avoid the problems of "inconsistency in diagnostic decision-making in pathology, poor reproducibility in grading disease and the variation that exists in image interpretation."

Madabhushi and Lee provide a review of image analysis and machine learning in digital pathology in 2016 [132]. The first big research avenue they identify is "segmentation and detection of histologic primitives", such as glands and nuclei. They see this as an important prerequisite for quantitative histomorphometry, the detailed characterization of the morphologic landscape of the

tissue. Another direction for image analysis is "tissue classification, grading and precision medicine."

These applications may be used as aid to clinical diagnosis, but also for research purposes: as the digitization of pathological slides becomes routine practice, huge datasets can be created, and automated measures may be extracted for large retrospective studies.

### 2.2.3 Cancer grading systems

Cancer staging or grading systems provide a codification of the assessment of a tumour, which is useful to provide prognosis (i.e. cancer outcome prediction, such as risk of recurrence after treatment or death) and to compare groups of patients [133].

The Tumour-Node-Metastasis (TNM) system, maintained by the American Joint Committee on Cancer (AJCC) and the International Union for Cancer Control (UICC) is the "general purpose" tumour grading system, in the absence of another system specifically targeted to the cancer type. The grades are based on an aggregation of three categorical assessments [134]:

- The **Primary Tumour** category, based on its size and extent. Categories generally include TX (unknown), T0 (no evidence of primary tumour), Tis (*in situ* carcinoma), T1-T4 (primary invasive tumour, with higher categories based on size and/or local extension).
- The **Regional Lymph Node** category, assessing the existence and extent of a regional lymph node involvement. Categories generally include NX (unknown), N0 (no regional lymph node involvement), N1-N3 (regional node(s) containing cancer cells, with higher categories for increasing number of nodes, size of the nodal cancer deposit, etc.)
- The **Distant Metastasis** category, which specifies whether there is evidence for the presence of distant metastasis (M1) or not (M0) at the time of diagnosis.

It is important to note that the histopathological examination of the tumour is only a part of the grading criteria, alongside other imaging modalities, physical examination, clinical history, etc. While there are general criteria and trends that are common in all cases, the specific staging is tuned to the specific tumour types based on relevant survival studies.

For some cancer types, alternative systems have been fully validated and are now largely adopted worldwide [134], including the Gleason scoring system [135] for prostate cancer (and its 2016 Epstein modification [136]) and the Nottingham system [137] for breast cancer.

The **Nottingham system** is based on the assessment of tubule formation (with three categories based on their extent relative to the total tumour), nuclear pleomorphism (three categories based on the size and regularity of the nuclei) and mitotic count (three categories with specific numbers depending on the examined field area).

The **Gleason** system is based on a "relatively low magnification" assessment of the pattern of growth of the tumour, with five distinct pattern types of increasing apparent malignancy being identified (see Figure 2.4). The predominant ("primary") pattern and lesser ("secondary") pattern are recorder for each case, and the final score is based on the addition of the two pattern categories (so if the "pattern 3" is predominant with some "pattern 2" present, the score would be 3+2 = 5).

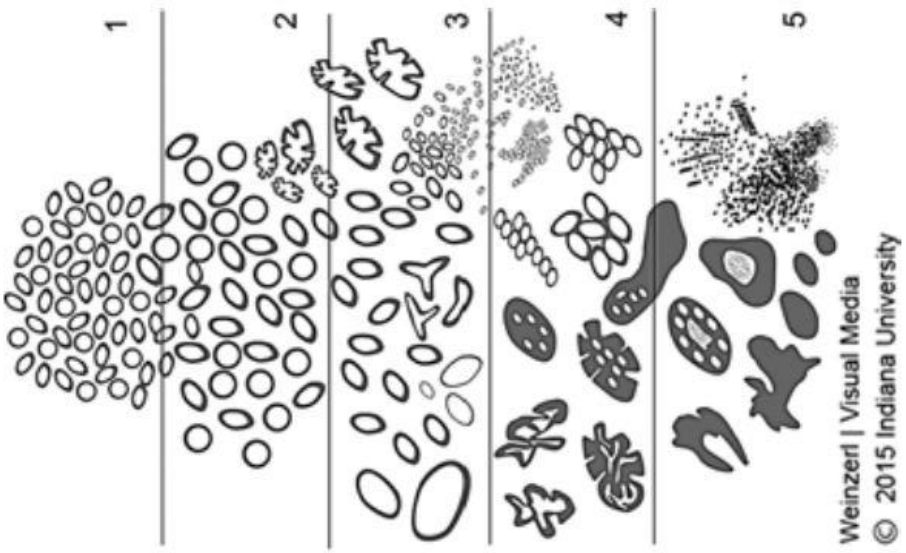



**Figure 2.4. ISUP Gleason schematic diagrams for Gleason histological patterns (reproduced from** [138]**).**

These two systems in particular have driven a lot of research in computer vision for digital pathology, as their criteria are directly based on the analysis of the pathology samples. It is therefore not surprising that lots of publications and challenges relate to nuclear detection and segmentation, mitosis detection, or Gleason scoring.

### 2.2.4 Immunohistochemistry (IHC)

Immunohistochemistry use antigen-antibody interactions to selectively bind the staining agent to target proteins. This allows for a specific targeting of cells that express those proteins. By carefully selecting the antibodies, it is therefore possible to discriminate between the benign and malign nature of certain cell proliferations, or to identify micro-organisms or materials secreted by cells [139].

IHC techniques can therefore be extremely useful at the clinical level for diagnosis. It is also very valuable from a research perspective, in the search of biomarkers that may or may not correlate with tumour aggressiveness or patient outcome. The interpretation of IHC images, however, can be very difficult and subjective, often based around an estimation of the proportion of tissue or cells where the stain is present, leading to large inter- and intra-observer variability [140].

The overall structure of the cellular tissue is usually made visible through a "counterstain" (often haematoxylin) which makes all the cells visible (see Figure 2.3).

## 2.3 Image analysis in digital pathology, before deep learning

As we will show in the next chapter, deep learning techniques quickly spread in histopathological image analysis in the 2010s. Before we get to those algorithms, however, it is interesting to take a look at what existed before.

In 2009, Gurcan et al. published a large review of histopathological image analysis [11]. The next year, they organised the ICPR 2010 "Pattern Recognition in Histopathological Image Analysis" (PR in HIMA) challenge, where several image analysis algorithms were tested on lymphocyte segmentation on breast cancer images, and centroblasts detection in follicular lymphoma [141].

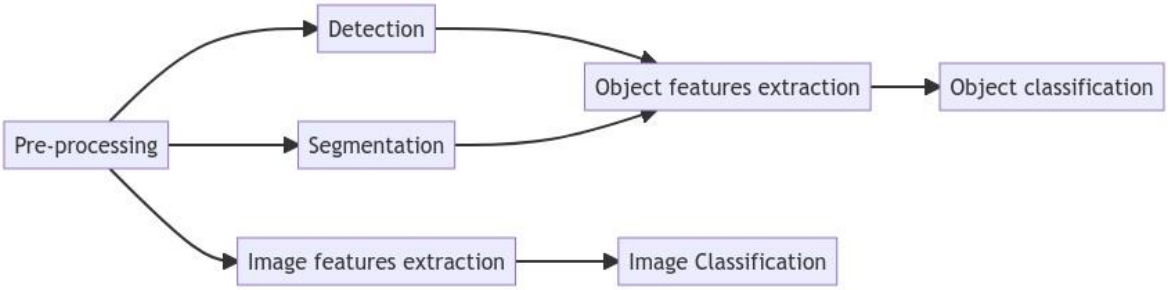



**Figure 2.5. Classic image analysis pipelines for classification. These pipelines typically either extract global features from the entire image (for image classification) or have a first step to detect or segment candidate objects from which object-specific features can be extracted for the classification.**

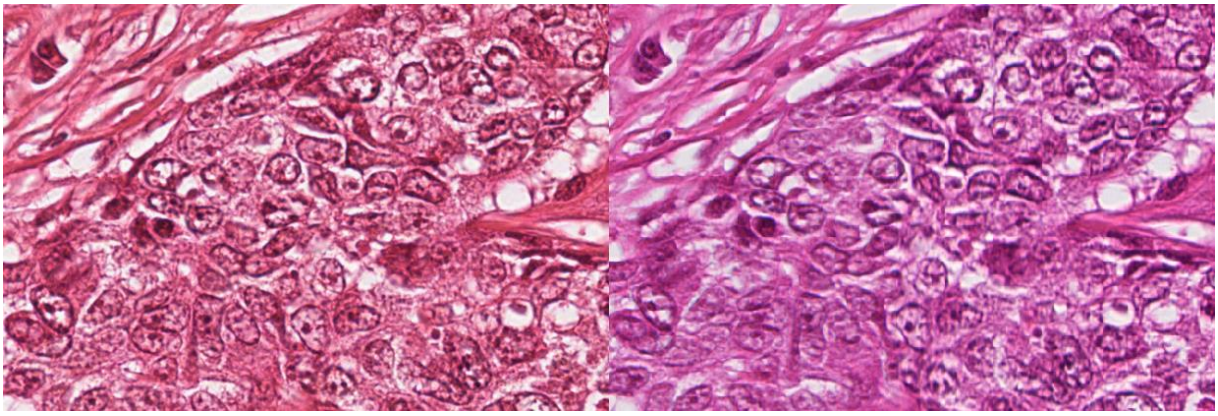


**Figure 2.6. Same region cropped from an image in the ICPR 2012 MITOS dataset, acquired with two slide scanners: (left) an Aperio ScanScope XT and (right) a Hamamatsu NanoZoomer 2.0-HT, showing the difference in colour and image quality in the digital image.**

A 2011 review by Fuchs et al. [142] also provide some interesting insights on the state of "computational pathology" at the time, with a particular focus on questions of interobserver variability and ground truth generation which will be discussed more fully in later chapters of this thesis.

From these reviews, we can see how the classic image analysis pipeline (presented in Figure 2.5) has been adapted for digital pathology.

### 2.3.1 Pre-processing

The interpretation of histopathological images relies a lot on the staining process. A very common pre-processing step in digital pathology image analysis consists in **stain** or **colour normalisation**. The objective of this step is to reduce the differences due to the exact protocol used for the staining process, or to differences in the acquisition setup or the acquisition hardware. For instance, Figure 2.6 illustrates how the same slide scanned by two different machines can lead to large differences in the colours of the resulting digital images.

Typically, stain normalization methods are based on the knowledge that, in general, histopathological slides use a combination of two stains. The most common combination is H&E, where the blue haematoxylin stains the nuclei, and the pink-red eosin stains the cytoplasmic

elements [127], whereas in immunohistochemistry there is generally a combination of an IHC marker, often using 3,3'-Diaminobenzidine (DAB) as chromogen which exhibits a brown colour, with a haematoxylin counterstain. In 2009, Macenko et al. [143] proposed to look at the pixel distribution in the RGB colour space and to use Singular Value Decomposition to find the two main directions of the distribution. Colour deconvolution [144] can then be performed to separate the contributions to the two "stain" channels, and colour normalization consists in aligning the vectors of the "target" image to the vectors of a "reference" image. Several improvements on the same general concept have been proposed over the years, as stain normalisation is still often used as a pre-processing step even with deep learning techniques [145], [146].

Another important characteristic of histopathological images is their **size**. It is not uncommon for WSIs to have a full resolution size of around 80.000x60.000px, taking up about 15 gigabytes of data[15]. Even with today's hardware, it is impractical to apply any image analysis algorithm to the entire slide at once. A typical pre-processing step would therefore consist in extracting **image patches** from the larger WSI. If the entire WSI is to be processed, then a common strategy would be to use a regular tiling grid on the image to extract evenly spread patches that cover the whole slide. As a large part of the slide will often only contain the glass slide itself and no tissue (see Figure 2.3), a simple thresholding method can be applied to filter empty patches.

### 2.3.2 Detection and segmentation

While detection and segmentation are often "end goals" of computer vision methods by themselves, they generally are not the desired outcome for pathologists, but rather an important step towards disease grading or diagnosis [11]. However, as fully automated diagnosis systems are even today largely out of reach (and arguably not particularly desirable in the first place), detection and segmentation results still hold a lot of interest. In clinical practice, such applications may for instance provide pathologists with measures such as cell counts or mitotic counts, which are very time consuming for human experts to provide and obvious candidates for automatization.

Nuclei segmentation has been the subject of a lot of research. While it seems at first glance that relatively simple thresholding techniques may give good results, as the nuclei are typically much darker than their surroundings (see Figure 2.7), the large variability in image sets leads to inconsistent results [11]. Watershed-based methods suffer from the same issues. Doyle et al. [147] proposed a slightly more complex method. A first thresholding is made to keep all low intensity pixels. It is followed by a Euclidian Distance Transform (EDT) to compute the distance to the closest background pixels. Another threshold is applied to the EDT results to keep pixels that are far from the background (and therefore should be closer to the centre of the objects of interest), and template matching with different elliptical templates representing usual nuclei shapes and sizes is then applied and pixels with a high correlation to at least one of the templates are kept as "nuclear centroids".

Similar pipelines were developed for gland segmentation, where first low-level colour and texture information are used to give a general pre-classification of each pixel, then further constraints based on shape, size and spatial relationships are used to refine the results [11].

[15] https://dicom.nema.org/dicom/dicomwsi/

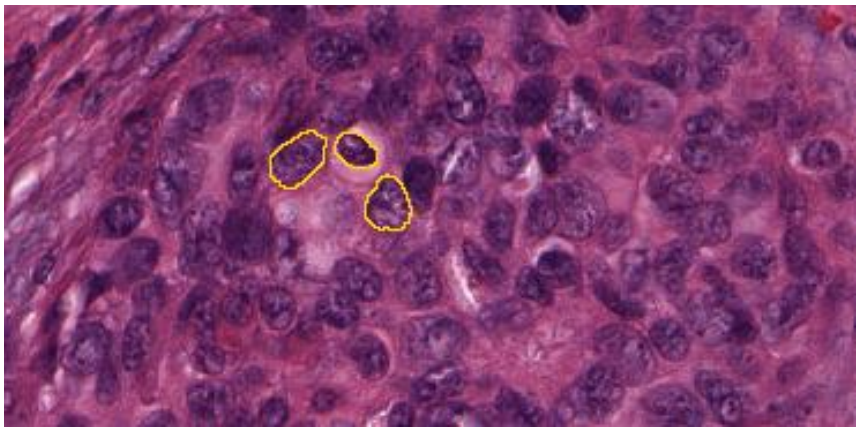


**Figure 2.7. Example of an H&E stained oestrogen receptor positive (ER+) breast cancer image with some manually annotated nuclei, from Janowczyk's nuclei dataset** [148]**.**

### 2.3.3 Feature extraction

Feature extraction is where domain knowledge particularly needs to be present in classic image analysis pipelines, as the useful features are often inspired by the attributes identified by pathologists as relevant for grading and diagnosis.

Object-level features are computed on the pixels that intersect with the binary mask of the previously segmented object. They include information about the shape and size (area, eccentricity, perimeter, centre of mass, symmetrical properties…), the colour information (optical density, hue…), or the texture (co-occurrence matrix features, wavelet features…) [11].

Colour and texture-based features can also be extracted directly on full images or image regions to provide patch-level classification.

Another approach to use the detected object is to build a graph of their spatial relationship across the entire image, and to then compute graph-based features to characterize these relationships [149].

It is also common to use a multi-resolution approach to compute those features, especially if working with larger WSI images, as they tend to contain different types of information at different scales, from the very high-resolution nuclei to larger tissue structures such as glands.

### 2.3.4 Feature selection and Classification

Long before deep learning methods were applied, machine learning approaches were already used to perform classification based on the extracted features. While many algorithms in the 2000s mostly relied on rule-based expert systems, it was also very common to see a more statistical approach to pattern recognition [142].

The machine learning approach to pattern recognition consists in computing as many independent features as possible in the feature extraction phase, then to reduce the dimensionality of the problem based on the information present in the dataset. For instance, features can be added or removed from the feature set based on whether their presence improves the classifier. Dimensionality reduction can also be done using statistical methods such as Principal Component Analysis (PCA), transforming the feature space so that its dimensions are as informative to the classifier as possible.

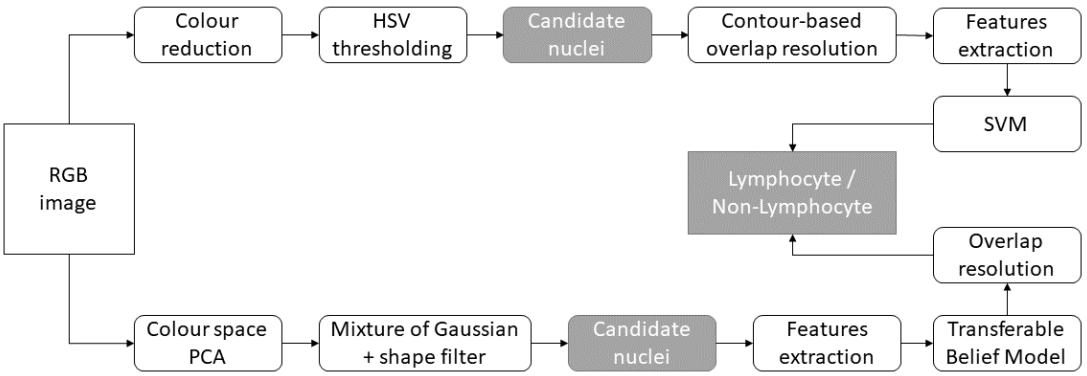



**Figure 2.8. Comparison of two pipelines from the ICPR 2010 PR in HIMA challenge, from Kuse et al.** [150] **(top) and Panagiotakis et al.** [151] **(bottom).**

In terms of the classifiers themselves, the most popular in the 2000s in digital pathology image analysis were probably the Support Vector Machines (SVMs). It was also established that ensemble classifiers were, in general, the preferred solution to reduce the bias and variance of individual classifiers [11].

### 2.3.5 Methods from the ICPR 2010 PR in HIMA competition

These different approaches are well illustrated by the methods proposed by two participating teams at the ICPR 2010 PR in HIMA competition for lymphocyte segmentation. Their overall pipelines are illustrated side-by-side in Figure 2.8.

In Kuse et al. [150], a pre-processing is applied to reduce the number of colours in the image using a mean-shift clustering. A thresholding operation is then done based on the Hue value in HSV space to find candidate nuclei. The candidates are labelled using connected components analysis, and a contour-based approach is used to separate overlapping nuclei. Eighteen texture features were extracted and used to train a SVM classifier on a supervised set of lymphocyte / non-lymphocyte nuclei.

Meanwhile, Panagiotakis et al. [151] first reduced the dimensionality of the image by performing a PCA in colour space and selecting the first dimension. A Mixture of Gaussian model was then used to determine three separate pixel values distributions representing stroma, non-lymphocyte nuclei and lymphocyte nuclei. Regions detected as candidate nuclei are filtered based on handcrafted area and eccentricity criteria. Two features are then extracted from each remaining candidate region (mean value and variance), and a Transferable Belief Model filters out remaining non-lymphocytes. Finally, the area and eccentricity are used again to split overlapping nuclei in an iterative method that tries to maximize a criterion describing the "expected shape" of the objects.

### 2.3.6 Mitosis detection

Mitotic count is often used as part of the grading system for different types of cancer [134], such as with the Nottingham system used for grading breast cancer tumours [137]. The development of automated counting, detection or segmentation methods has therefore been a topic of interest for some time. Figure 2.9 illustrates the pipelines of several methods proposed over the years.

Between 1984 and 1997, a Dutch team developed and improved a mitosis detection method using a "classic" image analysis pipeline [152]–[154]. Their first attempt used greyscale photographs of

microscopic images. They selected candidate nuclei based on simple thresholding and morphological filtering. Linear discriminant analysis was then used on some contour and histogram features to classify the candidate between mitosis and non-mitosis. Over the years, they improved the method by obtaining better acquisition devices, moving to colour images (and therefore colour features), and using a region growing approach to select the candidate nuclei. While their False Positive Rate remained relatively high (22-42%), they concluded that the false negatives were rare enough (5-8%) that "the current system may serve well as a pre-screening device" [154].

Progress, however, was slow. The 2008 method by Dalle et al. [155] is not that different from the 1997 attempt by Beliën et al. They used global thresholding and morphological filtering to detect candidate nuclei, and then a Gaussian model to discriminate between mitotic and non-mitotic cells based on a few shape and colour features. The main difference of Dalle et al.'s method is that they go one step further in the "automated diagnosis", as they compute the mitotic score (three categories based on the mitotic count) and evaluate their method based on the agreement on the score with a pathologist, instead of the per-mitosis detection performance. While it is certainly interesting to get closer to a clinical application, it is also difficult to compare their performance to other methods as they do not provide any per-mitosis results.

Classic pipelines for mitosis detection did not entirely disappear after the start of the deep learning era. In 2015, Paul et al. [156] claimed state-of-the-art results on several mitosis detection challenge datasets (MITOS12, AMIDA13, MITOS-ATYPIA-14) without deep learning. In their pipeline, they first use only the red channel for pre-processing and the segmentation of candidate nuclei. They then use the red and green channel normalized histograms of the candidates with a random forest classifier to discriminate between mitosis and non-mitosis. The selection of candidate nuclei in Paul et al. is significantly more complex than in the previous methods, with an iterative entropy-based pre-processing and segmentation step, and the Random Forest classifier is able to manage the higher dimensionality of the very low-level features computed.

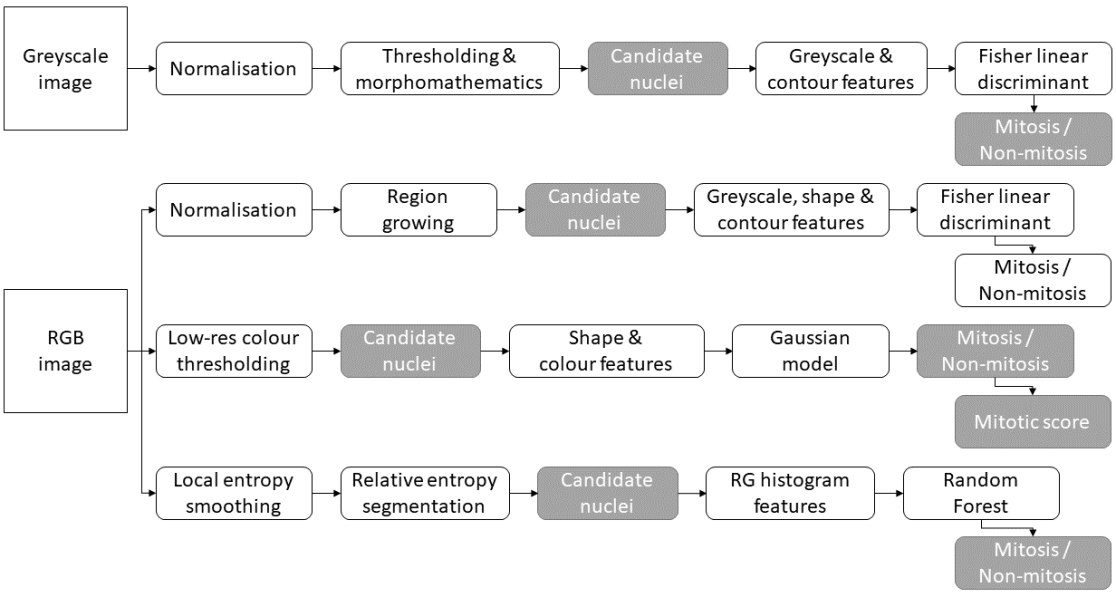



**Figure 2.9. Evolution of the "classic" image analysis pipeline for mitosis detection, showing from top to bottom the methods of Kaman et al.** [152]**, Beliën et al.** [154]**, Dalle et al.** [155] **and Paul et al.** [156]**.**

## 2.4 Characteristics of histopathological image analysis problems

To conclude this chapter, we will summarize the main characteristics of histopathological image analysis problems.

### 2.4.1 Colour spaces

Modern whole-slide image scanners produce colour images encoded with either 8-bits or 16-bits RGB channels. The distribution of the pixels in the colour space is very different from what can be found in natural images (such as photographs, for instance). The staining process used in the preparation of the tissue sample means that there will generally be few distinct hues in the image, as illustrated in Figure 2.10. As we previously mentioned, pre-processing steps such as stain normalisation or deconvolution are very common in digital pathology image analysis pipelines.

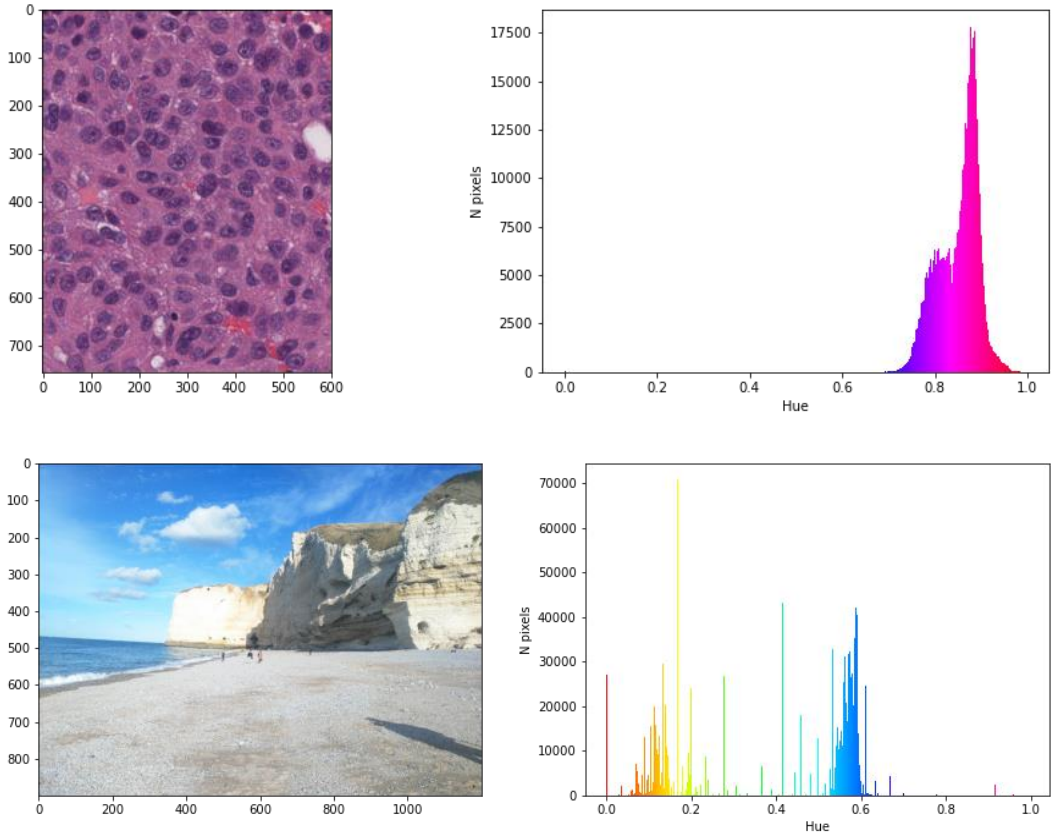



**Figure 2.10. Difference in hue histograms between an H&E-stained histopathological image (taken from the MoNuSAC dataset** [24]**) and a natural photographic image. Even though there are dominant hues in the photographic image, the concentration of the values into narrow peaks is much more visible in the histopathological image.**

### 2.4.2 Image size and multi-scale information

WSIs are very large images. At full resolution, they can take gigabytes of disk space, and therefore cannot be processed all at once. WSI scanners can acquire the image at different levels of magnification. It is therefore possible to use multi-scale information not just from interpolation to lower resolutions (with the risk of aliasing artefacts), but also by switching between the different acquisition magnification levels. Histopathological image analysis pipelines have to determine

which resolution, or resolutions, contain the information needed to solve the task. Some tasks, such as mitosis detection, typically work better at the highest resolutions available, so that nuclei features are visible [154], but for other applications such as Gleason grading it may be beneficial to use lower magnification levels (which implies seeing a larger context in a same-sized patch) to detect useful features [157].

### 2.4.3 Scope and explainability

In digital pathology, the scope of the problem is generally much larger than the specific image analysis task that the algorithm is trying to solve. The output of an image analysis system will typically be either an image-level assessment (class, grade…), an object-level assessment (count, localisation…) or a pixel-level assessment (segmentation). The expected output of a fully automated "AI for digital pathology" system, however, would be a patient-level diagnosis or prognosis or therapy indication, and would have to incorporate data outside of the image into its inputs (such as patient history, other imaging modalities, laboratory data, etc. [158]). The potential trap of a direct data-to-diagnosis approach, however, would be to create a "black box" effect where it becomes impossible (or at the very least impractical) to determine the "reasoning" behind an algorithm's output.

Intermediate, lower-level outputs such as those typical in image analysis problems are therefore still very relevant to produce, as they provide the potential for better explainability of the algorithms' results. The clinical end result, however, should not be forgotten either. It should at the very least inform the evaluation process and metrics by which the results on a task should be assessed.

The criteria for object classification or tumour assessment are often fuzzy and ill-defined. Despite great efforts to standardise and objectify these criteria, there is still often a lot of room left for the subjective (and experienced) view of the pathologist. This also makes it hard to describe the image analysis tasks formally, and to design expert rules. Digital pathology tasks are therefore good candidates for deep learning methods, which are capable of designing their own features directly from the data. The acquisition and annotation of the data, however, is very challenging for the same reasons, and obtaining accurate supervision is, as we will see in the rest of this thesis, a constant struggle and one of the largest issues that deep learning faces in digital pathology.

### 2.4.4 Classification and scoring

In many digital pathology classification tasks, the categories are *ordered*. As we have seen, many tumour assessment standards revolve around "scores" or "grades". These problems are therefore straddling the line between "classification" and "regression". However, even though they are ordered, they are typically treated as distinct categories. We will generally refer to these sorts of tasks as scoring tasks, which can be seen as a subcategory of classification tasks.

# 3 Deep learning in digital pathology

Finding the "first" application of deep learning in digital pathology is difficult, as the definitions are fuzzy. The earliest example that we could find that would qualify as such comes from Malon et al. in 2008 [159]. Using a network architecture based on LeNet-5 (LeCun et al.'s 1998 version of their architecture [61], which is very similar as well to Krizhevsky et al.'s 2012 ImageNet-winning network [35]), they showed that convolutional neural networks were capable of encouraging results on epithelial layer, mitosis and signet ring cell detection.

It is in 2012, however, that deep learning really took off, in digital pathology and in general, with the ICPR mitosis detection challenge ("MITOS12"), won by the IDSIA team of Cireşan et al. with a convolutional neural network [160]. Around the same time, Malon et al. [161] also used a convolutional network for mitosis detection, while Cruz-Roa et al. used a deep learning architecture for basal-cell carcinoma cancer detection [162].

The history of deep learning in digital pathology is strongly linked to the history of "grand challenges". The first digital pathology competition was organized in 2010 at ICPR ("Pattern Recognition in Histopathological Images") [141]. It proposed two tasks: counting lymphocytes and counting centroblasts. Five groups submitted their results, none of which used deep learning methods. All subsequent digital pathology challenges, starting with MITOS12, would be won by deep learning algorithms. The ICPR 2010 challenge was mostly an experiment laying the groundwork for future challenges. To quote the authors:

> *"Given that digital pathology is a nascent field and that application of pattern recognition and image analysis methods to digitized histopathology is even more recent, there is not yet consensus on what level of performance would be acceptable in the clinic."*

The main purpose of the contest was, therefore, "to encourage pattern recognition and computer vision researchers in getting involved in the rapidly emerging area of histopathology image analysis." The number of challenges grew steadily over the next years, often linked with large conferences such as ICPR, MICCAI and ISBI [130]. As these challenges often published at least their training datasets and annotations, they have been an important source of data for researchers in the domain, either through participation in the challenge themselves, or through further use of the data in later years.

In 2017, when Litjens et al. published their survey on deep learning in medical image analysis [163], they listed 64 publications between 2013 and 2017 on various tasks, including nucleus detection, segmentation and classification; large organ segmentation; and disease detection and classification. They also note how challenges demonstrated the superiority of deep learning methods over "classical" methods, citing the performance of Cireşan et al. [160] in the MITOS12 & AMIDA13 challenges (mitosis detection), alongside the winners of other challenges such as GlaS (gland segmentation), CAMELYON16 (cancer metastasis detection in lymph nodes) and TUPAC16 (tumour proliferation assessment).

The 2018 study by Maier-Hein et al [164] provides a critical analysis of biomedical challenges and their rankings. It demonstrated the lack of robustness of challenge rankings with regard to small variations in the metrics used, in the results aggregation methodology, in the selection of experts for the annotations and in the selection of the teams that are ranked.

In 2020, Hartman et al. published a review of digital pathology challenges [130] listing 24 challenges between 2010 and 2020. Their review focuses on the characteristics of the dataset and the practical value that the challenges have for the pathology community. They note in their conclusion that there is "a disconnect between the types of organs studied and the large volume specimens typically encountered in routine clinical practice" and that this mismatch may "limit the wider adoption of AI by the pathology community." They also emphasize their value through the fact that "a common evaluation method and dataset allow for a better comparison of the performance of the algorithms," and that "public challenges also foster the development of AI by reducing the start-up costs to commence with AI development."

In our own 2022 review [8], we focus on segmentation challenges and on how they create consensus ground truths from multiple experts, how the evaluation metrics, and the transparency (or lack thereof) that challenges offer with regard to their evaluation code, their datasets and the detailed results of the algorithms.

In this section of the thesis, we will review the use of deep learning in digital pathology, mostly through the lens of challenges. We will first look at the different tasks that challenges have proposed to solve (see Table 3.1), and at how they relate to the generic computer vision tasks that we previously defined. We will then look at how the classic deep learning pipeline adapts to the specificities of digital pathology tasks and datasets.

The main source for finding those challenges is the "grand-challenge.org" website. Some additional challenges were found from being mentioned in other challenge publications, in Maier-Hein's study [164], or in "The Cancer Imaging Archive" (TCIA) wiki[16], . The inclusion criteria for this list are that the dataset is composed of histological images (WSI, Tissue Microarray or patches extracted from either), with either H&E or IHC staining.

**Table 3.1. List of digital pathology image analysis challenges and their associated tasks, held between 2010 and 2021.**

| **Name, Year** | **Post-challenge publication or website** | **Task(s)** |
|---|---|---|
| PR in HIMA, 2010 | Gurcan, 2010 [165] | Lymphocyte **segmentation**, centroblast **detection**. |
| MITOS, 2012 | Roux, 2013 [166] | Mitosis **detection**. |
| AMIDA, 2013 | Veta, 2015 [167] | Mitosis **detection**. |
| MITOS-ATYPIA, 2014 | Challenge website | Mitosis **detection**, nuclear atypia **scoring**. |
| Brain Tumour DP Challenge, 2014 | Challenge website | Necrosis region **segmentation**, gliobastoma multiforme / low grade glioma **classification**. |
| Segmentation of Nuclei in DP Images (SNI), 2015 | Description in TCIA wiki | Nuclei **segmentation**. |
| BIOIMAGING, 2015 | Challenge website | Tumour **classification**. |
| GlaS, 2015 | Sirinukunwattana, 2017 [168] | Gland **segmentation**. |

[16] https://wiki.cancerimagingarchive.net/display/Public/Challenge+competitions

| | | |
|---|---|---|
| TUPAC, 2016 | Veta, 2019 [169] | Mitotic **scoring**, PAM50 **scoring**, mitosis **detection**. |
| CAMELYON, 2016 | Ehteshami Bejnordi, 2017 [170] | Metastases **detection**. |
| SNI, 2016 | Challenge website | Nuclei **segmentation**. |
| HER2, 2016 | Qaiser, 2018 [171] | HER2 **scoring**. |
| Tissue Microarray Analysis in Thyroid Cancer Diagnosis, 2017 | Wang, 2018 [172] | Prediction of BRAF gene mutation (**classification**), TNM stage (**scoring**), extension status (**scoring**), tumour size (**regression**), metastasis status (**scoring**). |
| CAMELYON, 2017 | Bandi, 2019 [173] | Tumour **scoring** (pN-stage) in lymph nodes. |
| SNI, 2017 | Vu, 2019 [174] | Nuclei **segmentation**. |
| SNI, 2018 | Kurc, 2020 [175] | Nuclei **segmentation**. |
| ICIAR BACH, 2018 | Aresta, 2019 [176] | Tumour type patch **classification**, tumour type region **segmentation**. |
| MoNuSeg, 2018 | Kumar, 2020 [177] | Nuclei **segmentation**. |
| C-NMC, 2019 | Gupta, 2019 [178] | Normal/Malignant cell **classification**. |
| BreastPathQ, 2019 | Petrick, 2021 [179] | Tumour cellularity assessment (**regression**). |
| PatchCamelyon, 2019 | Challenge website | Metastasis patch **classification**. |
| ACDC@LungHP, 2019 | Li, 2019 [180] | Lung carcinoma **segmentation**. |
| LYON, 2019 | Swiderska-Chadaj, 2019 [181] | Lymphocyte **detection**. |
| PAIP, 2019 | Kim, 2021 [182] | Tumour **segmentation**, viable tumour ratio estimation (**regression**). |
| Gleason, 2019 | Challenge website | Tumour **scoring**, Gleason pattern region **segmentation.** |
| DigestPath, 2019 | Zhu, 2021 [183] | Signet ring cell **detection**, lesion **segmentation**, benign/malign tissue **classification**. |
| LYSTO, 2019 | Challenge website | Lymphocyte **assessment**. |
| BCSS, 2019 | Amgad, 2019 [184] | Breast cancer regions **semantic segmentation**. |
| ANHIR, 2019 | Borovec, 2020 [185] | WSI **registration**. |
| HeroHE, 2020 | Conde-Sousa, 2021 [186] | HER2 **scoring**. |
| MoNuSAC, 2020 | Verma, 2021 [24] | Nuclei **detection**, **segmentation,** and **classification**. |
| PANDA, 2020 | Bulten, 2022 [187] | Prostate cancer Gleason **scoring**. |
| PAIP, 2020 | Challenge website | Colorectal cancer MSI **scoring** and whole tumour area **segmentation**. |
| Seg-PC, 2021 | Challenge website | Multiple myeloma plasma cells **segmentation**. |

| | | |
|---|---|---|
| PAIP, 2021 | Challenge website | Perineural invasion **detection** and **segmentation**. |
| NuCLS, 2021 | Amgad, 2021 [188] | Nuclei **detection**, **segmentation** and **classification**. |
| WSSS4LUAD, 2021 | Challenge website | Tissue **semantic segmentation** from weak, image-level annotations. |
| MIDOG, 2021 | Aubreville, 2022 [189] | Mitosis **detection**. |

## 3.1 Mitosis detection in breast cancer

In 2008, Malon et al. propose what may be the first "deep learning" algorithm in digital pathology [159], with a CNN "loosely patterned" on the LeNet-5 architecture [61]. For mitosis detection, their algorithms still used a "traditional" technique to detect candidate nuclei, and then used the CNN to discriminate between the mitotic and non-mitotic classes. To manage the very large class imbalance, they oversample mitotic examples so that they constitute 20% of the training set. The network alternates between subsampling layers and 5x5 convolutional layers, and they use simple data augmentation based on rotations by 90° increments and symmetrical reflection. They used a slightly updated version of this method [161] in the MITOS12 challenge, which would really start the deep learning era for mitosis detection and for digital pathology.

### 3.1.1 MITOS12 and MITOS-ATYPIA-14 ICPR challenges

The 2012 Mitosis Detection in Breast Cancer Histological Images (**MITOS12**) challenge was the second digital pathology image analysis competition, after the "PR in HIMA" from 2010. Organised by the IPAL Lab from Grenoble with the Université Pierre et Marie Curie and the Pitié-Salpêtrière Hospital in Paris, it is notable as the first to be won by a deep learning method. The IDSIA team of Cireşan et al. won by a comfortable margin from a total of 17 submissions [166]. The challenge used H&E stained 2048x2048px image patches from 50 "High Power Fields", high resolutions area extracted from 5 WSIs. Each WSI was scanned using three different scanners, evaluated separately, with most participants focusing on the "Aperio" scanner images.

Most participants used a classic pipeline much like those discussed in the previous chapter, with an initial candidate selection using thresholding and morphological operations, followed by feature extraction and candidate classification. The best result among those came from the IPAL team of Humayun Irshad [190], which obtained a F1 score of 0.72 on the Aperio set.

Cireşan et al.'s winning method largely surpassed their result with a F1 score of 0.78 on the Aperio images. They used three DCNNs with a (now) classic "classification" architecture, with a sequence of convolutional and max-pooling layers followed by several dense layers applied to 101x101 image patches. The first DCNN was trained on an artificially "balanced" dataset by taking every mitosis example and an equivalent, randomly sampled amount of non-mitosis example ("mitosis example" meaning any 101x101px patch whose central pixel was closer than 10 pixels from an annotated mitosis centroid). This DCNN will have a very large number of False Positives, but it will tend to have a larger probability of mitosis on samples which look similar to mitosis: other nuclei. The output of this DCNN is therefore used to resample the dataset, this time taking all mitosis example and a much larger number (about 14x as many) non-mitosis examples preferably sampled with a weighting factor based on the output of the first DCNN.

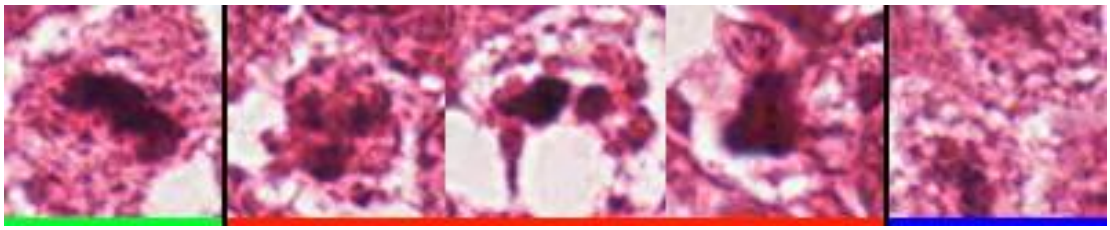


**Figure 3.1. Example of certain mitosis (green), certain non-mitosis (red) and probably mitosis (blue) nuclei from an image patch of the MITOS-ATYPIA-14 dataset.**

Simple data augmentation (rotations and mirroring) is then used and two DCNNs are trained (with the same macro-architecture, but one slightly smaller than the other), and the final prediction on a patch is given by averaging the results of both on 8 variations of the patch (4 rotations with and without mirroring). The predicted result is used to create a "probability map" on a whole 2048x2048 image with a sliding window method. A post-processing step is then used to clean-up the noisy probability map and find local maxima, which are the final "detection" of the algorithm.

This pipeline is slightly different from the classic method, yet some similarities remain. While the final trained DCNNs used for prediction do not have a "candidate selection" step and directly process the entire image, this "candidate selection" idea can still be found in the training step. The results of the ensemble of two DCNNs are also not used directly, as they produce a very large number of false positives: a post-processing step is necessary to obtain a reasonable detection.

The challenge itself, however, suffers from a few important issues that make its results hard to take at face value. First, it uses a very small amount of WSIs, meaning that the generalisation capabilities of any algorithm are hard to measure. Second, it uses only a single expert for the annotations. As we will see in a later chapter, interobserver variability is very high in mitosis detection, so the results based on a single annotator on a small set are going to be very unreliable. Finally, there is a fundamental design flaw in the challenge dataset, in that the split between the training and the test set was done at the "image patch" level and not at the "patient" level, meaning that patches extracted from the same slide are found in both the training and the test set.

This has a very significant impact on the results, as demonstrated by experiments performed by Élisabeth Gruwé during her Master Thesis [191]. Her results show that switching from a cross-validation scheme that mixes the patients to a leave-one-patient-out scheme reduced the score of her method from a 0.68 F1-score to a 0.54 F1-score.

The **MITOS-ATYPIA-14** challenge, by the same organisers, addresses those problems by increasing the number of slides to 16, with 5 WSI set aside for testing. It is unclear from the dataset description if some slides came from the same patient. Annotations were independently provided by two pathologists, with a third pathologist's opinion used to break ties. The supervision files contained information about the potential disagreement by providing a "confidence degree" to each annotated mitosis (see Figure 3.1). No journal publication was made by the 2014 challenge organisers with the results, which are only available on the challenge's website[17].

A post-challenge publication from the winning team of Chen et al. from the Chinese University of Hong Kong [192] explains their method (improved after the challenge), which uses a "cascade" of

[17] https://mitos-atypia-14.grand-challenge.org/Results2/

two patch classification networks. The first "coarse" network is a Fully Convolutional Network, which mostly follows the classic classification macro-architecture, but replaces the "dense" layers by convolutions with a 1x1 kernel, which makes it possible to process an image of an arbitrary size in a single pass. The network is trained with normal dense layers on 94x94px patches, but the dense layers are then converted into 1x1 convolutional layers for the prediction phase so that the network can be applied to the larger image, with each pixel in the output prediction map corresponding to a 94x94px region in the original image. The second, "fine discrimination" model is a patch classification network from Caffe [193], with the weights of the convolutional layers pre-trained on the ImageNet dataset. Three slightly different versions of the model are trained, and their outputs are averaged to provide the final classification.

### 3.1.2 AMIDA13 & TUPAC16

The **AMIDA13** challenge was organised by the UMC Utrecht team of Veta et al. [167]. The dataset contained 2000x2000px image patches extracted from WSIs from 23 patients. Annotations were performed by two pathologists, with two additional pathologists reviewing cases of disagreement. All slides were scanned using an Aperio scanner. As in MITOS12, the IDSIA team of Cireşan et al. was the only "deep learning" approach and won by a clear margin, with an F1-score of 0.61 compared to the next best result of 0.48 by the DTU team of Larsen et al. Cireşan et al.'s method was slightly upgraded from the MITOS12 challenge and used an ensemble of three networks for the prediction, with a post-processing step still necessary.

The **TUPAC16** challenge, with the same organising team [169], moved away from just mitosis detection and required the participants to predict the mitotic scores WSIs, in order to make the clinical relevance of the results more important. The mitosis detection performance was also evaluated separately on selected regions from the WSIs. The scoring task used a dataset of more than 800 WSIs, and the mitosis detection task used more than 100 image patches (including the 23 from the AMIDA13 challenge). Both tasks were won by the Lunit Inc.'s team of Paeng et al. [194], which used a ResNet architecture [106] and a two-pass approach where false positives from the first pass are oversampled with data augmentation in the second pass so that the network sees more difficult cases. The panel of competing methods, however, was very different from the previous challenge, as "[w]ith the exception of one team, **all participants/teams used deep convolutional neural networks**". The deep neural networks, however, were always included in a larger pipeline, and did not process the WSIs directly: a ROI detection step was almost always included, and for the mitotic score prediction the result of the mitosis detection were often combined with other features in classic machine learning classifiers such as SVMs or Random Forests.

One key insight from looking at the MITOS12, AMIDA13, MITOS-ATYPIA-14 and TUPAC16 results is how important the specificities of the dataset are, even on an identical task, when comparing different methods. Cireşan et al. obtained an F1-score of 0.78 on the MITOS12 set and of 0.61 on the AMIDA13 set with an upgraded method. Chen et al's post-challenge publication obtained scores of 0.79 on the MITOS12 set, and of 0.48 on the MITOS-ATYPIA-14 set with the same method used in both. The winner of TUPAC16, meanwhile, obtained an F1 score of 0.65 on the mitosis detection score using a ResNet architecture. This makes the evaluation of the potential for clinical applications of these algorithms extremely difficult. As the ground truth for the test sets were not provided for the MITOS-ATYPIA-14, AMIDA13 and TUPAC16 challenges, post-challenge publications are even harder to compare, as even within the same "dataset" the conditions can differ wildly.

As examples, Nateghi et al. [195] showed excellent results using a traditional image analysis pipeline on the 2012 (F1=0.88), 2013 (0.75) and 2014 (0.84) datasets, but they used a five-fold cross-validation scheme and do not specify whether they took the WSI into account in how they created their splits. Similarly, Albaryak et al. [196] have near perfect results (F1=0.97) on the MITOS-ATYPIA-14 dataset with a combination of classic and deep learning methods, but from their description of how they used the dataset, it seems that they worked a balanced subset of the training data, making the problem considerably easier. It is unfortunate that the only dataset where the test data is available is the MITOS12, where the challenge train/test split did not properly take the WSI into account in the first place.

Several deep learning methods have been proposed in recent years, with results that are, as we just saw, difficult to compare. It is still interesting to look at how the methods have evolved since the four challenges.

Cai et al. [197] use the Faster R-CNN architecture [198], a very common object detection network, with a ResNet-101 encoder pre-trained on the ImageNet dataset. Their results illustrate the good performance that can be obtained by retraining general purpose models to the specific tasks of digital pathology. Faster R-CNN was also used by Mahmood et al. [199] in 2020, also with ImageNet pre-training, but this time reverting to a two-pass approach where the Faster R-CNN is used to detect candidate nuclei, and an ensemble of ResNet and DenseNet classification networks is then used to discriminate between the mitosis and non-mitosis nuclei. The pipeline also includes a classic feature extraction & classification step in the middle to refine the candidate list based on handcrafted features. Cai, Mahmood, and a few other methods mentioned in Mahmood et al. used the same train/test split for the MITOS-ATYPIA-14 dataset, making the comparison between them more accurate. Mahmood et al., with a F1 score of 0.69, obtained the best results. It is interesting to note that traditional image analysis methods clearly still play a role in this particular task in state-of-the-art methods, even though off-the-shelf detection networks are starting to perform reasonably well.

### 3.1.3 MIDOG 2021

The **MIDOG** challenge was hosted at MICCAI 2021[18]. The dataset contains patches extracted from 280 WSIs acquired with four different scanners. In all the previous challenges there was a lot of variability due to the choice of scanners, and most methods focused on the Aperio datasets, where the results tended to be better. MIDOG chose not to evaluate the different scanners separately, but instead to focus on the "domain generalization" capabilities of the algorithms, pushing participants to create a single model capable of detecting mitosis from any scanner. Two teams obtained essentially equal results: the Tencent AI team of Yang et al. [200] and the TIA Lab team of Jahanifar et al. [201]

Yang et al. use a pre-trained HoVer-Net to detect candidate cells. To help with domain adaptation, they augment the dataset by swapping the low-frequency information in Fourier space of different images, to create new images containing the same high-frequency information (which will contain the object borders) but in the "style" of the images from which the low frequencies (which will contain the overall "background" colour information) were extracted. A SK-UNet [202] segmentation architecture is used to segment the mitotic cells, and a post-processing step cleans-up the segmentation and find the centroid of the object's bounding boxes as a final detection.

[18] https://midog2021.grand-challenge.org/

Jahanifar et al. use a stain normalization technique [203] as a pre-processing step, matching the colour space of all images to the same target tile. An Efficient-UNet segmentation model, pretrained on ImageNet, was then used to segment the mitoses. Under-sampling of the negative examples was used during training to provide a balanced dataset to the model. This first network provides a set of segmented candidate nuclei, with many false positives remaining. An Efficient-Net-B7 classification model is then used to further discriminate between mitosis and non-mitosis. An ensemble of three models trained with different subsets of the data during cross-validation is used for inference as the final prediction.

It is clear from this latest challenge that the overall pipeline of mitosis detection remains remarkably similar from one challenge to the next, with the addition of a colour normalization or specific data augmentation step in MIDOG. A two-step process with candidate selection followed by a classification network to discriminate between the candidates seems to invariably provide the best results. From the MIDOG result, we can see a certain shift that occurred in the choice of macro-architectures after 2015 and the introduction of the U-Net architecture for biomedical image segmentation [66]. Both methods use some variations of U-Net, even though the problem was initially framed as a detection task and no ground truth segmentation was provided by the organisers. The other networks used in addition to the U-Net were also repurposed, with the "generic" Efficient-Net [204] model on the one hand, and the HoVer-Net model [103], designed for nuclei segmentation and classification in digital pathology, on the other. It is interesting to note that the Tencent AI team used the HoVer-Net, which was designed by the TIA Lab, that opted for the Efficient-Net in this case.

## 3.2 Tumour classification and scoring

Being able to directly assess tumour malignancy from WSIs is an important area of research for deep learning in digital pathology, with many challenges and publications exploring different aspects of this problem. Some of the mitosis detection challenges mentioned above already moved in that direction, with MITOS-ATYPIA-14 looking for nuclear atypia score, and TUPAC16 evaluating the mitotic score instead of the raw mitosis detections. There are two distinct approaches to the definition of these "tumour scoring" tasks: the first is to define very generic categories, such as benign / malign, and to classify image patches or WSIs accordingly. The other approach is to ask the algorithms to replicate some specific grading criteria used by pathologists.

### 3.2.1 Generic classification

One of the first applications of deep learning in digital pathology came from Cruz-Roa et al. in 2013 [162]. Their method proposed to perform patch-level binary classification ("non-cancer" vs "cancer") in H&E stained Basal Cell Carcinoma (BCC) tissue. They used a convolutional auto-encoder attached to a softmax classifier to produce the cancer / non-cancer probability prediction. They also recognized a very important aspect of such "high-level" tasks, which is the necessity for explainability. In order to be trustworthy, a deep model's prediction needs to be able to show which components of the image were considered relevant for the prediction. This makes it possible to review the results and determine whether the model has learned features that make sense from a pathologist's perspective, or if it may be influenced by confounding factors. To solve this issue, Cruz-Roa et al. produce a "digital staining" on the patch by combining the feature maps of the auto-encoder with the weights of the prediction layer, so that the regions of the image that contributed to the "cancer" or "non-cancer" prediction can be visualized (see Figure 3.2).

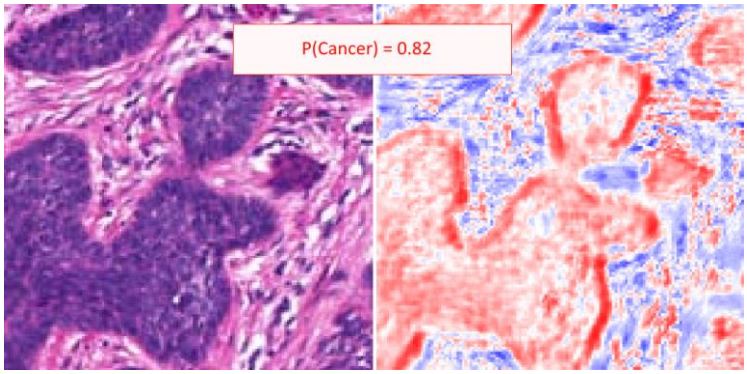



**Figure 3.2. Image patch (left), digital stain (right) and patch-level prediction (top) for BCC classification. Image adapted from Cruz-Roa, 2013** [162]**. In the digital stain, red indicates region that positively contributed to the cancer prediction, blue to the non-cancer prediction.**

The next year, Cruz-Roa et al. published another deep learning pipeline [205] using a CNN to classify all the tissue patches extracted from breast cancer WSIs as "invasive ductal carcinoma" (IDC) or not. The result is an "IDC" probability map on the entire WSI. Cruz-Roa et al. compared their relatively simple DCNN method with several traditional methods based on handcrafted features and found a clear improvement from the deep learning method.

Several challenges have since then been held with these types of "generic" categories. **BIOIMAGING 2015** used 2048x1536px image tiles extracted from H&E stained biopsy sample WSIs, classified as either normal tissue, benign lesion, carcinoma *in situ* or invasive carcinoma. The team of Araújo et al. [206] (which includes the organisers of the challenge) obtained the best results with a typical DCNN classifier. They split the large image tiles into smaller patches which were individually classified, then perform a majority vote on the patch predictions to determine the prediction on the whole image tile.

A few years later, the **BACH 2018** challenge extended the BIOIMAGING dataset and evaluated tile-level classification at the same time as WSI-level semantic segmentation based on the same four classes. There were 500 2048x1536px tiles in the classification dataset, and 40 WSIs in the segmentation dataset. The best results on both tasks were obtained by Kwok et al. [207], who used an Inception-ResNet-v2 model [208] pre-trained on ImageNet. For the segmentation task, they simply stitch together the patch-level results of all tissue patches in the WSI. To improve the results in the WSI part of the challenge, they first used the network trained on the classification task only to find the "hard examples" in the WSIs (patches for which the prediction is largely different from the ground truth), and to retrain the network specifically on those hard examples. Another method by Chennamsetty et al. [209] obtained equal results on the classification task but did not participate in the segmentation task. They used an ensemble of ResNet-101 and DenseNet-161, pre-trained on the ImageNet dataset. They normalized the images using two different normalization parameters: one based on the statistics of the ImageNet dataset used in the pre-training, and one based on the statistics of the BACH dataset. Some of the networks in the ensemble are trained using the first scheme, others with the second.

### 3.2.2 Scoring systems

Aside from the mitotic score, challenges have been organized to replicate several other tumour grading schemes. The **HER2 2016** and **HeroHE 2020** challenges, for instance, both attempt to assess the level of HER2 protein in breast cancer. This is important to guide the choice of treatment, as "HER2-positive" cancers require drugs that specifically target that protein[19]. The **Tissue Microarray Analysis in Thyroid Cancer Diagnosis 2017** challenge, meanwhile, asked participants to determine several indicators used to assess thyroid cancer: presence of the BRAF gene mutation, TNM stage, tumour size, metastasis score, and extension score. Metastasis score was also the target of the **CAMELYON17** challenge, while **BreastPathQ 2019** proposed a regression task of "tumour cellularity assessment". A particularly challenging scoring system that we presented in the previous chapter is Gleason scoring for prostate cancer. Two challenges have been held for that task: the **Gleason 2019** challenge and the **PANDA 2020** challenge.

The **Gleason 2019** challenge asked participants to segment and grade the Gleason "patterns" present in prostate cancer Tissue Microarray (TMA) spots. From these pattern grades, a per-TMA Gleason score also had to be predicted, and the participants were evaluated based on a mix of both predictions. An interesting aspect of that challenge (which will be explored more thoroughly further in this thesis) is that the organisers provided individual annotation maps from six expert pathologists, thus including the interobserver variability into the training set itself. The winning method from the challenge, by Qiu et al. [210] (with code released by Yujin Hu[20]), uses a PSPNet [211] architecture trained on consensus annotation maps derived from the individual expert's annotations with the STAPLE algorithm [212], which was also used by the challenge organisers to build consensus maps for the evaluation. No particular pre- or post-processing is applied, and the final per-TMA prediction is directly computed from the pattern found in the image according to the Gleason scoring rules.

The **PANDA 2020** challenge used a very large set of WSIs (12.625) from six different sites, and only required the final per-WSI Gleason score. With more than 1.000 participating teams, many teams were able to get very similar results at the top of the standings. The post-challenge publication [187] identifies three key features of top-ranking methods:

a) A "bag of patches" approach, where the encoder part of the DCNNs concatenates features from patches sampled from the WSI, so that the discriminator part of the network sees features from multiple parts of the WSI for the classification.
b) To account for potential errors in the annotations, which are necessarily noisy in such a large dataset, top-ranking methods included a "label cleaning" step wherein samples which diverged too much from the predictions in the training set were iteratively relabelled or excluded before retraining.
c) A generalized usage of ensembles of different models, combining different pre-processing approaches and/or network architectures so that the average of the predictions would smoothen any bias of the individual pipelines.

Data augmentation was largely used, but the challenge results make it difficult to determine which types of data augmentation were more effective, with some of the top teams including colour augmentation and affine transformations on the WSI, while others only performed simple spatial

[19] https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/breast-cancer-her2-status.html
[20] https://github.com/hubutui/Gleason

transforms at the patch level. The most commonly used network architectures were EfficientNet [204] and ResNeXt [107] variants.

## 3.3 Detection, segmentation, and classification of small structures

While the ability to directly grade a tumour has the potential to be clinically very useful, a big problem of such "high-level" tasks is that they do not necessarily encourage explainability, as Cruz-Roa already realized in his 2013 paper [162]. Many challenges therefore focus on a more "bottom-up" approach, where the goal is to detect, segment and classify small structures such as different types of cells or nuclei, metastasis, etc., which can then potentially be used to assess the tumour in a way that ensures that the criteria used for the grading are in line with the pathologist's criteria. The first digital pathology challenge, **PR in HIMA 2010**, focused on lymphocyte segmentation and centroblast detection. Lymphocytes were also the target of the **LYON 2019** challenge and the **LYSTO 2019** hackathon. The **CAMELYON 2016** and **PatchCamelyon 2019** targeted metastasis, **DigestPath 2019** looked at the detection of signet ring cells, **Seg-PC 2021** was about the segmentation of multiple myeloma plasma cells, and **PAIP 2021** focused on the detection and segmentation of perineural invasion.

The most common targets, however, are cell nuclei. Between 2015 and 2018, a yearly **Segmentation of Nuclei in Digital Pathology Images (SNI)** challenge was organized by the Stony Brook Cancer Center and hosted at the MICCAI workshop in Computational Precision Medicine. While few information remain about the 2015 and 2016 editions, post-challenge publications are available for the 2017 [174] and 2018 [175] editions. Nuclei segmentation was also one of the tasks used as an example of deep learning in digital pathology in Janowczyk and Madabhushi's 2016 "tutorial" [148]. **MoNuSeg 2018** looked at nuclei segmentation in multiple organs, and its successor **MoNuSAC 2020** added a classification aspect, with four types of nuclei being identified (epithelial, lymphocyte, neutrophil and macrophage). A more recent benchmark dataset, **NuCLS 2021**, largely extended the classification part, with a whole hierarchy of classes and super-classes being identified (see Figure 3.3). Additionally, that dataset includes "multi-raters" annotations, where individual annotations from the different experts are available, allowing for a deeper study of interobserver variability. Other large nuclei datasets have been proposed in recent years, with PanNuke [213] proposing samples from 19 tissue types, while CoNSeP [103] and Lizard [214] focus on colon tissue (all three datasets were produced under the supervision of Nasir Rajpoot of the TIA Lab at the University of Warwick).

The winning methods from SNI 2017, by the Sejong University team of Vu et al. [174], used a classic encoder-decoder segmentation architecture with "residual" short-skip connections and "U-Net like" long-skip connections, with a ResNet-50 forming the backbone of the "encoder" part. Three different networks are trained at different scales and their results are aggregated through some additional layers, with the whole architecture repeated twice: once to segment the inside of the objects, and the other to segment the borders. A post-processing step then takes these two outputs and uses the watershed algorithm to get to the final segmentation and labelling of individual nuclei.

By contrast, the winning method of the 2018 edition by the Shanghai Jiao Tong University team of Ren et al. [175] used a much more generic approach with a Mask R-CNN network (designed for detection and segmentation), pre-trained on the MSCOCO dataset, with some morphological post-processing to clean-up the results.

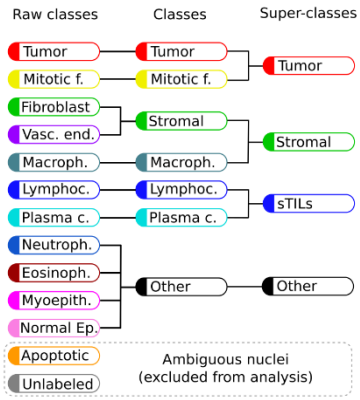



**Figure 3.3. Hierarchical organisation of the classes identified in NuCLS 2021** [188]**, from the dataset's website[21].**

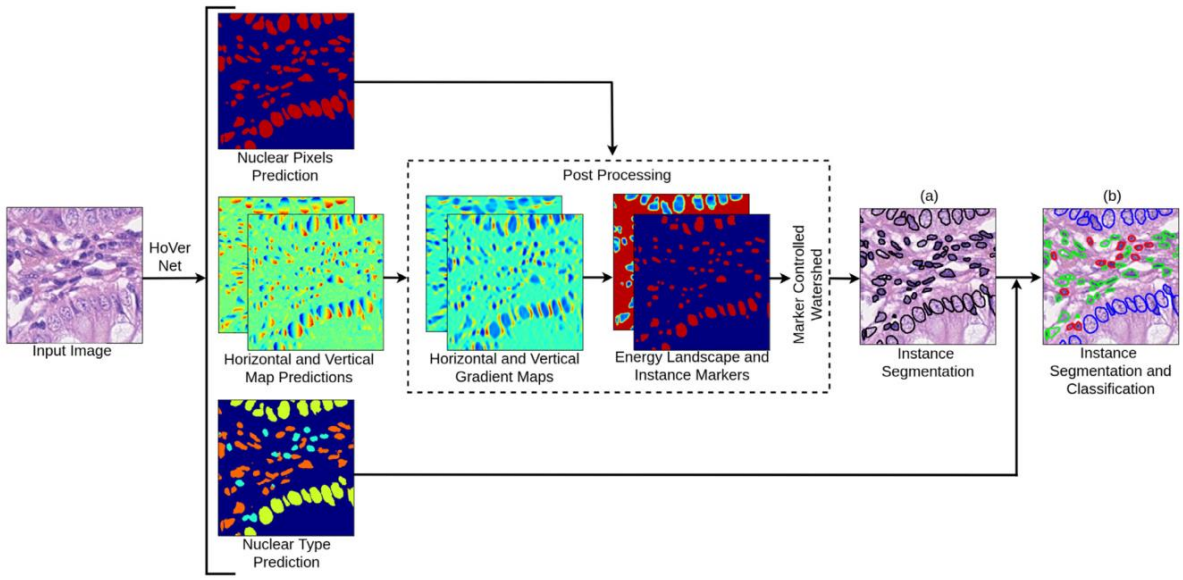



**Figure 3.4. Overall structure of the HoVer-Net architecture, with a segmentation path, a "vector" to the centre of the nuclei path, and a classification path. Image from Graham, 2019** [103]**.**

The Chinese University of Hong Kong team of Zhou et al, winners of MoNuSeg 2018 [215], reverted to a method similar to SNI 2017, a "Multi-Head Fully Convolutional Network" with distinct paths for prediction of the nucleus interior and contours, using a U-Net like architecture and short-skip connections. The encoder part of the architecture is shared between the two paths, but the decoders are different. They also used stain normalization as a pre-processing step.

A slightly different upgrade on that concept is proposed by Graham et al. with their HoVer-Net [103] architecture, that won the MoNuSAC challenge in 2020 [216]. As MoNuSAC added nuclei classification to the instance segmentation task, HoVer-Net adds a third classification path to the network. Another difference is that instead of predicting the contours, HoVer-Net predicts a vector pointing towards the centre of the nuclei, which also helps to separate the nuclei that are fused together by the segmentation path (see Figure 3.4).

[21] https://sites.google.com/view/nucls/data-generation?authuser=0

## 3.4 Segmentation and classification of regions

Aside from the small objects such as nuclei, digital pathology image analysis tasks can also target larger regions to segment, such as necrosis region (**Brain Tumour Digital Pathology Challenge 2014**), glands (**GlaS 2015**), tumour region (**PAIP 2019-2020, ACDC@LungHP 2019**) or lesions (**DigestPath 2019**). This segmentation can also be associated with a classification of each region, such as in the segmentation task of **BACH 2018** where the algorithm had to find benign, in situ and invasive tissue regions, **Gleason 2019** for Gleason patterns, or **BCSS 2019**, where regions have to be classified as tumours, stroma, inflammatory, necrosis, etc.

Some of the best methods proposed for those challenges use similar ideas to those previously discussed for the nuclei. For instance, the Chinese University of Hong Kong team of Chen et al. [102] won the **GlaS 2015** challenge with a "Deep Contour-Aware Network" (DCAN) that used the "inside / border" twin decoder paths, the same idea they later used for their previously discussed MoNuSeg 2018 entry. Another participant worth mentioning for the GlaS challenge are the fourth place finisher from Freiburg University with the U-Net architecture that they had just introduced [66], winning the "Cell Tracking Challenge 2015" and showing results better than the state-of-the-art at the time in a neuron segmentation in electron microscopy ISBI challenge. U-Net has had a lot of success in biomedical image segmentation tasks and has been very influential in the development of deep learning methods in digital pathology. As Ronneberger et al. made their code publicly available, it was widely re-used, adapted and re-purposed in the following years [202], [217], [218].

As the classes in some of these problems are ordered, they can also be treated as "regression" problems. This is done by Kwok et al.'s winning entry to the **BACH 2018** challenge [207], that we already presented in section 3.2.1 as they treated the WSI semantic segmentation problem as a per-patch classification problem, then stitching the patch-level predictions together. In the more recent 2019-2020 challenges, we mostly see variations on the U-Net and/or ResNet architectures, often with ensemble of networks being used.

## 3.5 Public datasets for image analysis in digital pathology

The largest resource of digital pathology WSIs is provided by the US National Cancer Institute, in the "TCGA[22]" (The Cancer Genome Atlas) and "TCIA[23]" (The Cancer Imaging Archive) databases. The TCIA histopathology archive[24] contains images from around 5.000 human subjects (and about 300 canines) from various organs, with some containing clinical data or expert annotations. The TCGA repository, meanwhile, includes about 30.000 slide images, with no expert annotations. These repositories are very commonly used by challenges as the primary source for the images, with experts from the challenge organisation providing the annotations. In the challenges analysed in our review of segmentation challenges [8] (which includes all the segmentation challenges listed in Table 3.1), about half of the challenges used TCGA and/or TCIA images.

Most digital pathology challenges are now hosted on the grand-challenge.org website, which is maintained by a team from Radboud University Medical Center in the Netherlands. Their training

[22] https://portal.gdc.cancer.gov
[23] https://www.cancerimagingarchive.net/
[24] https://www.cancerimagingarchive.net/histopathology-imaging-on-tcia/

data are often publicly available with the corresponding annotations. Many challenges, however, keep their testing data annotations private even after the challenge ended.

Some additional WSI datasets (mostly without annotations) are listed on the website of the Digital Pathology Association[25]. These datasets are generally aimed at pathologists for education purpose rather than for training image analysis algorithms.

Other research groups have made datasets from their publications available. Notable examples are the "Deep learning for digital pathology image analysis" tutorial datasets[26] of Janowczyk and Madabhushi [148], and the datasets from the TIA Warwick publications and challenges[27].

This thesis includes several experiments that used publicly available datasets. A complete description of these datasets is presented in Annex A.

## 3.6 Summary of the state-of-the-art

The different methods that were mentioned in the previous sections are referenced in Table 3.2. In this section, we will summarize the main characteristics of the state-of-the-art methods in digital pathology in this era of "deep learning" solutions.

**Table 3.2. Selection of publications on deep learning methods for digital pathology.**

| Reference, Year | Task type | Model / Method | Challenge / Dataset |
|---|---|---|---|
| Malon, 2008 & 2013 [159], [161] | Detection | LeNet-5 | MITOS12 |
| Cireşan, 2013 [160] | Detection | Cascade of DCNNs | MITOS12, AMIDA13 |
| Cruz-Roa, 2013 [162] | Classification | Convolutional AE + Softmax classifier | Private Basal-Cell Carcinoma dataset |
| Cruz-Roa, 2014 [205] | Classification | Classic classification macro-architecture. | Private Invasive Ductal Carcinoma dataset |
| Ronneberger, 2015 [66] | Segmentation | U-Net | GlaS, others. |
| Chen, 2016 [192] | Detection | Cascade of DCNNs | MITOS12, MITOS-ATYPIA-14 |
| Paeng, 2016 [194] | Detection, Scoring | ResNet + SVMs | TUPAC16 |
| Araújo, 2017 [219] | Classification | Classic classification macro-architecture. | BIOIMAGING15 |
| Chen, 2017 [102] | Instance segmentation | Deep Contour-Aware Network / Multi-Head FCN (border / inside paths) | GlaS 2015, MoNuSeg 2018 |
| Kwok, 2018 [207] | Classification | Inception-ResNet-v2 | BACH18 |
| Cai, 2019 [197] | Detection | Faster R-CNN | MITOS-ATYPIA-14, TUPAC16 |

[25] https://digitalpathologyassociation.org/whole-slide-imaging-repository
[26] http://www.andrewjanowczyk.com/deep-learning/
[27] https://warwick.ac.uk/fac/cross_fac/tia/data/

| | | | |
|---|---|---|---|
| Graham, 2019 [103] | Instance segmentation and classification | HoVer-Net | MoNuSAC, CoNSeP |
| Vu, 2019 [174] | Instance segmentation | U-Net-like with Residual Units, multi-scale input, separate "border" and "inside" paths | SNI 2017 |
| Mahmood, 2020 [199] | Detection | Faster R-CNN, ResNet, Densenet | MITOS12, MITOS-ATYPIA-14, TUPAC16 |
| Kurc, 2020 [175] | Instance segmentation | Mask R-CNN with morphological post-processing | SNI 2018 |
| Yang, 2021 [200] | Detection | HoVer-Net + SK U-Net | MIDOG |
| Jahanifar 2021 [201] | Detection | Efficient-UNet + Efficient-Net | MIDOG |

### 3.6.1 Pre-processing

A common pre-processing step in digital pathology pipelines is **stain normalisation** [11], [143], [220]. Stain normalisation aims at reducing the differences between images that are due to variations in the staining process or in the acquisition hardware and setup (see Figure 3.5). Several top-ranking challenge methods include this step in their pipeline (such as the MoNuSeg 2018 and PAIP 2020 winners). In most cases, however, researchers prefer to let the deep neural network become invariant to stain variations, often with the help of colour jittering in the data augmentation (see below). It remains more common, however, to perform a normalisation step in RGB space, either to **zero-centre** the pixel data and set the per-channel variances to one, or to simply **rescale** the value range to [0, 1] or [-1, 1].

As many deep neural network architectures require fixed-sized input images, it is very common to see a **patch extraction** step at the beginning of the pipeline. For inference, these patches can either be non-overlapping tiles with independent predictions (e.g. used by the winner of BACH 2018) or overlapping tiles in a sliding window process, where the results in the overlapping region have to be merged, typically with either the average prediction or the maximum prediction (e.g. winner of MoNuSeg 2018). The details of this operation are left unclear in many of the published methods.

### 3.6.2 Data augmentation

Some form of data augmentation is explicitly included in almost every method that were mentioned in this chapter, although the level of details on which operations are done vary wildly. Almost every method includes basic affine transformations, vertical and horizontal flips and random crops. It is also very common to include elastic distortions, random Gaussian blur or noise, scaling, or brightness/contrast variations (see Figure 3.6).

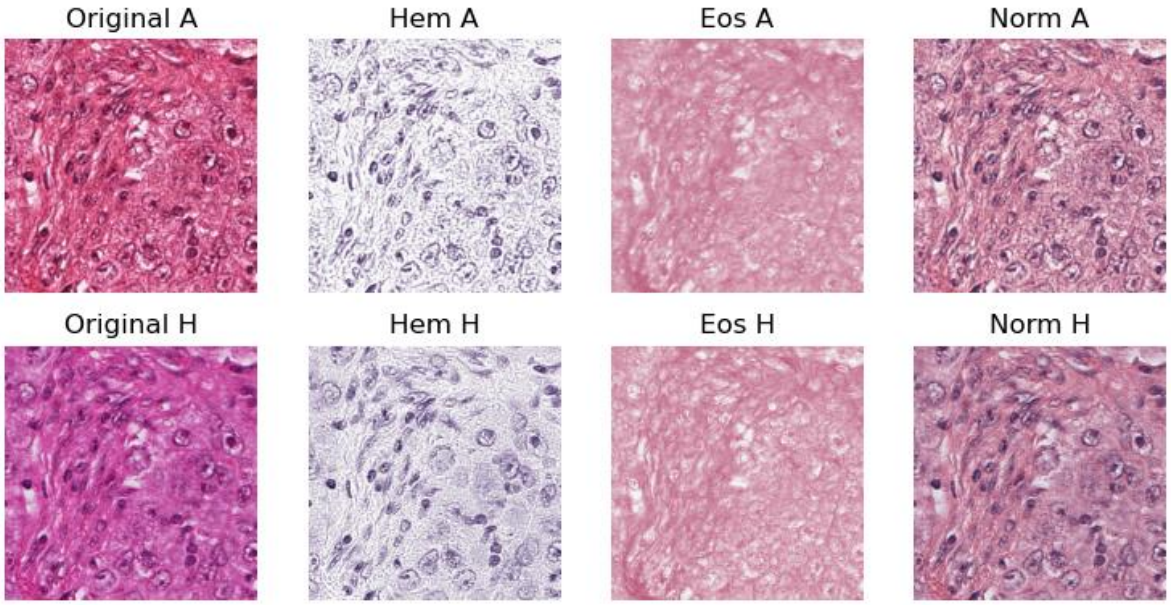



**Figure 3.5. Example of stain normalisation on a patch extracted from the MITOS12 challenge dataset. On the left are two patches from the same region of an image acquired with an Aperio scanner (A) and a Hamamatsu scanner (H), on the right are the two images after normalisation using the method from Anghel et al.** [220]**. In the middle are the separated Haematoxylin and Eosin stain concentrations.**

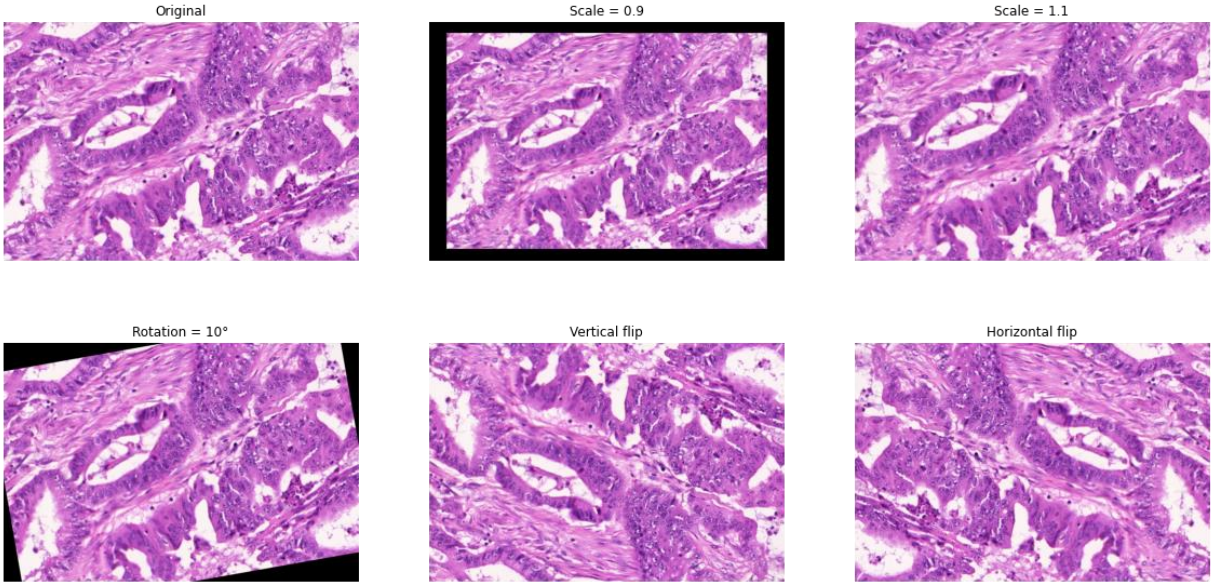



**Figure 3.6. Example of basic transformations for data augmentation on an image from the GlaS 2015 challenge.**

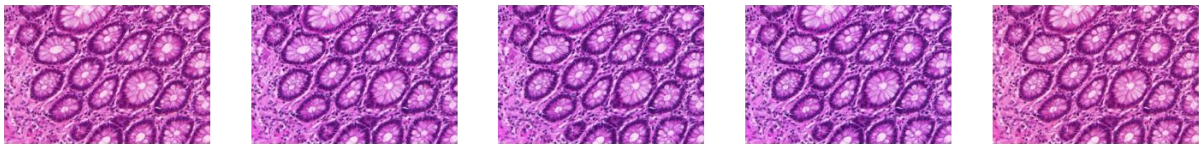


**Figure 3.7. Example of random colour augmentation on an image from the GlaS 2015 challenge. Left image is the original, the four others have had their "Hue" channel value shifted by a random value.**

As mentioned above, colour data augmentation is also a common practice, often done in the HSV space (as in the example in Figure 3.7) or in a transformed, stain-specific colour space. For instance, "colour deconvolution" can be used to find each stain's "colour vector", which can then be used either to normalize the stains (by matching the stain's vectors of an image to a reference target) or to perform realistic colour augmentation [145], [146]. While it is clear from challenge results and from other studies [146] that there is a very large consistent improvement in using basic morphological augmentation such as affine and elastic transforms over no data augmentation, the impact and importance of adding more complex augmentation methods is harder to assess and may depend a lot more on the specificities of a given dataset (e.g., from single or multiple source(s)) and application.

### 3.6.3 Network architectures

The results of digital pathology image analysis challenges and the evolution of the state-of-the-art make it very difficult to get to any sort of conclusion on which architecture(s) work best for which type of task. Some trends, however, can be noted. Until 2015, most challenges and publications focused on classification and detection tasks, and often used a **classic classification macro-architecture**, with several convolutional and max-pooling layers for feature extraction followed by some dense layers for discrimination. Those architectures are very close to the "pioneering" LeNet-5 and AlexNet, with some micro-architectural adaptations that the methods often do not really justify, nor systematically test their impact [161], [205], [219]. The main adaptation for the specificities of digital pathology problems come in the pre-treatment of the data, and the popularity of the "**cascading**" approach for highly imbalanced problems such as mitosis detection, as exemplified by Cireşan et al. [160] and Chen et al. [192] in their MITOS12, AMIDA13 and MITOS-ATYPIA-14 winning methods.

The situation slightly evolved after 2015 and the introduction of **U-Net** [66] and **ResNet** [221], which quickly became the standard "baseline" for segmentation and classification, respectively. As popular software libraries such as Tensorflow and PyTorch became available, so did open-source code for network architecture and even the weights of networks pre-trained on popular general-purpose datasets such as ImageNet. A standard approach to image analysis problems therefore became to **start with a pretrained network and fine-tune it on the task-specific dataset**. Digital pathology tasks, however, often require some additional tweaks, with many solutions adding **multiple outputs** (such as object borders and object inside [103], [174], [222]) or using **ensemble of networks** trained on subsets of the data, on multiple scales, or using different architectures [199]–[201]. In segmentation problems, however, our review noted that "when the methods are published with detailed results for the individual components, the improvement due to the ensemble over the best individual network is usually small (although ensemble methods do seem to perform consistently better, but generally without statistical validation)" [8].

Digital pathology detection problems appear to be challenging for classic detection networks such as Faster R-CNN or instance segmentation networks like Mask R-CNN, which need to either be included in an ensemble [199] or require significant post-processing [175].

