Deep Learning for medical imaging: perspectives from real use cases

Università degli studi di Torino

SCUOLA DI SCIENZE DELLA NATURA

Dottorato di ricerca in Informatica

Ciclo: 33°

Deep Learning for medical imaging: perspectives from real use cases

Tesi presentata da: Daniele Perlo

Tutor: Prof. Marco Grangetto

Coordinatore del Dottorato:
Prof. Marco Grangetto

Anni Accademici 2020/2021
Settore Scientifico Disciplinare di afferenza:
INF/01 - COMPUTER SCIENCES

A Marco, Enzo e ai ragazzi del gruppo,
la cui mano è sempre tesa;
Alla mia famiglia,
che sopporta e supporta con calore;
Ai miei amici,
che sanno dare continua energia;
Alla splendida Federica, faro senza il quale sarei perduto,
come un piccolo peschereccio in una notte di tempesta.

Challenges are what make life interesting.
Overcoming them
is what makes life
meaningful.

Joshua Marine

Abstract

The interest in inserting automatic elements to support diagnostics, within the healthcare system, is one of the main actors that in recent years has driven the research and development of Artificial Intelligence (AI) applications. Technological advances, the refinement of algorithms, the computational resources increase and the extraordinary capacity of prediction, have contributed to the proliferation of AI research. Major advances have been driven by the spread of Deep Learning (DL). The healthcare institutions are still not ready to integrate AI applications in their diagnostic workflows. In spite of the advances in the research fields, the publicly available datasets are often unrepresentative of real noisy data. This difficulty lies in the inherent complexity of using data from patient screening, whose sharing and precise annotation for automatic tools is not part of the medical routine, yet. As a consequence, collaboration between institutions is not guaranteed and data are not easily shared, and therefore AI solutions exhibit little diagnostic generalization. The European Union stands at this challenge funding projects aiming to contribute filling this gap, such as *DeepHealth*. In this thesis we investigate two challenging problems related to the diagnostic prediction from visual reports; in particular, we highlight the complexity of data management and pre-processing, and how DL can support physicians in their diagnostic process. In the beginning, we will address the generalization problem from a methodological point of view. We will present our useful tool for generalization of DL models, called *PostSynaptic Potential* (PSP), based on a training regularization approach. Specifically to the medical image processing domain, we will first tackle a histo-pathological tissue classification, in particular colorectal polyps and their degree of dysplasia, and we show that DL can achieve comparable accuracy as human pathologists. Then, we switch from image classification to an image generation task. In this

scenario, we will discuss the generation of brain perfusion maps from patients with ischemia. Perfusion maps are the result of patient exams that shows vessel occlusion in the brain. We take into account brain computed tomography scans in order to predict three types of perfusion map. With the perspective of making these analyses effective tools for diagnostic pipelines, an analysis of the solutions is proposed, based on the AI development tools proposed by *DeepHealth*.

Contents

Chapter 1

Healthcare and Artificial intelligence

1.1 Overview

The landscape of healthcare and biomedical research is changing rapidly through involvement of artificial intelligence into their processes. Artificial Intelligence (AI) has recently received new scientific and public attention, with technology companies and researchers releasing new breakthroughs and technologies at a breakneck pace. For such reasons, AI technologies applications start to become increasingly prevalent also in business and society other than in healthcare and in research. Many aspects of patient care, pharmaceutical and medical processes can be heavily influenced and transformed by the potential of these technologies. Several studies have already shown that artificial intelligence can perform basic healthcare tasks, such as disease diagnosis, at least as well as humans. As an example, researchers from Google Inc., by collaborating with medical institutions, demonstrate how an AI system can be trained on a huge amount of images and how it can reach levels of sensitivity and specificity performance, comparable to clinician's ones, in diagnosing relative diabetic retinopathy and also can detect associations, previously unrecognized, between the images [32]. AI is essentially a branch of computer science that aims to understand and build intelligent

beings that are implemented as software. The first usage of an AI system in the medical field can be found at Dartmouth Conference in 1956 [70]. In the early 2000's, many practical limitations were overcome by new AI paradigms. Since 2012, the rapid and promising development of image classifiers has deeply contributed to the rise of artificial intelligence contribution in the last years. Although there have been significant advances and progress in recent decades, AI has been plagued by a constant question that is far to be resolved: "What the "true AI" is?". A "true AI", or also "strong AI", is can be defined as computer programs that can experience sentience, self-awareness and consciousness [73]. By contrast, "weak AI" is limited to the use of software accomplish specific problem solving tasks [25]. A well-known feature of AI research is that, once a certain performance goal is achieved in resolving a certain task, future progress needs to be measured by different criteria. For example, AI technology achievements from the 1970s to the 1990s, such as automatic interpretation of electrocardiograms, are probably not examples of true AI, despite once considered medical breakthroughs and now considered useful. Currently, algorithms are already better than radiologists at detecting malignancies and creating cohorts for high-risk clinical trials. More recently, the application of medical imaging systems has pushed the boundaries of AI into areas that used to be the domain of medical specialists. The AI application limits are far to be conquered and they continue to expand, also into several other medicine areas, including clinical practice, translational medical research, and basic biomedical research. However, artificial intelligence can be considered many years away from replacing humans in a wide range of medical processes. Several reviews reports the AI influence in healthcare, can be referred in order to take inspiration and information [98, 21] and to build our own perspective for the near future.

1.1.1 AI history in medicine

Researchers have developed and presented systems for clinical decision support since the mid-twentieth century. Intuitively, the medicine support was the most studied application for Artificial Intelligence from the dawn of his birth [98]. Rule-based expert systems and approaches, based on "if-then" rules, are an AI technology that became mainstream from late 1970s and and have been widely used commercially since then. In healthcare, they have been widely used for clinical decision support in recent decades and continue to be used today: the common usage is to interpret electronic health records (EHRs), then they provide support for disease diagnosis, appropriate patient treatments selection, clinical conclusions interpretations, and diagnostic hypotheses generation. Nowadays EHR vendors offer their own sets of rules for their systems. Expert systems require experts or knowledge engineers to create rule sets for specific knowledge areas. That makes rule-based systems expensive and vulnerable because they require explicit decision rules and, like textbooks, must be updated and validated by expert humans for each specific competence field. Moreover, it has a non-trivial issue: the rule systems increase in complexity by trying combine both deterministic and probabilistic elements in their decision processes. However, the cardinality of these rules increases usually more than a thousand and the rules begin to conflict with each other and raise the probability to create failure points. In addition, when changes occur in the knowledge domain, changing the rules is time consuming and difficult also for experts in their domain. These reasons are the cause of the slow replacement of expert systems in healthcare by more comprehensive approaches based on data collection and machine learning algorithms [98].

During the last decade, a massive media interest raised around *IBM Watson* [68] and its focus on precision medicine, specifically cancer diagnosis and treatment. *IBM Watson* uses a combination of machine learning and machine speech processing capabilities. However, *IBM Watson* is designed to be a set of "cognitive services", delivered through application pro-

gramming interfaces (APIs), and not a standalone product. Its services include speech and language data analysis, vision and machine learning programs. The initial enthusiasm for using the technology is diminished as the clients realized the *IBM Watson*'s training complexity: in cancer-specific treatments, integrating it into treatment processes and systems seams almost infeasible [68]. While *IBM Watson*'s APIs are technically superior, most agree that the effort to cure cancer is too ambitious. *IBM Watson* left room for others "open source" AI builders solutions, such as Google's *TensorFlow* and Facebook's *PyTorch* [98].

In the first generation of AI systems, experts organized medical knowledge and developed correct decision rules to make automatic inferences, but the recent AI research starts to develop machine learning (ML) techniques, useful to explore complex task interactions, in order to identify patterns in data [24].

As we said before, the ML research is focused today in building artificial neural networks and in developing deep learning algorithms for deep multilayer ANN. The community is hardly working to build neural networks for biological and medical applications. However, deep learning algorithms are very "data-hungry". In most of the cases, data have to come labeled and pre-processed in order to achieve useful performances for real applications. Large datasets of medical cases have become available only in recent times, thanks to the demand of large scale studies, data-collection platforms and financial incentives. EHR systems, mentioned before, will not only accelerate the collection of large amounts of clinical data, but also enable more rational use of AI systems into the clinical workflow [98]. This is only a first step, but it is often not enough to guarantee a real data collection sufficient to train ANN-based AI algorithms for many clinical tasks. In addition to the difficulty of data collection, there is also the difficulty of generalizing the use of data-based AI algorithms. By generalization of an AI algorithm, we mean the ability to predict with respect to unseen data. Consequently, for training a machine learning algorithm, the collected data

must be divided into at least two different sets, called "training set" and "test set", respectively. Consequently, the generalization capacity refers to the goodness of predictions over the "test set". Different institutions use different data acquisition techniques and machines for the same types of patient screening. Indeed, we can think at data from external medical institutions as additional "test sets" for AI algorithms. This heterogeneity is a major challenge for the reusability of the same ANN models across multiple medical institutions. We will treat the generalization problem in more detail in Sec. 3.1.

Traditionally, physicians have collected medical information about patients, performed clinical assessments, and recorded diagnoses and treatments in the medical record. After decision support systems have entered the market, with which it is possible to collect medically relevant information and make recommendations to physicians. Automatic decision support systems can be integrated into clinical workflows in a variety of ways. For example, decision support systems can proactively collect information about patients and electronic health records, make recommendations to physicians, and store the results of the system in EHR systems. In many proposed fully automated clinical systems, autonomous tools collect patient data, make decisions, and export results to the patient, but this kind of integration is far from being a consistent, usable and reliable real clinical workflow [98]. The explosive growth of healthcare artificial intelligence in recent years is due to the collection of large annotated clinical datasets, the development of machine learning techniques, open-source training packages, and affordable and rapidly increasing computing power. This can lead to a valuable transformation in medical practice in the near future, as AI systems can perform many expert-level diagnostic tasks and predict patient prognosis better than physicians[100]. As machine learning models become more sophisticated, AI could be involved in medical practice and redefine the role of physicians in the process [41].

1.1.2 Future challenges in Healthcare

We believe that artificial intelligence will play a key role in the medical service proposal in the near future. In the form of machine learning, it is a key component in enabling the development of precision medicine, which needs to constantly improve provided care. Hinton, as one of the pioneers of neural networks, and many AI researchers predicts dramatic changes in medical practice [60]. Since the early stages, AI is expected to play a role in this field, although diagnosing and suggesting treatments will be difficult. By reducing human error and physician fatigue in routine clinical tasks, artificial intelligence has the potential to drastically improve the quality of health care. However, it will not necessarily reduce the expert amount of work: as clinical guidelines recommend, high-risk patients need more frequent testing. Until now, most medical decisions have been made by humans, but the use of intelligent machines to make and support decisions raises side implications that are not strictly medical.