In general, it seems that apart from very specifically designed networks such as HoVer-Net [103], very good results can generally be obtained with standard, off-the-shelf networks, with most domain-specific adaptations coming either in the pre- or post-processing stages.

Three different network architectures have been used (with some variations) in the different experimental works of this thesis. Their description can be found in Annex B, alongside those of the most commonly used networks in the literature.

### 3.6.4 Pre-training and training

As previously mentioned, many challenges top ranked methods use pre-trained networks, often from **general-purpose datasets** such as ImageNet, PASCAL VOC or ADE20K. Another option, used for instance by the winners of MoNuSAC 2020, is pretrain the network on data from **similar datasets** (from previous challenges, or other benchmark datasets).

In the methods analysed in our review, the training or re-training itself is generally done using the **Adam optimizer** (or some adaptation, such as the Rectified Adam), with the **cross-entropy** loss function or, alternatively, the "**soft Dice**" loss (or a combination of both).

To deal with the recurring problem of **class imbalance**, several methods have been proposed. One option is to **weights the classes** in the loss function so that errors on minority classes are more heavily penalized. Another is to use an **adapted loss function** such as the Focal Loss [84], which gives more weights to hard examples in the training set. A more direct approach is to **balance the batches** presented to the model by sampling the patches so that minority classes are "over-represented" compared to the actual distribution. **Data augmentation** can also be used to **rebalance the dataset**, by producing more "augmented" examples from the minority classes than from the majority classes.

### 3.6.5 Post-processing and task adaptation

Multi-outputs model typically require a post-processing step to put together the different parts of the results. A clear example of that can be found in the HoVer-Net model previously illustrated in Figure 3.4. The outputs of the "nuclear pixels prediction" branch are first combined with the outputs of the "horizontal and vertical map predictions" (which predict the vector to the centre of the nucleus) to compute an "energy landscape" and a set of markers that can be used to perform the watershed algorithm, thus providing the "instance segmentation". The class of each segmented object is then determined by taking a majority vote from all per-pixel class prediction of the "nuclear type prediction" branch within the pixels of the object, to give the final "instance segmentation and classification".

Another very common step is the stitching of patch predictions into a larger image, particularly for WSIs. In a classification task, this may take the form of a majority vote (where the WSI-level prediction is the majority vote of the per-patch prediction), but other task-specific rules may be used (for instance, in a grading problem, the final WSI grade may be the "worst"[28] of the per-patch grades). In a segmentation task, an important choice is whether to include overlap or not in the tiling of the patches. If overlapping tiles are used, the final per-pixel prediction may be the majority

[28] i.e. the highest one, as the grade is positively correlated with the malignancy level.

vote of all patch predictions that included that pixel, but other heuristics may be applied. The fact that segmentation predictions tend to be worse close to the borders of a patch, for instance, may be used to weights the pixel prediction of a patch according to how close that pixel was to the centre of the patch.

## 3.7 The deep learning pipeline in digital pathology

With all these information in mind, we can now look at how the specific characteristics of digital pathology impact the deep learning pipeline that was presented in Chapter 1. Figure 3.8 illustrates the pipeline, and where the different elements that will be discussed in the rest of the thesis intervene.

In Chapter 4, we will study the **evaluation processes** that are used, in challenges and in other publications, to assess the performances of deep learning algorithms. This includes not only the choice of evaluation metrics, but also the process of aggregating the results while considering the specificities of digital pathology datasets (such as the intra-patient dependence of samples), and the methods for ranking or otherwise comparing competing algorithms. We perform several experiments and theoretical analyses to better understand the biases of commonly used metrics, and the potential effects of class imbalance in classification and detection metrics.

In Chapter 5, we will investigate the question of how **imperfect annotations** affects the training process of deep learning model, through our **SNOW** (Semi-supervised, Noisy and/or Weak) framework. A practical example, the problem of **artefact segmentation** in whole-slide images, will be studied in Chapter 6. Artefacts that occur in the manipulation of the physical samples and in the acquisition of the images are very common in digital pathology. Segmenting these artefacts so that they can be safely removed from further automated methods, and so that the quality of the produced WSI can be objectively assessed, is a good practical example of the different challenging aspects of digital pathology datasets.

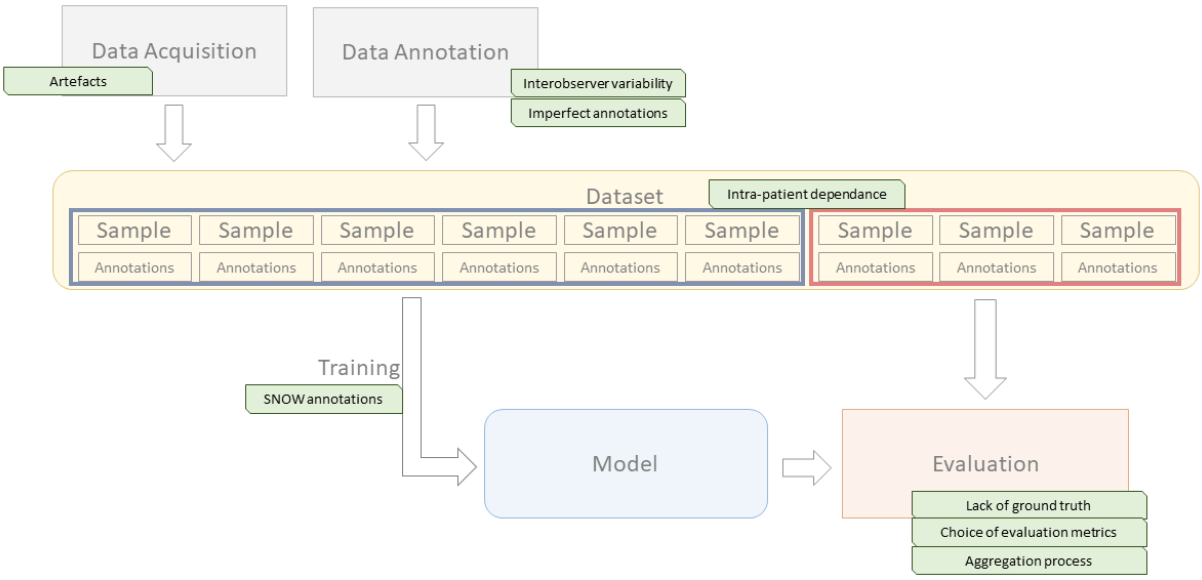



**Figure 3.8. The deep learning pipeline, with the specific characteristics of digital pathology datasets that will be further explored in the rest of the thesis.**

In Chapter 7, the more specific question of **interobserver variability** will be analyzed. Even if individual experts could provide their "perfect" annotations (meaning that the annotated samples perfectly correspond to their assessment of the images), the fact that individual experts would reach different assessments means that there can be no ground truth in a digital pathology task. This has an impact both on the training process, and on the evaluation process of deep learning algorithms.

Finally, in Chapter 8, we will study the question of **quality control in challenges**. This is transversal to the whole pipeline, as errors can occur anywhere from the constitution of the dataset to the evaluation process. Through the analysis of selected examples, we will examine the types of error that can undermine the trust that we can have on the results presented by challenges and provide some recommendations for future challenge organisers on how to ensure that the insights that can be gained from their efforts are maximized.

# 4 Evaluation metrics and processes

In challenges or in any publication that aims to determine if a particular method improves on the state-of-the-art for a particular task, the question of how to evaluate the methods is extremely important. At the core of any evaluation process, there is one or several **evaluation metrics**. Different metrics have been proposed and have achieved a more-or-less standard status for all types of tasks, and many studies have been made to examine their behaviour in different circumstances. Moreover, the evaluation process is not limited to the metric itself.

In this chapter, we will start in section 4.1 by providing the necessary definitions for the description of the evaluation process of digital pathology image analysis tasks, and of the common (and less common) evaluation metrics used for detection, classification, and segmentation tasks, as well as for more complex tasks that combine these different aspects. The use of these metrics in digital pathology challenges will then be discussed in section 4.2, and the state-of-the-art of the analysis of their biases and limitations in section 4.3.

We will then present our additional analyses and experiments in section 4.4, and provide recommendations on the choices that can be made when determining the evaluation process for a digital pathology task in section 4.5.

## 4.1 Definitions

Most of the formal definitions of the metrics used in this section are adapted from several publications and reworked to use the same mathematical conventions. For the detection metrics, we largely rely on the works of Padilla et al. [223], [224]. Most of the classification metrics can be found in Luque et al. [225], while the definitions for the segmentation metrics are given in Reinke et al. [226]. The evaluation process in general described here is largely based on our analysis of the processes described in all the digital pathology challenges referenced in section 4.2.

### 4.1.1 Evaluation process

The evaluation of an algorithm in an image analysis task starts from a set of "target" samples (usually from expert annotations and considered to be the "ground truth"), associated to a set of predictions from the algorithm. In the final evaluation of an algorithm, the samples would come from the "test set" extracted from the overall dataset, which should be as independent as possible from the training and validation set. The exact nature of the samples will depend on the task.

Digital pathology datasets have a hierarchical nature, with the top-level being the patient. From each patient, different whole-slide images (WSI) may have been acquired. From each of those WSIs, different image patches may have been extracted, and often each of these patches will contain a set of individual objects of interest (as illustrated in Figure 4.1).

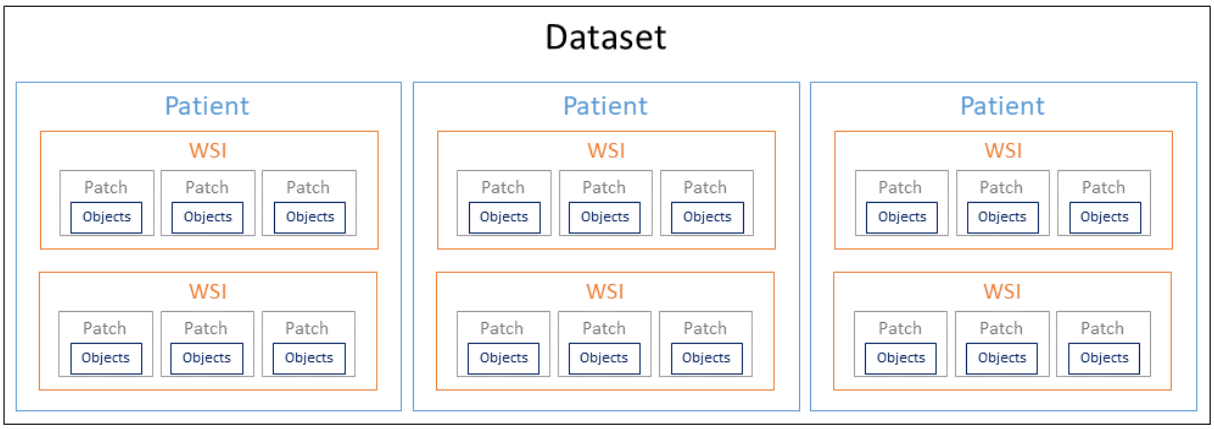



**Figure 4.1. Hierarchical representation of a typical digital pathology dataset. Evaluation metrics can be computed and/or aggregated at these different levels.**

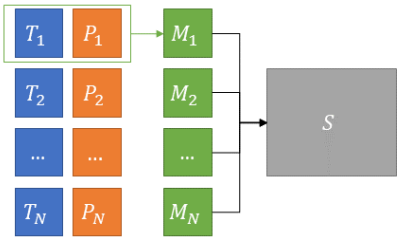



**Figure 4.2. Illustration of the evaluation process. Metrics $M$ are computed on pairs of "Target" ($T$) / "Predicted" ($P$) items, then aggregated to produce a final score $S$.**

In the DigestPath 2019 challenge[29], for instance, two different datasets are described. The "signet ring cell dataset" contains objects (the signet ring cell) in 2000x2000px image patches extracted from WSIs of H&E stained slides of individual patients. In the "colonoscopy tissue segment dataset", however, WSIs have been directly annotated at the pixel-level, so there are no "patch-level" or "object-level" items.

A **metric** is thus defined as a function $M = Metric(T, P)$ that evaluates a pair of "target" (ground truth) and "predicted" items (e.g. object localization, patient diagnostic class…). Most metrics will produce outputs either in the $[0, 1]$ or the $[-1, 1]$ range, but there are exceptions (such as distance-based metrics, which are typically in $[0, \infty[$ ). The **aggregation process**, on the other hand, describes how the set of metric values $\{M_i\}$ at a given level is reduced to a single score $S$ at a higher level (see Figure 4.2).

With these definitions in mind, we can now turn our attention to the three main "task types" that we previously identified in Chapter 1, as well as their combinations.

In a **detection** task, there will typically be an **object-level** annotation. To evaluate such a task, the first step is therefore to find the **matching pairs** $(T_i, P_j)$ of objects in the images. The matching criteria are therefore an important factor in the evaluation process. They typically involve a **matching threshold**, for instance based on the centroid distance or the surface overlap. Once the

[29] https://digestpath2019.grand-challenge.org/Dataset/ (Archive link 19/04/2022)

matching pairs of objects have been found, there are two separate aspects of the results that can be evaluated. First, a **partial confusion matrix** can be built:

$$\begin{pmatrix} N.A. & FP = |\{P_i \notin M\}| \\ FN = |\{T_i \notin M\}| & TP = |\{T_i \in M\}| \end{pmatrix}$$

With TP the "True Positives" (number of matching pairs), FN the "False Negatives" (number of un-matched ground truth objects), FP the "False Positives" (number of un-matched predicted objects), $M = \{Matches(T_k, P_l)\}$ and $|\cdot|$ is the cardinality of a set. It should be noted that, at the object-level, there are no countable "True Negatives", which will limit the available metrics for evaluating the detection score [223].

The other possible aspect of the evaluation is the **quality of the matches**. This can take different forms depending on what exactly the target output was: bounding box overlap, centroid distance, etc.

In a **classification task**, the annotation could be at the object level (instance classification), at the pixel level (semantic segmentation), at both levels (instance segmentation and classification), or at any of the "higher" levels in Figure 4.1: patch, WSI, or even patient. The latter cases correspond to "pure" classification tasks, whereas the others are combining classification with detection and/or segmentation. The classification part of the problem can typically be summed up with a **confusion matrix**, built from pairs of $(T_i, P_i)$ where $T_i \in [1, m]$ and $m$ is the number of classes in the problem, and $P_i = argmax_c(\pi_{ic})$ with $\pi_{ic}$ a vector of $m$ class probabilities so that $\sum_c \pi_{ic} = 1$. The confusion matrix will therefore be a $m \times m$ matrix with $CM_{jk} = |\{(T_i, P_i) \mid T_i = j, P_i = k\}|$.

The distinction between "classification" tasks and "detection" tasks can sometimes be difficult to make. In many digital pathology applications, there will be one "meaningful" class (for instance: nuclei, tumour, etc.), and a "no class" category that includes "all the rest". In those cases, there will be clear "positive" and "negative" categories, and the confusion matrix will include the TP, FP, FN (and, in this case, countable TNs) of the detection tasks. As we will see further in this chapter, this can be a source of confusion in the definition of the metrics, as some metrics, such as the F1-Score, take a different form if a single "positive" class is considered (as in a detection task) or if two or more classes are considered on equal grounds.

The combination of "detection" and "classification" is **instance classification**, which can also be seen as **multi-class detection**. In such problems, the results can be described in a confusion matrix that includes multiple classes *and* a background or no-instance category. The confusion matrix for an m-class problem would therefore look like:

$$\begin{pmatrix} \text{N. A.} & CM_{01} & \dots & CM_{0\text{m}} \\ CM_{10} & CM_{11} & \dots & CM_{1\text{m}} \\ \vdots & \vdots & \ddots & \vdots \\ CM_{\text{m}0} & \dots & \dots & CM_{\text{mm}} \end{pmatrix}$$

Where the first line corresponds to falsely positive detections (i.e. detections for any of the classes that correspond to no ground truth object) and the first column to falsely negative detections (i.e. ground truth objects that have no predictions). As in the previously described detection confusion matrix, the top-left corner is uncountable and corresponds to the true negative detections.

In **segmentation tasks**, the annotations will be at the pixel-level. These annotations can be simply binary (an "annotation mask"), or also contain class labels ("class map") and/or instance labels

("instance map"). We will therefore have $T = \{T_i\}$ and $P = \{P_i\}$, with $i$ the pixel position. In an **instance segmentation** case, there will once again need to be a **matching criterion** to find the matching subsets $T_k = \{T_i \mid label(T_i) = k\}$ and $P_l = \{P_i \mid label(P_i) = l\}$, so that the evaluation can be reduced to several binary "object-level" annotation masks. In **semantic segmentation**, there is no matching step necessary. Instead, a segmentation metric will often simply be computed per-class, then an average score can be computed. The most common evaluation metrics can be defined from pixel-level confusion matrices, or as distance metrics that compare the contours and/or the centroids of the binary masks.

When all three tasks are combined together, we get **instance segmentation and classification**. In such problems, each pixel can be associated both to a class and to an instance label. The evaluation of such problems can be quite difficult, as will be discussed through this chapter.

### 4.1.2 Detection metrics

As mentioned above, detection tasks typically require a **matching** step before the evaluation, often involving a **matching threshold** $\mu$. The matching step takes the sets of Target objects $\{T_i\}$ and Predicted objects $\{P_i\}$ and outputs three sets: the true positives, the un-matched false negatives, and the un-matched false positives.

These sets are related to a **confidence threshold** $\tau$ on the detection probability (which is the raw output of the detection algorithm), that determines if a candidate region is considered as a detected object or not: $P(\tau) = \{P_i; \pi_i \geq \tau\}$, with $\pi_i$ the detection probability for the object $i$ [223], with $\tau$ usually set to 0.5.

Higher confidence thresholds will yield a smaller set of $P_i$, which means that the following relationships are always verified:

$$\tau_1 < \tau_2 \Rightarrow TP_{\tau_1} \geq TP_{\tau_2}, FP_{\tau_1} \geq FP_{\tau_2} \; and \; FN_{\tau_1} \leq FN_{\tau_2}$$

Detection metrics will therefore be either "fixed-threshold" metrics, bound to a particular choice for the matching and confidence thresholds, or "varying-threshold", capturing the evolution of the metric as the thresholds are moved.

#### *4.1.2.1 Object Matching*

In most cases, object detection outputs include some sort of localisation, which gives an indication of where is the object in the image, and what is its size. In the most precise case, we have the full segmentation of the object, and are therefore in an "instance segmentation" problem. More often, the localisation will be given in the form of a bounding box with a centroid. In the extreme opposite case, there is no localisation at all, and simply an indication of whether an object is present in the image or not. There is then no "matching" step, and the problem is formulated as a binary classification problem, with the "no object" and "object" class being determined at the image patch level.

The matching step generally is "**overlap-based**" and/or "**distance-based**".

Given a pair of target and predicted objects $T_i$ and $P_j$, represented as a set of pixels belonging to this object (which may come either from bounding boxes or per-pixel instance masks), overlap-based methods will look at the **intersection** area ($|T_i \cap P_j|$) and the **union** area ($|T_i \cup P_j|$) of the two objects. These two measures can be combined in the **intersection over union** (IoU), also known as the Jaccard Index [227]:

$$IoU(T_i, P_j) = \frac{|T_i \cap P_j|}{|T_i \cup P_j|}$$

A common criterion is to apply a threshold on this IoU, so that a pair of objects is "a match" if $IoU(T_i, P_j) \geq \mu_{IoU}$, with $\mu_{IoU}$ often set at 0.5 or 0.75 [224]. If $\mu_{IoU} < 0.5$, it is possible for multiple target objects to be matched to the same predicted objects (and vice-versa), in which case a "maximum IoU" criterion is typically added to the rule (so that the match is the predicted object with the maximum IoU among those that satisfy the threshold condition). In that case, we will therefore have: $Matches(T, P) = \{(T_i, P_j)\}$ where all three following conditions are respected:

$$IoU(T_i, P_j) \geq \mu_{IoU},$$

$$IoU(T_i, P_j) > IoU(T_i, P_{k \neq j}) \; \forall P_k \in P$$

$$IoU(T_i, P_j) > IoU(T_{k \neq i}, P_j) \forall T_k \in T$$

The maximum IoU can also be used as a single criterion, with no threshold attached, in which case any overlap may be detected as a match (this is equivalent to setting $\mu_{IoU} = 0$).

Another strategy to determine a match relates to the **distance between the centroids** of the objects. Similarly to the IoU criterion, this will imply a "closest distance" rule, which may be associated with a "maximum distance threshold". The criteria will therefore be:

$$d(T_i, P_j) \leq \mu_{dist}$$

$$d(T_i, P_j) < d(T_i, P_{k \neq j}) \; \forall P_k \in P$$

$$d(T_i, P_j) < d(T_{k \neq i}, P_j) \; \forall T_k \in T$$

With $d(T_i, P_j)$ usually defined as the Euclidian distance between the centroids of $T_i$ and $P_j$.

The same rule can be adapted to other distance definitions, such as for instance the Hausdorff's distance (defined in the segmentation metrics below) between the contours of the objects.

#### *4.1.2.2 Fixed-threshold metrics*

Fixed-threshold metrics are computed for a specific value of the matching threshold (commonly, $\mu_{IoU} = 0.5$) and of the confidence threshold (also usually $\tau = 0.5$), which determine the sets TP, FP and FN.

The **precision** (PRE) of the detection algorithm is the proportion of "positive detections" that are correct, and is defined as:

$$PRE = \frac{TP}{TP + FP}$$

The **recall** (REC) is the proportion of "positive target" that have been correctly predicted, and is defined as:

$$REC = \frac{TP}{TP + FN}$$

The **F1-score** (F1) is the harmonic mean of the two:

$$F1 = 2 \times \frac{PRE \times REC}{PRE + REC} = 2 \times \frac{TP}{2 \times TP + FP + FN}$$

#### 4.1.2.3 *Varying-threshold metrics*

To better characterize the robustness of an algorithm's prediction, it is sometimes useful to look at how the performance evolves with different values of the confidence and/or matching thresholds.

Decreasing the confidence threshold means being less restrictive on what's considered a "prediction", and therefore to more true positives, more false positives and less false negatives. There is thus a monotonic relationship between the confidence threshold and the recall. The relationship between the precision and the confidence threshold, however, is less predictable.

This dynamic relationship between PRE, REC and the confidence threshold can be captured in a "precision-recall curve" (**PR curve**). An example of a PR curve is shown in Figure 4.3, constructed from synthetic data. As we can see, while the overall trend is for the precision to decrease as the recall increases, the relationship is not monotonic, and the precision has a characteristic "zig-zag" pattern.

To summarize the performance shown by the PR curve, the "area under the PR curve" (AUPRC) can be computed. An ideal detector would have $AUPRC = 1$. A very common approximation of the AUPRC, that is less sensitive to the small oscillations of the precision, is the "Average Precision" (AP). Adapting the formulation from Luque et al. [224], we can express the PR curve as a function $PRE = Pre(REC)$, and the AP is obtained by:

a) Interpolating the PR curve so that $Pre^*(REC) = \max_{k \geq REC} Pre(k)$, thus removing the zigzag pattern by replacing each point by the "highest value to its right" (see green line in Figure 4.3).
b) Sampling $Pre^*$ with $n$ equally spaced recall values, so that $AP = \frac{1}{n}\sum_{k=1}^{n} Pre^*(k)$.

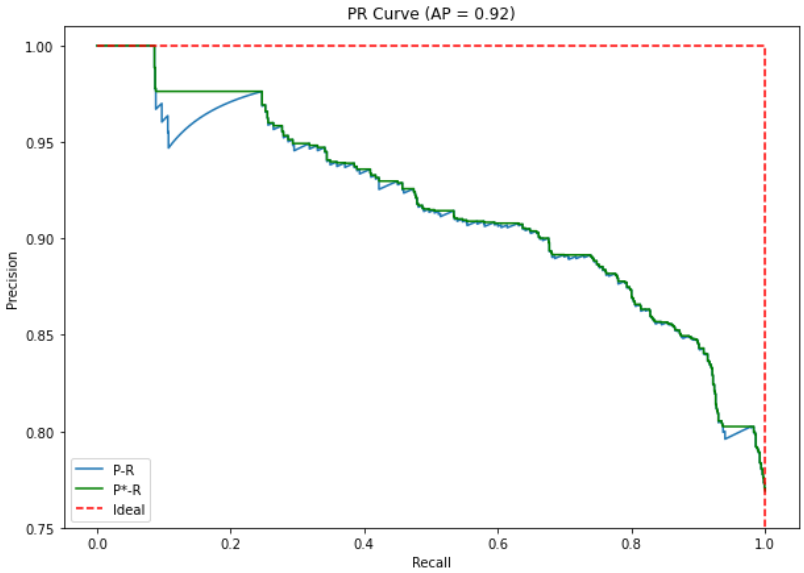



**Figure 4.3. Example of a PR curve (blue) created from synthetic data. The dotted red line corresponds to the "ideal" detector, the green line is the interpolated curve. The AP was computed based on a sampling of 100 equally spaced points.**

The AP is therefore a **varying-threshold** metric for the confidence threshold, but a **fixed-threshold** metric for the matching rule. To also take into account this matching rule variability, the AP can be averaged over different matching threshold values. As an example, in the NuCLS 2021 challenge [188], one of the metrics used to evaluate the detection score is the mean AP using an IoU threshold varying between 0.5 and 0.95, by step of 0.05. If we set $AP@k$ to mean the AP using the matches found with $\mu_{IoU} \geq k$, then we have:

$$AP@.5{:}.95 = \frac{1}{10}\sum_{k=0}^{9} AP@(0.5 + 0.05k)$$

### 4.1.3 Classification metrics

The results of a classification problem with $m$ categories are summarized in a $m \times m$ confusion matrix. We will use here the convention that the *rows* of that matrix correspond to the "ground truth" class, while the *columns* correspond to the "predicted" class, and $CM_{ij}$ corresponds to the number of examples with "true class" $i$ and "predicted class" $j$. It should be noted, however, that some of the classification metrics may be used to simply characterize the agreement between two sets of observations, without one being assumed to be the "truth". To be useful for that latter purpose, a metric needs to be **observer invariant**. Practically, this is the case if $Metric(CM) = Metric(CM^T)$ (or, from the sets of ground truth and predicted observations $T = \{T_i\}, P = \{P_i\}$ : $Metric(T, P) = Metric(P,\ T)$).

Another characteristic of classification metrics is that they can be **class-specific** or **global**. A class-specific metric $Metric_c$ will evaluate class $c$ as a "positive" class against all others, while global metrics aggregate the class-specific values.

Metrics can also assume an **order** to the classes or, on the contrary, that they are purely **categorical**. In the former case, switching the classes in the confusion matrix would lead to a different result, while categorical metrics are class-switch invariant.

Finally, another aspect (that we more thoroughly examine in sections 4.3 and 4.4) is the **class imbalance bias** that many metrics exhibit. This means that changing the class balance of the dataset (e.g. by changing the sampling method) alters the results.

#### *4.1.3.1 Class-specific metrics*

Several per-class metrics can be defined. These use the notions of the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). Using our general notation for the confusion matrix, we can have for class $c$:

$$TP_c = CM_{cc}$$

$$FP_c = \sum_{k \neq c} CM_{kc}$$

$$FN_c = \sum_{k \neq c} CM_{ck}$$

$$TN_c = n - TP_c - FP_c - FN_c$$

A visual example on a 5 classes problem is shown in Table 4.1.

**Table 4.1. Confusion matrix in a 5 classes problem. The colours show the elements of the matrix that contribute to (yellow) $TP_\beta$, (orange) $FP_\beta$, (blue) $FN_\beta$ and (white) $TN_\beta$.**

| T \ P | $\alpha$ | $\beta$ | $\gamma$ | $\delta$ | $\epsilon$ |
|---|---|---|---|---|---|
| $\alpha$ | | | | | |
| $\beta$ | | | | | |
| $\gamma$ | | | | | |
| $\delta$ | | | | | |
| $\epsilon$ | | | | | |

The **sensitivity** (SEN), also called **recall** (REC) or **True Positive Rate** (TPR) measures the proportion of "True Positives" among all the samples that are "positive" in the Target (i.e. that truly belong to the class *c*):

$$SEN_c = \frac{TP_c}{TP_c + FN_c}$$

The **specificity** (SPE), or **True Negative Rate** (TNR) is conversely the proportion of TN in the negative Target samples:

$$SPE_c = \frac{TN_c}{TN_c + FP_c}$$

The **precision** (PRE) or **Positive Predictive Value** (PPV) is the proportion of TP in the positive Predicted samples:

$$PRE_c = \frac{TP_c}{TP_c + FP_c}$$

The **Negative Predictive Value** (NPV) is similarly the proportion of TN in the negative Predicted samples:

$$NPV_c = \frac{TN_c}{TN_c + FN_c}$$

The **False Positive Rate** (FPR) is the proportion of FP in the negative Target samples:

$$FPR_c = \frac{FP_c}{TN_c + FP_c} = 1 - SPE_c$$

The **F1-Score** is better defined as a detection metric (see section 4.1.2), but it is often used in classification problems as well.

The **per-class F1-Score** ($F1_c$) is defined as the harmonic mean of the precision and sensitivity:

$$F1_c = 2 \times \frac{PRE_c \times SEN_c}{PRE_c + SEN_C}$$

The **per-class Geometric Mean** ($GM_c$) of the SEN and SPE is also sometimes used [225]:

$$GM_c = \sqrt{SEN_c \times SPE_c}$$

#### 4.1.3.2 Global metrics

To get a global classification score that aggregates the results from all classes, several metrics are possible.

The simplest performance measure from the confusion matrix is the **accuracy** (ACC), which is given by the sum of the elements on the diagonal (the "correct" samples) divided by the total number of samples $n$:

$$ACC = \frac{\sum_i CM_{ii}}{n}$$

It is also possible to extend the F1-Score so that it becomes a global metric. The **macro-averaged F1-Score** has been defined in two different ways in the literature [228].

First, as a simple average of the per-class F1-scores. We call this version the **simple-averaged F1-Score** ($sF1$):

$$sF1 = \frac{1}{m}\sum_c F1_c$$

The other definition first computes the average of the per-class PRE and SEN, before computing the harmonic mean. We will call it the **harmonic-averaged F1-Score** ($hF1$):

$$hF1 = 2 \times \frac{\mathrm{M}PRE \times \mathrm{M}SEN}{\mathrm{M}PRE + \mathrm{M}SEN}$$

With:

$$\mathrm{M}PRE = \frac{\sum_k PRE_k}{m}$$

$$\mathrm{M}SEN = \frac{\sum_k SEN_k}{m}$$

Meanwhile, the **micro-averaged F1-Score** ($\mu F1$) will first aggregate the TP, FP and FN before computing the **micro-precision** and **micro-recall**, and finally the F1-Score. It has been used in challenges such as Gleason 2019, but it should be noted that it is equivalent to the accuracy:

$$\mu PRE = \frac{\sum_k TP_k}{\sum_k (TP_k + FP_k)} = \frac{\sum_k TP_k}{n} = ACC$$

$$\mu SPE = \frac{\sum_k TP_k}{\sum_k (TP_k + FN_k)} = \frac{\sum_k TP_k}{n} = ACC$$

$$\mu F1 = 2 \times \frac{\mu PRE \times \mu SPE}{\mu PRE + \mu SPE} = ACC$$

Like the F1-Score, the $GM_c$ can also be extended as a global metric. It will then be the **Geometric Mean** (GM) of the per-class $SEN_c$:

$$GM = \left(\prod_c SEN_c\right)^{\frac{1}{m}}$$

The GM has zero bias due to class imbalance [225]. It is, however, **not observer invariant**.

The **Matthews Correlation Coefficient** (MCC), often called $R_K$ in the multiclass case (and sometimes phi coefficient), uses all the terms of the CM and is defined as:

$$MCC = \frac{n \times \sum_k CM_{kk} - \sum_k (\sum_l CM_{lk} \times \sum_l CM_{kl})}{\sqrt{n^2 - \sum_k (\sum_l CM_{kl})^2} \sqrt{n^2 - \sum_k (\sum_l CM_{lk})^2}}$$

It is bound between -1 and 1, where 0 means that the target and prediction sets are completely uncorrelated.

**Cohen's kappa** is a measure of the agreement between two sets of observations and can be used as a classification metrics to rate the agreement between the target and the predictions. It measures the difference between the observed agreement $p_o$ and the agreement expected by chance $p_e$, and is defined as [229]:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

So that a "perfect agreement" ($p_o = 1$) leads to $\kappa = 1$, a perfect disagreement ($p_o = 0$) to a negative $\kappa$ (the exact minimum value depends on the distribution of the data). From the confusion matrix, $p_o$ and $p_e$ are defined as:

$$p_o = ACC = \frac{\sum_i CM_{ii}}{n}$$

$$p_e = \frac{\sum_i (\sum_k CM_{ik} \sum_k CM_{ki})}{n^2}$$

Cohen's kappa is also often used for **ordered** categories, where errors between classes which are "close" together are less penalised.

Let $W$ be a weight matrix ($W$ should verify $w_{ij} = w_{ji}$, and $0 \leq w_{ij} \leq 1$).

We can define the matrix of expected observations from random chance with:

$$e_{ij} = \frac{\sum_k CM_{ik} \sum_k CM_{kj}}{n}$$

From that, we define **Cohen's kappa** generally with:

$$\kappa = 1 - \frac{\sum_i \sum_j w_{ij} CM_{ij}}{\sum_i \sum_j w_{ij} e_{ij}}$$

The three most common formulations of the weights are:

**Unweighted kappa $\kappa_U$**: $w_{ij} = 1 - \delta_{ij}$ (using the Kronecker delta), which resolves to the same value as the original definition of the $\kappa$ shown above.

**Linear weighted kappa $\kappa_L$**: $w_{ij} = |i - j|$

**Quadratic weighted kappa $\kappa_Q$**: $w_{ij} = (i - j)^2$

Cohen's kappa is bound between -1 and 1, with 0 corresponding to as much agreement as expected by random chance.

#### 4.1.3.3 Varying-threshold metrics

All the classification metrics described thus far are "no threshold" metrics, where the predicted class of a sample is simply taken as the class which has the maximum predicted probability. It can however also be interesting to take into account the "confidence" of a model with regards to its predictions. This information can be captured in a "**Receiver Operating Characterstic**", or **ROC Curve**. Where the detection PR-curve plotted the PRE and REC for varying confidence thresholds, the ROC curve plots the SEN (=REC) and the FPR (=1-SPE).

As the SEN and SPE are "per-class" metrics, so is the ROC a "per-class" visualisation, where the "varying threshold" is used in a "one class vs all" manner. For a class $c$, a threshold $\tau$, and the set of predicted class probabilities vectors $\{P_i\}$ where $\pi_{ic}$ is the probability that sample $i$ has the class $c$, we can define:

$$TP_{c,\tau} = \left|\{P_i\ ;\ \pi_{ic} \geq \tau\ \&\ T_{i,c} = 1\}\right|$$

$$FP_{c,\tau} = |\{P_i\ \ ;\pi_{ic} \geq \tau\ \&T_{i,c} = 0\}|$$

$$TN_{c,\tau} = |\{P_i\ ;\pi_{ic} < \tau\ \&T_{i,c} = 0\}|$$

$$FN_{c,\tau} = \left|\{P_i\ ;\pi_{ic} < \tau\ \&\ T_{i,c} = 1\}\right|$$

From which $SPE_{c,\tau}$ and $SEN_{c,\tau}$ can be computed.

To summarize the information contained in the ROC curve, the Area Under the ROC ($AUROC_c$) can be computed for a class, so that an "ideal classifier" for that class would have an $AUROC_c$ of 1. As for the F1-Score, we can also define micro-averaged and macro-averaged summaries of the $AUROC$.

The macro-average $MAUROC$ is defined as:

$$MAUROC = \frac{1}{c}\sum_{c} AUROC_c$$

While the micro-averaged is defined by first summing the TP, FP, TN and FN:

$$\mu TP_\tau = \frac{1}{c}\sum_c TP_{c,\tau}, \text{ etc.}$$

Then computing the resulting $\mu SPE_\tau$, $\mu SEN_\tau$, and finally the $\mu AUROC$ from those micro-averaged values. An example on synthetic data is presented in Figure 4.4.

A summary of the main characteristics of the metrics discussed here can be found in Table 4.2.

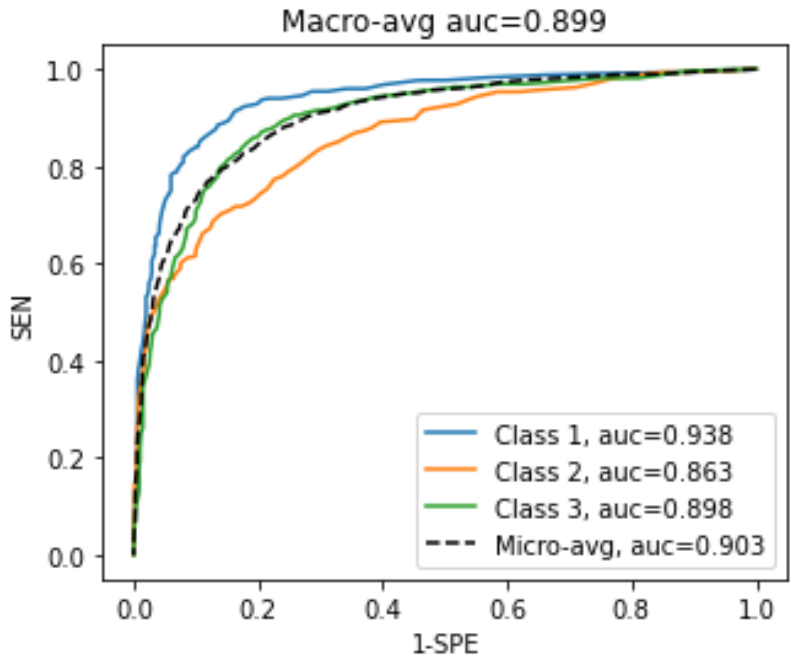



**Figure 4.4. Example of ROC curves for a 3-class problem on synthetic data.**

**Table 4.2. Range of values and key properties of the most common classification metrics.**

| Metric | Range of values | Class-specific or global | Observer invariant | Class imbalance bias | Ordered |
|---|---|---|---|---|---|
| $SEN_c$ | [0, 1] | Class-specific | No | No | No |
| $SPE_c$ | [0, 1] | Class-specific | No | No | No |
| $PRE_c$ | [0, 1] | Class-specific | No | High | No |
| $NPV_c$ | [0, 1] | Class-specific | No | High | No |
| $F1_c$ | [0, 1] | Class-specific | Yes | High | No |
| $GM_c$ | [0, 1] | Class-specific | No | No | No |
| $ACC$ | [0, 1] | Global | Yes | Medium | No |
| h$F1$ | [0, 1] | Global | Yes | Low | No |
| $sF1$ | [0, 1] | Global | Yes | Medium | No |
| $MCC$ | [-1, 1] | Global | Yes | Medium | No |
| $\kappa_U$ | [-1, 1] | Global | Yes | High | No |
| $\kappa_L, \kappa_Q$ | [-1, 1] | Global | Yes | High | Yes |
| $GM$ | [0, 1] | Global | No | No | No |
| $AUROC_c$ | [0, 1] | Class-specific | No | No | No |
| $\mu AUROC$ | [0, 1] | Global | No | No | No |
| $MAUROC$ | [0, 1] | Global | No | No | No |

### 4.1.4 Segmentation metrics

We consider in this section pure "segmentation" tasks that separate a foreground region from a background region. Instance and semantic segmentation will be considered in the "combined metrics" section. Segmentation metrics therefore compare two binary masks in an image. Let $T$ be

the set of pixels that belong to the target ("ground truth") mask, and $P$ the set of pixels that belong to the predicted mask. There are two main categories of segmentation metrics: those that measure the **overlap** between the two sets, and those that measure a **distance** between the contours of the $T$ and $P$ masks, labelled $T_o$ and $P_o$ respectively.

In the following definitions, $|\cdot|$ is the cardinality of the set and $\sim$ indicates all the elements that are not in the set.

#### *4.1.4.1 Overlap metrics*

The overlap metrics can also be defined using a "confusion matrix" based on the binary segmentation masks $\boldsymbol{T}$ and $\boldsymbol{P}$ (we use here boldface to denote the mask matrices instead of the sets), where $\boldsymbol{T_i} = 1$ for all $\boldsymbol{T_i} \in T$, 0 otherwise. The TP, FP, FN and TN are then defined as:

$$TP = \sum_i \boldsymbol{T_i} * \boldsymbol{P_i} = |T \cap P|$$

$$FP = \sum_i (1 - \boldsymbol{T_i}) * \boldsymbol{P_i} = |\sim T \cap P|$$

$$FN = \sum_i \boldsymbol{T_i} * (1 - \boldsymbol{P_i}) = |T \cap \sim P|$$

$$TN = \sum_i (1 - \boldsymbol{T_i}) * (1 - \boldsymbol{P_i}) = |\sim T \cap \sim P|$$

The two most commonly used overlap metrics are the **Dice Similarity Coefficient** (DSC) and the **Intersection over Union** (IoU), while the most common distance-based metric is **Hausdorff's Distance** (HD). The IoU, DSC and HD are illustrated in Figure 4.5.

The **IoU** is defined as:

$$IoU = \frac{|T \cap P|}{|T \cup P|}$$

It is bounded between 0 (no overlap) and 1 (perfect overlap).

Using the confusion matrix notation, it can also be defined as:

$$IoU = \frac{TP}{TP + FP + FN}$$

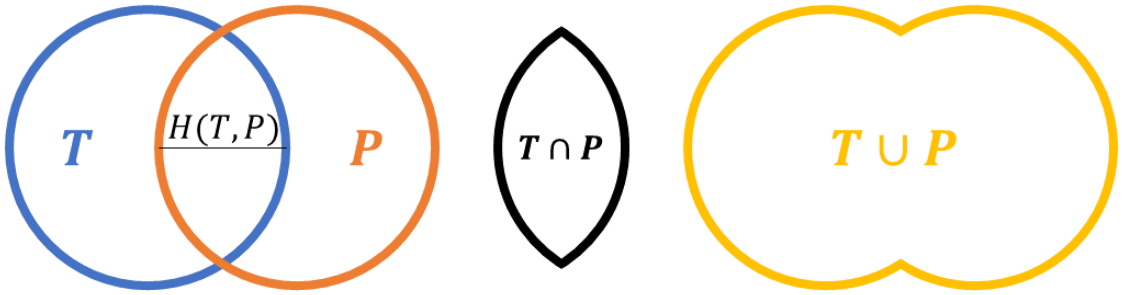
The **DSC** is defined as:

$$DSC = \frac{2|T \cap P|}{|T| + |P|}$$

It is also bounded between 0 and 1, and heavily correlated to the IoU, as the two are linked by the relationship:

$$DSC = \frac{2 \times IoU}{1 + IoU}$$

That relationship also means that $DSC \geq IoU$ is always verified.




$$DSC = \frac{2|T \cap P|}{|T| + |P|} \qquad IoU = \frac{|T \cap P|}{|T \cup P|} \qquad H(T,P) = \max(h(T,P), h(P,T))$$

**Figure 4.5. Illustration of the three "basic" segmentation metrics. Adapted from our review** [8]**.**

Like the IoU, we can define it in terms of the confusion matrix values:

$$DSC = \frac{2 \times TP}{2 \times TP + FP + FN}$$

Which shows that it simply corresponds to a "per-pixel" definition of the F1-score.

#### *4.1.4.2 Distance metrics*

For **Hausdorff's distance** (HD), only the contours $T_o$ and $P_o$ are considered. First, the Euclidean distances from all points in $T_o$ to the closest point in $P_o$ are computed, and the maximum value is taken:

$$h_{TP} = \max_{t \in T_o} \min_{p \in P_o} ||t - p||$$

Then, the same thing is done starting from all the points in $P_c$:

$$h_{PT} = \max_{p \in P_o} \min_{t \in T_o} ||p - t||$$

Hausdorff's distance is then defined as the maximum of those two values:

$$HD = \max(h_{TP}, h_{PT})$$

This process is illustrated in Figure 4.6.

As the HD is extremely sensitive to outliers, it is also sometimes preferred to slightly relax the "maximum", and to replace it with a percentile $P_\lambda$:

$$h_{\lambda,TP} = \mathrm{P}_\lambda \min_{t \in T_o} \min_{p \in P_o} ||t - p||, HD_\lambda = \max(h_{\lambda,TP}, h_{\lambda,PT})$$

With $HD_{95}$ being the most frequent choice.

While a HD of 0 corresponds to a perfect segmentation, its maximum value is unbounded. This makes it a particularly tricky metric to use in aggregates, as it is difficult to assign a value, for instance, to a "missing prediction" (i.e. an image where $P = \emptyset$).

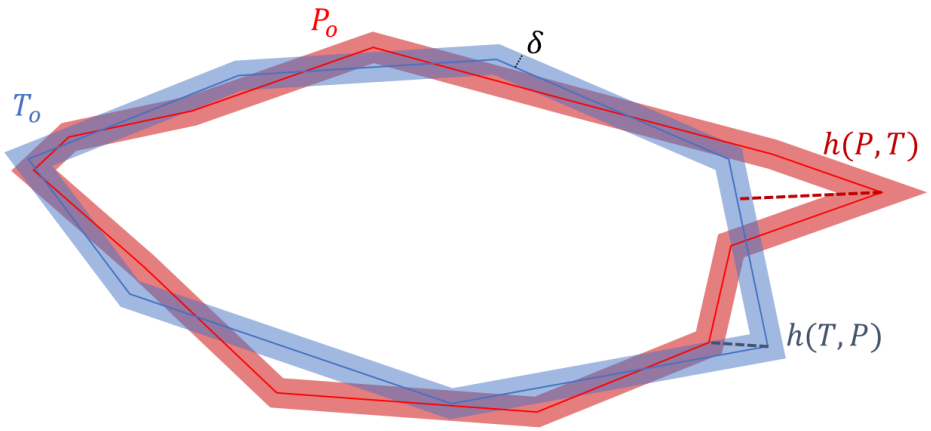



**Figure 4.6. Illustration of how the HD is computed on a ground truth contour $T_o$ and a predicted contour $P_o$. $h(T,P)$ is the distance from the point in $T_o$ that is the furthest from any point in $P_o$, and conversely for $h(P,G)$. The HD is the maximum of these two distances, which in this case is $h(P,T)$. A boundary region using a tolerance $\delta$, which could be used in the NSD or in the uncertainty-aware HD, is shown in light colours for both contours.**

While less commonly used, some other distance-based metrics exist. Reinke et al.'s review [226] notably mention the **Average Symmetric Surface Distance** (ASSD) and the **Normalized Surface Distance** (NSD).

The **ASSD** is defined as:

$$ASSD = \frac{\sum_{t \in T_o} \min_{p \in P_o} ||t - p|| + \sum_{p \in P_o} \min_{t \in T_o} ||p - t||}{|T_o| + |P_o|}$$

For each point of the contour, the distance to the other contour is computed. The final value is the average distance computed from both contours. Compared to the HD, it will tend to penalize more a segmentation that is consistently wrong by a small amount than a segmentation that is almost perfect but with a few large errors.

The **NSD**, meanwhile, is a metric that takes into account the **uncertainty** of the annotations by first defining "boundary regions" $B_T$ and $B_P$, which are the set of all pixels within a certain tolerance distance $\delta$ of the contours $T_o$ and $P_o$:

$$B_{T,\delta} = \left\{ b \,;\, \min_{t \in T_o} |b - t| < \delta \right\}, B_{P,\delta} = \left\{ b \,;\, \min_{p \in P_o} |b - p| < \delta \right\}$$

The NSD is then defined as:

$$NSD_\delta = \frac{\left|P_o \cap B_{T,\delta}\right| + \left|T_o \cap B_{P,\delta}\right|}{|P_o| + |T_o|}$$

This same idea of a "tolerance" could easily be applied to the HD as well by redefining the $h_{TP}$ and $h_{PT}$ as:

$$h_{TP,\delta} = \max_{t \in T_o} \min_{b \in B_{P,\delta}} ||t - b||, h_{PT,\delta} = \max_{p \in P_o} \min_{b \in B_{T,\delta}} ||p - b||$$

So that the "uncertainty-aware" $HD_\delta$ is given by:

$$HD_\delta = \max\left(h_{TP,\delta}, h_{PT,\delta}\right)$$

#### *4.1.4.3 Segmentation as binary classification*

The IoU and DSC metrics, in their "confusion matrix" definitions, are essentially "pixel detection" metrics, which do not take into account the "true negatives" and consider that there is a "positive" class, the foreground, and a "negative" class, the background. But once we have established the confusion matrix, it is also possible to treat the problem as a "binary classification" problem, *including* the true negatives. The classification metrics defined in 4.1.3 can therefore also be used. The ACC or the MCC, for instance, can easily be computed.

The main difference of this approach is that it does not consider that the "foreground class" is inherently more important than the "background class", and that the metric should reflect that correctly identifying that a pixel is in the background is just as important as identifying one in the foreground.

A key benefit is that it becomes trivial to extend the metrics to the semantic segmentation case, as adding new classes just means adding new rows and columns to the confusion matrix, but the formula for computing the metrics remains unchanged.

The main drawback of this approach is that, when the segmentation metric is computed on patches extracted from a larger image (for instance, comparing just the masks of a detected object and its matching ground truth object), the result of the metrics becomes highly dependent on the size of the patches, and the amount of background included in it.

Binary segmentation problems are indeed typically not really "two classes" problems, but rather "one class" class problems where we detect one class and put all others in the same bin. The "per-pixel binary classification" approach would therefore only make sense if the problem can actually be defined as a two classes problem.

### 4.1.5 Metrics for combined tasks

The most common approach for the evaluation of combined tasks (instance segmentation, semantic segmentation, instance classification, instance segmentation and classification) is to combine "basic" metrics to create a new one which attempts to summarizes all aspects of the task into a single score.

The Gleason 2019 challenge, for instance, combines a classification $\kappa$ with a micro-averaged and a macro-averaged segmentation F1-Score into a score $S = \kappa + \frac{1}{2}(\mu F_1 + MF_1)$. The SNI challenges, meanwhile, combine a simple "binary" DSC (on the whole image and not on separate instances) with a sort of micro-averaged DSC where the numerator and denominator of the DSC are separately aggregated over the set of matching instances. A similar idea is used in the MoNuSeg challenge with the "Aggregated Jaccard Index" (AJI), where the Intersection and the Union are aggregated over the matches.

For the most complex task of instance segmentation and classification, the "Panoptic Quality" (PQ), originally proposed by Kirillov et al. [230] for natural scenes, has recently been introduced to digital pathology by Graham et al. [103].

The PQ considers each class separately. For each class $c$, $T_c = \{t_i\}$ is the set of ground truth instances in the class. Given a set of corresponding class predictions $P_c = \{p_i\}$, the $PQ_c$ is computed in two steps.

First, the **matches** between ground truth instances and predicted instances are found (using the $\mu_{IoU} = 0.5$ matching threshold). Using this strict matching rule, each segmented instance in $T_c$ and $P_c$ can be counted as TP, FP or FN.

Then, the ***PQ* of the class *c* in the image *i* is computed as**:

$$PQ_{c,i} = \frac{\sum_{(p_i,t_j)\in TP} IoU(p_i, t_j)}{TP + \frac{1}{2}FP + \frac{1}{2}FN}$$

Which can be decomposed into:

$$RQ_{c,i} = \frac{TP}{TP + \frac{1}{2}FP + \frac{1}{2}FN}$$

$$SQ_{c,i} = \frac{\sum_{(p_i,t_j)\in Matches} IoU(p_i, t_j)}{TP}$$

$$PQ_{c,i} = SQ_{c,i} \times RQ_{c,i}$$

With $RQ_c$ the "Recognition Quality", corresponding to the per-object F1-score of the class $c$, and $SQ_c$ the "Segmentation Quality", corresponding to the average IoU of the matching pairs of ground truth and predicted instances. The $RQ$ is also often referred to as the Detection Quality, and therefore noted as $DQ$ [24], [103].

### 4.1.6 Aggregation methods

A critical step to arrive at a final "score" for an algorithm on set of test samples is the aggregation of the per-sample metric(s). Let us therefore go back to the hierarchical representation of digital pathology datasets that we used at the beginning of the chapter (see Figure 4.1).

The aggregation process can happen in different dimensions: across that hierarchical representation, across multiple classes, and across multiple metrics.

#### *4.1.6.1 Hierarchical aggregation*

The aggregation across the hierarchical aggregation has two aspects: *where* are the metrics computed, and *how* are they averaged?

Object-level metrics would typically be those used in instance segmentation (per-object IoU, HD…). In a "bottom-up" approach, all the object-level measures could be averaged over a patch, then the patch-level measures over a WSI, the WSI-level over a patient, and finally the patient-level measures over the whole dataset. The problem of this approach is that, if the patches are very dissimilar in their object distribution, it may lead to cases where a single error on a patch with few objects is penalized a lot more harshly than an error on a larger patch with a larger population of objects. A more robust approach will therefore generally be to rather aggregate the per-object metrics over a WSI or a patient. It is also possible to directly average the object-level metrics on the entire dataset, skipping intermediate levels entirely. In the digital pathology context, having a patient-level distribution of a metric is however often desirable for a clinically relevant analysis of the results of an algorithm, as the goal will generally be to be able to validate

that the performance of an algorithm is good across a wide range of patients. It also makes it possible to use powerful statistical tests to compare different algorithms together, as patients can then be used as paired samples.

Patch-level metrics are very commonly used in all sorts of tasks. It could be a region segmentation metric like the IoU or DSC, an object detection metric like the F1-Score, or a classification metric like the ACC or the MCC. It is once again possible to directly average those results on the entire dataset, or to go through the patient-level first.

Another possibility for many of those metrics is to not compute the metric at the patch level, but to aggregate its base components over a patient or WSI. For instance, a confusion matrix can be built over multiple patches of a WSI, before computing the relevant classification or detection metrics on this aggregated confusion matrix. This reduces the risk of introducing biases from the patch selection and focuses the results on the object of interest of pathology studies: the patient. We therefore find here again the same choices of micro- and macro-averaging that were described in the classification metrics.

#### *4.1.6.2 Multiclass aggregation*

When multiple classes are present, detection and segmentation metrics are generally first computed independently for each class before being aggregated together. This aggregation can also happen at any point in the hierarchy of the dataset. For the PQ metric, for instance, some aggregate the $PQ_{c,i}$ at the image patch level as $PQ_i = \frac{1}{m}\sum_c PQ_{c,i}$, then find the global $PQ = \frac{1}{n}\sum_i PQ_i$ [24], [103], [214], while others averages each $PQ_c$ separately over all patches as $PQ_c = \frac{1}{n}\sum_i PQ_{c,i}$, then define the global $PQ = \frac{1}{m}\sum_c PQ_c$ [231] (with $n$ the number of images and $m$ the number of classes). As always, the micro- or the macro-averaging approaches can also be considered. In a micro-averaging approach, the metric must first be decomposed into its base components, which are aggregated separately before computing the metric. For most classification or detection metrics, the base components would be the elements of the confusion matrix. For combined metrics such as the PQ or some custom score, then a choice must be made as to what constitutes a "base component". Practically, almost all publications and challenges use a macro-averaging approach.

#### *4.1.6.3 Multi-metrics aggregation*

Single metrics are not capable of fully representing the capabilities of a given model or algorithm. This is true even for simple tasks: as we have seen, different metrics have distinct behaviours and biases, and may therefore miss some important insights about a particular algorithm. A clear example would be the IoU and the HD for segmentation metrics, which give very different information about the same task. As noted in section 4.1.5, combined tasks are often evaluated by creating complex metrics that combine basic detection, classification and segmentation metrics in some ways. Another approach, however, is to compute and report those simple metrics independently, to provide more detailed information about the predictive performance. Computing multiple metrics, however, means that algorithms cannot necessarily be ranked based on a single score, thus making the overall ranking (and therefore the choice of the "best method") more difficult to obtain.

A possible solution is to compute the ranks separately for the different metrics, then to combine them with, for instance, the "sum of ranks". As we will discuss below in section 4.4.7, this approach comes with its own pitfalls and challenges.

## 4.2 Metrics in digital pathology challenges

We examine here the evaluation metrics used in the challenges previously presented in Chapter 3, to get a sense of the common choices made by challenge organizers when determining the evaluation process. We look here mainly at the "basic" metrics (i.e. pure detection, classification or segmentation), and we will note when those metrics are included as part of a more complex evaluation score.

### 4.2.1 Detection tasks

A summary of the evaluation metrics used in digital pathology detection challenges can be found in Table 4.3. Almost all challenges used the F1-Score as the primary metric for ranking the participants' submissions. Some challenges reported the PRE and REC separately. The only "challenge" to use varying-threshold metrics is NuCLS 2021. This dataset, however, is presented more as a benchmark for future usage by researchers than as a challenge, despite being listed on the grand-challenge.org website. In the baseline results reported by Amgad et al. [188], the AP using a fixed matching threshold of 0.5 on the IoU, and the mAP using varying matching thresholds from 0.5 to 0.95 are used, so that both the confidence threshold and the IoU threshold are varying in the metric. DigestPath ranks three different metrics separately, then use the average rank as the overall rank for each competing team.

**Table 4.3. Summary of the metric(s) used in digital pathology detection challenges. Bolded values are used for the final ranking.**

| Challenge | Target | Metric(s) |
|---|---|---|
| PR in HIMA 2010 | Centroblasts in follicular lymphoma. | REC, SPE[30] |
| MITOS 2012 | Mitosis in breast cancer. | **F1**, PRE, REC |
| AMIDA 2013 | Mitosis in breast cancer. | **F1**, PRE, REC |
| MITOS-ATYPIA 2014 | Mitosis in breast cancer. | **F1**, PRE, REC |
| GlaS 2015 | Prostate glands | **F1** |
| TUPAC 2016 | Mitosis in breast cancer. | **F1** |
| LYON 2019 | Lymphocytes in breast, colon and prostate. | **F1** |
| DigestPath 2019 | Signet ring cell carcinoma. | PRE, **REC**, **FPs per normal region**, **FROC**[31]. |
| MoNuSAC 2020 | Nuclei | **F1** (as part of the PQ) |
| PAIP 2021 | Perineural invasion in multiple organs. | **F1** |
| NuCLS 2021 | Nuclei in different organs. | AP@.5[32], mAP@.5:.95 |
| MIDOG 2021 | Mitosis in breast cancer. | **F1** |
| Conic 2022 | Nuclei | **F1** (as part of the PQ) |

[30] See classification metrics. It is unclear how the "true negatives" are counted.
[31] Similar to AP, but with average REC computed for different numbers of FPs.
[32] AP computed at a minimum IoU threshold of 0.5 for a match.

### 4.2.2 Classification challenges

A summary of the metrics used in digital pathology classification challenges is presented in Table 4.4. Compared to the detection tasks, there is a lot more diversity in the choices made by challenge organisers. All of the multi-class tasks in these challenges (except NuCLS 2021) have some form of ordering present in their categories. For the binary classification tasks, many challenges are assessed with a positive-class only metric, as shown with CAMELYON 2016 and PatchCamelyon 2019 (AUROC of the metastasis class), DigestPath 2019 (AUROC of the malignant class), HeroHE 2019 (F1-Score, AUROC, SEN and PRE of the HER2-positive class) and PAIP 2020 (F1-Score of the MSI-High class). In the multi-class case, the TUPAC 2016 and PANDA 2020 evaluations use the weighted quadratic kappa, while in MITOS-ATYPIA 2014 a "penalty" system is used for errors of more than one class. In several challenges, however, this ordering is not taken into account at all. This is the case with the Brain Tumour DP 2014, BIOIMAGING 2015 and BACH 2018 challenges, where the simple accuracy is used, and for the C-NMC 2019 challenge, where a weighted macro-averaged $sF1$ is preferred.

**Table 4.4. Summary of the metric(s) used in digital pathology classification challenges. Bolded values are used for the final ranking.**

| Challenge | Classes | Metric(s) |
|---|---|---|
| Brain Tumour DP 2014 | Low Grade Glioma / Gliobastoma | **ACC** |
| MITOS-ATYPIA-14 | Nuclear atypia score (1-3) | **ACC** with penalty[33] |
| BIOIMAGING15 | Normal, benign, *in situ*, invasive | **ACC**[34] |
| TUPAC16 | Proliferation score (1-3) | $\boldsymbol{\kappa_Q}$ |
| CAMELYON16 | Metastasis / No metastasis | $\boldsymbol{AUROC_{metastasis}}$ |
| BACH18 | Normal, benign, *in situ*, invasive | **ACC** |
| C-NMC19 | Normal, malignant | **Weighted sF1**[35] |
| Gleason 2019 | Gleason grades | $\boldsymbol{\kappa}$ (included in a custom score) |
| PatchCamelyon19 | Metastasis / No metastasis | $\boldsymbol{AUROC_{metastasis}}$ |
| DigestPath19 | Benign / Malignant | $\boldsymbol{AUROC_{malignant}}$ |
| HeroHE20 | HER2 positive / negative | $\boldsymbol{F1}_+, AUROC_+, SEN_+, PRE_+$ |
| PANDA20 | Gleason group (1-5) | $\boldsymbol{\kappa_Q}$ |
| PAIP20 | MSI-High / MSI-Low | $\boldsymbol{F1_{High}}$ |
| NuCLS21 | Different types of nuclei | $ACC, MCC, \mu AUROC, MAUROC$ |

These classification tasks make the boundaries between detection, classification and regression tasks sometimes difficult to determine. In HeroHE 2020, for instance, the ranking of the algorithm is based on the "F1-Score of the positive class", which is a typical detection metric, but the AUROC of the positive class is also computed, which takes the "true negatives" into account and is typically a classification metric. The tasks that are explicitly about predicting a "score" all attempt to

[33] Points are given for a $(T_i, P_i)$ sample as $|P_i - T_i| + 1$, so that correct predictions give one point, predictions that are incorrect by one give zero point, and predictions that are incorrect by two give minus one point, with the ranking based on the sum of points. This is therefore equivalent to an "accuracy" with penalty points for larger errors.

[34] With a slight modification which does not impact the ranking: $ACC^* = \frac{N_{correct}-5}{N-5}$

[35] $F1_c$ are weighted based on the class distribution so that minority classes have less impact on the final score.

incorporate the notion of "distance" to the ground truth into their metric, which brings them closer to a regression task. The use of the quadratic kappa is also clearly related to its popularity in pathology and medical sciences in general, as it makes the results more relatable for medical experts. The danger of that particular metric, however, is that it may give a false sense of "interpretability", as the number of classes and the target class distribution have a large effect on the perceived performance of an algorithm using that metric.

### 4.2.3 Segmentation challenges

Almost all segmentation challenge use overlap-based metrics (either the IoU or the DSC) as their main metric for ranking participating algorithms, as shown in Table 4.5. The only challenge to really use a distance-based metric as part of the ranking was the GlaS 2015 challenge, which ranked the per-object average HD and the per-object average DSC (as well as the detection F1 score) separately. PAIP 2021 used the HD as a matching criterion but did not use it for the ranking.

The BACH 2018 challenge is an outlier in this, as they use a custom score that loosely corresponds to a "per-pixel" accuracy, with ordered classes so that "larger" errors are penalized more.

**Table 4.5. Summary of the metric(s) used in digital pathology segmentation challenges.**

| Challenge | Target | Metric(s) |
|---|---|---|
| PR in HIMA 10 | Lymphocytes | DSC, IoU, HD, MAD |
| Brain Tumour DP 14 | Necrosis region | **DSC** |
| GlaS 2015 | Prostate glands | **DSC, HD** |
| SNI 15-18 | Nuclei | **DSC**[36] |
| MoNuSeg 18 | Nuclei | **IoU**[37] |
| BACH 18 | Benign / in situ / invasive cancer regions | **Custom score** |
| Gleason 19 | Gleason patterns | **DSC**[38] (included in a custom score) |
| ACDC@LungHP 19 | Lung carcinoma | **DSC** |
| PAIP 19 | Tumour region | **IoU** |
| DigestPath 19 | Malignant glands | **DSC** |
| BCSS 19 | Tumour / stroma / inflammatory / necrosis / other tissue segmentation. | DSC |
| MoNuSAC 20 | Nuclei (epithelial / lymphocyte / neutrophil / macrophage) | **IoU** (as part of the PQ) |
| PAIP 20 | Tumour region | **IoU** |
| SegPC 21 | Multiple myeloma plasma cells | **IoU** |
| PAIP 21 | Perineural invasion | HD |
| NuCLS 21 | Nuclei (many classes) | IoU, DSC |
| WSSS4LUAD 21 | Tumour / stroma / normal tissue | **IoU** |
| CoNIC 22 | Nuclei (epithelial / lymphocyte / plasma / eosinophil / neutrophil / connective tissue) | **IoU** (as part of the PQ) |

[36] Adapted to include a "detection" aspect, as explained in section 4.1.5.
[37] Idem
[38] Unclear, as will be explained in the Quality Control chapter (Chapter 8).

## 4.3 State-of-the-art of the analyses of metrics

There is a growing amount of literature on the topic of evaluation metrics, within the context of biomedical imaging or in general. Reinke et al. [226] thoroughly examine the pitfalls and limitations of common image analysis metrics in the context of biomedical imaging. Luque et al. [225] examine the impact of class imbalance in binary classification metrics. Their methods and conclusions will be explored more deeply in section 4.3.3, and we extend their analysis in section 4.4.2. Chicco et al. [232] compare the MCC to several other binary classification metrics and find it generally more reliable. Delgado et al. [233] describes the limitations of Cohen's kappa compared to the Matthews correlation coefficient, and some of their results are discussed below in section 4.3.2. Grandini et al [234] propose an overview of multi-class classification metrics, a topic generally less explored in the literature. Padilla et al. [223], [224] compare object detection metrics and note the importance of precisely defining the metrics used, as the AP, for instance, which can be computed in several ways.

Oksuz et al. [235] focus on object detection imbalance problems. They review the different sampling methods that can help to counteract the foreground-background imbalance to train machine learning algorithms. They identify four classes of such methods: hard sampling (where a subset of positive and negative examples with desired proportions is selected from the whole set), soft sampling (where the samples are weighted so that the background class samples contribution to the loss is diminished), sampling-free (where the architecture of the network, or the pipeline, are adapted to address the imbalance directly in the training with no particular sampling heuristic), and generative (where data augmentation is used also to correct the imbalance by increasing the minority class samples). In section 4.4.1, we also explore the effect of foreground-background imbalance on the evaluation process.

### 4.3.1 Relations between classification and detection metrics

Many evaluation metrics are related or take similar forms. The most obvious relationship is between the different forms of the F1-Score, which can be used as a detection measure, a classification measure, and a segmentation measure (as the DSC).

A somewhat less obvious relation exists between the F1-Score and unweighted Cohen's kappa, as identified by Zijdenbos et al. [236]. In a binary classification where we consider a "positive" and a "negative" class, we have the confusion matrix:

$$\begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$$

And $\kappa_U$ resolves to:

$$\kappa_U = \frac{2 \times \big((TN \times TP) - (FN \times FP)\big)}{(TN + FP)(FP + TP) + (TN + FN)(FN + TP)}$$

In a detection problem, TN is uncountable, but it can generally be assumed to verify:

$$TN \gg TP, FP, FN$$

With this assumption, $\kappa_U$ for detection can be simplified to:

$$\kappa_{U,d} = \frac{2 \times TN \times TP}{TN(FP + TP) + TN(FN + TP)} = \frac{2 \times TP}{2 \times TP + FP + FN} = \frac{2 \times PRE \times REC}{PRE + REC}$$

Which is the **harmonic mean** of the PRE and REC, i.e. the detection F1-Score.

Meanwhile, the same exercise applied to the MCC leads us to a "detection" version of:

$$MCC_d = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} = \frac{TP \times TN}{\sqrt{(TP + FP)(TP + FN)TN^2}}$$

$$MCC_d = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} = \sqrt{PRE \times REC}$$

Which is the **geometric mean** of the PRE and REC (note that this is different from the classification GM we used before, which was the geometric mean of the REC and *SPE*, not REC and *PRE*).

While the harmonic mean is typically the preferred averaging methods for ratios (such as the PRE and REC), the geometric mean will penalize less harshly small values for one of the averaged elements. For instance, for $PRE = 0.1$ and $REC = 0.9$, the geometric mean will still be equal to 0.3 while the harmonic mean will drop to 0.18.

### 4.3.2 Interpretation and problems of Cohen's kappa

Cohen's kappa is very popular in digital pathology. It is often associated to an "interpretation" scale such as this one [237]:

- < 0: less disagreement than random chance.
- 0.00-0.20: slight agreement
- 0.21-0.40: fair agreement
- 0.41-0.60: moderate agreement
- 0.61-0.80: substantial agreement
- 0.81-1.00: almost perfect agreement

There is however some debate on the validity of these interpretation, with some authors suggesting that, for medical research in particular, a less optimistic view of the $\kappa$ value should be used [238]. It should also be noted that the kappa values are highly dependent on the weights used, and that this interpretation can therefore be highly misleading if a weighted kappa is used. For instance, in McLean et al.'s study of interobserver agreement in Gleason scoring [239], looking at the *unweighted* kappa would lead to an interpretation of "slight agreement", the *linear* kappa to "slight to fair agreement", and the *quadratic* kappa to "fair to moderate agreement".

This can be problematic when, for instance, Humphrey et al. [240] compare the average *unweighted kappa* of 0.435 of general pathologists on Gleason scoring from a study [241] with the 0.6-0.7 *weighted kappa*[39] for urologic pathologists from another [237] to find an "enhanced interobserver agreement" for the specialists, instead of using the 0.47-0.64 *unweighted kappa* which was also reported in the same study, marking an improvement that is still very substantial but not quite as extreme as Humphrey's paper suggests.

[39] Which in this case did not specify if it was the linear or quadratic version, which is another problem altogether…

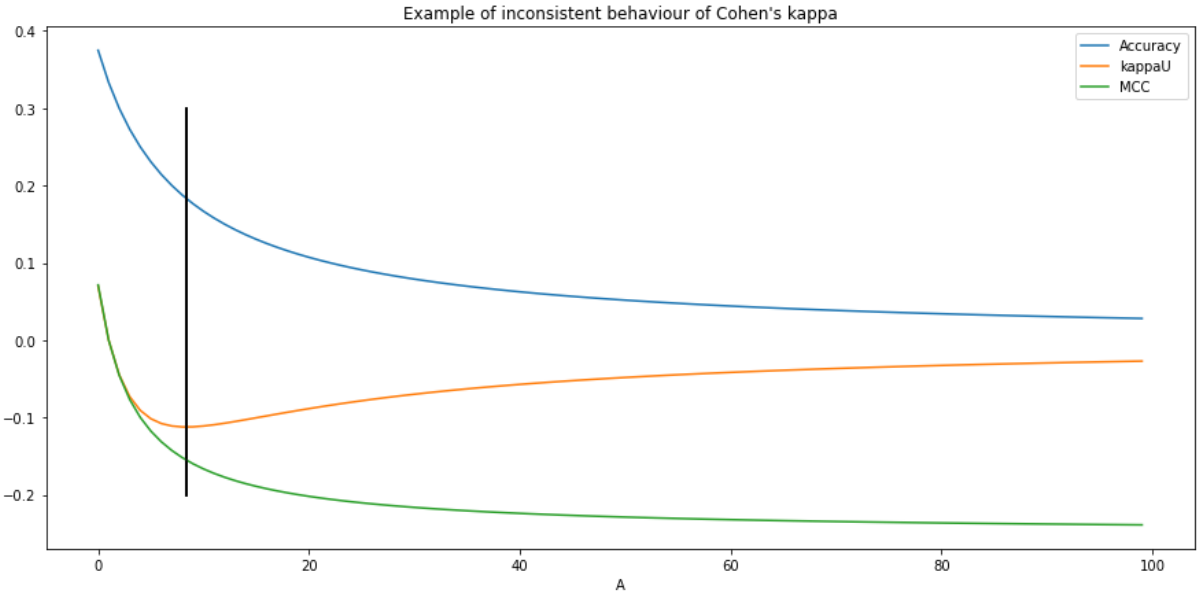



**Figure 4.7. Behaviour of the accuracy, the unweighted kappa and the MCC on a 3 classes confusion matrix of the $Z_A$ family with increasing values of $A$. While the accuracy and MCC are monotically decreasing, the unweighted kappa has an inflexion point shown here with the black line marking $A = 1 + 3\sqrt{6}$.**

As noted by Delgado et al. [233], in some circumstances Cohen's kappa can have an inconsistent behaviour, where a worse agreement leads to better results. In general, this can be demonstrated using the $Z_A$ family of imbalanced confusion matrices, defined by:

$$Z_A = \begin{pmatrix} 1 & \cdots & A \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

So that $Z_{A,ij} = 1$ everywhere except for $Z_{A,0m} = A$, with $A \geq 0$. Intuitively, increasing $A$ means that at the same time we increase the imbalance, and we reduce the agreement. While we would expect in that case $\kappa$ to be monotically decreasing, Delgado et al. show that it is not the case, and that after the inflexion point at $A = 1 + m\sqrt{m(m-1)}$ the $\kappa$ values start to increase. This can be verified experimentally, as we can see in Figure 4.7.

However, this is unlikely to happen, and in practice the $\kappa_U$ in general agrees well with the MCC, as shown in our experiments in section 4.4.5.

### 4.3.3 Imbalanced datasets in classification tasks

The question of how classification metrics behave with imbalanced datasets was systematically explored by Luque et al. [225] for binary classification. In this section, we will briefly explain their method and their results. In section 4.4.2, we will extend their analysis to some metrics that they did not include, and to multi-class problems.

To analyse the metrics' behaviour, the first step is to parametrize the confusion matrix so that it becomes a function of the sensitivity of each class, and of the class imbalance. Let $\lambda_c$ be the sensitivity of class $c$ and $\pi_c$ the proportion of the dataset that is of class $c$, so that $\sum_c \pi_c = 1$ and $0 \leq \lambda_c \leq 1$. As it is a binary classification problem, Luque et al. further define one of the classes as the *positive* class, and the other as the *negative* class. We therefore have $\pi_P = 1 - \pi_N$, and we can

then define the *balance parameter* $\delta = 2\pi_P - 1 = 1 - 2\pi_N$, so that $\delta = 0$ corresponds to balanced data, $\delta = -1$ to an "all-negative" dataset, and $\delta = 1$ to an "all-positive" dataset. The binary confusion matrix can be formulated as:

$$CM = N \begin{pmatrix} \lambda_N \frac{1-\delta}{2} & (1-\lambda_N)\frac{1-\delta}{2} \\ (1-\lambda_P)\frac{1+\delta}{2} & \lambda_P \frac{1+\delta}{2} \end{pmatrix}$$

With $N$ the total number of samples in the dataset.

The study then shows how different classification metrics behave for different values of $(\delta, \lambda_P, \lambda_N)$. The **bias** of a metric due to class imbalance can be characterized by looking at the difference between the metric measured for $(\delta, \lambda_P, \lambda_N)$ with the same metric measured at $(0, \lambda_P, \lambda_N)$ (i.e. for the same per-class sensitivities in a balanced dataset). Figure 4.8 shows for instance how the ACC behaves with imbalanced datasets. The interpretation of the top row is that, as the balance skews towards one of the two classes, only the performance of that class influences the result of the metric, as the isocontours (shown at 0.1 intervals in the figures) become parallel to either the horizontal or the vertical axis. In the bottom row, we can visualise the bias that it introduces for medium ($\delta = \pm 0.5$) and very high ($\delta = \pm 0.99$) imbalance according to per-class sensitivities. By contrast, the unbiased behaviour of the GM is shown in Figure 4.9.

In Table 4.6, we reproduce the range of bias found in Luque et al. [225] for several classification metrics. It should be noted that, for the MCC, the metric is "normalized" with $MCC_n = \frac{1+MCC}{2}$ so that its value is limited to the same [0, 1] range as the other metrics, to make the comparisons more accurate.

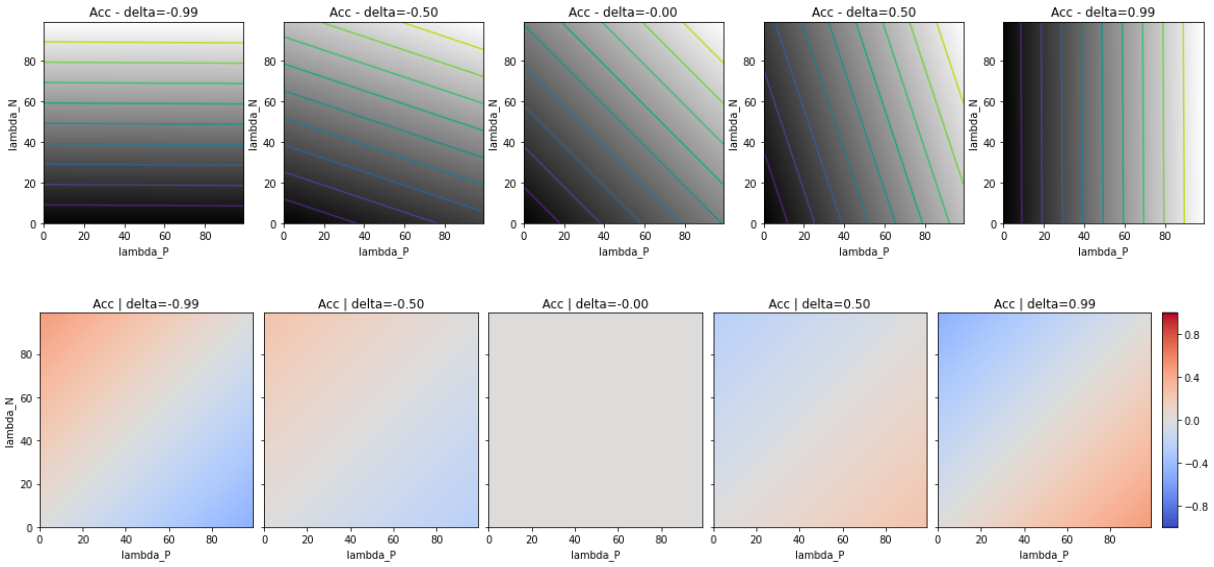



**Figure 4.8. Behaviour of the ACC in a binary classification task with different class imbalance. Top row: result of the metric with isocontours at 0.1 intervals. Bottom row: imbalance bias (difference with the metric at $\delta = 0$), with negative values in blue and positive values in red. For the accuracy, the range of the bias is [-0.49, 0.49].**

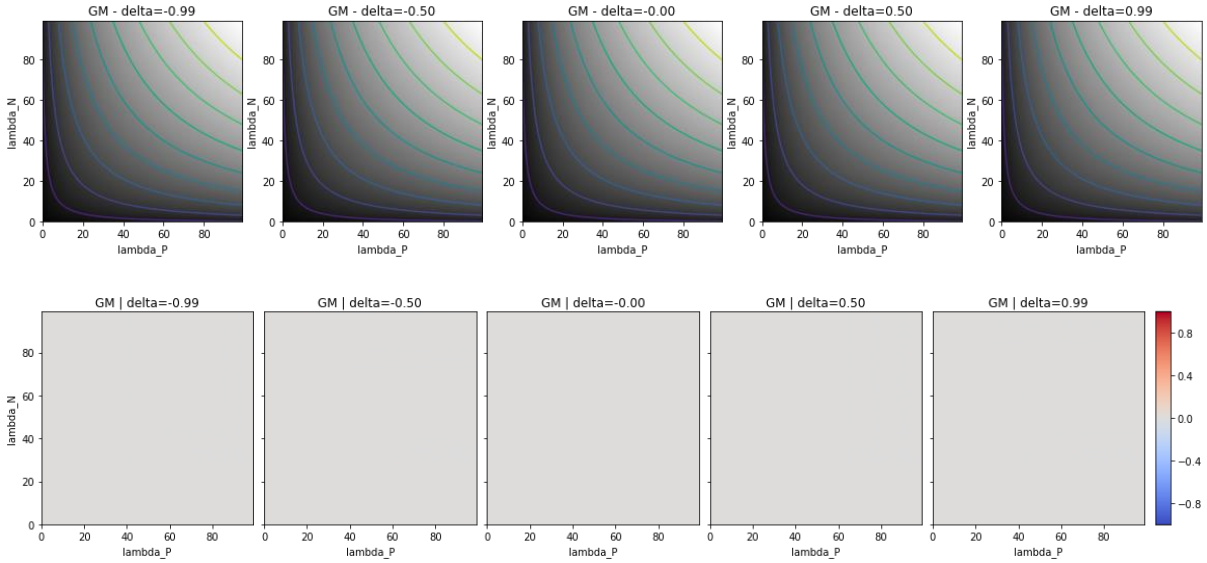



**Figure 4.9. Behaviour of the GM in a binary classification task with different class imbalance. The GM is completely independent from the balance parameter $\delta$, as it only depends on the $SEN$ of the two classes. Top row: result of the metric with isocontours at 0.1 intervals. Bottom row: imbalance bias (difference with the metric at $\delta = 0$), with negative values in blue and positive values in red.**

**Table 4.6. Range of bias for different metrics in binary classification tasks, as reported by Luque et al.** [225]

| Metric | Range of bias |
|---|---|
| $ACC$ | [-0.49, 0.49] |
| $F1_P$ | [-0.86, 0.33] |
| $GM$ | [0, 0] |
| $MCC_n$ | [-0.34, 0.34] |

A limitation of their study is that the F1-Score that they used is the $F1_P$, the "per-class" F1-score of the "positive" class. This is a very common choice in binary classification problems, but it makes the comparison with the $ACC$, $GM$ and $MCC$ slightly uneven, as the $F1_P$ is a *class-specific* metric while the others are all *global* metrics. In fact, the $F1_P$ here is not really a binary classification metric, but rather a **detection** metric, as it does not consider a "two classes" problem but rather a "one class-vs-all" problem.

### 4.3.4 Statistical tests

When looking at results from different algorithms on a given task, it can be very difficult to determine if the difference in metrics between methods can be considered "significant". Part of the question is related to the various sources of uncertainty that come with the annotation process, and will be discussed more in the following chapters, such as interobserver variability and imperfect annotations. Even if we assume "perfect" annotations, however, it is still necessary to determine if a difference in results may be attributed simply to random sampling, or if we can safely reject that hypothesis.

Let's consider datasets $D_i$, which are sampled from a larger, potentially infinite population $P$, and a performance metric $M(D_i, A_j)$ computed on the output of an algorithm $A_j$ for the dataset $D_i$.

Ideally, the null hypothesis would be that the samples of $M(D_i, A_j)$ for the algorithms that we want to compare come from the same distribution.

Practically, however, we generally have a single test dataset which we want to use for our comparison. Different statistical tests have been proposed to provide some confidence in the detection of significant differences in algorithms, depending on the situation. Dietterich [242], for instance, proposes to use the McNemar test for comparing the performances of two classifiers. McNemar's test is based on the contingency table between two classifiers:

| | |
|---|---|
| $n_{00}$ = Number of examples misclassified by $A_1$ and $A_2$ | $n_{01}$ = Number of examples well classified by $A_2$ only |
| $n_{10}$ = Number of examples well classified bv $A_1$ only | $n_{11}$ = Number of examples well classified by $A_1$ and $A_2$ |

The statistic computed is:

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \sim \chi_1^2$$

It follows the $\chi^2$ distribution with one degree of freedom under the null hypothesis.

In general, however, the goal is to be able to compare multiple algorithms based on arbitrary evaluation metrics and the same test set. The recommended test for comparing multiple algorithms on the same data is the **Friedman** non-parametric test [243]–[245]. The Friedman test tells if the null hypothesis that all the tested and dependent samples (in this case the values of the metrics) come from the same distribution can be rejected. To determine the pairwise significance of the differences between algorithms, a *post hoc* test must be conducted. The **Nemenyi** post hoc is a recommended choice [243], although some suggest replacing the post hoc by pairwise Wilcoxon signed-rank tests [244]. The Friedman and Nemenyi tests were used in the statistical score we used in our experiments on imperfect annotations [2], [4] (see Chapter 5).

The **Friedman** test is the non-parametric version of the repeated-measures ANOVA [243]. If we consider a test set $T$ that can be divided into $n$ non-overlapping subsets $S_i, i \in \{1, 2, \dots, n\}$ (for instance, $S_i$ may contain all the patches from a patient in a patch classification task, or all the objects from a patch in a detection or instance segmentation task), so that a performance metric $M$ for the $k$ algorithms $A_j, j \in \{1, 2, \dots, k\}$ can be computed as $m_{ij} = M(S_i, A_j)$.

First, the *rank* of each algorithm is computed on each subset: $r_{ij} = rank(S_i, A_j)$, where $rank(S_i, A_j) = 1$ for the algorithm where $M(S_i, A_j) = \min_{j \in [1,k]} M(S_i, A_j)$. For each algorithm, the *ranks average* $R_j$ is then computed: $R_j = \frac{1}{n}\sum_{i \in [1,n]} r_{ij}$. If all algorithms are equivalent, then their average ranks should be equal. The statistic used to test if this null hypothesis can be rejected is [243]:

$$S = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right]$$

Under the null hypothesis, this statistic follows a $\chi^2$ distribution with $k - 1$ degrees of freedom. If the null hypothesis is rejected according to a p-value threshold (typically $p < 0.05$), the **Nemenyi post hoc** can be used to determine *which* algorithms are significantly different from each other.

The Nemenyi test is based on the average rank difference, computed during the Friedman test, between pairs of algorithms. The ranks depend not only on the performance of the two algorithms being considered, but also on the performance of all the other algorithms compared in the Friedman test. This can be problematic, as it can lead to situations where two algorithms are considered to be significantly different if compared as part of one set of algorithms, and not significant if compared as part of another. For this reason, some recommend using the **Wilcoxon signed-rank test** instead of the Nemenyi for the pairwise analysis (adjusting the significance level to correct for the Type-I error, for instance by dividing the confidence level by the number of compared methods minus one) [243], [244].

Practically, this type of analysis is unfortunately rarely used when comparing deep learning algorithms. At best, challenges will sometimes provide box-plots of the distributions of results for the top teams (as for instance in the ACDC@LungHP 2019 [180] or in the PAIP 2019 [182] challenges), or 95% confidence intervals alongside the average results (as in MoNuSeg 2018 [177]).

## 4.4 Experiments and original analyses

In this section, we present several experiments that we performed in order to extend these analyses, or to explore more thoroughly some of the most interesting aspects of the behaviour of the metrics. Specifically, we will look at the effects of different types of class imbalance (4.4.1, 4.4.2), the consequences of the fuzzy boundary between detection and classification tasks (4.4.3, 4.4.4), the agreement between the scores given by different classification metrics for the same set of predictions (4.4.5), the biases of common segmentation metrics (4.4.6), the difficulties in multi-metrics aggregation (4.4.7) and, finally, at why the Panoptic Quality should be avoided for instance segmentation and classification problems (4.4.8).

### 4.4.1 Foreground-background imbalance in detection metrics

As detection tasks are "one class versus all" problems, they are often associated with a very large "foreground-background" imbalance, meaning that the objects of interest are sparsely distributed in the images. This is particularly true in the most popular detection task in digital pathology: mitosis detection.

In the MITOS12 mitosis detection challenge, for instance, the training dataset consists of 35 image patches, each with dimensions of 2048×2048 pixels. There are 226 mitoses in this training set, occupying a total mitosis area of around 135.000 pixels, which corresponds to about 0.09% of the whole area.

The detection process, illustrated in Figure 4.10, typically involves three main phases: first finding "candidate" regions with a "candidate selector", then classifying each candidate as a positive or negative detection with a "candidate classifier", then finally merging overlapping regions into detected object instances.

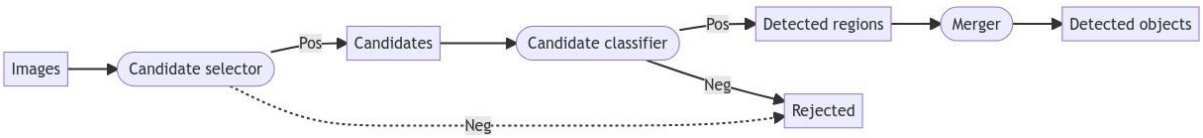



**Figure 4.10. Typical detection process. The dashed line is used to note that the "candidates" rejected at the selector stage are generally not countable (as most technically possible regions will never be considered at all)**

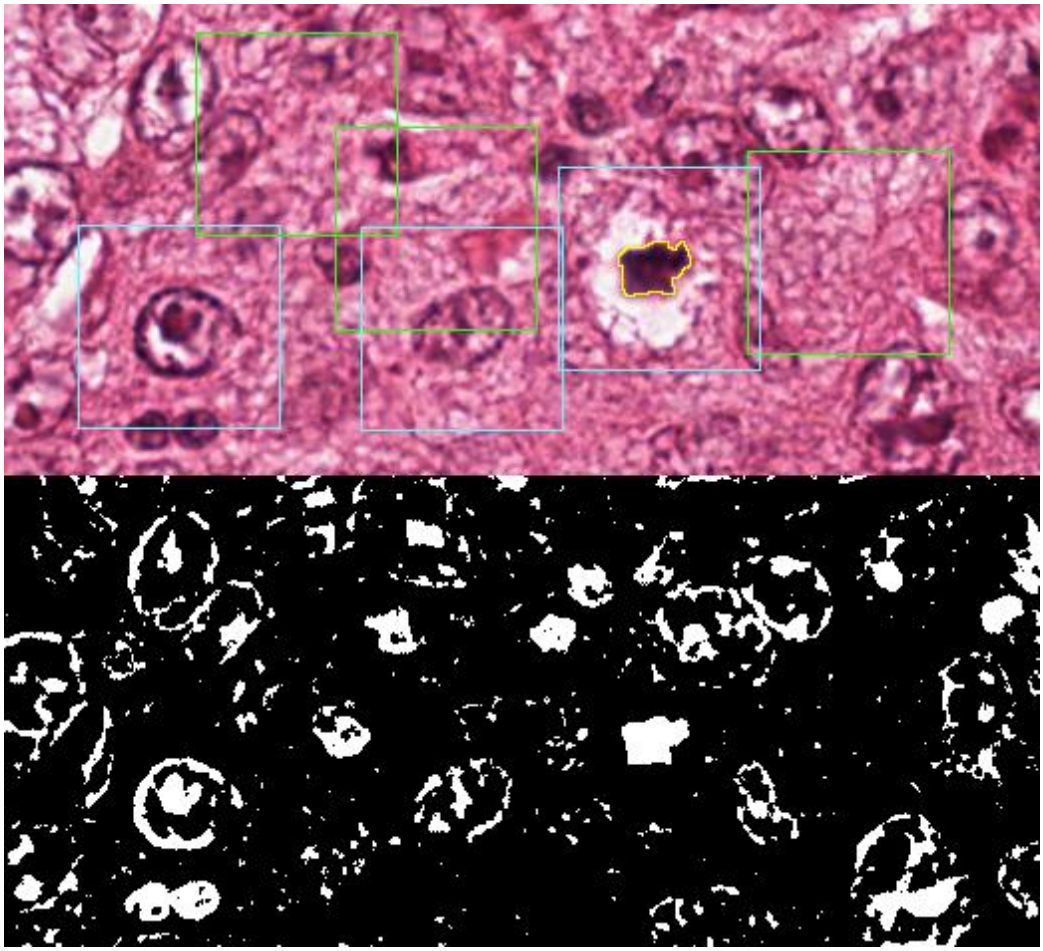


**Figure 4.11. Small region from an image patch in the MITOS12 dataset. A mitosis is highlighted in yellow. Green bounding boxes denote candidate regions that could be trivially rejected, while cyan bounding boxes show candidate regions that are more difficult to classify. The bottom image shows in simple thresholding that could trivially remove a large portion of the negative candidates in a pre-selection step.**

For the first phase, there are two main strategies: either create a *very large number of candidates* by densely sampling the space of possible bounding boxes, or create a *restricted set of candidates*, using the image features already in the first phase to reject "obvious" negatives. The main advantage of the first approach is that it generally ensures that no positive object is accidentally rejected in the first phase, and therefore not even presented to the candidate classifier. The main advantage of the second is that it makes the task of the candidate classifier a lot easier.

The overall performance of the whole detection pipeline, however, is not just affected by the performance of the selector, classifier and merger, but also, as in all machine learning tasks, by the

selection of the possible data used for the evaluation. In the MITOS12 dataset, the 35 image patches correspond to small portions of 5 WSI, selected because of the presence of several mitosis. When the choice of regions of interest is not random, the distribution of positive & negatives changes with the size of the selected region. Taking a larger region "around" the mitosis that first attracted the attention of the pathologist is likely to increase the proportion of negatives, while taking a narrower region would increase the proportion of positives. Given the sparsity of the objects of interest, the effect can quickly become large.

To quickly simulate how this may affect the results of a detection pipeline, we define a detector with the following characteristics:

- The "candidate selection" step filters out 99.99% of the possible candidates at the pixel level, assuming that we define the set of possible candidates as same-sized patches centred on each pixel of the image (so there are as many possible candidates before selection as there are pixels in the image). This assumes that most possible candidates will be trivially dismissed, which is not unrealistic in a mitosis detection problem, as all pixels that are relatively lighter are certain not to be part of a mitosis (see Figure 4.11).
- The "candidate classifier" step has a 99% specificity and 75% sensitivity rate.

We then look at three scenarios. In the first one, we take the characteristics of the MITOS12 dataset (226 mitosis occupying 0.09% of the total pixel area). In the second, we imagine a more restricted set where a smaller region of interest was considered. We look at what happens if, when restricting the total area by half, we keep about 75% of the mitosis. In the third scenario, we do the opposite and double the total area, and simulate that this includes 150% of the original amount of mitosis. The results presented in Table 4.7 show that, while the recall in all cases remains the same (as it is equal to the sensitivity that we fixed as a parameter of the pipeline), the precision varies as the foreground-background imbalance changes. As proportionally more negative examples are shown, the number of false positives can only increase and the number of true positives can only decrease, so the precision and F1-score can only get worse.

**Table 4.7. Precision, recall and F1-score of a simulated detection pipeline on different foreground-background imbalance scenarios based around the MITOS12 distribution.**

| Scenario | Precision | Recall | F1 |
|---|---|---|---|
| MITOS12 distribution | 0.55 | 0.75 | 0.63 |
| Smaller region | 0.64 | 0.75 | 0.69 |
| Larger region | 0.47 | 0.75 | 0.58 |
| MITOS-ATYPIA-14 distribution | 0.18 | 0.75 | 0.29 |

Of course, as long as methods are compared on exactly the same test set, a ranking of the methods based on the F1-Score should still be representative of their relative performance. However, the characteristics of the dataset are still important to keep in mind when interpreting the results, particularly when the same task is measured on different datasets. The MITOS-ATYPIA-14 dataset, for instance, has a much lower ratio of mitosis to total area, with mitosis representing 0.02% of the total area of the patches. A detection pipeline with the same behaviour as the one we simulated on MITOS12 would see its precision fall from 0.55 to 0.18, and its F1-score from 0.63 to 0.29 just by this change of distribution.

It is tempting to see that as the main explanation for the difference in results observed between the two challenges, as the top result for MITOS12 was a 0.78 F1-score, while it was 0.36 for MITOS-ATYPIA-14. There were, however, less participants in the mitosis detection part of the challenge in the 2014 version, and the MITOS12 challenge was also designed with a problematic train/test split, with patches extracted from the same WSI appearing in both parts of the dataset. The difference in the mitosis density is very likely, however, to be a contributing factor.

### 4.4.2 Imbalanced datasets in classification tasks

#### *4.4.2.1 Extension to other metrics*

As noted in section 4.3.3, the analysis of Luque et al. [225] on the effects of imbalanced datasets in classification tasks uses a "detection" F1-Score, which is not a global metric for a classification problem. We therefore reproduce the experiment with the two version of the macro-averaged F1-score, the $sF1$ and the $hF1$. We also add the $\kappa_{Un}$, normalized similarly as the $MCC_n$ (cf. section 4.3.3), for completeness' sake. Our results are presented in Table 4.8, and a comparison between the biases of the $F1_P$, the $hF1$, the $sF1$ and the $MCC_n$ is shown in Figure 4.12.

As we can see, the shape of the bias is very similar between the $hF1$ and the $MCC_n$. While the $F1_P$ is clearly extremely biased with imbalanced datasets and should generally be avoided as a binary classification metric, the use of a macro-averaged version of the metric reduces its bias quite significantly and puts it in the same range as the $MCC$ (see Table 4.6) for the $sF1$, and even lower for the $hF1$. The latter therefore appears to be more robust to class imbalance than the MCC.

**Table 4.8. Range of bias for different metrics in binary classification tasks, from our experiments (complement to Table 4.6).**

| Metric | Range of bias |
|---|---|
| $sF1$ | [-0.43, 0.16] |
| $hF1$ | [-0.28, 0.14] |
| $\kappa_{Un}$ | [-0.41, 0.49] |

#### *4.4.2.2 Extension to multi-class problems*

In a multi-class problem, the notion of a "positive" and "negative" class has to be removed. The generic parametrized confusion matrix is therefore, for $m$ classes:

$$CM = N\begin{pmatrix} \lambda_{11}\pi_1 & \cdots & \lambda_{1(m-1)}\pi_1 \\ \vdots & \lambda_{ii}\pi_i & \lambda_{i(m-1)}\pi_i \\ \lambda_{(m-1)0}\pi_{m-1} & \lambda_{(m-1)i}\pi_{m-1} & \lambda_{(m-1)(m-1)}\pi_{m-1} \end{pmatrix}$$

Where $N$ is the total number of samples, $\pi_c$ is the proportion of class $c$ in the dataset, and $\lambda_{ij}$ is the proportion of class $i$ samples that were misclassified as class $j$, so that $\lambda_{ii} = SEN_i$ and $\sum_j \lambda_{ij} = 1$.

This leads to a parametric space with a very high dimensionality, that is quite complex to explore. For a 3-class problem, we would have two "balance" parameters $\pi_1$ and $\pi_2$ (which set $\pi_3 = 1 - \pi_1 - \pi_2$) and six "error distribution" parameters $\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}, \lambda_{31}, \lambda_{32}$ (which set $\lambda_{13} = 1 - \lambda_{11} - \lambda_{12}$, etc.). In general, for a $m$ class problem, we would have $m - 1$ balance parameters and $m(m-1)$ error distribution parameters.

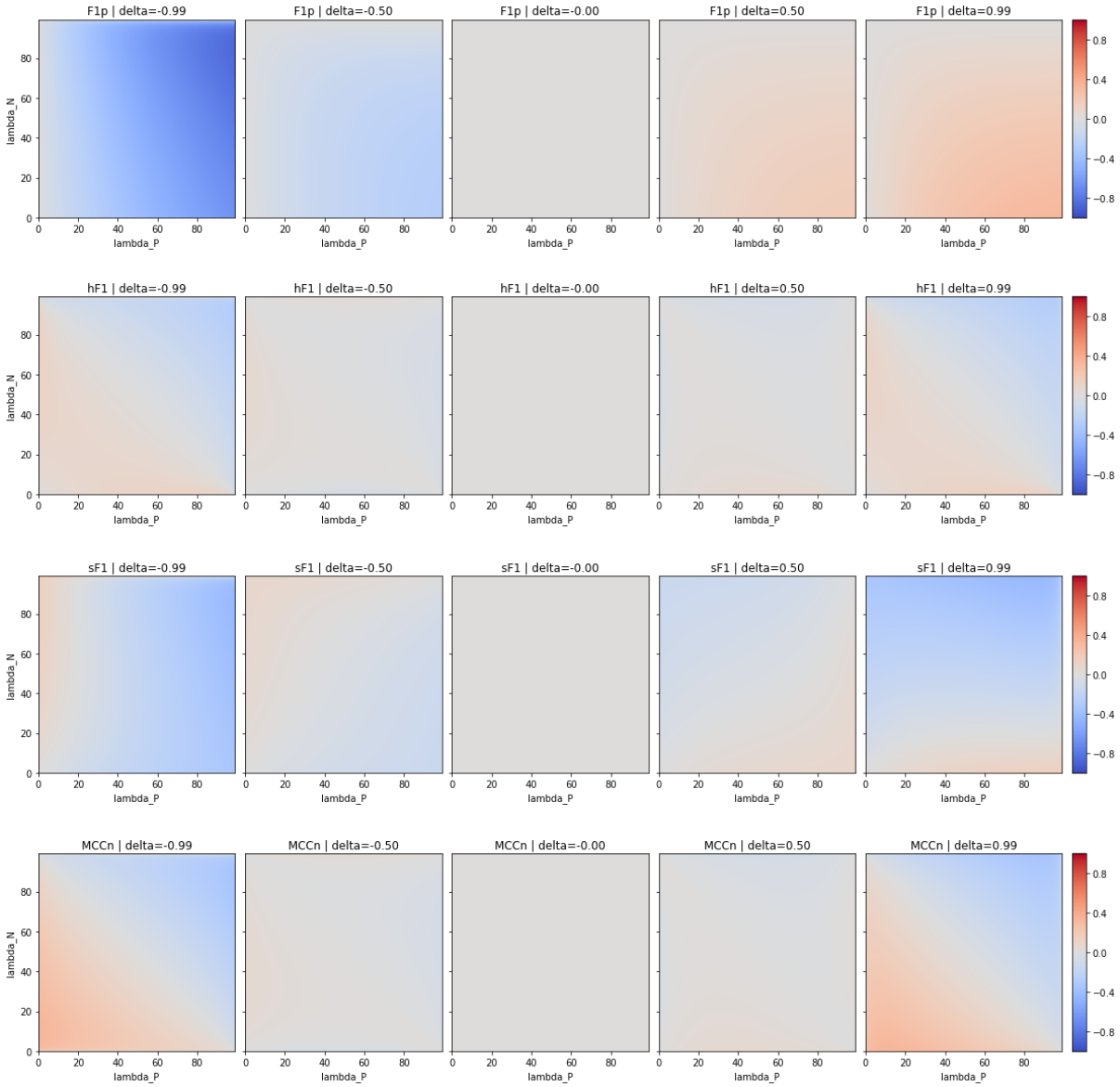



**Figure 4.12. Comparison between the biases of (top to bottom) the $F1_P$, the $hF1$, the $sF1$ and the $MCC_n$ with imbalanced datasets in binary classification tasks.**

To reduce this dimensionality, we define a single **balance parameter $\boldsymbol{\beta}$** that will vary from 0 (balanced) to 1 (extremely imbalanced). We also define $\mathrm{B}(\beta, m) = \beta + \frac{1-\beta}{m}$, and then recursively $\pi_1 = \mathrm{B}(\beta, m)$ and $\pi_i = \mathrm{B}\big(\beta, m-(i-1)\big)\prod_{j=1}^{i-1}\Big(1-\mathrm{B}\big(\beta, m-(j-1)\big)\Big)$. An example of the class distributions yielded by this method for a 4-class problem and different $\beta$ values is shown in Figure 4.13.

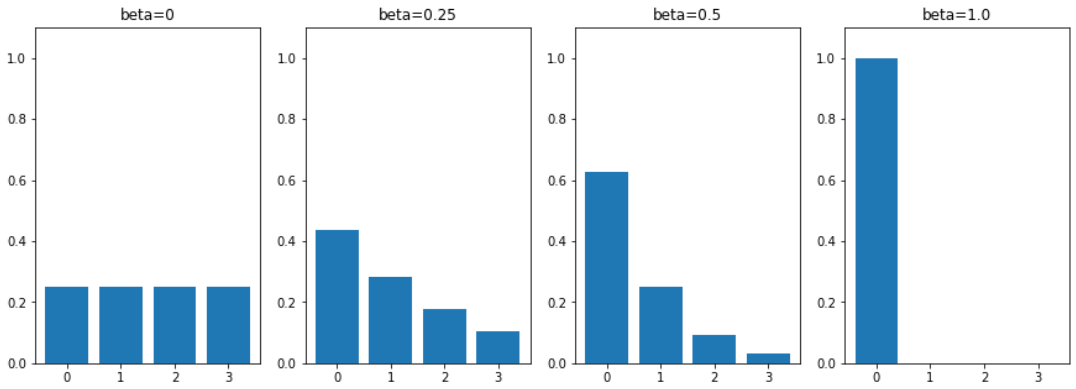



**Figure 4.13. Class distribution given by the $\beta$-parametrization of the imbalance in a 4-class problem.**

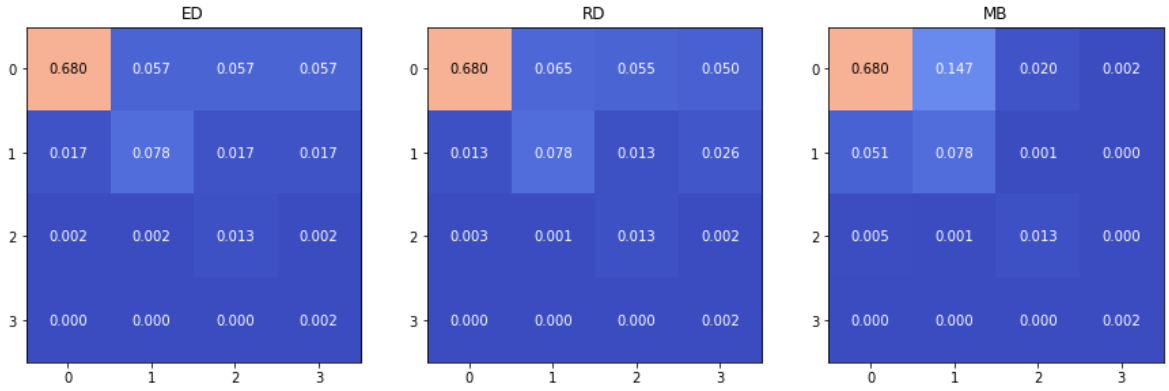



**Figure 4.14. Illustration of the CM (normalized by the total number of samples) obtained with the "evenly distributed" (ED), "randomly distributed" (RD) and "majority bias" (MB) scenarios in a 4-class problem with $\beta = 0.8$ and a class sensitivity vector $\lambda_{ii} = [0.8, 0.6, 0.7, 0.9]$.**

For the **error distribution**, we consider three scenarios that allow us to only have as parameters the $m$ class sensitivity values:

- The **evenly distributed error (ED)** scenario, where we set $\lambda_{ij} = \frac{1-\lambda_{ii}}{m-1}$ $\forall j \neq i$.
- The **randomly distributed error (RD)** scenario, where we first generate a random "error likelihood" matrix $e$ so that $e_{ii} = 0$ and $\sum_j e_{ij} = 1$, and the $\lambda_{ij}$ are then simply computed as $\lambda_{ij} = e_{ij}(1 - \lambda_{ii})$ $\forall\, i \neq j$. The error matrix needs to be computed first so that the same "error likelihood" is used when exploring the space of $\lambda_{ii}$ values.
- The **majority bias error (MB)** scenario, where the errors are distributed more towards the majority class, so that $\lambda_{ij} = (1 - \lambda_{ii})\frac{\pi_j}{\sum_{k \neq i} \pi_k}$.

The three scenarios are illustrated in Figure 4.14. The biases related to the class imbalance are reported in Table 4.9 for $m = 3$ and $m = 4$, in the three error distribution scenarios. The imbalance bias for the $ACC$ and the $hF1$ increases as the number of classes increases. For the $MCC_n$, the main effect seems to be that the negative bias (i.e. the imbalanced result is worse than the balanced result at the same sensitivity values) becomes more important than the positive bias, so the $MCC_n$ in general will be lower in a multi-class, imbalanced dataset. The $sF1$ keeps a relatively constant bias. The $\kappa_{Un}$ sees the range of its bias reduced, but only on the positive side. Finally, the GM remains unbiased.

**Table 4.9. Range of bias found in the three error distribution scenarios for $\beta \in \{0.5, 0.99\}$ and $m = 3$ and $m = 4$, with the previous results for $m = 2$ repeated to ease the comparison.**

| $m = 2$ | $ACC$ | $GM$ | $MCC_n$ | $hF1$ | $sF1$ | $\kappa_{Un}$ |
|---|---|---|---|---|---|---|
| - | [-0.49, 0.49] | [0, 0] | [-0.34, 0.34] | [-0.28, 0.14] | [-0.43, 0.16] | [-0.41, 0.49] |
| $m = 3$ | $ACC$ | $GM$ | $MCC_n$ | $hF1$ | $sF1$ | $\kappa_{Un}$ |
| ED | [-0.65, 0.65] | [0, 0] | [-0.36, 0.2] | [-0.42, 0.13] | [-0.42, 0.18] | [-0.42, 0.24] |
| RD | [-0.65, 0.65] | [0, 0] | [-0.37, 0.21] | [-0.43, 0.12] | [-0.42, 0.17] | [-0.42, 0.24] |
| MB | [-0.65, 0.65] | [0, 0] | [-0.37, 0.17] | [-0.42, 0.14] | [-0.44, 0.18] | [-0.42, 0.24] |
| $m = 4$ | $ACC$ | $GM$ | $MCC_n$ | $hF1$ | $sF1$ | $\kappa_{Un}$ |
| ED | [-0.73, 0.73] | [0, 0] | [-0.38, 0.19] | [-0.52, 0.11] | [-0.41, 0.19] | [-0.43, 0.20] |
| RD | [-0.73, 0.73] | [0, 0] | [-0.38, 0.18] | [-0.52, 0.11] | [-0.41, 0.23] | [-0.43, 0.20] |
| MB | [-0.73, 0.73] | [0, 0] | [-0.45, 0.19] | [-0.52, 0.14] | [-0.45, 0.18] | [-0.43, 0.20] |

#### *4.4.2.3 Normalized confusion matrix to counter class imbalance*

To counteract the bias induced by class imbalance, the most obvious solution is to compute the metrics on the **balanced** (or **normalized**) **confusion matrix** (NCM). The NCM is computed by simply dividing each element by the sum of the values on the same row:

$$NCM_{ij} = \frac{CM_{ij}}{\sum_k CM_{ik}}$$

This acts as a "resampling" of the data so that each class has the same number of samples, while keeping the per-class sensitivity values intact. This solution, however, is not without its own drawbacks. To illustrate these, we need to use some real data. The results of the MoNuSAC 2020 challenge can be used in this case: as the prediction maps of four teams are available, it is possible to compute alternate metrics to those reported in the challenge results.

**Table 4.10. Confusion matrices for the classification part of the MoNuSAC challenge, recomputed for the four teams based on the available test set annotations and teams' prediction maps.**

| Team 1 | E | L | N | M | Team 2 | E | L | N | M |
|---|---|---|---|---|---|---|---|---|---|
| **E** | 6098 | 260 | 8 | 12 | **E** | 6240 | 88 | 0 | 57 |
| **L** | 79 | 7214 | 2 | 1 | **L** | 162 | 7131 | 4 | 6 |
| **6** | 5 | 39 | 118 | 2 | **N** | 1 | 18 | 133 | 10 |
| **M** | 16 | 11 | 8 | 170 | **M** | 16 | 1 | 11 | 164 |
| **Team 3** | **E** | **L** | **N** | **M** | **Team 4** | **E** | **L** | **N** | **M** |
| **E** | 5960 | 96 | 0 | 1 | **E** | 6193 | 302 | 2 | 22 |
| **L** | 76 | 6864 | 3 | 0 | **L** | 179 | 7274 | 1 | 0 |
| **N** | 1 | 20 | 137 | 3 | **N** | 3 | 38 | 117 | 2 |
| **M** | 30 | 8 | 11 | 159 | **M** | 30 | 7 | 25 | 155 |

The MoNuSAC challenge is a nuclei instance segmentation and classification challenge. We will focus here on the classification part of the challenge, which has four classes (epithelial, neutrophil, lymphocyte and macrophage). From the published prediction maps, we can compute the classification confusion matrices (based on the matching pairs of detected nuclei, which will therefore vary between the teams), shown in Table 4.10, and the number of per-class TP, FP and FN for the detection task, shown in Table 4.11. A full description of the MoNuSAC dataset is given in Annex A.

**Table 4.11. Detection errors in the MoNuSAC challenge, recomputed for the four teams based on the available test set annotations and teams' prediction maps. FP detections are the number of predicted objects, for each class, with no corresponding ground truth (of any class). FN detections are the number of ground truth objects, for each class, with no corresponding prediction (of any class). TP detections are the sum of each row in Table 4.10**

| **FP detections** | **E** | **L** | **N** | **M** |
|---|---|---|---|---|
| **Team 1** | 1338 | 829 | 14 | 59 |
| **Team 2** | 962 | 629 | 5 | 285 |
| **Team 3** | 2932 | 1545 | 50 | 99 |
| **Team 4** | 1035 | 770 | 10 | 13 |
| **FN detections** | **E** | **L** | **N** | **M** |
| **Team 1** | 831 | 507 | 8 | 102 |
| **Team 2** | 824 | 500 | 10 | 115 |
| **Team 3** | 1152 | 860 | 11 | 99 |
| **Team 4** | 690 | 349 | 12 | 90 |
| **TP detections** | **E** | **L** | **N** | **M** |
| **Team 1** | 6378 | 7296 | 164 | 205 |
| **Team 2** | 6385 | 7303 | 162 | 192 |
| **Team 3** | 6057 | 6942 | 161 | 208 |
| **Team 4** | 6519 | 7454 | 160 | 217 |

**Table 4.12. Class proportions $\pi_C$ in the annotations, and range of values extracted from Table 4.10 and Table 4.11 for the predicted class proportions $\pi_C$, the class detection recall values $REC_C$ and classification sensitivity values $SEN_C$ for the four teams whose predictions are available in the MoNuSAC challenge.**

| **Range** | **Epithelial** | **Lymphocyte** | **Neutrophil** | **Macrophage** |
|---|---|---|---|---|
| Annotations $\pi_C$ | 0.47 | 0.50 | 0.01 | 0.02 |
| Predicted $\pi_C$ | $[0.46 - 0.50]$ | $[0.47 - 0.52]$ | $[0.01 - 0.01]$ | $[0.01 - 0.03]$ |
| Detection $REC_C$ | $[0.84 - 0.90]$ | $[0.89 - 0.96]$ | $[0.93 - 0.95]$ | $[0.63 - 0.71]$ |
| Classif. $SEN_C$ | $[0.95 - 0.98]$ | $[0.98 - 0.99]$ | $[0.72 - 0.85]$ | $[0.71 - 0.85]$ |

From these results, we can look at the relationship between the $SEN_C$ values and the class imbalance. From Table 4.12, we can see that the two majority classes are associated to much higher sensitivity values than the two minority classes. If we are trying to judge how the algorithm would perform "in a world where the data is balanced", this normalization may therefore not be completely accurate.

Another possible problem that may arise from the normalization is that, if we have a very large imbalance with some very rare classes, we may artificially increase what is simply "sampling noise". To illustrate this, we again turn to the MoNuSAC results. In this experiment, we keep all the properties of the dataset (class distribution of the ground truth) and of each team's results (detection recall and per-class distribution of classification errors, represented by the $NCM$). We then randomly sample N nuclei from an "infinite" pool of samples that has the same class proportions as the ground truth and compute each team's confusion matrix based on that sampling. On average, we will obtain confusion matrices that are the same as those of the challenge, but we will also be able to see the variation due to random sampling.

**Table 4.13. Uncertainty due to random sampling for each metric, based on 5000 simulations using the MoNuSAC class distribution and the computation of maximum divergence on the four teams' performances. At 100.000 samples, all differences are ≤ 0.001. The maximum value(s) for each line are bolded.**

| **15.000 samples** | $\boldsymbol{ACC}$ | $\boldsymbol{GM}$ | $\boldsymbol{MCC_n}$ | $\boldsymbol{hF1}$ | $\boldsymbol{sF1}$ | $\boldsymbol{\kappa_{Un}}$ |
|---|---|---|---|---|---|---|
| CM | 0.001 | 0.003 | 0.001 | **0.006** | 0.001 | 0.001 |
| NCM | **0.003** | **0.003** | 0.002 | **0.003** | 0.002 | 0.002 |
| **5.000 samples** | $\boldsymbol{ACC}$ | $\boldsymbol{GM}$ | $\boldsymbol{MCC_n}$ | $\boldsymbol{hF1}$ | $\boldsymbol{sF1}$ | $\boldsymbol{\kappa_{Un}}$ |
| CM | 0.002 | 0.011 | 0.002 | **0.011** | 0.001 | 0.002 |
| NCM | 0.010 | **0.011** | 0.006 | 0.009 | 0.007 | 0.006 |
| **1.000 samples** | $\boldsymbol{ACC}$ | $\boldsymbol{GM}$ | $\boldsymbol{MCC_n}$ | $\boldsymbol{hF1}$ | $\boldsymbol{sF1}$ | $\boldsymbol{\kappa_{Un}}$ |
| CM | 0.006 | **0.056** | 0.005 | 0.046 | 0.003 | 0.005 |
| NCM | 0.052 | **0.056** | 0.035 | 0.054 | 0.035 | 0.034 |

In Table 4.13, we report the results from 5.000 runs of the simulation based on 1.000, 5.000 or 15.000 samples (15.000 being close to the real number of annotated nuclei in the MoNuSAC test set). We show, for each classification metric, the maximum divergence observed across the four teams ($\max_i(\mathrm{P}_{97.5}(metric, team_i) - \mathrm{P}_{2.5}(metric, team_i))$, with $P_x$ the x percentile) as an uncertainty measure. As we can see, if we have many annotated samples like in MoNuSAC, the uncertainty due to the random sampling remains very low even in the NCM. If we have a smaller set of samples, however, this uncertainty can quickly grow, at least for the $ACC, MCC_n, sF1$ and $\kappa_{Un}$. As the $GM$ metric is unbiased relative to the imbalance, the results do not change between CM and NCM, but the random sampling uncertainty is very high. The $hF1$ also has a high uncertainty overall.

Overall, the uncertainty added by using the NCM instead of the CM remains relatively low but, if the sample size is small, it is still in a range that could potentially affect the comparison between algorithms (as many biomedical datasets have much fewer than 1.000 samples available).

#### *4.4.2.4 Conclusions on imbalanced datasets in classification tasks*

Given the above analysis, the flowchart presented in Figure 4.15 attempts to summarize how to decide which classification metrics to use based on the class imbalance of the dataset.

With **mostly balanced** datasets, the confusion matrix can be used directly to compute any global metrics. Which of these to use will mostly depend on whether the distribution of the errors among the classes is important for the evaluation or not. Figure 4.16 shows three confusion matrices corresponding to a balanced dataset with the same total error, but with different distributions per class. Table 4.14 shows the results of the different global metrics for these confusion matrices.

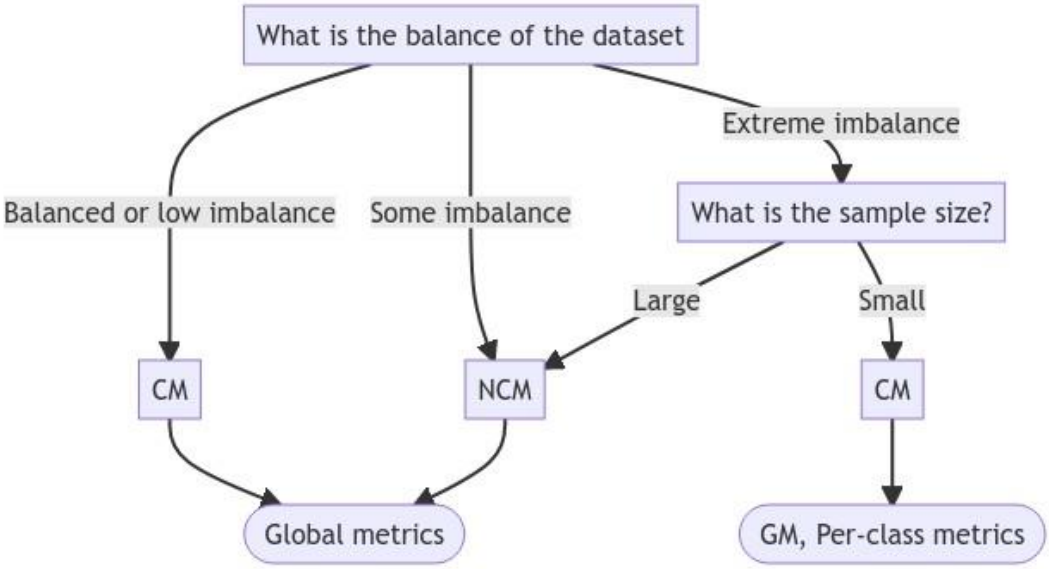



**Figure 4.15. Flowchart on which classification metrics to use based on the dataset balance.**

Obviously, the **accuracy** is the same for all three: it is completely independent from how the error is distributed, as long as the total error remains the same. Less obvious is the invariance of the **unweighted kappa**, which is expected to depend also on the distribution of the predicted classes, which here varies between the confusion matrices. With a *perfectly balanced* dataset, however, we can show that the $\kappa_U$ is actually independent from the error distribution.

Using the original definition of the kappa:

$$\kappa_U = \frac{p_o - p_e}{1 - p_e}$$

It is clear that $p_o$, which is the accuracy, will not vary. Meanwhile, we have for $m$ classes and $n$ samples:

$$p_e = \frac{\sum_i (\sum_k CM_{ik} \sum_k CM_{ki})}{n^2}$$

And, in a balanced dataset:

$$\sum_k CM_{ik} = \frac{n}{m} \; \forall i \in \{1, 2, \dots, m\}$$

That is, the sum of each row will be the same. Therefore:

$$p_e = \frac{1}{mn} \sum_i \sum_k CM_{ik} = \frac{1}{m}$$

And finally:

$$\kappa_U = \frac{ACC - \frac{1}{m}}{1 - \frac{1}{m}}$$

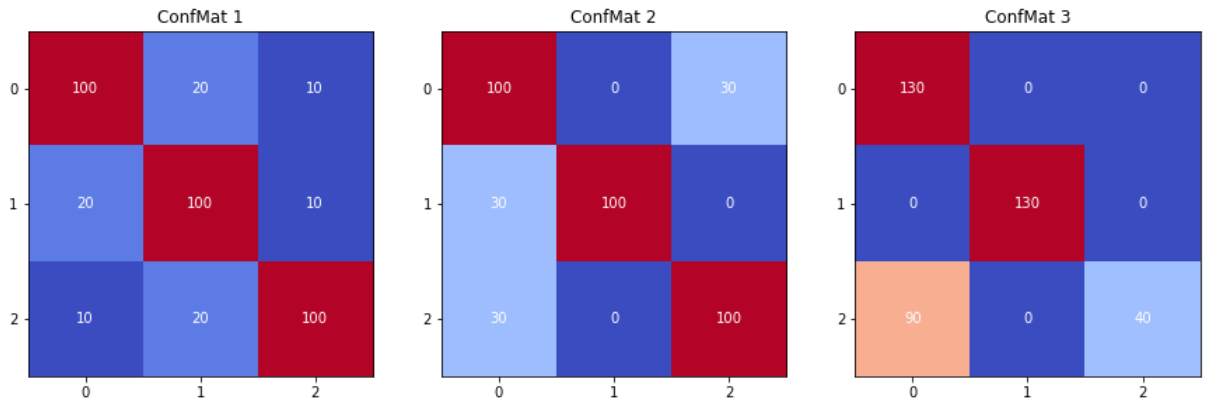



**Figure 4.16. Example of different error distributions in 3 class confusion matrices.**

Which is illustrated in Table 4.14 with $m = 3$. If there is any class imbalance, this relationship no longer holds true.

The **Geometric Mean** is equal for the first two confusion matrices, for which the $SEN_C$ are equal whereas the errors in each class are distributed differently. The **Matthews Correlation Coefficient** and the two **macro-averaged F1 scores**, on the other hand, have different values for the three confusion matrices, and are therefore affected by changes in the distribution of errors. Whether this is a desired trait for the metric depends on the task and the evaluation criteria. It is interesting to note, however, that the latter three *increase* as the error gets progressively more *concentrated* in the three confusion matrices, even in the third one where all the prediction errors concern one class only. This contrasts greatly with the GM, which penalises the third confusion matrix a lot more, as the much smaller $SEN_C$ for class 2 will greatly affect the score.

If the dataset is **imbalanced** but with still a reasonable number of samples of each class (what is a reasonable amount will depend on how much sampling uncertainty is acceptable for the purpose of the evaluation of the task), using the **normalized confusion matrix** to reduce the problem to a balanced problem is a good solution, with the same choice of metrics available.

In cases of **extreme imbalance** with a **small sample size**, where there are very few samples available for the minority class, the normalization may induce too much sampling uncertainty. It is therefore probably advisable in those cases to focus on an unbiased global metric like the **GM**, or on **class-specific metrics** separately.

**Table 4.14. Classification metrics computed on the three confusion matrices from Figure 4.16.**

| | $ACC$ | $GM$ | $\kappa_U$ | $MCC$ | $hF1$ | $sF1$ |
|---|---|---|---|---|---|---|
| **ConfMat 1** | 0.769 | 0.769 | 0.654 | 0.654 | 0.771 | 0.755 |
| **ConfMat 2** | 0.769 | 0.769 | 0.654 | 0.660 | 0.783 | 0.780 |
| **ConfMat 3** | 0.769 | 0.675 | 0.654 | **0.713** | **0.814** | **0.871** |

### 4.4.3 Limit between detection and classification

An interesting aspect of object detection tasks is their close relationship to binary classification tasks. The main difference, in the definition we have used for the task, is that an "object detection" task does not have a countable number of true negatives, but as we have seen a detection pipeline is often composed of a "candidate selection" step and a "classification" step.

The danger of this confusion between detection and classification is that it is tempting to use the confusion matrix of the classification part and its associated metrics to measure the detection performance of an algorithm. For instance, Giusti et al. [246] compare humans and algorithms on a mitosis detection task using pre-sampled image patches forming a balanced dataset, and score them with a ROC curve, which takes the TN into account. To illustrate the impact that the definition of the task has on the evaluation, we simulate four scenarios on synthetic data.

In each scenario, we define five categories of "candidates":

- Negative, trivial ($N_0$): corresponds to negative candidates from a sampling method that will always be correctly rejected by the classifier.
- Negative, easy ($N_E$): negative candidates that the classifier will rarely fail to reject.
- Negative, hard ($N_H$): negative candidates that are similar enough to the objects of interest that the classifier will regularly misclassify them as positive.
- Positive, hard ($P_H$): positive candidates that are hard to separate from the non-objects and that the classifier will regularly reject.
- Positive, easy ($P_E$): positive candidates that the classifier will rarely miss.

Two of the scenarios correspond to "handpicked" candidates, where in one case negative examples are chosen for being "difficult" to classify, and in the other they are chosen to be easier. We also simulate two "candidate selectors" (see Figure 4.10), the first one a non-specific selector that largely sample the set of possible candidates, leading to many candidates that are trivially easy to reject to be added to the set, while the other simulates a more targeted selector which tends to add more "hard" negative examples but also includes some trivial examples.

Each scenario differs by the number of candidates presented from each category. In all four, the number of $P_H$ and $P_E$ are the same (as they correspond to the positive objects, which will be the same regardless of the "candidate selection" procedure) and is set to 500. In the "Handpicked Hard" scenario, negative examples are added with a majority of $N_H$ and no $N_0$. In the "Handpicked Easy" scenario, negative examples are added with a majority of $N_E$ and no $N_0$. Finally, in the "Large selector" scenario, 5000 trivial candidates are added, and $N_E = N_H = 800$. All the values are reported in Table 4.15.

**Table 4.15. Number of candidates in each category for the four scenarios.**

| Scenario | $N_0$ | $N_E$ | $N_H$ | $P_H$ | $P_E$ |
|---|---|---|---|---|---|
| Handpicked Hard | 0 | 200 | 800 | 500 | 500 |
| Handpicked Easy | 0 | 800 | 200 | 500 | 500 |
| Large selector | 5000 | 800 | 800 | 500 | 500 |
| Targeted selector | 50 | 600 | 800 | 500 | 500 |

**Table 4.16. Detection (REC, PRE, AP) and classification (SPE, MCC) metrics computed for the four scenarios with the same "classifier". To account for the random sampling in the classifier, each metric is the arithmetic mean over 500 repetitions of the experiment. The maximum standard deviation observed was 0.017. Bolded values show the best scenario according to each metric.**

| Scenario | REC | PRE | AP | SPE | MCC |
|---|---|---|---|---|---|
| Handpicked hard | 0.85 | 0.77 | 0.92 | 0.75 | 0.60 |
| Handpicked easy | 0.85 | **0.93** | **0.98** | 0.94 | **0.79** |
| Large selector | 0.85 | 0.77 | 0.92 | **0.96** | 0.78 |
| Targeted selector | 0.85 | 0.77 | 0.92 | 0.83 | 0.67 |

We define a single classifier which outputs a random "confidence" value for each class, sampled from a normal distribution $Normal(\mu, \sigma)$ with the following characteristics:

- $N_0 \rightarrow Normal(0, 0)$
- $N_E \rightarrow Normal(0.1, 0.1)$
- $N_H \rightarrow Normal(0.45, 0.1)$
- $P_H \rightarrow Normal(0.55, 0.1)$
- $P_E \rightarrow Normal(0.8\ 0.05)$

The REC, PRE and AP are used as detection metrics, and the SPE and MCC as classification metrics. The results obtained by the "classifier" based on these parameters and using a confidence threshold $\tau = 0.5$ are reported in Table 4.16.

The REC, which only depends on the positive examples, are equal in all cases. The PRE and SPE, however, are unsurprisingly much larger when the handpicked negative examples are easier. Looking at detection metrics (PRE and AP), the "large selector", "targeted selector" and "handpicked hard" scenarios are identical. The classification metrics, however, vary widely depending on the scenario, and on which negative candidates were added to the dataset.

This illustrates some of the pitfalls of confusing detection and classification tasks: it may lead to using metrics which are not adapted to the problem. If a manual selection of the candidates is done, it includes a potential bias to the results, as the candidate distribution may no longer be representative of the actual data present in the images in a real situation.

### 4.4.4 Use of detection metrics for instance classification

As previously noted, the results of an **instance classification** task can be described with a confusion matrix which combines the detection and classification aspects of the results, with a negative ("no object/background") class alongside the target classes.

Such problems are often treated as "**multi-class detection**", so that detection metrics are computed per target class and then averaged, with for instance a macro-averaged F1-Score ($sF1$ or $hF1$). It should be noted, however, that a "macro-averaged F1-Score" in a *multi-class detection* problem is not exactly the same as a "macro-averaged F1-score" in a *classification* problem. In the latter, as there is no "background class", all the elements on the diagonal are both true positives (of their own class) and true negatives (of the other classes), while all the elements off the diagonal are both false positives (of their "predicted" class) and false negatives (of their "ground truth" class). In multi-class detection, however, this is not true for the elements on the first row (which

are each "false positives" of their predicted class only) and for the elements on the first column (which are each "false negatives" of their ground truth class only).

Indeed, this leads to certain biases to these averaged metrics, as they will penalize "misclassifications" more harshly than "missed detections" and "false detections". To illustrate that, we can start from a "perfect" 2-classes detection confusion matrix with 10 objects in each class:

$$CM_{perfect} = \begin{pmatrix} N.A. & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

If we transform one of the correct predictions from class 1 into a "**misclassification**", we get:

$$CM_{mc} = \begin{pmatrix} N.A. & 0 & 0 \\ 0 & 9 & 1 \\ 0 & 0 & 10 \end{pmatrix}$$

For the two target classes, the computed TP, FP and FN are:

$$TP_1 = 9, FP_1 = 0, FN_1 = 1, TP_2 = 10, FP_2 = 1, FN_2 = 0$$

And therefore:

$$F1_1 = 0.947, F1_2 = 0.952, \boldsymbol{sF1 = 0.950}$$

If the misclassification is transformed into a "**missed detection**", we get:

$$CM_{md} = \begin{pmatrix} N.A. & 0 & 0 \\ 1 & 9 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

Leading to:

$$TP_1 = 9, FP_1 = 0, FN_1 = 1, TP_2 = 10, FP_2 = 0, FN_2 = 0$$

And:

$$F1_1 = 0.947, F1_2 = 1.0, \boldsymbol{sF1 = 0.974}$$

Finally, if, instead of a "missed detection", we add a "**false detection**", we get:

$$CM_{fd} = \begin{pmatrix} N.A. & 1 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

$$TP_1 = 10, FP_1 = 1, FN_1 = 0, TP_2 = 10, FP_2 = 0, FN_2 = 0$$

$$F1_1 = 0.952, F1_2 = 1.0, \boldsymbol{sF1 = 0.976}$$

We have two different biases in action here. Falsely positive detections (with no corresponding ground truth objects) are penalized less than falsely negative detections (missed objects). The worse type of error, however, is misclassification. This can lead to very counterintuitive results, as methods that completely miss objects may be ranked higher than methods that find the objects but fail to classify them properly.

To avoid this issue, a better strategy if possible is to separate the two problems into a "single-class" detection problem, where all the "target" classes are grouped into one, and a "multi-class"

classification problem, where only the objects that were properly detected are considered. In our example, we therefore obtain detection ($CM_{d,.}$) and classification ($CM_{c,.}$) matrices as follows:

$$CM_{d,mc} = \begin{pmatrix} N.A. & 0 \\ 0 & 20 \end{pmatrix}, CM_{d,md} = \begin{pmatrix} N.A. & 0 \\ 1 & 19 \end{pmatrix}, CM_{d,fd} = \begin{pmatrix} N.A. & 1 \\ 0 & 20 \end{pmatrix}$$

And:

$$CM_{c,mc} = \begin{pmatrix} 9 & 1 \\ 0 & 10 \end{pmatrix}, CM_{d,md} = \begin{pmatrix} 9 & 0 \\ 0 & 10 \end{pmatrix}, CM_{d,fd} = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$$

The detection and classification metrics can then be computed separately. The advantage of this approach is that it makes it immediately clear what the specific weaknesses of the algorithm are. The "missed detection" and "false detection" cases will both have an imperfect detection score, but a perfect classification score, while the "misclassification" case will have a perfect detection score but an imperfect classification score. The insights that we gain from the evaluation metrics are therefore much improved, and the ranking of the methods can be adjusted according to the needs of the clinical application being considered. This is particularly appropriate if the target classes are naturally part of a "superclass": for instance, the "epithelial", "lymphocyte", "neutrophil" and "macrophage" classes of MoNuSAC can be grouped into a "nuclei" superclass for the purpose of measuring the detection performance.

### 4.4.5 Agreement between classification metrics

There is a large diversity of classification metrics, and it can be difficult to really get a sense of their levels of agreement or disagreement. To quantify inter-metrics agreement, we perform a simple experiment. A synthetic dataset is constructed with some randomized features: the number of classes, the size, the number of features, the difficulty, and the imbalance. A Support Vector Machine (SVM) classifier is trained on 90% of the samples and tested on the remaining 10%. The values of the different classification metrics are then computed on the predicted test set. This experiment is repeated 50 times, leading to 50 values for each metric. The Spearman correlation coefficient between each pair of metrics is then computed to form a "similarity matrix", which is shown in Figure 4.17. Such a matrix is, however, not particularly easy to interpret. Two interesting ways of visualising the (dis)similarity are Multi-Dimensional Scaling (MDS) [247] and dendrograms.

Dendrograms (see Figure 4.18) are very useful to quickly visualize clusters from a dissimilarity matrix and are for instance used to compare classification metrics in a study by Ferri et al. [248], and to visualize the similarity of different tasks based on challenge results by Wiesenfarth et al. [249].

With MDS (see Figure 4.19), the dissimilarities (defined here as $1 - r$, with $r$ the Spearman correlation coefficient) are projected into a 2D space so that the distance between the points on this 2D space correspond, as much as possible, to the computed dissimilarities. This was proposed by Bouix et al. [250] as a way to evaluate classifiers without a ground truth, but it is an interesting way to visualize (dis)similarities in general.

These figures clearly show that the MCC and $\kappa_U$ are the most similar metrics and mostly behave identically, with the $\kappa_L$ often relatively close and the $\kappa_Q$ joining them in a loose cluster. The ACC and hF1 also tend to be relatively close to each other, as do the $MAUROC$ and $\mu AUROC$. The $sF1$ and $GM$ are further apart from every other metric.

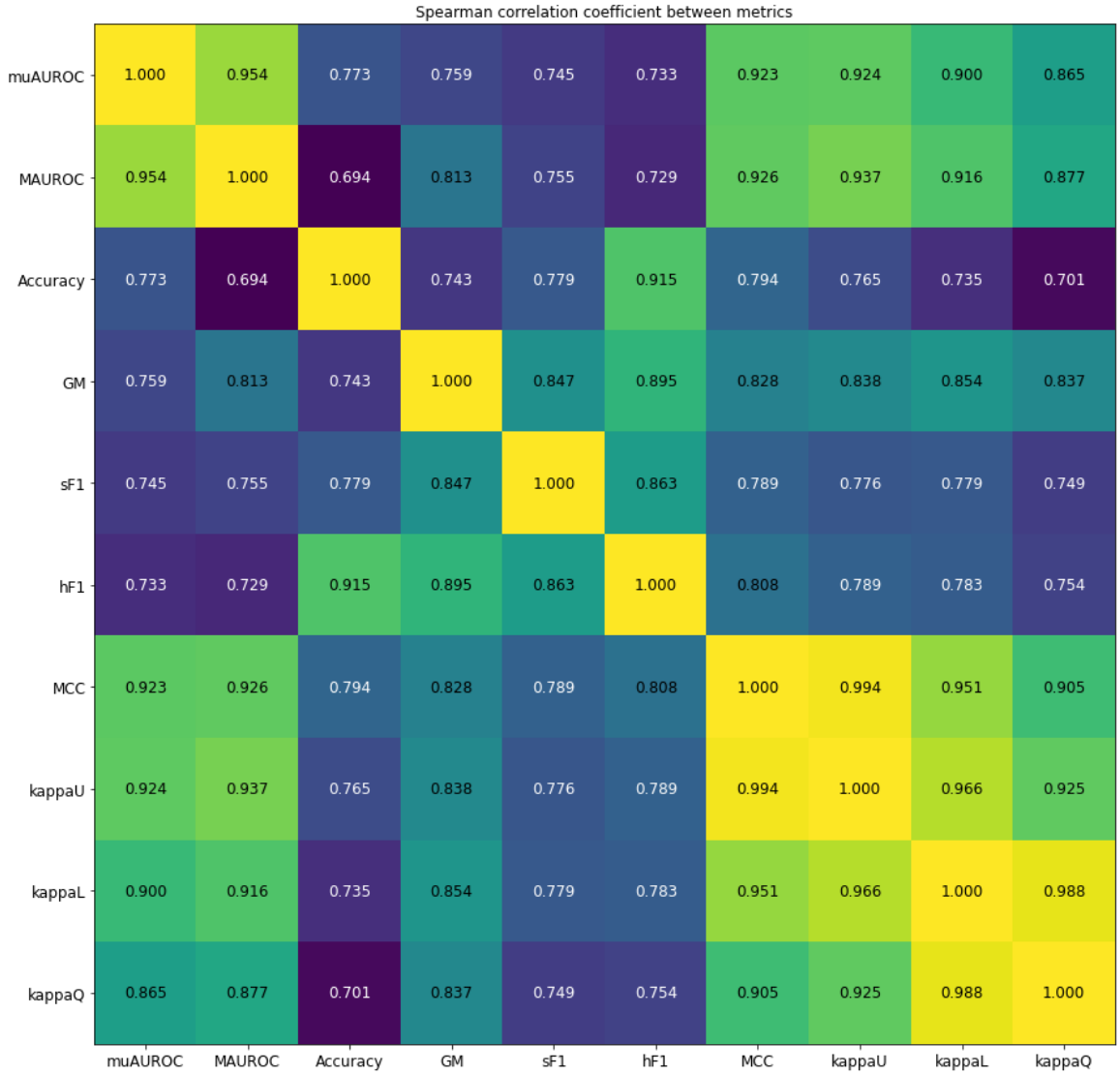



**Figure 4.17. Spearman correlation coefficients between pairs of metrics based on the results of a SVM classifier on 50 randomly created datasets with different number of classes, imbalance, and difficulty.**

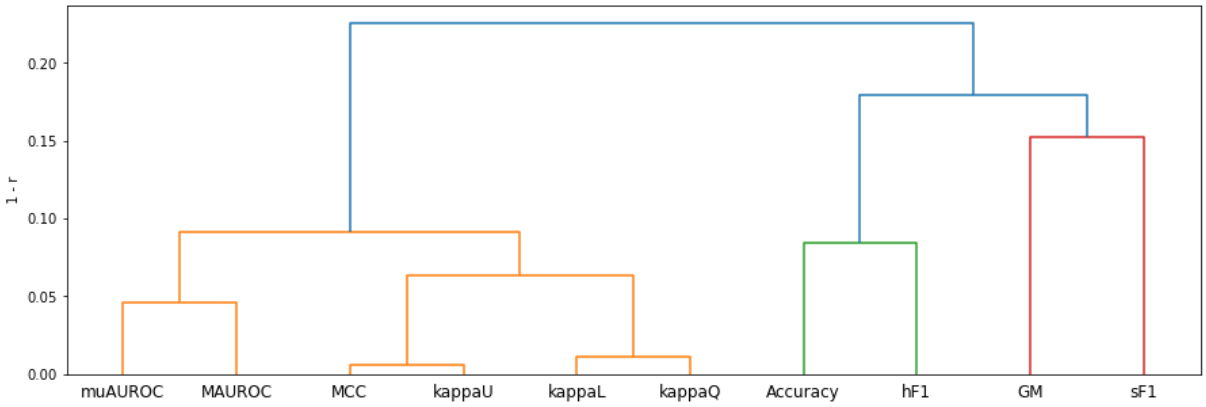



**Figure 4.18. Dendrogram visualisation of the similarity matrix shown in Figure 4.17, using the "average" (UPGMA) inter-cluster dissimilarity update method (as implemented in the Scipy library[40])** [251].

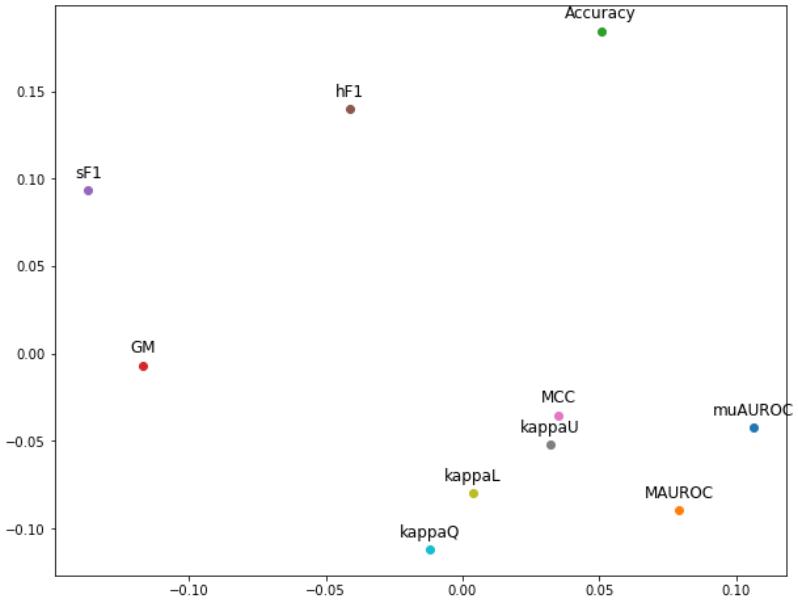



**Figure 4.19. MDS visualisation of the similarity matrix shown in Figure 4.17, with the "dissimilarity" defined as $1 - r$.**

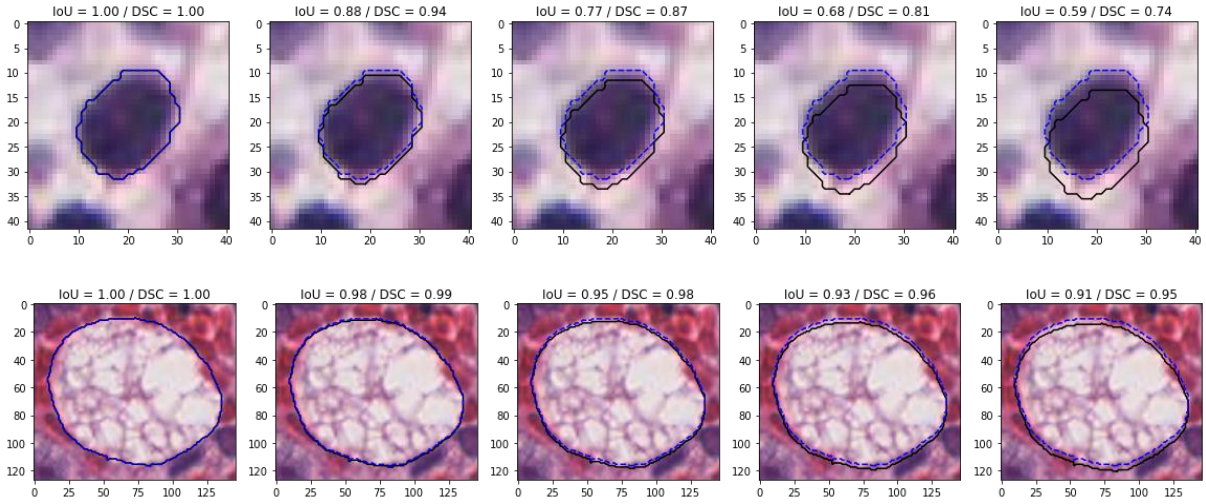



**Figure 4.20. Effect on the IoU of shifting the position of the predicted segmentation by steps of one pixel, based on an annotated lymphocyte nucleus of ~300px (top) and macrophage of ~10000px (bottom) from the MoNuSAC 2020 dataset. The dashed blue line corresponds to the true annotation, the black line to the the prediction shifted by increments of 1 pixel. The corresponding HD would be 0, 1, 2, 3 and 4px (or about 0, 0.25, 0.5, 0.75 and 1 µm) in both cases.**

### 4.4.6 Biases of segmentation metrics

Overlap-based metrics and distance-based metrics behave very differently with regards to the characteristics of the segmented objects, and in particular with their relative size. Overlap-based metrics, for instance, will be invariant to a rescaling of both prediction and ground truth masks. A distance-based metric, meanwhile, will be rescaled as well if it is expressed in pixels. Whenever

[40] https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html

possible, these metrics should therefore be expressed in distances on the physical sample (for instance, in µm).

#### 4.4.6.1 *Effect of small object size on overlap metrics*

Overlap-based metrics, however, tend to be very sensitive to small prediction errors when the objects of interest have very small areas. Because of the fuzziness of the object borders in many digital pathology tasks, **any predicted segmentation, even accurate, is likely to have some misalignment in the pixels around the border of the object**. This will cause a number of FP and FN pixels that will tend to be correlated to the perimeter of the object. The TP region, meanwhile, will tend to be correlated with the object area. As the $\frac{Perimeter}{Area}$ ratio will generally be higher for smaller objects, the corresponding IoU will therefore tend to be lower, even for a very good segmentation. For objects which are very small even at high levels of magnification, such as nuclei, this can lead to very problematic results. A shift of a single pixel of the annotation contour (which could correspond to a small bias of the annotation stylus, for instance) can have a very large impact on the IoU, as shown in Figure 4.20. On a lymphocyte with an area of around 300px, the single pixel shift brings the perfect IoU of 1.0 down to 0.88, even though it is arguably as "exact" as the original given the fuzzy nature of the actual border, while a 4px shift brings it down to 0.59, close to the typical matching threshold of 0.5. For the much larger (around 10000px) macrophage, the single pixel shift has an almost perfect IoU of 0.98, and the 4px shift only brings it down to 0.91.

If we look at all the annotated objects in the MoNuSAC dataset, we can plot the relationship between their area and this "single pixel shift" IoU, as shown in Figure 4.21. Once objects are smaller than a ~2.500px area, single pixel shifts start dropping the IoU very quickly. The IoU obtained in the different classes of the dataset therefore has a very different interpretation, as an IoU of 0.8 for a (large) macrophage would correspond to a much "worse" segmentation than for a (small) lymphocyte.

A possible solution to this problem is to include the uncertainty into the computation of the metrics. Similarly to the $HD_\delta$ and $NSD_\delta$ defined in section 4.1.4, we can define uncertainty-aware version of the IoU and DSC. If we define the "outer object" $T_\delta^+ = \{x_i \in {\sim}T; d(x_i, T) \leq \delta\}$ and the "inner object" $T_\delta^- = \{x_i \in T; d(x_i, {\sim}T) > \delta\}$, we can redefine the per pixel TP, FP and FN as:

$$TP_\delta = |P \cap T_\delta^+|, FP_\delta = |P \cap {\sim}T_\delta^+|, FN_\delta = |{\sim}P \cap T_\delta^-|$$

And therefore:

$$IoU_\delta = \frac{TP_\delta}{TP_\delta + FP_\delta + FN_\delta}, DSC_\delta = 2 \times \frac{TP_\delta}{2 \times TP_\delta + FP_\delta + FN_\delta}$$

The results of this uncertainty-aware overlap metrics are illustrated in Figure 4.22 for $\delta = 1$, showing that the $IoU_\delta$ and $DSC_\delta$ remain high when the prediction stays within the uncertainty region.

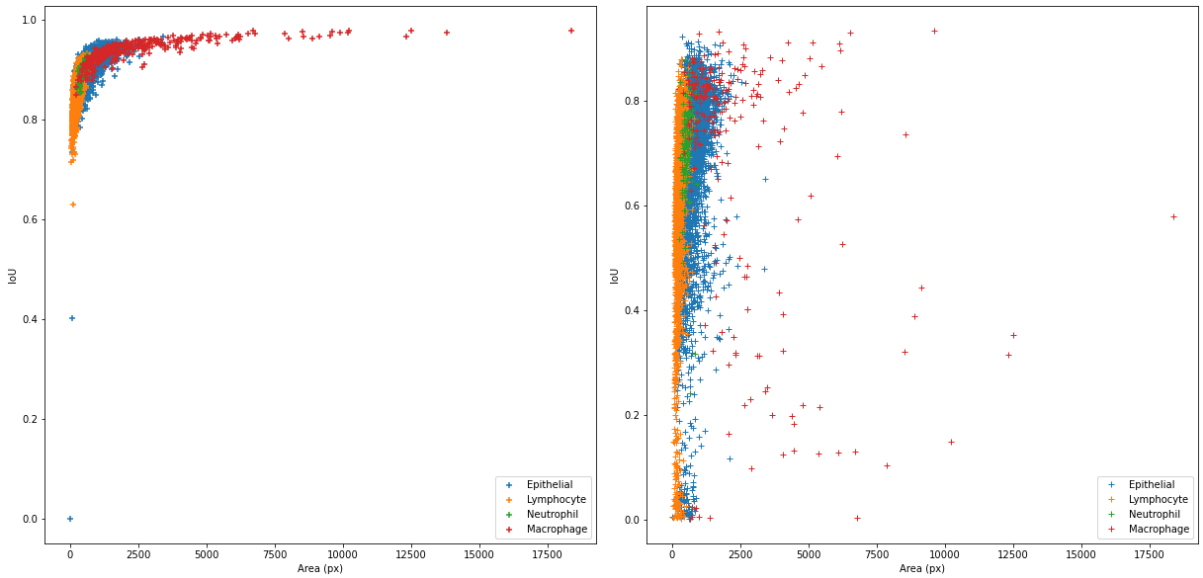



**Figure 4.21. Relationship between object area (in pixel) and IoU obtained (left) after a single pixel vertical shift, and (right) on the predictions of "Team 1", on all annotated objects of the MoNuSAC dataset.**

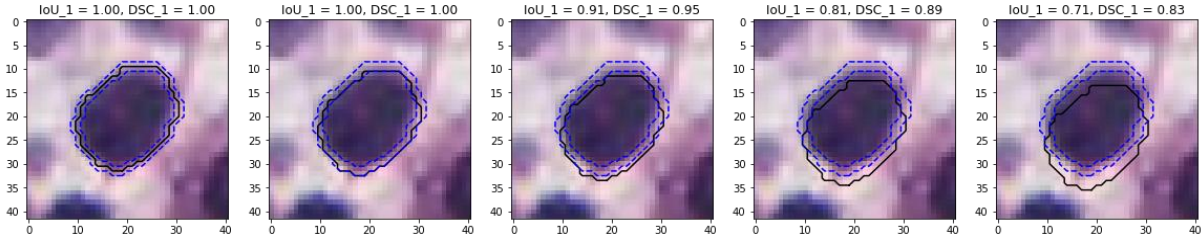



**Figure 4.22. Pixel-shift experiment using the uncertainty-aware DSC and IoU with $\delta = 1$ on an image from the MoNuSAC test set. Dashed blue lines indicate the limits of the "outer" and "inner" regions, while the black line represents the prediction shifted by increments of 1 pixel, as in Figure 4.21.**

To explore how the characteristics of the IoU on small objects can affect the performance evaluation of competing algorithms, we turn to the MoNuSAC challenge results. The evaluation metric used in MoNuSAC is the Panoptic Quality, which includes the IoU as a "segmentation" component. Several examples of the results from the four teams whose predictions are available on nuclei of different types are shown in Figure 4.23. On the first row, Team 3 and 4 are harshly penalized for having very slightly overestimated the size of the object compared to the ground truth. Team 3's segmentation only gets an IoU of 0.57, while Team 4, which is still very good, is near the 0.5 matching threshold used in the challenge. On the other hand, Team 1 and 2 are not penalized by their more biologically problematic shape mismatch.

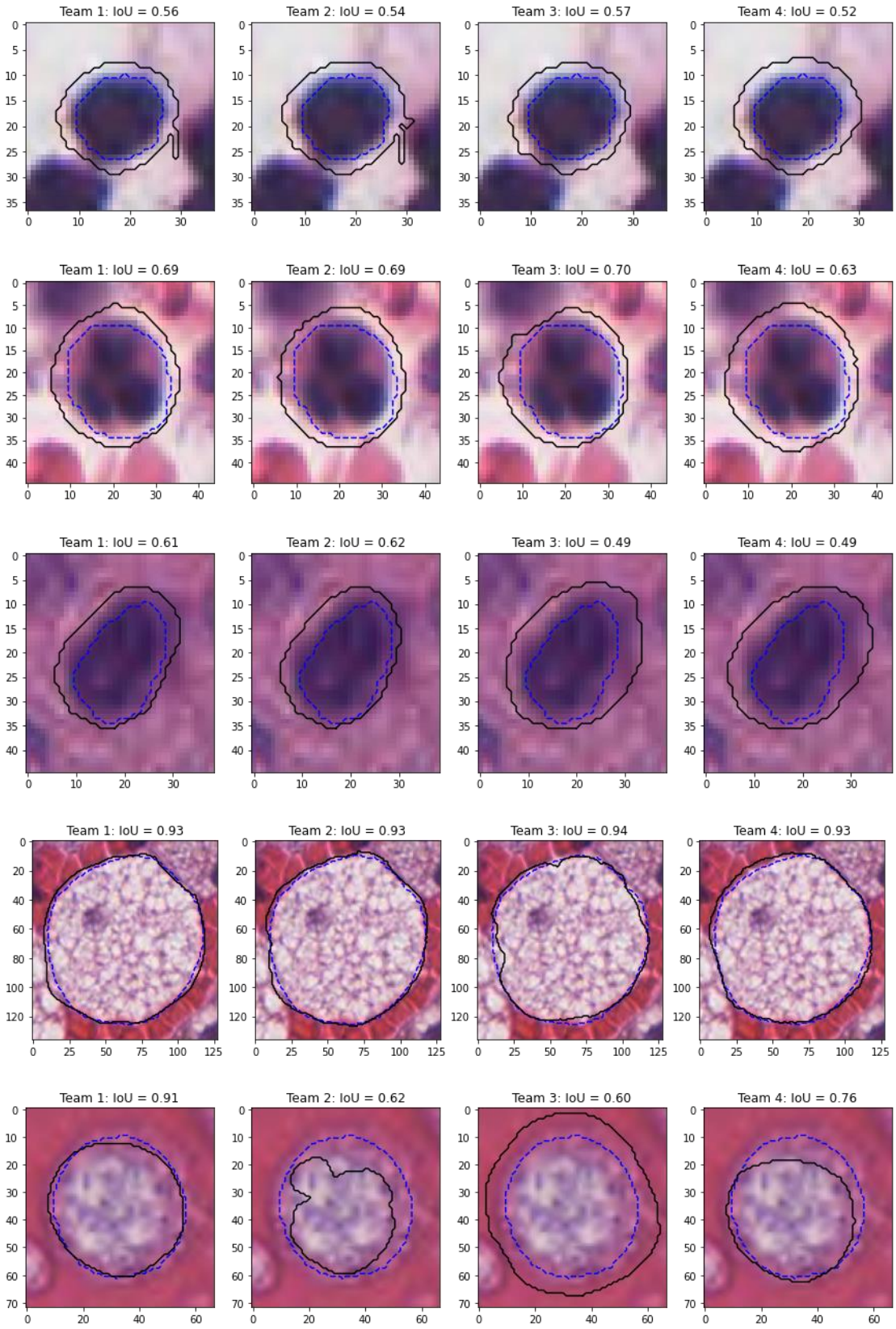



**Figure 4.23. Predictions of four competing teams on some nuclei from the MoNuSAC 2020 challenge. From top to bottom: lymphocyte (~200px), neutrophil (~500px), epithelial (~300px), and two macrophages (~9600px and ~1800px). Ground truth annotations are shown with dashed blue lines, predicted segmentation with black lines.**

In the second row, the four predictions are mostly equivalent, yet there is still a wide range of IoU, from 0.63 to 0.70 for Team 3, which provides the least similar overall shape and yet the best score. On the third row, Team 3 and 4 both fall under the 0.5 matching threshold, largely due to the fact that the annotation appears to be slightly off to the right and bottom. On the fourth row, we see that for a much larger object the score are a lot higher, despite a segmentation that is not necessarily "better" biologically than those of the smaller objects above. On the last row, we see that for those larger objects the differences between the teams' results make a lot more sense, but still fail to recognize the more meaningful types of errors, such as the bad shape of Team 2's result, compared to the size overestimation by Team 3. The HDs computed on the last rows yield results of 3.0, 13.6, 9.0 and 9.2px respectively, more accurately capturing in this case the problematic nature of Team 2's results.

If we look at the IoUs per cell class in Figure 4.24 for one of the participating teams, it is clear that while the segmentation score for the lymphocytes is lower than for the other classes, using the uncertainty-aware version of the score makes the results even across the classes. This indicates that the poor results for the lymphocytes may be largely due to the fuzziness of the annotations combined with the small size of the objects, rather than to the quality of the predicted segmentation.

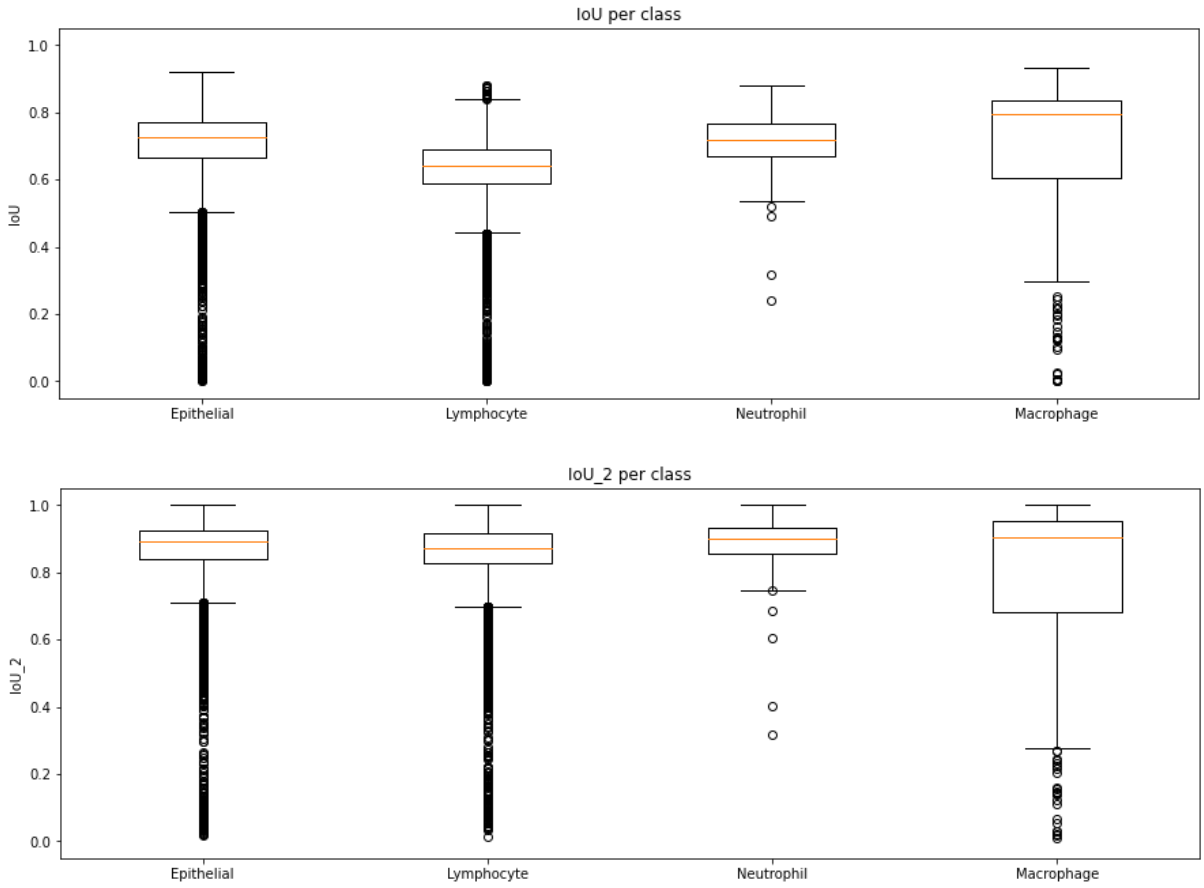



**Figure 4.24. Distribution of the IoU per class for the predictions of "Team 1" on the MoNuSAC test set. (top) Using the simple IoU definition, (bottom) using the uncertainty-aware $IoU_\delta$ with $\delta = 2$.**

#### 4.4.6.2 Over- and under-estimation of objects sizes

Overlap metrics have another bias that should be noted: they do not penalize in the same way over- and under-estimation of the sizes of the objects. This can be easily shown using the formulation of the metrics based on the TP, FP and FN. If, starting from a perfect prediction ($TP = |G|, FP = 0, FN = 0$) we add $n$ "background" pixels to the predicted positives (corresponding to an overestimation of the object size), we have $FP = n, TP = |T|$, and:

$$IoU_+ = \frac{|T|}{|T| + n}$$

Whereas if we remove the same number of pixels from the true positives, we have $TP = |T| - n, FN = n$, and:

$$IoU_- = \frac{|T| - n}{|T| - n + n} = \frac{|T| - n}{|T|}$$

Therefore, for an equal number of erroneous pixels, we have a bias $B$ equal to:

$$B = IoU_+ - IoU_- = \frac{|T|^2 - (|T| - n)(|T| + n)}{|T|(|T| + n)}$$

$$B = \frac{|T|^2 - |T|^2 + n^2}{|T|(|T| + n)} = \frac{n^2}{|T|(|T| + n)} > 0$$

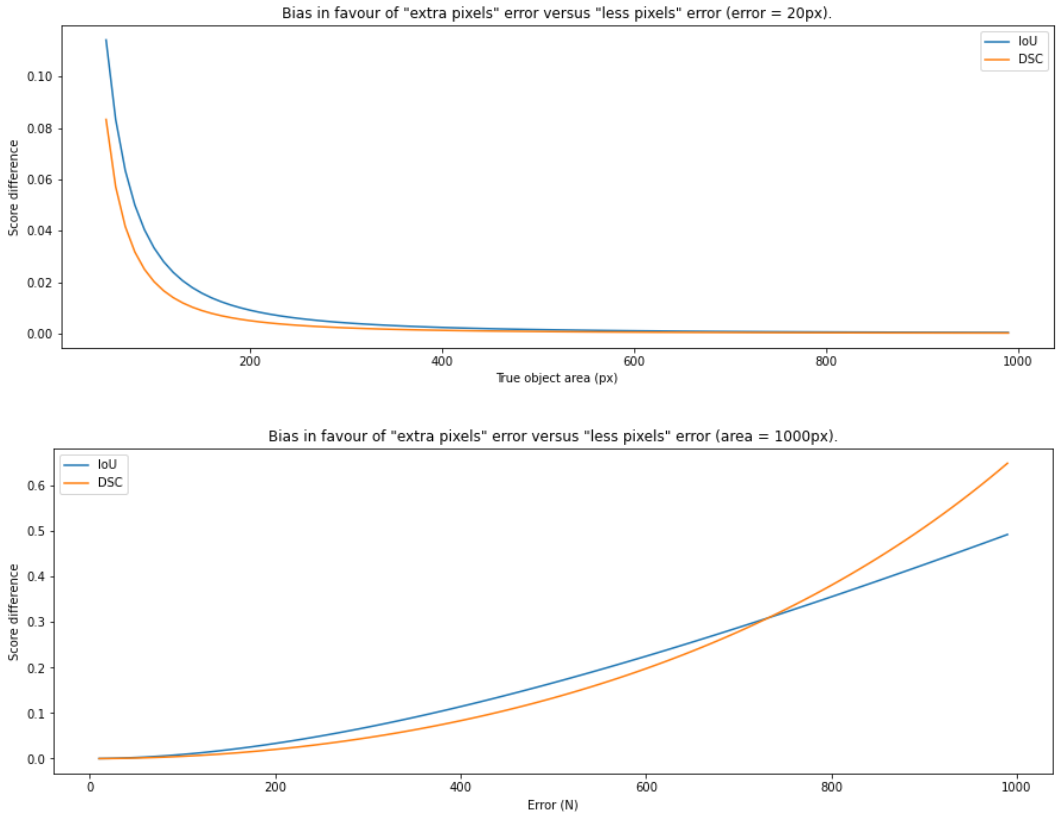



**Figure 4.25. Difference in IoU and DSC computed after (top) adding or removing 20px from a perfectly predicted ground truth object of varying area, (bottom) adding or removing N pixels from an object of 1000px area, showing the bias of the metrics in favour of algorithms that overestimate the size of the object.**

This means that there is a bias in favour of "overestimation" of the object size, which is proportional to the size of the error and inversely proportional to the true size of the object. This effect is clearly visible in Figure 4.25. In the top plot, we can see that for a fixed error of 20px, the bias in favour of overestimation becomes clear as the object area gets smaller. In the bottom plot, we can see that the bias increases as the size of the error increases for a fixed object area.