Ethical implications

This has led to a number of ethical issues regarding the use of AI in healthcare. Perhaps the most difficult problem to solve with current technology is transparency. Many artificial intelligence algorithms, especially deep learning algorithms used for image analysis, are almost impossible to explain or account for. If a patient is told they have been diagnosed with cancer because of a certain image, they will want to know why. *Deep learning algorithms may be impossible to explain, even to doctors who have a general understanding of how they work [21].* Artificial intelligence systems are not error free, and they could make mistakes when diagnosing and treating patients, and it can be difficult to place the blame on them. It is also possible that patients may receive medical information from AI systems, which has, from the patient's point of view, a different psychological impact than consulting an experienced physician. Machine learning systems in health care can also

be affected by algorithmic bias, predicting a higher probability of illness based on gender or race, when in fact there is no causal relationship. As AI is used in healthcare, it will face many ethical, medical, professional and technological changes. It is important that healthcare organizations and government and regulatory agencies have governance structures in place to monitor key issues, respond responsibly, and mitigate negative impacts [21].

Physician workflow implications

Successful implementation of AI for routine clinical tasks will allow physicians to focus their time on more complex tasks and devote more time to patients [98]. It also seems increasingly clear that AI systems are not meant to replace human doctors on a large scale, but to complement patient care. Over time, doctors may shift their approach to the job in favor of assignments that rely on mostly human skills such as empathy, persuasion, and general thinking. Replacing routine tasks with AI for some health care providers may result in re-design the healthcare workforce needs.However, there is little empirical evidence of such an impact on clinical practice. Current AI applications will not reach their full potential unless they are integrated into clinical workflows [9, 53]. Machine learning-based models pose a unique challenge for regulators because models evolve rapidly as data and user feedback are collected. Evaluating and updating adopted models is not trivial. For example, a new model may be better on average but perform poorly on a subset of patients. As data collection becomes a more common practice, we need a consensus framework to guide health-related data sharing. As automated AI becomes more prevalent in certain clinical tasks, the skills needed to diagnose, treat, support, and assist will need to be updated, and the roles of healthcare professionals will continue to evolve as various AI modules are incorporated into standard treatments.

Regulatory implications

For artificial intelligence systems to be widely adopted, they will require regulatory approval, integration into *EHR* systems, standardization so that similar products work in the same way, guidance to clinicians, payment by public or private payers, and updates over time in the field. These challenges will be overcome over time, but that will take much longer than the maturation of the technology itself. To address these challenges, AI researchers and clinicians will need to work together to prioritize and develop applications that address critical clinical needs. Hospital administrators must assess and mitigate disruptions to clinical workflows when implementing new AI applications. Organizations need to define an appropriate framework for prospective clinical trials to evaluate the performance of AI systems in the clinical environment. Insurers need to assess the value of AI systems in healthcare and potentially change their reimbursement policies to reduce healthcare costs and improve quality. Interdisciplinary and cross-sectoral collaboration is needed to facilitate the development and deployment of AI applications in healthcare. This is one of the most powerful and influential technologies in society and will require sustained attention and thoughtful policies for years to come.

1.2 Thesis goals and contributions

Healthcare is different from other areas where machine learning can be applied. While progress has been made in other data driven fields, what complicates medicine is not the volume of collectible data itself, but the problem arises in extracting useful insights from both complex and high dimensional data. It is these kinds of problems that make medicine the most interesting field for those who are really interested in pushing forward the AI limits. This means that we can solve real problems that are relevant to society and can affect all of us. In this thesis we want to address different issues concerning the sphere of biomedical image analysis, a perfect example of bi-

ased and high dimensional data. In particular, the focus will be on the study of Deep Learning AI techniques applicability to solve these delicate tasks. In this thesis, we will show multiple types of contributions, each one linked to the main AI application support to healthcare.

Methodological contribution

First, a careful analysis of generalization guarantees is proposed. In this context, we will see how some innovative techniques, such as "regularizations", can mitigate the undesirable effects of biased data or poor data collections. We propose, as DL "regularization" technique, *Post Synaptic Potential* (PSP) [6], detailed in Sec. 3.2, and how it can be applied to improve the overall generalization of Deep Learning applications on different datasets. This methodological contribution to the generalization support techniques pool can be extended beyond the experimental domain and applied to contexts such as the medical domain. Real-world problems, as in healthcare, are more complex to handle rather than research tasks on well-known dataset. Solutions to this tasks have to be tuned and rethought in order to fit the complexity of the data and the retrieved insights must be relevant to the domain expert. Therefore, methodologies and studies on multi resolution classification in biomedical image analysis trough Convolutional Neural Network (CNN): Histopathological colorectal tissue analysis and dysplasia grade prediction for polyp classification are will be further discussed, by also proposing a novel class inference method, above our analysis results [5], based on stacked multi resolution classifiers [1].

Experimental contribution

We will see how AI can be intended to solve these problems and we look at these solutions in the context of EU *DeepHealth* project. *DeepHealth* provides us the support to create large-size medical image datasets and propose its own Deep Learning framework ecosystem to face these challenges. In an

era in which large, foreign and private technological companies can expand the use of their software resources into very sensitive domains, such as healthcare, a common reference tool becomes necessary for the evaluation of AI-based solutions. As experimental contribution to the project, we use the *DeepHealth Toolkit* to reproduce a specific medical image analysis task based on computed tomography. The application performs an automatic processing for perfusion map generation in patients with ischemic strokes and cerebral vessel occlusions [4]. Moreover, the experimental contribution continues with an overview of technical metrics comparison will be provided in order to find strong and weak points in the current European toolkit for deep learning development. This comparison will take into account the Computer vision and the Deep Learning libraries provided in the European toolkit, at the current release stage, and the most popular DL library for research purpose, *PyTorch*.

Dataset contribution

The dataset collection in the healthcare context is the key factor to train AI models and to develop AI applications able to generalize well across institutes data. The data itself is also challenging to collect due to many factors. For example, the raw data samples usually have very high resolution and the sample distribution follows the disease appearance in the patients. For both medical cases, histopathological colorectal polyps classification and brain perfusion maps generation, we share to the community the datasets we collect. As a dataset contribution to the healthcare research community, we created UniToPatho [2] and UniToBrain [3].

Figure 1.1: The logo of *DeepHealth*

1.3 European Union in Healthcare and AI: DeepHealth Project

Part of the research we are going to discuss in this thesis have contributed to the development and to the evaluation of the *DeepHealth* project products. Deep-Learning and HPC to Boost Biomedical Applications for Health (*DeepHealth*) project [22] is funded by the European Community *Horizon 2020* research and innovation program under the topic *ICT-11-2018-2019 HPC and Big Data enabled Large-scale Test-beds and Applications*. Kicked-off in mid January 2019, *DeepHealth* is a 3-year project and is expected to conclude its work in June 2022. The aim of *DeepHealth* is to offer a community available unified framework completely adapted to exploit underlying heterogeneous and parallel High-Performance Computing (HPC) and Big Data architectures. The *DeepHealth* framework is assembled with state-of-the-art techniques in Deep Learning (DL) and Computer Vision (CV). In particular, the project will combine HPC infrastructures with Deep Learning and Artificial Intelligence (AI) state-of-the-art developing tools to support and integrate biomedical applications that require the analysis of large, complex and biased biomedical datasets and thus, new and more efficient ways of automated diagnosis, monitoring and treatment of severe diseases. Healthcare is one of the driving key sectors of the global economy, especially in Europe: the level of current healthcare expenditure was over EUR 1000 billion in 2018 [28]. Any improvement in healthcare systems has a high

impact on the welfare of the society. The use of technology in health care is undoubtedly an important path to improve its efficiency: both individuals and government budgets can benefit from the effectiveness of the use of technology. Today, European public health systems generate large data sets, mostly biomedical images and other general biomedical data, since most medical examinations use image-based processes. These datasets are constantly growing and represent a large untapped knowledge base because much of their value comes from interpretations by medical experts, and this process is usually time-consuming and manual. In addition, global knowledge sharing between institutions is complex due to different or unclassified information systems, as well as data confidentiality restrictions and limits.

In the context of automation and acceleration of data and process analysis in healthcare, scientific discovery and innovation in healthcare is expected to move rapidly toward the so-called "fourth science paradigm", which is based on the integration of traditionally unrelated and heterogeneous environments for high-performance computing and big data analysis. Under this paradigm, the *DeepHealth* project will provide HPC computing power at the service of biomedical applications; and apply Deep Learning (DL) techniques on large and complex biomedical datasets to support new and more efficient ways of diagnosis, monitoring and treatment of diseases.

The aim of *DeepHealth* is to offer a unified framework completely adapted to exploit underlaying heterogeneous HPC and Big Data architectures; and assembled with state-of-the-art techniques in Deep Learning and Computer Vision. In particular, *DeepHealth* framework is envisioned to tackle real needs of the health sector and facilitate the daily work of medical personnel and the expert users in terms of image processing and the use and training of predictive models without the need of combining numerous tools. To this end, the project will combine High-Performance Computing (HPC) infrastructures with Deep Learning (DL) and Artificial Intelligence (AI) techniques to support biomedical applications that require the analysis of large and complex biomedical datasets and thus, new and more

efficient ways of diagnosis, monitoring and treatment of diseases. Moreover, two new libraries, the European Distributed Deep Learning Library (EDDLL) and the European Computer Vision Library (ECVL), will be developed and incorporated in the *DeepHealth* framework for manipulating and processing the images in a more efficient way and thus, for increasing the productivity of professionals working on biomedical images. As a result, improved diagnostics will greatly improve the quality of health care for society, making health care systems more efficient and affordable for all.

Great expectations come from the massive usage of innovation technologies in the heath care sector. The exploitation of jointly collected health big data [18] and the powerful promises in Artificial Intelligence (AI)/Machine Learning (ML) [17] can lead to a significant metamorphosis in diagnosis process. The *DeepHealth* project aims at contributing to such epochal change exploiting one of the "biggest big data", i.e. biomedical multidimensional images and state of the art AI based on Deep Learning (DL) methods [47, 21, 93, 12]. DL is obtaining astonishing results in many sectors, with healthcare making no exception. The DL approach is based on training deep and complex Neural Networks – more specifically, Convolutional Neural Networks (CNNs) which are well suited to images. The training of neural networks from large datasets of images requires significant amounts of computing resources and this is why one of the main goals of *DeepHealth* is to combine and optimize the computing power of HPC systems with Big Data software for efficiently training DL algorithms. Moreover, in order to develop industry-ready solutions, it is of utmost importance to consider the flexibility, scalability and efficiency of these solutions. Within this context, *DeepHealth* aims at providing not only the maximum raw performance, but also at designing a framework capable of automatically selecting the most efficient setups, balancing performance, energy and accuracy.

Chapter 2

Introduction to Deep Learning

2.1 Deep Learning overview

In its early days, artificial intelligence (AI) focused on rule-based systems that made predictions by using predefined sets of rules. These rules have to be provided and verified by experts. However, these systems were fragile and relied on expert advice, which eventually led to their obsolescence. As the size and volume of data increased, these approaches were replaced by a more data-driven approach, called *machine learning* (ML). Machine learning is a set of algorithms and processes that allow machines to discover and understand patterns in data and take advantage of that structure to reason and make inference about specific tasks. In order to understand these unknown patterns, many approaches are possible. As we know, one of the most notable advances in AI in recent years has been called Deep Learning. Machine learning techniques, and the subsequent development of AI applications, aim to detect previously unrecognized patterns in data. These pattern discoveries are made without having to define, for each specific task, manually crafted decision rules by domain experts, by taking account complex similarities between inputs. Unravelling data patterns directly from the sources is an advantage in terms of domain experts effort and working time. Indeed,

machine learning has become the mainstay of AI tools. Natural language translation, pattern recognition and gaming are some of the tasks where deep learning models can match or exceed human performance. The recent success of AI has been also driven by the use of now available large data sets. In the past, collecting huge amounts of data samples in a given domain was prohibitive in terms of human resources and physical resources, like storage or bandwidth to share the dataset. Nowadays, many of the technological limits have become less constraining, therefore the data collection effort is mitigated. These large sources of data are needed in order to train multi-level Artificial Neural Networks (ANN), also called *Deep* neural networks. An important factor of the rising success of Deep Learning (DL) is both the availability of real-world data sets and the constantly growing hardware capability to handle their processing. What is the relationship between machine learning and Deep Learning? How does deep learning fit into the field and what are the main applications and challenges? There is a misconception that deep learning is a competing technology in the field of machine learning. Deep Learning is a sub-field of machine learning. Deep learning is a machine learning technique that is characterized by its computational depth. With improved computer performance in terms of memory, storage end GPUs capability, and the availability of large data sets, deep learning algorithms are able to learn patterns that are hidden in the data itself, and thereby make predictions. Deep learning can be thought of as a branch of ML that learns from huge amounts of data and takes advantage of a large number of computing units at the same time to output the designed predictions. To create a learning system that works like a human, the architecture of deep learning draws inspiration from the structure of the human brain. Thus, many of the fundamental concepts used in deep learning are familiar to neuroscience. Just as neurons are the basic building blocks of the brain, the architecture of deep learning has its own "neuron" as a minimal computational unit. The first ANN composed by a single neuron was called *perceptron* and its concept was introduced in 1958 by the psychologist Frank Rosenblatt. It is with this

artificial neuron that the magic of deep learning begins. This basic computational unit weights its input and produces a single output signal, much like the neurons in the human brain transmit electrical impulses that cross all the human nervous system. The artificial neuron is designed to filter, or catch, some elementary pattern of the data: by stacking multiple neuron layers, we have a structure that is sensitive to much more complex hierarchical patterns, and, at the end, capable of understanding a representation also in the most complex real-world data sets. A layer can be seen as a large set of smaller computational units that learn to identify repeated input values. Each perceptron layer is sensible for interpreting a particular pattern in the data. Because the perceptron design and these perceptron networks remind the way neurons form networks in an animal brain, their architecture is called a neural network (NN or artificial neural network (ANN). NN are so become the natural model for deep learning because of their power and scalability. We are going to observe and analyze in details the neuron/perceptron structure later in the discussion, that is represented in Fig. 3.2 in Sec. 3.2.2.

2.2 Deep Learning applied to images

The basic architecture of a deep neural network consists of an input layer, an output layer, and several hidden layers in between. The feedforward neural network is the simplest model, which consist in one single hidden layer mades of perceptrons, that forward the inputs in the input layer to the output layer. From there, the research in the AI field gives us a multitude of more complex models in order to fit each sort of task and manage multiple types of inputs. As proof, the success of Deep Learning was raised by the models capability to solve complex tasks on processing images. For example, *Autoencoders* are used for dimension reduction or provide useful additional outputs, like reconstructing images. Autoencoder variants, like *U-Net*, are designed to produce segmentation/semantic maps. Generative Adversarial Networks (GAN) can even generate

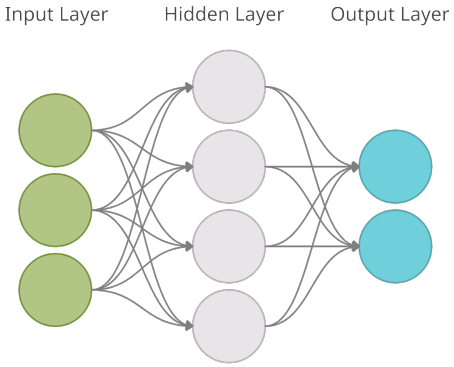


Figure 2.1: Visual representation of feedforward neural network with an hidden layer made of 4 perceptrons. Each perceptron receive as inputs all the outputs signals from the previous layer.

completely new images with the same characteristic of the input ones. Recurrent neural networks are useful for processing time-series data. Deep residual neural networks, introduced as *ResNet*, are used for image classification and they are based on traditional deep feed-forward neural networks to avoid saturating model performance by missing connections [35]. Modern ANNs have millions of parameters, and training them requires enormous computational resources. Fortunately, most recent advances in computer and processor design have provided the necessary processing power for deep learning. For example, modern graphics processing units (GPUs) could perform over a trillion floating point operations per second, so they can process millions of medical images per day at relatively low cost. Some modern neural networks can reach more than 100 layers in depth. Neural networks with multiple layers can model complex relationships between inputs and outputs, but may require more data, computation time, or complex architectural design to achieve optimal performance. Various layers, mathematical operations on neurons, and normalization methods have been developed to achieve this results. For example, recurrent layers use circular connections to model temporal events, while convolutional layers are useful for extracting spatial or temporal connections. Deep Learning relies on processing power,

large data sets, and "convolutional" neural networks (CNN) to handle tasks in the image domain. CNNs use convolutional types of layers to transform and and compress local groups of pixels in an image to extract some high-level features. Convolutional layers introduction into ANNs have become a standard technique for extracting local spatial correlations on multidimensional inputs such as images. Before CNNs, image features had to be identified and extracted by hand-crafted or rule-based models, and the performance of machine learning models depended on the quality of those features. A major improvement of CNNs is that they work with the original image and can learn useful features from the training set, simplifying the learning process and facilitating the recognition of image models. This is critical to the success of deep learning in image analysis and has contributed to the revolution that has emerged also in medical imaging.

Depending on the type of task the AI wants to solve, the main machine learning algorithms can be broadly classified as "supervised" and "unsupervised".

- Supervised DL algorithms are possible by collecting a large number of "training" instances containing input data (e.g. in medical domain they can be tissue images) and desired output labels (e.g. a carcinoma diagnosis for such tissue). Supervised ML methods require both data and the prediction (also called *label*, *class* or *ground truth*) that the model is supposed to infer from the data itself. The model is trained to produce the correct prediction for a given data instance, by learning the relevant features in the training data samples. The optimal set of parameters of the model by minimizing the difference between model predictions and data labels for the training instances, with the expectation that the learned class-discriminative features can be generalized to data instances outside the training dataset. As we have seen before, classification, regression, and characterization of similarities between instances with similar final labels are some of the most widely used tasks for supervised

machine learning models.

- Unsupervised learning can identify potential patterns in unlabeled data and clusterize them, by creating a low-dimensional representation of it. This is useful to identify outliers in the data. It is important to note that the identification of low-dimensional representations of labeled instances can be achieved in a more efficient supervised manner, but the labeling is always an expensive task in terms of expert effort, cost and time consumption. Commonly to all the DL algorithms, Unsupervised DL methods need also a wide training dataset, that contains input data but do not require labels.

2.3 Deep Learning tasks

As mentioned earlier, there are many different tasks that deep learning algorithms attempt to solve. Each of these goals is usually defined through the definition of a *loss* function, or *objective* function. This allows to evaluate and quantify, during each iteration of the training algorithm, the error that the network commits in predicting the expected result with respect to a certain input x. In the following we will denote by y the prediction of a neural model $M(\cdot)$ with respect to x, and by $\hat{y}$ the expected label result. As described above, $\hat{y}$ will only be available for supervised tasks. The training algorithm of a neural network aims, therefore, to find a configuration of the model parameters for which the loss function is minimal. The evaluation of the loss function itself is the means that provides the direction of updating of the parameters. How is this possible? By calculating the *gradient* (i.e., the *derivative*) of the function with respect to each model parameter and updating them by a *gradient descent* based rule, for example *Stochastic Gradient Descent*. Stochastic gradient descent (SGD) is a way to minimize some objective function $L(\Theta, x^i, \hat{y}^i)$, in which the arguments are model's parameters Θ and the i-th input-label pair in the training set, by updating the parameters in the opposite direction of the gradient of the objective function.

In other words, by following the direction of the slope of the ideal surface outlined by $L(\cdot)$, the error will be downhill until it reaches a valley. We can describe the update step as

$$\Theta = \Theta - \eta \cdot \nabla_\Theta L(\Theta, x^i, \hat{y}^i) \tag{2.1}$$

where η is the *learning rate*. The *learning rate* is an hyper-parameter of the training process, it can be interpreted as how much the evaluation of $L(\Theta, x^i, \hat{y}^i)$ influences the network parameters each training iteration.

Now we can see some widely used cost functions to solve deep learning some tasks.

- Multi-class classification: Each image, or sample, can belong to one of C classes. This task is treated as a single classification problem of samples in one of C classes. The *ground truth* (or label) $\hat{y}$ is represented by a one-hot vector t with a single positive class. Usually, the ANN M have C output neurons, one for each class, and we referenced it as y. The common choice for this task is to use a *Categorical Cross-Entropy* (CCE) loss. The CCE loss compares two given probability distribution (y and t in our case) and it have his minimum when $y = t$. In order to use the CCE as $L(\cdot)$ function, we need to define $\sigma(\cdot)$ as the *softmax* function. $\sigma(\cdot)$ basically transform the model output values y in to a probability distribution.

  $$\sigma(y^i) = \frac{e^{y^i}}{\sum_j^C e^{y^j}} \tag{2.2}$$

  where i is a given class index. $\sigma(y^i)$ gives the probability of predicting the class i for $y = M(\Theta, x)$. Therefore CCE can be defined by using $\sigma(\cdot)$:

  $$CCE = -\sum_i^C t^i \cdot \log(\sigma(y^i)) \tag{2.3}$$

  Note that t is a one-hot vector, so the CCE is evaluated only for the true label $\hat{y}$. By adjusting Θ to minimize the

cross entropy function, we find a configuration in which $y - t < \epsilon$, where ϵ is a low error value.