### 4.4.7 Multi-metrics aggregation: study of the GlaS 2015 results

To aggregate individual metrics into a single final ranking, one solution is to first rank the metrics separately, then to combine them using, for instance, the "sum of ranks". This approach was notably used by the GlaS 2015 challenge for gland segmentation. The main advantage is that the separate ranks provide a more insightful interpretation of the results when comparing different algorithms. It also makes it possible to combine rankings that have a completely different scale, such as the DSC and the HD. An important weakness, however, is that the "final score" of an algorithm (which is the sum of ranks) becomes dependant also on the results of the *other* algorithms being compared, and overall rankings may be undesirably affected by which methods are included in the study. It also does not differentiate between a case where most methods are essentially equivalent according to a metric to a case where some methods are largely better or worse. This is apparent in the table of results from the GlaS 2015 challenge, which we reproduce in Table 4.17. Six separate rankings are computed, on three metrics and two separate test sets. Some of the difference in ranks, however, come from very small differences in metrics which are very unlikely to be significantly different and, for the segmentation metrics, are probably well within the uncertainty on the border position. For instance, for the HD in Part A, ranks 3-6 are within a single pixel distance, and in DSC part B ranks 2-5 are within the same percent. A very slight difference in the annotations could easily change the rankings of #2 and put it in first place or in third.

**Table 4.17. Table of results from the GlaS 2015 challenge** [168]**. Highlighted results in each column show very close scores, which are unlikely to be significantly different.**

| Team | $F_1$ | | $DSC$ | | $HD$ | | Rank sum |
|---|---|---|---|---|---|---|---|
| | Part A | Part B | Part A | Part B | Part A | Part B | |
| #1 | 0.912 (1) | 0.716 (3) | 0.897 (1) | 0.781 (5) | 45.42 (1) | 160.3 (6) | 17 |
| #2 | 0.891 (4) | 0.703 (4) | 0.882 (4) | 0.786 (2) | 57.41 (6) | 145.6 (1) | 21 |
| #3 | 0.896 (2) | 0.719 (2) | 0.886 (2) | 0.765 (6) | 57.35 (5) | 159.9 (5) | 22 |
| #4 | 0.870 (5) | 0.695 (5) | 0.876 (5) | 0.786 (3) | 57.09 (3) | 148.5 (3) | 24 |
| #5 | 0.868 (6) | 0.769 (1) | 0.867 (7) | 0.800 (1) | 74.60 (7) | 153.6 (4) | 26 |
| #6 | 0.892 (3) | 0.686 (6) | 0.884 (3) | 0.754 (7) | 54.79 (2) | 187.4 (8) | 29 |
| #7 | 0.834 (7) | 0.605 (7) | 0.875 (6) | 0.783 (4) | 57.19 (4) | 146.6 (2) | 30 |
| #8 | 0.652 (9) | 0.541 (8) | 0.64 (10) | 0.654 (8) | 155. (10) | 176.2 (7) | 52 |
| #9 | 0.777 (8) | 0.31 (10) | 0.781 (8) | 0.617 (9) | 112.7 (9) | 190.5 (9) | 53 |
| #10 | 0.64 (10) | 0.527 (9) | 0.737 (9) | 0.61 (10) | 107.5 (8) | 210. (10) | 56 |

A possible solution could be to allow for ex aequo rankings when results are within a certain tolerance of each other. A difficulty, however, is that for instance rank #1 and #2 may be within the ex aequo tolerance, and rank #2 and #3 as well, but not rank #1 and #3. To get around that problem, we can use a **statistical score** that replaces the "rank" by a score corresponding, for an algorithm A, to the number of other algorithms that are *significantly* worse (according to an adequate statistical test, as discussed in section 4.3.4) to which we subtract the number of other

algorithms that are *significantly* better. This number can then be summed across the metrics and/or datasets. The resulting statistical score will be higher for algorithms which are often significantly better than their counterparts. This method was introduced in one of our publications [2].

As we do not have the detailed per-image results of the GlaS challenge, we cannot perform these tests, but we can get an idea of what the results can look like using some basic heuristics. Let's assume that any F1-Scores or DSC value pairs that differ by more than 0.05 are "significantly different", as are any HD value pairs that differ by more than 5 pixels. Applied to Table 4.17, this would give us the results presented in Table 4.18. This method provides a relatively easy interpretation of the scores: any method that has a negative score is, on average, significantly worse than most other methods. As this is true regardless of the number of methods considered, it is easier to see method which can safely be discarded as underperforming (in this case, #8-#10). Meanwhile, metrics where the results were extremely close no longer contribute much to the overall standing: according to the DSC, for instance, we can see quickly see that methods #1-#7 are essentially equivalent. There is also a big reward for methods that manage to be significantly better than all or most others in one of the rankings, like #1 on HD Part A, #5 on $F_1$ Part B or #2, #4 and #7 in HD Part B.

The overall ranking based on the score sum shows some small changes, with #2 having the best scoring, followed by #1 and #4, and then #3 and #5. This mostly acknowledges that #3 was "lucky" in the rankings of the challenge, as they tended to be at the "front" of groups of mostly identical results, and therefore may have seen their overall ranking artificially improved. This is, however, just an illustration of how the insights on the challenge results may change when using such a scoring method and should not be considered as an "alternative ranking" of the challenge results, as this would require access to the per-image results of the different teams.

**Table 4.18. Statistical scores estimated from the the GlaS challenge results, assuming some simple heuristics to determine the "significance" of differences in metrics. This should not be considered a true alternative ranking for the GlaS 2015 challenge, as this would require the per-image scores of the different teams to be available for proper statistical testing.**

| Team | $F_1$ | | $DSC$ | | $HD$ | | Score sum |
|---|---|---|---|---|---|---|---|
| | Part A | Part B | Part A | Part B | Part A | Part B | |
| #1 | 4 | 3 | 3 | 3 | 9 | 0 | 22 |
| #2 | 4 | 3 | 3 | 3 | 3 | 7 | 23 |
| #3 | 4 | 4 | 3 | 3 | 3 | 0 | 17 |
| #4 | 3 | 3 | 3 | 3 | 3 | 7 | 22 |
| #5 | 3 | 8 | 3 | 3 | -3 | 3 | 17 |
| #6 | 4 | 3 | 3 | 3 | 3 | -6 | 10 |
| #7 | -1 | -3 | 3 | 3 | 3 | 7 | 12 |
| #8 | -8 | -6 | -9 | -7 | -9 | -3 | -42 |
| #9 | -5 | -9 | -6 | -7 | -7 | -6 | -40 |
| #10 | -8 | -6 | -8 | -7 | -5 | -9 | -41 |

### 4.4.8 Why panoptic quality should be avoided for nuclei instance segmentation and classification

The notion of "Panoptic Segmentation", and its corresponding evaluation metric "Panoptic Quality" (PQ), was introduced by Kirillov et al [230]. Panoptic segmentation, per Kirillov's definition, attempts to unify the concepts of *semantic segmentation* and *instance segmentation* into a single task, and a single evaluation metric. In Panoptic Segmentation tasks, some classes are considered as *stuff* (meaning that they are *regions* of similar semantic value, but with no distinct instance identity, such as "sky" or "grass"), and some as *things* (countable objects). The concept was initially applied to natural scenes using the Cityscapes, ADE20k and Mapillary Vistas datasets. It was then applied to the digital pathology task of nuclei instance segmentation and classification in the paper that introduced the HoVer-Net deep learning model [103].

The PQ was then adopted as the ranked metric of the MoNuSAC 2020 challenge [24], then the CoNIC 2022 challenge [231], and has been adopted by several recent publications [214], [252]–[254]. It combines a "recognition metric" and a "segmentation metric" into a single score. It uses the IoU both for the matching rule, which only recognizes matches if the IoU is strictly above 0.5, and as the segmentation metric. As we previously showed, this can lead to very problematic results when applied to nuclei segmentation, as the target objects tend to be very small. Besides the IoU problem, however, there are other fundamental issues with the metric which we will discuss here:

a) The PQ in used in digital pathology on *instance segmentation and classification* tasks, but these tasks are fundamentally different from the *panoptic segmentation* task that the metric was designed to evaluate.
b) The summarization of the performances of a complex, multi-faceted task into a single entangled metric leads to *poor interpretability of the results*.

#### *4.4.8.1 Panoptic segmentation vs instance segmentation and classification*

In the original definition of a "Panoptic Segmentation" problem in [230], each pixel of an image can be associated with a ground truth class $c$ and to a ground truth instance label $z$. A pixel cannot have more than one class or instance label (i.e. no overlapping labels are allowed), but a pixel does not necessarily have an instance label (i.e. $z$ can be undefined). The distinction between *things* and *stuff* therefore becomes that *stuff* are the classes that do not require instance labels, while *things* are classes that do require them.

In the HoVer-Net publication [103], the PQ is used for *nuclei instance segmentation* only, with all "nuclei" classes grouped into a superclass, and a separate "instance classification" $F_c$ metric which similarly combines detection and classification accuracy (as $F_c = F_{1,d} \times ACC_c$) where $F_{1,d}$ is the detection F1-Score on the nuclei superclass and $ACC_c$ is the classification accuracy computed on all correctly detected instances. In subsequent uses of the PQ from the same group [214], [231] and in the MoNuSAC challenge, however, the multi-class PQ is used for the whole *instance segmentation and classification* task.

The first problem of the PQ in digital pathology is that nuclei **instance segmentation and classification, is *not* a panoptic segmentation task**. Panoptic segmentation is characterized by two key factors:

a) Every pixel is associated with one single *class label*.
b) Every pixel is associated with one single optional *instance label*.

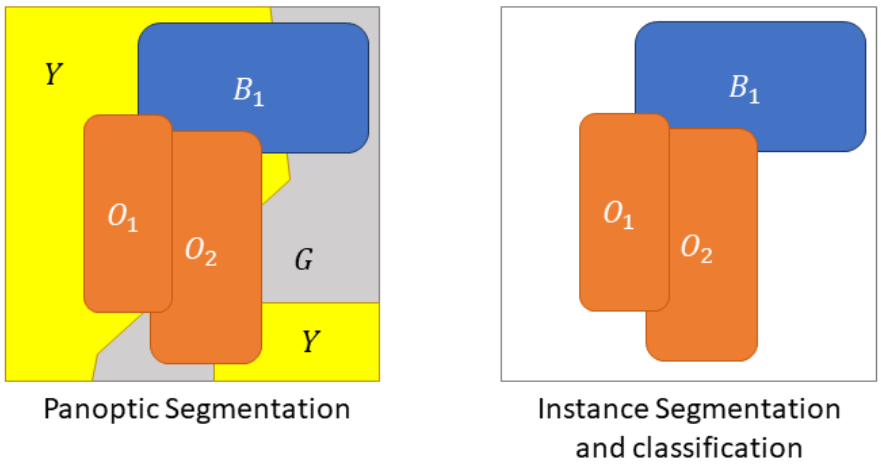



**Figure 4.26. Difference between a Panoptic Segmentation and an Instance Segmentation and Classification task. In the former, every pixel of the image is associated to a class and an optional instance. Some classes ("stuff") always count as a single instance, even if disjointed (Y and G on the left). In the second task, however, it is possible for pixels to have neither class nor instance and be part of the "background".**

In instance segmentation and classification, however, the class label is *also* optional, as there is typically a "background class" that corresponds to everything that is *not* an object of interest (which can be the glass slide itself, or simply tissue areas that are not part of the target classes). Additionally, if a pixel is associated with a class label, it also needs to have an instance label (i.e. there is no *stuff*, only *things*, using Kirillov et al's terminology), as illustrated in Figure 4.26.

This by itself is not necessarily a problem. Metrics can find uses outside of their original, intended scope: the IoU is generally traced to Paul Jaccard's study of the distribution of flora in the Alps [227], long before "image segmentation" was on anyone's radar. There is, however, a problem with the transition between tasks in the present case. The "Recognition Quality" in Kirillov et al's definition corresponds to the *classification* F1-Score, whereas the "Detection Quality" in Graham et al's definition [103] is the *detection* F1-Score. As we discussed before, there is a key difference between the confusion matrices associated with detection and classification problems, with the presence or absence of the background class. The application of the F1-Score to a classification problem leads to the bias that a **higher penalty** is given to a ***good detection* with the *wrong class*** (which will be counted as a "false negative" in the ground truth class, and as a "false positive" in the predicted class) **than to a *missed detection*** (which will only be a "false negative" in the ground truth class), as previously demonstrated in section 4.4.4, and this bias is therefore transmitted to the PQ.

Figure 4.27 illustrates this problem. In this example from the MoNuSAC results, Team 3 is the only one to detect the nucleus of a macrophage. However, they misclassify it as an epithelial cell. Team 3's detection is therefore counted both as a false positive for the epithelial class, and as a false negative for the macrophage class. In contrast, the three other teams that did not detect it are only penalized for the macrophage false negative. In a real panoptic segmentation problem, this could not happen, as there is no "background" class: any false negative is *always* a false positive of another class, as there must be some predicted object or region at that particular location.

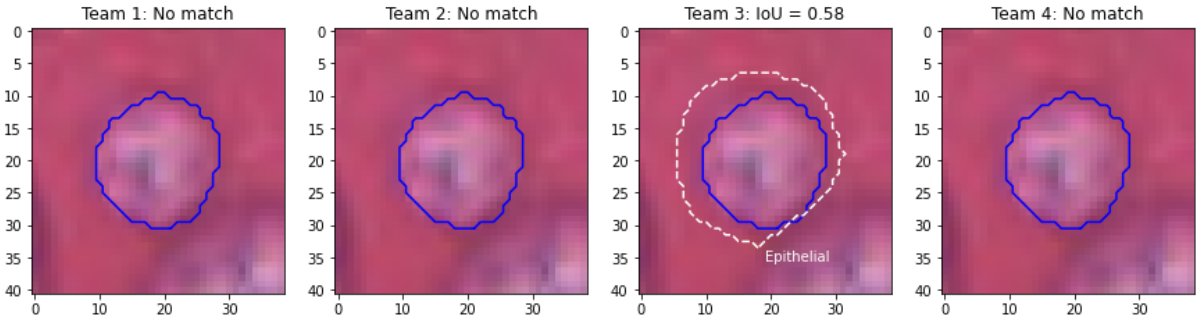



**Figure 4.27. Predictions of the four teams (dashed line) on the nuclei of a macrophage, with the corresponding IoU compared to the ground truth segmentation (solid blue).**

#### *4.4.8.2 Intersection over Union for digital pathology objects*

The reliance of the PQ on the IoU impacts it at two levels: the **matching rule** and the **segmentation quality**. For the matching rule, it means that the conjunction of a small object and an algorithm that underestimate its size can easily lead to "false false detections", where clearly matching objects are rejected due to an IoU under 0.5. For the segmentation quality, the problem lies with the interpretability, and with the aggregation. When objects of different classes have different sizes, the limit of what would constitute a "good" IoU within that class are different. When the PQ are averaged between the classes, this therefore adds a hidden "weight" to the metric. Indeed, algorithms that perform poorly on classes with smaller objects will necessarily tend to have a lower average IoU (and therefore PQ) than those that perform poorly on classes with larger objects.

Additionally, it is well known that **the IoU does not consider the shape of the object** (like other overlap-based metrics such as the DSC). As demonstrated in [226] and shown in Figure 4.23, predictions that completely miss the shape of the object can end up with the same IoU as predictions that match the shape well, but are slightly offset, or slightly under- or overestimate its size. To get a better sense of the segmentation performance of an algorithm, it is often useful to refer both to an overlap-based metric like the IoU and to a border distance metric such as Hausdorff's Distance (HD). By using the PQ, an important aspect of the evaluation is therefore completely missed. In digital pathology tasks, the *shape* of the object of interest is often very relevant to the clinical and research applications behind the image analysis task. It is therefore ill-advised to base a choice of algorithm on a metric that ignores that particular aspect.

#### *4.4.8.3 Interpretability of the results*

As we have shown in [7], the PQ **metric hides a lot of potentially insightful information** about the performances of the algorithms by merging together information of a very different nature. While the SQ and RQ have the same range of possible values, being bounded between 0 and 1, the implication of multiplying these values to get the PQ is that the impact of a change in SQ by a factor $\alpha$ is exactly the same as a change of RQ by the same factor.

The significance of these changes for the underlying clinical applications, however, can be very different. A 10% reduction in the SQ may only indicate a small underestimation of each segmented object's size (which, for small objects, would probably be within the typical interobserver variability), whereas a 10% reduction in the DQ indicates potentially much more significant errors, with entire objects being added as false positives, or missed as false negatives. The interpretation of the relative change in SQ is dependent on the size of the ground truth object, while the interpretation of the relative change in RQ is more likely to depend on the class

distribution. Ranking different algorithms with the PQ therefore leads to results that cannot really be related to the needs of clinical applications.

## 4.5 Recommendations for the evaluation digital pathology image analysis tasks

In this chapter, several important aspects of evaluation processes have been highlighted. An attempt will now be made to extract some practical recommendations on how to properly assess the performances of algorithms in a digital pathology context. It should be noted here that, at this stage, we do not yet include in these recommendations the aspects related to interobserver variability, imperfect annotations, and quality control, as those will be considered in the next chapters. Instead, we consider here that we have access to a dataset with annotations that are treated as a mostly correct "ground truth", outside of a small uncertainty on the boundaries of the objects.

These recommendations follow five axes: the choice of metric(s) depending on the task, the benefits of using simulations to provide context for the results, the importance of using disentangled metrics to independently assess the individual "sub-tasks", the necessity of proper statistical testing, and the benefits of going beyond the "ranking" paradigm. Other important recommendations for biomedical image analysis challenge organizers can be found in the BIAS guidelines by Maier-Hein et al. [255], and some recommendations for digital pathology evaluation have also been proposed by Javed et al. [256].

### 4.5.1 Choice of metric(s)

While Figure 4.15 focused on the choice of classification metrics depending on the class imbalance, the expended flowchart in Figure 4.28 attempts to summarize the factors that impact the choice of metric(s) depending on the type of task and the characteristics of the dataset. Such a flowchart is bound to be incomplete, but it highlights some of the main questions that need to be asked when deciding how to evaluate algorithms.

The first of those questions relate to the nature of the target: are there image-level (patch or WSI/patient) labels, regions of interest, or distinct objects to be found? Are there one or more 'positive' or 'target' class(es) compared to a 'background', 'others' or 'negative' class, or are all categories considered equal? Are the categories ordered? Is there some "hierarchy" in the classes, so that the target classes can be grouped into one or more "superclass"?

As we move further down the flowchart, questions also relate to what information or insight we are trying to gain from the experiment, as well as the characteristics of the dataset: do we want to penalize larger errors with ordered categories? Do we care about the quality of the segmentation? Is the dataset balanced?

It should be noted that it is possible to end up with several answers when dealing with complex tasks, which illustrates the importance of computing multiple metrics to evaluate all the important aspects of the task. For example, in an instance segmentation and classification problem, following the flowchart from the "objects" target, we could conclude that we need to compute the binary detection $F_1$ or $AP$ on a "superclass", then use the object confusion matrix to compute an Accuracy or MCC, and measure the segmentation quality using the per-class IoU, DSC or HD.

The final choice for a specific metric within the same "endpoint" of the flowchart would further depend on the precise characteristics of the dataset and of the real-world application that the

dataset attempts to represent. For instance, as we discussed before, very small objects of interest would make the use of overlap-based segmentation metrics inadvisable.

To help in that final choice, and to help contextualize the results, simulations can be used.

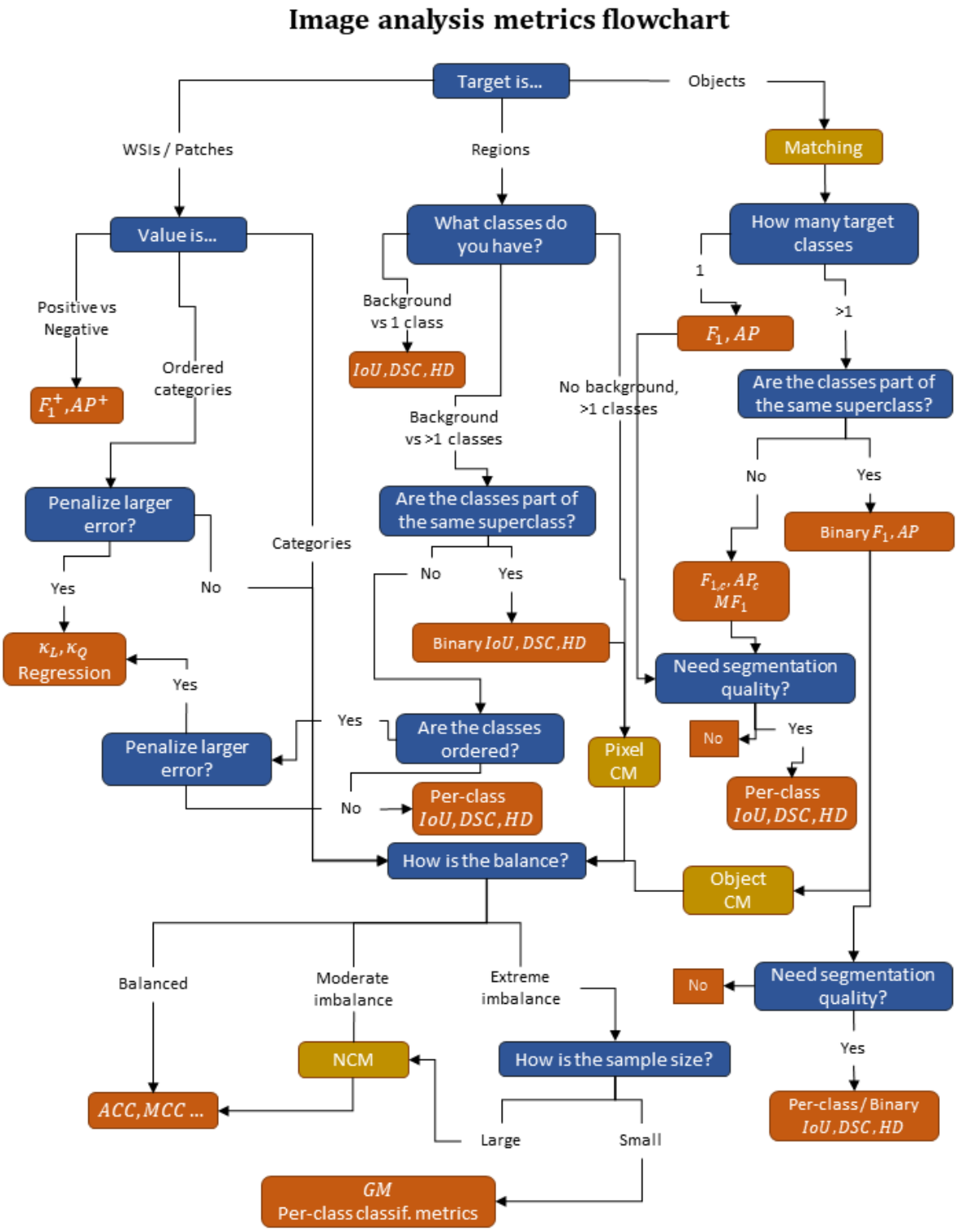



**Figure 4.28. Flowchart for aiding in the choice of evaluation metrics based on the characteristics of the task. Questions are in blue, metrics in red, and partial steps in orange.**

To illustrate the use of the flowchart, we can look at the task of nuclei instance segmentation and classification. At the top of the chart, we first ask the nature of the target: in this case, objects (nuclei). The chart thus indicates a first point of attention in the evaluation metric: the *matching criterion*. The next question then relates to the number of classes, which will be superior to one (as we want to classify different types of cell nuclei), but the classes belong to the same superclass (nuclei). The chart thus recommends using *binary detection metrics* (such as $F_1$ and $AP$) on the "superclass versus background" detection problem. Two paths can now be further followed. The first one recommends computing the *confusion matrix* for the classes of the detected objects and then, depending on the (im)balance of the dataset, to compute *classification metrics* either on the normalized or on the original confusion matrix (or, in extremely imbalanced cases, to stick to per-class metrics such as SPE…). The second path asks if we need to evaluate the segmentation quality, which we do. It therefore recommends the use of per-class and/or binary segmentation metrics. Which one to use will depend on the exact characteristics of a given dataset.

### 4.5.2 Using simulations to provide context for the results

Most evaluation metrics appear to be easily interpretable at a surface level. Their range is often strictly bounded, and very often presented as a "percentage" (outside of distance-based segmentation metrics). This can give the illusion of metrics providing an absolute scale, with "50%" neatly splitting "failure" from (limited) success, and results above "90%" or "95%" being perceived as very good.

As shown in this chapter, however, evaluation metrics are highly susceptible to differences in the characteristics of the dataset. Even within the same well-defined task, the performance of an algorithm may vary considerably depending on the composition of the test set. Several simulations have been used in this chapter to illustrate some of those differences. Such simulations can be very useful in general to provide context to the results and to help in the choice of metrics and correct interpretation of their values.

Two main types of simulations have been used here: alterations to the annotations, and simulations of performances. In the first type, small changes are done to the annotations to determine how quickly the evaluated performance deteriorates given different types of errors. If done on different metrics, this can help choose which metrics penalize the types of errors that are more relevant to the real-world application under consideration. In the second type of simulation, the performance of a "fake" algorithm is probabilistically pre-determined (with, for instance, distributions of per-class sensitivities), and results are randomly computed based on the known characteristics of the dataset (such as the class imbalance). Multiple simulated runs can provide an indication on the effects of random sampling, and the range of results on the metric being considered that would correspond to that performance level.

### 4.5.3 Disentangled metrics

It is clear from our experiments and analyses that metrics that attempt to capture multiple aspects of a task in a single score should be avoided. While they may provide an easy way of ranking algorithms, they do so at the cost of interpretability, and produce ranking that cannot reliably be related to the real-world applicability of the results.

If a single ranking for a complex task is desirable, then subtasks should first be ranked independently, and then combined. Ideally, these ranking should take into account the uncertainty of the metric, so that close results can be considered "ex aequo".

The uncertainty due to the random sampling of test set examples can for instance be captured through adequate statistical tests.

### 4.5.4 Statistical testing

Deep learning algorithms are full of randomness, from the initialization process to the training itself. Coupled with the randomness inherent in the selection of data samples for the dataset (and in the split between training and test set), this makes it difficult to draw definitive conclusions about small differences in results between algorithms.

Statistical testing provides a way of bringing some objectivity to this interpretation of the difference in results. To compare multiple algorithms based on an arbitrary metric, the Friedman test with a post hoc such as the Nemenyi test are recommended, with the "samples" being either the patients or the image patches (if they are reasonably similar in sizes, to avoid giving too much importance to small errors in small images).

### 4.5.5 Alternatives to ranking

When evaluating sets of algorithms on a digital pathology task, it is important not to get focused so much on the ranking that we forget the point of the experiment. In most deep learning experiments on digital pathology today, the goal is not yet to validate a product for clinical use, or to find the absolute best trained model to solve a particular task. Rather, the goal is to gather knowledge and insights on the algorithms themselves, to guide future research that may eventually lead us towards methods that bring us closer to clinical applicability.

Rankings can certainly help with providing some insights on the performances of the algorithms, but they are neither sufficient nor necessary in this regard. Alternatives to ranking should are therefore recommended to better capture the behaviour of the methods being compared. Such alternatives include using similarity metrics not just between the methods and the ground truth (particularly when this ground truth, as is often the case in digital pathology, is imperfect and uncertain), but also between the methods themselves. Representations of these similarities can be made using methods such as Multi-Dimensional Scaling or dendrograms, so that the relations between the choices in hyper-parameters and the behaviour of the methods can be better understood.

Clusters of similarly behaved methods can then be found, which can provide additional insights as to which hyper-parameters and which elements of the deep learning pipeline have a larger effect on the performances of an algorithm, as well as what that effect is.

## 4.6 Conclusion

To conclude this chapter, it is important to note once again that this analysis of the evaluation metrics and processes considered that the ground truth of the task is known and considered as reliable. As we will see in the following chapters, this assumption can be questioned for digital pathology datasets (as for other fields based on medical imaging). To the uncertainties related to random sampling and the inherent biases of the metrics should therefore be added the uncertainties related to the unreliability of the data.

As we move further into the specificities of digital pathology datasets and their effect on the training and evaluation of deep learning algorithms, however, we should always keep in mind the lessons that we have already drawn here.

# 5 Deep learning with imperfect annotations

As noted in the 2017 survey of deep learning in medical imaging by Litjens et al. [163], "the main challenge [in medical image analysis] is not the availability of image data itself, but the acquisition of relevant annotations/labelling for these images." With the increasing availability of whole slide scanners, large datasets of pathology images can now be assembled, but their annotations require highly trained experts and a lot of time [3], [257].

This is particularly the case for segmentation problems, where the "ground truth" is determined at the pixel level. Pixel precise annotations, however, would require a prohibitively large time commitment from the experts, and in many cases would in any case be made impossible by the limitations of the resolution of the image. The exact position of the boundary between regions is often fuzzy and ill-determined in digital pathology datasets, even ignoring the time constraints. As objects of interest can be very numerous and very small, annotations may often not be done at the highest possible available resolution, which means that annotations which seem very precise at the level at which they were made be a lot less accurate at a higher level of magnification (as illustrated in Figure 5.1).

The uncertainty on the accuracy of the annotations can have several causes. As we have already established, there is the inherent uncertainty caused by the acquisition process, with the fuzzy boundaries of the target objects. Another source may be the annotation software or hardware used by the expert. Annotations made with a mouse on a standard computer monitor may be less accurate than annotations made with a stylus on a high-end drawing tablet. The expert may also make mistakes, either in the interpretation of the image or in the manipulation of the annotation tools. A different type of annotation uncertainty comes from the absence of a unique "ground truth": independent expert annotators, faced with the same image and the same instructions, may produce different annotations.

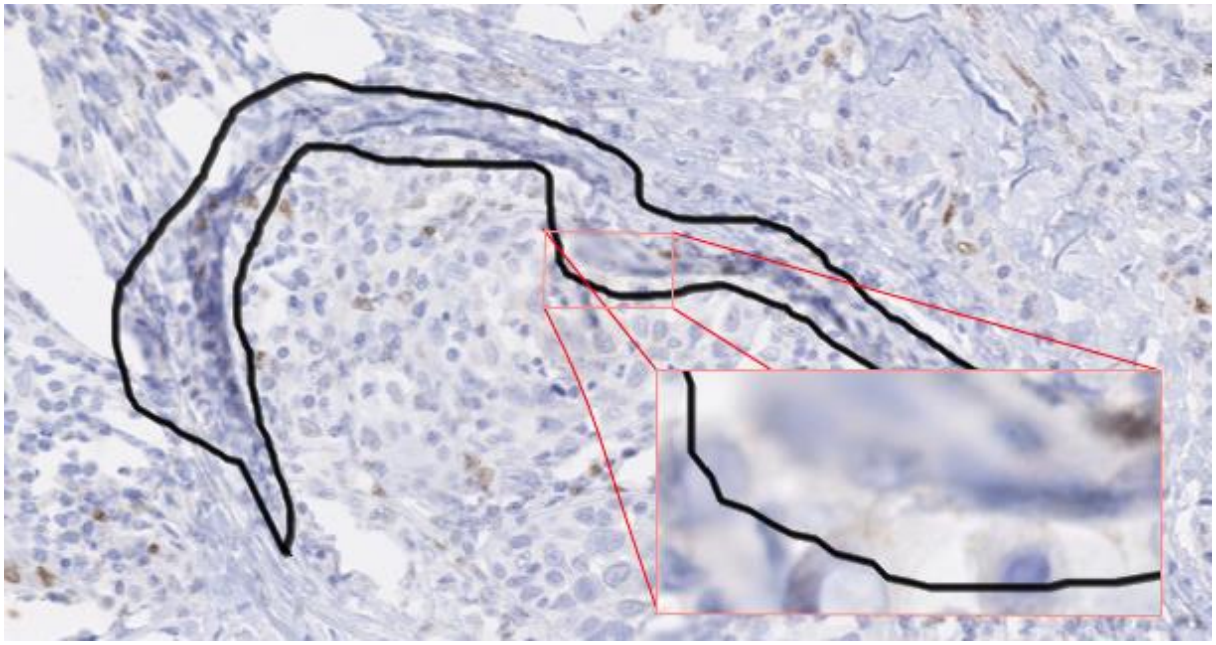


**Figure 5.1. Expert annotation of an artefact in a WSI, shown at 10x magnification with detail at 40x magnification.**

In this chapter, we will focus on the first type: uncertainties that are caused by *imperfections* in the annotations. In the next chapter, we will consider uncertainties that come from *interobserver variability*, or the absence of a singular "ground truth".

First, we will characterize imperfect annotations using well-known paradigms in machine learning: semi-supervised learning, learning with noisy label, and weakly supervised learning. This will lead us to the concept of "SNOW" supervision (Semi-supervised, NOisy, and/or Weak), that we introduced in [2] and developed in [4]. We will review the state of the art on dealing with the different types of imperfections. We will then present our experiments on the effect of SNOW on segmentation tasks, and on which learning strategies are useful to counteract these effects. Finally, we will examine how our knowledge of these imperfections should affect our evaluation process.

## 5.1 Imperfect annotations

In the previous chapter, we have defined datasets in the best-case scenario, where each sample is associated to a corresponding ground truth annotation. To describe the different types of imperfections, we need to introduce some notations for the rest of this chapter. $X = \{x_i\}, i \in [1, N]$ is the set of *images*, and $x_i = \{x_{ij}\}, j \in [1, M_i]$ is the set of pixel values in the image $x_i$. $Y = \{Y_i\}$ is the set of available *annotations*, and $T = \{t_i\}$ is the set of *true target values* that would exist in an ideal dataset.

We define three main types of imperfections, illustrated in Figure 5.2:

a) **Incomplete** annotations

In incomplete annotations, there is a subset of $X$ for which there is no corresponding annotation. This can happen at the pixel level (i.e. within an image $x_i$, there are some $x_{ij}$ with a corresponding $y_{ij}$, and some without) or at the image level. Using classical machine learning paradigms, this corresponds to *semi-supervised learning*. In digital pathology segmentation or detection tasks, it is for instance very common to have expert annotations only for selected focal regions in a WSI. The majority of the WSI, however, may be left unannotated. This can also apply to higher-level tasks such as image classification, where some images may not have expert labels associated to them.

b) **Imprecise** annotations

This corresponds to annotations where the *level of details* of the annotations is less than that of the target ground truth. Using our notation, this would mean that labels $y_{ik}$ are assigned only for sets of $t_{ij}$. Practically, this could mean in segmentation tasks that only bounding boxes, or in extreme cases only image-level labels were provided on some or all of the examples, even though the target of the algorithm is to provide a segmentation. This type of imperfection is generally addressed through *weakly supervised learning*.

c) **Noisy** annotations

With noisy annotations, even when each target ground truth has a corresponding annotation, this annotation may be incorrect. In other words, for each $t_{ij}$, it is possible that $y_{ij} \neq t_{ij}$. The likelihood of such errors determines the level of noise of the dataset. This type of imperfections can take different forms depending on the type of task. In segmentation or detection tasks, this could

correspond to objects that are missed by the annotator. In classification tasks, this could correspond to a wrong label being assigned to the object or image.

Several recent studies have been published in recent years looking at different aspects of imperfect annotations in medical imaging. Karimi et al. [85] experimented on three types of label noise: systematic error by a human annotator, interobserver variability, and label noise generated by an algorithm. Tajbakhsh et al. [258] categorized imperfections into "scarce annotations" (similar to our "incomplete" definition) and "weak annotations" (which include both imprecise and inaccurate annotations). Zhang et al. [259] surveyed techniques for "small sample learning", which include aspects of incomplete and imprecise annotations. Vădineanu et al. [260] studied annotation errors in cell segmentation, with both inaccurate and imprecise annotations being considered. While this topic has only recently started to be more thoroughly investigated for deep learning and digital pathology, it has been largely covered in classical machine learning through the notions of semi-supervised learning, weakly supervised learning, and noisy labels.

### 5.1.1 Semi-supervised learning from incomplete annotations

In semi-supervised learning, the dataset can be divided in two subsets: the *labelled* set $L = \{(x_i, y_i)\}, i \in [1, n_L]$, and the unlabelled set $U = \{(x_i, \emptyset)\}, i \in [n_L + 1, n_L + 1 + n_U]$ where $n_L$ and $n_U$ are the number of labelled and unlabelled instances in the dataset, respectively. It will generally be the case that $n_U \gg n_L$, as obtaining unlabelled instances is much easier than obtaining annotations.

Zhu and Goldberg [261] describe two main approaches to semi-supervised learning. The first approach is to start from a standard supervised algorithm, and to use the unlabelled data to strengthen the classification from the few labelled data. The second is to start from an unsupervised algorithm, such as clustering, and to use the labelled data to obtain a better clustering (and to label the clusters). The key assumptions made in both approaches are *local consistency* (i.e. similar samples share the same label) and its counterpart *exotic inconsistency* (i.e. samples with low similarity are likely to come from different classes) [262]. As noted in Su et al. [263], these assumptions may not always hold for histopathological images, which can have high inter-class similarity and intra-class variation.

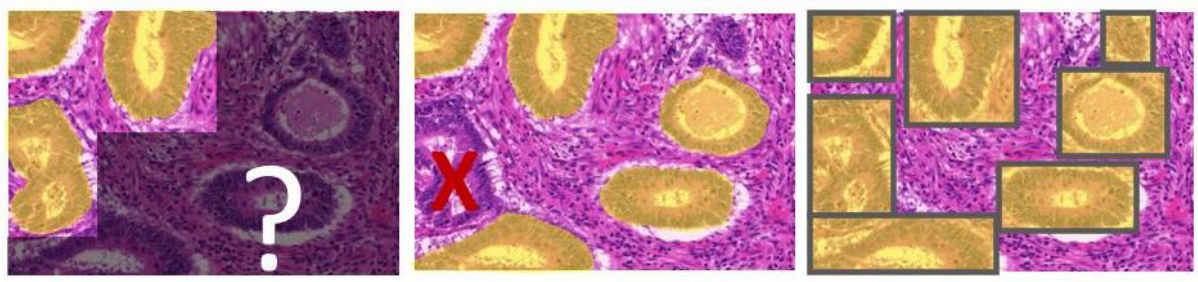



**Figure 5.2. Illustration of the three types of imperfection, using as an example an image from the GlaS challenge** [168]**. From left to right: incomplete (semi-supervised), noisy and imprecise (weak).**

An example from classical machine learning applied to digital pathology can be found for instance in Peikari et al. [264]. A "seeded clustering" method is first used to propagate the labels from $L$ to $U$ and to estimate the density function of the feature space. This density is then used to train a SVM classifier so that it finds the maximum margin boundary region that passes through sparse regions of the feature space.

These ideas have also been applied to deep learning in digital pathology in recent years. Chen et al. [265] use a DCNN to learn features based on sparse labels, then propagate these labels based on affinity in feature space. They apply this method to gland segmentation and tumour region segmentation with promising results based on very few annotations. Shaw et al. [266] propose a "teacher-student chain" based around a ResNet-50 architecture, where $L$ is first used to train a teacher model, which predicts "pseudolabels" on $U$. The pseudolabels with a high level of certainty (according to a set threshold) are kept forming a new "supervised" set $P$. $P$ is then used to train a "student" model, which is then fine-tuned on $L$. The student then becomes the teacher, and the cycle is repeated. Their method is tested on the **NCT-CRC-HE-100K multi-class patch classification dataset** [267]**. A similar approach is used by Jaiswal et al.** [268] **for lymph node metastases patch classification on the PatchCamelyon dataset**[41]**.**

### 5.1.2 Weakly supervised learning from imprecise annotations

The typical framework for weakly supervised learning (WSL) is Multiple-Instance Learning (MIL). In MIL, instances $x_i$ of the data are grouped into bags $b_j = \{x_i\}$, so that labels are only assigned to the bags. Instead of the fully supervised dataset $D = \{(x_i, y_i)\}$, we therefore have a weakly supervised dataset $W = \{(b_j, y_j)\}, b_j = \{x_i\}$. The most common application of this framework to visual recognition is to have image-level labels for an object-level task like detection, or a pixel-level task like segmentation. As introduced by Dietterich et al. [269], MIL describes binary problems with a "positive" and a "negative" class, and assumes that any bag that contain at least one positive instance is "positive".

T. Durand, in his Ph.D. thesis [270] identifies a way of implementing MIL in a deep learning framework. The idea is to use global spatial pooling (for example, max pooling) from a layer that still contains spatial information. With max pooling, the image-level score is therefore the score of the region with the maximum predicted value, which is in line with the principle of MIL that one positive "instance" (in this case: region) is enough to classify the whole bag (in this case: image) as positive. The localization of the detected object can therefore be inferred from the activation of the feature maps. A similar idea is used by methods such as Grad-CAM [271], which uses the activation of the feature maps to highlight the regions of the image that contribute to a class prediction. The main challenges that face WSL algorithms are the difficulty of differentiating strongly co-occurring objects (for instance: *boat* and *water*, or *chair* and *table*), and of identifying the whole object instead of its most discriminative part (for instance: the face instead of the whole *person*) [272].

An early application of MIL in digital pathology was proposed by Xu et al. in 2014 [273], in which image-level binary labels (cancer vs non-cancer) are used to obtain pixel-level segmentation and patch-level clustering of distinct cancer types. A deep learning adaptation of MIL was proposed in 2017 by Jia et al. [274], where a DCNN is used to produce pixel-level segmentation at different scales, from which the image-level prediction is determined. The loss function can therefore be computed based on the image-level labels, but the network has to produce a pixel-level prediction

[41] https://github.com/basveeling/pcam

as well. Additional area constraints are placed on the segmented regions to penalize unlikely predictions. Chen et al. [265], which we already mentioned for the label propagation in semi-supervised learning, also proposes a WSL method by using superpixels, computed using the SLIC algorithm [275], to propagate points annotations to a larger region.

### 5.1.3 Learning from noisy labels

Label noise means that, for any annotated label $y_i$, there is a certain probability that the corresponding ground truth $t_i$ is different. Frénay and Verleysen proposed a taxonomy of label noise [276] that describes three models of noise:

a) Noisy completely at random: the occurrence of an error is completely independent from either the true class or the observed data: $p(y_i \neq t_i | x_i, y_i) = p_e$ is constant $\forall i$. In such models, it is furthermore assumed that in a multiclass problem erroneous labels are also uniformly distributed between the other classes.
b) Noisy at random: the occurrence of an error depends on the true class but not on the observed data. This means that the noise distribution can be described using a *noise transition matrix*: $N_{ij} = P(y_k = i \mid t_k = j)$.
c) Noisy not at random: the occurrence of an error depends on the true class and on the observed data. This model recognizes that, within a class, there will be examples that will be more likely to be mistakenly labelled (for instance, because they are more similar to examples from other classes, or because they are dissimilar to most other examples, meaning that they are in a more sparsely populated region of the input space).

Typical methods outside of deep learning to deal with label noise include data cleaning (i.e. excluding samples that are more likely to be wrong based on, for instance, outliers detection), adapted loss functions, or soft labels reflecting the uncertainty [277]. Frénay and Verleysen categorize three types of methods for handling noisy labels [276]: those that adapt the *model selection and design* (including loss function and training procedures), those that reduce the noise in the training data, and those that train the classifier and model the noise concurrently.

Karimi et al. [85] review deep learning methods that include some form of noisy label management. The main categories that they identify are: label cleaning and pre-processing; adapted network architectures that include specialized layers that reduce the influence of noisy labels [278], adapted loss functions such as the IMAE loss [279]; data re-weighting that down-weight samples that are more likely to have incorrect labels; adding constraints on data consistency (i.e. samples of the same class should be close in feature space); and adapted training procedures such as co-teaching [280]. In their own experiment on MRI data, Karimi et al. [85] obtain good results with a simple data re-weighting method where data samples with high loss values are ignored, and with a more involved method dubbed “iterative label cleaning”, where a model for detecting noisy labels is updated alongside the main classifier. Using adapted loss functions, meanwhile, did not provide any improvement compared to their baseline network.

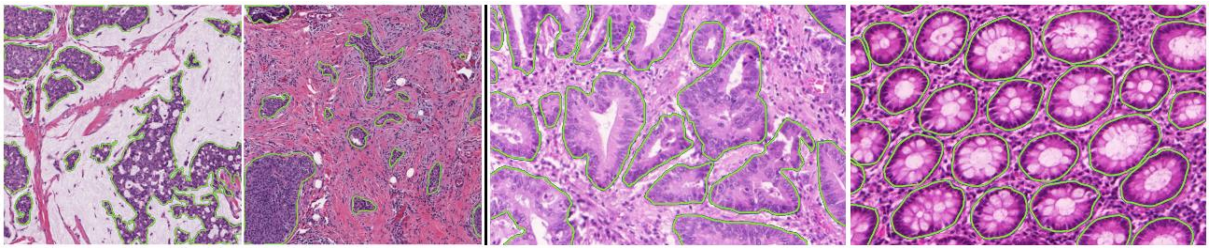


**Figure 5.3. Annotated images from the Epithelium (left) and GlaS (right) datasets.**

## 5.2 Datasets and network architectures

### 5.2.1 Datasets

The two datasets used in our experiments are the GlaS challenge dataset [168] and the Epithelium dataset from Janowczyk and Madabhushi [148]. Both contain high quality segmentation annotations of all the target objects in the images. The GlaS dataset targets glands in colorectal tissue (from normal and tumour regions), while the Epithelium dataset separates epithelial from stroma regions in breast cancer tissue. Examples of images and annotations from these datasets are shown in Figure 5.3.

The GlaS dataset has a very high density of foreground objects, with 50% of the pixels in the training set being annotated as positive. The Epithelium dataset, meanwhile, has a slightly lower density with 33% positive pixels in the training set. Our baseline networks are trained using small patches extracted from the larger images of the datasets. For the GlaS dataset, we used 256x256px patches, and around 95% of the extracted patches contain at least some part of a gland, and would therefore be considered "positive" patches in a MIL framework. For the Epithelium dataset, 128x128px patches were used, with around 87% "positive" patches.

The GlaS dataset is used for the experiments on the effects of SNOW, while both datasets are used for the experiments on the learning strategies. Additional information on the datasets can be found in Annex A.

### 5.2.2 Baseline networks

Three networks were used for our experiments: ShortRes, U-Net and PAN. They all follow a classic segmentation macro-architecture with an encoder and a decoder. The "ShortRes" network uses short-skip connections similar to ResNet [106] in the encoder and in the decoder, and has around 500k parameters. U-Net is directly adapted from Ronneberger et al.'s version [66], with long-skip connections between the encoder and the decoder and Dropout layers. It has around 30M parameters. The third network, which we called "Perfectly Adequate Network" (PAN), uses both short and long-skip connections, and combines the outputs from different layers to produce the final segmentation using multi-scale predictions. It has around 10M parameters. These 3 architectures allow us to study networks with a priori different learning capacities and to measure their respective resistance to different types and/or levels of supervision defects. In all the networks, the Leaky ReLU [281] activate function are used in the convolutions and transposed convolutions. A schematic representation of the ShortRes and PAN architectures is shown in Figure 5.4.

In all our experiments, the same basic data augmentation scheme is applied to the training sets that are then used by all networks, left as is (baseline) or combined with one of the learning

strategies described below. We modify each mini-batch on-the-fly before presenting it to the network, using the following methods:

- Random horizontal and/or vertical flip.
- Random uniform noise on each of the three RGB channels (maximum value is 10% of maximum image intensity).
- Random global illumination changes on each of the three RGB channels (maximum value of ± 5% of maximum image intensity).

## 5.3 Experiments on the effects of SNOW supervision

To assess the effects of SNOW supervision on deep neural networks, we introduced "corrupted annotations" to clean datasets. These corrupted annotations were generated to simulate different levels of supervision and annotation effort. In this section, we will detail the methodology for the corruption, and show the effect on the three baseline networks. In the next section, we will look at how different learning strategies can help to deal with the effects of SNOW supervision. The results presented here were originally reported in our SNOW publications [2], [4], with some additional assessment.

### 5.3.1 Corruption methodology

As illustrated in Figure 5.5, random corruptions are introduced in the training set annotations of both datasets mimicking imperfections commonly encountered in real-world datasets. The test set with correct annotations is kept to evaluate the impact of these imperfections on DL performance. Pseudo-code for the procedure used to generate the corrupted annotations is given in Table 5.1.

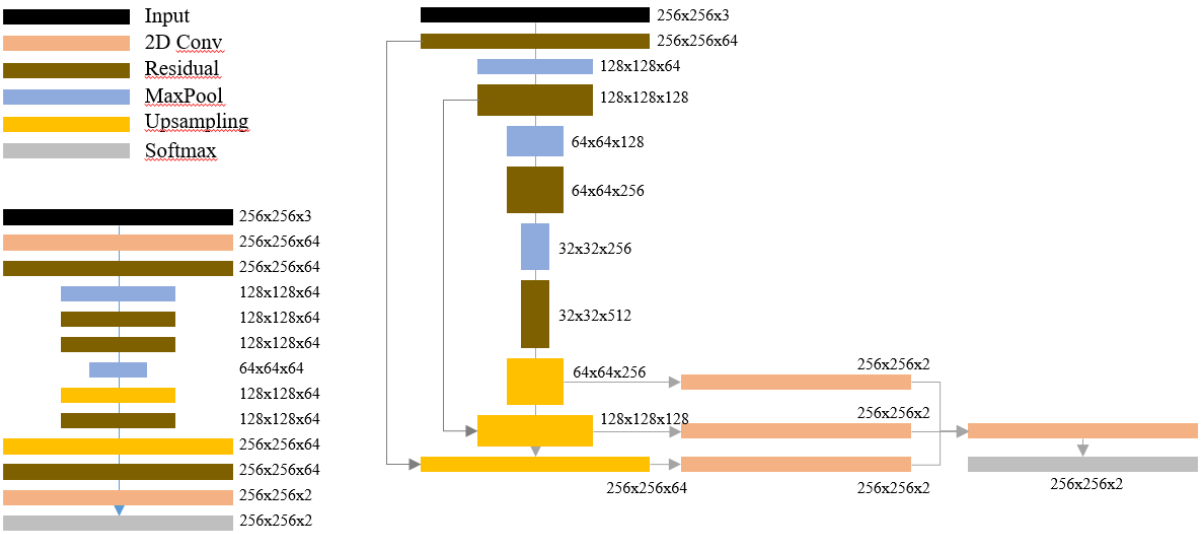



**Figure 5.4. Architectures of the ShortRes (left) and PAN (right) networks. Upsampling layers are Transposed Convolutions. Residual units contain three convolutional layers with a short-skip connection that adds the input $x$ to their output, so that $R(x) = C_3\left(C_2(C_1(x))\right) + x$.**

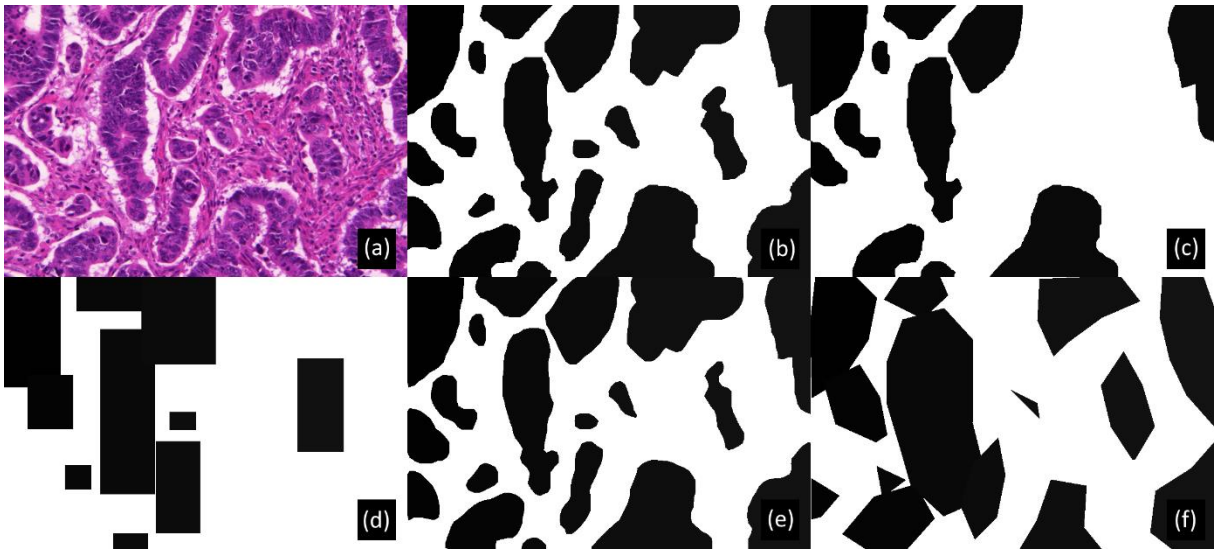



**Figure 5.5. Examples of corrupted annotations generated on the GlaS dataset to simulate different levels of supervision and annotation effort. (a) Original image, (b) Original annotations, (c) 50% Noise (i.e., 50% of the objects of interest are labelled as background), (d) 50% noise + Bounding Boxes, (e) Low contour deformations, (f) High contour deformation.**

**Table 5.1. Pseudo-code for the corruption procedure to create fabricated datasets. For "bounding boxes", lines 9-11 are changed so that the object is replaced by its bounding box rather than using the simplification factor.**

| | **Corruption of the annotations** |
|---|---|
| | Let $v$ be the **noise** level. |
| | Let $\sigma_R$ be the standard deviation of the radius of the **erosion/dilation** disk. |
| | Let $f$ be the **simplification factor** for the contour. |
| | Let $D = \{(x_i, y_i)\}$ be the original dataset, with images $x_i$ and corresponding annotations $y_i$ |
| | **Output**: $D^* = \{(x_i, y_i^*)\}$, the corrupted dataset |
| 1 | **For each** $(x_i, y_i) \in D$: |
| 2 | **For each** annotated object $a_j$ in $y_i$ and corresponding object $a_j^*$ in $y_i^*$: |
| 3 | **Draw** $r$ from uniform distribution $U(0,1)$ |
| 4 | **If** $r \leq v$: remove $a_j^*$ from $y_i^*$ |
| 5 | **Else:** |
| 6 | **Draw** $s$ from normal distribution $N(0, \sigma_R)$ |
| 7 | **If** $s < 0 \rightarrow a_j^* = erode(a_j, disk(\lvert s \rvert))$ |
| 8 | **Else:** $a_j^* = dilate(a_j, disk(s))$ |
| 9 | **Compute** all points in contour of $a_j^* \rightarrow C$ |
| 10 | **Sample** $\frac{\lvert C \rvert}{f}$ points in $C$ (regular sampling) to create contour $C^*$ |
| 11 | **Compute** filled polygon raster of $a_j^*$ from $C^*$ |

Because creating pixel-perfect annotations is very time-consuming, experts may choose to annotate faster by drawing simplified outlines. They may also have a tendency to follow "inner contours" (underestimating the area of the object) or "outer contours" (overestimating the area). We generate deformed dataset annotations in a two-step process. First, the annotated objects are

eroded or dilated by a disk whose radius is randomly drawn from a normal zero-centered distribution, a negative radius being interpreted as erosion and a positive radius as dilation. The standard deviation ($\sigma_R$) of this distribution enables us to adjust the level of deformation. The second step consists in simplifying the contour of each object, as follows. The contour pixels are identified and only a fraction of them, determined by a simplification factor $f$, are kept to create a polygonal approximation of the original contour. The median number of points in the object contours in the original annotations is 358. We introduce low deformations using $\sigma_R = 5px$ and $f = 10$ (median of 36 remaining points per contour), medium deformations using $\sigma_R = 10px$ and $f = 40$ (median of 9 remaining points per contour), and high deformations (see Figure 5.5) using $\sigma_R = 20px$ and $f = 80$ (median of 5 remaining points per contour).

In addition to deformed annotations, we also simulate the case where the expert chooses a faster supervision process using only bounding boxes to identify objects of interest. In this case, we replace each annotation by the smallest bounding box which includes the entire object.

Experts who annotate a large dataset may miss objects of interest. We create “noisy datasets" by randomly removing the annotations of a certain percentage of objects. A corrupted dataset with “50% of noise" is therefore defined as a dataset where 50% of the objects of interest are relabelled as background. As there is some variation in the size of the objects, we verified that the percentage of pixels removed from the annotations ranged linearly with the percentage of omitted objects, as shown in Figure 5.6.

Different imperfections are also combined: noise with deformations and noise with bounding boxes, resulting in different “SNOW datasets", as illustrated in Figure 5.5.

### 5.3.2 Experiments and evaluation

The following experiments were performed on the GlaS dataset:

a) Compare the **corrupted annotations** to the **ground truth annotations**. This estimates the performance of a model that *perfectly matches* the imperfect annotations.
b) Measure the performance of the three networks based on different **levels of noise, erosion/dilation radii** and **simplification factors**.

The DSC was used as a general-purpose binary segmentation metric in these experiments, as the goal was not to focus on any particular use case but to look at the overall effect of SNOW imperfections on the performances of the algorithms. Each network was trained once on each of the randomly corrupted training dataset, then evaluated based on the original annotations of the test set.

### 5.3.3 Results

The results of the first experiment are presented in Table 5.2. A low amount of deformation is associated with a 4% loss in the DSC. This indicates that when a pixel-perfect segmentation is difficult to define (for instance with objects with fuzzy or debatable boundaries), results based on typical segmentation metrics should be interpreted carefully. In such a situation, a difference of a few percent between two algorithms could thus be considered as irrelevant.

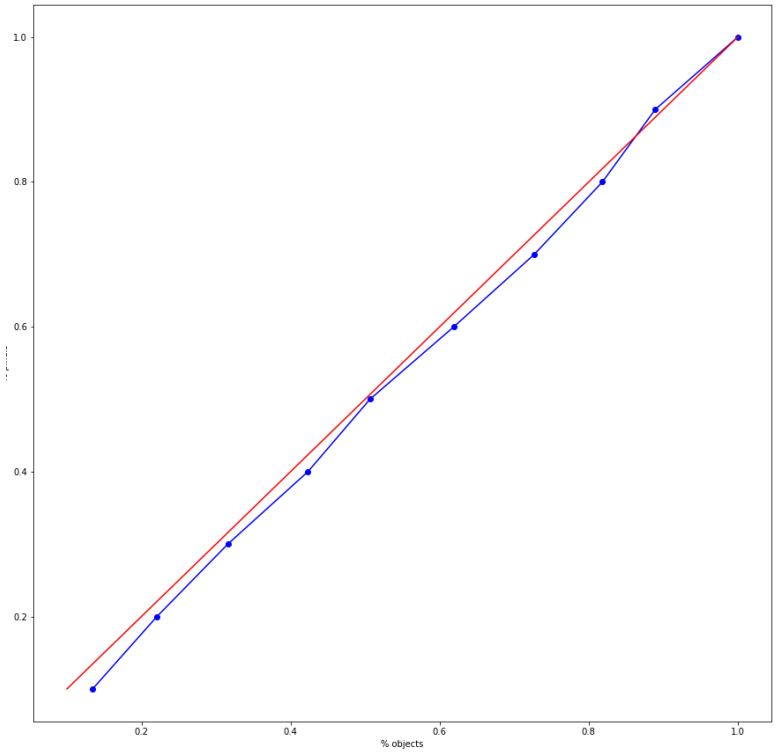



**Figure 5.6. Percentage of remaining pixels in the annotations as a function of the percentage of remaining objects in the noisy datasets.**

Figure 5.7 shows the effects of increasing noise levels introduced in the supervision of the GlaS training set on the performance of the 3 baseline DCNNs. Despite their differences in terms of size and architecture, the three networks behave very similarly, with some robustness up to 30% of noisy labels. However, a clear decrease in performance is observed from 40% or 50% of supervision noise. The effects of annotation erosion or dilatation are much less drastic, and polygonal approximations seem to have no significant effects in the range of $f$ considered in these tests. For ShortRes, the performance on the "bounding boxes" dataset, which can be seen as an extreme form of polygonal simplification, are about the same as the effects of large erosion/dilations. The three networks behave in a very similar way with regard to these types of annotation corruption. It should be noted that the the variability of the results due to the randomness of the corruption process is not taken into account in those results. However, the overall trends identified in the results are unlikely to be affected by this randomness. Uncertainty in the measured results may increase for the largest values of the corruption parameters, but these values correspond to such a large performance drop on the three networks that the effect is very unlikely to be a random anomaly.

Somewhat surprisingly, as shown in Table 5.2, the ShortRes network trained on the combined "50% noise + bounding boxes" corruptions actually performs better than the network trained on the 50% noise with no deformation dataset. This is likely to be a result of the very high object density of this particular dataset. As the bounding boxes cover more tissue area, they may give the networks a bias in favour of the positive pixel class, which helps them get a better score on the uncorrupted test set: as we have shown in Chapter 4, overlap-based metrics such as the IoU or the DSC are biased in favour of results that overestimate the segmented region.

**Table 5.2. DSC of different corrupted datasets compared to the original ground truth annotations (in the training set), alongside the DSC for the ShortRes trained on the corrupted annotations (measured on the original test set).**

| Dataset | DSC (Corrupted vs Original, training set) | DSC (ShortRes vs Original, test set) |
|---|---|---|
| Original (GlaS) | 1.000 | 0.841 |
| 10% Noise | 0.931 | - |
| 50% Noise | 0.589 | 0.231 |
| Low deformation ($\sigma_R = 5px, f = 10$) | 0.960 | - |
| Medium deformation ($\sigma_R = 10px, f = 40$) | 0.917 | - |
| High deformation ($\sigma_R = 20px, f = 80$) | 0.830 | - |
| Bounding boxes | 0.836 | 0.724 |
| 50% Noise + High deformations | 0.455 | 0.212 |
| 50% Noise + Bounding boxes | 0.557 | 0.511 |

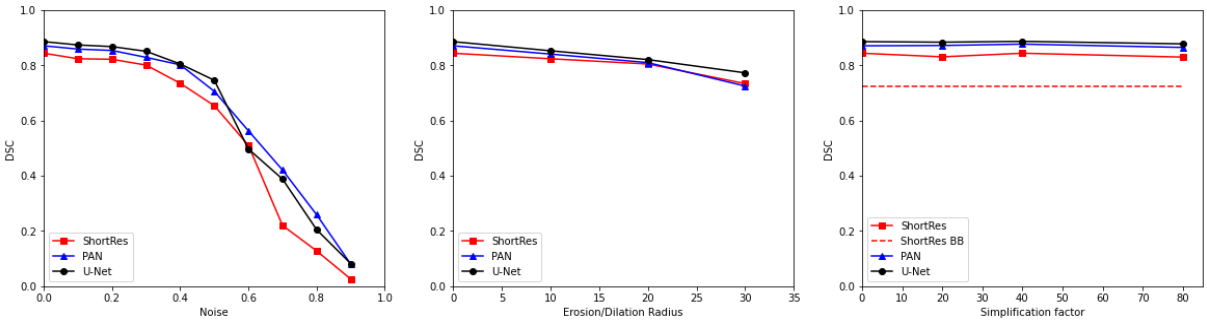



**Figure 5.7. Effects of SNOW on trained models using the ShortRes, PAN and U-Net architectures. (left) Effects of increasing levels of label noise, (middle) of increasing the random erosion/dilation radius standard deviation $\sigma_R$ of the annotations, (right) of increasing the simplification factor $f$ of the object contours polygonal approximation, with the performance on bounding boxes shown with a dashed line for the ShortRes network. Each point corresponds to the results of one trained network on one corrupted dataset.**

## 5.4 Experiments on learning strategies

The next set of experiments we performed aimed at assessing the performance of different learning strategies on the corrupted datasets. Given the very similar behaviours observed above for the three baseline networks with respect to SNOW supervision, only the ShortRes network is used in the first experiments on the GlaS datasets to investigate the effects of different learning strategies. This allows us to draw the first lessons that we then confirm on the Epithelium dataset using the ShortRes and PAN networks, knowing that original versions of both datasets are similar in terms of the quality and nature of the annotations.

### 5.4.1 Learning strategies

For the following strategies, we consider two subsets of the training data. The *positively annotated* regions ($R^+$) corresponds to all pixels which are within a 20px margin of the bounding box of an annotated object in the dataset. The *negatively annotated* regions ($R^-$) corresponds to all remaining pixels. Practically, we first compute the bounding boxes of all objects annotated in the training set and extend their boundaries by 20px in all directions. When sampling patches, a patch is considered "in $R^+$" if it intersects with at least one pixel of these extended bounding boxes, and "in $R^-$" otherwise.

#### *5.4.1.1 Only Positive (OnlyP)*

In this approach, only input patches in $R^+$ are used for training the model.

#### *5.4.1.2 Semi-supervised learning (SSL)*

A two-step approach (see Figure 5.8) is used for the semi-supervised strategy and is based on the fact that all our networks follow the classic encoder-decoder segmentation architecture described in section 1.5.2.

First, an auto-encoder (AE) is trained on the entire dataset by replacing the original decoder (with a segmentation output) by a shorter decoder with a reconstruction output. The Mean Square Error loss function between the network output and the input image is used to train the auto-encoder, with an L1 regularization loss on the network weights to encourage sparsity.

The second step consists of replacing the decoder part of the AE by the decoder of the segmentation network, and then training the whole network on the supervised dataset. The encoder part of the network is therefore first trained to detect features as an auto-encoder, and then fine-tuned on the segmentation task, while the decoder of the final network is trained only for segmentation. In the experiments on the SNOW datasets reported below, we test two variants of the semi-supervised strategy depending on the supervised data on which the network is fine-tuned: either the full supervised (and corrupted) dataset, or only patches from $R^+$. This latter version will be referred to as "SSL-OnlyP" in the result tables.

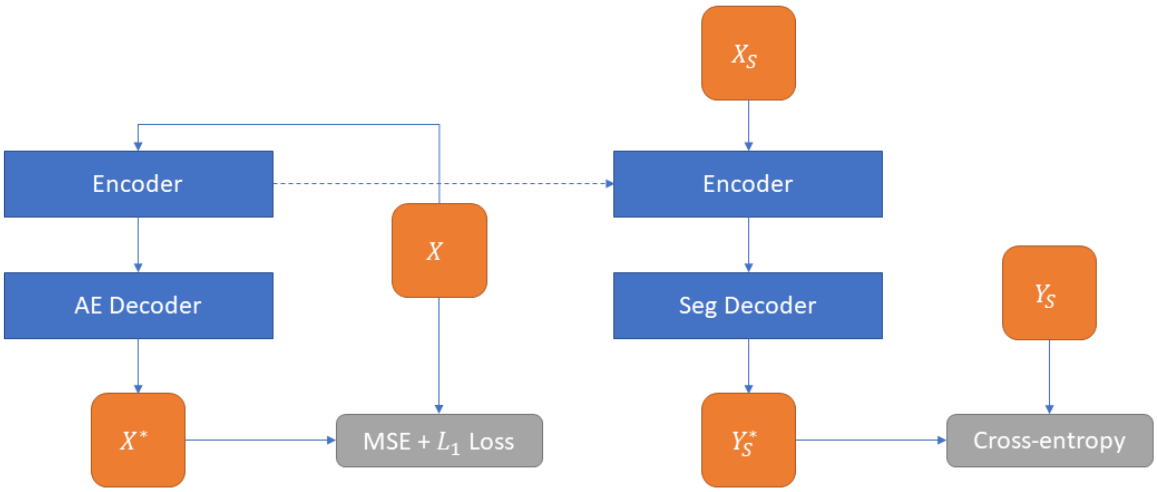



**Figure 5.8. Principle of the semi-supervised learning method based on the auto-encoder. The AE is first trained on the whole unsupervised dataset. The encoder is kept, and the decoder is replaced to be trained on the supervised part of the dataset.**

#### 5.4.1.3 Generated Annotations (GA)

The "Only Positive" strategy may tend to overestimate the likelihood of the objects of interest, especially in cases where they have a fairly low prior (such as in our artefact dataset, used in Chapter 6). We propose a slightly different, original approach based on a two-step method detailed as follows (with pseudo-code in Table 5.3). First, we train an **annotation generator** network with patches sampled from $R^+$ (i.e. using the "Only Positive" strategy). This annotation generator is then used to reinforce the learning of the final **segmentation** network, which will be trained on the whole dataset using the following procedure. If the patch is in $R^+$, the original annotations are used as supervision. If it is in $R^-$, it randomly chooses between using the original (empty) annotations *or* the output of the annotation generator as supervision. The probability of each choice should depend on the object prior. As the original datasets are quite strongly biased towards the presence of objects of interest, we use a probability of either 75% (GA75) or 100% (GA100) of using the annotation generator output.

This strategy can be seen as a version of semi-supervised learning because the regions without annotations are sometimes treated as "unsupervised" rather than with a "background" label. But it is also based on label noise estimation. The assumption that positive regions are more likely to have correct annotations results in a highly asymmetric noise matrix, with $P(y_i = 1 \mid t_i = 0) \gg P(y_i = 0 \mid t_i = 1)$, where $t_i$ is the true class and $y_i$ the class provided by the imperfect supervision. The Generated Annotations strategy includes this information by treating positive annotations as correct for training and negative annotations as uncertain.

**Table 5.3. Pseudo-code for the process of constructing minibatches in the "Generated Annotations" learning strategy, where $\mu$ is the hyper-parameter determining the probability of using the annotation generator as supervision**

**Generated annotations strategy**

Let $G$ be the **generator** model
Let $S$ be the **segmentation** model
Let $R^+ = \{p_i = (x_i, y_i)\}$ be all the image patches and corresponding annotations in the positive regions, and $R^-$ for the negative regions.
**Train** $G$ using the patches sampled from $R^+$
**Train** $S$: procedure for each minibatch
  **Initialize** $M = \emptyset$ as the current minibatch
  **Sample** patch $p_i = (x_i, y_i)$
  **If** $p_i \in R^+$:
    **Append** $(x_i, y_i)$ to $M$
  **Else:**
    **Draw** $r \in U(0,1)$ (uniform distribution)
    **If** $r > \mu$:
      **Append** $(x, y)$ to $M$
    **Else**
      **Compute** $y_G = G(x)$
      **Append** $(x, y_G)$ to $M$
  **Repeat** until desired *batch size*

#### 5.4.1.4 Label augmentation (LA)

Knowing that labels could be imperfect, especially around the borders, we create slightly modified versions of the supervision via morphological erosion or dilatation (with a 5 pixels radius disk) of

the objects of interest that are randomly presented during learning. Following a purpose similar to that of classical data augmentation, this strategy aims at making networks robust to annotation modifications.

#### *5.4.1.5 Patch-level annotation strategies*

As mentioned above, typical weak strategies rely on patch-level annotations. However, such strategies are not appropriate for the datasets described in section 5.2.1, because these sets include very few examples of negative patches (5% for GlaS and 13% for Epithelium). This means that with original or noiseless datasets, “weak" networks would see almost only positive examples, whereas with noisy data sets, they would see either correct positive examples or incorrect negative examples. In either case, they will not be able to learn. Therefore, these strategies were not used in our experiments.

### 5.4.2 Evaluation procedure

The networks are trained with patches randomly drawn from the training set images. The patch size is determined for each dataset by preliminary testing on the baseline network, with the goal of finding the smallest possible patch size on which the network can learn. 256x256 pixels patches were selected for the GlaS dataset and 128x128 pixels patches for the Epithelium dataset. To evaluate the results on the test set, images are split in regular overlapping tiles, with 50% overlap between two successive tiles. For each tile, the networks produce a probability map. As most pixels (except those close to the borders) are seen as part of multiple tiles, the maximum probability value for the “positive" class is assigned as the final output. A mask is then produced using a 0.5 threshold applied to this final output. It should be noted that contrary to what is usual in image segmentation, no further post-processing is applied on the results. This aims to avoid contaminating the experiences by external factors but with the consequence of somewhat penalizing our baseline networks compared to what is reported in the literature.

We initially used the DSC as a general-purpose metric for both publicly available datasets and their corrupted versions, as the objective of this experiment is not to solve a particular digital pathology task, but to compare the effects of the learning strategies on segmentation accuracy. The DSC is computed for each image of the test set. To determine significant differences between the strategies, in terms of performance achieved with a given training set, the DSC obtained on the same test image are compared by means of the Friedman test and the Nemenyi post-hoc test, as described in Chapter 4. The DSC, however, is (like the IoU) asymmetrical in its treatment of over-estimation vs under-estimation of the segmented region. In a dataset where the proportion of positive tissue is so large, the DSC of a method that mostly predict positive pixels everywhere will be much better than that of a method that is more conservative in its predictions. We therefore also compute the per-pixel MCC to take the “true negatives” into account in the evaluation, thus treating the segmentation task as a per-pixel binary classification task. While the MCC is not a commonly used segmentation metric, its symmetrical properties are nonetheless useful in gathering insights on the behaviour of the different learning strategies and can capture information that are not easily detected with the DSC.

On the GlaS dataset, a “statistical score" is also computed to highlight the actual differences in performance between the tested strategies. For each corrupted dataset and for each pairwise strategy comparison, if the difference between the two strategies is judged significant by the post-hoc test ($p < 0.05$), a positive point is assigned to the best learning strategy and a negative point to the other. The statistical score is computed by summing those points for each learning strategy. The best learning strategies so determined are then applied on the Epithelium dataset.

### 5.4.3 Results

#### *5.4.3.1 Results on the GlaS dataset*

The average DSC and MCC on the GlaS test set of the learning strategies are reported in Table 5.4, alongside the statistical score. The first notable thing is that all learning strategies outperform the baseline method on most corrupted training sets. Using the DSC, it appears that the three most effective learning strategies overall are the "Only Positive" and the two "Generated Annotations" methods. The "Only Positive" results, however, are far less impressive when looking at the MCC, particularly on the "50% Noise + High deformations" (NoisyHD) dataset. This is due to a strong bias towards positive predictions on that particular dataset. From the MCC results, the two "Generated Annotations" methods appear to outperform the others, with the SSL-OnlyP method as the only other method that, on average, outperforms most others.

The per-dataset results show, however, that the learning strategies have different strengths and weaknesses when it comes to the types of imperfection in the annotations. The SSL-OnlyP strategy, for instance, significantly outperforms all others when trained on the "50% Noise + Bounding boxes" (NoisyBB) set, where the OnlyP method also performs relatively well. Both perform very poorly, however, when trained on the NoisyHD set. The SSL and LA methods, meanwhile, have the worse performances overall (outside of the baseline), yet they still perform well when trained on the "Bounding boxes" (BB) set (although all methods have very similar results when using those annotations).

**Table 5.4. Average DSC and MCC computed on the (non-corrupted) test set (80 images) of the GlaS dataset for the ShortRes network trained with different learning strategies on the different corrupted annotations. Scores in bold are not found significantly different (i.e. $p > 0.05$) from the score of the best strategy for that dataset using the Nemenyi post-hoc test (comparing the scores obtained on the same test image). The statistical score calculates a balance between the number of significant pairwise comparisons where the result of the strategy is the worst and those where it is the best (see main text for details).**

| DSC | Original | Noisy | BB | NoisyBB | NoisyHD | Stat. score |
|---|---|---|---|---|---|---|
| Baseline | **0.841** | 0.231 | 0.724 | 0.511 | 0.212 | -22 |
| OnlyP | **0.836** | **0.768** | 0.730 | **0.697** | **0.660** | 10 |
| SSL | 0.831 | 0.467 | **0.756** | 0.522 | 0.207 | -11 |
| SSL-OnlyP | 0.819 | **0.729** | 0.740 | **0.730** | 0.428 | 5 |
| GA100 | 0.837 | **0.764** | **0.755** | **0.700** | **0.621** | 12 |
| GA75 | **0.843** | **0.736** | **0.754** | 0.695 | **0.608** | 10 |
| LA | **0.837** | 0.575 | **0.761** | 0.631 | 0.449 | -4 |
| **MCC** | **Original** | **Noisy** | **BB** | **NoisyBB** | **NoisyHD** | **Stat. score** |
| Baseline | **0.683** | 0.231 | 0.355 | 0.328 | 0.200 | -18 |
| OnlyP | **0.668** | **0.577** | 0.409 | 0.450 | 0.012 | -1 |
| SSL | 0.664 | 0.372 | **0.474** | 0.347 | 0.186 | -6 |
| SSL-OnlyP | 0.647 | **0.537** | 0.413 | **0.487** | 0.070 | 5 |
| GA100 | 0.664 | **0.588** | **0.455** | 0.413 | **0.438** | 11 |
| GA75 | **0.687** | **0.573** | **0.453** | 0.410 | **0.433** | 11 |
| LA | **0.674** | 0.450 | **0.488** | 0.337 | 0.322 | -2 |

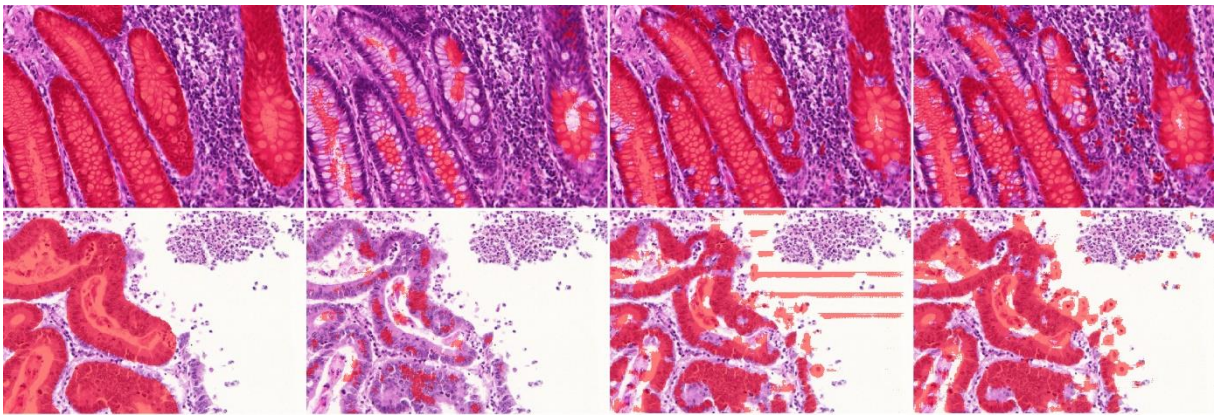


**Figure 5.9. Results on two images from the GlaS test set obtained with the ShortRes network trained on the Noisy set with different learning strategies. From left to right: correct segmentation, Baseline, OnlyP, GA100. Positive pixels are shown in red.**

Results of the "50% Noise" (Noisy) training on some images from the test set are shown in Figure 5.9. The baseline network severely underestimates the target regions. The OnlyP strategy recovers a lot of the performance but has large regions of false positives. The GA100 offers a form of compromise between the two, and generally has the best results overall. The raw results could be considerably improved in all cases with some basic post-processing (such as morphology operations), but these raw results make the effects of the learning strategies more visible.

#### *5.4.3.2 Results on the Epithelium dataset*

The average DSC and MCC on the Epithelium dataset of the different learning strategies are reported in Table 5.5. To get some additional perspective on the results, we also report the Average Precision (AP), which is the area under the precision-recall curve. A single AP is computed on all test images at once, so no statistical test was done for that metric.

Those results confirm that the baseline networks are robust even to large deformations of their annotations. They also confirm that, for the noisy annotations, strategies that focus the training on the areas with "positive" pixels (which are more likely to be correctly annotated) manage to recover well even from a dataset with 50% of annotations removed. Even though the difference is not statistically significant according to the Nemenyi post-hoc, the Semi-Supervised method does seem to perform slightly better with the ShortRes network. The Label Augmentation method, meanwhile, does not outperform the baseline even on the deformed dataset.

An interesting insight from computing the AP is that the baseline appears more robust than when looking at the single-threshold metrics, even on the noisy labels. This indicates that, while a lot of positive tissue is missed using the standard 0.5 threshold at the output, the predicted value is still generally higher in positive tissue than in negative tissue. The Precision-Recall curve on the noisy labels dataset, shown in Figure 5.10, provides some additional information on the behaviour of the strategies. The SSL-OnlyP method has a better AP than the other strategies, but it also has an "inverted U" shape that indicates that it has some very high confidence false positive predictions.