- Multi-label classification: Each sample can belong to more than one class. The target vector t will be not an one-hot vector, and it can have more than a positive class. This task can be seen the sum of independent classification problems, where each output neuron decides if a sample belongs to a class or not. In this case *cross entropy* loss can be used, in its binary formulation, but not the *softmax* function: we do not want to predict exclusively a single class, but each class can be predicted exclusively. The $\sigma(\cdot)$ function, in this case, is the *logistic* function. It can be used to transform the ANN output y into a vector of independent probabilities, therefore each value is in the range $[0, 1]$ as for each element of t.

$$\sigma(y^i) = \frac{1}{1 + e^{-y^j}} \tag{2.4}$$

Straightforwardly, the total loss value have to be the mean of the *cross entropy* applied to each y independently, therefore *binary cross entropy* (BCE) is defined as

$$BCE = -\frac{1}{|C|} \sum_{i}^{C} t^i \cdot \log(\sigma(y^i)) + (1 - t^i) \cdot \log(1 - \sigma(y^i)) \tag{2.5}$$

- Image generation: For each example x we want to generate an image y that respects certain characteristics. In this case $\hat{y}$ can be a segmentation map, a binary map, or even x itself. Therefore, $\hat{y}$ has a cardinality (or number of pixels) that is a some function $d(\cdot)$ of the cardinality of x: $|\hat{y}| = d(|x|)$. A neural network designed for this task, usually CNN, produces an output y for which $|y| = |\hat{y}|$. Usually $\hat{y}$ has values in range $[0, 1]$, a sigmoid function has to be applied to y values. For this kind of problems, a standard objective function choice is the *mean squared*

error (MSE) loss.

$$MSE = \sum_{i}^{n} \frac{(\sigma(y^i) - \hat{y}^i)^2}{n} \tag{2.6}$$

in which n is the number of $\hat{y}$ pixels and $\sigma(\cdot)$ is the logit function defined in Eq. 2.4. Note the *mean squared error* loss is proportional to the ℓ_2 norm function. Other regression losses can be obtained changing the exponent value. In this case, the error of each output neuron contributes to evaluate the mean error evaluation. MSE minimal value is 0 when $y = \hat{y}$. By finding a configuration for parameters Θ, that minimizes MSE, the model $M(\cdot)$ generates images y which are very similar to $\hat{y}$.

In the following chapters, by exploring in detail some *DeepHealth* AI use-cases, we will encounter two different types of problems: multi-class classification and image generation.

Chapter 3

Regularize Deep Neural Networks

3.1 The regularization problem

Training a deep neural network with high generalizationis a challenging task. A model with too few parameters will fail to learn the problem in its complexity, while a model with too much power will learn not only the data features but also it can retrain the whole dataset. In both cases, we get models that generalize poorly to unseen data samples. Next, we briefly discuss the problem of overfitting in neural network training and how it can be solved (or mitigated) by performance-enhancing regularization techniques [11]. As we seen before, the goal of training a NN model is to have a predictor that can performs well on training samples as for external data, in the same domain, that make the predictor useful in real applications.

"The central challenge in machine learning is that we must perform well on new, previously unseen inputs — not just those on which our model was trained. The ability to perform well on previously unobserved inputs is called generalization." [31]

The model is designed to learn from known cases and generalize these known cases to new cases in the future. Techniques such as *K-fold cross-validation* or *train/validation/test splits* are used to evaluate the ability of the model to generalize to

new data. Learning and generalizing to new examples is difficult: if too little time is spent on training, the model will perform poorly on the training dataset and on new data and the model will be not able to scale to the given problem. This can be referenced as an example of *underfit model* and it is is characterized by high bias and low variance. Otherwise, if

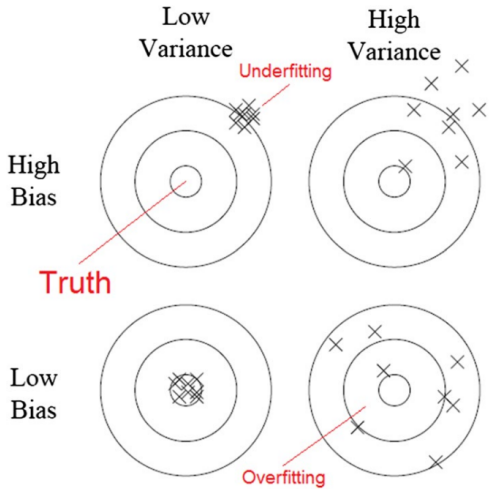


Figure 3.1: Visual representation of solutions given by model with high bias or variance. The picture is taken from Moradi *et al.* [58]

too much time is spent on training, the model will perform too well on the training dataset, but it will have inaccurate predictions on the new data: this is an occurrence of *overfit model*. It has low bias and high variance. The model performance can varies widely even with statistical noise added to samples in the training dataset. In both cases, the model does not generalize.

"In order to generalize well, a system needs to be sufficiently powerful to approximate the target function. If it is too simple to fit even the training data then generalization to new data is also likely to be poor. [...] An overly complex system, however, may be able to approximate the data in many different ways that give similar errors and is unlikely to choose the one that will generalize best ..."[63]

We can solve the problem of underfitting by improving the capacity of the model: the capacity refers to the ability of the

model itself to handle different features; the greater the capacity, the more types of features the model can handle and transform the inputs into outputs. The performance of a model can be easily improved by changing the architecture of the model, such as adding more nodes in its layers, adding more layers and/or substituting them. Because poorly fitted models are easy to deal with, it is more common to find overfitted models. An overfit model can be easily diagnosed by evaluating the model on both the training dataset and the validation dataset and examining the model's performance during training. If we maintain a graph for model performances during the training phase, we can see what is called *learning curve* or *training error*. A good fit model should have a low training error and a low validation error. If the model has a high training error and a low validation error then it could be an overfit model. For example, the loss curve of the model on the training and validation datasets (which we try to minimize) shows a line for the training dataset that drops and may reach a plateau (*learning curve*), while a line for the validation dataset first drops and then starts to rise again at some point.

"As training progresses, the generalization error may decrease to a minimum and then increase again as the network adapts to idiosyncrasies of the training data."[63]

At the very end, we want boosting performance but also we want to prevent data overfit. Regularization is a way to introduce constraints to parameters: the learning algorithm finds difficulties to both fit the constraints and learn entirely the available data. This "auxiliary information" can be very valuable to the learner when the data itself is insufficient. In general, each part of the learning process or prediction procedure that is added to reduce flaws in the data, is called regularization. In fact, the regularization process reduces the variance of the model by assuming some sample's correlation exists. In machine learning, we can separate the sampled data point into two parts: the description and the random noise. Any machine learning algorithm must be able to model the description and ignore the noise. Also, algorithms that focus on fitting noise and features as well as the description, cre-

ate model overfit [58]. In this case, regularization allows us to address the model complexity choice, so that the model can make better predictions the future. Deep models are capable of learning complex representation spaces, which are needed to solve complex learning problems. However, the depth and capacity of the model needed to cover such representations is often excessive. Therefore, the development of new efficient regularization methods is inevitable to solve the problems of model overfitting and generalization. These techniques are critical features that have to be taken into consideration by developing a learning algorithm based on training deep neural networks. These regularization strategies can be divided in different families, so a more structured grouping is needed.

- Regularization by data manipulation: the concept is to manipulate the images (or whatever kind of data) that feed the neural network during the training loop. The aim is to make the training robust to input perturbation during the inference phase. This regularization family includes all data augmentation techniques as affine transformations like flip or rotation and domain specific transformations (like contrast enhancing, color jiggling, grayscale transformation, etc.. for images). Also cropping, adding gaussian noise or input dropout fall into this category. The aim is the same as before, reached by hiding random features.

- Regularization by architecture selection: each task and dataset can require a specific artificial network architecture that fits properly to the purpose. By regularization through architecture selection, we do not just refer to selecting state-of-the-art model architectures. All different choices of layers (convolutions, pooling layers, dropout) or layer blocks, by which we can edit or compose models, are also included. The design of a network can also be a simplified or pruned [86, 87] of well known models or, otherwise, doesn't have to follow a specific model.

- Regularization by optimizer choice: Lin *et al.* [52] shows

how the objective function "sharp" minima does not generalize as well as "wide" minima; by choosing between different and properly designed optimizers, we can avoid to stuck our training loop in a local or "sharp" minima [44], like Adam [45] or SGD. Beside the optimizer, we include in this regularization family also network initialization techniques together with early stopping and cross validation techniques.

- Regularization by loss term regularizer: to the global objective function (also called loss or cost function), a weights constraint term is added. The simplest and perhaps most common regularization method is to add a penalty to the loss function in proportion to the size of the weights in the model. This approach will minimize the cost function in the train loop, by taking in consideration also these constraints and boosting the generalization performances. ℓ_2 regularization is a widely used representative for this category.

"Unless your training set contains tens of millions of examples or more, you should include some mild forms of regularization from the start." [31]

3.2 Post Synaptic Potental

Generalization improvement of prediction results is always the main goal and challenge in training a predictive model such deep neural network. During this enthusiastic deep learning era, several tricks and techniques, aiming to boost model performances on unseen data, have been proposed. Research effort can not be confused to the more general deep neural network architecture enhancing, which also increase generalization performances, and helps the model evolution through time, from AlexNet [49] to VGG [77] or ResNet [35] and much more in the years that followed. Just to name some performance boosting examples, that are retrieved from state of the art training

pipeline, we can cite data augmentation techniques in the input domain, or neural networks layer, such Dropout-[82] or Batch Normalization [39]. In addition, many other refinements have been proposed with the same objective, like optimization methods [16, 45], or even cost function regularization like ℓ_2 (also known ad *weight decay*) regularization.

Of course, during the experimentation and exploration of such techniques and their effective application to our context, we face benefits and disadvantages. ℓ_2 or *weight decay* regularization is generally used combined with other types of regularization choices. This method has been widely used in the training of most artificial neural networks and even with its simple formulation, it improves generalization to various scenarios. About this topic, similarity was observed between weight decay and dropout by Wager *et al.* [90]. By investigating the meanings and underlying implications of ℓ_2 regularization, we figured out a novel technique that achieves more sophisticated learning strategies performances and we called it *PostSynaptic Potential regularization* (PSP) [6]. Like weight decay, PSP falls in the large regularization family by loss term regularizer. The following subsection is dedicated to the definition of PSP and to the underlying theoretical explication. Images, notation and experiments are taken from the original publication *Post-synaptic Potential Regularization Has Potential* at *ICANN2019*) [6].

3.2.1 PSP definition

In order to define *PostSynaptic Potential regularization*, we have to introduce some notation that help us to understand how it works. Let our artificial neural network to be an acyclic, multi-layer, for a total of N layers, where layer $l = N$ the output layer and $l = 0$ is the input layer. The set of all trainable and trained parameters in the neural network is defined in Sec. 2.3 as Θ. Let the l-th layer contain K_l neurons. If the the l-th layer is a convolutional layer, K_l will be the the number of filters. Therefore, let $k \in [1, K_l]$, we consider the k-th neuron in the l-th layer with:

- $\mathbf{y}_{l-1}$ as input vector.
- $y_{l,k}$ as its own output signal.
- $\Theta_{l,k}$ as its set of parameters: $b_{l,k}$ is the neuron bias, while $\mathbf{w}_{l,k}$ are the weights. The j-th weight is identified as $w_{l,k,j}$

An affine function $f_{l,k}(\cdot)$ takes the weighted input vector. This affine function can be a dot product, convolution, batch normalization or any of their combination. An activation function $\varphi_{l,k}(\cdot)$ is applied later to produce the neuron output. So, the neuron output can be expressed by

$$y_{l,k} = \varphi_{l,k}\left[f_{l,k}\left(\Theta_{l,k}, \mathbf{y}_{l-1}\right)\right] \tag{3.1}$$

The (3.1) is simplified by defining the *post-synaptic potential* $z_{l,k}$ as

$$z_{l,k} = f_{l,k}\left(\Theta_{l,k}, \mathbf{y}_{l-1}\right) \tag{3.2}$$

Post-synaptic potential can also be referred as the pre-activation potential. Hence, the core of this regularization strategy is $z_{l,k}$, that is the main actor of the following analysis.

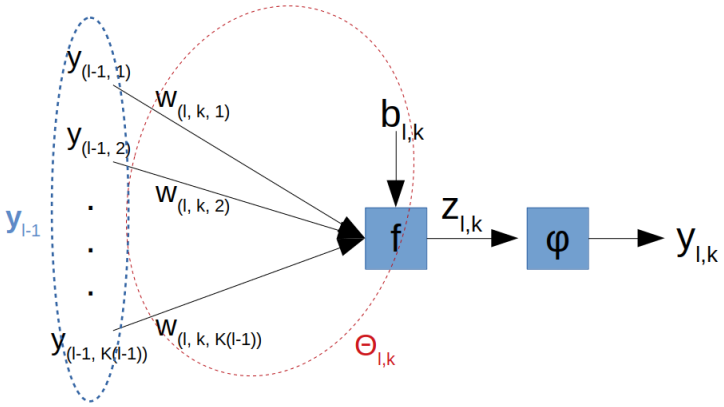


Figure 3.2: Representation of a neuron. In this schema $z_{l,k}$, or also the *post-synaptic potential*, is reported. The image and the notation is taken from Tartaglione *et al.* [6]

A graphic summary of the above notation is reported in Fig. 3.2.

3.2.2 Weight decay consequences on the post-synaptic potential

We have to introduce first the well-known ℓ_2 regularization term, that is used in several learning strategies:

$$R_{\ell_2}(\Theta) = \frac{1}{2} \sum_l \sum_k \sum_j \Theta_{l,k,j}^2 \tag{3.3}$$

Together with the objective function $L(\cdot)$, Eq. (3.3) is minimized. The overall minimized function $J(\cdot)$ is defined as:

$$J(\Theta, x, \hat{y}) = \eta L(\Theta, x, \hat{y}) + \lambda R_{\ell_2}(\Theta) \tag{3.4}$$

where η, λ are positive real numbers, commonly in range $(0, 1)$, and $\hat{y}$ is the desired output. The back-propagation strategy is used to compute all the update contributions to the weights. Let the focus be the regularization term (3.3), minimizing it corresponds to adopt the commonly named *weight decay* strategy. The update rule for *weight decay* is the following:

$$\Theta_{l,k,j}^{t+1} := (1 - \lambda)\Theta_{l,k,j}^{t} \tag{3.5}$$

This obviously generates perturbations of the output for the corresponding neuron, therefore the result is a perturbation of the post-synaptic potential:

$$\Delta z_{l,k} = z_{l,k}^{t+1} - z_{l,k}^{t} \tag{3.6}$$

How does the ℓ_2 regularization influence $z_{l,k}$?
If we minimize (3.3), we expect $\Theta_{l,k,j} \to 0 \forall l, k, j$. Now, if we have an input pattern for our network $\mathbf{y}_0$, it is straightforward, according to (3.2) and (3.5), that $z_{l,k} \to 0$, because all the parameters $\Theta_{l,k}$ will be zero.
This conduces to thinking to the effect to weight decay, if $z_{l,k} \to 0$, the output of the neuron $y_{l,k}$ will have a limited range of values, because $y_{l,k} \to \varphi(0)$. This is a characteristic region in the mostly-used activation functions: for the ReLU activation we are close to the derivative discontinuity of the

function, while, in the case we use *sigmoid* or *hyperbolic tangent*, we have the maximum value for the $z_{l,k}$ derivative. If focusing on this region is the reason why weight decay boosts the performance in generalization, we can formulate a regularization term in which $|z_{l,k}|$ is minimized explicitly.

3.2.3 Post-synaptic regularization

We have observed that, if we take into account the typical deep learning scenario, the weight decay regularization minimizes the post-synaptic potential of the neurons, by focusing the output of each neuron around some characteristic regions. This might help the back-propagated signal and, indirectly, favor the generalization performances.
If we impose an ℓ_2 regularization on $z_{l,k}$, we will explicitly drive the post-synaptic potential of the neuron and its output $y_{l,k}$. We define the regularization term as:

$$R = \frac{1}{2} \sum_{l} \sum_{k} \left(z_{l,k}\right)^2 \tag{3.7}$$

where k is an index ranging for all the neurons in the l-th layer. For each of the k neurons in the l-th layer, we can rewrite (3.7):

$$R_{l,k} = \frac{1}{2} \left(z_{l,k}\right)^2 \tag{3.8}$$

What is the perceived update by the parameters of our model? We can check the update contribution by using the chain rule on applying the (3.8) regularization:

$$\frac{\partial R_{l,k}}{\partial \Theta_{l,k,j}} = \frac{\partial R_{l,k}}{\partial z_{l,k}} \cdot \frac{\partial z_{l,k}}{\partial \Theta_{l,k,j}} = z_{l,k} \cdot \frac{\partial z_{l,k}}{\partial \Theta_{l,k,j}} \tag{3.9}$$

Expanding (3.9), we have

$$\frac{\partial R_{l,z}}{\partial \Theta_{l,k,j}} = \frac{\partial z_{l,k}}{\partial \Theta_{l,k,j}} \cdot \left(b_{l,k} + \sum_{i} w_{l,k,i} y_{(l-1,i)} \right) \tag{3.10}$$

Bias and weight cases are slightly different. Let $\Theta_{l,k,j}$ the bias, then (3.10) can be written as:

$$\begin{aligned}\frac{\partial R_{l,k}}{\partial b_{l,k}} &= 1 \cdot \left(b_{l,k} + \sum_i w_{l,k,i} y_{(l-1,i)} \right) \\ &= b_{l,k} + \sum_i w_{l,k,i} y_{(l-1,i)} \end{aligned} \tag{3.11}$$

Otherwise, let $\Theta_{l,k,j}$ be one of the neuron weights. The derivative, by isolating $w_{l,k,j}$ contribute from the summation in Eq. 3.10, the will be rewritten as follows:

$$\begin{aligned}\frac{\partial R_{l,k}}{\partial w_{l,k,j}} &= \frac{\partial z_{l,k}}{\partial w_{l,k,j}} \cdot \left[w_{l,k,j} y_{(l-1,j)} + \left(b_{l,k} + \sum_{i \neq j} w_{l,k,i} y_{(l-1,i)} \right) \right] \\ &= \frac{\partial z_{l,k}}{\partial w_{l,k,j}} \cdot \left[w_{l,k,j} \frac{\partial z_{l,k}}{\partial w_{l,k,j}} + \left(b_{l,k} + \sum_{i \neq j} w_{l,k,i} y_{(l-1,i)} \right) \right] \\ &= w_{l,k,j} \left(\frac{\partial z_{l,k}}{\partial w_{l,k,j}} \right)^2 + \left(b_{l,k} + \sum_{i \neq j} w_{l,k,i} y_{(l-1,i)} \right) \frac{\partial z_{l,k}}{\partial w_{l,k,j}} \\ &= w_{l,k,j} \left(\frac{\partial z_{l,k}}{\partial w_{l,k,j}} \right)^2 + C_{l,k,j} \frac{\partial z_{l,k}}{\partial w_{l,k,j}} \end{aligned} \tag{3.12}$$

where $C_{l,k,j}$ is the contribution to $z_{l,k}$ from all the parameters except for $w_{l,k,j}$. By assuming $\frac{\partial z_{l,k}}{\partial w_{l,k,j}} = 1$ and $C_{l,k,j} = 0$, we can recover the weight decay formulation.
According to (3.9), the parameter value variation is

$$\Delta \Theta_{l,k,j} = -\lambda z_{l,k} \frac{\partial z_{l,k}}{\partial \Theta_{l,k,j}} \tag{3.13}$$

We can see that $z_{l,k}$ is a bounded term, because we are minimizing (3.7). In addition, we need to distinguish between two different cases for $y_{(l-1,j)}$:

- $l \neq 1$: $y_{(l-1,j)}$ is defined as the output of the $(l-1)$-th layer: if the post-synaptic potential is minimized also in previous layers, taking into account commonly-used activation functions, it is definitely a bounded quantity.

- $l = 1$: in this case, $y_{(0,j)}$ represents the network input, which we assume to be bounded.

Hence, also (3.13) is a ranged quantity, because it defined as product of bounded quantities.
However, we want to minimize the whole sum in (3.7), rather than (3.8). By explicitly writing the regularization contribution to the parameter $\Theta_{l,k,j}$ and by considering R application to the layer p and neuron h, we found

$$\frac{\partial R_{p,h}}{\partial \Theta_{l,k,j}} = z_{p,h} \cdot \frac{\partial z_{p,h}}{\partial \Theta_{l,k,j}} \tag{3.14}$$

and that take us into the analysis of three different cases:

- $p = l$: the gradient term $\frac{\partial z_{p,h}}{\partial \Theta_{l,k,j}} = y_{(l-1,j)}$ if $h = k$, zero otherwise. The regularization effect on a neuron is not affected from adjacent neurons in the same layer.