**Table 5.5. Average DSC, MCC and AP computed on the (non-corrupted) test set (7 images) of the Epithelium dataset for the ShortRes and PAN network trained with different learning strategies on the different corrupted annotations. Scores in bold are not found significantly different (i.e. $p > 0.05$) from the score of the best strategy for the Noisy dataset using the Nemenyi post-hoc test (comparing the scores obtained on the same test image). The Friedman test is not significant for the Original and Deformed (HD) datasets.**

| **DSC** | | **Original** | **Noisy** | **Deformed** |
|---|---|---|---|---|
| Shortres | Baseline | 0.853 | 0.545 | 0.811 |
| | OnlyP | 0.848 | **0.730** | 0.808 |
| | GA100 | 0.848 | **0.671** | 0.809 |
| | LA | 0.837 | 0.577 | 0.808 |
| | SSL-OnlyP | 0.853 | **0.788** | 0.816 |
| PAN | Baseline | 0.860 | 0.639 | 0.828 |
| | OnlyP | 0.857 | **0.762** | 0.826 |
| | GA100 | 0.861 | **0.681** | 0.827 |
| | LA | 0.858 | 0.648 | 0.814 |
| | SSL-OnlyP | 0.853 | **0.768** | 0.788 |
| **MCC** | | **Original** | **Noisy** | **Deformed** |
| Shortres | Baseline | 0.781 | 0.466 | 0.717 |
| | OnlyP | 0.767 | **0.646** | 0.714 |
| | GA100 | 0.770 | **0.595** | 0.717 |
| | LA | 0.777 | 0.489 | 0.733 |
| | SSL-OnlyP | 0.778 | **0.718** | 0.747 |
| PAN | Baseline | 0.786 | 0.559 | 0.746 |
| | OnlyP | 0.783 | **0.685** | 0.746 |
| | GA100 | 0.788 | **0.606** | 0.744 |
| | LA | 0.785 | 0.572 | 0.731 |
| | SSL-OnlyP | 0.774 | **0.684** | 0.740 |
| **AP** | | **Original** | **Noisy** | **Deformed** |
| Shortres | Baseline | 0.952 | 0.845 | 0.929 |
| | OnlyP | 0.952 | 0.913 | 0.921 |
| | GA100 | 0.951 | 0.912 | 0.923 |
| | LA | 0.948 | 0.861 | 0.932 |
| | SSL-OnlyP | 0.955 | 0.931 | 0.948 |
| PAN | Baseline | 0.953 | 0.886 | 0.938 |
| | OnlyP | 0.953 | 0.924 | 0.934 |
| | GA100 | 0.954 | 0.919 | 0.936 |
| | LA | 0.957 | 0.888 | 0.933 |
| | SSL-OnlyP | 0.957 | 0.924 | 0.943 |

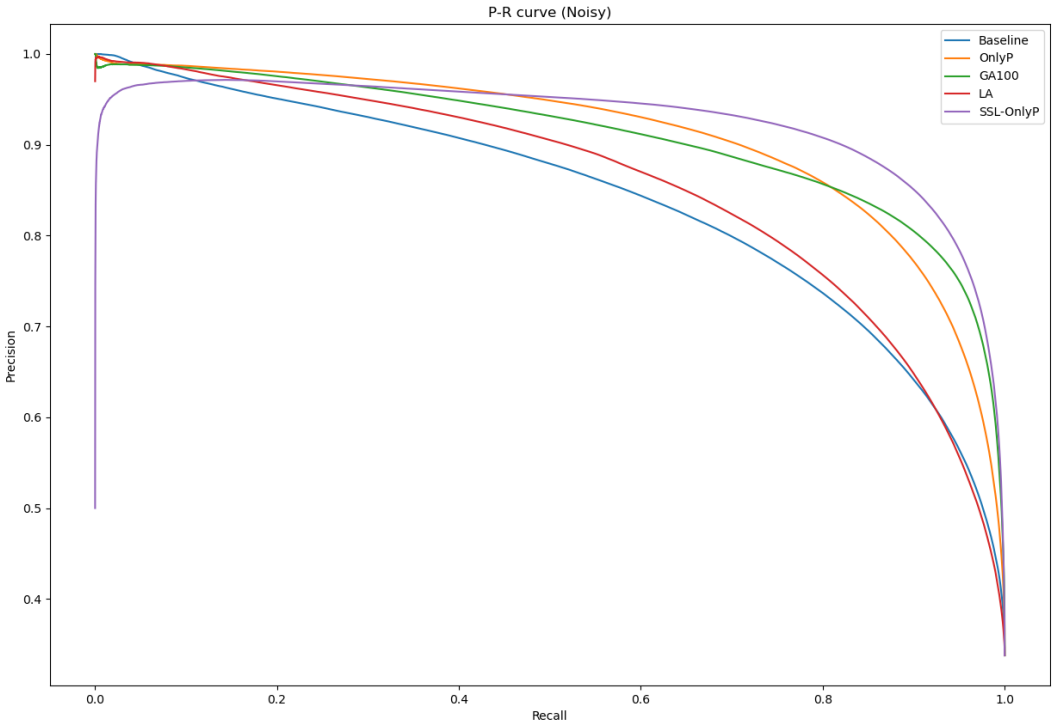



**Figure 5.10. Precision-Recall curve of the ShortRes network trained using the different learning strategies on all images of the test set.**

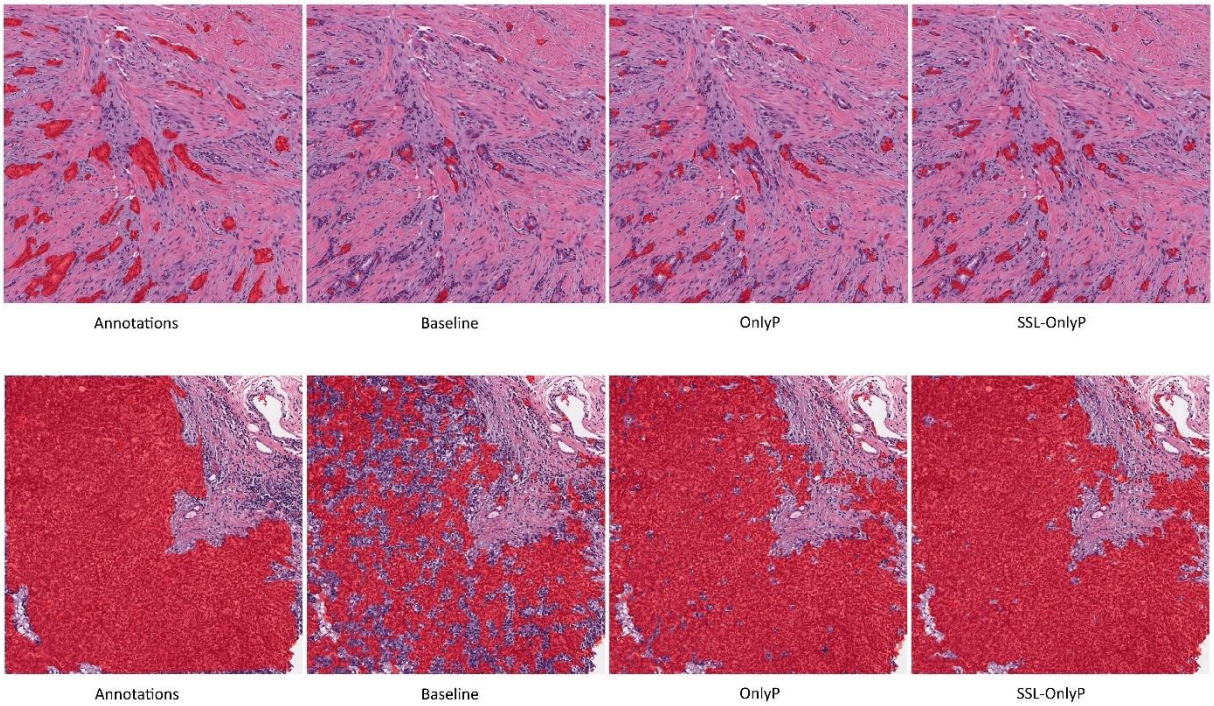



**Figure 5.11. Example results on the images with the worst (top) and best (bottom) average results from the Epithelium test set for the Baseline, OnlyP and SSL-OnlyP strategies, alongside the ground truth annotations (left).**

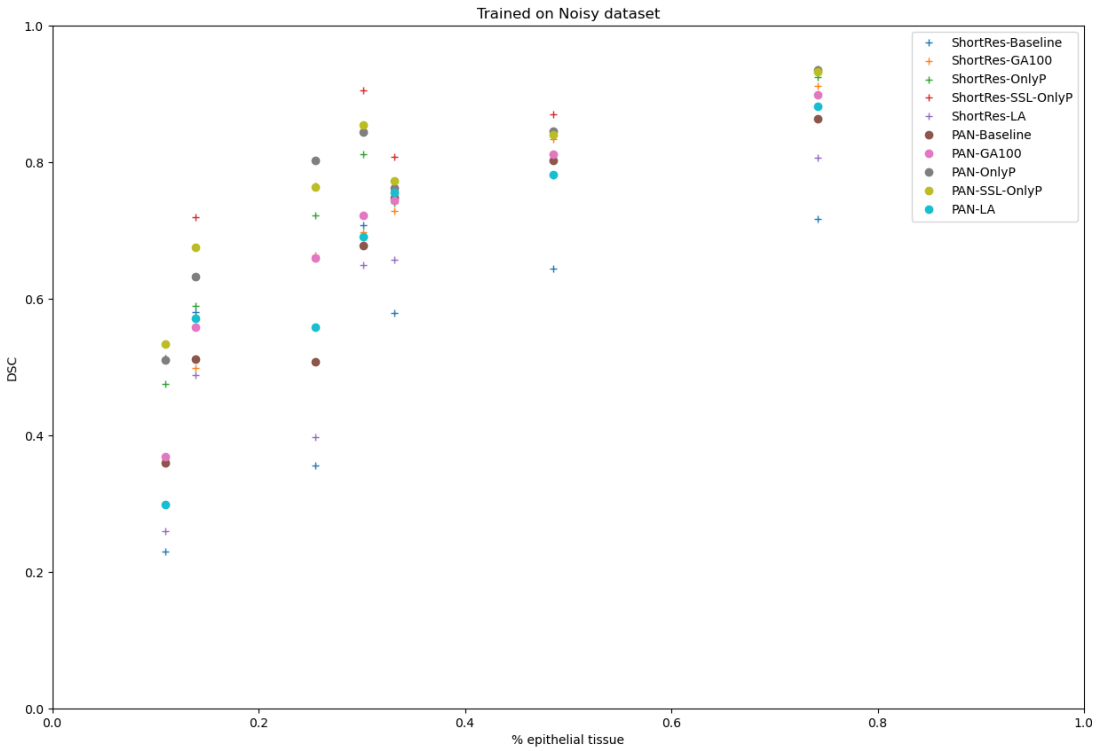



**Figure 5.12. Relationship between the proportion of epithelial tissue in the test set images and the DSC of the different networks and strategies trained on the "50% label noise" dataset.**

The results on the Epithelium dataset also illustrate how important the distribution of the dataset is to the results. As Figure 5.11 shows, the image where the different models have the worst performance according to the DSC and MCC metrics is an image with relatively few objects of interest. Conversely, the image where the models perform best is one which is nearly entirely covered in "positive" (epithelial) tissue. This is confirmed in Figure 5.12, which plots the relationship between the ratio of epithelial tissue in an image and the DSC of the networks and learning strategies. Images with sparse epithelial tissue are clearly associated with lower values for the DSC and MCC.

## 5.5 Comparison with similar experiments

Similar experiments to our 2019 [2] and 2020 [4] publications have since then been made by Karimi et al. [85] and by Vadineanu et al. [260].

Karimi et al. introduced increasing levels of realistic deformations on foetal brain annotations in diffusion weighted MR images. They show that their baseline network (with a U-Net like architecture) is fairly robust to levels of deformation similar to our experiments, and that the results only start to fall when the DSC of the deformed set compared to the original annotations becomes lower than around 0.8 (our "high deformation" dataset had a DSC of 0.830). They compare two learning strategies: using an adapted loss function, and an "iterative label update" method which is similar in principle to our "GA" method. The latter has better performances in general for most noise levels, although the adapted loss function is slightly better for extreme levels of deformations. However, no statistical analysis was provided and the results in general

are relatively close even with the baseline, so it is difficult to draw strong conclusions from their results.

Vadineanu et al., meanwhile, introduced label noise and deformations in a very similar fashion to our experiments (with the addition of "inclusion" errors, i.e. "negative" objects labelled as "positive") for cell segmentation in immunofluorescence images, and for epithelial nuclei segmentation using the MoNuSAC 2020 dataset. They only studied the effects of the noise on baseline networks but did not introduce additional learning strategies. Their results show that the networks are relatively robust to up to 30% of label noise with, like in our experiments, a big fall in performance starting at 40-50% noise. They show a stronger sensitivity of their networks to deformations. However, their target objects are very small and the metric used is the DSC, which means that even small deviations from the annotated "ground truth" are harshly penalized, as evidenced in Chapter 4, section 4.4.6.

## 5.6 Impact on evaluation metrics

While we have focused in this chapter on the impact of SNOW annotations when training a DCNN, we should not neglect the impact on evaluation metrics. Even when experts are instructed to take more time on the test set annotations, or when senior experts are more involved in the validation of these annotations, there will always remain some level of imprecision and noise (even neglecting interobserver variability). This impact is, by its very nature, extremely difficult to objectively assess. It should, however, not be neglected. As an example, Cruz-Roa et al. [205] note in their 2014 publication on invasive ductal carcinoma detection that "[a] closer examination of the qualitative segmentation results suggests that some misclassifications (both FP and FN errors) are a result of imperfect manual annotation."

A first possibility for evaluating an algorithm using SNOW annotations is to use a very small subset of the test set so that several experts can agree on the best possible annotations for that subset, then to compare those annotations to the corresponding test set "ground truth". This can provide a lower boundary to the range of differences in metrics between algorithms that cannot reasonably be considered to be true differences in performances. For instance, if the DSC between the "optimal annotations" and the "test set annotations" in a segmentation problem is 0.98, then a difference lower than 0.02 in the DSC should always be considered as an ex aequo, before any statistical testing even needs to be applied.

The other approach is to accept the uncertainty on the value of the metric, and to instead look at the SNOW annotations in the context of the discussion of the results. This can include having an expert look at randomly selected errors from the evaluated algorithms and assessing the likelihood of the algorithm being correct. One possibility for a fair assessment would be to show examples of regions where the "ground truth" and the "prediction" are different and ask the expert to choose the best of the two (ideally in a blind setting). While it would generally be impractical to apply this kind of analysis to all errors and all algorithms (particularly in the context of a challenge where tens or hundreds of methods are being compared), performing it even on a small subset of the data would at least provide an idea of the difference in performance that needs to be observer between the algorithms to be reasonably certain that one outperforms the other.

## 5.7 Conclusions

Deep learning models are trained and evaluated on a "ground truth", which is assumed to be exact. Practically, this is however not the case. Imperfections are unavoidable in any dataset, but they are particularly important in digital pathology. Segmentation tasks are the most difficult to properly annotate, as they require pixel-precise supervision on objects which are often very small, and which can have irregular, fuzzy boundaries. It is also impractical to annotate entire WSI for detection or segmentation: supervision will therefore always be limited to a set of regions of interest.

There has been a lot of work on how to manage those imperfections for training deep neural networks, and to best leverage the annotations that are available despite their imperfections. For segmentation, it is clear that deep neural networks are relatively robust to strong deformations, and to limited amounts of noise. This suggests that the best annotation strategy is to be *exhaustive* within a region of interest in the identification of the objects, without necessarily being *precise* about the exact boundaries. Using polygonal approximations rather than trying to pixel-precise, for instance, would generally be sufficient and considerably speed up the annotation process. The best learning strategies, meanwhile, harness the full unsupervised dataset to determine the best feature space, and focus the learning of the discriminative part of the network on the well annotated regions. Re-labelling unannotated regions (or mislabelled regions in the case of multi-class problems) may be useful as well, particularly when the annotated subset if small.

The question of how to deal with imperfect annotations for the evaluation of the algorithms has been less explored. All evaluation metrics assume the existence of a perfect ground truth, so that predictions can be categorised as "correct" or "incorrect". It is generally difficult to really include the uncertainty due to imperfections to this computation, as evaluating the level of imperfection would require, in turn, a perfect "ground truth" for comparison. To get out of this circular problem without ignoring it, evaluation processes should include a review by experts of the errors made by the algorithms to assess the proportion of these errors that may be attributed to mistakes in the annotations themselves rather than in the predicted labels.

Imprecise, noisy and incomplete annotations are unavoidable in real-life applications. Ensuring that deep neural networks are capable of handling SNOW annotations in their learning process is crucial to their clinical application. Furthermore, recognizing these imperfections in the evaluation procedure is necessary to avoid rewarding algorithms that reproduce the imperfections and penalizing algorithms that may be as good or better at the clinical task despite being less similar to the available "ground truth".

# 6 Artefact detection and segmentation

Rolls and Farmer define an artefact in histology or cytology as "a structure that is not normally present in the living tissue" [282]. Artefacts can be by-products of every step of the histopathology tissue processing workflow. Kanwal et al. [283] list and show examples of some common types of artefacts at various stages of the WSI acquisition pipeline:

- Damage of the **tissue**, which can happen during the biopsy or resection, during paraffin block sectioning (e.g. tears, shown in the bottom-left of Figure 6.1), or during staining.
- Artefacts on the **slide** during coverslipping, such as dirt, air bubbles or pen markings on the glass coverslip.
- **Scanning** artefacts, with the most common being blur and stitching artefacts (as scanners often acquire "tiles" which are then stitched together to form the WSI).

While some artefacts are easily distinguished by pathologists, the distinction between "artefact" and "normal tissue" may sometimes be difficult to make [284]. In automated image analysis of histological slides, artefacts can cause potential mistakes in quantitative analyses. Automated segmentation of artefacts may therefore **improve the results of such quantitative processing** by excluding artefactual tissue. It also provides a mean of **controlling the quality of the workflow of a laboratory**, to assess the usability of a WSI for diagnostic purpose or to signal if a re-cut, re-staining or re-scanning of the tissue sample may be necessary [285].

The problem of artefact segmentation also illustrates well the difficulties of digital pathology image analysis covered in this thesis. As the frontier between "artefact" and "normal tissue" is not well-defined, pathologist scoring of the WSI quality has a relatively high interobserver variability [286]. As the boundaries of the artefact are very difficult to strictly define, there will be a large uncertainty on the exact extent of the artefactual region. The diversity in nature, shape and size of the artefacts (see Figure 6.1) also make it extremely difficult for an expert to exhaustively annotate artefactual regions, making the quantitative evaluation of the automatic artefact segmentation results very challenging as well.

In our 2018 publication [1], we tested several deep learning strategies for the segmentation of artefactual regions. We later used our artefact dataset as a case study for some of the "SNOW" experiments [4], and developed a prototype for an automated tool for analysing WSI in a pathology workflow. Several similar tools were in the meantime developed by other research teams, such as Case Western Reserve University's HistoQC [287], or more recently PathProfiler from the Institute of Biomedical Engineering of the University of Oxford [285].

In this chapter, we will first present the state of the art of artefact detection and segmentation as it stood before our 2018 publication, i.e. before deep learning methods were proposed for this application. We will then present our experiments and results. We will finally explore the latest advances and the current state-of-the-art and discuss how these results may help the pathology practice.

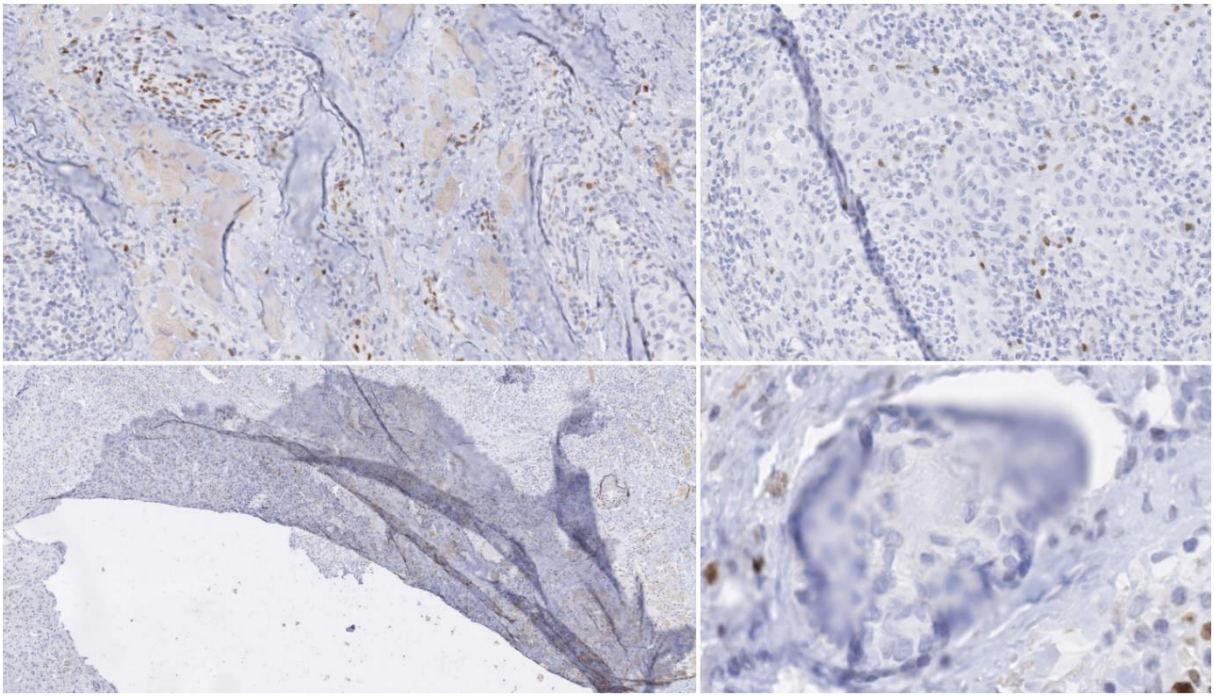


**Figure 6.1. Examples of artefacts extracted from an anti-NR2F2-stained IHC image from head and neck carcinoma.**

## 6.1 State of the art before deep learning

Most of the literature on artefact detection focuses either on one or few specific artefacts (most commonly, blur or tissue-folds), or on computing an overall "quality score" for the slide, generally based on blur, noise or contrast [283]. Before our 2018 experiments, all the proposed methods were based on traditional image analysis pipelines with handcrafted features. In this section, we present the state-of-the-art methods for overall quality assessment, blur detection and tissue-fold detection for those traditional methods.

### 6.1.1 Quality assessment

As the quality assessment of a slide is an important use case for artefact segmentation, many algorithms focus solely on determining an overall objective quality score rather than trying to find specific instances of artefacts. Moreover, these quality scores are often mostly concerned with **acquisition** artefacts, such as blur or noise. "Sharpness" and "noise", with sometimes the addition of other general features such as contrast or colour separation, serve as the basis for several quality scores, such as those proposed by Hashimoto et al. [288], Ameisen et al. [289], Shrestha et al. [290] or Shakhawat et al. [291]. The evaluation of such methods is particularly difficult, and generally consists in verifying a correlation with a subjective assessment by a panel of experts. All of these algorithms are relatively straightforward, using gradients, edge detection, and simple neighbourhood operators to determine the relevant statistics, with a linear combination of the individual components providing an overall score.

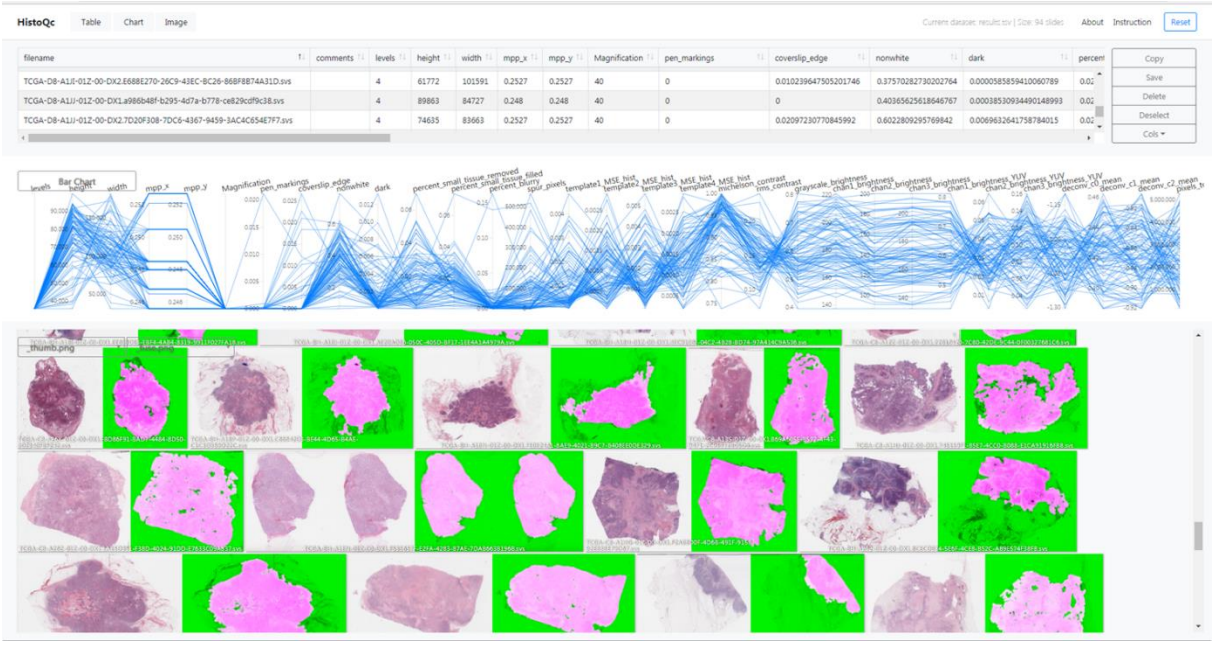



**Figure 6.2. Web interface of HistoQC, from the project's GitHub page (https://github.com/choosehappy/HistoQC).**

### 6.1.2 Blur detection

Blur is the most common acquisition artefact. As the focus parameters of the scanner needs to be adapted during the acquisition depending on the local tissue thickness of the slide, some regions may end up out-of-focus [292]. If quickly detected, it is also a relatively easy artefact to fix, as it only requires the re-acquisition of the out-of-focus regions.

Methods aiming at specifically finding the blurry regions in the whole slide generally take the form of a tile classification algorithm. If the size of the tile is small enough, this provides sufficient precision for the localization of the blurry region, particularly if the aim is to guide re-acquisition rather than to exclude the region from further processing.

Moles Lopez et al. [292] combine the "sharpness" and "noise" features from Hashimoto et al. [288]'s quality score with several other features based on the gradient image and the grey-level cooccurrence matrix statistics. A Decision Tree classifier is then trained on the features to provide a tile classification model.

Similarly, Gao et al. [293] use a diverse set of local features (contrast, gradient, intensity statistics, alongside some wavelet features) and train a set of "weak" Linear Discriminant Analysis classifiers, combined together using AdaBoost. A large set of intensity-based, gradient-based and transform-based features is used by Campanella et al. [294] to train a Random Forest classifier.

These methods show that relatively simple feature-based algorithms obtain good results for blur detection.

### 6.1.3 Tissue-fold detection

Tissue-folds are very commonly found in pathology tissue, caused by the thin tissue folding on itself while being manipulated. This is particularly common around tears in the tissue, as shown in the bottom-left of Figure 6.1.

One of the earliest tissue-fold detection method was proposed by Palokangas et al. in 2007 [295]. They observed that tissue-fold regions are characterized by a high saturation and low intensity, and used an unsupervised approach with k-means clustering to segment the folds from the rest of the slide. This "high saturation, low intensity" characteristic was also exploited by Bautista et al. [296], who proposed a simple image enhancement function: $I_e = I + \alpha(S - V)$, where $I_e$ is the enhanced image, $I$ the original image, $S$ the saturation and $V$ the intensity value. A simple luminosity thresholding can then be applied on the enhanced image to segment the tissue folds.

This method is further refined in Kothari et al. [297] in 2013, which also uses $S - V$ to separate the folds from the rest of the tissue but add a method for adapting the threshold to the image being considered. These authors note that, by varying the threshold on the $S - V$ image, the number of connected components is initially low (as most pixels are segmented and therefore merged into a few large regions), then grows rapidly to a peak (as the segmentation includes tissue-folds regions as well as many disjointed nuclei) before going down as only the tissue-fold regions remain. Two thresholds are computed based on the number of connected objects at the peak, and pixels are labelled as "tissue-fold" if their $S - V$ value is larger than the lower threshold and they are in the 5x5px neighborhood of a pixel whose $S - V$ value is larger than the higher threshold.

More recently in 2020, Shakhawat et al. [291] use gray-level cooccurrence matrix statistics and pixel luminance as features to train a SVM on tile-based tissue-fold classification. They also use the same approach for air bubble detection.

### 6.1.4 HistoQC

HistoQC is an "open-source quality control tool for digital pathology slides"[42]. It was initially presented in a 2019 publication by Janowczyk et al. [287], with further experiments and validations presented in a 2020 publication by Chen et al. [286]. It proposes a collection of modules that each target a specific type of artefact. Ad-hoc pipelines can be created in a configuration file to apply those modules to a target WSI. It also provides a web-based interface to visualize the results, shown in Figure 6.2. The result of the pipeline is a set of statistics describing the different aspects of the quality of the slide, and a mask separating the normal tissue from the background glass slide and the artefacts (which include air bubbles, pen markings, tissue-folds, and blurry regions).

The software was validated in the 2019 publication [287] in a qualitative analysis involving two expert pathologists who determined whether the results proposed by the software were "acceptable" (defined as "an 85% area overlap between the pathologists' visual assessment and the computational assessment by HistoQC of artefact-free tissue") on a set of 450 slides. They reported an agreement on 94% and 97% of the slide with the two experts (similar to the 96% inter-expert agreement). In the larger 2020 experiment involving three pathologists and 1800 slides [286] (but limited to renal biopsies), they showed the potential of this type of software as an aid to the pathologist, reporting a considerable improvement in inter-expert agreement when using the software than when assessing the quality of the slide without the help of the application.

[42] https://github.com/choosehappy/HistoQC

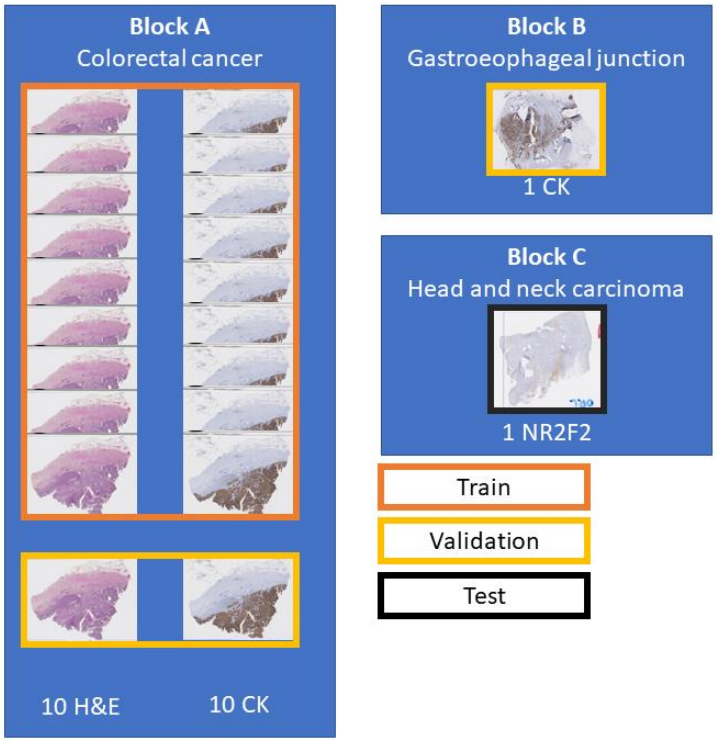



**Figure 6.3. Constitution of the artefact dataset for the experiments: 18 slides from Block A (9 H&E, 9 IHC with anti-pan-cytokeratin) are used for training, 3 slides from Block A and B (1 H&E, 2 IHC) for validation, and one final slide (Block C) with a different IHC stain (anti-NR2F2) for final testing. Block C was annotated by an histology technologist, while Block A and B were annotated by the author.**

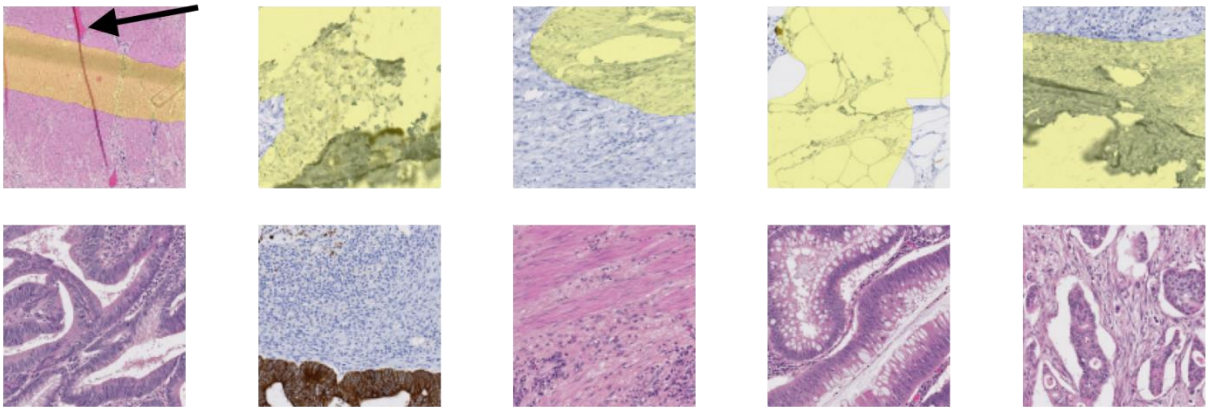


**Figure 6.4. Examples of 128x128px tiles extracted from the WSIs in block A, with their corresponding annotation mask (positive artefactual regions in yellow). A missing annotation can be seen in the first tile (black arrow).**

## 6.2 Experimental results

Our first experiments on artefact segmentation were presented at the CloudTech conference in 2018 [1], but our main work on the topic was included as a case study in our SNOW experiments in 2020 [4]. We will focus here on the latter results, which we report and extend.

### 6.2.1 Materials

A full description of our artefact dataset can be found in Annex A, and a visual summary is shown in Figure 6.3. The dataset contains WSIs extracted from three tissue blocks. Twenty slides come from a colorectal cancer tissue block ("Block A", 10 with H&E staining, and 10 with anti-pan-cytokeratin IHC staining). One from a gastroesophageal junction tissue block ("Block B", stained with anti-pan-cytokeratin IHC), and one from head and neck carcinoma ("Block C", with anti-NR2F2 IHC staining). The annotations for the training set (18 slides from Block A) and the validation set (2 slides from block A and one from block B) were done by the author. They are weak (with approximative contours) and noisy (few of the annotated objects should be incorrect, but many small artefacts are not annotated). The test slide (from block C) is annotated by a histology technologist. However, while the quality of the annotations is much better than in the training and evaluation set, there are still many missing small artefacts, and the exact boundaries are still uncertain. An important characteristic of the dataset (especially compared with the challenge datasets used in Chapter 5) is the scarcity of the "positive" class (in this case, artefactual regions). In the training set, only 2% of the pixels are annotated as "positive". In the validation set and the test slide, where more care was taken to be as exhaustive as possible in the annotations, this ratio increases to 7% and 9%, respectively. Examples of annotated tiles from block A are shown in Figure 6.4.

### 6.2.2 Methods

As in the previous SNOW experiments, the ShortRes, PAN and U-Net architectures were used. In terms of learning strategies, the baseline, GA50 and Only Positive strategies were chosen based on the results of the experiments on the corrupted GlaS and Epithelium datasets reported in Chapter 5.

An insight from our early experiments was that the very large uncertainty on the annotations made the quantitative assessment of the performance of the algorithms difficult. Typical segmentation or per-pixel classification metrics (such as DSC, SEN, SPE…) generally failed to capture much useful information about the relative strengths and weaknesses of the trained networks. The qualitative evaluation was generally more informative and easier to relate to the potential for practical use of the system in a laboratory setting.

The evaluation method was therefore adapted to have a more qualitative approach. Twenty-one tiles or varying dimensions (between around 400x400 and 800x800px) were extracted from the three "validation" slides (see Figure 6.3). Eight of the 21 test tiles have no or very few artefact pixels. The others show examples of tissue tears & folds (6), ink stains (2), blur (2), or other damage.

For each slide and network, we classify the result on each test tile as Good (results are acceptable), False Negative (some artefacts are not detected or the segmented region is too small), False Positive (some tissue regions without artefact are segmented), or Bad (completely misses artefacts or detects too much normal tissue as artefacts). Examples of such results are illustrated in Figure 6.5, where tissue regions considered as normal are shown in pink and those considered as artefactual are shown in green, using the visualisation convention proposed in HistoQC [287].

To compare the results of the different strategies and networks, we score the predictions on each tile by giving penalties according to the type of error (Good = 0, False Positive = 1, False Negative = 2, Bad = 3). False positives are given a lower penalty than false negative, as it is typically better to overestimate an artefactual region than to misidentify an artefact as normal tissue. We compute

the sum of the penalties on all 21 tiles to get a final penalty score, a lower penalty score thus meaning a better strategy. While we have seen that quantitative metrics are poor indicators for the performances of the algorithms in the absence of an objectively and correctly annotated ground truth, they can still be useful as a measure of similarity between the predictions of the different methods (thus comparing the methods with each other rather than with the annotations). We compute the DSC and the per-pixel MCC of each algorithm compared with each other to form similarity matrices, which are used to generate an MDS visualisation of the similarity between algorithms (as already used in section 4.4.5).

The last slide from Block C, which is distinguished from the others by the tissue origin and the IHC marker (see Figure 6.3), is used to visually assess the results on an independent whole slide image. Additionally, four H&E slides from the TCGA database containing different types of artefacts (identified in the "HistoQC Repository[43]") are added as an external dataset for the qualitative evaluation.

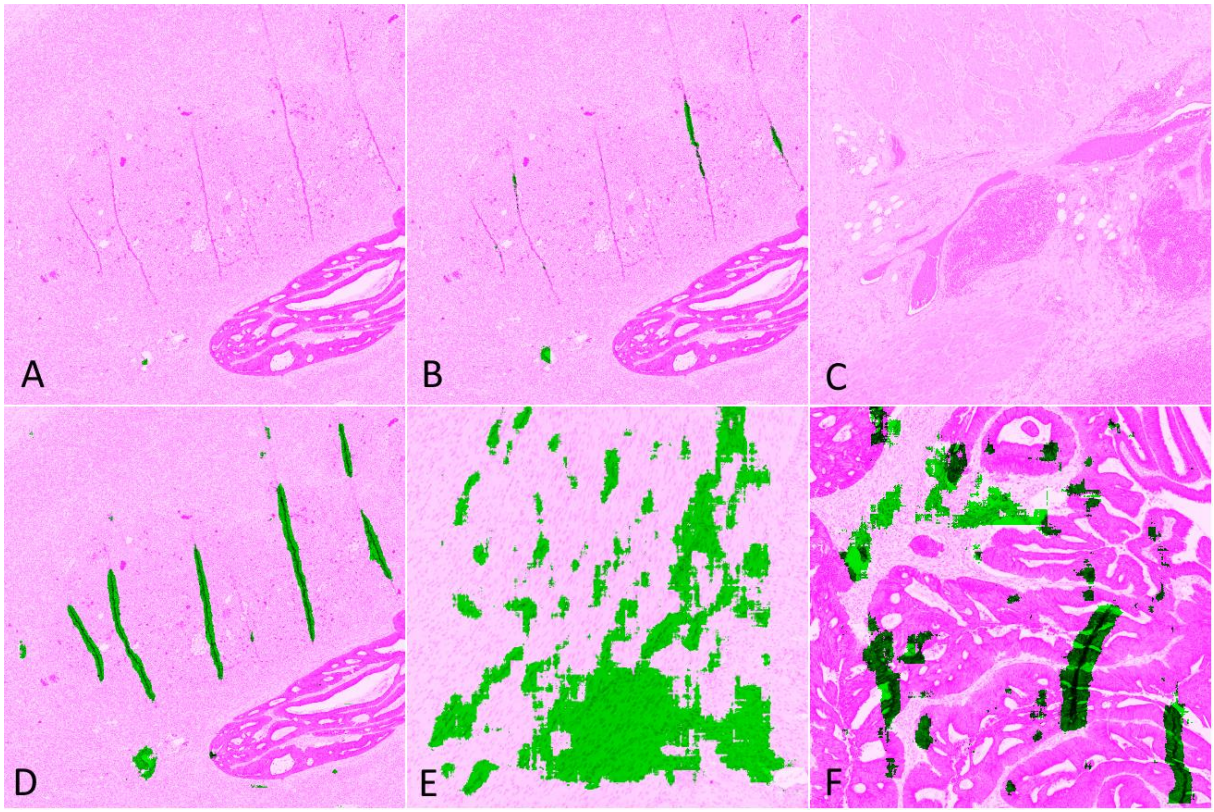



**Figure 6.5. Illustration of the classification of results. Detected artefacts are shown in green, normal tissue in pink. (A) "Bad" – none of the artefacts found, (B) "False Negative" – most of the artefactual region missing, (C) "Good" – normal region correctly classified, (D) "Good" – most artefacts found, (E) "Bad" – most normal tissue incorrectly identified as artefact, (F) "False Positive" – some normal tissue incorrectly identified as artefact.**

[43] http://histoqcrepo.com/

**Table 6.1. Results of the selected networks and strategies on the 21 validation tiles of the artefact dataset, with the computed penalty score (see main text). The results in bold identify the best strategy for each network architecture.**

| ShortRes | Good | FP | FN | Bad | Penalty Score |
|---|---|---|---|---|---|
| Baseline | 14 | 0 | 5 | 2 | 16 |
| **GA50** | **16** | **1** | **4** | **0** | **9** |
| Only Positive | 13 | 7 | 0 | 1 | 10 |
| **PAN** | **Good** | **FP** | **FN** | **Bad** | **Penalty Score** |
| Baseline | 13 | 0 | 5 | 3 | 19 |
| **GA50** | **19** | **0** | **2** | **0** | **4** |
| Only Positive | 8 | 12 | 0 | 1 | 15 |
| **U-Net** | **Good** | **FP** | **FN** | **Bad** | **Penalty Score** |
| Baseline | 13 | 0 | 6 | 2 | 18 |
| **GA50** | **19** | **1** | **1** | **0** | **3** |
| Only Positive | 3 | 10 | 0 | 8 | 34 |

For whole-slide prediction, we first perform background detection (i.e. glass side without tissue) by downscaling the image by a factor of 8, converting the image to the HSV colour space, and finding background with a low saturation ($S < 0.04$). The resulting background mask is rescaled to the original size and fused with the artefact segmentation result. All slides are analysed at 1.25x magnification. We use a regular 128x128 pixels tiling of the whole slide with 50% overlap and keep the maximum output of the artefact class for every pixel.

### 6.2.3 Results on the validation tiles

The results on the validation tiles are shown in Table 6.1. The GA50 strategy consistently performs better than the baseline and Only Positive for all three network architectures. The Only Positive strategy consistently overestimate the artefactual region, with the largest FP number, while the baseline network tends to underestimate it. In this case of low density of objects of interest, the results show that limiting the training only to annotated regions and regions close to annotations is too restrictive. It should be also noted that for the three networks the GA50 strategy is able to retrieve all the “bad" cases from the baseline, which seems less accurate with PAN and U-Net than with ShortRes.

The comparisons between the different algorithms using the DSC and per-pixel MCC are reported in Figure 6.6, and the MDS visualisation of the dissimilarity (using the DSC) is shown in Figure 6.7. It is immediately apparent that the main driver of the similarity between the results of the methods is the learning strategy rather than the network architecture. The Only Positive strategy has a much larger spread to its cluster than the two others, indicating that its results are more network-dependent than the two other strategies. This is illustrated in Figure 6.8, where we can see for one of the validation tiles that the Only Positive strategy predicts a lot more false positive artefacts with the U-Net than with the ShortRes network, while the GA50 has much more similar predictions between the two networks.

The bad results of the Only Positive method, however, may be in part attributable to the fixed 0.5 threshold used to produce the prediction mask from the pixel probabilities. If, instead of the MCC or DSC, we compute the average cross-entropy between the probability maps of the networks and build a similarity matrix, the MDS visualisation (Figure 6.9) shows that the predictions of the PAN

and ShortRes Only Positive networks are much more similar to the GA50 networks than was apparent from the binarized predictions. The GA50 appears in both cases to be a “compromise” between the OnlyP and baseline strategies.

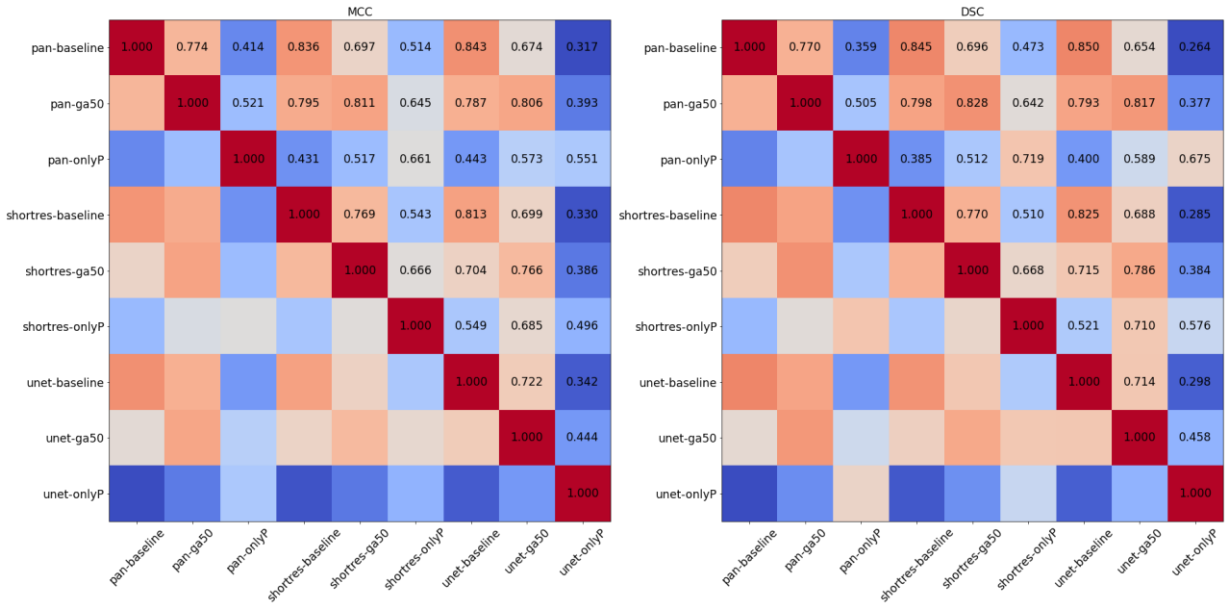



**Figure 6.6. Similarity matrix of the different algorithms, using the per-pixel MCC (left) and the DSC (right) as metrics. Blue indicates low similarity, red high similarity.**

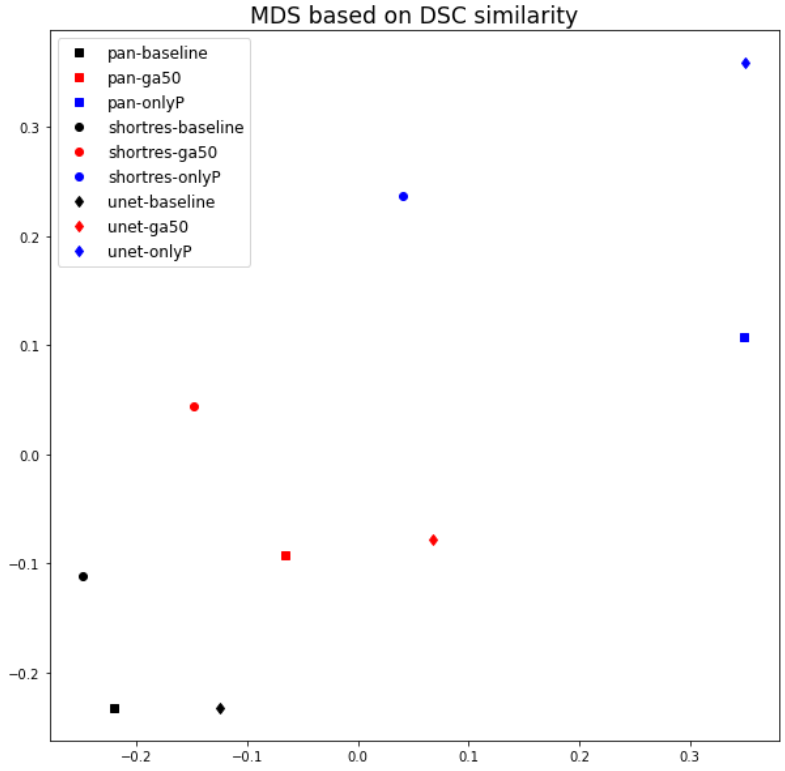



**Figure 6.7. MDS visualisation of the DSC similarity shown in Figure 6.6 between the different algorithms. The shapes of the points indicate the network architecture (squares = PAN, circles = ShortRes, diamonds = U-Net), and the colour the learning strategy (black = baseline, red = GA50, blue = Only Positive).**

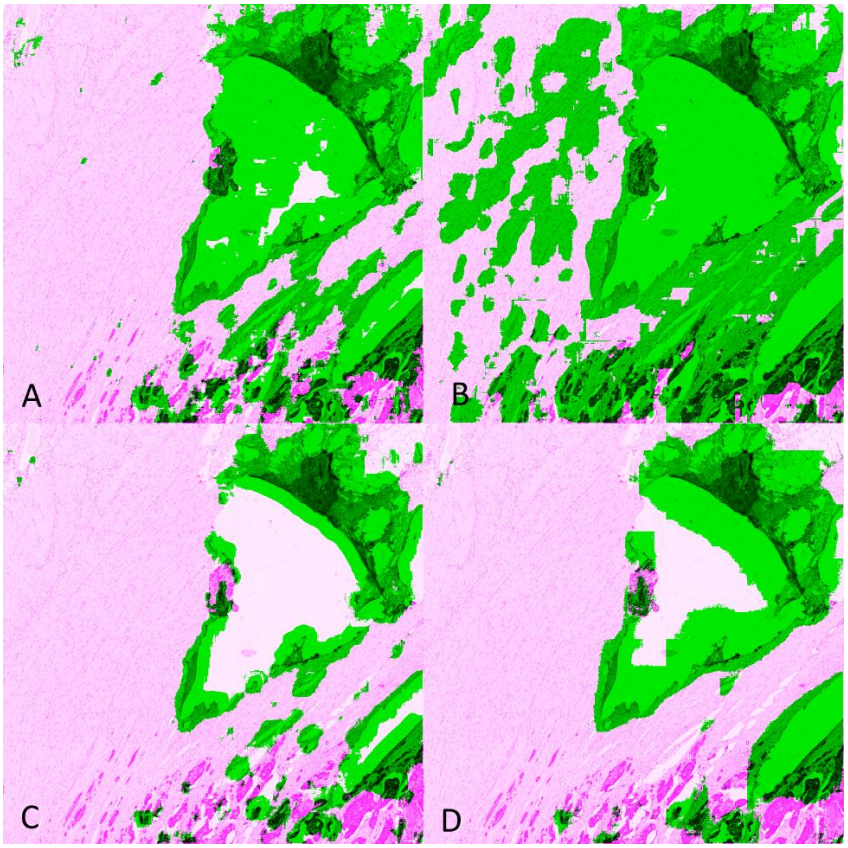



**Figure 6.8. Predictions on a validation tile for (A) ShortRes / Only Positive, (B) U-Net / Only Positive, (C) ShortRes / GA50, (D) U-Net / GA50. Detected artefacts are shown in green, normal tissue is shown in pink.**

### 6.2.4 Results on the test slides

Given the results on the validation tiles, the PAN-GA50 network was used on five test whole-slide images for a qualitative evaluation: the "Block C" slide (see Figure 6.3) and four slides from the TCGA dataset, chosen for the presence of different types of artefacts as identified on the HistoQC repository. Our observations are summarized in Table 6.2, and all the predictions are shown in Figure 6.10-Figure 6.14. It should be noted that the processing time for PAN-GA50 took around 2 minutes 20 seconds for the 4 TCGA slides using an NVIDIA TITAN X GPU.

**Table 6.2. Qualitative results of PAN-GA50 on the test WSI (including TCGA ones).**

| Slide | (Main) artefacts | PAN-GA50 result |
|---|---|---|
| Block C | Tears and folds | PAN-GA50 misses some small artefacts but its results are generally acceptable. (Figure 6.10) |
| A1-A0SQ | Pen marking | Pen marking is correctly segmented and small artefacts are found. Some intact fatty tissue is mistakenly labelled (see black arrows in Figure 6.11). |
| AC-A2FB | Tissue shearing, black dye | The main artefacts are correctly identified. (Figure 6.12) |
| AO-A0JE | Crack in slide, dirt | Some intact fatty tissue is mistakenly labelled (see black arrows in Figure 6.13), but all artefacts are found and almost all intact tissue is kept. |
| D8-A141 | Folded tissue | The main artefacts are correctly identified. (Figure 6.14) |

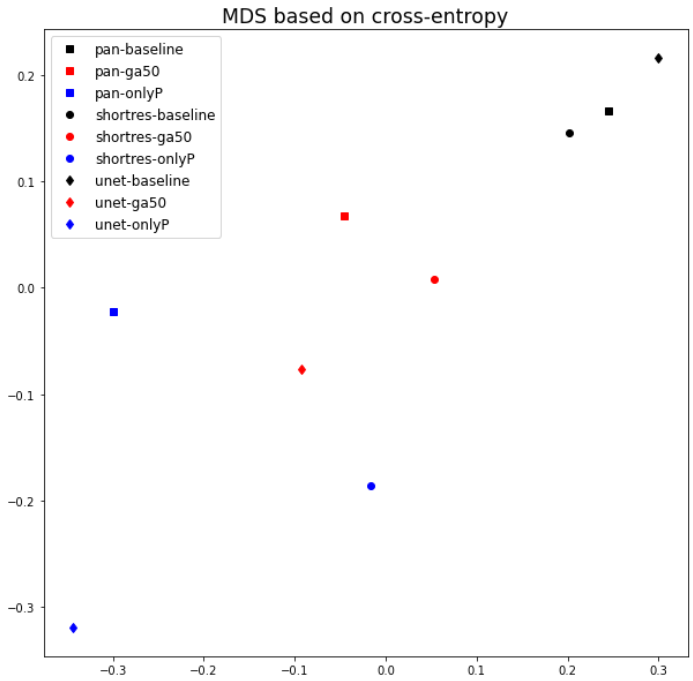



**Figure 6.9. MDS visualisation of the Cross-Entropy similarity between the output probability maps of the different algorithms. The shapes of the points indicate the network architecture (squares = PAN, circles = ShortRes, diamonds = U-Net), and the colour the learning strategy (black = baseline, red = GA50, blue = Only Positive).**

The most recurrent error in the test slide predictions is found in fatty tissue. This tissue has a very distinct appearance (looking like large, empty cells), and were not present in the training dataset. They are therefore likely confused with small holes, and thus identified as artefactual regions.

### 6.2.5 Insights from the SNOW experiments

Our GA method succeeds in learning from a relatively small set of imprecise annotations, using images from a single tissue type. It generalizes well to new tissue types and previously unseen IHC markers. This method provides a good compromise between using as much of the available data as possible (as in semi-supervised methods) and giving greater weight to the regions where we are more confident in the quality of the annotations (as in the Only Positive strategy). The baseline method underestimates the artefactual region, as expected from the low density of annotated objects in the dataset. The Only Positive strategy, on the other hand, is too limited in the data that it uses and, therefore, has too few examples of normal tissue to correctly identify the artefacts.

While the PAN network was slightly better than the ShortRes network with the GA50 strategy on the test tiles, it performed worse with the Baseline version. Since ShortRes is significantly simpler (20x less parameters), these observations suggest that for problems such as artefact detection, better learning strategies do not necessarily involve larger or more complex networks.

By using strategies adapted to SNOW annotations, we were able to obtain very good results on artefact segmentation with minimal supervision. Extending the network to new types of artefacts or particular appearances of the tissue (such as fat) should only require the addition of some examples with quick and imprecise annotations for fine-tuning.

**Figure 6.10. Prediction of artefactual regions of the PAN-GA50 network on the Block C slide. Detected artefacts and background are shown in green, normal tissue is shown in pink.**

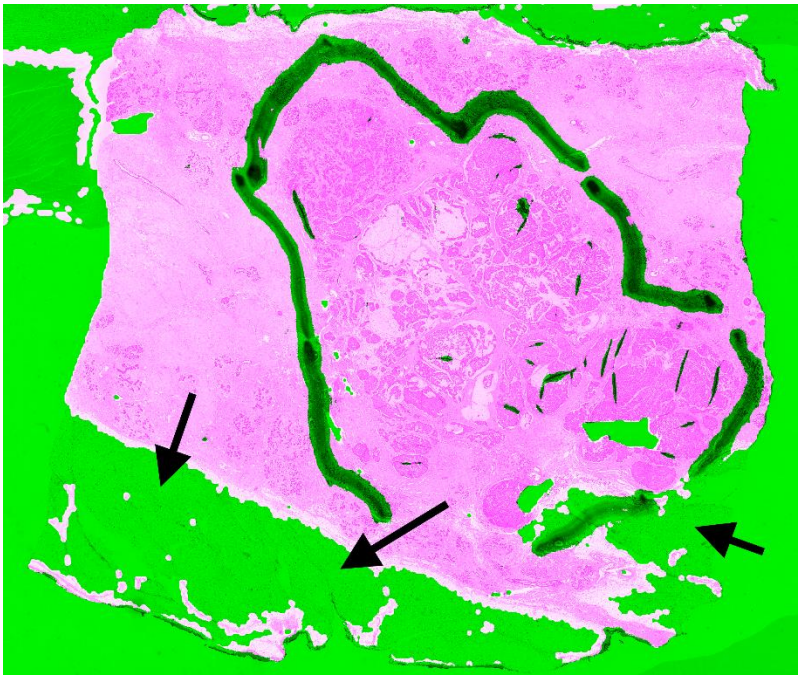


**Figure 6.11. Prediction of artefactual regions of the PAN-GA50 network on the TCGA A1-A0SQ slide. Black arrows point to intact fatty tissue incorrectly labelled as artefacts.**

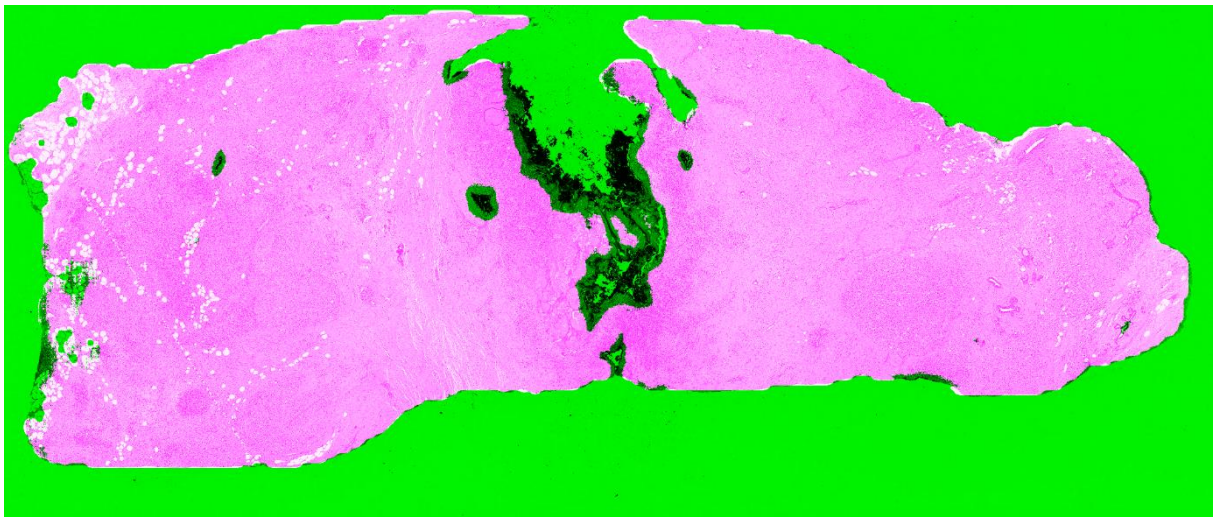


**Figure 6.12. Prediction of artefactual regions of the PAN-GA50 network on the TCGA AC-A2FB slide.**

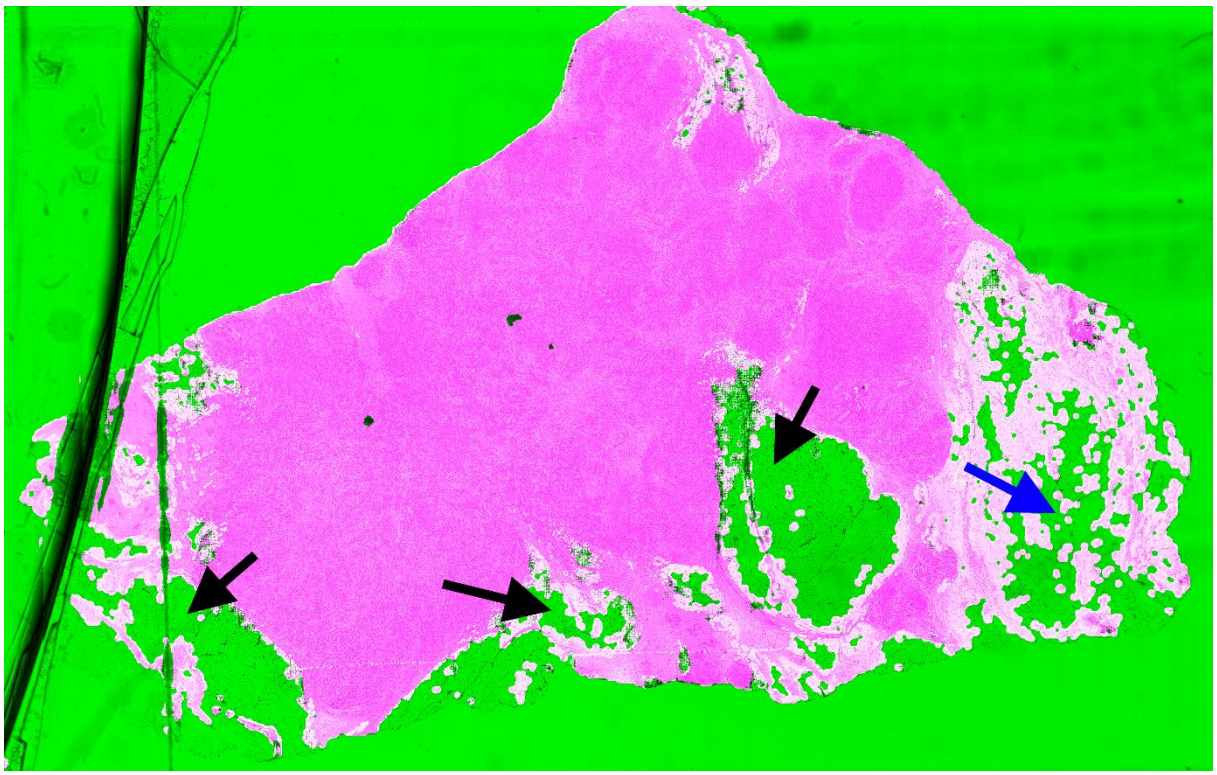


**Figure 6.13. Prediction of artefactual regions of the PAN-GA50 network on the TCGA AO-A0JE slide. Black arrows point to intact fatty tissue incorrectly labelled as artefacts. The blue arrow points to damaged fatty tissue.**

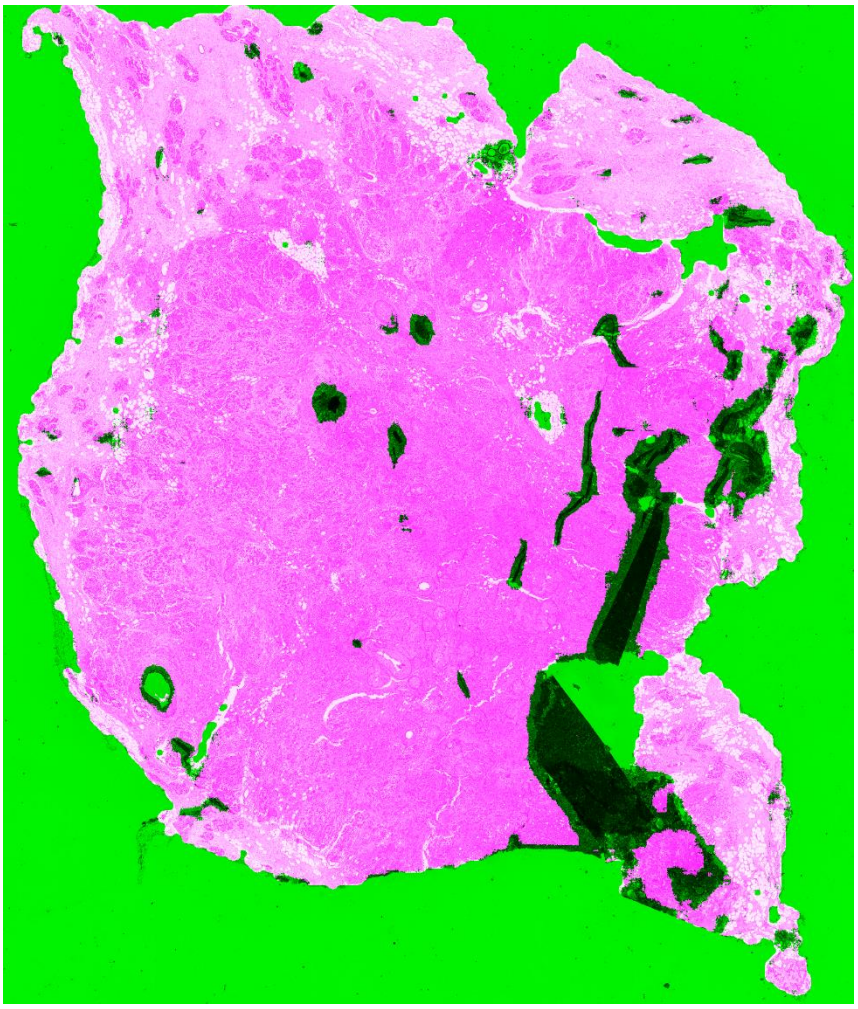


**Figure 6.14. Prediction of artefactual regions of the PAN-GA50 network on the TCGA D8-A141 slide. Detected artefacts and background are shown in green, normal tissue is shown in pink.**

## 6.3 Prototype for a quality control application

To study the possibility of including our network trained for artefact segmentation in a laboratory environment for quality control, a prototype application was developed with a web-based interface for quickly visualising the quality of newly acquired WSIs. A diagram showing the general concept of the application is shown in Figure 6.15, and a screenshot of the web interface in Figure 6.16. The overall architecture of the application is largely based on HistoQC, except that it uses our trained PAN-GA50 network to segment all artefacts instead of their classic image analysis approach, and it requires no specific configuration for each WSI, as our model generalizes well to different stains and organs.

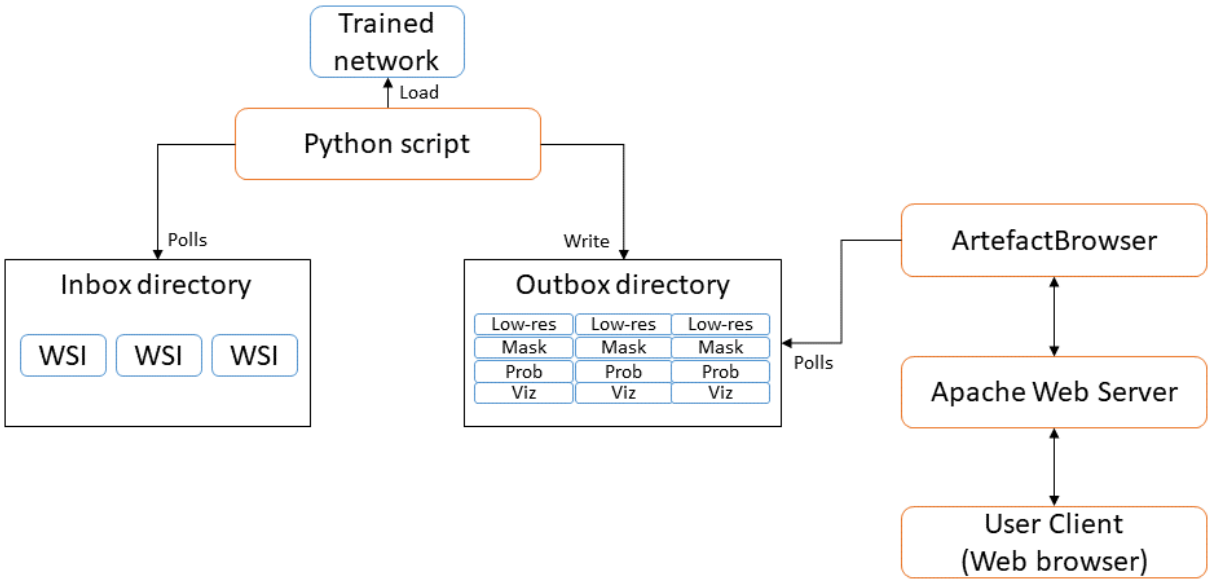



**Figure 6.15. Prototype for a quality control application developed around the trained PAN-GA50. A python script polls an "inbox" directory where newly acquired WSI are sent. The lower-resolution mask and probability map for the artefactual regions, as well as the color-coded visualisations (see Figure 6.16) are written to an outbox directory, which is in turn polled by the web application so that the results are quickly visible to an operator.**

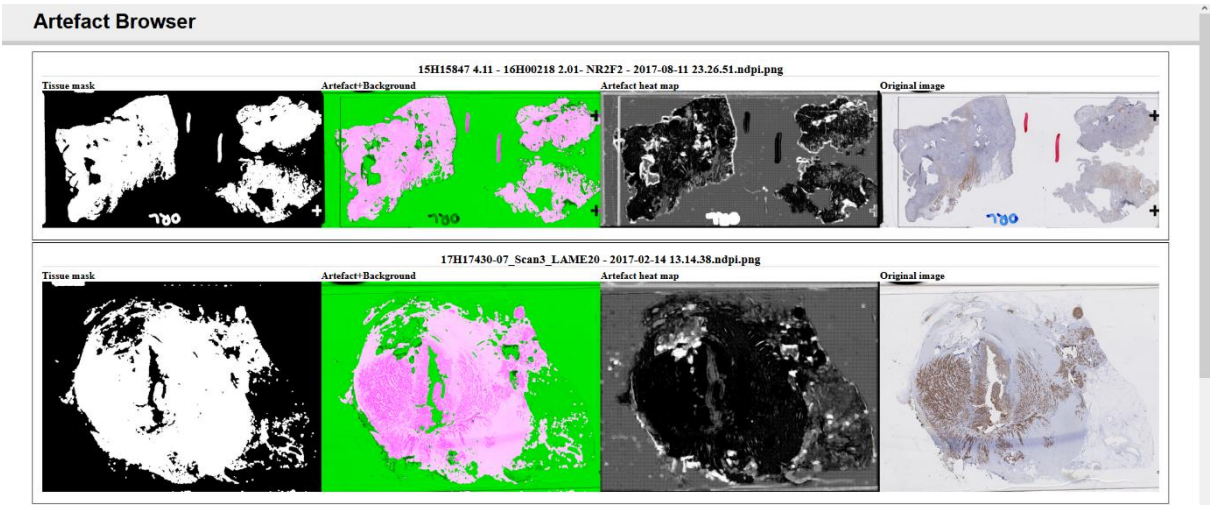



**Figure 6.16. Screenshot of the web interface developed for visualising the predictions of the artefact segmentation network.**

Such a system could be introduced into a pathology laboratory workflow to provide quick feedback on the quality of the acquisitions. Quantitative measurements can additionally be made based on, for instance, the ratio of artefactual tissue to normal tissue (excluding the background glass slide). Development and validation of the prototype was however postponed as the coronavirus crisis necessarily shifted the priorities of the pathology laboratory involved in the process.

## 6.4 Recent advances in artefact segmentation

Several recent software and publications have addressed the problem of artefact segmentation more recently. The PathProfiler open-source software[44] by Haghighat et al. [285] uses a U-Net network to separate tissue from the glass slide background, and a multi-label regression network with a ResNet-18 backbone to predict the quality of 256x256px patches extracted at 5x magnification. The scores predicted per patch are a general "usability" score, the probability of "no artefact" in the patch, and the quality of the patch with regards to staining issues, focus issues, folding artefacts and "other artefacts". Performances are evaluated based on the Pearson correlation coefficient between the algorithm and the expert on the overall usability score, and patch-level ROC-AUC based on expert annotations.

Deep learning methods have also been proposed more specifically for the detection of blurry regions, like Senaras et al.'s DeepFocus [298] which uses a very straightforward classification architecture similar to AlexNet. Swiderska-Chadaj et al. [299], meanwhile, train a U-Net network on "damaged tissue, which include "distortions, deformations, folds and tissue breaks" in Ki-67-stained brain tumour specimens. Their qualitative results seem good, but their evaluation is based only on annotated patches and do not include WSI predictions, making it harder to fully assess the quality of their final product.

The effect of artefacts on deep learning algorithms for digital pathology image analysis was analyzed by Schömig-Markiefka et al. [300]. They used different types of artificially generated artefacts (focus, compression, contrast, deformation, bad staining, dark spots…) and observed the decrease in performance of a tumour patch classification algorithm. They notably found that "substantial losses of accuracy can occur when the focus quality levels are still visually perceptible as adequate by pathologists", and that "any [JPEG] compression levels under 80% can result in accuracy deterioration". In general, they demonstrate that all artefacts "might result in accuracy deterioration and warrant quality control measures".

## 6.5 Discussion

From our experiments and the state-of-the-art, it seems relatively clear that the difficulty in artefact detection and segmentation is found not so much in the ability of deep neural networks to find relevant and accurate features, but rather in the very loose definition of the problem itself. This leads to a scarcity of good expert annotations, and to the impracticality of quantitative evaluations. Most of the proposed evaluation methodologies (including ours) rely on subjective assessment by experts of the "quality" of the slide, or of the predicted artefactual region.

These results, however, tend to show that automated segmentation of artefactual regions, either as a pre-processing step before running other image analysis tasks or as a quality control step just after the acquisition of the slide, is a good use case for deep learning methods. Our experiments show that even a limited supply of low-quality annotations (with all WSIs from the training set coming from the same block) results in a trained network with good generalization capabilities (to other organs and staining agents).

Tools such as HistoQC, PathProfiler, or our prototype application can easily be integrated into a digital pathology workflow, providing a quick automated analysis of the quality of the WSI to warn

[44] https://github.com/MaryamHaghighat/PathProfiler

against the necessity of re-staining or re-acquiring a slide, and to provide objective quality control metrics for a laboratory's assessment of the quality of their output.

The performance of HistoQC shows that even relatively simple methods can give good results for this application. The main drawback of that system, however, is the necessity to tweak the configuration depending on the nature of the slide to analyse. The deep learning methods such as PathProfiler's or ours, meanwhile, are easier to use out-of-the-box on new examples but require some time investment to annotate examples of the type of artefactual regions that are relevant to detect for the purposes of the laboratory. As the annotations do not have to be very precise, however, this time investment is more limited than in most "clinical" applications and can more easily be outsourced to non-experts based on a smaller set of expert examples.

# 7 Interobserver variability

A key characteristic of digital pathology tasks is the absence of a 100% reliable ground truth. Pathology is a domain where inter-expert disagreement is typically high. There have been many studies on the importance of interobserver variability for clinical practice, and many computer vision publications note and often measure in some way this disagreement for the particular dataset that they are working on, it is rarely, if ever, taken into account into the evaluation process. In this chapter, we will review the importance of interobserver variability in pathology tasks in general (section 7.1), then more particularly in the context of image analysis challenges and other image analysis publications (section 7.2). We will particularly look at how the "ground truth" of the dataset is determined to train and evaluate algorithms, and how researchers' knowledge about inter-expert disagreement is incorporated into the process.

After these analyses of the state-of-the-art, we will also take a closer look in section 7.4 at a specific digital pathology challenge: Gleason 2019. This challenge is the only image segmentation challenge in digital pathology to date that released individual expert annotations instead of a "consensus" annotation on their dataset, making it a very useful resource when exploring the effects of inter-expert agreement. Our analyses and experiments on the Gleason 2019 dataset in particular were presented at the SIPAIM 2021 conference and published in SPIE [5].

## 7.1 Pathology

### 7.1.1 Tumour grading

Inter-expert or interobserver agreement is a long-studies topic in pathology. Cancer grading often relies on a combination of criteria which can be highly subjective, such as the shape irregularity. Interobserver variability is an important factor in assessing the usefulness of cancer grading systems [301]. Indeed, a grading system where the criteria are too subjective and have a high degree of interobserver variability will necessarily be less trustworthy, and more dependent on the level of experience of the pathologist. The 1991 study establishing the Nottingham grading system for breast cancer [302], for instance, clearly states that their modification to the earlier "Bloom & Richardson" method aimed "to introduce greater objectivity," and they measure their success in part by the fact that "tumours […] graded independently by two experienced histopathologists obtain[ed] over 90% agreement on first assessment". They also note that "[r]eproducibility between different centres is more difficult to achieve."

Inter-expert agreement in cancer grading is generally measured as the "rate of agreement" ($R$ in Table 7.1), which is simply the percentage of cases where the two experts assigned the same grade, or with Cohen's kappa [229] (see Chapter 4, section 4.1.3). Cohen's kappa is an extremely popular agreement metric in medical studies (despite some important known weaknesses [233] discussed in section 4.3.2). While $R$ varies between 0 (no agreement) and 1 (perfect agreement), Cohen's kappa varies between -1 (complete disagreement) and 1 (perfect agreement), with 0 meaning that the agreement is equal to that expected from random chance.

As detailed in Chapter 4, section 4.1.3.2, Cohen's kappa can take different forms, depending on whether the categories used by the observers are ordered on not, and if ordered depending on how much "large" errors should be penalized. The "unweighted" kappa ($\kappa_U$ henceforth) penalizes all errors equivalently, while the linear ($\kappa_L$) and quadratic ($\kappa_Q$) kappa values penalize errors that

are "smaller" less harshly, so that $\kappa_Q < \kappa_L < \kappa_U$ in situations where most errors are between neighbouring categories.

Interobserver agreement does not just depend on the task, but also on the specificities of the dataset, the instructions given to the observers, the conditions of the test, etc. This makes a direct comparison of interobserver agreement between studies difficult, even when looking at the same underlying pathology task. For instance, in a meta-analysis of lung cancer interobserver variability studies, Paech et al. [303] find that even after reanalysing the results to ensure that all studies measure the same classes, rates of agreement vary between 77% and 94% (with the $\kappa_U$ ranging between 0.48 and 0.88). Such large variations in results can be also seen for other cancer types, such as ovarian cancer [304], [305] or breast cancer [306]–[309].

The reason for interobserver disagreement is often found either in subjective criteria, or in criteria that are difficult for humans to perceive objectively, such as relative proportions. As noted in Hernandez et al. when discussing their results [304], "Although observers A and B agreed as to what cell types were present, they did not always agree as to which cell type was more prevalent and representative of the tumor."

Meyer et al. also note regarding breast cancer grading that "[t]his and other studies show that while disagreements of one step (grade 1 vs grade 2, grade 2 vs grade 3) have been common, discrepancies of more than one grade seldom occur" [309]. It is interesting to note that, while this is obviously a very important criterion in assessing interobserver variability, many studies appear to use the $\kappa_U$ as their agreement metric, even though it is completely invariant to the degree of the disagreements.

### 7.1.2 Mitosis detection

Mitotic count is an important factor in tumour grading, particularly in the Nottingham system for breast cancer grading. Most of the studies on breast cancer grading therefore also report the agreement on mitotic grade. The mitotic grade is a grouping of the mitotic count in three categories, using thresholds that depend on the field area of the microscope [302]. Meyer et al. [309] report the results from several studies between 1982 and 2000, finding $\kappa_U$ values ranging from 0.36 to 0.70 for mitotic grade.

It is noteworthy that mitosis detection, as it is generally conceived as a *computer vision* task, will typically be evaluated on a per-mitosis basis, whereas the clinical relevance is mostly found in the categorized mitotic grades. The evaluation of computer vision algorithms should therefore at the very least discuss the potential clinical impact of their performance in addition to the classical image analysis metrics.

### 7.1.3 Gleason scoring

Several grading systems have been proposed for prostate cancer over the years, but by the mid to late 1990s the Gleason system had become the most widely used [239]. Part of the reason for its widespread adoption (and for later revisions) is its higher correlation with patient outcome compared to competing systems [310], another cited factor has been its relatively higher inter- and intraobserver agreement, making it more reliable and less dependent on a particular pathologist. As Özdamar et al. [311] state, "[t]hose items with poor agreement cannot be regarded as reliable indicators and cannot be used for making a decision", and only a system with similar intra- and interobserver variability "may be safely used by different pathologists".

The high subjectivity of the Gleason pattern grading, however, is made apparent in a 1997 study by McLean et al. [239], where they find that, without a prior "consensus meeting" between the pathologists to agree on the precise criteria they would use to assess those patterns, they obtained relatively poor rates of agreements (between 0.18 and 0.37 on three possible pairings of pathologists, with $\kappa_L$ ranging from 0.15 to 0.29).

Two large studies by Allsbrook et al. in 2001 have been largely cited since then, measuring interobserver agreement between 10 urologic pathologists [237], then measuring the performance of 41 general pathologists against the consensus of the urologic pathologists [241]. They found markedly higher agreement than in the McLean study, with pairwise $\kappa_L$ ranging between 0.48 and 0.84. General pathologists had an even wider range of results, with $2/41$ achieving almost perfect agreement with the "consensus" of urologic pathologists, yet $7/41$ fairing barely better than random chance, showing the importance of specialised training and experience on such a complex problem.