- $p < l$: the gradient term is $\frac{\partial z_{p,h}}{\partial \Theta_{l,k,j}} = 0$ and its contribution falls to zero.

- $p > l$: in the most interesting case, our regularization applied to the last layer affects all the previous ones. All the network parameters on previous levels receive a contribution for each following regularized level. By back-propagation, such contribution is automatically computed.

If we take in consideration the general case, the whole update contribution, resulting by minimize (3.7), on the k-th neuron's j-th weight in layer l is indeed

$$\Delta \Theta_{l,k,j} = -\lambda \left[z_{l,k} \frac{\partial z_{l,k}}{\partial \Theta_{l,k,j}} + \sum_{p=l+1}^{L} \frac{\partial R_p}{\partial \Theta_{l,k,j}} \right] \tag{3.15}$$

where

$$\frac{\partial z_{l,k}}{\partial \Theta_{l,k,j}} = \begin{cases} 1 & if\ \Theta_{l,k,j}\ is\ bias \\ y_{(l-1,j)} & if\ \Theta_{l,k,j}\ is\ weight \end{cases} \tag{3.16}$$

In order to better understand the above update contribution, a graphic visualization is shown in Fig. 3.3. So we have seen how a post-synaptic potential regularization, which minimizes $z_{l,k}$ explicitly, works in all the neurons of an artificial neural network. The analysis shows how the update term for a parameter includes global information employs a global information originating from succeeding layers. This "feedback" favors the regularization itself.

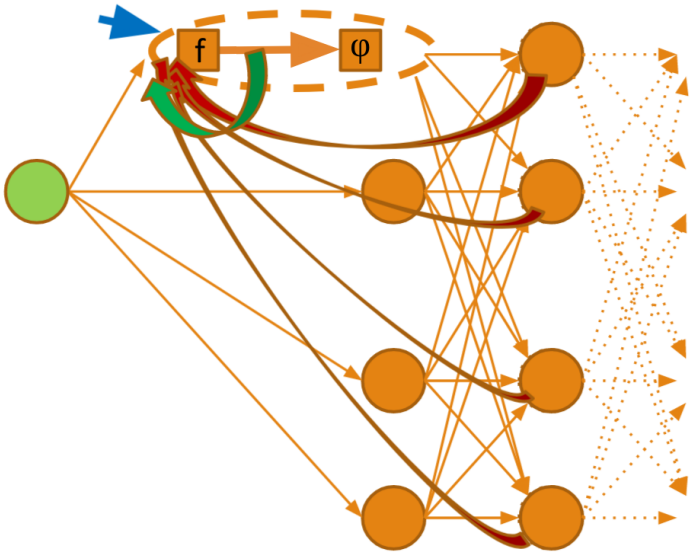


Figure 3.3: Representation of the update contribution in Eq. 3.15. By considering the on the k-th neuron (blue arrow), we highlight $z_{l,k}\frac{\partial z_{l,k}}{\partial \Theta_{l,k,j}}$ contribution in back-propagation from z in the same neuron (green arrow) and $\sum_{p=l+1}^{L}\frac{\partial R_p}{\partial \Theta_{l,k,j}}$ contribution from subsequent layers (red arrows). Therefore, the parameters in previous layers are tuned to regularize z also in following layers.

3.3 PSP experiments

We report the performance reached by some of the commonly used artificial neural networks by applying our post-synaptic potential regularization (PSP). We compare those performances to the results obtained by applying weight decay.

We tested our regularization techniques on three different datasets: MNIST, Fashion-MNIST and CIFAR-10 by using LeNet5, ResNet-18 [35], MobileNet v2 [71] and All-CNN-C [81] architectures. For each reported training result, the used model hyper-parameters are selected by using grid search over multiple initialization seeds. All the simulations are performed using the standard SGD with CUDA 8 on a NVIDIA Tesla P-100 GPU. Our regularization has been implemented using PyTorch 1.1.[1]

We start by showing our attempted to train the widely studied LeNet-5 model over the standard MNIST dataset [51]. The MNIST dataset is a database of digitized handwritten digits, in form of 28x28 pixels grey-scale pictures. This database is provided with 60k training images and 10k test images. We use SGD with a learning parameter $\eta = 0.1$, mini-batch size 100. The training lasts here 50 epochs. In Fig. 3.4, we show a typically observed scenario in our experimental setting, where we compare standard SGD optimization strategy without using any regularization, the effect of both ℓ_2 and ℓ_1 regularization (by using hyper-parameter $\lambda = 0.0001$), the effect of dropout regularization on the network (by using a commonly used drop probability $p = 0.5$) and our pre-activation signal potential regularization (PSP, $\lambda = 0.001$). The results are shown in Table 3.1, that reports the obtained classification errors, averaged on 10 different runs along with the corresponding standard deviation. While the weight decay average performance is 0.74%, PSP performance is 0.55%. Please note that this result is deemed statistically significant since the t-test rejects the null hypothesis that there's no difference between the means with p-value in order of 10^{-5}.

We would like to emphasize that, using this simple training strategy, the performance reached on LeNet-5 is among the best recorded, nevertheless having a much lower computational complexity [16]. We find interesting the behavior of $\langle z^2 \rangle$ for all the three techniques (Fig. 3.5). In the case of standard SGD, the averaged $\langle z^2 \rangle$ value, as it is not controlled,

[1] All the source code is publicly available at `https://github.com/EIDOSlab/PSP`

typically grows until the gradient on the loss will not be zero; the same behaviour is shown for dropout regularization. For ℓ_2 regularization, interestingly, it slowly grows until it reaches a final plateau. Finally, in PSP regularization, the $\langle z^2 \rangle$ value is extremely low, and still slowly decreases. According to the results in Fig. 3.4, this is helping in the generalization.

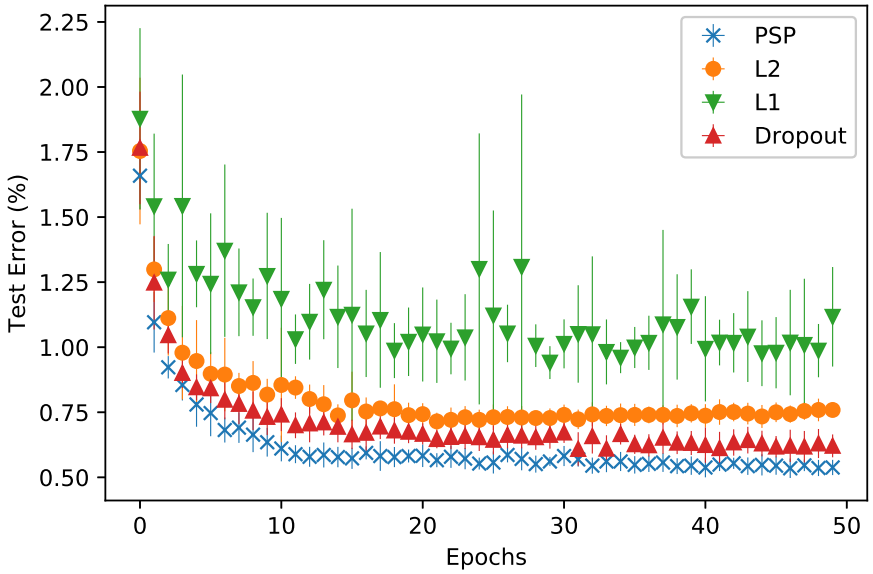


Figure 3.4: Error on the test set during the training phase. We report the average error across multiple network initialization seeds. Vertical lines indicate the standard deviation on error values.

We want to have a further look at what is happening at the level of the distribution of the parameters layer-by-layer. A typical trained parameters distribution for LeNet5, trained on MNIST, is shown in Fig. 3.6. While ℓ_2 regularization typically shrinks the parameters around zero, PSP regularization does not constrain the parameters with the same strength, while still constrains the pre-activation signal (Fig. 3.5). However, contrarily to this, the first convolutional layer, with ℓ_2 regularization, is less constrained around zero than with PSP (Fig. 3.6a). Such a behavior can be explained by (3.15): all

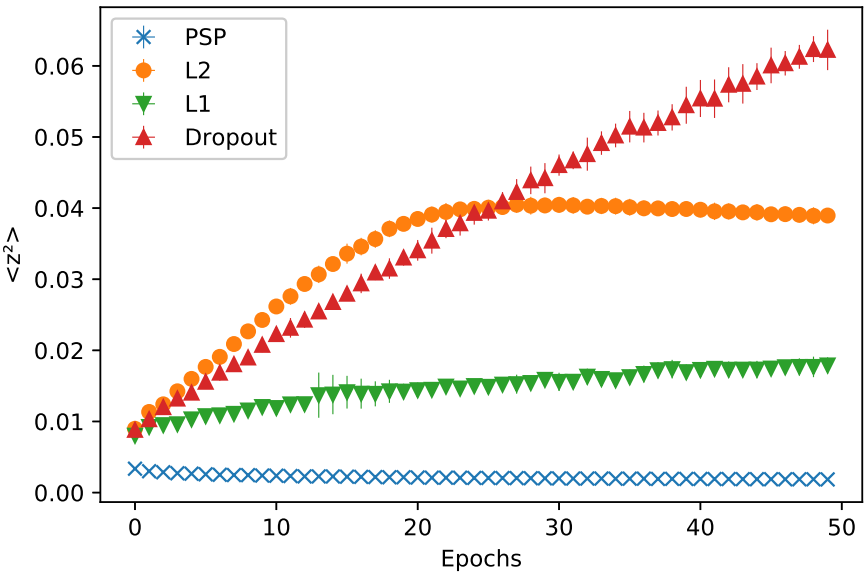
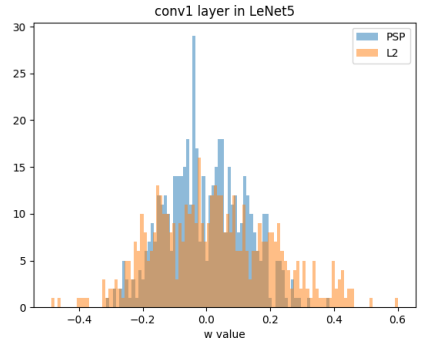


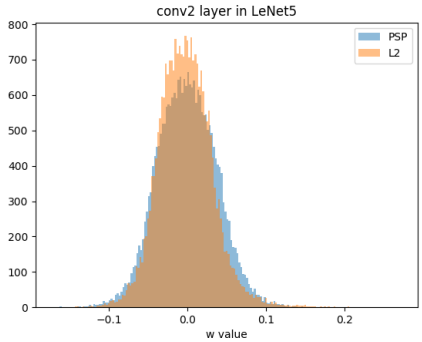
Figure 3.5: Performance comparison in LeNet5 trained on MNIST between ℓ_2 regularization, ℓ_1 regularization, dropout and post synaptic potential regularization (PSP). All the plots show an average on 10 runs. The graph shows Average values for z^2. Vertical lines indicate the values standard deviation. We can see how these are almost invisible in the case of using PSP.

the regularization contributions coming from all the forward layers (in this case, conv2, fc1 and fc2) affect the parameters in conv1, which are directly conditioning all the z computed in forward layers.

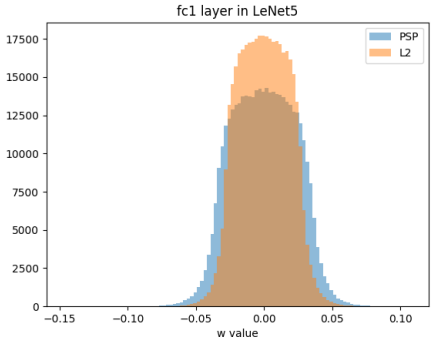
As a further step, We decided to test LeNet-5 on a more complex dataset: hence, we have chosen the Fashion-MNIST dataset [96]. It is made of 10 classes of 28×28 grey-scale images representing various pieces of clothing. They are divided in 60k examples for the training set and 10k for the test set. Such a dataset has two main advantages: the problem dimensionality (input, output) is the same as MNIST; hence, the same ANN can be used for both problems, and it is not as trivial as MNIST to solve. Training results are shown in



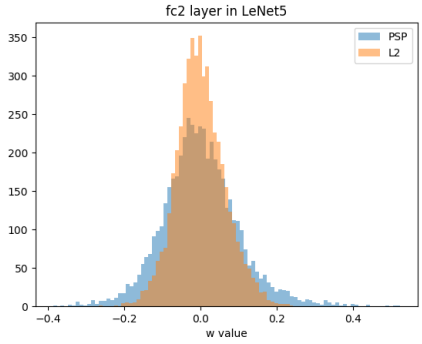
(a) First conv. layer (conv1)



(b) Second conv. layer (conv2)



(c) First fullyconnected layer (fc1)



(d) Output layer (fc2)

Figure 3.6: Typical distribution of the parameters in a LeNet5 trained on MNIST with ℓ_2 regularization and with post synaptic potential regularization (PSP). We can see how PSP usage pushes the top layers parameters to a more homogeneous value distribution than ℓ_2.

Table 3.1. The simulations are performed with $\eta = 0.1$, batch size 100 and $\lambda = 0.0001$ for both ℓ_2 and ℓ_1 regularization while $\lambda = 0.001$ for PSP.
The training lasts here 150 epochs. Here, the difference in the generalization between ℓ_2 and PSP is wider than the one presented for MNIST: we are able, with the same architecture, to improve the performance by around the 1% without any other heuristics.

Moving towards deep architectures, we decided to use CIFAR-10 as dataset. It is made of 32x32 color images (3

Table 3.1: Classification error on LeNet-5 for MNIST and Fashion-MNIST evaluated on an average of 10 runs (error % ± standard deviation)

Regularization	MNIST	Fashion-MNIST
ℓ_2	0.74(±0.06)	8.28(±0.73)
PSP	**0.55**(±0.05)	**7.88**(±0.28)
ℓ_1	1.02(±0.26)	9.56(±0.64)
Dropout	0.63(±0.08)	8.22(±0.25)

channels) divided in 10 classes. The training set is made of 50k images and the test set of 10k samples. This dataset is a good compromise to make the first tests on deep architectures as the training is performed from scratch.
Three convolutional architectures have been here tested: MobileNet v2 [71], ResNet-18 [35] and All-CNN-C [81]. In order to separate the contribution of our regularizer towards other state-of-the-art regularizers, we are going to compare our results with our baseline (same data augmentation, no dropout). All these networks were pre-trained with $\eta = 0.1$ for 150 epochs and then learning rate decay policy was applied (drop to 10% every 100 epochs) for 300 epochs. Mini-batch size was set to 128 and momentum to 0.9. For standard training, the ℓ_2 λ was set to 0.0005 while for PSP regularization to 0.001.

All the results are shown in Table 3.2. Please note that

Table 3.2: Top-1 classification error on CIFAR-10 (Error %)

Architecture	ℓ_2	PSP	ℓ_1	Dropout ($p = 0.1$)
ResNet-18	5.1	**4.6**	6.2	6.4
MobileNet v2	7.0	**6.4**	6.9	7.7
All-CNN-C	9.1	**8.6**	9.4	N/A

ALL-CNN-C includes by design some dropout layers, having $p = 0.5$; as a consequence, all the presented results for this architecture includes dropout. Our results show that PSP-based regularization generalizes better than ℓ_2, ℓ_1 and

dropout.

Chapter 4

Colorectal polyps classification

4.1 Histopathological tissue diagnosis

Histopathology literally refers to the study of tissues associated with some disease. Histopathology is the study of organic tissues and the analysis of biopsies, performed by a pathologist. Therefore, it is also the examination of tissues under a microscope for signs of disease. In clinical medicine, it refers specifically to the examination of specimens from biopsies or surgical procedures that are processed by a pathologist and histological sections are placed on glass slides. The histopathology report describes the microscopic features of the tissue and cancer sent for examination. Therefore, histopathological reports are sometimes referred to as "biopsy or pathology report". Pathological examination of tissue begins with surgery, biopsy or autopsy. Tissue is removed from the body or plant, usually freshly dissected by a specialist, and then placed in a fixative to stabilize the tissue and prevent degradation. The most common fixative is formalin. A physician who specializes in microscopic examination is called a pathologist. The tissue to be examined is taken from a biopsy or surgical procedure, and a sample of the suspect tissue is selected and sent to the laboratory. The tissue is then cut into very thin layers (called sections), stained, and examined under a microscope

to reveal the details of the cells in the tissue [33]. This can be done on chemically fixed slides or frozen section slides. In order to examine the tissue under the microscope, the slides are stained with one or more stains. The purpose of the stain is to show the cellular composition, and the contrast color is used to create contrast. Histochemistry is the science of studying the chemical reactions between laboratory chemicals and tissue components. The most common stain used in histology is a combination of hematoxylin and eosin (often abbreviated as H&E). Hematoxylin is used to stain the nucleus blue, and eosin is used to stain the cytoplasm and extracellular connective tissue matrix pink. There are hundreds of other methods to selectively stain cells. Saffron, silver salts and artificial dyes are also other techniques used to stain tissue sections. On the other hand, histopathological sections provide a more complete picture of the disease and its effect on the tissue because the basic structure of the tissue is preserved during the creation process. Thus, disease features, such as lymphocytic infiltration of cancer, can only be inferred from histopathological images [33]. The grading system varies depending on the type of cancer being evaluated, but is generally based on the abnormalities of the cells under the microscope, with low grade tumors appearing more normal and higher grade tumors showing more abnormalities [72]. In other words, high-grade tumors are usually tumors with many cellular abnormalities. Grading is different from staging. The staging of cancer surgery is related to the part of the body where the cancer is located and how much it has spread. The importance of quantitative pathology image analysis has been recognized by image analysis and pathology researchers. Since most current pathologic diagnoses are based on the subjective (but empirical) opinion of the pathologist, there is a clear need for quantitative evaluation based on digital slide images of pathology. Such quantitative analysis of digital pathology images is important not only for diagnosis but also for understanding why a particular diagnosis was made (e.g., specific chromatin structures in the nucleus of a cancer cell may indicate a specific genetic abnormality). In addition, histopathological imaging has be-

come the *"gold standard"* for the diagnosis of many diseases, including almost all types of cancer [37]. The appropriate use of such image processing will enable a more accurate characterization of images from a diagnostic perspective. For example, enhancing cell nuclei features can be indicative of cancer status diagnosis. While the additional structures in these images provide a wealth of information, they also pose new challenges for automated image analysis.

4.2 AI and histopathological images

Today, automated diagnostics based on medical imaging is probably the most successful application area for medical AI. As we said before, any medical specialties, such as radiology, ophthalmology, dermatology, pathology, etc., use diagnostic imaging. Histopathological evaluation is the gold standard for the diagnosis of many cancers. In this process, biopsies are processed on tissue sections, stained with dyes, and the sections are then interpreted by expert pathologists based on visual assessment under a microscope. This process is commonly referred to as histopathology. However, differences between pathologists have been reported [83, 23] and proposed procedures cannot be easily extended. In addition, extended studies on histopathological images can only be done by collecting them from multiple source institutions. The ability of the human expert pathologist to predict survival in cancer patients [99] suggests that pathology sections contain a wealth of information that has not yet been exploited. DL models have been used for classification and segmentation tasks of histological images since 2012 [40, 78, 97]. Janowczyk *et al.* [40] provide a detailed survey on the distribution of the application of DL across different pathology domains, like tubule segmentation, mitosis detection and nuclei segmentation. Advanced applications of automated image processing for microscopic tissues scans also include detection of metastatic breast cancer [57, 59], grading gliomas [27], basal-cell carcinoma [19]. Janowczyk *et al.* describes well known models,

such AlexNet[48] and VGG [76], today almost used for benchmarks. Regarding the colorectal cancer domain, Sirinukunwattana *et al.* [79] present a DL-based method to perform nucleus center detection and classification on four different types of nuclei from microscopic whole-slides. An extensive literature is focusing on applying deep learning in image analysis in order to evaluate endoscopic images for the detection of polyps [54]. In contrast, a fewer number of studies have attempted to address the challenges of histopathological examination, by using deep learning analysis, of whole digital slides for colorectal polyps [47, 80, 92] that are the core of this thesis. These efforts have yielded promising results, but the sequences analyzed contain a small amount of samples and, most importantly, no attempt has been made to evaluate both of the important mandatory features of diagnostic evaluation (i.e., histological type and degree of dysplasia) simultaneously. Korbar *et al.* [47] present a crop-based framework, developed using a ResNet architecture [34], to classify different types of colorectal polyps from whole-slide images. This work provides empirical suggestions the residual network architecture achieves better performance than other models. Following their previous work, Korbar *et al.* introduce a revised version of Grad-CAM (gradient driven class activation mapping) [74] to visualize the attention map of the network for the annotated whole-slide [47].This work was able to show guided Grad-CAM with boxes can visualize the most relevant and informative regions for the network, in according with pathologists annotation. A similar approach can be seen in Kather *et al.* [43] on colorectal cancer classification. Subsequently Bychkov *et al.* [12] use convolutional and recurrent neural network and combine different architectures predictions, in order to predict five-years probabilities of disease survival, caused by colorectal cancer, and estimate risk for each individual. Bychkov *et al.* discusses the idea of feeding features extracted from CNN image clipping into an LSTM network to use spatial information. Due to the high impact of DL frameworks on medical images processing performances, a modern open-source plat-

form, *ORBIT*[1][84] has included also some deep models for multiple tasks like image segmentation, classification, tissue quantification. Reyes *et al.* [65] enhanced data formats to describe whole-slides at multiple resolution level. Recently the focus shifted in detecting and predicting polyp type during colonoscopy without submitting it for histology, saving time, and costs [101, 75, 94, 13]. Recently, Wei *et al.* [92] propose an analysis model to evaluate the automatic predictions of annotated tissue. It perform an evaluation of the generalization capability of neural model across datasets coming from external medical institutions. Automated systems can address the shortage of full-time pathologists [66] and improve the quality of care for cancer patients through rapid and objective evaluation of histopathology specimens. In general, successful applications in radiology, dermatology, ophthalmology, and pathology have used labeled big data, computational power, and deep learning techniques to achieve expert-level diagnostic accuracy. Applying this research to clinical practice will not be easy, but it has the potential to fundamentally change current medical practice.