A major update to the Gleason system was made first in 2005, then in 2014 at the International Society of Urological Pathology Conference [138], leading to the "Epstein grouping" system. Some changes were made over the year in the exact definition of each pattern, but the Epstein system brings some major changes in the final categories. This major update largely simplifies the interpretation of the results (5 groups instead of 10 scores), and was validated as better correlated with patient outcome [136].

The effect on interobserver agreement, however, is less clear. A 2016 study by Ozkan et al. [312] found very little difference in agreement between the Gleason score ($\kappa = 0.43$) and the Epstein groups ($\kappa = 0.39$)[45]. The importance of which "kappa" is used is clearly shown if we try to compare the 2016 study with the recent 2020 study by Bulten et al. [313], which reports a $\kappa_Q = 0.71$ for two pathologists on 245 Tissue Micro Arrays samples for Epstein grouping. This seems like a much higher agreement than the 2016 study, but that interpretation depends on which kappa they actually used, as recomputing the kappa from Bulten et al. gives us $\kappa_L = 0.52$ and $\kappa_U = 0.31$, meaning that the agreement is either vastly improved, slightly improved or even slightly *worse*. As Bulten et al. provide the confusion matrix in their supplementary materials, it is easy to recompute the different values as needed, but the Ozkan et al. study gives little details on how they came to their numbers.

**Table 7.1. Selection of studies on interobserver variability in pathology. Findings in grey were recomputed based on the data available in the publication to facilitate comparisons between studies. $R$ is the rate of agreement, $\kappa_U$, $\kappa_L$ and $\kappa_Q$ are the unweighted, linear and quadratic kappa values.**

| Reference | Description of the experiment | Main findings |
|---|---|---|
| Hernandez, 1983 [304] | Two pathologists from different institutions asked for histologic type (5 possibilities) and histologic grade (4 possibilities) of 68 **ovarian tumours** from 34 patients. | $R = 0.60$ for histologic type, $R = 0.66$ for histologic grade ($\kappa_U = 0.365, \kappa_L = 0.511, \kappa_Q = 0.625$). |
| Gilchrist, 1985 [306] | Eleven surgical pathologists studied microscopic sections from 45 masectomy specimens of node | $R = 0.85$ pairwise for cell type ($\kappa_U = 0.32$). |

[45] The authors did not specify which version of the $\kappa$ was used.

| | | |
|---|---|---|
| | positive **breast cancer** patients. Asked for cell types (2) and nuclear grades (3). | $R = 0.54$ for nuclear grades ($\kappa_U = 0.21$). |
| Robbins, 1995 [307] | Consensus of 3 pathologists in an institution compared to consensus of 2 in another on 50 cases of **breast cancer** in B5 and BFS fixed tissue. Asked for cancer grade, tubule formation, nuclear size/pleomorphism and mitotic count (3 categories each). | $R = 0.74 - 0.83$ on nuclear grade ($\kappa_U = 0.58 - 0.73$, $\kappa_L = 0.65 - 0.79$, $\kappa_Q = 0.74 - 0.85$).<br>$R = 0.78 - 0.81$ on mitotic grade[46] ($\kappa_U = 0.65 - 0.69$, $\kappa_L = 0.70 - 0.77$, $\kappa_Q = 0.75 - 0.82$) |
| Özdamar, 1996 [311] | Two pathologists asked to grade **prostate cancer** using the WHO grading system (4 categories) and the Gleason scores (9 categories) in 96 specimens. | $R = 0.60$ (WHO)<br>$R = 0.71$ (Gleason) |
| McLean, 1997 [239] | Three urology pathologists asked to grade **prostate cancer** using the Gleason scores in 71 specimens. | Using all Gleason scores (5-9): $R = 0.18 - 0.37$, $\kappa_U = 0.03 - 0.17$, $\kappa_L = 0.15 - 0.29$, $\kappa_Q = 0.28 - 0.46$.<br>Regrouped in 2 categories[47]: $R = 0.51 - 0.79$, $\kappa = 0.15 - 0.29$[48]. |
| Allsbrook, 2001 [237], [241] | 41 general pathologists and 10 urologic pathologists asked to grade **prostate cancer** using the Gleason score in 38-46 cases. The study measured the agreement between urologic pathologists, and between general pathologists and the consensus of the urologic pathologists. | Urologic pathologists: $\kappa_U = 0.31 - 0.79$[49], $\kappa_L = 0.48 - 0.84$[50] pairwise agreement.<br>General pathologists (measured against the consensus of urologic pathologists): $\kappa_U = 0.44$ [0.0 − 0.88] |
| Meyer, 2005 [309] | Seven pathologists asked to grade **breast cancer** in five trials. Also reports several earlier studies. | $R = 0.67 - 0.74$ on tumour grade ($\kappa_U = 0.5 - 0.59$).<br>$R = 0.55 - 0.68$ on mitotic grade ($\kappa_U = 0.45 - 0.67$). |
| Malpica, 2007 [305] | Nine pathologists (incl. 2 non-specialists) asked for grading of 80 cases of **ovarian serous carcinoma** (in two rounds to also measure intraobserver agreement) | Pairwise $k_U = 0.717 - 1$ |

[46] The 3 categories are "$N \leq 7$", "$8 \leq N \leq 14$", "$N \geq 15$" in a 0.196mm$^2$ field from the edge of the tumour.
[47] Corresponding to $Score \leq 7$ and $Score \geq 8$.
[48] In a binary problem, $\kappa_U = \kappa_L = \kappa_Q$, so we only report a single $\kappa$ value.
[49] Measured after grouping in 4 score groups.
[50] Assumed to be $\kappa_L$, as it is only described as the "weighted kappa". Measured on all possible Gleason scores.

| | | |
|---|---|---|
| Paech, 2011 [303] | Meta-analysis of six studies on interobserver variability in **lung cancer** grading. | $R = 0.77 - 0.94$<br>$\kappa_U = 0.48 - 0.88$ |
| Davidson, 2019 [308] | 208 pathologists interpret 1 of 4 **breast cancer** biopsy sets including 5 or 6 invasive carcinomas, then interpret the same set 9 months later in a randomized order (for intraobserver agreement), either using a glass side or a WSI. | $\kappa_U = 0.36$ on Nottingham grade.<br>$\kappa_U = 0.40$ on Mitotic count. |
| Bulten, 2020 [313] | Two pathologists asked to grade **prostate cancer** TMA cores using Epstein groups. | $\kappa_U = 0.31$<br>$\kappa_L = 0.52$<br>$\kappa_Q = 0.71$ |

## 7.2 Computer vision

### 7.2.1 Comparing automated methods with pathologists

Interobserver variability has been a known problem for a long time in digital pathology. In fact, the introduction of computer vision algorithms to the digital pathology pipeline has often been justified in part by the need to provide more objective, quantitative assessments to reduce this interobserver (and intra-observer) variability in diagnosis [11]. Interobserver variability generally affects computer vision pipelines (and more particularly machine learning algorithms) in two aspects: in the generation of the ground truth (often through a consensus mechanism), and in the evaluation. Very often, for the evaluation, interobserver variability is only considered as a way to compare the performance of algorithms to the performance of pathologists, both evaluated against the "consensus ground truth".

This type of comparison, however, may be difficult to do fairly. As an example, Bauer et al. [314] compare the performances on a benign/malignant cancer classification task of a ResNet to a classical "handcrafted features" pipeline and to inter-pathologist accuracy. The inter-pathologist accuracy, however, is computed as the proportion of cells which receive the same label from two pathologists (in this case, 1272/1633, or 78%). Meanwhile, the performances of the algorithms are computed based on the 1272 cells where the pathologists were in agreement. This can obviously introduce a large bias, as the cases where the experts are in disagreement are more likely to be more difficult to correctly classify. In the same publication, a more meaningful comparison between the algorithms and the pathologists is also proposed, with both being tested on follow-up survival data.

A different approach is proposed in Turkki et al. [315] to evaluate an automated method for the quantification of tumour-infiltrating immune cells in breast cancer samples stained with H&E. The accuracy is first computed based on a collegial supervision by a group of experts aided by additional IHC images. Then, the results on another set of images are compared to two pathologists separately, and the pathologists are similarly compared with each other.

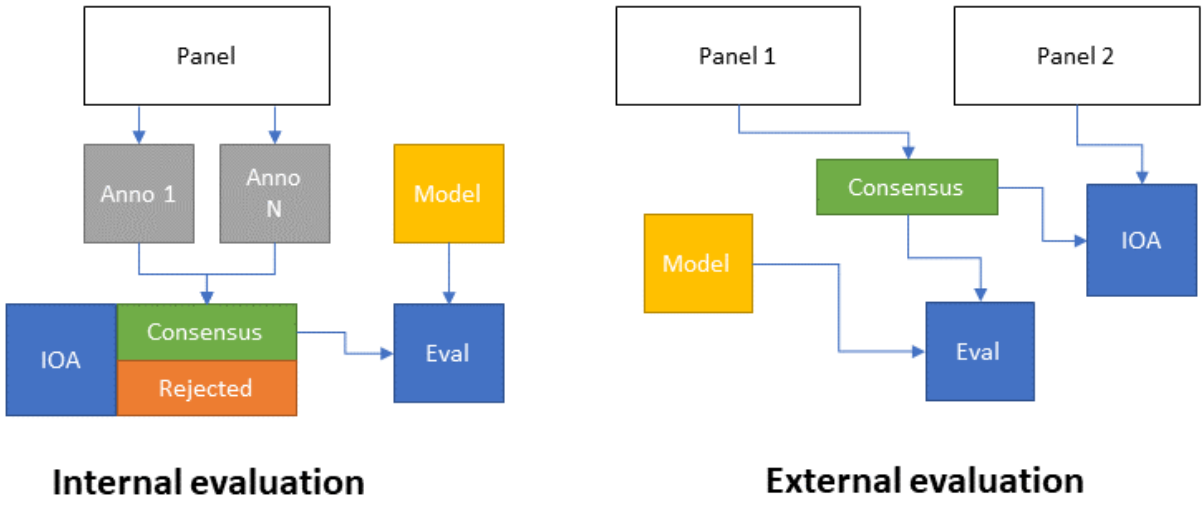



**Figure 7.1. Main approaches for the evaluation of interobserver agreement (IOA) for comparison with the performances of an algorithm: in the internal approach, the same panel of experts is used to create the dataset on which the model is evaluated, and to evaluate the IOA. In the external approach, a second panel is used to evaluate the IOA while being put in the same conditions as the algorithm.**

Malon et al. [316] studied inter-pathologists agreement for mitosis detection. To include an automated method in their comparison, they used the majority vote from three pathologists as a ground truth, and then evaluated each pathologist and the automated method against this ground truth. While this is certainly less biased than removing the contentious cases, it still poses a potential bias problem, as the pathologists in this case are evaluated on a ground truth that they participated in. Indeed, while the algorithms in this case has to agree with 2 of the 3 pathologists for a prediction to be counted as correct, a pathologist only has to agree with one of their two colleagues. This weakness was noted by the authors. One of the effects of this methodology was that it gave a large advantage to one of the pathologists in particular: as the other two were very often in disagreement, this pathologist typically acted as the "tie-breaker" and was therefore largely responsible for the determination of the "consensus".

A much stronger comparison (which also require significantly more effort and the involvement of more experts) is performed in Bulten et al. [313] for Gleason grading. The ground truth was determined by three pathologists specialised in urological pathology, first independently, then collegially for cases with disagreements. Fifteen additional pathologists with varying levels of experience (including two pathologists in training) and a deep learning algorithm were compared to the consensus. The automated method obtained a better quadratic kappa than the median of the pathologists, outperforming 10 out of 15.

Several deep learning competitions have evaluated interobserver variability in some way. In the AMIDA 2013 mitosis detection challenge [167], the ground truth was initially determined independently by two pathologists, with two additional pathologists reviewing the cases with a disagreement. In the challenge results, they observed that "many of the false positives produced by the top performing methods closely resemble mitotic figures", and subsequently performed a "re-annotation" of the false positives by two of the pathologists (one from the "first pass", one from the "reviewers"). About 30% of the false positives from the top method were re-annotated as true positives. The ground truth test dataset was also re-annotated, with only 71% of the objects

originally annotated as “mitosis” being relabelled as such. Clearly, such large uncertainty on the real “ground truth” label has the potential to influence the results of a challenge. The BACH 2018 challenge [176] created the “ground truth” with two pathologists which had access to information outside of the H&E images (such as IHC or other regions of tissue), and three other pathologists were tested while being given only the same information as the competing algorithms. A large panel of “external” pathologists was also used in the Camelyon 2016 challenge [170] for metastatic region segmentation in lymph nodes. The Gleason 2019 challenge will be more thoroughly discussed in section 7.4 below.

In general, we can therefore see two main approaches to the evaluation of interobserver agreement as a means of providing a comparison to algorithmic performances (as illustrated in Figure 7.1). In **internal evaluation**, the expert panel creates the dataset by first annotating it separately, then creating a consensus. At the same time, the interobserver variability is computed based on the individual annotations. In some cases, parts of the dataset where no consensus can be reached may be rejected, and the consensus data is used to evaluate the model. In **external evaluation**, an independent panel of experts is used for the evaluation of interobserver agreement. The main advantage of this approach is that it puts the evaluation of the experts on a “fair” ground with the model. Indeed, for the creation of the consensus dataset, experts can often use more information than what’s available to the model (as the goal is to have the best possible annotations). This fairness comes at the cost of requiring additional experts.

### 7.2.2 Consensus methods for ground truth generation in challenges

Even when the interobserver variability is not explicitly computed or reported, its existence always means that challenge organizers have to choose how to address it to create the “ground truth” annotations of their dataset.

In our study of segmentation challenges in digital pathology [8], a diversity of strategies was identified for the generation of “ground truth” annotations from multiple experts. The simple solution of using a single expert as ground truth was used by the PR in HIMA 2010, GlaS 2015 and ACDC@LungHP 2019 challenges, with the latter using a second expert to assess interobserver variability.

Several challenges use students (in pathology or in engineering) as their main source of supervision, under the supervision of an expert pathologist reviewing their work. This was the case for the Segmentation of Nuclei in Images challenges of 2017 and 2018 (no information was found for the earlier editions of the challenge), as well as for the MoNuSeg and MoNuSAC challenges.

The Seg-PC challenge in 2021 relied on a single expert for identifying nuclei of interests, then on an automated segmentation method which provided a noisy supervision [317], at least for the training set. It is unclear whether the same process was applied for the test set used for the final evaluation, as that information is not present in the publicly available documents of the challenge.

When using multiple experts, the processes vary and are sometimes left unclear. BACH and PAIP 2019 used one expert to make detailed annotations, and one to check or revise them. DigestPath 2019 and the 2020 and 2021 editions of PAIP all mention “pathologists" involved in the annotation process, but do not give details on how they interacted and came to a consensus. Gleason 2019 used six experts who annotated the images independently. A consensus ground truth was automatically generated from their annotation maps using the STAPLE algorithm [212].

BCSS, NuCLS and WSSS4LUAD used a larger cohort of experts and non-expert with varying degrees of experience, with more experienced experts reviewing the annotation of least experienced annotators until a consensus annotation was produced.

Conic 2022, meanwhile, uses an automated method to produce the initial segmentation, with pathologists reviewing and refining the results.

In general, we can identify the following common strategies, illustrated in Figure 7.2:

a) A **single expert** being considered as ground truth. In this case, the interobserver variability is unknown or needs to be evaluated separately (having another expert which has no impact on the ground truth produce their own annotations).
b) **Multiple experts working collegially** (either working on all cases together, or discussing the cases where there is a disagreement). The interobserver variability is also unknown in this case, as there is no "individual annotations" being produced.
c) **Multiple experts working independently on subsets** of the data. The big advantage of this approach is that it is easier to get more data without overworking a single expert. The dataset will also naturally include the diversity in biases and habits of individual experts. There may however be some induced biases if, for instance, some experts are over-represented in the test set.
d) **Automated method refined by expert(s)**. In this case, it is very easy to quickly get a lot of "noisy" data (using the best algorithms from the current state-of-the-art), with the expert only needed to provide corrections when needed. The resulting annotations may however remain noisy, particularly in segmentation tasks, as the limits of what an expert would consider an acceptable annotation may vary.
e) **Expert with senior review**. In this approach, the original annotations are provided by junior experts (one or several working on subsets of the data), and a senior expert checks their annotations. Some insight on the interobserver variability may in this case be extracted from the number of corrections that the senior expert had to make.
f) **Multiple experts producing individual annotations**. This approach makes it easy to evaluate the interobserver variability at the same time as the ground truth annotations are generated. Producing those ground truth annotations, however, requires the additional step of "merging" the individual annotations. This can either be done by an automated method (such as majority voting, or the previously mentioned STAPLE algorithm), or with the help of a senior expert which resolves conflicts.

An additional strategy, which bypasses the need for a "consensus", is to generate the supervision based on additional data. This could come from available information on patient outcome, or from using data from IHC as a supervision for an adjacent H&E-stained slide. The latter approach is used for instance in Turkki et al. [315], where consecutive slides were stained with H&E and the anti-CD45 antibody. The CD45 expression was then used to annotate the H&E slide with "immune cell-rich" and "immune cell-poor" regions. An example of such technique is illustrated in Figure 7.3 using images from our "Artefact" dataset, enabling the identification of epithelial cells (using anti-pan-cytokeratin IHC). The difficulty of such technique is that it requires a correct registration of the IHC image onto the H&E image, as there will always be some deformations, potential tissue damage, different artefacts, etc., between the two. This registration task was the target of the ANHIR challenge in 2019[51].

[51] https://anhir.grand-challenge.org

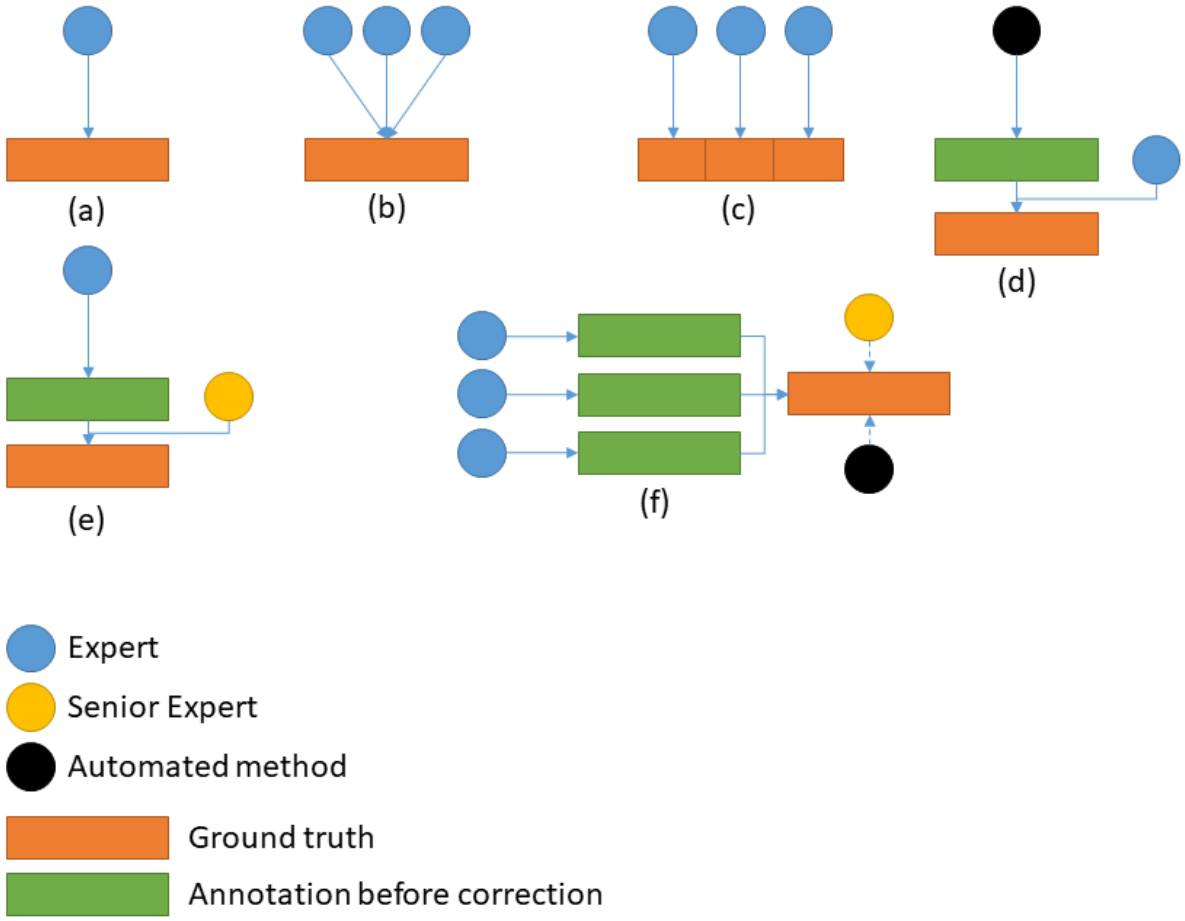



**Figure 7.2. Workflows for generating ground truth annotations in digital pathology datasets. (a) Single expert, (b) Multiple experts working collegially, (c) Multiple experts working independently on subsets, (d) Automated method refined by expert, (e) Expert with senior review, (f) Multiple experts producing individual annotations, automated (or senior expert) consensus.**

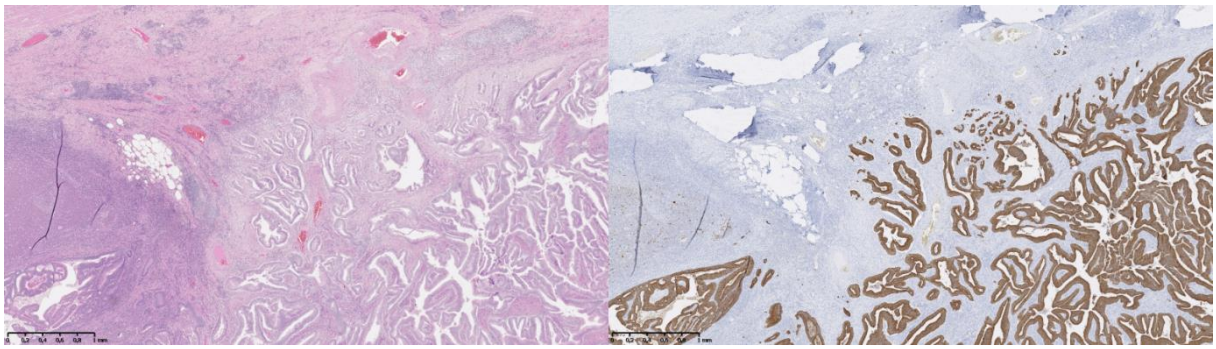


**Figure 7.3. Regions of a WSI from two adjacent slides extracted from the same tissue block, stained with (left) H&E and (right) anti-pan-cytokeratin with haematoxylin counterstain.**

### 7.2.3 Influence on training deep learning models

Just as there are many different approaches to creating a "consensus" ground truth from annotations by multiple experts, several options are available for training deep learning models with these annotations. In most of the ground truth generation strategies, only the consensus annotations set is available in the end, leaving no choice but to train the models based on this "ground truth". Interobserver variability, in that case, has to be considered as "noisy labels", as in the previous chapter. When annotations from multiple experts are available, however, more options are available.

In digital pathology, the Gleason 2019 dataset is the only publicly available segmentation dataset to have released those individual annotations. The team from the University of British Columbia that produced the dataset tested several strategies in publications made before the challenge [157], [318], [319]. In a first study [157], a "probabilistic approach", based on a method by Raykar et al. [320], in which the classifier is trained simultaneously to the "consensus label" and the annotators' accuracy using an Expectation Maximization algorithm. In another study [319], pixel-wise majority voting is used as the ground truth label. The results are then compared with the same algorithm trained and tested using ground truth labels created with the STAPLE algorithm. They report results that are marginally better (but "very close") with the STAPLE than the majority vote. In a third publication [318], they compare training on a single expert (and testing on each expert separately) with training and testing on the majority vote. They show that, unsurprisingly, the algorithm performs better when tested on the same pathologist as it was trained on. They also obtain better results using the majority vote labels. However, they do not test using individual annotations from all experts versus using the majority vote labels. Lastly, Karimi et al. [85] compared training on single pathologists, the majority vote labels, and the STAPLE labels, all on a test set ground truth computed with the STAPLE algorithm. Results with majority vote and STAPLE were similar (and better then a single pathologist). They also tested using the separate annotations from three pathologists for training, and the STAPLE consensus of the three others for testing. While these results are not directly comparable (since the ground truth labels are not the same as in the other experiments), they point to similar performances than the STAPLE and majority vote training.

Post-challenge publications from other teams of researchers using the publicly available dataset have also used different strategies. Khani et al. [321] and Jin et al. [322] use the STAPLE labels for training and testing. Ciga et al. [323] and Yang et al. [324] both choose to only use the annotations from a single expert. Zhang et al. [325] merge the annotations to create a per-pixel "grade probability" map (so that if, for instance, three experts predict grade 3 and three experts predict grade 4, the pixel will be associated with a probability vector with p=0.5 for the grade 3 and 4 labels). Xiao et al. [326] use all individual annotations with a custom loss function that give different weights to each expert and each pixel in the images, and additionally weight the pixels based on their "roughness" in the image (defined as the absolute difference between the image before and after a Gaussian filter). They do not specify which "ground truth" they use for their test set, although their illustrations are consistent with either a STAPLE consensus or a majority vote.

In his 2021 Master Thesis at the LISA laboratory [327], Alain Zheng studied the effect of training a deep neural network either on all six separate expert annotation sets, or on the STAPLE consensus, with the test set ground truth always being set to the STAPLE consensus. The results from this Master Thesis show that better predictive performances on the image-level Epstein groups (which are more clinically relevant than the per-pixel predictions) were obtained by the

network trained on the separate annotation sets, but that the difference is only significant (according to a McNemar test) if it is combined with a filter that only take into account the pixels where the highest class probability $p_1$ and the second-highest class probability $p_2$ verify $P_1 > 2 \times P_2$.

## 7.3 Evaluation from multiple experts

While many different strategies are used to generate the ground truth and to use the multi-expert annotations for training, the evaluation of the algorithms is always done on a single reference "ground truth", typically the result of a consensus process. It is however not necessarily the only way to use annotations from multiple experts when they are available. In this section, we study possible alternatives that may provide better insights on the performances of the evaluated method.

In a multi-expert dataset, each image $i$ will be associated with a set of annotations $y_{ij}, j \in [1, E_i]$ with $E_i$ the number of experts who annotated that particular image (this number may vary between images, as not all experts will necessarily annotate all images). The typical process for using these annotations is to first apply a consensus mechanism to obtain a reference ground truth $t_i = C(\{y_{ij}\}_j)$, where the "consensus" function $C$ may be a mathematical process (such as the STAPLE algorithm or the majority vote) or a human process (such as a discussion between the experts to resolve the issue, or a decision by a senior expert). Then, the score per image of a prediction $p_i$ is computed using a metric $M$ as:

$$S_i = M(p_i, t_i)$$

Statistics based on these scores per image can then be computed, such as the average score on the $N$ images, which is usually the "summary statistic" from which algorithms being compared are ranked:

$$aS = \frac{1}{N}\sum_i S_i$$

Another approach, however, is to compute the scores independently for each expert annotation, so as to obtain a "per-image, per-expert" score $S_{ij}$:

$$S_{ij} = M(p_i, y_{ij})$$

Several options are then available. The "consensus" score can be replaced by the average of the per-expert scores $S_i^*$:

$$S_i^* = \frac{1}{E_i}\sum_j S_{ij}$$

Which can then be used similarly to $S_i$, for instance to compute the average score on the dataset:

$$aS^* = \frac{1}{N}\sum_i S_i^*$$

With this approach, another set of information can give some interesting insights on the algorithm's behaviour: the degree to which the scores are different depending on which expert

provides the annotation. For instance, the per-image standard deviation, or more realistically the range (as there will generally not be many experts per image):

$$Range_i = \max_j S_{ij} - \min_j S_{ij}$$

To fully explore the multi-expert aspect of the annotations, however, another strategy is to compare the summary statistics on the dataset per-expert directly, without first computing a per-image average. In that case, a "per-expert" average score $S_j^+$ can for instance be computed:

$$S_j^+ = \frac{1}{N}\sum_i S_{ij}$$

If all experts annotated all images (so that $E_i = E$ is constant), a single average performance can be obtained as:

$$aS^+ = \frac{1}{E}\sum_j S_j^+$$

Otherwise, the per-expert scores $S_j^+$ should really be considered as separate measures.

Approaching the evaluation from a multi-expert perspective can allow us to make a distinction between algorithms whose errors (relative to the consensus) tend to follow the biases of some of the experts in the dataset, from algorithms whose errors are due to other factors. As the error becomes a set of dissimilarities to different experts, it also becomes possible to use alternative methods to visualize the results, such as the Multi-Dimensional Scaling (MDS) that we previously used to look at the dissimilarity between metrics. In a multi-expert evaluation setup, MDS can be used to visualize at the same time the interobserver variability (how dissimilar the experts are to each other) and the performance of the algorithm (how dissimilar the algorithm is to each of the experts).

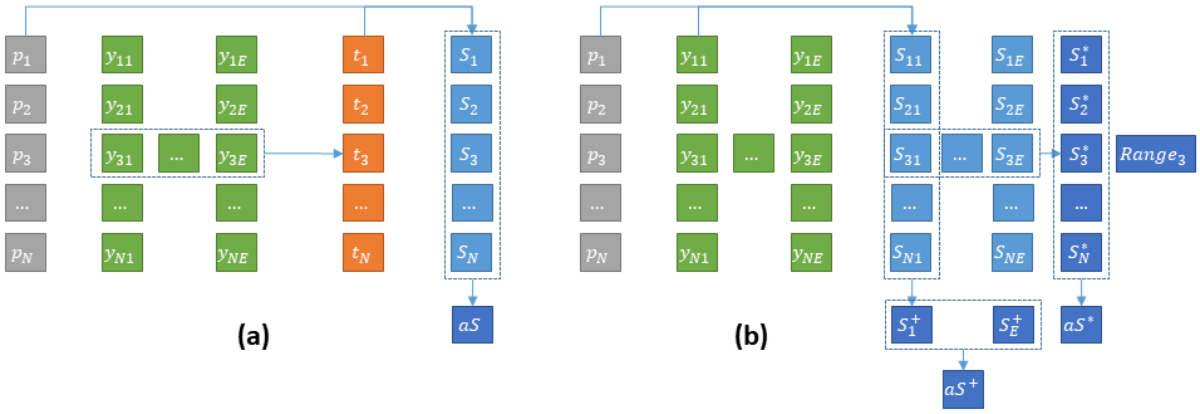



**Figure 7.4. Strategies to evaluate a multi-expert dataset. (a) Based on a reference "consensus" ground truth, (b) computing the scores independently for each expert on each image. In these simplified diagrams, the number of experts is constant for each image, but this is not necessarily the case.**

## 7.4 Insights from the Gleason 2019 challenge

In our 2021 SIPAIM publication [5], we further explored the Gleason 2019 dataset and the impact of multi-expert annotations. In this section, we present those results, with some improvements to make our results more relevant to the clinical perspective.

The Gleason 2019 dataset uses "Tissue Microarray" (TMA) slide, which contain many small tissue samples (the "cores") assembled in a grid pattern on a single slide [328]. In the dataset, individual cores are extracted from those slides and presented as separate images.

Using the publicly available training set of 244 cores, we studied several aspects:

a) How are inter-expert agreement and agreement to the consensus affected by the Gleason score computation method on each image.
b) How these agreement levels are affected by the consensus method.
c) How the comparison between an algorithm and the experts against the consensus may be affected by the participation of the experts in the consensus.
d) What insights may be gained from a visualisation method such as MDS.

Another aspect that was explored in this publication was the presence of clear mistakes in the annotation maps. This will be left for the next chapter on quality control.

### 7.4.1 Effect of the Gleason score computation method

The first step in the Gleason grading system is to identify the "Gleason patterns" that are present in the image (see Figure 7.5). The patterns are graded from 1 to 5. The annotations in the Gleason 2019 dataset are segmented glandular regions with their associated Gleason pattern. According to the challenge definition, "[t]he final Gleason score is reported as the sum of the most prominent and second most prominent patterns; e.g., a tissue with the most prominent pattern of Gleason grade of 4 and the second most prominent pattern of Gleason grade of 3 will have a Gleason score of 4+3"[52] (if a single pattern is present, the value is doubled, e.g. 4+4). This appears to follow the 2005 ISUP guidelines [329], where the final score is therefore on a scale from 2 to 10. These guidelines were revised at the 2014 ISUP conference [138]. According to these new guidelines, Gleason pattern 1 and 2 are removed. The first group (after "benign" tumours) is therefore constituted by tumours where only pattern 3 is present (3+3 = 6 in the Gleason system). The two next groups are defined are "3+4" (predominant pattern 3 with some pattern 4 present), "4+3" (predominant pattern 4 with some pattern 3), which were previously merged as "score 7". Finally, we have a "score 8" group and a "score 9-10" group. The scores are thus replaced by "groups" (hereafter "Epstein groups"):

- Group 1 = Gleason scores <= 6
- Group 2 = Gleason patterns 3+4
- Group 3 = Gleason patterns 4+3
- Group 4 = Gleason score 8
- Group 5 = Gleason score > 8

[52] https://gleason2019.grand-challenge.org/Home/, June 17th, 2022.

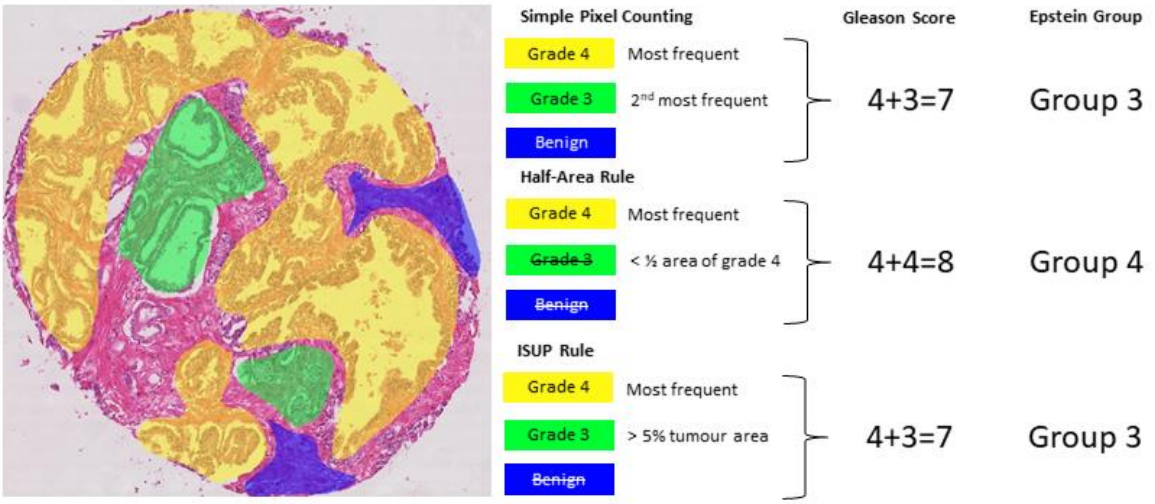



**Figure 7.5. Impact of the "simple pixel counting", "half-area rule" and "ISUP rule" on the determination of the Gleason Score and the Epstein group.**

Identifying what counts as "predominant patterns" is not trivial either. For instance, the 2005 guidelines specify that "in the setting of high-grade cancer one should ignore lower-grade patterns if they occupy less than 5% of the area of the tumor" [329, p. 1236]. However, higher-grade patterns should be considered even if they occupy a very small area. This, however, may not necessarily be adapted to *automated* methods of identifying the Gleason patterns, as it is more likely to have "noisy" results with very small pockets of badly identified patterns influencing the results.

In our analysis, we look at the inter-expert agreement measured by the unweighted and quadratic weighted Cohen's kappa ($\kappa_U, \kappa_Q$) for all inter-expert agreements using the Gleason scores (ranging between 2-10) and the Epstein groups (ranging from 1-5). Two rules were tested to identify the two "predominant patterns". First, a **simple pixel counting (SP)** rule consists in identifying and summing the two most frequent grades in terms of pixels, except if only one grade is present which is then doubled (e.g. a single grade 4 gives a score of 4+4). In our SIPAIM publication, we also defined a "half-area" rule, according to which the second most frequent grade is considered only if the area with this grade is at least half of the area occupied by the first most frequent grade. This latter rule aimed to give more weight to the majority grade and to avoid possible small "contaminations" due to segmentation errors. However, this rule is too aggressive at removing "smaller" regions compared to what pathologists would typically do (particularly in the presence of small regions of high-grade patterns). We therefore replace this rule here to follow the **ISUP** guidelines more closely and consider the secondary pattern either if it has a higher grade than the primary pattern, or if it has a smaller grade *and* occupies at least 5% of the annotated tumour regions. Our SIPAIM publication also used the "grade 1" pattern annotation in the score and group computations (i.e. a core with predominant grade 4 and some grade 1 would lead to a score a 5 and to Epstein group 1). However, the label "1" in the annotations corresponds to benign tissue, so only grades 3 to 5 should be considered. For these reasons, the tables presented here have different values than those presented in the original publication [5]. The different scoring methods are illustrated in Figure 7.5.

**Table 7.2. Variations in inter-expert agreement evaluation. The kappas are averaged from all head-to-head comparisons and shown alongside the value range in parenthesis, and the ranking in square brackets. G = Gleason grouping; E = Epstein grouping; SP = simple pixel counting; ISUP = "5%" rule. For each method an average is computed by weighting the experts' kappas by the number of their annotated maps. The highest average score for each expert is bolded, the highest average score for each method is shown in italic. N = number of annotation maps considered for each expert.**

| $\kappa_U$ **Expert (N)** | **G-SP** | **G-ISUP** | **E-SP** | **E-ISUP** |
|---|---|---|---|---|
| Expert 1 (237) | .36 (.32-.39) [6] | **.37 (.30-.38) [6]** | .34 (.25-.36) [6] | .34 (.25-.36) [6] |
| Expert 2 (20) | **.43 (.36-.48) [4]** | **.43 (.30-.67) [4]** | .37 (.25-1.) [5] | .37 (.25-1.) [5] |
| Expert 3 (186) | *.51 (.39-.59) [1]* | ***.52 (.38-.60) [1]*** | *.47 (.34-.58) [1]* | *.48 (.34-.59) [1]* |
| Expert 4 (235) | .45 (.30-.55) [3] | **.46 (.32-.57) [3]** | .44 (.30-.53) [3] | .45 (.32-.56) [3] |
| Expert 5 (244) | .46 (.34-.59) [2] | **.47 (.36-.60) [2]** | .45 (.32-.58) [2] | .46 (.33-.59) [2] |
| Expert 6 (64) | .38 (.30-.47) [5] | **.41 (.32-.67) [5]** | .39 (.30-1.) [4] | **.41 (.31-1.) [4]** |
| Average | .44 | **.45** | .42 | .43 |
| $\kappa_Q$ **Expert (N)** | **G-SP** | **G-ISUP** | **E-SP** | **E-ISUP** |
| Expert 1 (237) | **.71 (.41-.92) [5]** | **.71 (.41-.92) [5]** | .54 (.48-.57) [6] | .55 (.48-.57) [6] |
| Expert 2 (20) | ***.88 (.76-.96) [1]*** | ***.88 (.76-.95) [1]*** | *.74 (.48-1.) [1]* | *.75 (.48-1.) [1]* |
| Expert 3 (186) | **.79 (.59-.96) [2]** | **.79 (.60-.95) [2]** | .72 (.57-.86) [2] | .73 (.57-.86) [2] |
| Expert 4 (235) | **.74 (.70-.82) [3]** | **.74 (.71-.82) [3]** | .68 (.52-.76) [5] | .68 (.53-.77) [5] |
| Expert 5 (244) | .72 (.45-.87) [4] | **.73 (.45-.88) [4]** | .72 (.56-.89) [2] | .72 (.56-.92) [3] |
| Expert 6 (64) | .54 (.41-.89) [6] | .55 (.41-.95) [6] | .71 (.53-1.) [4] | **.72 (.53-1.) [3]** |
| Average | **.73** | **.73** | .67 | .67 |

Our experiment (see Table 7.2) shows to what extent the results vary depending on the scoring method. On average, the unweighted kappa shows moderate (0.4-0.6) agreement between experts, whereas the quadratic weighted kappa shows substantial (0.6-0.8) agreement, indicating that most of the disagreement happens between scores or groups that are close from each other. The individual head-to-head comparisons (see ranges) using the unweighted kappa vary from moderate (0.2-0.4) to substantial (0.6-0.8), with a perfect (1.) agreement between experts 2 and 6 in the case of the Epstein groups. However, these two experts have very few annotation maps in common and their head-to-head comparison only concerns 4 maps. The quadratic kappa again shows superior values, with several experts having "almost perfect agreement" (0.8-1.).

The ISUP rule also leads to a very slightly higher average agreement with the $\kappa_U$, as disagreements on very small glands are no longer taken into account, but most often the SP and ISUP rules lead to the same score. We should note that this particular heuristic is probably better suited for algorithms in particular (which are more subject to noisy results) than for annotations from pathologists, who can take small tissue regions into account when they estimate the Gleason or Epstein groups of a sample (which is generally larger than a tissue core in practice). It is also interesting to note that a similar ranking of experts is maintained regardless of the scoring method used. It is also interesting to note that Expert 1 is the "least in agreement with the others" in almost all scenarios and in both metrics.

**Table 7.3. Agreement of the experts' scoring with the consensus based on either STAPLE (ST), Majority Vote (MV) or Weighted Vote (WV). The scoring is computed based on the Epstein grouping with the ISUP rule. The ranks of the experts compared with the consensus are shown in parenthesis. Bolded values correspond to the best score for each expert.**

| $\kappa_U$ **Expert (N)** | **ST** | **MV** | **WV** |
|---|---|---|---|
| Expert 1 (237) | 0.470 (6) | **0.472 (6)** | 0.427 (6) |
| Expert 2 (20) | 0.605 (5) | **0.686 (2)** | **0.686 (4)** |
| Expert 3 (186) | 0.673 (3) | 0.655 (4) | **0.702 (2)** |
| Expert 4 (235) | 0.614 (4) | **0.637 (5)** | 0.587 (5) |
| Expert 5 (244) | 0.680 (2) | 0.710 (1) | **0.739 (1)** |
| Expert 6 (64) | 0.691 (1) | 0.672 (3) | **0.693 (3)** |
| $\kappa_Q$ **Expert (N)** | **ST** | **MV** | **WV** |
| Expert 1 (237) | **0.646 (6)** | 0.644 (6) | 0.622 (6) |
| Expert 2 (20) | **0.958 (1)** | 0.940 (1) | 0.940 (1) |
| Expert 3 (186) | 0.889 (2) | 0.894 (2) | **0.902 (3)** |
| Expert 4 (235) | 0.848 (5) | **0.852 (5)** | 0.835 (5) |
| Expert 5 (244) | 0.871 (4) | 0.883 (5) | **0.904 (2)** |
| Expert 6 (64) | 0.883 (3) | 0.891 (3) | **0.894 (4)** |

### 7.4.2 Impact of consensus method

We similarly computed the $\kappa_U$ and $\kappa_Q$ to evaluate the expert's agreements with the "ground truth" of the challenge, i.e. the **STAPLE** consensus. The Epstein grouping and the ISUP rule were used to compute the core-level groups for both the expert's annotation maps and the consensus maps, as it corresponds to the method closest to current guidelines for pathologists. As noted above, some publications on the Gleason 2019 dataset used a simple "**majority vote**" (**MV**) as a consensus method, which we also computed to investigate its effect on the agreement. We also computed a "**weighted vote**" (**WV**) which weights each expert by its average head-to-head agreement with the other experts (computed with the unweighted kappa).

The results are shown in Table 7.3. The unweighted kappa values range from 0.470 to 0.691 (moderate to substantial agreement) between each expert and the STAPLE consensus. The majority and weighted votes have similar ranges, with slightly different rankings. Unsurprisingly, Expert 1 is consistently worse in agreement with the consensus, as they were also the last in the average head-to-head agreement measures. Using the quadratic kappa changes the rankings of the expert and the range of values. The values compared to the STAPLE range from 0.646 to 0.958 (substantial to almost perfect agreement).

The agreement between the consensus methods (see Table 7.4) is high, although they clearly do not produce identical labels (even at the core level). Once again, the quadratic kappa gives higher values than the unweighted kappa, showing that the disagreements tend to be between adjacent classes. Figure 7.6 illustrates the different annotations provided by the experts, the result of the consensus methods, and the difference that the interobserver variability makes in the computation of the core-level scores depending on the scoring method.

**Table 7.4. Agreement between the consensus methods, based on the E-ISUP core level scores.**

| $\kappa_U$ Consensus | ST | MV | WV |
|---|---|---|---|
| **ST** | 1. | 0.915 | 0.871 |
| **MV** | 0.915 | 1. | 0.911 |
| **WV** | 0.871 | 0.911 | 1. |
| $\kappa_Q$ **Consensus** | **ST** | **MV** | **WV** |
| **ST** | 1. | 0.983 | 0.970 |
| **MV** | 0.983 | 1. | 0.978 |
| **WV** | 0.970 | 0.978 | 1. |

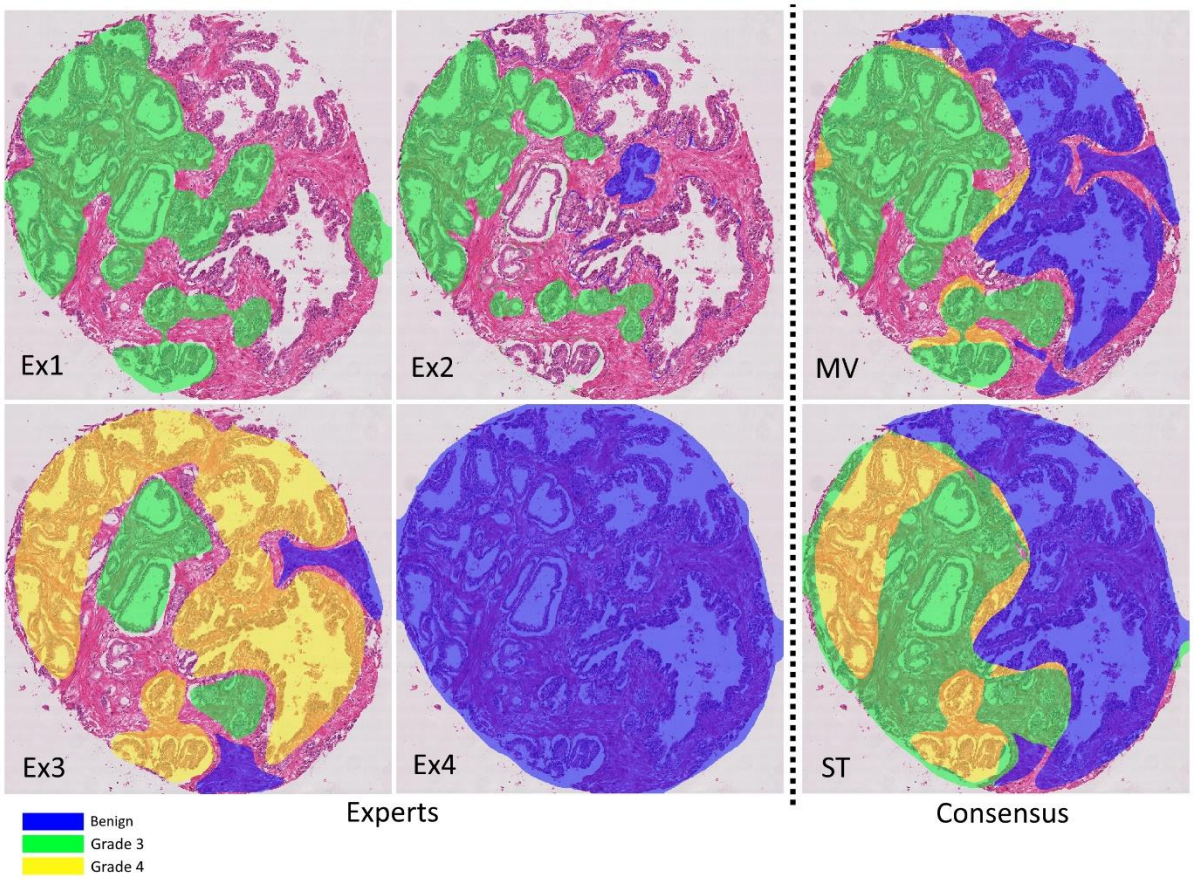



**Figure 7.6. Six annotation maps for the same core sample: four experts, two consensus methods (Majority Vote (MV) and STAPLE (ST)). The second expert shows clear mistakes in the annotation with contours (mostly blue and a bit of green) which were not properly closed. With the ISUP rule and the simple pixel counting rule, the core-level scores are 6,6,7,0 (experts) and 7 (both consensus), and the Epstein groups are 1, 1, 3, 1 (experts) and 3 (both consensus).**

**Table 7.5. Agreement between expert and the "Leave-One-Out" (LoO) consensus (i.e. consensus made from all other experts). In parenthesis is the difference between the LoO score and the corresponding score with the consensus that includes the expert. All results are based on the E-ISUP core level scores. Rankings per consensus method are shown in square brackets.**

| $\kappa_U$ **Expert (N)** | **LoO-ST** | **LoO-MV** | **LoO-WV** |
|---|---|---|---|
| Expert 1 (237) | 0.367 (-0.103) [6] | 0.344 (-0.129) [6] | 0.351 (-0.076) [6] |
| Expert 2 (20) | 0.487 (-0.118) [4] | 0.489 (-0.197) [3] | 0.556 (-0.131) [1] |
| Expert 3 (186) | 0.545 (-0.128) [1] | 0.565 (-0.090) [1] | 0.533 (-0.169) [3] |
| Expert 4 (235) | 0.488 (-0.126) [3] | 0.478 (-0.158) [4] | 0.468 (-0.119) [4] |
| Expert 5 (244) | 0.504 (-0.176) [2] | 0.542 (-0.168) [2] | 0.534 (-0.205) [2] |
| Expert 6 (64) | 0.483 (-0.209) [5] | 0.470 (-0.202) [5] | 0.448 (-0.246) [5] |
| $\kappa_Q$ **Expert (N)** | **LoO-ST** | **LoO-MV** | **LoO-WV** |
| Expert 1 (237) | 0.550 (-0.096) [6] | 0.532 (-0.112) [6] | 0.567 (-0.055) [6] |
| Expert 2 (20) | 0.936 (-0.023) [1] | 0.906 (-0.034) [1] | 0.919 (-0.021) [1] |
| Expert 3 (186) | 0.830 (-0.058) [2] | 0.836 (-0.058) [2] | 0.809 (-0.093) [2] |
| Expert 4 (235) | 0.757 (-0.091) [4] | 0.756 (-0.095) [5] | 0.765 (-0.070) [5] |
| Expert 5 (244) | 0.748 (-0.123) [5] | 0.788 (-0.095) [3] | 0.808 (-0.096) [4] |
| Expert 6 (64) | 0.807 (-0.076) [3] | 0.825 (-0.065) [4] | 0.809 (-0.084) [2] |

These data provide performance ranges to be achieved by the algorithms to produce results equivalent to those of experts. However, the comparison with algorithm performance would not be fair because each expert participated in the consensus. To correct this evaluation and show how that may affect the perception of the results, we compute the agreement levels with a "leave-one-out" strategy consisting in evaluating each expert against the consensus (STAPLE, majority vote and weighted vote) computed on the annotations of all the others.

Our results (Table 7.5) show that all experts, as expected, have a lower agreement with the ground truth when their own annotations are excluded from the consensus. However, not all experts are affected in the same way, and the rankings are not preserved. This shows that the comparisons to a consensus may not reflect the complexity of the relationships between experts.

### 7.4.3 Visualizing expert and consensus method agreement

The results shown in the previous tables are averages computed from the head-to-head comparisons between experts, and from comparisons between experts and the consensus "ground truth" obtained with the different scoring methods. MDS allows us to clarify how the experts relate to each other and to the consensus methods, and these latter between them.

An MDS visualization of the dissimilarity between all experts and consensus methods (of all experts, i.e. not the LoO consensus) is shown in Figure 7.8. The circles are used to represent the average "projection error" of the visualization (i.e. the average difference between the dissimilarity between two measures, reported in Figure 7.7, and the Euclidian distance between the points that represent them). The three consensus methods are very close to each other. Experts 2 and 6 are "identical" in their head-to-head comparison, but their annotations only overlap on a few examples, so their distance to all other points are very different (hence the

relatively large circles in the MDS plot). Between the four experts that provided more annotations (1, 3, 4 and 5), Experts 3 and 5 are often in agreement, but Expert 1 clearly appears as an outlier. This latter point is interesting to note because both of the post-challenge publications that used a single expert as their ground truth [323], [324] used Expert 1, indicating that the dataset they used may not be fully representative of a typical pathologist. The lower agreement of Expert 1 with their colleagues was reported in the original publication by Nir et al. [318], but it was computed on "patch-wise grading" (i.e. grading of the pattern present in small patches within each core) instead of the core-level score, leading to a much smaller apparent difference between the experts. As we will more thoroughly explore in Chapter 8, the dataset used in Nir et al. and the subsequent publications by their team are also not exactly the same as the publicly available dataset, making it difficult to compare their results with post-challenge independent publications.

| | E1 | E2 | E3 | E4 | E5 | E6 | ST | MV | WV |
|---|---|---|---|---|---|---|---|---|---|
| E1 | 0 | 0,52 | 0,43 | 0,47 | 0,44 | 0,47 | 0,35 | 0,36 | 0,38 |
| E2 | 0,52 | 0 | 0,14 | 0,33 | 0,08 | 0 | 0,04 | 0,06 | 0,06 |
| E3 | 0,43 | 0,14 | 0 | 0,23 | 0,16 | 0,26 | 0,11 | 0,11 | 0,1 |
| E4 | 0,47 | 0,33 | 0,23 | 0 | 0,25 | 0,24 | 0,15 | 0,15 | 0,17 |
| E5 | 0,44 | 0,08 | 0,16 | 0,25 | 0 | 0,18 | 0,13 | 0,12 | 0,1 |
| E6 | 0,47 | 0 | 0,26 | 0,24 | 0,18 | 0 | 0,12 | 0,11 | 0,11 |
| ST | 0,35 | 0,04 | 0,11 | 0,15 | 0,13 | 0,12 | 0 | 0,02 | 0,03 |
| MV | 0,36 | 0,06 | 0,11 | 0,15 | 0,12 | 0,11 | 0,02 | 0 | 0,02 |
| WV | 0,38 | 0,06 | 0,1 | 0,17 | 0,1 | 0,11 | 0,03 | 0,02 | 0 |

**Figure 7.7. Dissimilarity matrix used to create the MDS visualization in Figure 7.8. The dissimilarity is simply defined as $1 - \kappa_Q$. Results computed from E-ISUP scores.**

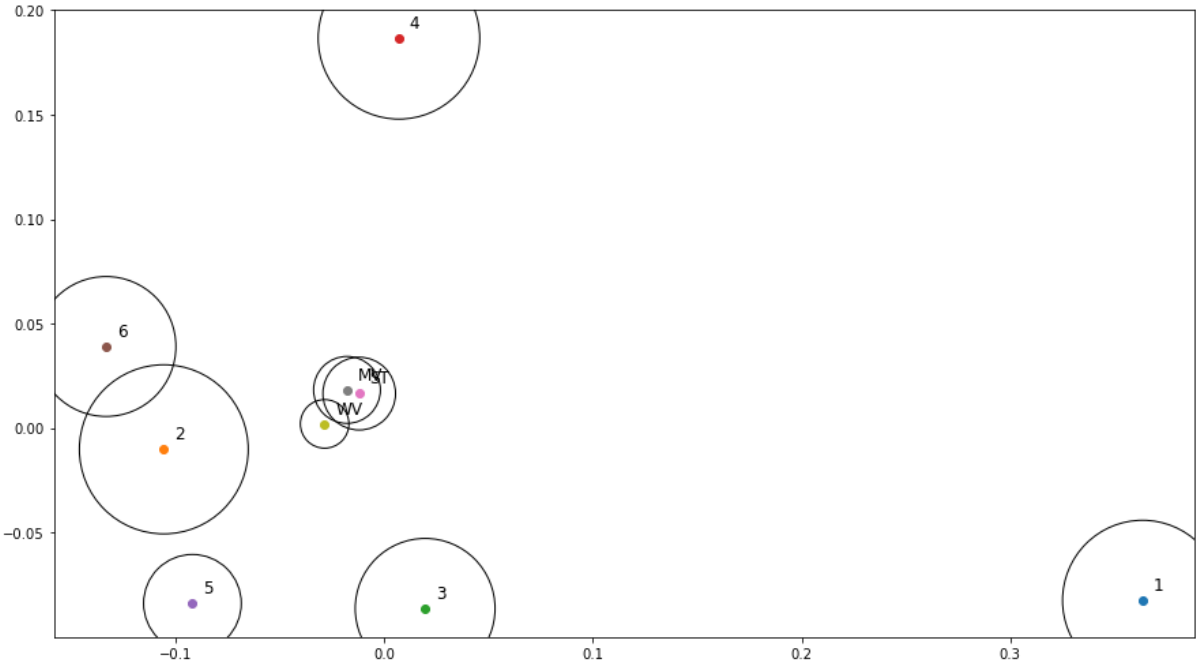



**Figure 7.8. MDS visualisation of the interobserver agreement in the Gleason 2019 dataset (as well as the agreement between experts and the consensus) using the Epstein groups and the ISUP rules. The circles represent the average "projection error" of the visualisation.**

## 7.5 Discussion

Interobserver variability is an important factor in exploiting digital pathology datasets. When the image analysis task is to reproduce the diagnosis of a human expert, it implies that there is no singular "ground truth" to train machine learning algorithms on, or to evaluate against. This ground truth can be approached through consensus mechanisms, particularly one where pathologists are able to communicate and discuss their difference in opinion. Such a process, however, requires a large amount of time and effort, and therefore makes it very difficult to gather large datasets. Most existing public datasets from digital pathology competitions involve either a single expert, or a small number of experts, leading to a "consensus" that is still likely to contain large uncertainties (as exemplified by the "re-labelling" experiments of, for instance, the AMIDA13 challenge).

Training and evaluating on the consensus annotations may therefore not be the best approach for machine learning, as the information on interobserver variability is thus hidden to the model itself and can only serve as a tool for later discussion. Having access to individual annotation maps brings lots of opportunities for better leveraging this information in the training and in the evaluation process, and to go beyond the "agreement with the consensus" that is typically used as the definitive measure of an algorithm's performance.

Comparisons between the performance of an algorithm and the performance of "human experts" need to be made carefully. If the comparison is made with a "consensus" annotation, it is necessary to make sure to be fair to both the algorithm and the experts. This can be achieved either through a panel of experts external to the "consensus panel", used solely for the performance comparison, or through a leave-one-out strategy on the generation of the consensus if an automated method such as STAPLE or majority voting is used.

Our results also show the importance of being extremely precise and detailed when translating rules made for pathologists (such as the ISUP guidelines for prostate cancer grading) into algorithms applicable to machine learning outputs. For the Gleason grading, small differences in the implementation of how to get the Gleason groups from the identified patterns can lead to differences in results. The Gleason 2019 challenge lacks precision in this regard, and this clearly leads to difficulties in interpreting the published results of different researchers, as different assumptions about the data and the task itself are made. This precision is particularly important for competitions, as they typically encourage researchers from outside the specialized field of the task to compete. Gleason 2019 winner Yujin Hu[53], for instance, states on GitHub[54] that "I don't quite understand task2, and got it wrong when I participated this challenge", task2 being the core-level Gleason score prediction task and task1 the per-pixel Gleason pattern semantic segmentation task.

When training on data with interobserver variability, results from Alain Zheng's master thesis on Gleason grading point to potentially better performances when using individual annotations from all pathologists than when using the consensus only. This, however, required some adequate strategy to focus the final predicted grade on the regions where the per-pixel prediction was more certain.

[53] https://gleason2019.grand-challenge.org/Results/
[54] https://github.com/hubutui/Gleason

The evaluation of algorithms based only on agreement to consensus may not give the full picture, particularly if the goal is to compare algorithms with experts. Head-to-head comparisons between experts and algorithms give a more detailed view of the algorithm's behaviour. This also implies that the results may be more difficult to interpret. The use of adapted visualisation techniques such as MDS can quickly point us towards the potentially interesting insights. This does require some care, however, as there can be large differences between the distances in the MDS visualization and the values in the dissimilarity matrix. It is important to always check the projection error and to always validate any insights with the data in the dissimilarity matrix.

Another clear insight from our studies is that comparing results from post-challenge publications with challenge results is extremely complicated when the test set annotations are not publicly released after the challenge has ended, and when key steps of the evaluation process (such as, for Gleason 2019, the precise method used to compute the core-level scores from the segmented patterns) are not fully described. This encourages researchers to implement their own methods, which will inevitably vary between researchers, leading to performances measured on essentially different datasets. When the challenge has multiple annotations available, the implementation of the consensus mechanism in particular is a really important component to publicly release. Otherwise, it encourages researchers to simplify the problem, leading to the problem in Gleason 2019 of several post-challenge publications choosing to reduce the dataset to the expert most in disagreement with the others only.

Releasing test set annotations and all the code necessary to evaluate results from a set of prediction maps is the only way to remove ambiguities and to ensure that results are reproducible and comparable.

# 8 Quality control in challenges

The organisation of digital pathology challenges is a very complex process that is inevitably subject to errors. These errors can happen anywhere from the annotation process to the evaluation code or the analysis of the results. They can also significantly impact the results of the challenge, and therefore the conclusions drawn from them, influencing the trends in designing models and algorithms to solve similar tasks in the future.

In this chapter, some examples of mistakes in challenges will be examined. More specifically, our focus will be on the problems that those mistakes can cause during and after the challenge, and what can be done to mitigate those effects. The three main challenges that will be used as examples are the MITOS 2012, Gleason 2019 and MoNuSAC 2020 challenges. We already mentioned the main issues with the MITOS 2012 challenge, which were examined by Elisabeth Gruwé in her Master Thesis [191] and acknowledged by the organisers in the post-challenge publication [166]. Some of the problems of the Gleason 2019 challenge were reported by Khani et al [321] in 2019, and we added our own analysis in different publications [5], [8]. Our analysis of the MoNuSAC 2020 errors were published as a "comment article" in response to the challenge's publication [6], alongside the author's response [23], while some additional analyses were provided in a research blog publication[55].

It is important to emphasize here that these analyses should not be taken as a condemnation of the challenge organisers. Part of the reason that we were able to conduct these analyses is because the challenges were more open and transparent about their process than most. The MITOS12 challenge organisers, for instance, published their full test set annotations, organized one of the first digital pathology challenges, and corrected most of their problems by the time they organised the 2014 edition of their challenge, MITOS-ATYPIA 14. Gleason 2019, as we have established in the previous chapter, is the only digital pathology challenge to publish multiple individual expert annotations, allowing for a much finer analysis of interobserver variability than what is usually possible. MoNuSAC 2020, meanwhile, is the only digital pathology challenge to publish the predictions of some of the participating teams on the whole test set, instead of the summarized result metrics and selected examples for the post-challenge publication. This alone makes it a very valuable resource, despite the problems with the results of the challenge themselves.

## 8.1 MITOS12: dataset management

Mitosis detection, as we have seen through this thesis, has been a very popular topic for image analysis. The ICPR 2012 Mitosis Detection challenge was very influential in opening up the field of digital pathology to deep learning methods, thanks to the excellent performance obtained by the IDSIA team of Cireşan et al. The dataset was made out of selected regions from five H&E-stained slides, each coming from a different patient. Each slide was scanned with three different whole-slide scanners. Ten regions per slide were selected, and they were split into a training set of 35 regions and a test set of 15 regions. The annotations were made by a single pathologist, who annotated 326 mitoses.

[55] https://research.adfoucart.be/monusac-error

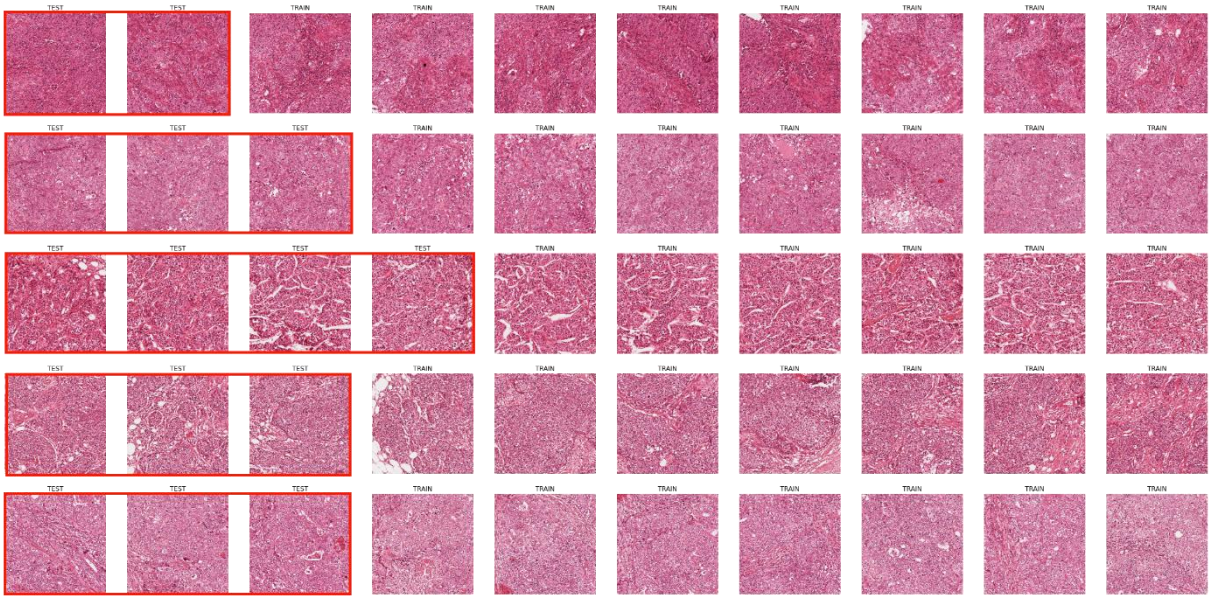


**Figure 8.1. All (heavily down-sampled) images from the MITOS12 dataset. Each row corresponds to a different WSI. Images within the red rectangles are part of the test set.**

The detection task evaluated in the challenge consisted in predicting the centroids of the mitosis. A predicted mitosis was determined to be a "true positive" if its centroid was within a range of 5µm from the centroid of a ground truth mitosis. The Precision, Recall and F1-Score were used as metrics, with the F1-Score being the primary measure for the ranking[56].

The use of a single pathologist for determining the ground truth is an obvious limitation of the dataset, but the most important problem is clearly that the dataset is split at the *image region* level instead of at the WSI level (which in this case is the same as the patient level). All the image regions from the dataset are shown in Figure 8.1, with the test set images framed in red. It is clear, even looking from a very low-resolution point of view, that images extracted from the same patients will share common features.

In Gruwé's Master Thesis [191], a comparative study was made between evaluating with a five-fold cross-validation with the split at the image level, with a five-fold cross-validation at the WSI level (which in this case corresponds to a "leave-one-patient-out" cross-validation). With the image level cross-validation, the average F1-Score obtained was 0.68, whereas for the leave-one-patient-out cross-validation, it was 0.54.

It is clear that keeping the test set as independent as possible from the training set should be standard practice to avoid learning patient-specific features. However, as public challenge datasets attract researchers which may not be familiar with the specificities of biomedical data, splits at the image level can still be found in recent publications. For instance, Rehman et al. [330] demonstrate excellent results on the four usual mitosis datasets (MITOS12, AMIDA13, MITOS-ATYPIA-14 and TUPAC16), but their experimental setup was that "[e]ach dataset was divided into five chunks each containing 20% randomly selected images of the dataset". In doing so, they actually propagate the "mistake" of the MITOS12 dataset to the later datasets where the challenge split was correctly made. A similar error is done in Nateghi et al. [195], where "each dataset is

[56] http://ludo17.free.fr/mitos_2012/results.html

divided into five random subsets and 5-fold cross-validation is performed in each of these subsets", with the MITOS12, AMIDA13 and MITOS-ATYPIA-14 datasets being used.

A key factor in this reoccurring error may be the unavailability of the test set annotations for the later challenges, even years after the end of the competition. Researchers thus have to make their own train/test split from the available training data, which can easily lead to them overlooking some key aspects of the methodology. This in turns leads to improper comparisons between methods, made on different subsets of the data.

## 8.2 Gleason 2019: annotation errors and evaluation uncertainty

The Gleason 2019 publicly released training dataset contains 244 H&E-stained TMA cores annotated by 3 to 6 expert pathologists (see Annex B). These maps show a large variability between the experts, which is expected given the high interobserver variability typically observer for that task. Part of the variability, however, is not just due to differences of opinions, but to clear mistakes in the creation of the annotation maps from the expert annotations. These mistakes were first noted by Khani et al. [321], who noted that "some of the contours drawn by the pathologists were not completely closed", which led to the corresponding regions not being filled in the annotations. Some examples are shown in Figure 8.2. For their own experiments, Khani et al. identified and manually corrected 162 of these problematic annotation maps. From our own analysis, several more annotation maps with mistakes were identified, bringing the total to 185 of the 1171 available annotations, meaning that around 15% of the provided annotation maps are incorrect.

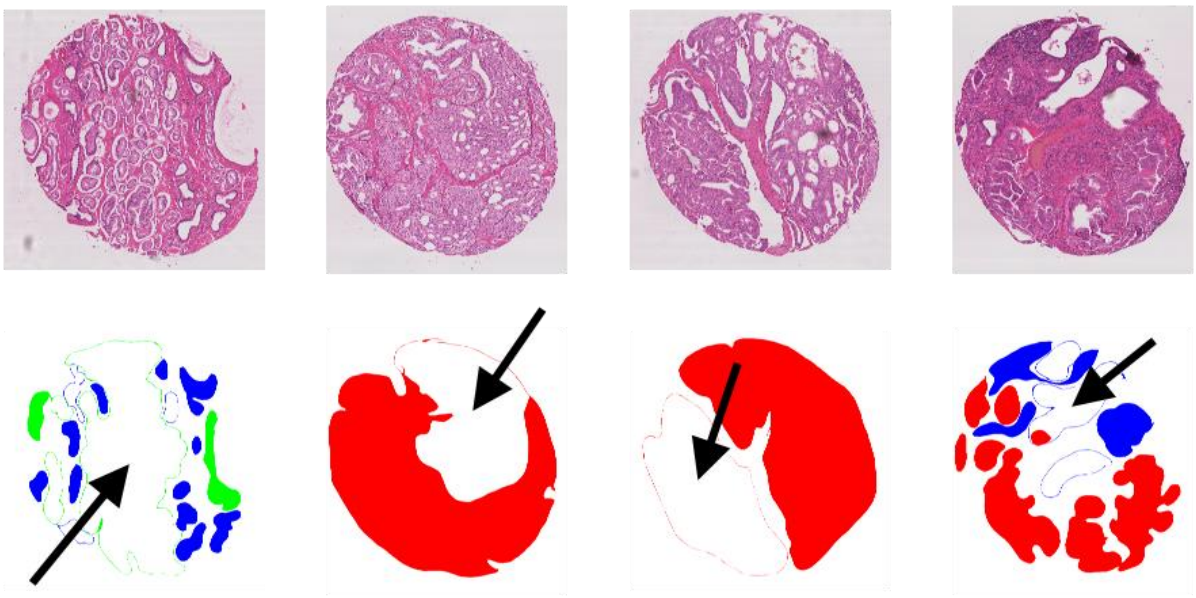


**Figure 8.2. Examples of mistakes in the publicly released annotation maps. (top) RGB images of the TMA spots, (bottom) associated annotation map with improperly closed contours leading to large regions incorrectly labelled (see black arrows). Colours correspond to the different Gleason patterns (blue = benign, green = grade 3, red = grade 4).**

These mistakes are also not randomly distributed in the dataset, as some experts are largely more affected than others, as shown in Table 8.1. Pathologist 2 is particularly affected by this problem. Taking a more detailed look at the annotations show that, in general, this pathologist attempted to be a lot more detailed in their annotations compared to the others. In our own experiments on interobserver variability presented in Chapter 6, we chose to remove those annotations from the dataset instead of potentially adding our own biases and mistakes by attempting a manual correction.

**Table 8.1. Number of annotation maps that contain clear mistakes with unclosed contours, per pathologist.**

| Pathologist 1 | Pathologist 2 | Pathologist 3 | Pathologist 4 | Pathologist 5 | Pathologist 6 |
|---|---|---|---|---|---|
| 5 | 119 | 54 | 6 | 0 | 1 |

It is unclear whether these problems are also present in the test set used by the challenge, as those annotations were not made public. Other publications using the same dataset make no mention of these mistakes [331]–[334]. The dataset used by the challenge's organizing team in their publications [85], [157], [318], [319] does not seem affected by those mistakes, which suggests that they may have happened during the process to generate the data files for the challenge itself. An example of clear differences between the annotations from the challenge organizers' publications and those found in the public dataset is shown in Figure 8.3. Unfortunately, the challenge organizers did not respond to our requests for clarification to the contact address provided on the challenge website. One contacted author of subsequent studies responded that he agreed "that the data uploaded on the challenge website has the problem that you indicated", without elaborating further.

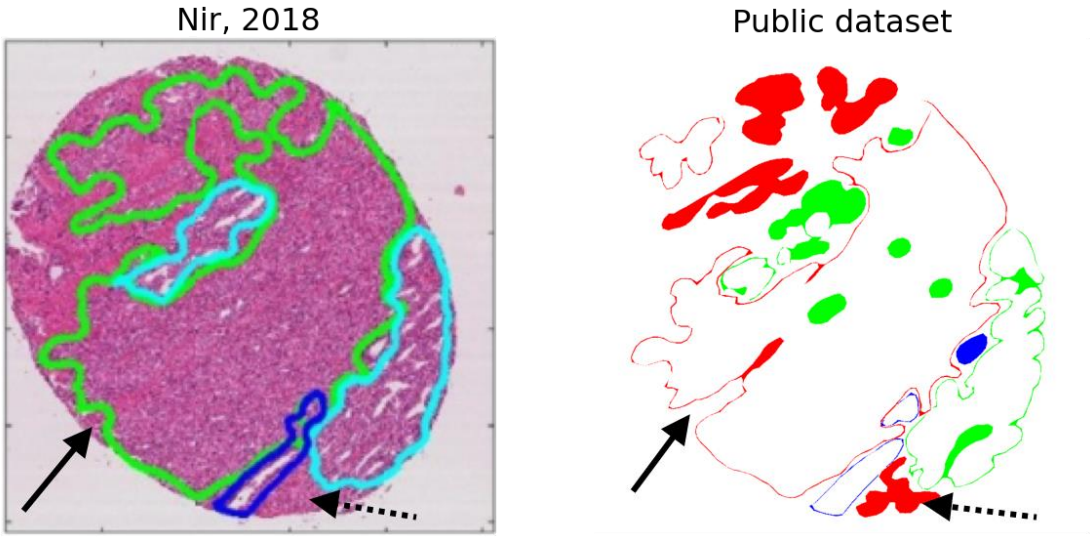



**Figure 8.3. Difference in the annotation map for "Pathologist 2" on a TMA core between an illustration from Nir, 2018** [157]**, and the publicly available annotation map from the challenge website**[57] **("slide005_core063"). In addition to the "closed contour" difference, some details from the publicly released dataset are missing from the one used by Nir et al., such as some detailed indentations in the contour (solid black arrow), or some smaller annotated regions (dotted black arrow).**

[57] https://gleason2019.grand-challenge.org/Register/

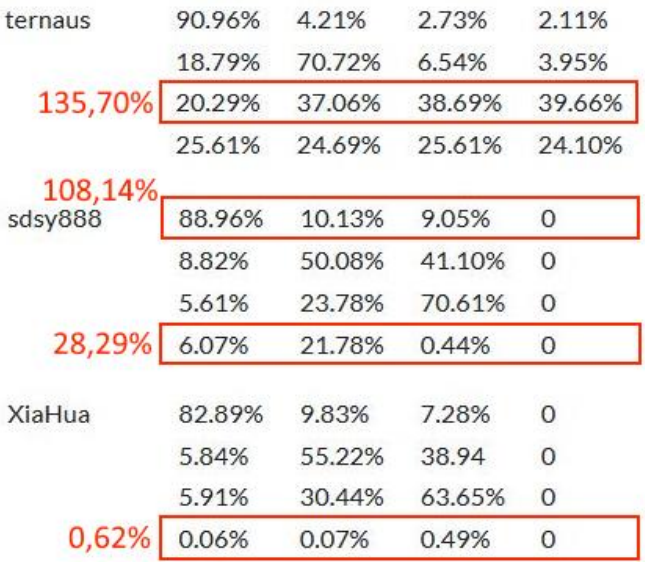



**Figure 8.4. Inconsistencies in the reported confusion matrices from the Gleason 2019 challenge website. While each row is supposed to be "normalized to sum up to 1", many reported rows sum up to higher or lower values well outside the scope of rounding errors.**

The evaluation of the challenge also shows some signs of potential issues. The results reported on the challenge's website include a ranking of all the teams based on a custom score which combines the Cohen's kappa, the micro-averaged F1-Score and the macro-averaged F1-Score. They also provide a confusion matrix of the pixel grading for each of the top 8 teams, which is announced as being normalised so that each row (representing the "ground truth" grade) sums up to 1, providing the sensitivity of each grade on the diagonal. There is no available evaluation code, and the exact definition of the metric is unclear (Cohen's kappa could be unweighted, linear or quadratic, the F1 score could be computed on the image-level classification task or on the pixel-level segmentation task, and the macro-averaging could be done as the arithmetic mean of the per-class $F_1$ scores or as the harmonic mean of the average precision and recall). In any case, the values presented on the challenge website[58] for the confusion matrices have some inconsistencies, as illustrated in Figure 8.4. Many rows do not sum up to 1, with differences that cannot possibly be explained by rounding errors. Some of the results, such as for the algorithms ranked 5th and 7th in Figure 8.4, imply that no predictions for Grade 5 were made at all.

As the evaluation code is not available, there is no way to know if the errors were only made in the reporting of the results on the website, or if they impacted the final score on which the rankings were made. They also appear in the recently released post-challenge publication of the challenge winners [210], in which the $\kappa$ and both F1-scores are used for the pixel-level evaluation, and accuracy, precision, sensitivity and specificity are used for the core-level evaluation. Confusingly, the reported individual components of the challenge metric (i.e. $\kappa$, $MF1$ and $\mu F1$) do not match with the reported score using the formula provided by the challenge organizers. As of June 2022, no official "challenge" publication has been made (in fact, none of the publications by the University of British Columbia team refer to the competition itself), making it very difficult to fully understand the exact evaluation process.

[58] https://gleason2019.grand-challenge.org/Results/

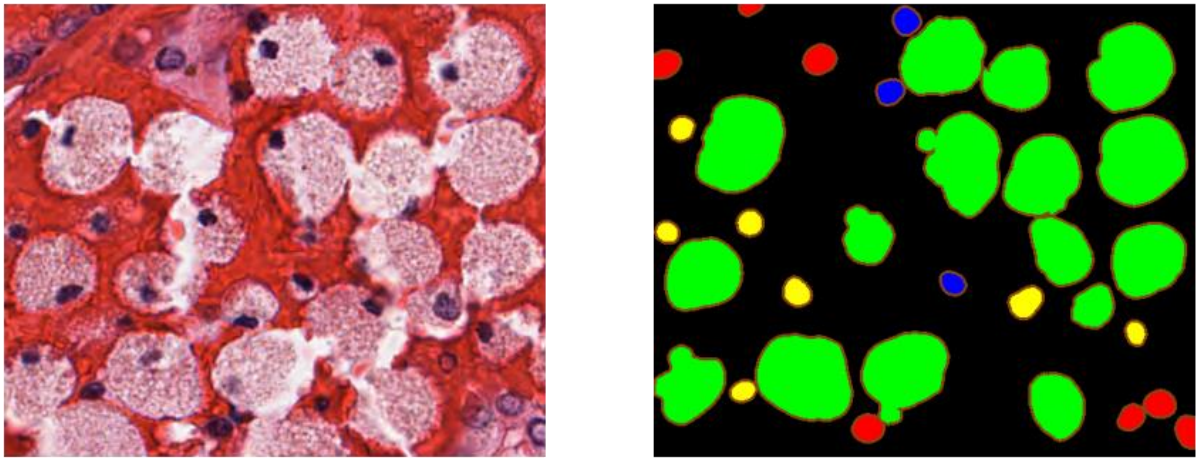


**Figure 8.5. Example of an image from the MoNuSAC test set, with the "color-coded" annotations from one of the top teams. Epithelial nuclei are in red, lymphocytes in yellow, neutrophils in blue, macrophages in green, and boundaries are highlighted in brown.**

## 8.3 MoNUSAC 2020: errors in the evaluation code

The MoNuSAC 2020 challenge was remarkably transparent about their data, the results of the participant, and their code. A pre-challenge preprint was published on ResearchGate[59] detailing the challenge procude, the evaluation metric, and the expected submission format. Code for reading the annotations and for computing the evaluation metric was released on GitHub[60]. Examples of the submission file's format were provided via Google Drive[61]. After the challenge, in addition to the post-challenge official publication [24], supplementary materials were released with all of the participating teams' submitted methods[62].

All the images and annotations from the training and the test set were also released on the challenge website[63], as well as the "color-coded […] predictions of top five teams". These "color-coded" predictions were not the raw submissions of the challenge participants, but a visualization that emphasized the contours of the segmented objects (as shown in Figure 8.5). As the challenge's task is instance segmentation and classification of cell nuclei, being able to visually assess the separation between objects is important. The ground truth annotations were provided as XML files with the vertex position of all annotated objects.

Providing the teams' predictions, the ground truth and the evaluation code should in theory be sufficient for independent researchers to reproduce the results of the challenge. Several issues make that reproduction difficult in this case. First, the "color-coding" process makes it difficult to recover the exact pixel-precise predictions of the participating team. Thanks to the collaboration of Dr. Amirreza Mahbod, who kindly shared with us the raw submission files of his team, we can see that this visualization was made by adding a 3px wide border outside of the predicted object (see Figure 8.6). For relatively isolated object, it is therefore possible to automatically recover the teams' predictions from the color-coded masks. However, for objects that are close together

[59] https://www.researchgate.net/publication/339227864_Multi-organ_Nuclei_Segmentation_and_Classification_Challenge_2020
[60] https://github.com/ruchikaverma-iitg/MoNuSAC
[61] https://drive.google.com/file/d/1f_dpKpS4z8DGzw_xHvxxPYMSW0ReHW_R/view
[62] https://drive.google.com/file/d/1kdOl3s6uQBRv0nToSIf1dPuceZunzL4N/view
[63] https://monusac-2020.grand-challenge.org/Data/

(frequent in cell nuclei segmentation), this coding introduces uncertainty on the exact boundary between objects due to thickened contours that create artefactual overlaps (see Figure 8.6). Second, a careful analysis of the public evaluation code shows that it had to use more than the publicly released ground truth annotations and evaluation code. The code that reads the XML annotation files[64] produces TIF image files (one file per nucleus class and per RGB image), whereas the code that computes the Panoptic Quality evaluation metric[65] reads the ground truth from MATLAB files, which was the required format for the participants submissions.

These are, however, relatively minor problems, and the results of the challenge should still be reproducible from the available data, within a reasonable range of error. The evaluation code itself, however, contains several mistakes that make the published results highly unreliable. Our analysis of these errors was published as a comment article to the challenge publication [6]. In this section, we will present our analysis and the author's reply [23].

### 8.3.1 Error in the Detection Quality computation

The Panoptic Quality, as we previously established in Chapter 4, section 4.4.8, combines the "detection quality" (with the F1-Score) and the "segmentation quality" (with the average $IoU$ of the true positive objects). In MoNuSAC, these values are aggregated by first computing the $PQ_{ci}$ (per-image $i$ and class $c$), then computing the average per-image $PQ_i = \frac{1}{m_i}\sum_c PQ_{ci}$, where $m_i$ is the number of classes in image $i$. Finally, the average $PQ$ on the whole test set is computed as $aPQ = \frac{1}{n}\sum_i PQ_i$, with $n$ the number of images in the test set.

In the provided code, this is done in the "PQ_metric.ipynb" notebook available on the challenge's GitHub. The method `Panoptic_quality(ground_truth_image, predicted_image)` computes the $PQ_{ci}$. It takes as input the ground truth "n-ary mask" (i.e. the mask of labelled objects in the class) and the predicted n-ary mask. Some regions from the images were also marked as "ambiguous" in the ground truth annotations and were thus removed from the ground truth and the predicted images before computing the metric. This method, however, contains a bug, which is detailed in our analysis available on GitHub[66].

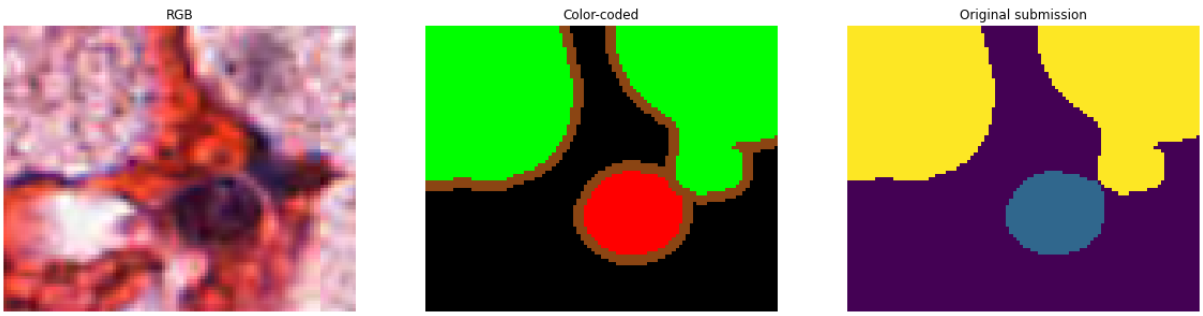



**Figure 8.6. Detail from the MoNuSAC image in Figure 8.5, with (left) the RGB image, (middle) the color-coded prediction mask publicly released by the challenge organizers, and (right) the per-pixel predicted class from the submission file shared by Dr. Mahbod.**

[64] https://github.com/ruchikaverma-iitg/MoNuSAC/blob/master/n-ary_mask_generation.ipynb
[65] https://github.com/ruchikaverma-iitg/MoNuSAC/blob/master/PQ_metric.ipynb
[66] https://github.com/adfoucart/monusac-results-code-analysis

The error causes the list of “false positives” in the image to be overestimated in some cases. The effect of the error can be very different depending on the strategy used by the team to generate the instance number in the “n-ary masks”. Indeed, there are two strategies when generating the per-class prediction masks. As each class prediction has to be sent separately, it makes sense to reset the instance count for each class, so that the "n-ary mask" will always start with instance index 1. Conversely, as the goal of the challenge is to provide "segmentation and classification" for all classes in an image, it would also make sense to provide a unique instance number for each object in the image, and then to separate the objects by class. In that case, the instance indices could follow each other from class to class (for instance: [1,2, ..., N] for the epithelial instances, [N+1, N+2, ..., M] for the lymphocyte instances, etc.), or they could be completely mixed. Nothing in the provided instructions require one strategy over the others.

The provided example .mat files illustrating the format chose the "indices following each other" method, so that the indices for a given class will typically only start at 1 for the first class present in the image. The code building the "n-ary masks" from the .xml annotations for the ground truth follow that same rule. In the submission file provided to us by Dr Mahbod, however, the indices are mixed. For the image shown in Figure 8.5, the epithelial indices are for instance [1, 4, 5, 33, 36, 37, 38], and the lymphocyte indices are [11, 16, 17, 23, 25, 28, 32]. We can use that same image to see the effects of this error on the computation of the metric.

The error in the metric computation is stronger when the indices are “misaligned”. In Figure 8.7, the instance labels from the ground truth annotations and the submission file are shown for the same image. In Table 8.2, the results from the computation of the $PQ_{ci}$ and the number of FP found are reported, using the code used for the challenge and our code with the typo corrected. Apart from the neutrophils, where there was no ground truth object, all other classes are affected by the error in the code. For the macrophages in particular, 9 of the 16 predicted objects are counted as both TP and FP. Notably, the range of the error is somewhat diluted by the nature of the PQ itself. As the PQ multiplies the detection F1-Score with the average $IoU$ of the TP, the error on the F1-Score is less visible.

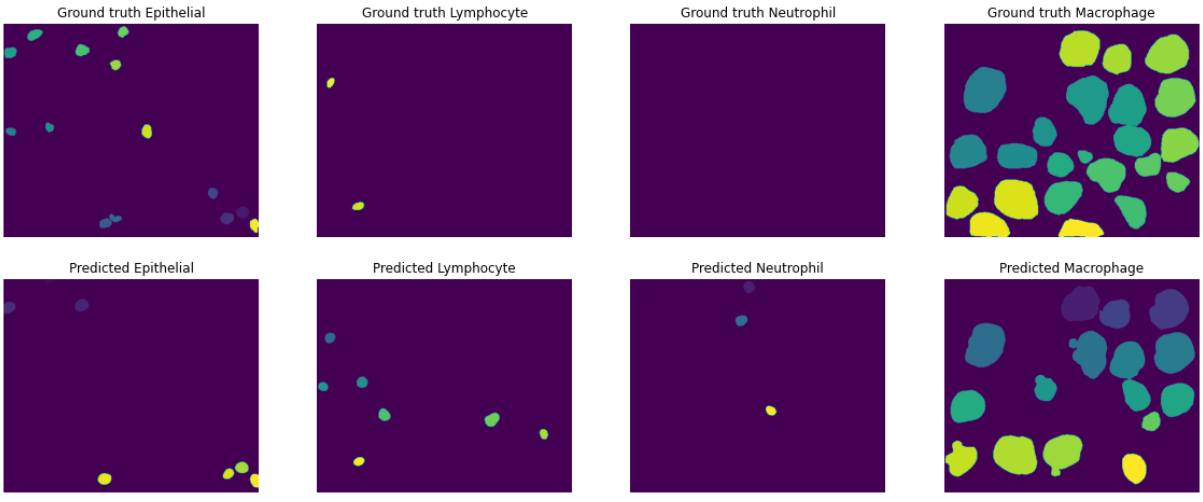



**Figure 8.7. Per class labels on the image previously shown in Figure 8.5, with the ground truth computed from the XML annotations and the predicted labels from the submission files shared by Dr Mahbod (one colour per object).**

After our comment article [6], the authors acknowledged this error and recomputed all the scores of the challenge in their reply [23]. Their updated results show that for most teams the difference between the updated PQ and the initial PQ is around 0.04, but that some teams are more strongly affected. Dr Mahbod's team, for instance, see their initial submission's score move up from 0.389 to 0.548, gaining three places in the ranking. It is possible that their labelling strategy is the main cause of their poor performance in the original ranking. As we can see in Table 8.3, re-labelling the predicted labels so that they follow each other from class to class considerably lower the effect of the error.

**Table 8.2. $PQ_{ci}$ and FP per-class computed from the labels shown in Figure 8.7, using the challenge code and our corrected code.**

| | Challenge | | | Corrected | | | Both |
|---|---|---|---|---|---|---|---|
| | $PQ_{ci}$ | $FP$ | $F_1$ | $PQ_{ci}$ | $FP$ | $F_1$ | $m\_IoU$ |
| **Epithelial** | 0.379 | 5 | 0.480 | 0.451 | 1 | 0.571 | 0.789 |
| **Lymphocyte** | 0.298 | 6 | 0.400 | 0.331 | 5 | 0.444 | 0.746 |
| **Neutrophil** | 0. | 3 | 0. | 0. | 3 | 0. | 0. |
| **Macrophage** | 0.555 | 9 | 0.667 | 0.683 | 0 | 0.821 | 0.832 |

**Table 8.3. Average PQ on the whole dataset recomputed from the shared submission files of Dr Mahbod, compared with a re-labelled version (where labels are assigned incrementally from one class to the next) and a version with the corrected code. The scores computed after correcting the "undetected false positive" error are also reported.**

| | Labels from submission file | Re-labelled | Corrected code |
|---|---|---|---|
| **With undetected FP error.** | 0.536 | 0.566 | 0.596 |
| **FP error corrected.** | 0.410 | 0.429 | 0.456 |

### 8.3.2 Undetected false positives

In addition to the error in the computation of the $PQ_{ci}$, the challenge code also contains an error in the aggregation of the $PQ_{ci}$ into the per-image $PQ_c$. The way that the code processes a team's predicted mask is to start from a list of files taken from the *ground truth directory*. Each of this file will contain the n-ary mask for one class on one image. Iterating through this list, the code then finds a corresponding .mat file in the team's predictions directory, which had to follow the same structure (`PATIENT_ID/PATIENT_ID_IMAGE_ID/Class_label/xxx.mat`). The $PQ_{ci}$ is then computed based on these two n-ary masks and added to an Excel file where the $PQ_i$ and finally averaged $PQ$ could then be computed.

There is, however, no code provided that checks if there were additional files in the *team predictions directory* that have no corresponding file in the ground truth directory. In the example that we used above, for instance, there are predicted neutrophils but there are no ground truth neutrophils. As the code is written, the prediction file of the neutrophil class is never opened, and the false positive neutrophils are never counted. The $PQ_i$ for that image would therefore only be computed based on the $PQ_{ci}$ of the three other classes, and the fact that three additional false positives were found is ignored. In this case, all three falsely predicted objects were present in another ground truth class, which means that the error is at least counted in the "false negatives" of the other classes. Contrary to the assertion in the author's reply, however, this is not necessarily

always the case: these "false positive" could be made on a region that is not annotated at all, and thus be a "false negative" of the background class, which does not contribute to the metric. In that case, the error would not be counted at all.

Even when the false positives really are all misclassifications of objects of another class, the challenge's approach would still not work. We can see in Figure 8.7 that some predicted lymphocytes are found in the ground truth of the epithelial class. As there are also some ground truth lymphocytes, those misclassifications are counted both as "epithelial false negatives" and as "lymphocytes false positives", whereas the same error with the neutrophil class is only counted as "epithelial false negative".

To avoid this problem, the logical solution would be to set the $PQ_{ci}$ to 0 when there is a false positive and no ground truth object. As we can see in Table 8.3, this has a considerable effect on the scores. It does seem very harsh to set the $PQ_{ci}$ to 0 for what may be a single error, which would have a big impact on the $PQ_i$ and eventually the $PQ$. This is in large part due to the next problem: the aggregation strategy.

**Table 8.4. Effect of the error on the aggregation method, with or without the correction on the undetected FP. All the values are computed with the corrected version of the $PQ_{ci}$ computation.**

| | **Original aggregation method** | **Corrected aggregation method** |
|---|---|---|
| **With undetected FP error.** | 0.596 | 0.618 |
| **FP error corrected.** | 0.456 | 0.563 |

### 8.3.3 Aggregation strategy

The third big issue with the challenge evaluation is the aggregation strategy. As explained above, the challenge first computes the $PQ_{ci}$ per-class and per-image, then computes a $PQ_i$ per-image, and finally computes the average over the 101 images of the test set. Those images, however, are actually small regions extracted from 25 WSI (each one from a different patient). The images, meanwhile, have a very large range of sizes and number of objects. The smallest image only has 2 annotated nuclei, whereas the larger images can have hundreds of them. The published methodology from the challenge organisers is a bit unclear on what they intended to do. In the author's reply, they clearly state that this was a methodological choice and not an implementation error. The confusion may have stemmed in part from their post-challenge publication, which states that the final $aPQ$ is computed from the "[a]rithmetic mean of the 25 $PQ_i$", and that "[p]articipants submitted a separate output file for each of the 25 test images", while the pre-challenge preprint often refers to the 101 images as "sub-images". Regardless of whether the choice of aggregating per-image rather than computing the $PQ_{ci}$ per-patient (first compiling the TP, FP, FN and IoUs for each "sub-image" of the patient) was intentional or not, it is clearly at the very least a large methodological flaw.

The diversity in image sizes means that errors in small images have a much bigger impact on the overall score as errors in large images. The author's reply justifies this strategy by the need to compute statistics (in their case, confidence interval on the scores). However, if the samples are the image regions, those that come from the same patient will not be independent. The size diversity also makes the distribution of results much noisier. Even though the number of samples

is higher, the statistical analysis is necessarily flawed if the metrics themselves are flawed. A per-patient aggregation still provides enough samples (N=25) for some statistical analysis, with samples that make a lot more sense from a clinical point of view. As we can see in Table 8.4, the errors in aggregation method and with the missed false positives partially "cancel" each other: the correct aggregation method leads to higher PQ overall (as no single errors are penalized too harshly), while adding the previously missed false positives obviously lowers the score, and correcting the $PQ_{ci}$ computation, which *added* false positives, increases the score once again.

### 8.3.4 Error in the reporting of the per-class results

A minor issue, which was corrected by the authors in their reply, also appeared in the detailed results published as supplementary materials, where the scores were computed per-organ and per-class. The macrophage and neutrophil classes were inverted in the results. This was simply the result of an "inadvertent column-header swap" [23]. This is very likely given our own analysis, as everywhere in the code the same class order is used with neutrophils before macrophage, but the order is reversed in the supplementary material's table.

### 8.3.5 Corrections and scientific publishing

The challenge organisers were contacted and notified of the first error (in the computation of the Detection Quality) in September 2021. They initially replied that the code was correct and, upon receiving evidence that it was not the case, stopped responding. The IEEE Transactions on Medical Imagining editor in chief was contacted in October 2021 and made aware of the problem, and our comment article summarizing our findings was submitted on October 20th, 2021. On February 23rd, 2022, we received notice that our comment article was accepted. It appeared online on the April 2022 issue of the journal, alongside the authors' reply.

Between October 2021 and April 2022, there was no notice on the original publication that there was any dispute on the validity of the results. The original article is still accessible (and available for purchase) in its original state, with a link to the comment article provided in the "Related Content" tab but no clear indication that the results published inside are incorrect. As not all corrections were accepted and implemented by the authors, the actual results of the challenge are still uncertain at this stage and are likely to remain so.

While it is perfectly normal that editors take the time to make sure that comments and notice of errors on published materials are properly reviewed before issuing a correction, it does not seem right to take no action at all once it is clear that there is at least some credibility to the claims. The choice of not correcting the original article or have a clear notice pointing to the correction, but to publish the comments and authors' reply separately in another issue of the journal is also strange, as it reduces the trust that a reader can have in currently published results.

## 8.4 Discussion and recommendations: reproducibility and trust

The terms "reproducibility" and "replicability" are used sometimes interchangeably and with contradicting conventions to refer to two distinct ideas [335]:

- The ability to **regenerate the results** using the original researcher's code and data.
- The ability to **arrive at the same scientific findings** by an independent group using their own data.

This first definition is crucial in ensuring trust in the published results and is particularly important in the context of digital pathology, deep learning, and challenges. Challenges on

complex tasks, as in most digital pathology competitions, often require ad hoc code for the processing of the participants submissions and their evaluation. The opportunity for errors to appear in the implementation of this code is fairly high. There are also many small implementation choices that may influence the results and are not necessarily always reported in the methodology, whether by oversight or by space constraints in the publications.

It is therefore crucial for the source code of the evaluation process to be made available. After publication of the competition results, making the test set "ground truth" annotations available is also necessary to avoid future researchers having to re-create their own train/test split and making improper comparisons of their results with the state-of-the-art. While it obviously creates the possibility of improper management of this dataset by those researchers (i.e. training their model also on the test set to obtain better results), this risk would be best countered by requiring the code of those publications to also be publicly released, rather than by keeping an embargo on the annotations.

While some competitions may prefer to keep an "online leaderboard", with the possibility for researchers to keep submitting new results that are automatically evaluated, this does not fulfil the requirements for independent reproducibility of the results. To have full reproducibility, the participants submissions and/or their own source code must also be made available.

These may seem like extreme requirements for a competition, but our analysis of the challenges in this thesis shows that the potential for error or misinterpretation is very high. The Seg-PC 2021 challenge, for instance, requires participants to submit segmentation masks for instances of multiple myeloma plasma cells, with labels indicating the pixels that belong to the cytoplasm, and the pixels that are part of the nucleus. The evaluation code, available on the Kaggle platform[67], shows however that the ground truth and submission masks are binarized so that the "nucleus" and "cytoplasm" labels are merged into a single "cell" label, on which the IoU is computed. The ranking of the challenge therefore does not take into account whether a particular method correctly identifies the nucleus from the cytoplasm. This is a fairly important information for correctly interpreting the results of the challenge, and the fact that the source code is public makes it much more obvious in this case, as the methodology was not clear on that issue.

Mistakes can happen in all parts of the deep learning process in digital pathology pipeline: from the collection and labelling of the data, to the generation of the predictions, the evaluation of those predictions, and the reporting of the results on a website or in a peer-reviewed publication. Transparency on the whole process is key in building trust in the published results, which is undoubtedly necessary before deep learning algorithm can be safely included in clinical practice.

A proposed timeline for the release of the different data necessary to ensure the reproducibility of challenge results is presented in Figure 8.8. As soon as the challenge opens (or before), the images and annotations from the training set should be released, as well as the evaluation code and an example of the expected submission format. This ensures that participants are able to participate in the quality control of the challenge and are able to independently verify that the evaluation process corresponds to the methodology described by the organisers. This also allows participants to give feedback on aspects of the evaluation that the organisers may have overlooked well before the evaluation period, so that the code and methodology can still be adjusted. After the announcement of the results at the end of the evaluation period, the test set annotations and

[67] https://www.kaggle.com/datasets/sbilab/segpc2021dataset

the participants' submission files (or code) should be made available to at least all challenge participants. In this way, when preparing the post-challenge publication, participants can again independently verify that the evaluation was done correctly and fairly. This can also stimulate challenge participants to contribute interesting insights on the results, thus "crowd-sourcing" the results analysis to a larger pool. Participants can therefore play a more active role in preparing the post-challenge publication, with all the necessary data at their disposal. After publication, all the data should be available to the larger research community, to ensure full transparency and reproducibility.

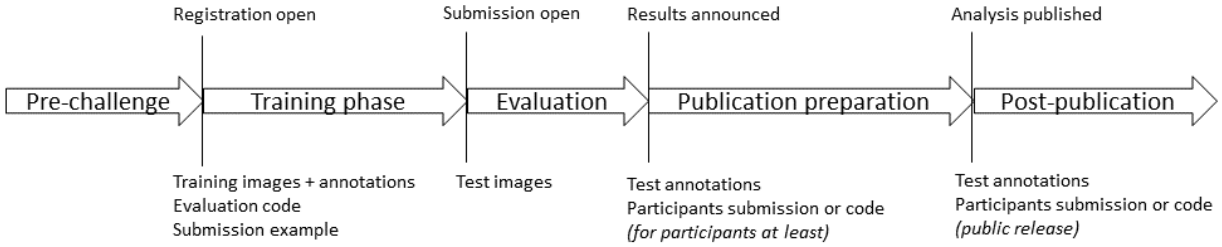



**Figure 8.8. Timeline of transparency requirements for a better reproducibility of challenge results.**

# 9 Discussion and conclusions

Large convolutional neural networks have become the dominant technique in image analysis. The transition from "traditional" methods based on "handcrafted features" to these "deep learning" techniques has been very quick. In digital pathology, while deep learning methods are the state-of-the-art for the most common tasks of detection, classification, and segmentation, they seem to remain in a constant state of "promising results" for future clinical applications [336], [337].

The adoption of such methods into clinical practice require to build trust in their results. Trust, in turn, requires explainability (*why* do these methods produce their results) and replicability of the results. Explainability and replicability are important for the inner workings of the deep learning algorithm itself, but it must also apply to the other elements of an "automated" digital pathology pipeline, from the collection of the dataset to the evaluation procedure.

A key characteristic of digital pathology that make this trust-building difficult is the unavoidable uncertainty that surrounds the *annotation* process. Supervised deep learning algorithms and evaluation processes are built around the existence of a singular, absolute "ground truth" to which the predictions of the algorithm can be compared. As we have seen through this thesis, this absolute ground truth does not exist in digital pathology.

Real-world annotations in digital pathology are imperfect. For segmentation problems, precise annotations are so time-consuming, and deep learning algorithms require so much data, that obtaining these annotations from senior experts is impractical. Even for less time-consuming forms of annotations (such as patient-level or image-level labels for classification or grading), high interobserver variability implies a large uncertainty on the validity of any label, which can only be mitigated through time-consuming consensus processes. If imperfections cannot be avoided, then they must be a part of the learning process and of the evaluation process.

In this discussion, we will now summarize the key findings of this thesis:

- What are the effects of imperfect annotations on deep learning algorithms, and how can we mitigate those effects?
- How can we include these imperfections into the evaluation process, and how do we ensure that our quantitative results are as relevant as possible to the clinical or biological reality of the data?
- How do "digital pathology challenges" deal with the reality of the data, what are their current shortcomings, and how can they be improved in the future?

Predicting what the future holds is a risky proposition in any field. It is impossible to know whether deep learning will be the technology that brings the long-awaited prospect of automating at least the most tedious pathology tasks to fruition. We however conclude this thesis with our best vision for what lies ahead.

## 9.1 Deep learning with real-world annotations

A perfect dataset is not necessary for deep learning algorithms. Our SNOW experiments, and other independent works on the topic [260], have shown that, for segmentation tasks in particular, standard network architectures such as U-Net were robust to large imprecisions on the object boundaries, and to a small amount of noise on the class labels. These results have some important

implications on the best strategies for annotating digital pathology datasets for the purpose of training deep learning algorithms.

There is always a trade-off between the quality and the quantity of annotations provided. For the constitution of a *training set*, focusing on the quantity would therefore appear to be a better strategy. For segmentation tasks, this could mean making quick polygonal approximations of the objects of interest rather than detailed outlines. The most important aspect may be to make sure to provide a diverse set of examples: patients, acquisition device, laboratory executing the cutting, fixation, and staining procedure, etc. More generally, our results highlight the importance of a continuous collaboration between the actors from the medical fields – such as pathologists – and developers of image analysis systems. By critically analysing datasets and the impact of the different types of imperfections they may contain, the latter can direct the annotation efforts of the medical experts where they are most needed. Likewise, pathologists can use their expertise to ensure that the definition of the tasks, the constitution of the datasets, and the nature of the annotations are relevant to their practice.

While deep learning algorithms are naturally robust to some level of imperfections, their performance can be further improved by adapted learning strategies. From our experiments, it seems that relatively simple methods can be applied to mitigate the effects of imperfect annotations. In general, methods that work well seem to rely on expanding the dataset from a core of "more certain" labels. In our "GA" method, this was done by a first training pass on only the "positive" regions of the datasets (i.e. regions containing some annotated objects of interest), followed by a second training pass that included the full dataset and re-labelled "uncertain" regions based on the output of the first pass. Recent works have shown that this approach can be pushed further by iteratively improving the labels, essentially propagating the labels to noisy or unlabelled regions of the latent feature space [85], [277], [338], [339]. Unsupervised or self-supervised approaches may also help identifying "natural" classes present in the data, either as a pre-training step before supervised fine-tuning, or with *a posteriori* expert input to associate the "natural" classes to the target. As an example, in her 2022 master thesis, Rania Charkaoui [340] used the self-supervised method from Ciga et al. [341] to find a feature space in which artefact detection could be framed as an outlier detection problem.

Special care has to be taken around the question of class imbalance. Digital pathology datasets often feature large class imbalance in their distributions. The impact of noise may therefore be particularly strong on minority classes. In the annotations of the Gleason 2019 dataset, for instance, Karimi et al. [342] find a much larger inter-pathologists disagreement on the minority "grade 5" Gleason pattern label than on the much more common benign, grade 3 and grade 4 annotations. Typical remedies for class imbalance based on simply over-sampling the minority class may therefore amplify this noise and hurt the overall performances of the algorithms. In such case, having access to annotations from multiple experts could make it easier to focus the learning (and data augmentation) on examples that are part of a larger consensus.

Using annotations from multiple experts rather than a consensus may in general enrich the learning process. While a consensus of experts provides a more "certain" annotation by itself, it also removes the information about which examples are more likely to be incorrect (or at least subject to diverging opinions). A practice that should be avoided is to completely remove contentious examples from the dataset. While this is likely to improve the performances of algorithms trained on that consensual data, it would be at the cost of a false sense of confidence in the results (if this removal of "harder" examples is also done on the test set). Several protocols

used for the generation of consensus annotations in challenges used a "multi-pass" approach where at first individual annotations are made by several experts, then non-consensual cases are debated and corrected. With this approach, it would therefore be possible to provide, ideally, the individual annotations alongside the consensus. At the very least, an indication of which cases were not immediately consensual (as was done for instance in the MITOS-ATYPIA-14[68] challenge) would greatly help in identifying which examples are likely to be difficult.

## 9.2 Evaluation with real-world annotations

Parallel to the problem of *learning* from imperfect annotations is that of *evaluating* algorithms given these imperfections. As the lack of a ground truth often cannot be avoided in digital pathology tasks, any quantitative evaluation metric is therefore bound to have an uncertainty on its value. The effects and scope of that uncertainty is highly dependent on the type of task and the nature of the target object.

In segmentation problems, uncertainty on the exact boundaries of the objects can for instance have a large effect on overlap-based metrics like the IoU if the target objects are small. Noise or interobserver disagreement in the class labels will likewise add an uncertainty on classification or detection metrics. These uncertainties are furthermore to be added to those that are inherent to all machine learning problems, from the aspects of the pipeline that are subject to the effects of randomness. This includes the sampling of the data, the data augmentation, but also, in the case of deep neural networks, the initialization of the network parameters.

With all those sources of uncertainty, it can be very easy to draw inadequate conclusions on the comparative performances of different algorithms based on the quantitative evaluation. A critical analysis of the dataset and of the metrics is necessary to ensure a proper interpretation. Simulations and analysis of the annotations can greatly help in providing ranges of values for different sources of uncertainties, and to set correspondences between the values of the metrics and their interpretation. An example would be to compare individual experts to the consensus or between themselves using the same evaluation metrics as for the algorithms. This can inform on whether the difference between two algorithms could possibly be attributed to the selection of the experts or the consensus method. Simulations based on small perturbations of the annotations are similarly useful.

The effects of randomness on the training process of deep learning methods can be harder to evaluate. Ignoring it, however, may also lead to incorrect conclusions. In most cases, research on deep learning for digital pathology tasks is still in the process of improving the overall pipelines and methodology, rather than validating a specific trained network for a clinical setting. The question that challenges and other publications based on performance comparisons asks is therefore not "is this trained model the best?", but rather "is this proposed methodology the best?". Assessing how robust the results are to changes in the random conditions, or to small changes in the constitution of the dataset, is a necessary part of such evaluation. Likewise, the question of the resources necessary to obtain a given set of results (such as computing power and time) is an important factor that is often neglected in the evaluations. How useful to clinical practice is a method that cannot be replicated without having access to large clusters of GPUs or TPUs, or putting clinical data into the hands of third-party, for-profit companies? The benefits of

[68] https://mitos-atypia-14.grand-challenge.org/Dataset/

more computational efficiency for the practical implementation of automated methods in digital pathology are certainly important.

A possible way to mitigate the uncertainty of the evaluation is to take a greater care in the annotation process for the test set. This can be done either by involving more pathologists to form a stronger consensus, or by directing them to focus most of their time and effort to the test set, possibly at the detriment of quality of the training set annotations. As the test set is generally smaller than the training set, this may be a good compromise to make the quantitative analysis of the results more trustworthy. While this will reduce the uncertainty, it will however not eliminate it. Properly analysing the uncertainty that remains is important for the discussion of the results. As was the case with the learning process, keeping trace of the individual annotations, when possible, will also enrich this discussion. It allows for comparisons not only to the consensus, but also to the individual experts. This can help make the distinction between algorithms that diverge from the consensus but remain within the realm of what some experts may predict, and algorithms that are mistaken in their own unique way. It also removes any bias that may come from the consensus mechanism itself, whether it is an automated process like STAPLE or a majority vote, or a consensus from a discussion. In the latter case, for instance, the seniority of the experts may have a large influence on their weight in the final decision, but so could other factors that influence teams' decisions, such as their persuasiveness or their gender [343], will be less correlated with their likelihood of being correct.

The choice of the evaluation metric itself is far from trivial, as we have shown in Chapter 4. It has to be informed by the nature of the task, by the characteristics of the dataset, and by the relative impact of different types of errors in the case of the clinical application where, for instance, confusion between some classes may be more or less desirable than for others. Digital pathology tasks are often complex and multi-faceted. Our analyses show that using simple independent metrics is generally preferable to trying to capture all the desired aspects in a single value. Simple metrics are easier to interpret and provide richer information on the relative merits of different methods.

The example of the "Panoptic Quality" for nuclei instance segmentation and classification clearly shows the risks of using a metric that is not adapted to the studied problem. The influence of object size on overlap-based metrics like the IoU can cause important issues in tasks such as nuclei segmentation. This, compounded by the confusion between the classification and detection definitions of the F1-Score, creates lots of possible sources of confusion in the interpretation of the results.

An over-reliance on quantitative metrics may therefore sometimes be detrimental to the best interpretation of the results. Qualitative analysis can be a richer source of information, although such analysis comes with its own constraints and biases. It is impractical when comparing many algorithms and requires the test set to remain relatively small. In tasks where the annotation process is particularly time-consuming, like segmentation, it may however be both more informative and easier to ask pathologists to compare the results of different algorithms (for instance by scoring how much they agree with each algorithm's prediction) than to ask them to provide a full set of annotations on which quantitative measurements could be made.

Another possibility to reduce the uncertainty of the annotations is to use some external source of validation for the test set annotations, which are not available to the algorithms. A common case is to have information from IHC slides or other modalities available to the test set annotators,

while the algorithm only has access to H&E-stained WSIs. The drawback of such a system is that, to compare the algorithms' performance to those of the experts, another independent expert panel needs to be constituted and given only the same information as the algorithm. This setup therefore largely increases the human resources necessary for conducting the study.

A common thread to this question of real-world annotations is the importance of keeping human experts in the loop through the entire process. The constitution of the dataset, the annotation process, the evaluation process and metrics, and the interpretation of the results, cannot be done without the active involvement of medical experts. To do otherwise increases the risk of wasting a lot of energy chasing performances which will not translate to practical improvements for the pathologist or the patient.

## 9.3 Improving digital pathology challenges

Several digital pathology challenges have been featured extensively in this thesis. The organisation of such challenges has been largely beneficial to the research community, by bringing attention to important tasks where automated image analysis may be of help, and by making large datasets available.

The complexity of organizing these challenges, however, makes the likelihood of mistakes or weaknesses relatively high. From the challenges that provide a large amount of transparency on their methods, datasets and/or evaluation code, it seems that mistakes may be relatively common. We identified the incorrect annotation maps in Gleason 2019 [5], the mistakes in the evaluation code of MoNuSAC 2020 [6], and mentioned in Chapter 8 the confusion between the stated task and the evaluation code in the Seg-PC 2021 challenge. Many competitions do not provide the information that allows other researchers to properly analyse their results.

This lack of transparency is an important weakness that have a detrimental impact on the trust that we can have on their results. The main hub that centralizes information on biomedical challenges today is the grand-challenge.org website[69]. Many challenges hosted on that platform, unfortunately, keep important information on their methodology and their data locked for participants only (or, for the test set annotations, sometimes for everyone) long after the challenge ended. When post-challenge publications are made, many challenges do not update their information to provide a reference to it. In our review of segmentation challenges, it sometimes required significant effort to find substantial information. In some cases, the websites have disappeared and only some snapshots on archiving services are available. As all of those challenges are rather recent, the oldest being the 2010 PR in HIMA competition, this trend is very worrying for the future.

The most visible part of the results, in most cases, is the "leaderboard", with the ranking of the participants and the values of the metric(s) chosen by the organizers on the predictions of the algorithms. In the absence of the test set annotations, and either the participants' predictions or the code of their algorithm, such a leaderboard is very limited in the information it provides. It is impossible, for instance, to recompute alternate metric(s) which may provide additional insights relevant to tasks adjacent to the one envisioned by the challenge organizers, or simply another point of view on that same task. If the evaluation code is not available, it is furthermore impossible to verify that the implementation of the evaluation corresponds to the published methodology. Aside from the potential for malicious manipulations of the results, the main risk is simply to have

[69] https://grand-challenge.org/

mistakes such as those made in the MoNuSAC challenge. Such mistakes are easy to make in digital pathology pipelines, where the evaluation methodology may involve multiple steps and heavily customized metrics not available in standard machine learning libraries. Trust requires replicability, and for challenges replicability requires transparency. Focusing on the leaderboard also emphasizes the competitive aspect of the challenges, rather than its scientific output, which is generally to be found in the discussion of the strengths and weaknesses of the proposed methods.

Transparency is not just good for future researchers. It can also outsource some of the responsibility for the quality control of the challenge to all participants. The mistakes in the MoNuSAC challenge evaluation, for instance, were visible to all participants from the start of the submission period, as all the code was publicly available. Encouraging challenge participants to audit the code and the data could be a good way to involve them as partners in the process of scientific discovery, rather than limiting their role to being competitors. The responsibility for quality control is also shared by the organizers of the conferences where challenges are often hosted, such as MICCAI or ISBI, and by reviewers and editors of journals where their results are published. While it may be unrealistic to expect reviewers or conference organizers to perform the quality control themselves, they could implement stronger requirements in terms of transparency and quality control procedures. While the efforts made by the MICCAI society with the requirements of the "BIAS" transparency reports [255] go in the right direction, replicability is still far from achievable from the information that challenges generally release.

A way forward for challenge organizers could be to move beyond adversarial competitions and towards a more cooperative dynamic. In current competitions, dozens or hundreds of participants compete in parallel, many with very similar solutions and encountering the same problems. The incentives are such that competitors are adversary between themselves, but also, in a way, with the challenge organizers: if errors exist in the data or evaluation code that can be exploited to obtain better results according to the challenge metrics, then the adversarial challenge model rewards the exploitation of those errors rather than their disclosure. Finding the right incentives to reward cooperation is certainly tricky. This could mean focusing the results on solutions rather than teams, with participants being able to contribute to different solutions and being rewarded according to those contributions. Instead of having multiple teams independently working on slight variations of the same pipeline, each intermediary result could then inform all the other participants on which directions may be of interest. The post-challenge publication should similarly be focused on the solutions and not on the teams, to emphasize the scientific insights rather than the often small differences in the quantitative metrics for the top teams. As an example, the PANDA challenge publication [187] does not contain any team rankings in the main text. The competition results are reported in supplementary materials, but the main article focuses on an overall perspective of the proposed methods and the shared characteristics of those that performed well.

## 9.4 Conclusions: predicting the future

Deep learning solutions to digital pathology tasks now routinely outperform train pathologists… if the tasks are performed in controlled settings. The gap between outperforming a pathologist on a curated dataset and being ready for deployment in a clinical setting, however, is as large as the gap between trusting a self-driving car on a private circuit and letting them run autonomously in the middle of a large city.

It is unlikely that deep learn methods will bridge that gap in the near future in the form of fully automated pathology pipeline. This, however, does not mean that such method cannot find a place in daily clinical or laboratory practice.

If automated methods should not be entrusted to pose an initial diagnosis, they could however serve as a line of defence against errors in a pathologist's diagnosis. As an automated method can operate in the background on any slide that is processed in a digital pathology pipeline, it can potentially flag cases where its automated diagnosis differs from that of the pathologist, prompting a potential second opinion, or review by the initial pathologist. For such a system to be accepted by clinicians, however, would probably require a great emphasis on the explainability of the algorithm: it is not enough to contradict a pathologist's opinion, the features or WSI region that prompted the alternative opinion should be highlighted. Pathologists could then for instance quickly assess if they potentially overlooked a useful part of the slide.

Tasks that are not as sensitive in nature as diagnosis but can otherwise facilitate or speed up the pathology workflow are also good candidates for automation. Artefact detection (and quality assessment on the slide preparation and acquisition) is a typical example. Such tasks can make the work of histology technologists and of pathologists easier, without any risk to the patient's outcome if some data fall outside of the algorithm's working domain, with potentially unpredictable outputs.

Finally, there is a large potential for deep learning methods as a way to find new biomarkers. Exploiting the very large amount of data from routine scanning of WSIs to find learned features that are correlated with patient outcome is a good way to potentially harness the strengths of deep learning methods with limited potential for harm. The development of such methods would therefore be focused on the exploration and understanding of the features and their potential relationship with biological processes rather than on their pure predictive performance.

As usage of machine learning methods in pathology becomes a reality, the ethical and regulatory issues surrounding deep learning also become more pressing. While most of these issues are largely outside of the scope of this thesis, they cannot be completely left aside. Several recent studies outline some of those issues [344], [345].

Selection biases in the datasets (such as having a large majority of white male subjects) can have a large impact on such data-centric methods as deep learning algorithms, which may perform poorly on samples outside of the majority population. Chauhan et al. [344] give the example of a commercial prediction algorithm used to identify high-risk patients based on several biomarkers, demographic information and known comorbidities. The algorithm was trained to predict healthcare costs for the patients, which was seen as a proxy for their healthcare needs. Obermeyer et al. [346], however, found that because the United States healthcare system spends less money on average for patients identifying as "Black" than for patients identifying as "White", the algorithm tended to predict lower costs for Black patients. This means that, based on the algorithm's recommendations, they would receive less additional care at similar risks level than White patients. This type of bias can be very difficult to discover in very complex deep learning models and highlight the importance of good data selection and of good model explainability.

Another important issue is the collection and usage of patient data. Deep learning models are improved by compiling very large datasets. These datasets may be further improved by linking patient information across multiple modalities: digital pathology WSIs, alongside radiological and genetic information, blood test results, and any available personal data. As more personal data is

collected, however, anonymisation of the data becomes more difficult, and risks of privacy violations become much more important (particularly, as noted in Sorell et al. [345], when the data contain information that relates to the patient's wealth, sexual orientation and practices, or political orientations). They contend, however, that while "absolutely irreversible deidentification is, if possible at all, very difficult" [345, p. 280], anonymized data in large datasets require in practice very sophisticated and expensive methods to be linked to known identities. Still, this at the very least require great care in the collection and curation of the dataset to avoid exposing sensitive personal data and violating existing privacy laws such as the GDPR.

The potential use of these data in commercial applications may pose a more serious ethical problem. Patients who agreed to the collection of their data for research purposes may not agree to the inclusion of these data in the model trained by a commercial entity in order to sell a product. The "informed consent" form of the TCGA project[70] warns that the collected data may "lead to the development of new diagnostic tests, new drugs or other commercial products". Publicly available datasets do not necessarily always disclose the exact terms that the patients agreed to, however. Furthermore, there is a difference between the data *leading* to the development of commercial products, and the data *being integrated* into the model used in a commercial product. Controlling on which data companies trained their model, and whether they respected license requirements, is extremely complicated. It is not yet clear that the current regulatory framework will be capable of dealing with all the questions that arise from the development of commercial products based on deep learning methods and digital pathology data.

Deep learning in digital pathology is a multi-disciplinary field, requiring the collaboration of scientists coming from very different worlds. To wildly caricature: machine learning scientists, used to nicely curated benchmark datasets and to results backed up by mathematics, meeting pathologists, working in a world of constantly evolving guidelines, and the messy variability and unpredictability of human diseases. The success of such a collaboration requires machine learning scientists to be better attuned to the realities of the clinical practice, and the impact of this reality on the datasets that they will work on. It also requires pathologists to better understand the strengths and limitations of the algorithms whose purpose is to help them in their decision making. If this cross-disciplinary understanding is absent, then we run the risk of producing models whose performances only appear good because of faulty dataset design or a poor understanding of the underlying clinical needs behind the computer vision task. Conversely, pathologists may not properly assess the extent to which some results may or may not be trusted and be led towards faulty diagnosis (or to stop trusting those results altogether).

The potential for improving patient care, however, is undeniable. By analysing the impact of real-world annotations on deep learning methods in digital pathology, we hope to help better understand the interaction between those two worlds. In this way, we can work towards ensuring that the large investment in time and resources by pathologists and machine learning scientists alike is well spent and moves the state-of-the-art in directions that best serve the needs of the patients.

[70] http://web.archive.org/web/20211015004430/https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga/history/policies/suggested-prospective-informed-consent.pdf [Archived on October 15th, 2021]

# References


[1] A. Foucart, O. Debeir, and C. Decaestecker, “Artifact Identification in Digital Pathology from Weak and Noisy Supervision with Deep Residual Networks,” in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, Nov. 2018, pp. 1–6. doi: 10.1109/CloudTech.2018.8713350.

[2] A. Foucart, O. Debeir, and C. Decaestecker, “SNOW: Semi-Supervised, Noisy And/Or Weak Data For Deep Learning In Digital Pathology,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Apr. 2019, pp. 1869–1872. doi: 10.1109/ISBI.2019.8759545.

[3] Y.-R. van Eycke, A. Foucart, and C. Decaestecker, “Strategies to Reduce the Expert Supervision Required for Deep Learning-Based Segmentation of Histopathological Images,” *Front Med (Lausanne)*, vol. 6, Oct. 2019, doi: 10.3389/fmed.2019.00222.

[4] A. Foucart, O. Debeir, and C. Decaestecker, “Snow Supervision in Digital Pathology: Managing Imperfect Annotations for Segmentation in Deep Learning,” 2020, doi: 10.21203/rs.3.rs-116512.

[5] A. Foucart, O. Debeir, and C. Decaestecker, “Processing multi-expert annotations in digital pathology: a study of the Gleason 2019 challenge,” in *17th International Symposium on Medical Information Processing and Analysis*, Dec. 2021, p. 4. doi: 10.1117/12.2604307.

[6] A. Foucart, O. Debeir, and C. Decaestecker, “Comments on ‘MoNuSAC2020: A Multi-Organ Nuclei Segmentation and Classification Challenge,’” *IEEE Trans Med Imaging*, vol. 41, no. 4, pp. 997–999, Apr. 2022, doi: 10.1109/TMI.2022.3156023.

[7] A. Foucart, O. Debeir, and C. Decaestecker, “Evaluating participating methods in image analysis challenges: lessons from MoNuSAC 2020,” 2022, doi: 10.13140/RG.2.2.11627.00801.

[8] A. Foucart, O. Debeir, and C. Decaestecker, “Shortcomings and areas for improvement in digital pathology image segmentation challenges,” 2022, doi: 10.13140/RG.2.2.32389.63200.

[9] J. M. S. Prewitt and M. L. Mendelsohn, “THE ANALYSIS OF CELL IMAGES*,” *Ann N Y Acad Sci*, vol. 128, no. 3, pp. 1035–1053, Dec. 1966, doi: 10.1111/j.1749-6632.1965.tb11715.x.

[10] S. Doyle, M. Hwang, K. Shah, A. Madabhushi, M. Feldman, and J. Tomaszeweski, “AUTOMATED GRADING OF PROSTATE CANCER USING ARCHITECTURAL AND TEXTURAL IMAGE FEATURES,” in *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2007, pp. 1284–1287. doi: 10.1109/ISBI.2007.357094.

[11] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, “Histopathological image analysis: a review.,” *IEEE Rev Biomed Eng*, vol. 2, pp. 147–71, Jan. 2009, doi: 10.1109/RBME.2009.2034865.

[12] L. Pantanowitz, “Digital images and the future of digital pathology,” *J Pathol Inform*, vol. 1, no. 1, p. 15, 2010, doi: 10.4103/2153-3539.68332.

[13] S. Al-Janabi, A. Huisman, and P. J. van Diest, "Digital pathology: current status and future perspectives," *Histopathology*, vol. 61, no. 1, pp. 1–9, Jul. 2012, doi: 10.1111/j.1365-2559.2011.03814.x.

[14] L. Pantanowitz, A. Sharma, A. Carter, T. Kurc, A. Sussman, and J. Saltz, "Twenty years of digital pathology: An overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives," *J Pathol Inform*, vol. 9, no. 1, p. 40, 2018, doi: 10.4103/jpi.jpi_69_18.

[15] M. Garland *et al.*, "Parallel Computing Experiences with CUDA," *IEEE Micro*, vol. 28, no. 4, pp. 13–27, Jul. 2008, doi: 10.1109/MM.2008.57.

[16] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative Study of Deep Learning Software Frameworks," Nov. 2015, Accessed: Aug. 25, 2022. [Online]. Available: https://arxiv.org/abs/1511.06435

[17] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2013*, 2013, pp. 411–418.

[18] A. Cruz-Roa *et al.*, "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks," in *Medical Imaging 2014: Digital Pathology*, Mar. 2014, vol. 9041, no. 216, p. 904103. doi: 10.1117/12.2043872.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[20] C. Malon and E. Cosatto, "Classification of mitotic figures with convolutional neural networks and seeded blob features," *J Pathol Inform*, vol. 4, no. 1, p. 9, 2013, doi: 10.4103/2153-3539.112694.

[21] J. van der Laak, G. Litjens, and F. Ciompi, "Deep learning in histopathology: the path to the clinic," *Nat Med*, vol. 27, no. 5, pp. 775–784, May 2021, doi: 10.1038/s41591-021-01343-4.

[22] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int J Comput Vis*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[23] R. Verma, N. Kumar, A. Patil, N. C. Kurian, S. Rane, and A. Sethi, "Author's Reply to 'MoNuSAC2020: A Multi-Organ Nuclei Segmentation and Classification Challenge,'" *IEEE Trans Med Imaging*, vol. 41, no. 4, pp. 1000–1003, Apr. 2022, doi: 10.1109/TMI.2022.3157048.

[24] R. Verma *et al.*, "MoNuSAC2020: A Multi-Organ Nuclei Segmentation and Classification Challenge," *IEEE Trans Med Imaging*, vol. 40, no. 12, pp. 3413–3423, Dec. 2021, doi: 10.1109/TMI.2021.3085712.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[26] W. Buntine, "Machine learning after the deep learning revolution," *Front Comput Sci*, vol. 14, no. 6, 2020, doi: 10.1007/s11704-020-0800-8.

[27] T. Sejnowski, *The Deep Learning Revolution*. MIT Press, 2018.

[28] J. Kelleher, *Deep Learning*, no. 9. MIT Press, 2019.

[29] M. I. Razzak, S. Naz, and A. Zaib, "Deep Learning for Medical Image Processing: Overview, Challenges and the Future," 2018, pp. 323–350. doi: 10.1007/978-3-319-65981-7_12.

[30] M. Salto-Tellez, P. Maxwell, and P. W. Hamilton, "Artificial Intelligence - The Third Revolution in Pathology," *Histopathology*, 2018, doi: 10.1111/his.13760.

[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[32] J. Schmidhuber, "Deep learning in neural networks : An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[33] Q. V Le *et al.*, "Building high-level features using large scale unsupervised learning," *International Conference in Machine Learning*, 2012, doi: 10.1109/ICASSP.2013.6639343.

[34] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," *Applied Sciences*, no. February, p. 20, 2012, doi: 10.1109/CVPR.2012.6248110.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv Neural Inf Process Syst*, pp. 1–9, 2012.

[36] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int J Comput Vis*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[37] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[38] K. Fukushima, "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biol Cybern*, vol. 36, pp. 193–202, 1980.

[39] W. S. McCulloch and W. H. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[40] D. Hebb, *The organization of behavior*. New York: Wiley, 1949.

[41] F. Rosenblatt, "The Perceptron - A Perceiving and Recognizing Automaton," Buffaly, N. Y., Jan. 1957. Accessed: Aug. 25, 2022. [Online]. Available: https://blogs.umass.edu/brain-wars/files/2016/03/rosenblatt-1957.pdf

[42] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington DC: Spartan Books, 1962. doi: 10.2307/1419730.

[43] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Trans Syst Man Cybern*, no. 4, pp. 364–378, 1971.

[44] P. J. Werbos, "Applications of Advances in Nonlinear Sensitivity Analysis," in *Proceedings of the 10th IFIP Conference, 31.8 - 4.9, NYC*, 1981, pp. 762–770.

[45] P. J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Harvard University, 1974.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 9, 1986.

[47] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Comput*, vol. 1, no. 4, pp. 541–551, 1989, doi: 10.1162/neco.1989.1.4.541.

[48] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[49] R. Raina, A. Madhavan, and A. Y. Ng, “Large-scale deep unsupervised learning using graphics processors,” *ACM International Conference Proceeding Series*, vol. 382, 2009, doi: 10.1145/1553374.1553486.

[50] A. M. Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, 1937, doi: 10.1112/plms/s2-42.1.230.

[51] P. Husbands and O. Holland, “Warren McCulloch and the British cyberneticians,” *Interdisciplinary Science Reviews*, vol. 37, no. 3, pp. 237–253, 2012, doi: 10.1179/0308018812Z.00000000019.

[52] H. M. Sheffer, “A set of five independent postulates for Boolean algebras, with application to logical constants,” *Trans Am Math Soc*, vol. 14, no. 4, pp. 481–488, 1913, doi: 10.1090/S0002-9947-1913-1500960-1.

[53] F. Rosenblatt, “The Perceptron: a probabilistic model for information storage and organization in the brain,” *Psychol Rev*, vol. 65, no. 6, pp. 386–408, 1958.

[54] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, “An artificial neuron implemented on an actual quantum processor,” *npj Quantum Inf*, vol. 5, no. 1, p. 26, Dec. 2019, doi: 10.1038/s41534-019-0140-4.

[55] M. Olazaran, “A Sociological Study of the Official History of the Perceptrons Controversy,” *Soc Stud Sci*, vol. 26, pp. 611–59, 1996, Accessed: Aug. 25, 2022. [Online]. Available: http://www.jstor.org/stable/285702

[56] M. L. Minsky and S. Papert, *Perceptrons, An Introduction to Computational Geometry*. MIT Press, 1969.

[57] P. J. Werbos, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,” Ph.D. Thesis, Harvard University, Cambridge, 1974.

[58] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 9, 1986.

[59] J. Schmidhuber, “Deep Learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[60] Y. Le Cun *et al.*, “Handwritten Digit Recognition with a Back-Propagation Network,” *Adv Neural Inf Process Syst*, pp. 396–404, 1990, doi: 10.1111/dsu.12130.

[61] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[62] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proc. 27th ICML*, 2010, pp. 807–814.

[63] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, p. 6, 2013.

[64] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway Networks,” 2015. [Online]. Available: http://arxiv.org/abs/1505.00387

[65] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks Importance of Identity Skip Connections Usage of Activation Function Analysis of Pre-activation Structure,” 2016. doi: 10.1007/978-3-319-46493-0_38.

[66] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[67] S. Sagiroglu and D. Sinanc, “Big data: A review,” *Int’l Conf on Collaboration Technologies and Systems (CTS)*, 2013, doi: 10.1109/CTS.2013.6567202.

[68] R. Vuduc and J. Choi, “A Brief History and Introduction to GPGPU,” in *Modern Accelerator Technologies for Geographic Information Science*, Springer, 2013, pp. 9–23.

[69] J. Bergstra *et al.*, “Theano: Deep Learning on GPUs with Python,” *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2011.

[70] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

[71] M. Abadi and Others, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015. Accessed: Aug. 25, 2022. [Online]. Available: https://arxiv.org/abs/1603.04467

[72] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” *Adv Neural Inf Process Syst*, vol. 32, no. NeurIPS, 2019.

[73] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[74] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR*, vol. 9, pp. 249–256, 2010.

[75] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015, doi: 10.1109/ICCV.2015.123.

[76] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *CoRR*, vol. abs/1412.6, pp. 1–15, 2014, doi: 10.1145/1830483.1830503.

[77] G. Hinton, N. Srivastava, and K. Swersky, “Neural Networks for Machine Learning - Lecture 6a,” 2012. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[78] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," Dec. 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[79] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic Gradient Descent as Approximate Bayesian Inference," Apr. 2017. [Online]. Available: http://arxiv.org/abs/1704.04289

[80] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," Sep. 2016. [Online]. Available: http://arxiv.org/abs/1609.04836

[81] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug. 2014, pp. 661–670. doi: 10.1145/2623330.2623612.

[82] B. Hanin and D. Rolnick, "How to Start Training: The Effect of Initialization and Architecture," 2018. Accessed: Aug. 25, 2022. [Online]. Available: https://arxiv.org/abs/1803.01719

[83] R. M. Schmidt, F. Schneider, and P. Hennig, "Descending through a Crowded Valley - Benchmarking Deep Learning Optimizers," in *Proc. 38th Int'l Conf Machine Learning, PMLR*, 2020, pp. 9367–9376. [Online]. Available: http://arxiv.org/abs/2007.01547

[84] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[85] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis," *Med Image Anal*, vol. 65, p. 101759, Oct. 2020, doi: 10.1016/j.media.2020.101759.

[86] N. Akhtar and U. Ragavendran, "Interpretation of intelligence in CNN-pooling processes: a methodological survey," *Neural Comput Appl*, vol. 32, no. 3, pp. 879–898, Feb. 2020, doi: 10.1007/s00521-019-04296-5.

[87] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2018–2025, 2011, doi: 10.1109/ICCV.2011.6126474.

[88] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 12, pp. 2481–2495, 2017.

[89] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[90] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Feb. 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[91] F. Almasri and O. Debeir, "Multimodal Sensor Fusion In Single Thermal image Super-Resolution," in *Computer Vision – ACCV 2018 Workshops*, 2018, pp. 418–433. doi: 10.1007/978-3-030-21074-8_34.

[92] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep Convolutional AutoEncoder-based Lossy Image Compression," in *2018 Picture Coding Symposium (PCS)*, Jun. 2018, pp. 253–257. doi: 10.1109/PCS.2018.8456308.

[93] N. M. N. Leite, E. T. Pereira, E. C. Gurjao, and L. R. Veloso, "Deep Convolutional Autoencoder for EEG Noise Filtering," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2018, pp. 2605–2612. doi: 10.1109/BIBM.2018.8621080.

[94] R. Togo, H. Watanabe, T. Ogawa, and M. Haseyama, "Deep convolutional neural network-based anomaly detection for organ classification in gastric X-ray examination," *Comput Biol Med*, vol. 123, p. 103903, Aug. 2020, doi: 10.1016/j.compbiomed.2020.103903.

[95] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1646–1654.

[96] F. P. dos Santos, C. Zor, J. Kittler, and M. A. Ponti, "Learning image features with fewer labels using a semi-supervised deep convolutional network," *Neural Networks*, vol. 132, pp. 131–143, Dec. 2020, doi: 10.1016/j.neunet.2020.08.016.

[97] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv Neural Inf Process Syst*, vol. 27, 2014.

[98] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, no. February, pp. 3642–3649. doi: 10.1109/CVPR.2012.6248110.

[99] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, vol. 11-18-Dece, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[100] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[101] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2980–2988. doi: 10.1109/ICCV.2017.322.

[102] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, "DCAN: Deep contour-aware networks for object instance segmentation from histology images," *Med Image Anal*, vol. 36, pp. 135–146, Feb. 2017, doi: 10.1016/j.media.2016.11.004.

[103] S. Graham *et al.*, "Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Med Image Anal*, vol. 58, p. 101563, Dec. 2019, doi: 10.1016/j.media.2019.101563.

[104] A. Moyes, R. Gault, K. Zhang, J. Ming, D. Crookes, and J. Wang, "Multi-Channel Auto-Encoders and a Novel Dataset for Learning Domain Invariant Representations of Histopathology Images," Jul. 2021. [Online]. Available: http://arxiv.org/abs/2107.07271

[105] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.06434

[106] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[107] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," Nov. 2016. [Online]. Available: http://arxiv.org/abs/1611.05431

[108] A. Zela, A. Klein, S. Falkner, and F. Hutter, "Towards Automated Deep Learning: Efficient Joint Neural Architecture and Hyperparameter Search," Jul. 2018. [Online]. Available: http://arxiv.org/abs/1807.06906

[109] X. Xiao, M. Yan, S. Basodi, C. Ji, and Y. Pan, "Efficient Hyperparameter Optimization in Deep Learning Using a Variable Length Genetic Algorithm," Jun. 2020. [Online]. Available: http://arxiv.org/abs/2006.12703

[110] J.-Y. Kim and S.-B. Cho, "Evolutionary Optimization of Hyperparameters in Deep Learning Models," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2019, pp. 831–837. doi: 10.1109/CEC.2019.8790354.

[111] Y. Sucaet and W. Waelput, *Digital Pathology*. Springer, 2014. doi: 10.1159/isbn.978-3-318-05846-8.

[112] M. May, "A better lens on disease," *Sci Am*, vol. 302, no. 5, pp. 74–77, 2010, doi: 10.1038/scientificamerican0510-74.

[113] W. E. Tolles, "SECTION OF BIOLOGY: THE CYTOANALYZER-AN EXAMPLE OF PHYSICS IN MEDICAL RESEARCH," *Trans N Y Acad Sci*, vol. 17, no. 3 Series II, pp. 250–256, Jan. 1955, doi: 10.1111/j.2164-0947.1955.tb01204.x.

[114] C. C. Spencer and R. C. Bostrom, "Performance of the cytoanalyzer in recent clinical trials," *J Natl Cancer Inst*, vol. 29, no. 2, pp. 267–276, 1962, doi: 10.1093/jnci/29.2.267.

[115] A. I. Spriggs, "Automatic scanning for cervical smears.," *J Clin Pathol*, vol. s2-3, no. 1, pp. 1–7, Jan. 1969, doi: 10.1136/jcp.s2-3.1.1.

[116] M. L. Mendelsohn, W. A. Kolman, and R. C. Bostrom, "INITIAL APPROACHES TO THE COMPUTER ANALYSIS OF CYTOPHOTOMETRIC FIELDS*," *Ann N Y Acad Sci*, vol. 115, no. 2, pp. 998–1009, Jul. 1964, doi: 10.1111/j.1749-6632.1964.tb00071.x.

[117] M. L. Mendelsohn, W. A. Kolman, B. Perry, and J. M. S. Prewitt, "Computer Analysis of Cell Images," *Postgrad Med*, vol. 38, no. 5, pp. 567–573, Nov. 1965, doi: 10.1080/00325481.1965.11695692.

[118] R. Weinstein, M. Holcomb, and E. Krupinski, "Invention and early history of telepathology (1985-2000)," *J Pathol Inform*, vol. 10, no. 1, p. 1, 2019, doi: 10.4103/jpi.jpi_71_18.

[119] E. Krupinski, A. Bhattacharyya, and R. Weinstein, "Telepathology and Digital Pathology Research," in *Digital Pathology*, 2016. doi: 10.1159/isbn.978-3-318-05846-8.

[120] J. Greene, "When Television Was a Medical Device," *Humanities, Volume 38 (2)*, 2017.

[121] J. Ho, A. V. Parwani, D. M. Jukic, Y. Yagi, L. Anthony, and J. R. Gilbertson, "Use of whole slide imaging in surgical pathology quality assurance: Design and pilot validation studies," *Hum Pathol*, vol. 37, no. 3, pp. 322–331, 2006, doi: 10.1016/j.humpath.2005.11.005.

[122] L. Pantanowitz *et al.*, "Review of the current state of whole slide imaging in pathology," *J Pathol Inform*, vol. 2, no. 1, p. 36, 2011, doi: 10.4103/2153-3539.83746.

[123] M. D. Zarella *et al.*, "A Practical Guide to Whole Slide Imaging: A White Paper From the Digital Pathology Association," *Arch Pathol Lab Med*, vol. 143, no. 2, pp. 222–234, Feb. 2019, doi: 10.5858/arpa.2018-0343-RA.

[124] F. Ghaznavi, A. Evans, A. Madabhushi, and M. Feldman, "Digital imaging in pathology: Whole-slide imaging and beyond," *Annual Review of Pathology: Mechanisms of Disease*, vol. 8, no. October, pp. 331–359, 2013, doi: 10.1146/annurev-pathol-011811-120902.

[125] M. Satyanarayanan, A. Goode, B. Gilbert, J. Harkes, and D. Jukic, "OpenSlide: A vendor-neutral software foundation for digital pathology," *J Pathol Inform*, vol. 4, no. 1, p. 27, 2013, doi: 10.4103/2153-3539.119005.

[126] D. Wolfe, "Tissue processing," in *Bancroft's Theory and Practice of Histological Techniques (8th ed)*, Elsevier, 2019, pp. 73--83.

[127] M. Titford, "Progress in the development of microscopical techniques for diagnostic pathology," *J Histotechnol*, vol. 32, no. 1, pp. 9–19, 2009, doi: 10.1179/his.2009.32.1.9.

[128] R. W. Horobin, "Theory of histological staining," in *Bancroft's Theory and Practice of Histological Techniques (8th ed)*, Elsevier, 2019.

[129] J. Bury and J. Griffin, "Digital pathology," in *Bancroft's Theory and Practice of Histological Techniques (8th ed)*, Elsevier, 2019, pp. 476--492.

[130] D. Hartman, JeroenA. W. M. Van Der Laak, M. Gurcan, and L. Pantanowitz, "Value of public challenges for the development of pathology deep learning algorithms," *J Pathol Inform*, vol. 11, no. 7, 2020, doi: 10.4103/jpi.jpi_64_19.

[131] P. W. Hamilton *et al.*, "Digital pathology and image analysis in tissue biomarker research," *Methods*, vol. 70, no. 1, pp. 59–73, 2014, doi: 10.1016/j.ymeth.2014.06.015.

[132] A. Madabhushi and G. Lee, "Image analysis and machine learning in digital pathology: Challenges and opportunities," *Med Image Anal*, vol. 33, pp. 170–175, 2016, doi: 10.1016/j.media.2016.06.037.

[133] S. B. Edge and C. C. Compton, "The American Joint Committee on Cancer: the 7th Edition of the AJCC Cancer Staging Manual and the Future of TNM," *Ann Surg Oncol*, vol. 17, no. 6, pp. 1471–1474, Jun. 2010, doi: 10.1245/s10434-010-0985-4.

[134] F. L. Greene *et al.*, *AJCC Cancer Staging Manual, 8th edition*. Springer, 2018.

[135] D. F. Gleason and G. T. Mellinger, "Prediction of Prognosis for Prostatic Adenocarcinoma by Combined Histological Grading and Clinical Staging," *Journal of Urology*, vol. 111, no. 1, pp. 58–64, Jan. 1974, doi: 10.1016/S0022-5347(17)59889-4.

[136] J. I. Epstein *et al.*, "A Contemporary Prostate Cancer Grading System: A Validated Alternative to the Gleason Score," *Eur Urol*, vol. 69, no. 3, pp. 428–435, Mar. 2016, doi: 10.1016/j.eururo.2015.06.046.

[137] C. W. Elston and I. O. Ellis, "Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with long-term follow-up.," *Histopathology*, vol. 19, no. 5, pp. 403–410, Nov. 1991.

[138] J. I. Epstein, L. Egevad, M. B. Amin, B. Delahunt, J. R. Srigley, and P. A. Humphrey, "The 2014 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma," *American Journal of Surgical Pathology*, vol. 40, no. 2, pp. 244–252, Feb. 2016, doi: 10.1097/PAS.0000000000000530.

[139] L. L. De Matos, D. C. Trufelli, M. G. L. De Matos, and M. A. Da Silva Pinhal, "Immunohistochemistry as an Important Tool in Biomarkers Detection and Clinical Practice," *Biomark Insights*, vol. 5, p. BMI.S2185, Jan. 2010, doi: 10.4137/BMI.S2185.

[140] C. Verocq *et al.*, "The daily practice reality of PD-L1 (CD274) evaluation in non-small cell lung cancer: A retrospective study," *Oncol Lett*, Mar. 2020, doi: 10.3892/ol.2020.11458.

[141] M. N. Gurcan, A. Madabhushi, and N. Rajpoot, "Pattern Recognition in Histopathological Images: An ICPR 2010 Contest," in *Lecture Notes in Computer Science 6388*, 2010, pp. 226–234. doi: 10.1007/978-3-642-17711-8_23.

[142] T. J. Fuchs and J. M. Buhmann, "Computational pathology: Challenges and promises for tissue analysis," *Computerized Medical Imaging and Graphics*, vol. 35, no. 7–8, pp. 515–530, Oct. 2011, doi: 10.1016/j.compmedimag.2011.02.006.

[143] M. Macenko *et al.*, "A method for normalizing histology slides for quantitative analysis," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Jun. 2009, pp. 1107–1110. doi: 10.1109/ISBI.2009.5193250.

[144] A. Ruifrok and D. Johnston, "Quantification of histochemical staining by color deconvolution.," *Analytical and quantitative citology and histology*, vol. 23, no. 4, pp. 291–299, 2001.

[145] Y.-R. Van Eycke, J. Allard, I. Salmon, O. Debeir, and C. Decaestecker, "Image processing in digital pathology: an opportunity to solve inter-batch variability of immunohistochemical staining OPEN," *Sci Rep*, vol. 7, Mar. 2017, doi: 10.1038/srep42964.

[146] Y. R. Van Eycke, C. Balsat, L. Verset, O. Debeir, I. Salmon, and C. Decaestecker, "Segmentation of glandular epithelium in colorectal tumours to automatically compartmentalise IHC biomarker quantification: A deep learning approach," *Med Image Anal*, vol. 49, pp. 35–45, 2018, doi: 10.1016/j.media.2018.07.004.

[147] S. Naik, S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, "Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, May 2008, pp. 284–287. doi: 10.1109/ISBI.2008.4540988.

[148] A. Janowczyk and A. Madabhushi, “Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases,” *J Pathol Inform*, vol. 7, no. 1, 2016, doi: 10.4103/2153-3539.186902.

[149] L. E. Boucheron, “Object- and Spatial-Level Quantitative Analysis of Multispectral Histopathology Images for Detection and Characterization of Cancer,” Ph.D. dissertation, Ph.D. dissertation at Univ. of California Santa Barbara, 2008. Accessed: Aug. 25, 2022. [Online]. Available: https://vision.ece.ucsb.edu/abstract/518

[150] M. Kuse, T. Sharma, and S. Gupta, “A Classification Scheme for Lymphocyte Segmentation in H&E Stained Histology Images,” in *Lecture Notes in Computer Science*, vol. 6388 LNCS, 2010, pp. 235–243. doi: 10.1007/978-3-642-17711-8_24.

[151] C. Panagiotakis, E. Ramasso, and G. Tziritas, “Lymphocyte Segmentation Using the Transferable Belief Model,” in *Lecture Notes in Computer Science*, 2010, pp. 253–262. doi: 10.1007/978-3-642-17711-8_26.

[152] E. J. Kaman, A. W. M. Smeulders, P. W. Verbeek, I. T. Young, and J. P. A. Baak, “Image processing for mitoses in sections of breast cancer: A feasibility study,” *Cytometry*, vol. 5, no. 3, pp. 244–249, May 1984, doi: 10.1002/cyto.990050305.

[153] T. K. ten Kate, J. A. M. Beliën, A. W. M. Smeulders, and J. P. A. Baak, “Method for counting mitoses by image processing in feulgen stained breast cancer sections,” *Cytometry*, vol. 14, no. 3, pp. 241–250, 1993, doi: 10.1002/cyto.990140302.

[154] J. A. M. Beliën, J. P. A. Baak, P. J. van Diest, and A. H. M. van Ginkel, “Counting mitoses by image processing in Feulgen stained breast cancer sections: The influence of resolution,” *Cytometry*, vol. 28, no. 2, pp. 135–140, Jun. 1997, doi: 10.1002/(SICI)1097-0320(19970601)28:2<135::AID-CYTO6>3.0.CO;2-E.

[155] J.-R. Dalle, W. K. Leow, D. Racoceanu, A. E. Tutac, and T. C. Putti, “Automatic breast cancer grading of histopathological images,” in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2008, pp. 3052–3055. doi: 10.1109/IEMBS.2008.4649847.

[156] A. Paul and D. P. Mukherjee, “Mitosis Detection for Invasive Breast Cancer Grading in Histopathological Images,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4041–4054, Nov. 2015, doi: 10.1109/TIP.2015.2460455.

[157] G. Nir *et al.*, “Automatic grading of prostate cancer in digitized histopathology images: Learning from multiple experts,” *Med Image Anal*, vol. 50, pp. 167–180, Dec. 2018, doi: 10.1016/j.media.2018.09.005.

[158] D. N. Louis *et al.*, “Computational Pathology: An Emerging Definition,” *Arch Pathol Lab Med*, vol. 138, no. 9, pp. 1133–1138, Sep. 2014, doi: 10.5858/arpa.2014-0034-ED.

[159] C. Malon, M. Miller, H. C. Burger, E. Cosatto, and H. P. Graf, “Identifying histological elements with convolutional neural networks,” *5th International Conference on Soft Computing as Transdisciplinary Science and Technology, CSTST ’08 - Proceedings*, pp. 450–456, 2008, doi: 10.1145/1456223.1456316.

[160] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks," in *Lecture Notes in Computer Science*, vol. 8150, no. 2, 2013, pp. 411–418. doi: 10.1007/978-3-642-40763-5_51.

[161] C. Malon and E. Cosatto, "Classification of mitotic figures with convolutional neural networks and seeded blob features," *J Pathol Inform*, vol. 4, no. 1, p. 9, 2013, doi: 10.4103/2153-3539.112694.

[162] A. A. Cruz-Roa, J. E. Arevalo Ovalle, A. Madabhushi, and F. A. González Osorio, "A Deep Learning Architecture for Image Representation, Visual Interpretability and Automated Basal-Cell Carcinoma Cancer Detection," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2013*, vol. 8150, 2013, pp. 403–410. doi: 10.1007/978-3-642-40763-5_50.

[163] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med Image Anal*, vol. 42, no. December 2012, pp. 60–88, 2017, doi: 10.1016/j.media.2017.07.005.

[164] L. Maier-Hein *et al.*, "Why rankings of biomedical image analysis competitions should be interpreted with care," *Nat Commun*, vol. 9, no. 1, p. 5217, Dec. 2018, doi: 10.1038/s41467-018-07619-7.

[165] M. N. Gurcan, A. Madabhushi, and N. Rajpoot, "Pattern Recognition in Histopathological Images: An ICPR 2010 Contest," in *Lecture Notes in Computer Science*, vol. 6388, 2010, pp. 226–234. doi: 10.1007/978-3-642-17711-8_23.

[166] L. Roux *et al.*, "Mitosis detection in breast cancer histological images An ICPR 2012 contest," *J Pathol Inform*, vol. 4, no. 1, p. 8, 2013, doi: 10.4103/2153-3539.112693.

[167] M. Veta *et al.*, "Assessment of algorithms for mitosis detection in breast cancer histopathology images," *Med Image Anal*, vol. 20, no. 1, pp. 237–248, Feb. 2015, doi: 10.1016/j.media.2014.11.010.

[168] K. Sirinukunwattana, J. P. W. Pluim, H. Chen, and Others, "Gland segmentation in colon histology images: The glas challenge contest," *Med Image Anal*, vol. 35, pp. 489–502, 2017, doi: 10.1016/j.media.2016.08.008.

[169] M. Veta *et al.*, "Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge," *Med Image Anal*, vol. 54, pp. 111–121, May 2019, doi: 10.1016/j.media.2019.02.012.

[170] B. Ehteshami Bejnordi *et al.*, "Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer," *JAMA*, vol. 318, no. 22, p. 2199, Dec. 2017, doi: 10.1001/jama.2017.14585.

[171] T. Qaiser *et al.*, "HER2 challenge contest: a detailed assessment of automated HER2 scoring algorithms in whole slide images of breast cancer tissues," *Histopathology*, vol. 72, no. 2, pp. 227–238, Jan. 2018, doi: 10.1111/his.13333.

[172] C.-W. Wang *et al.*, "A benchmark for comparing precision medicine methods in thyroid cancer diagnosis using tissue microarrays," *Bioinformatics*, vol. 34, no. 10, pp. 1767–1773, May 2018, doi: 10.1093/bioinformatics/btx838.

[173] P. Bandi *et al.*, "From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient Level: The CAMELYON17 Challenge," *IEEE Trans Med Imaging*, vol. 38, no. 2, pp. 550–560, Feb. 2019, doi: 10.1109/TMI.2018.2867350.

[174] Q. D. Vu *et al.*, "Methods for Segmentation and Classification of Digital Microscopy Tissue Images," *Front Bioeng Biotechnol*, vol. 7, Apr. 2019, doi: 10.3389/fbioe.2019.00053.

[175] T. Kurc *et al.*, "Segmentation and Classification in Digital Pathology for Glioma Research: Challenges and Deep Learning Approaches," *Front Neurosci*, vol. 14, Feb. 2020, doi: 10.3389/fnins.2020.00027.

[176] G. Aresta *et al.*, "BACH: Grand challenge on breast cancer histology images," *Med Image Anal*, vol. 56, pp. 122–139, 2019, doi: 10.1016/j.media.2019.05.010.

[177] N. Kumar *et al.*, "A Multi-Organ Nucleus Segmentation Challenge," *IEEE Trans Med Imaging*, vol. 39, no. 5, pp. 1380–1391, 2020, doi: 10.1109/TMI.2019.2947628.

[178] A. Gupta and R. Gupta, Eds., *ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging*. Singapore: Springer Singapore, 2019. doi: 10.1007/978-981-15-0798-4.

[179] N. Petrick *et al.*, "SPIE-AAPM-NCI BreastPathQ challenge: an image analysis challenge for quantitative tumor cellularity assessment in breast cancer histology images following neoadjuvant treatment," *Journal of Medical Imaging*, vol. 8, no. 03, May 2021, doi: 10.1117/1.JMI.8.3.034501.

[180] Z. Li *et al.*, "Deep Learning Methods for Lung Cancer Segmentation in Whole-Slide Histopathology Images - The ACDC@LungHP Challenge 2019," *IEEE J Biomed Health Inform*, vol. 25, no. 2, pp. 429–440, 2021, doi: 10.1109/JBHI.2020.3039741.

[181] Z. Swiderska-Chadaj *et al.*, "Learning to detect lymphocytes in immunohistochemistry with deep learning," *Med Image Anal*, vol. 58, p. 101547, Dec. 2019, doi: 10.1016/j.media.2019.101547.

[182] Y. J. Kim *et al.*, "PAIP 2019: Liver cancer segmentation challenge," *Med Image Anal*, vol. 67, p. 101854, 2021, doi: 10.1016/j.media.2020.101854.

[183] C. Zhu *et al.*, "Multi-level colonoscopy malignant tissue detection with adversarial CAC-UNet," *Neurocomputing*, vol. 438, pp. 165–183, May 2021, doi: 10.1016/j.neucom.2020.04.154.

[184] M. Amgad *et al.*, "Structured crowdsourcing enables convolutional segmentation of histology images," *Bioinformatics*, vol. 35, no. 18, pp. 3461–3467, 2019, doi: 10.1093/bioinformatics/btz083.

[185] J. Borovec *et al.*, "ANHIR: Automatic Non-Rigid Histological Image Registration Challenge," *IEEE Trans Med Imaging*, vol. 39, no. 10, pp. 3042–3052, Oct. 2020, doi: 10.1109/TMI.2020.2986331.

[186] E. Conde-Sousa *et al.*, "HEROHE Challenge: assessing HER2 status in breast cancer without immunohistochemistry or in situ hybridization," Nov. 2021. [Online]. Available: http://arxiv.org/abs/2111.04738

[187] W. Bulten *et al.*, "Artificial intelligence for diagnosis and Gleason grading of prostate cancer: the PANDA challenge," *Nat Med*, vol. 28, no. 1, pp. 154–163, Jan. 2022, doi: 10.1038/s41591-021-01620-2.

[188] M. Amgad *et al.*, "NuCLS: A scalable crowdsourcing approach and dataset for nucleus classification and segmentation in breast cancer," *Gigascience*, vol. 11, no. Cche 57357, pp. 1–45, May 2022, doi: 10.1093/gigascience/giac037.

[189] M. Aubreville *et al.*, "Mitosis domain generalization in histopathology images -- The MIDOG challenge," 2022. [Online]. Available: https://www.researchgate.net/publication/359865353

[190] H. Irshad, "Automated mitosis detection in histopathology using morphological and multi-channel statistics features.," *J Pathol Inform*, vol. 4, p. 10, 2013, doi: 10.4103/2153-3539.112695.

[191] E. Gruwé, "Développement d'une approche de Deep Learning pour la détection automatique de mitoses dans des lames histologiques," Master Thesis, Université Libre de Bruxelles, Brussels, 2018.

[192] H. Chen, Q. Dou, X. Wang, J. Qin, and P. A. Heng, "Mitosis Detection in Breast Cancer Histology Images via Deep Cascaded Networks," *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1160–1166, 2016.

[193] Y. Jia *et al.*, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, Nov. 2014, pp. 675–678. doi: 10.1145/2647868.2654889.

[194] K. Paeng, S. Hwang, S. Park, and M. Kim, "A Unified Framework for Tumor Proliferation Score Prediction in Breast Histopathology," Dec. 2016. [Online]. Available: http://arxiv.org/abs/1612.07180

[195] R. Nateghi, H. Danyali, and M. S. Helfroush, "Maximized Inter-Class Weighted Mean for Fast and Accurate Mitosis Cells Detection in Breast Cancer Histopathology Images," *J Med Syst*, vol. 41, no. 9, p. 146, Sep. 2017, doi: 10.1007/s10916-017-0773-9.

[196] A. Albayrak and G. Bilgin, "Mitosis detection using convolutional neural network based features," in *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, Nov. 2016, pp. 000335–000340. doi: 10.1109/CINTI.2016.7846429.

[197] D. Cai, X. Sun, N. Zhou, X. Han, and J. Yao, "Efficient Mitosis Detection in Breast Cancer Histology Images by RCNN," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Apr. 2019, pp. 919–922. doi: 10.1109/ISBI.2019.8759461.

[198] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, 2015, vol. 28.

[199] T. Mahmood, M. Arsalan, M. Owais, M. B. Lee, and K. R. Park, "Artificial Intelligence-Based Mitosis Detection in Breast Cancer Histopathology Images Using Faster R-CNN and Deep CNNs," *J Clin Med*, vol. 9, no. 3, p. 749, Mar. 2020, doi: 10.3390/jcm9030749.

[200] S. Yang, F. Luo, J. Zhang, and X. Wang, “Sk-Unet Model with Fourier Domain for Mitosis Detection,” Sep. 2021. [Online]. Available: http://arxiv.org/abs/2109.00957

[201] M. Jahanifar *et al.*, “Stain-Robust Mitotic Figure Detection for the Mitosis Domain Generalization Challenge,” Sep. 2021. [Online]. Available: http://arxiv.org/abs/2109.00853

[202] X. Wang *et al.*, “SK-Unet: An Improved U-Net Model With Selective Kernel for the Segmentation of LGE Cardiac MR Images,” *IEEE Sens J*, vol. 21, no. 10, pp. 11643–11653, May 2021, doi: 10.1109/JSEN.2021.3056131.

[203] A. Vahadane *et al.*, “Structure-Preserving Color Normalization and Sparse Stain Separation for Histological Images,” *IEEE Trans Med Imaging*, vol. 35, no. 8, pp. 1962–1971, Aug. 2016, doi: 10.1109/TMI.2016.2529665.

[204] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019, vol. 97, pp. 6105–6114.

[205] A. Cruz-Roa *et al.*, “Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks,” in *Medical Imaging 2014: Digital Pathology*, Mar. 2014, vol. 9041, no. 216, p. 904103. doi: 10.1117/12.2043872.

[206] T. Araújo *et al.*, “Classification of breast cancer histology images using Convolutional Neural Networks,” *PLoS One*, vol. 12, no. 6, p. e0177544, Jun. 2017, doi: 10.1371/journal.pone.0177544.

[207] S. Kwok, “Multiclass Classification of Breast Cancer in Whole-Slide Images,” in *ICIAR 2018: Image Analysis and Recognition*, 2018, pp. 931–940. doi: 10.1007/978-3-319-93000-8_106.

[208] C. Szegedy, S. Ioffe, and V. Vanhoucke, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” 2016. Accessed: Aug. 25, 2022. [Online]. Available: arxiv.org/abs/1602.07261

[209] S. S. Chennamsetty, M. Safwan, and V. Alex, “Classification of Breast Cancer Histology Image using Ensemble of Pre-trained Neural Networks,” 2018, pp. 804–811. doi: 10.1007/978-3-319-93000-8_91.

[210] Y. Qiu *et al.*, “Automatic Prostate Gleason Grading Using Pyramid Semantic Parsing Network in Digital Histopathology,” *Front Oncol*, vol. 12, Apr. 2022, doi: 10.3389/fonc.2022.772403.

[211] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid Scene Parsing Network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html

[212] S. K. Warfield, K. H. Zou, and W. M. Wells, “Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation,” *IEEE Trans Med Imaging*, vol. 23, no. 7, pp. 903–921, Jul. 2004, doi: 10.1109/TMI.2004.828354.

[213] J. Gamper, N. A. Koohbanani, K. Benet, A. Khuram, and N. Rajpoot, "PanNuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification," in *European Congress on Digital Pathology*, 2019, pp. 11–19. doi: 10.1007/978-3-030-23937-4_2.

[214] S. Graham *et al.*, "Lizard: A Large-Scale Dataset for Colonic Nuclear Instance Segmentation and Classification," pp. 684–693, 2021, doi: 10.1109/iccvw54120.2021.00082.

[215] Y. Zhou, O. F. Onder, and Tsou, "MH-FCN: Multi-Organ Nuclei Segmentation Algorithm," 2020.

[216] R. Verma, N. Kumar, A. Patil, N. C. Kurian, S. Rane, and A. Sethi, "Multi-organ Nuclei Segmentation and Classification Challenge 2020," no. February, 2020, doi: 10.13140/RG.2.2.12290.02244/1.

[217] Z. Zeng, W. Xie, Y. Zhang, and Y. Lu, "RIC-Unet: An Improved Neural Network Based on Unet for Nuclei Segmentation in Histology Images," *IEEE Access*, vol. 7, pp. 21420–21428, 2019, doi: 10.1109/ACCESS.2019.2896920.

[218] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," 2016, pp. 424–432. doi: 10.1007/978-3-319-46723-8_49.

[219] T. Araújo *et al.*, "Classification of breast cancer histology images using Convolutional Neural Networks," *PLoS One*, vol. 12, no. 6, p. e0177544, Jun. 2017, doi: 10.1371/journal.pone.0177544.

[220] A. Anghel *et al.*, "A High-Performance System for Robust Stain Normalization of Whole-Slide Images in Histopathology," *Front Med (Lausanne)*, vol. 6, Sep. 2019, doi: 10.3389/fmed.2019.00193.

[221] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015, doi: 10.48550/arXiv.1512.03385.

[222] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, "DCAN: Deep contour-aware networks for object instance segmentation from histology images," *Med Image Anal*, vol. 36, pp. 135–146, Feb. 2017, doi: 10.1016/j.media.2016.11.004.

[223] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Jul. 2020, pp. 237–242. doi: 10.1109/IWSSIP48289.2020.9145130.

[224] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–28, 2021, doi: 10.3390/electronics10030279.

[225] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.

[226] A. Reinke *et al.*, "Common Limitations of Image Processing Metrics: A Picture Story," pp. 1–11, Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.05642

[227] P. Jaccard, "La distribution de la flore dans la zone alpine," *Revue générale des sciences pures et appliquées*, vol. 18, no. 23, pp. 961–967, 1907.

[228] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged F1 and macro-averaged F1 scores," *Applied Intelligence*, vol. 52, no. 5, pp. 4961–4972, Mar. 2022, doi: 10.1007/s10489-021-02635-5.

[229] J. Cohen, "A Coefficient of Agreement for Nominal Scales," *Educ Psychol Meas*, vol. 20, no. 1, pp. 37–46, Apr. 1960, doi: 10.1177/001316446002000104.

[230] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9396–9405, 2019, doi: 10.1109/CVPR.2019.00963.

[231] S. Graham *et al.*, "CoNIC: Colon Nuclei Identification and Counting Challenge 2022," 2021. [Online]. Available: http://arxiv.org/abs/2111.14485

[232] D. Chicco, N. Tötsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Min*, vol. 14, no. 1, p. 13, Dec. 2021, doi: 10.1186/s13040-021-00244-z.

[233] R. Delgado and X.-A. Tibau, "Why Cohen's Kappa should be avoided as performance measure in classification," *PLoS One*, vol. 14, no. 9, p. e0222916, Sep. 2019, doi: 10.1371/journal.pone.0222916.

[234] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," Aug. 2020. [Online]. Available: http://arxiv.org/abs/2008.05756

[235] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance Problems in Object Detection: A Review," *IEEE Trans Pattern Anal Mach Intell*, vol. 43, no. 10, pp. 3388–3415, Oct. 2021, doi: 10.1109/TPAMI.2020.2981890.

[236] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric analysis of white matter lesions in MR images: method and validation," *IEEE Trans Med Imaging*, vol. 13, no. 4, pp. 716–724, 1994, doi: 10.1109/42.363096.

[237] W. C. Allsbrook *et al.*, "Interobserver reproducibility of Gleason grading of prostatic carcinoma: Urologic pathologists," *Hum Pathol*, vol. 32, no. 1, pp. 74–80, Jan. 2001, doi: 10.1053/hupa.2001.21134.

[238] M. L. McHugh, "Interrater reliability: the kappa statistic.," *Biochem Med (Zagreb)*, vol. 22, no. 3, pp. 276–82, 2012.

[239] M. McLean, J. Srigley, D. Banerjee, P. Warde, and Y. Hao, "Interobserver variation in prostate cancer Gleason scoring: Are there implications for the design of clinical trials and treatment strategies?," *Clin Oncol*, vol. 9, no. 4, pp. 222–225, Jan. 1997, doi: 10.1016/S0936-6555(97)80005-2.

[240] P. A. Humphrey, "Gleason grading and prognostic factors in carcinoma of the prostate," *Modern Pathology*, vol. 17, no. 3, pp. 292–306, Mar. 2004, doi: 10.1038/modpathol.3800054.

[241] W. C. Allsbrook, K. A. Mangold, M. H. Johnson, R. B. Lane, C. G. Lane, and J. I. Epstein, "Interobserver reproducibility of Gleason grading of prostatic carcinoma: General pathologist," *Hum Pathol*, vol. 32, no. 1, pp. 81–88, Jan. 2001, doi: 10.1053/hupa.2001.21135.

[242] T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," *Neural Comput*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998, doi: 10.1162/089976698300017197.

[243] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[244] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 152–161, 2016.

[245] J. Liu and Y. Xu, "T-Friedman Test: A New Statistical Test for Multiple Comparison with an Adjustable Conservativeness Measure," *International Journal of Computational Intelligence Systems*, vol. 15, no. 1, p. 29, Dec. 2022, doi: 10.1007/s44196-022-00083-8.

[246] A. Giusti, C. Caccia, D. C. Ciresan, J. Schmidhuber, and L. M. Gambardella, "A comparison of algorithms and humans for mitosis detection," in *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2014, pp. 1360–1363. doi: 10.1109/ISBI.2014.6868130.

[247] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, 2nd Edition. Springer, 2005.

[248] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognit Lett*, vol. 30, no. 1, pp. 27–38, Jan. 2009, doi: 10.1016/j.patrec.2008.08.010.

[249] M. Wiesenfarth *et al.*, "Methods and open-source toolkit for analyzing and visualizing challenge results," *Sci Rep*, vol. 11, no. 1, p. 2369, Dec. 2021, doi: 10.1038/s41598-021-82017-6.

[250] S. Bouix *et al.*, "On evaluating brain tissue classifiers without a ground truth," *Neuroimage*, vol. 36, no. 4, pp. 1207–1224, Jul. 2007, doi: 10.1016/j.neuroimage.2007.04.031.

[251] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," Sep. 2011. Accessed: Aug. 25, 2022. [Online]. Available: https://arxiv.org/abs/1109.2378

[252] D. Liu, D. Zhang, Y. Song, H. Huang, and W. Cai, "Panoptic Feature Fusion Net: A Novel Instance Segmentation Paradigm for Biomedical and Biological Images," *IEEE Transactions on Image Processing*, vol. 30, pp. 2045–2059, 2021, doi: 10.1109/TIP.2021.3050668.

[253] K. Benaggoune, Z. Al Masry, C. Devalland, S. Valmary-degano, N. Zerhouni, and L. H. Mouss, "Data Labeling Impact on Deep Learning Models in Digital Pathology: a Breast Cancer Case Study," 2022, pp. 117–129. doi: 10.1007/978-981-16-7771-7_10.

[254] S. Butte, H. Wang, M. Xian, and A. Vakanski, "Sharp-GAN: Sharpness Loss Regularized GAN for Histopathology Image Synthesis," in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, Mar. 2022, pp. 1–5. doi: 10.1109/ISBI52829.2022.9761534.

[255] L. Maier-Hein *et al.*, "BIAS: Transparent reporting of biomedical image analysis challenges," *Med Image Anal*, vol. 66, p. 101796, Dec. 2020, doi: 10.1016/j.media.2020.101796.

[256] S. A. Javed, D. Juyal, Z. Shanis, S. Chakraborty, H. Pokkalla, and A. Prakash, "Rethinking Machine Learning Model Evaluation in Pathology," Apr. 2022. [Online]. Available: http://arxiv.org/abs/2204.05205

[257] C. L. Srinidhi, O. Ciga, and A. L. Martel, "Deep neural network models for computational histopathology: A survey," *Med Image Anal*, vol. 67, 2021, doi: 10.1016/j.media.2020.101813.

[258] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation," *Med Image Anal*, vol. 63, p. 101693, 2020, doi: 10.1016/j.media.2020.101693.

[259] P. Zhang, Y. Zhong, Y. Deng, X. Tang, and X. Li, "A Survey on Deep Learning of Small Sample in Biomedical Image Analysis," Aug. 2019. [Online]. Available: http://arxiv.org/abs/1908.00473

[260] Şerban Vădineanu, D. Pelt, O. Dzyubachyk, and J. Batenburg, "An Analysis of the Impact of Annotation Errors on the Accuracy of Deep Learning for Cell Segmentation," *MIDL*, pp. 1–17, 2022.

[261] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*, vol. 3, no. 1. Morgan & Claypool, 2009. doi: 10.2200/S00196ED1V01Y200906AIM006.

[262] Q. Miao, R. Liu, P. Zhao, Y. Li, and E. Sun, "A Semi-Supervised Image Classification Model Based on Improved Ensemble Projection Algorithm," *IEEE Access*, vol. 6, pp. 1372–1379, 2018, doi: 10.1109/ACCESS.2017.2778881.

[263] L. Su, Y. Liu, M. Wang, and A. Li, "Semi-HIC: A novel semi-supervised deep learning method for histopathological image classification," *Comput Biol Med*, vol. 137, p. 104788, Oct. 2021, doi: 10.1016/j.compbiomed.2021.104788.

[264] M. Peikari, S. Salama, S. Nofech-Mozes, and A. L. Martel, "A Cluster-then-label Semi-supervised Learning Approach for Pathology Image Classification," *Sci Rep*, vol. 8, no. 1, p. 7193, Dec. 2018, doi: 10.1038/s41598-018-24876-0.

[265] Z. Chen *et al.*, "Weakly Supervised Histopathology Image Segmentation With Sparse Point Annotations," *IEEE J Biomed Health Inform*, vol. 25, no. 5, pp. 1673–1685, May 2021, doi: 10.1109/JBHI.2020.3024262.

[266] S. Shaw, M. Pajak, A. Lisowska, S. A. Tsaftaris, and A. Q. O'Neil, "Teacher-Student chain for efficient semi-supervised histology image classification," Mar. 2020. [Online]. Available: http://arxiv.org/abs/2003.08797

[267] J. N. Kather, N. Halama, and A. Marx, "100,000 histological images of human colorectal cancer and healthy tissue," Apr. 2018, doi: 10.5281/ZENODO.1214456.

[268] A. K. Jaiswal, I. Panshin, D. Shulkin, N. Aneja, and S. Abramov, "Semi-Supervised Learning for Cancer Detection of Lymph Node Metastases," Jun. 2019. [Online]. Available: http://arxiv.org/abs/1906.09587

[269] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *Artif Intell*, vol. 89, no. 1–2, pp. 31–71, 1997, doi: 10.1016/S0004-3702(96)00034-3.

[270] T. Durand, “Weakly supervised learning for visual recognition,” Doctoral thesis, Université Pierre et Marie Curie - Paris VI, Paris, 2017. Accessed: Aug. 25, 2022. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01667325v2

[271] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,” Oct. 2016, doi: 10.1007/s11263-019-01228-7.

[272] L. Chan, M. S. Hosseini, and K. N. Plataniotis, “A Comprehensive Analysis of Weakly-Supervised Semantic Segmentation in Different Image Domains,” *Int J Comput Vis*, vol. 129, no. 2, pp. 361–384, Feb. 2021, doi: 10.1007/s11263-020-01373-4.

[273] Y. Xu, J.-Y. Zhu, E. I.-C. Chang, M. Lai, and Z. Tu, “Weakly supervised histopathology cancer image segmentation and classification,” *Med Image Anal*, vol. 18, no. 3, pp. 591–604, Apr. 2014, doi: 10.1016/j.media.2014.01.010.

[274] Z. Jia, X. Huang, E. I. C. Chang, and Y. Xu, “Constrained Deep Weak Supervision for Histopathology Image Segmentation,” *IEEE Trans. on Medical Imaging*, vol. 36, no. 11, pp. 2376–2388, 2017, doi: 10.1109/TMI.2017.2724070.

[275] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC Superpixels Compared to State-of-the-art Superpixel Methods,” *IEEE Trans Pattern Anal Mach Intell*, vol. 34, no. 11, pp. 2274–2282, 2012.

[276] B. Frénay and M. Verleysen, “Classification in the presence of label noise: A survey,” *IEEE Trans Neural Netw Learn Syst*, vol. 25, no. 5, pp. 845–869, 2014, doi: 10.1109/TNNLS.2013.2292894.

[277] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, “Learning From Noisy Labels With Deep Neural Networks: A Survey,” *IEEE Trans Neural Netw Learn Syst*, pp. 1–19, 2022, doi: 10.1109/TNNLS.2022.3152527.

[278] J. Yao *et al.*, “Deep Learning From Noisy Image Labels With Quality Embedding,” *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1909–1922, Apr. 2019, doi: 10.1109/TIP.2018.2877939.

[279] X. Wang, Y. Hua, E. Kodirov, and N. M. Robertson, “IMAE for Noise-Robust Learning: Mean Absolute Error Does Not Treat Examples Equally and Gradient Magnitude’s Variance Matters,” Mar. 2019. [Online]. Available: http://arxiv.org/abs/1903.12141

[280] B. Han *et al.*, “Co-teaching: Robust Training Deep Neural Networks with Extremely Noisy Labels,” 2018. [Online]. Available: http://arxiv.org/abs/1804.06872

[281] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, p. 6, 2013.

[282] G. O. Rolls, N. J. Farmer, and J. B. Hall, *Artifacts in Histological and Cytological Preparations*. Leica Microsystems, 2008.

[283] N. Kanwal, F. Perez-Bueno, A. Schmidt, K. Engan, and R. Molina, "The Devil is in the Details: Whole Slide Image Acquisition and Processing for Artifacts Detection, Color Variation, and Data Augmentation: A Review," *IEEE Access*, vol. 10, pp. 58821–58844, 2022, doi: 10.1109/ACCESS.2022.3176091.

[284] E. McInnes, "Artefacts in histopathology," *Comp Clin Path*, vol. 13, no. 3, pp. 100–108, Mar. 2005, doi: 10.1007/s00580-004-0532-4.

[285] M. Haghighat *et al.*, "Automated quality assessment of large digitised histology cohorts by artificial intelligence," *Sci Rep*, vol. 12, no. 1, p. 5002, Dec. 2022, doi: 10.1038/s41598-022-08351-5.

[286] Y. Chen *et al.*, "Assessment of a computerized quantitative quality control tool for whole slide images of kidney biopsies," *J Pathol*, vol. 253, no. 3, pp. 268–278, Mar. 2021, doi: 10.1002/path.5590.

[287] A. Janowczyk, R. Zuo, H. Gilmore, M. Feldman, and A. Madabhushi, "HistoQC: An Open-Source Quality Control Tool for Digital Pathology Slides," *JCO Clin Cancer Inform*, no. 3, pp. 1–7, Nov. 2019, doi: 10.1200/CCI.18.00157.

[288] N. Hashimoto, P. A. Bautista, M. Yamaguchi, N. Ohyama, and Y. Yagi, "Referenceless image quality evaluation for whole slide imaging," *J Pathol Inform*, vol. 3, no. 1, p. 9, Jan. 2012, doi: 10.4103/2153-3539.93891.

[289] D. Ameisen *et al.*, "Towards better digital pathology workflows: programming libraries for high-speed sharpness assessment of Whole Slide Images," *Diagn Pathol*, vol. 9, no. S1, p. S3, Dec. 2014, doi: 10.1186/1746-1596-9-S1-S3.

[290] P. Shrestha, R. Kneepkens, J. Vrijnsen, D. Vossen, E. Abels, and B. Hulsken, "A quantitative approach to evaluate image quality of whole slide imaging scanners," *J Pathol Inform*, vol. 7, no. 1, p. 56, Jan. 2016, doi: 10.4103/2153-3539.197205.

[291] H. M. Shakhawat, T. Nakamura, F. Kimura, Y. Yagi, and M. Yamaguchi, "Automatic Quality Evaluation of Whole Slide Images for the Practical Use of Whole Slide Imaging Scanner," *ITE Transactions on Media Technology and Applications*, vol. 8, no. 4, pp. 252–268, 2020, doi: 10.3169/mta.8.252.

[292] X. M. Lopez *et al.*, "An automated blur detection method for histological whole slide imaging," *PLoS One*, vol. 8, no. 12, 2013, doi: 10.1371/journal.pone.0082710.

[293] D. Gao, D. Padfield, J. Rittscher, and R. McKay, "Automated Training Data Generation for Microscopy Focus Classification," 2010, pp. 446–453. doi: 10.1007/978-3-642-15745-5_55.

[294] G. Campanella, A. R. Rajanna, L. Corsale, P. J. Schüffler, Y. Yagi, and T. J. Fuchs, "Towards machine learned quality control: A benchmark for sharpness quantification in digital pathology," *Computerized Medical Imaging and Graphics*, vol. 65, pp. 142–151, Apr. 2018, doi: 10.1016/j.compmedimag.2017.09.001.

[295] S. Palokangas, J. Selinummi, and O. Yli-Harja, "Segmentation of Folds in Tissue Section Images," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2007, pp. 5641–5644. doi: 10.1109/IEMBS.2007.4353626.

[296] P. A. Bautista and Y. Yagi, “Detection of tissue folds in whole slide images,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sep. 2009, pp. 3669–3672. doi: 10.1109/IEMBS.2009.5334529.

[297] S. Kothari, J. H. Phan, and M. D. Wang, “Eliminating tissue-fold artifacts in histopathological whole-slide images for improved image-based prediction of cancer grade,” *J Pathol Inform*, vol. 4, no. 22, 2013, doi: 10.4103/2153-3539.117448.

[298] C. Senaras, M. K. K. Niazi, G. Lozanski, and M. N. Gurcan, “DeepFocus: Detection of out-of-focus regions in whole slide digital images using deep learning,” *PLoS One*, vol. 13, no. 10, p. e0205387, Oct. 2018, doi: 10.1371/journal.pone.0205387.

[299] Z. Swiderska-Chadaj, T. Markiewicz, J. Gallego, G. Bueno, B. Grala, and M. Lorent, “Deep learning for damaged tissue detection and segmentation in Ki-67 brain tumor specimens based on the U-net model,” *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 66, no. 6, pp. 849–856, 2018.

[300] B. Schömig-Markiefka *et al.*, “Quality control stress test for deep learning-based diagnostic model in digital pathology,” *Modern Pathology*, vol. 34, no. 12, pp. 2098–2108, Dec. 2021, doi: 10.1038/s41379-021-00859-x.

[301] S. S. Cross, “Grading and scoring in histopathology,” *Histopathology*, vol. 33, no. 2, pp. 99–106, Aug. 1998, doi: 10.1046/j.1365-2559.1998.00495.x.

[302] C. W. ELSTON and I. O. ELLIS, “Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with long-term follow-up,” *Histopathology*, vol. 19, no. 5, pp. 403–410, Nov. 1991, doi: 10.1111/j.1365-2559.1991.tb00229.x.

[303] D. C. Paech *et al.*, “A Systematic Review of the Interobserver Variability for Histology in the Differentiation between Squamous and Nonsquamous Non-small Cell Lung Cancer,” *Journal of Thoracic Oncology*, vol. 6, no. 1, pp. 55–63, Jan. 2011, doi: 10.1097/JTO.0b013e3181fc0878.

[304] E. Hernandez, B. S. Bhagavan, T. H. Parmley, and N. B. Rosenshein, “Interobserver variability in the interpretation of epithelial ovarian cancer,” *Gynecol Oncol*, vol. 17, no. 1, pp. 117–123, Jan. 1984, doi: 10.1016/0090-8258(84)90065-9.

[305] A. Malpica *et al.*, “Interobserver and Intraobserver Variability of a Two-tier System for Grading Ovarian Serous Carcinoma,” *American Journal of Surgical Pathology*, vol. 31, no. 8, pp. 1168–1174, Aug. 2007, doi: 10.1097/PAS.0b013e31803199b0.

[306] K. W. Gilchrist *et al.*, “Interobserver reproducibility of histopathological features in stage II breast cancer,” *Breast Cancer Res Treat*, vol. 5, no. 1, pp. 3–10, Feb. 1985, doi: 10.1007/BF01807642.

[307] P. Robbins *et al.*, “Histological grading of breast carcinomas: A study of interobserver agreement,” *Hum Pathol*, vol. 26, no. 8, pp. 873–879, Aug. 1995, doi: 10.1016/0046-8177(95)90010-1.

[308] T. Davidson *et al.*, "Breast cancer prognostic factors in the digital era: Comparison of Nottingham grade using whole slide images and glass slides," *J Pathol Inform*, vol. 10, no. 1, p. 11, 2019, doi: 10.4103/jpi.jpi_29_18.

[309] J. S. Meyer *et al.*, "Breast carcinoma malignancy grading by Bloom–Richardson system vs proliferation index: reproducibility of grade and advantages of proliferation index," *Modern Pathology*, vol. 18, no. 8, pp. 1067–1078, Aug. 2005, doi: 10.1038/modpathol.3800388.

[310] G. K. Zagars, A. G. Ayala, A. C. von Eschenbach, and A. Pollack, "The prognostic importance of gleason grade in prostatic adenocarcinoma: A long-term follow-up study of 648 patients treated with radiation therapy," *International Journal of Radiation Oncology*Biology*Physics*, vol. 31, no. 2, pp. 237–245, Jan. 1995, doi: 10.1016/0360-3016(94)00323-D.

[311] Ş. O. Özdamar, Ş. Sarikaya, L. Yildiz, M. K. Atilla, B. Kandemir, and S. Yildiz, "Intraobserver and interobserver reproducibility of WHO and Gleason histologic grading systems in prostatic adenocarcinomas," *Int Urol Nephrol*, vol. 28, no. 1, pp. 73–77, Jan. 1996, doi: 10.1007/BF02550141.

[312] T. A. Ozkan, A. T. Eruyar, O. O. Cebeci, O. Memik, L. Ozcan, and I. Kuskonmaz, "Interobserver variability in Gleason histological grading of prostate cancer," *Scand J Urol*, vol. 50, no. 6, pp. 420–424, Nov. 2016, doi: 10.1080/21681805.2016.1206619.

[313] W. Bulten *et al.*, "Automated deep-learning system for Gleason grading of prostate cancer using biopsies: a diagnostic study," *Lancet Oncol*, vol. 21, no. 2, pp. 233–241, Feb. 2020, doi: 10.1016/S1470-2045(19)30739-9.

[314] S. Bauer, N. Carion, P. Schüffler, T. Fuchs, P. Wild, and J. M. Buhmann, "Multi-Organ Cancer Classification and Survival Analysis," Jun. 2016. [Online]. Available: http://arxiv.org/abs/1606.00897

[315] R. Turkki, N. Linder, P. E. Kovanen, T. Pellinen, and J. Lundin, "Antibody-supervised deep learning for quantification of tumor-infiltrating immune cells in hematoxylin and eosin stained breast cancer samples," *J Pathol Inform*, vol. 7, no. 1, p. 38, Jan. 2016, doi: 10.4103/2153-3539.189703.

[316] C. Malon *et al.*, "Mitotic figure recognition: Agreement among pathologists and computerized detector," *Analytical Cellular Pathology*, vol. 35, no. 2, pp. 97–100, 2012, doi: 10.3233/ACP-2011-0029.

[317] Á. García Faura, D. Štepec, T. Martinčič, and D. Skočaj, "Segmentation of Multiple Myeloma Plasma Cells in Microscopy Images with Noisy Labels," Nov. 2021. [Online]. Available: http://arxiv.org/abs/2111.05125

[318] G. Nir *et al.*, "Comparison of Artificial Intelligence Techniques to Evaluate Performance of a Classifier for Automatic Grading of Prostate Cancer From Digitized Histopathologic Images," *JAMA Netw Open*, vol. 2, no. 3, p. e190442, Mar. 2019, doi: 10.1001/jamanetworkopen.2019.0442.

[319] D. Karimi, G. Nir, L. Fazli, P. C. Black, L. Goldenberg, and S. E. Salcudean, "Deep Learning-Based Gleason Grading of Prostate Cancer From Histopathology Images—Role of

Multiscale Decision Aggregation and Data Augmentation," *IEEE J Biomed Health Inform*, vol. 24, no. 5, pp. 1413–1426, May 2020, doi: 10.1109/JBHI.2019.2944643.

[320] V. C. Raykar *et al.*, "Learning From Crowds," *The Journal of Machine Learning Research*, vol. 11, pp. 1297--1322, 2010.

[321] A. A. Khani, S. A. Fatemi Jahromi, H. O. Shahreza, H. Behroozi, and M. S. Baghshah, "Towards Automatic Prostate Gleason Grading Via Deep Convolutional Neural Networks," in *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, Dec. 2019, pp. 1–6. doi: 10.1109/ICSPIS48872.2019.9066019.

[322] C. Jin, R. Tanno, M. Xu, T. Mertzanidou, and D. C. Alexander, "Foveation for Segmentation of Mega-Pixel Histology Images," 2020, pp. 561–571. doi: 10.1007/978-3-030-59722-1_54.

[323] O. Ciga and A. L. Martel, "Learning to segment images with classification labels," *Med Image Anal*, vol. 68, p. 101912, Feb. 2021, doi: 10.1016/j.media.2020.101912.

[324] Y. Yang, F. G Farhat, Y. Xue, F. Y Shih, and U. Roshan, "Classification of Histopathology Images with Random Depthwise Convolutional Neural Networks," in *2020 7th International Conference on Bioinformatics Research and Applications*, Sep. 2020, pp. 22–27. doi: 10.1145/3440067.3440072.

[325] Y. Zhang, J. Zhang, Y. Song, C. Shen, and G. Yang, "Gleason Score Prediction using Deep Learning in Tissue Microarray Image," May 2020. [Online]. Available: http://arxiv.org/abs/2005.04886

[326] L. Xiao, Y. Li, L. Qv, X. Tian, Y. Peng, and S. K. Zhou, "Pathological Image Segmentation with Noisy Labels," Mar. 2021. [Online]. Available: http://arxiv.org/abs/2104.02602

[327] A. K. R. Zheng, "Deep Learning for Prostate Cancer Grading," Master Thesis, Université Libre de Bruxelles, Brussels, 2021.

[328] J. Packeisen, "Demystified ... Tissue microarray technology," *Molecular Pathology*, vol. 56, no. 4, pp. 198–204, Aug. 2003, doi: 10.1136/mp.56.4.198.

[329] J. I. Epstein, W. C. Allsbrook, M. B. Amin, and L. L. Egevad, "The 2005 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma," *American Journal of Surgical Pathology*, vol. 29, no. 9, pp. 1228–1242, Sep. 2005, doi: 10.1097/01.pas.0000173646.99337.b1.

[330] M. U. Rehman, S. Akhtar, M. Zakwan, and M. H. Mahmood, "Novel architecture with selected feature vector for effective classification of mitotic and non-mitotic cells in breast cancer histology images," *Biomed Signal Process Control*, vol. 71, p. 103212, Jan. 2022, doi: 10.1016/j.bspc.2021.103212.

[331] G. Raipuria, S. Bonthu, and N. Singhal, "Noise Robust Training of Segmentation Model Using Knowledge Distillation," in *Pattern Recognition, ICPR International Workshops and Challenges. ICPR 2021.*, 2021, pp. 97–104. doi: 10.1007/978-3-030-68763-2_8.

[332] A. H. Md. Linkon, Md. M. Labib, T. Hasan, M. Hossain, and M.-E.- Jannat, "Deep learning in prostate cancer diagnosis and Gleason grading in histopathology images: An extensive study," *Inform Med Unlocked*, vol. 24, p. 100582, 2021, doi: 10.1016/j.imu.2021.100582.

[333] Y. Mun, I. Paik, S.-J. Shin, T.-Y. Kwak, and H. Chang, "Yet Another Automated Gleason Grading System (YAAGGS) by weakly supervised deep learning," *NPJ Digit Med*, vol. 4, no. 1, p. 99, Dec. 2021, doi: 10.1038/s41746-021-00469-6.

[334] Y. Li, N. He, S. Peng, K. Ma, and Y. Zheng, "Deep Reinforcement Exemplar Learning for Annotation Refinement," in *MICCAI 2021*, 2021, pp. 487–496. doi: 10.1007/978-3-030-87237-3_47.

[335] E. and M. National Academies of Sciences, *Reproducibility and Replicability in Science*. Washington, D.C.: National Academies Press, 2019. doi: 10.17226/25303.

[336] J. van der Laak, G. Litjens, and F. Ciompi, "Deep learning in histopathology: the path to the clinic," *Nat Med*, vol. 27, no. 5, pp. 775–784, May 2021, doi: 10.1038/s41591-021-01343-4.

[337] V. Baxi, R. Edwards, M. Montalto, and S. Saha, "Digital pathology and artificial intelligence in translational medicine and clinical practice," *Modern Pathology*, vol. 35, no. 1, pp. 23–32, Jan. 2022, doi: 10.1038/s41379-021-00919-2.

[338] S. Li, Z. Gao, and X. He, "Superpixel-Guided Iterative Learning from Noisy Labels for Medical Image Segmentation," 2021, pp. 525–535. doi: 10.1007/978-3-030-87193-2_50.

[339] G. Algan and I. Ulusoy, "MetaLabelNet: Learning to Generate Soft-Labels From Noisy-Labels," *IEEE Transactions on Image Processing*, vol. 31, pp. 4352–4362, 2022, doi: 10.1109/TIP.2022.3183841.

[340] R. Charkaoui, "Artefacts detection and ischemic time prediction based on TCGA and GTEx samples quality assessment," Master Thesis, Université Libre de Bruxelles, 2022.

[341] O. Ciga, T. Xu, and A. L. Martel, "Self supervised contrastive learning for digital histopathology," *Machine Learning with Applications*, vol. 7, Mar. 2022, doi: 10.1016/j.mlwa.2021.100198.

[342] D. Karimi, G. Nir, L. Fazli, P. C. Black, L. Goldenberg, and S. E. Salcudean, "Deep Learning-Based Gleason Grading of Prostate Cancer from Histopathology Images - Role of Multiscale Decision Aggregation and Data Augmentation," *IEEE J Biomed Health Inform*, vol. 24, no. 5, pp. 1413–1426, 2020, doi: 10.1109/JBHI.2019.2944643.

[343] K. Coffman, C. B. Flikkema, and O. Shurchkov, "Gender stereotypes in deliberation and team decisions," *Games Econ Behav*, vol. 129, pp. 329–349, Sep. 2021, doi: 10.1016/j.geb.2021.06.004.

[344] C. Chauhan and R. R. Gullapalli, "Ethics of AI in Pathology," *Am J Pathol*, vol. 191, no. 10, pp. 1673–1683, Oct. 2021, doi: 10.1016/j.ajpath.2021.06.011.

[345] T. Sorell, N. Rajpoot, and C. Verrill, "Ethical issues in computational pathology," *J Med Ethics*, vol. 48, no. 4, pp. 278–284, Apr. 2022, doi: 10.1136/medethics-2020-107024.

[346] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations," *Science (1979)*, vol. 366, no. 6464, pp. 447–453, Oct. 2019, doi: 10.1126/science.aax2342.

# A. Description of the datasets

Several datasets, mostly coming from digital pathology competitions, are used throughout this thesis for our experiments and analysis. We provide here a description of their main characteristics, for reference.

## A.1 MITOS12

| | |
|---|---|
| **General information** | Dataset of the MITOS12 **mitosis detection** challenge (post-challenge publication by Roux et al. [1]), organized by IPAL Laboratory, TRIBVN Company, Pitié-Salpêtrière Hospital and CIALAB and hosted at **ICPR 2012**. |
| **Website** | http://ludo17.free.fr/mitos_2012/index.html |
| **Availability** | Training and test set images and annotations available. |
| **Main characteristics** | Patches from **H&E-stained** slides (each patch corresponding to a 512x512µm$^2$ region) extracted from **breast cancer biopsies**.<br>3 different acquisition devices: Aperio & Hamamatsu RGB scanners + multispectral microscope (only Aperio & Hamamatsu are used in our experiments), with resolutions of 0.25, 0.23 and 0.19 µm/px, respectively (corresponding to 40x magnification).<br>Number of samples: **5 patients**, 50 patches.<br>Training set: 35 images from all 5 patients, containing 226 mitosis.<br>Test set: 15 images from all 5 patients, containing 103 mitosis. |
| **Annotations** | **Single pathologist**, segmented mitotic nuclei. See Figure A.1 for an illustration of the acquisition and annotation pipeline. |
| **Balance** | The mitosis regions occupy 0.09% of the total tissue area. There is an average of 6.6 mitosis per 2048x2048px image patch. |

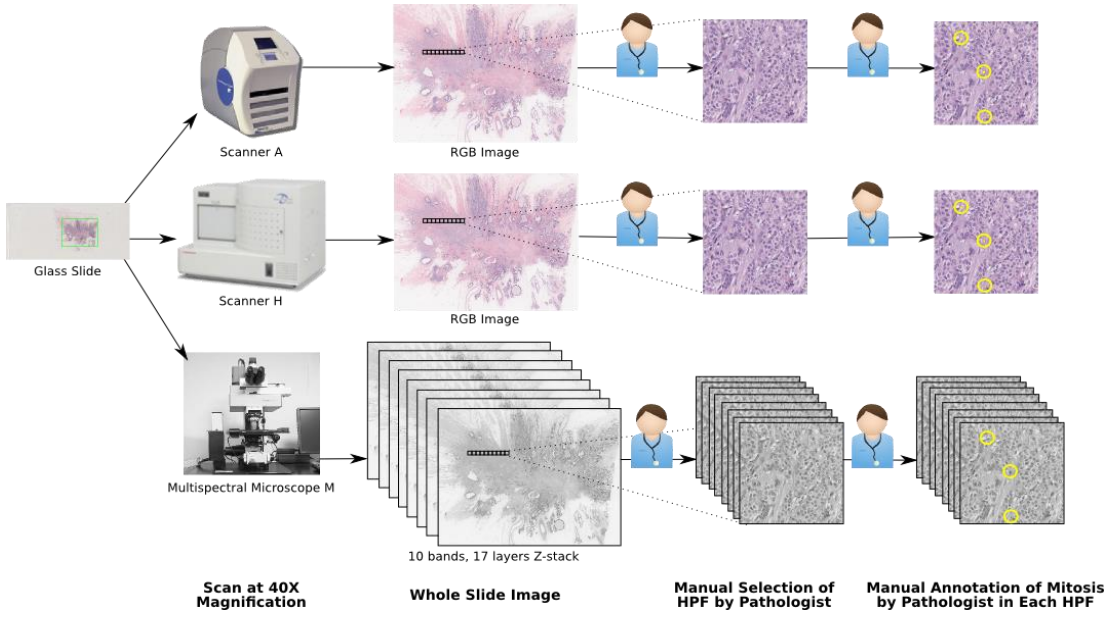



**Figure A.1. Acquisition and annotation process for the MITOS12 dataset, from the challenge's website.**

## A.2 GlaS 2015

| | |
|---|---|
| **General information** | Dataset of the GlaS 2015 **gland segmentation** challenge (post-challenge publication by Sirinukunwattana et al. [2]), organized by Bioimage Analysis Lab (University of Warwick), Qatar University, Eindhoven University of Technology & UMC Utrecht, and University Hospital of Coventry and Warwickshire and hosted at **MICCAI 2015**. |
| **Website** | https://warwick.ac.uk/fac/cross_fac/tia/data/glascontest/ |
| **Availability** | Training and test set images and annotations available. |
| **Main characteristics** | Patches from **H&E-stained** slides extracted from **colorectal cancer tissue**. Acquisition device: Zeiss scanner, resolution of 0.62 µm/px (corresponding to 20x magnification).<br>Number of samples: **16 patients**, 165 patches.<br>Training set: 85 images from all 16 patients (but from different "visual fields" than those in the test set). 37 images come from "benign" tissue, 48 from "malignant" tissue.<br>Test set: 80 images from all 16 patients. Further split in test set A (33 benign, 27 malignant) and B (4 benign, 16 malignant). In the challenge, test A was used for "off-site" testing, and test B for "on-site" testing on the day of the challenge event. |
| **Annotations** | **Single pathologist**, segmented glands. See Figure A.2 for examples of images and annotations from the training set. |
| **Balance** | Training set: 769 glands making up 50% of the total pixel area.<br>Test set: 761 glands making up 51% of the total pixel area. |

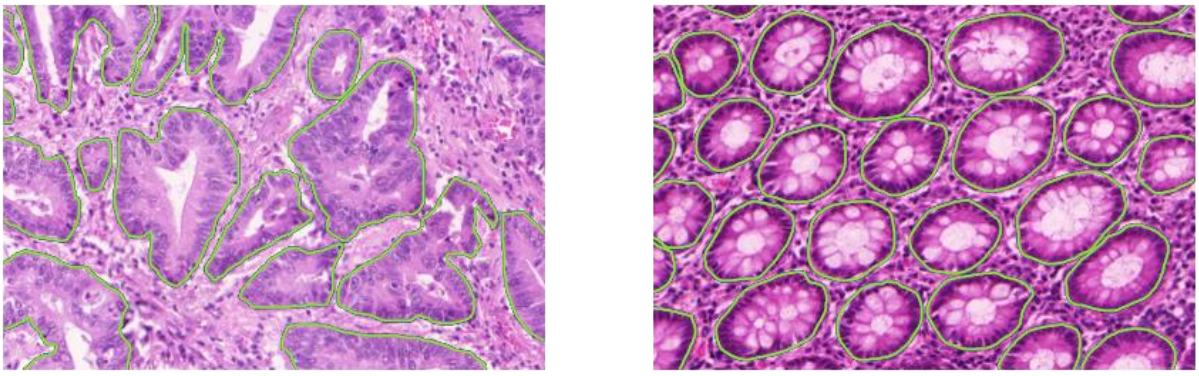


**Figure A.2. Example of images and annotated glands from the GlaS challenge training set.**

## A.3 Janowczyk's epithelium dataset

| | |
|---|---|
| **General information** | Dataset accompanying the "Deep learning for digital pathology image analysis" tutorial by Janowczyk and Madabhushi [3] for the **epithelium segmentation** task. |
| **Website** | http://www.andrewjanowczyk.com/deep-learning/ |
| **Availability** | Images and annotations available. |
| **Main characteristics** | Patches from **H&E-stained** slides extracted from oestrogen receptor positive (ER+) **breast cancer tissue**. Acquisition device: unspecified, 20x magnification.<br>Number of samples: **42 patients**, 42 patches. |
| **Annotations** | **Single pathologist**, segmented epithelium region (see Figure A.3). |
| **Balance** | The epithelium regions occupy 33.5% of the total pixel area |

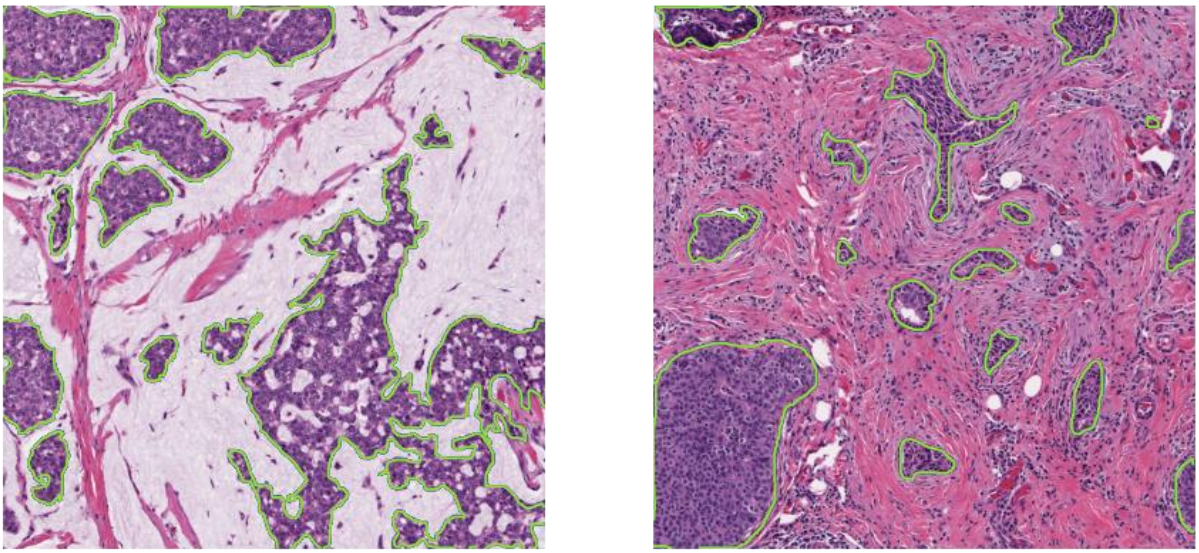


**Figure A.3. Example of images and annotated epithelium regions from Janowczyk's epithelium dataset.**

## A.4 Gleason 2019

| | |
|---|---|
| **General information** | Dataset of the Gleason 2019 **Gleason pattern segmentation and grading** challenge, organised by the University of British Columbia with data from the Vancouver Prostate Center, and hosted at **MICCAI 2019**. No post-challenge publication has been made by the organisers. A 2022 publication from the winning team is available in Qiu et al. [4], and several publications have been made by the organizing team using the data [5]–[8]. |
| **Website** | https://gleason2019.grand-challenge.org/ |
| **Availability** | Training set images and individual expert annotations, test set images. |
| **Main characteristics** | TMA cores from **H&E-stained prostate cancer** tissue. Acquisition device: SCN400 Slide Scanner, 40x magnification.<br>Number of samples: 331 cores from around 230 patients (the number of cores in the available data doesn't exactly match the numbers reported in the different publications).<br>Training set: 244 cores.<br>Test set: 87 cores. |
| **Annotations** | **Individual annotations from 6 different pathologists** (not all pathologists annotated all slides). |
| **Balance** | Large imbalance between the different grades, with Gleason pattern 5 very sparsely represented. |

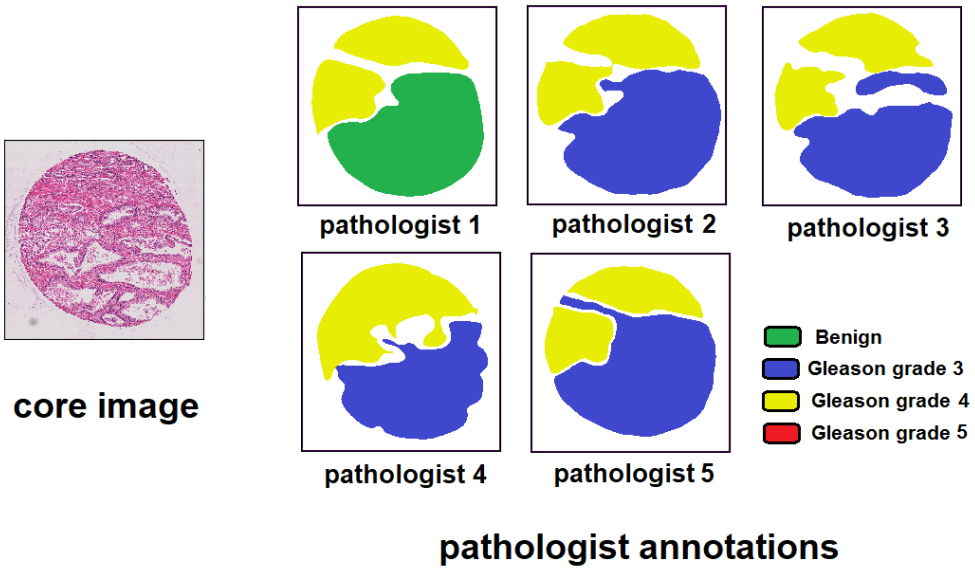



**Figure A.4. Image and individual pathologist annotations of a core from the Gleason 2019 challenge. Source of the image: https://gleason2019.grand-challenge.org/**

## A.5 MoNuSAC 2020

| | |
|---|---|
| **General information** | Dataset of the MoNuSAC 2020 **nuclei instance segmentation and classification** challenge, organised by the Case Western Reserve University in Cleveland, Ohio, and the Indian Institute of Technology in Bombay, India, and hosted at ISBI 2020. A post-challenge publication was made by Verma et al. [9]. We published a comment article noting some errors in the challenge results [10], leading to a response and partial correction of the results from Verma et al. [11]. |
| **Website** | https://monusac-2020.grand-challenge.org |
| **Availability** | Training and test set images and annotations available. Predictions of four of the top-five teams on the test set are also available. Source code for reading the annotation file and for the implementation of the evaluation metric available on GitHub[1]. |
| **Main characteristics** | Patches from **H&E-stained** tissue from four different organs (lung, prostate, kidney and breast) extracted from the TCGA portal at 40x magnification.<br>Number of samples: 71 patients, 310 patches (with more than 45.000 annotated nuclei).<br>Training set: 46 patients, 209 patches.<br>Test set: 25 patients, 101 patches. |
| **Annotations** | **Single annotation** per patch available. Annotations were made by "engineering graduate students" with quality control by "an expert pathologist" [9]. |
| **Balance** | Large class imbalance, with 20-30x more epithelial and lymphocyte nuclei than macrophages and neutrophils. However, the latter are much larger, so that the "per-pixel" imbalance is less strong. |

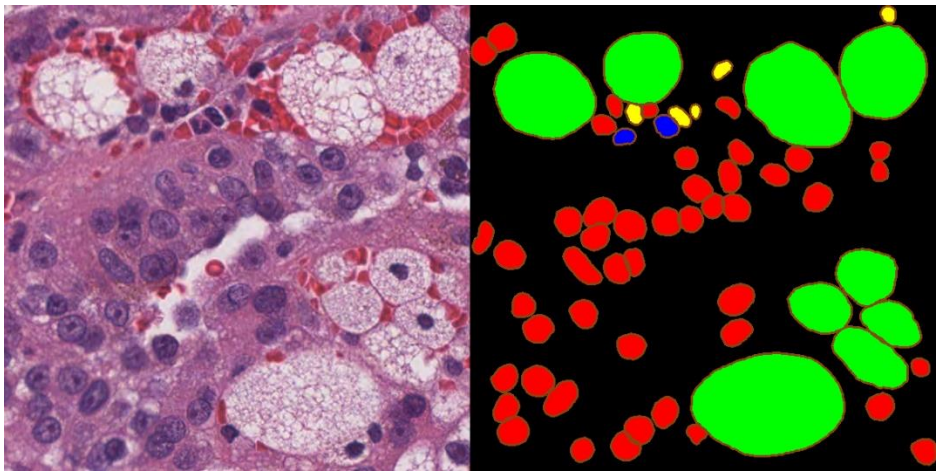


**Figure A.5. Image patch (left) and annotations (rights) from a kidney tissue slide of the MoNuSAC dataset. In the annotations, epithelial nuclei are in red, lymphocytes in yellow, neutrophils in blue, macrophages in green, and boundaries are highlighted in brown.**

[1] https://github.com/ruchikaverma-iitg/MoNuSAC

## A.6 Artefact dataset

| | |
|---|---|
| **General information** | Dataset used in several of our publications [12], [13] on **artefact segmentation** in digital pathology. |
| **Website** | https://doi.org/10.5281/zenodo.3773097 |
| **Availability** | Low-resolution (1.25x and 2.5x magnification) WSIs and annotation masks, as well as some extracted patches with patch-level annotations on the type of artefacts present. |
| **Main characteristics** | A total of 22 WSIs from 3 different tissue blocks:<br>• Block A (20 slides): 10 H&E-stained and 10 IHC (anti-pan-cytokeratin) from colorectal cancer tissue.<br>• Block B (1 slide): IHC (anti-pan-cytokeratin) from gastroesophageal junction (dysplasic) lesion.<br>• Block C (1 slide): IHC (anti-NR2F2) from head and neck carcinoma.<br>Training set: 18 slides from block A.<br>Validation set: 2 slides from block A + 1 slide from block B, as well as 21 image patches extracted from those 3 slides.<br>Test set: 1 slide from block C. (Additionally in our 2020 publication [13], 4 slides from TCGA are used for a qualitative assessment) |
| **Annotations** | Very rough annotations made by A. Foucart for block A and B, more precise annotations made by an expert technologist for block C. All types of artefactual regions are segmented (including tissue folds and tears, ink artefacts, pen markings, blur, etc.). A total of 918 distinct artefacts are annotated in the training set. |
| **Balance** | Very low density of annotated objects (2% positive pixels in the training set, 8% in the validation set and 9% in the test slide). |

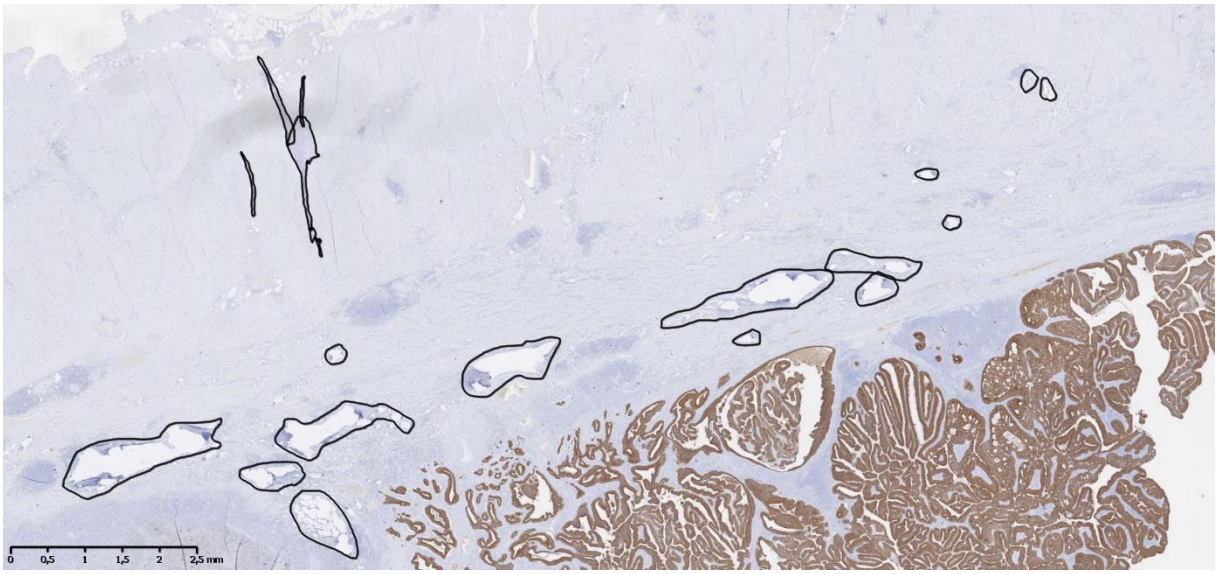



**Figure A.6. Annotated slide from the artefact training set, with imprecise delineation and many unlabelled artefacts, including blurry regions and smaller tears. Image reproduced from** [13].

## References

[1] L. Roux *et al.*, "Mitosis detection in breast cancer histological images An ICPR 2012 contest," *Journal of Pathology Informatics*, vol. 4, no. 1, p. 8, 2013, doi: 10.4103/2153-3539.112693.

[2] K. Sirinukunwattana, J. P. W. Pluim, H. Chen, and Others, "Gland segmentation in colon histology images: The glas challenge contest," *Medical Image Analysis*, vol. 35, pp. 489–502, 2017, doi: 10.1016/j.media.2016.08.008.

[3] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *Journal of Pathology Informatics*, vol. 7, no. 1, 2016, doi: 10.4103/2153-3539.186902.

[4] Y. Qiu *et al.*, "Automatic Prostate Gleason Grading Using Pyramid Semantic Parsing Network in Digital Histopathology," *Frontiers in Oncology*, vol. 12, Apr. 2022, doi: 10.3389/fonc.2022.772403.

[5] D. Karimi, G. Nir, L. Fazli, P. C. Black, L. Goldenberg, and S. E. Salcudean, "Deep Learning-Based Gleason Grading of Prostate Cancer from Histopathology Images - Role of Multiscale Decision Aggregation and Data Augmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 5, pp. 1413–1426, 2020, doi: 10.1109/JBHI.2019.2944643.

[6] D. Karimi, G. Nir, L. Fazli, P. C. Black, L. Goldenberg, and S. E. Salcudean, "Deep Learning-Based Gleason Grading of Prostate Cancer From Histopathology Images—Role of Multiscale Decision Aggregation and Data Augmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 5, pp. 1413–1426, May 2020, doi: 10.1109/JBHI.2019.2944643.

[7] G. Nir *et al.*, "Comparison of Artificial Intelligence Techniques to Evaluate Performance of a Classifier for Automatic Grading of Prostate Cancer From Digitized Histopathologic Images," *JAMA Network Open*, vol. 2, no. 3, p. e190442, Mar. 2019, doi: 10.1001/jamanetworkopen.2019.0442.

[8] G. Nir *et al.*, "Automatic grading of prostate cancer in digitized histopathology images: Learning from multiple experts," *Medical Image Analysis*, vol. 50, pp. 167–180, Dec. 2018, doi: 10.1016/j.media.2018.09.005.

[9] R. Verma *et al.*, "MoNuSAC2020: A Multi-Organ Nuclei Segmentation and Classification Challenge," *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3413–3423, Dec. 2021, doi: 10.1109/TMI.2021.3085712.

[10] A. Foucart, O. Debeir, and C. Decaestecker, "Comments on 'MoNuSAC2020: A Multi-Organ Nuclei Segmentation and Classification Challenge,'" *IEEE Transactions on Medical Imaging*, vol. 41, no. 4, pp. 997–999, Apr. 2022, doi: 10.1109/TMI.2022.3156023.

[11] R. Verma, N. Kumar, A. Patil, N. C. Kurian, S. Rane, and A. Sethi, "Author's Reply to 'MoNuSAC2020: A Multi-Organ Nuclei Segmentation and Classification Challenge,'" *IEEE Transactions on Medical Imaging*, vol. 41, no. 4, pp. 1000–1003, Apr. 2022, doi: 10.1109/TMI.2022.3157048.

[12] A. Foucart, O. Debeir, and C. Decaestecker, "Artifact Identification in Digital Pathology from Weak and Noisy Supervision with Deep Residual Networks," in *2018 4th International*

*Conference on Cloud Computing Technologies and Applications (Cloudtech)*, Nov. 2018, pp. 1–6. doi: 10.1109/CloudTech.2018.8713350.

[13] A. Foucart, O. Debeir, and C. Decaestecker, “Snow Supervision in Digital Pathology: Managing Imperfect Annotations for Segmentation in Deep Learning,” 2020, doi: 10.21203/rs.3.rs-116512.

# B. Description of the networks

This section provides a reference for the different DCNNs used for our experiments through this thesis (B.1.1), as well as some commonly found architectures from the state-of-the-art (B.2). The main characteristics of the described networks are summarized in Table B.1.

**Table B.1. Summary of the main characteristics of the described networks.**

| Name | Reference(s) | Short description | # parameters |
|---|---|---|---|
| ShortRes | Foucart et al. [1]–[3] | Small Segmentation network with "short-skip" connections. | ~500k |
| U-Net | Ronneberger et al. [4] | Segmentation network with long-skip connections. | ~30M |
| PAN | Foucart et al. [2], [3] | Segmentation network with long-skip connections and short-skip connections. | ~10M |
| LeNet-5 | LeCun et al. [5] | Classification network with convolutional, subsampling and dense layers. | ~60k |
| AlexNet | Krizhevsky et al. [6] | Classification network with convolutional, subsampling and dense layers. | ~60M |
| ResNet | He et al. [7] | Classification network with short-skip connections and batch normalization. | 250k-20M depending on the chosen depth. |
| DenseNet | Huang et al. [8] | Classification network with "dense blocks", where every convolutional layer has a direct connection to every subsequent layer. | 1M-25M depending on the chosen depth. |
| PSPNet | Zhao et al. [9] | Semantic segmentation network with a focus on multi-resolution processing of the image by the network. | Depends on the chosen encoder. |
| HoVer-Net | Graham et al. [10] | Instance segmentation and classification network, designed for digital pathology. | ~40M |
| EfficientNet | Tan et al. [11] | Classification network designed to be easily scalable with high performances at relatively low number of parameters. | 5M-70M depending on the scaling factor. |
| Faster R-CNN | Ren et al. [12] | Detection network combining a "region proposal network" which finds candidates bounding boxes and a classifier | Depends on the chosen encoder. |

| | | | |
|---|---|---|---|
| | | which is applied on those candidates, sharing an encoder. | |

## B.1 Networks used in our experimental work

### B.1.1 ShortRes

We introduced the "ShortRes" network in our work on artefact detection [1], and used it in our subsequent work on SNOW annotations [2], [3]. The goal of the architecture is to be very small (to be trainable on a single GPU in a reasonable amount of time in 2016), and to incorporate the short-skip connections of ResNet [7] for better convergence. A schematic representation of the architecture is presented in Figure B.1. The macro-architecture follows a classic encoder-decoder path. The encoder has a succession of residual blocks and max-pooling, while the decoder uses transposed convolutions and residual blocks. All 2D convolutions use 3x3 kernels with a Leaky ReLU activation function. The network has a total of ~500k trainable parameters.

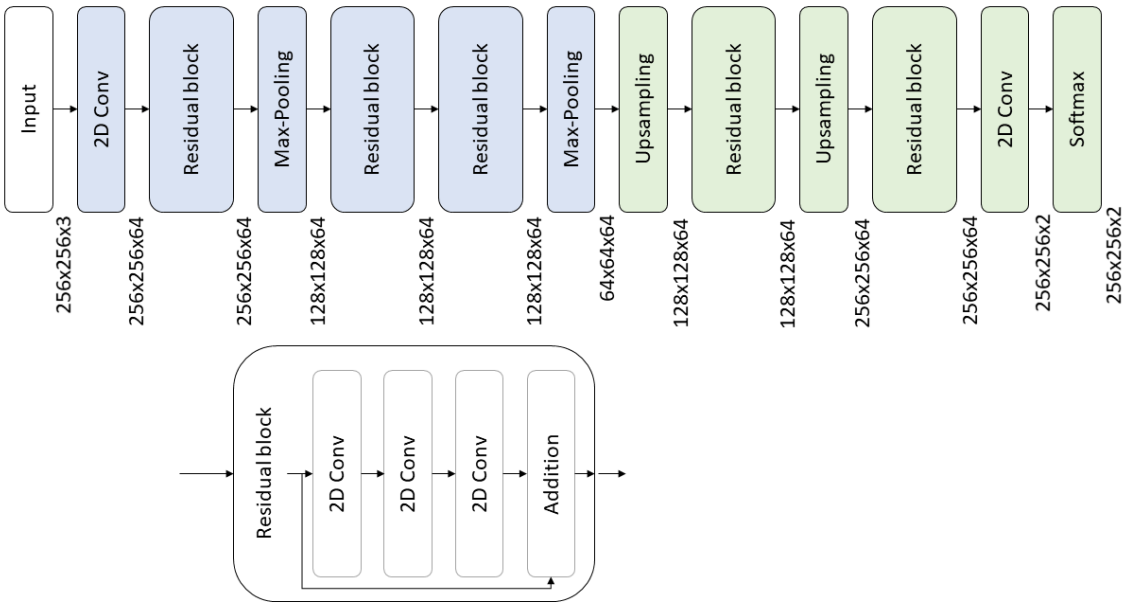



**Figure B.1. ShortRes architecture, shown here for a 256x256px input, a network width of 64 and a segmentation output, as used in our SNOW experiments** [2], [3]**. Upsampling layers are transposed convolutions. Encoder part is shown in blue, decoder in green.**

In our original artefact experiments [1], several versions of the network were tested: with different widths, added or removed residual blocks, and a "classification" version where the decoder was replaced with several dense layers.

### B.1.2 U-Net

U-Net was introduced in 2015 by Ronneberger et al. [4], and quickly became a state-of-the-art network for biomedical image segmentation. In its original version (see Figure B.2), the encoders and decoders were symmetrical, with a succession of 2D convolutions with 3x3 kernels and ReLU activation, and 2x2 max-pooling (or, conversely, transposed convolution). The main innovation of the architecture was the inclusion of "long-skip" connections, which re-introduced feature maps

from the encoder into the decoder, with the aim of providing a better spatial context to the decoder for a more accurate segmentation. The network has a total of ~30M trainable parameters.

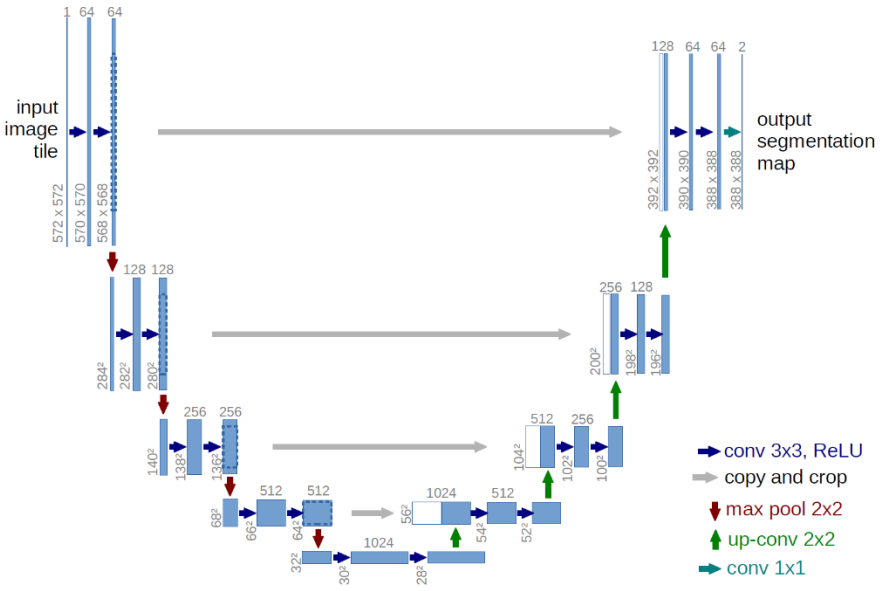



**Figure B.2. U-Net architecture, from Ronneberger et al.** [4].

As the network became more popular, and computing resources became more readily available, alternative versions of the network were developed, mainly by replacing the encoder part with encoders taken from other architectures (such as, for instance, ResNet or EfficientNet, presented in section B.2) and, often, pre-trained on large datasets such as ImageNet.

### B.1.3 Perfectly Adequate Network (PAN)

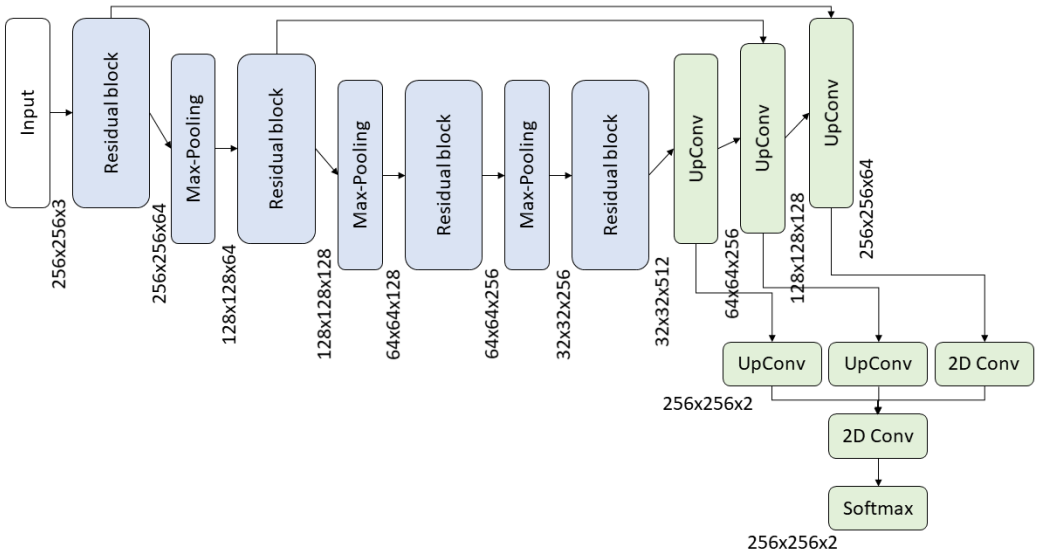



**Figure B.3. PAN architecture, used in our SNOW experiments** [2], [3].

We introduced the PAN architecture in our SNOW experiments [2], [3].

It combines the short-skip connections from ResNet, the long-skip connections from U-Net, and a final segmentation built from decoder feature maps taken at different levels (see Figure B.3), with the goal of ensuring that our results on the effects of imperfect annotations could be reproduced with different macro- and micro-architectural choices, and different sizes in terms of trainable parameters (~10M for PAN).

## B.2 Selected networks from the state-of-the-art.

### B.2.1 LeNet-5

The LeNet family of neural networks comes from the work of LeCun et al. on handwritten digits recognition [5], [13], and set the standard for the "classification" macro-architecture. LeNet-5, perhaps the best-known version, is shown in Figure B.4.

It includes a succession of convolutional and subsampling layers, followed by dense layers leading to the final class probabilities. Contrary to most commonly used pooling function today which usually don't include trainable parameters, the output $y_s$ of this subsampling layer was computed as $y_S = w \sum_i x_i + b$, with $w$ and $b$ being trainable parameters, and $x_i$ referring to the inputs, being a 2x2 neighbourhood in the previous layer.

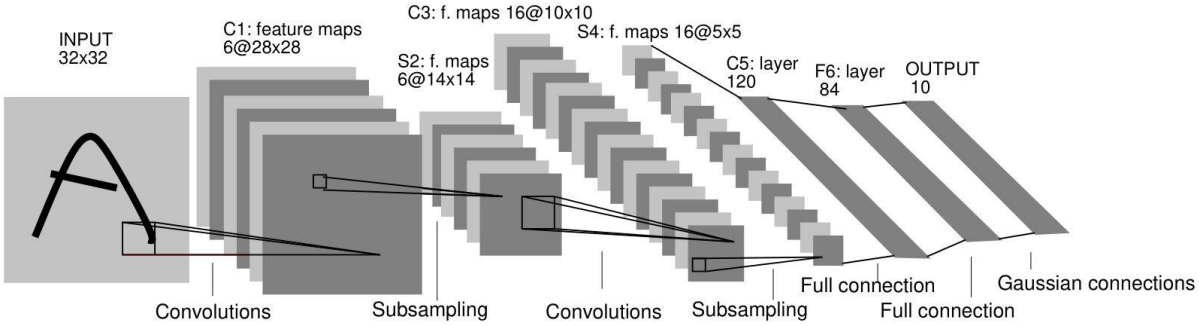



**Figure B.4. LeNet-5 architecture, originally published in LeCun et al.** [5].

Also of note, the connections in LeNet-5 did not cover the entire "width" of the network, meaning that not all feature maps of layer S2, for instance, were connected to all feature maps of layer C3. This was for performance reason, and to force feature maps "to extract different (hopefully complementary) features". LeNet-5 used hyperbolic tangents as an activation function through the network.

### B.2.2 AlexNet

AlexNet was Krizhevsky et al.'s [6] winning entry into the 2012 ImageNet competition. The macro-architecture was very similar to the LeNet architecture, following the classic encoder-discriminator suitable for classification tasks, but with a number of trainable parameters several orders of magnitude larger (~60M instead of ~60k).

The encoder part was split in two independent paths which could be trained on two separate GPUs, thus accelerating the training. Subsampling was done with a max-pooling operation, and the chosen activation function was the ReLU.

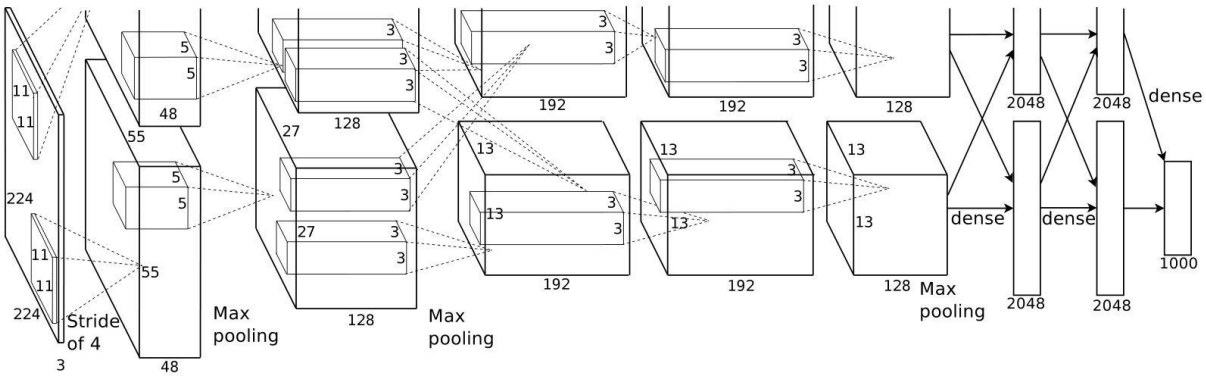



**Figure B.5. AlexNet architecture, originally published in Krizhevsky et al.** [6]

### B.2.3 ResNet

The "ResNet" family of architectures from He et al. [7] popularized short-skip connections by demonstrating great results on the ImageNet dataset. They contain a succession of "residual" blocks (shown in Figure B.6), made of two or three convolutional layers with ReLU activation functions. Subsampling is not done by pooling, but by using a stride of 2 in some of the convolutional layers. The number of parameters depends on the chosen depth, with the shallowest versions having ~250k parameters and the deepest version ~20M. Batch normalization is used after each convolution (before the activation function).

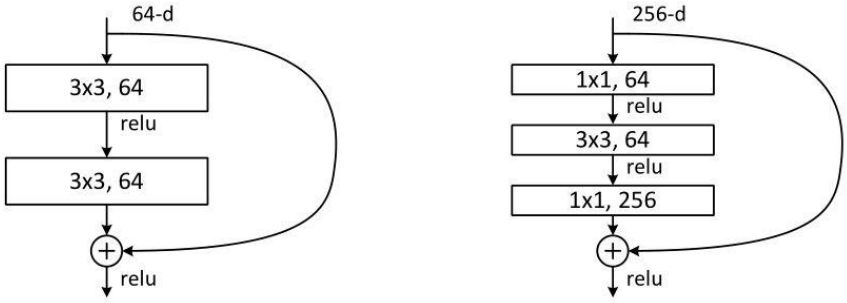



**Figure B.6. Two types of residual blocks from the ResNet architecture, originally published in He et al.** [7]. **The version of the left is for residual blocks where the dimensions of the feature maps remain the same, and the version of the right is a "bottleneck" block with subsampling in the 3x3 convolution.**

### B.2.4 DenseNet

DenseNet, introduced in Huang et al. [8], pushes the notion of skip-connections to the extreme by using "dense blocks", where every convolutional layer is connected to every subsequent convolutional layer. Subsampling is done between the dense blocks with a 1x1 convolution and a 2x2 average pooling (see Figure B.7). A particularity of DenseNet is that it works very well with a very "narrow" network, meaning that there are few feature maps per layer. ReLU activation functions are used through the network.

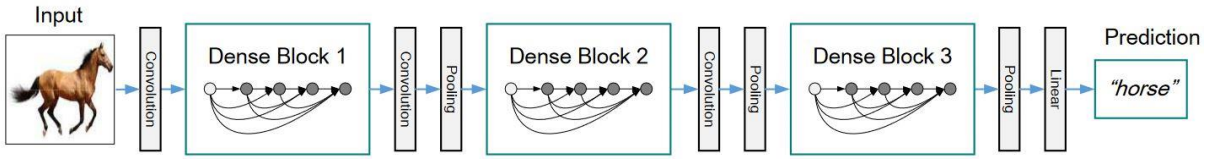



**Figure B.7. DenseNet architecture, originally published in Huang et al.** [8].

### B.2.5 PSPNet

PSPNet, introduced by Zhao et al. [9], is a semantic segmentation network with a strong focus on multi-resolution processing for natural scenes. The overall architecture is shown in Figure B.8. The network starts with a pre-trained DCNN encoder (ResNet-50 was used in the original publication). The feature map is then pooled with different pooling sizes to have a representation of the features at different levels of resolution. Each of these representations passes through a 1x1 convolution to reduce the number of features, then up-sampled with a linear interpolation and finally concatenated with the encoder's feature map, with a last convolutional layer providing the final prediction.

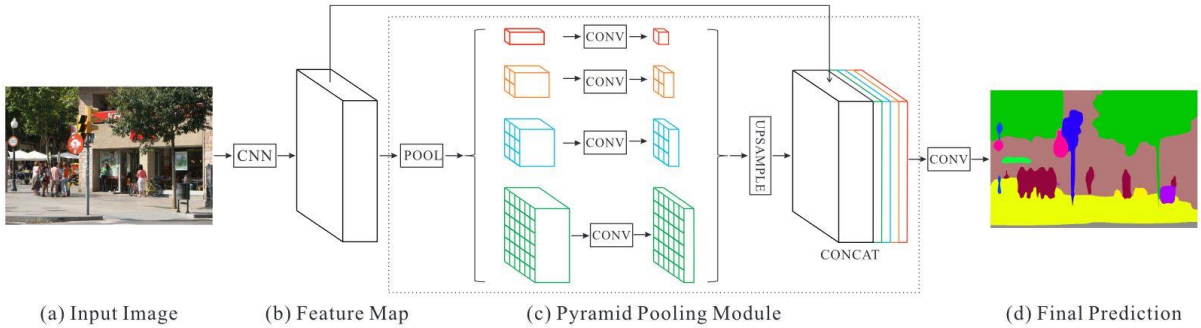



**Figure B.8. PSPNet architecture, originally published in Zhao et al.** [9].

### B.2.6 HoVer-Net

Originally designed as the "XY Network", the network now known as HoVer-Net was designed by Graham et al. [10] specifically for nuclei instance segmentation and classification in digital pathology. It is characterized by the use of three different paths for the decoder, which use the same architecture but are trained on different outputs (see Figure B.9). First, a classic binary segmentation decoder (the "Nuclear Pixel Branch") that produces a per-pixel nuclei/non-nuclei probability. Second, a decoder with two channels trained on the per-pixel regression task of predicting the X-Y vector pointing towards the centre of the nucleus (the "HoVer Branch"). The third decoder, meanwhile, is trained on the semantic segmentation task of predicting the nucleus type. The three branches are trained simultaneously. Residual blocks are used in the encoder, and dense blocks in the decoders.

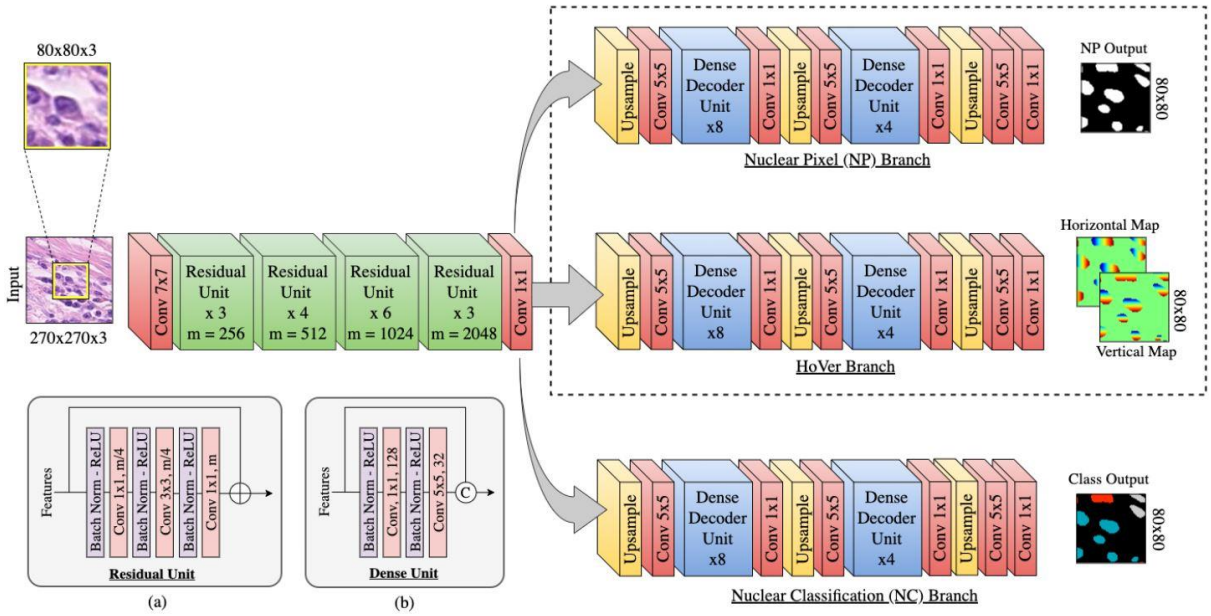



**Figure B.9. HoVer-Net architecture, originally published in Graham et al.** [10].

## B.2.7 EfficientNet

More than a network architecture, Tan et al. [11] propose with EfficientNet a method for scaling up convolutional neural networks so as to obtain high levels of performances with low numbers of parameters. The method, called "compound model scaling", uses a single parameter to simultaneously scale the depth (i.e. number of layers), width (i.e. number of channels in the feature maps) and image resolution. They propose a baseline architecture ("EfficientNet-B0") which uses residual blocks of increasing width and decreasing resolution, as usual for encoders. Then, using the compound scaling parameter, scaled-up versions of the architecture are built (from "B1" to "B7").

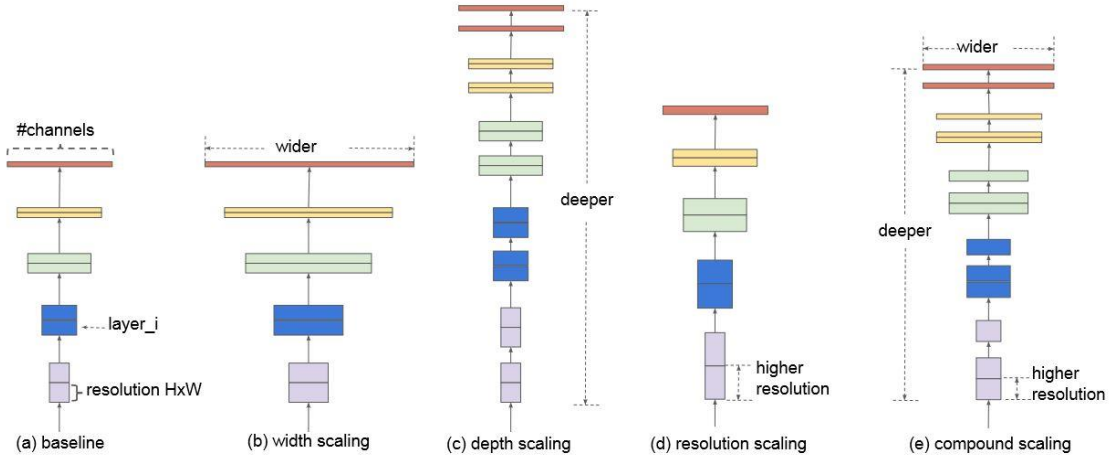



**Figure B.10. Compound model scaling in EfficientNet, originally published in Tan et al.** [11].

## B.2.8 Faster R-CNN and Mask R-CNN

Faster R-CNN, proposed by Ren et al. [12], is the successor of the R-CNN [14] and Fast R-CNN [15], previously introduced by Girshick et al. for object detection based on bounding box localisation.

It follows a classification macro-architecture, with two decoders. The first one is the "Region Proposal Network", which uses a sliding window to find candidate regions based on an "objectness" score. The second is trained to determine the class of candidate regions. The encoder is shared by both decoders.

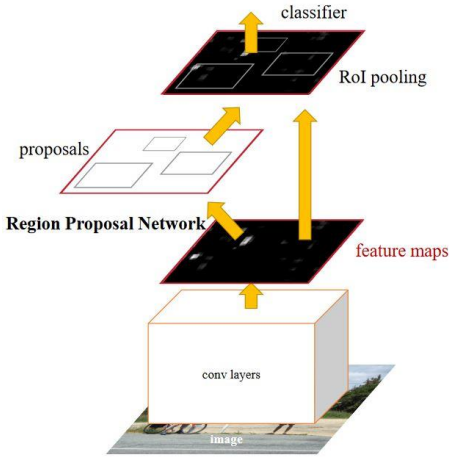



**Figure B.11. Faster R-CNN architecture, originally published in Ren et al.** [12].

## References

[1] A. Foucart, O. Debeir, and C. Decaestecker, "Artifact Identification in Digital Pathology from Weak and Noisy Supervision with Deep Residual Networks," in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, Nov. 2018, pp. 1–6. doi: 10.1109/CloudTech.2018.8713350.

[2] A. Foucart, O. Debeir, and C. Decaestecker, "SNOW: Semi-Supervised, Noisy And/Or Weak Data For Deep Learning In Digital Pathology," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Apr. 2019, pp. 1869–1872. doi: 10.1109/ISBI.2019.8759545.

[3] A. Foucart, O. Debeir, and C. Decaestecker, "Snow Supervision in Digital Pathology: Managing Imperfect Annotations for Segmentation in Deep Learning," 2020, doi: 10.21203/rs.3.rs-116512.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," 2017. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html

[10] S. Graham *et al.*, "Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Medical Image Analysis*, vol. 58, p. 101563, Dec. 2019, doi: 10.1016/j.media.2019.101563.

[11] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, vol. 97, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, 2015, vol. 28. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[13] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989, doi: 10.1162/neco.1989.1.4.541.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Nov. 2013.

[15] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, vol. 11-18-Dece, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.