4.3 Colorectal polyps classification

We already seen how the expansion of cancer screening programs and the demand of colonoscopy surveillance routines are leading the gastrointestinal histopathology to grow. In the medical image processing field, the raw data, e.g. 2D scans like tissues Hematoxylin & Eosin histology biopsy or 3D scans like brain Magnetic Resonance, has huge dimensions for each sample and high variance compared to their diagnosed class of belonging. By the definition of their collection process, high resolutions helps the specialist in pathology in his analysis and diagnostics.

A predictive signal of possible gastrointestinal cancer development are colorectal polyps, which are pre-cancerous lesions located in the lining of the colon. Colorectal tissue samples

[1]Website www.orbit.bio

are collected by biopsies and colonoscopies. Gastrointestinal pathologists examine them to find signs that predict tissue neoplastic process and invasive carcinomas. The histopathological characterization of colorectal polyps allows management correction, and follow-up, of the patient with the ultimate goal of prevention or early detection of invasive carcinoma. Colorectal polyps are characterized using histologic analysis of tissue samples to determine the degree of malignancy and polyp dysplasia. Deep neural networks have achieved superior accuracy in medical image recognition, but they require large sets of annotated images for training.

Within *DeepHealth*, a corresponding use-case is formulated: we focus our effort to develop a neural network-based pipeline to automatically diagnose colorectal cancers from Whole-Slide Images (WSI). WSI are super high-resolution images retrieved by scanning biopsies material, more specifically Hematoxylin & Eosin-stained slices of tissue, from patients undergoing cancer screening. These images can reach more than 100.000×100.000 pixels in their size. This description is taken mainly from the book chapter we wrote within *DeepHealth* activities [4].

The first goal of the use-case is to collect more than a thousand of labelled and annotated WSI, in accordance with the evaluation of the UNITO pathologists' team. The slides are acquired through a professional scanner "Hamamatsu Nanozoomer S210" at 20× magnification: the result is an image with physical/digital resolution image ratio of 0.4415 µm/pixel. Those images have to fit the data demanding for Deep Learning algorithms, in order to propose a complete and ready-to-use classification application for histopathological images.

The WSI corresponds to the whole specimen collected from the patient: it is provided with a label (the diagnosis) and the annotation on the subset of tissue taken into consideration by the pathologist to elaborate the diagnosis (the so-called diagnostic tissue). The experts collected six different types (or classes) of tissue:

- Normal tissue (**NORM**): biopsies of normal tissue, without any specific disease.

- Hyperplastic polyp (**HP**): it is a very frequent lesion, single or multiple, benign, without pre-neoplastic significance. It is due to an excess cell growth of glandular components. Hyperplastic polyps usually exhibit no malignant potential [88], while adenomas are more likely to progress into invasive carcinomas.

- Tubular adenoma, high-grade dysplasia (**TA.HG**): it is a rather frequent pathology with pre-neoplastic significance and the risk of neoplastic transformation into carcinoma is directly proportional to the number of adenomas and the size of each adenoma. Adenomas are associated with a grade of dysplasia, which measures the abnormality in cellular growth and differentiation [38]. The dysplasia is a pre-neoplastic lesion with a high-risk factor. Higher grade dysplasia indicates higher malignant potential.

- Tubular adenoma, low-grade dysplasia (**TA.LG**): such dysplasia degree has a lower risk factor.

- Tubulo-villous adenoma, high-grade dysplasia (**TVA.HG**): similar to tubular adenoma, it has a slightly different macroscopic structure and a major risk factor.

- Tubulo-villous adenoma, low-grade dysplasia (**TVA.LG**): Tubular and tubulo-villous are common colorectal adenomas, with villous adenomas generally presenting higher malignant potential given the larger surface [88].

Clinical and Surgical Pathology department of University of Turin provide the labelled set of whole-slide biopsy images that will be exploited by machine learning experts in the team for models training and testing on *DeepHealth* HPC infrastructures. In the context of *DeepHealth*, it was possible to publish

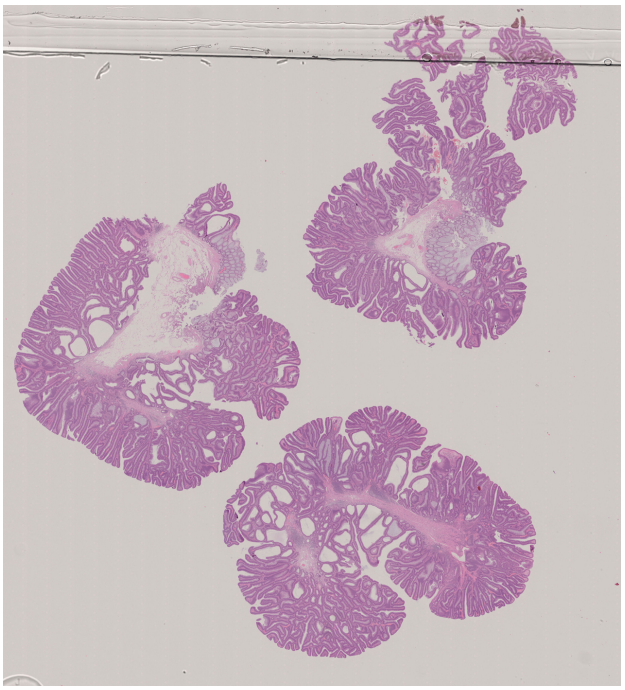


Figure 4.1: Example of Whole Slide Image (WSI)

part of the collected and annotated data in a public dataset that we called *UniToPatho*.

4.4 UniToPatho

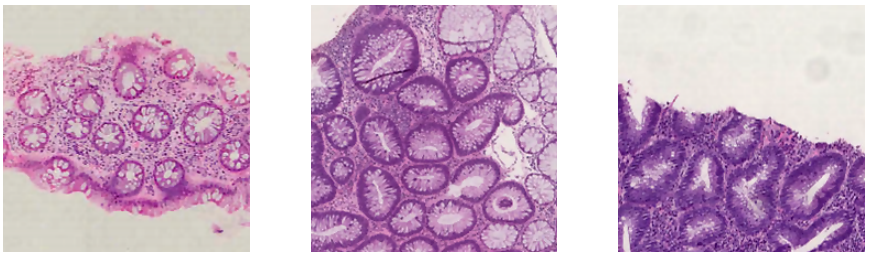
Figure 4.2: The logo of *UnitoPatho*

	HP	NORM	TA		TVA		**Total**
			HG	LG	HG	LG	
Slides	41	21	26	146	20	38	**292**
$\sigma = 7000$	59	74	98	411	93	132	**867**
$\sigma = 800$	545	950	454	3618	916	2186	**8699**
Total	604	1024	552	4029	1009	2318	**9536**
$A\,[cm^2]$	7.07	13.86	5.55	44.66	9.33	25.98	**106.44**

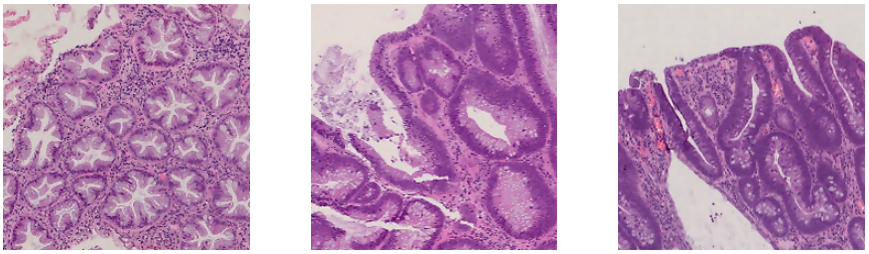
Table 4.1: *UniToPatho* class distribution for whole image slides (top) and the two patch scales made available (bottom).

UniToPatho [1, 2] is now public dataset of annotated high-resolution H&E-stained images, comprising multiple histological samples of colorectal polyps biopsies scans, collected from patients undergoing cancer screening. The dataset is a collection of the most relevant patch images extracted from 292 *whole-slide images* (or simply *slides* in the following), in accordance with UniTo pathologists' evaluation. The slides are acquired through a Hamamatsu Nanozoomer S210 scanner at 20× magnification (0.4415µm/px), as exemplified in Fig. 4.3. The following material and figures are taken from the original dataset proposal and our following research. [1, 5] Each slide belongs to a different patient and is annotated by expert UniTo pathologists, according to six classes as follows, as mentioned in Sec. 4.3: Normal tissue (**NORM**), Hyperplastic Polyp (**HP**), Tubular Adenoma with High-Grade dysplasia (**TA.HG**) and Low-Grade dysplasia (**TA.LG**), Tubulo-Villous Adenoma with High-Grade dysplasia (**TVA.HG**) and Low-Grade dysplasia (**TVA.LG**).

The *UniToPatho* dataset contains the slides splitted into a train set and a test set with a 70% to 30% ratio, by resulting in 204 slides used for training and 88 for testing. Therefore, each slide is represented either in the train set or in the test set, but not in both. Following the approach in [80, 42], we release square patches cropped from the 292 slides. Whole Slide images have large resolution and need to be cropped into patches. From each slide, we crop multiple non-overlapping square patches at different scales. We denote the side of the



(a) NORM (b) TA.LG (c) TA.HG



(d) HP (e) TVA.LG (f) TVA.HG

Figure 4.3: Example of 800×800µm patches for the six UniToPatho colorectal polyps classes.

underlying physical area with σ, measured in µm. The number of patches obtained for each slide hence depends on multiple factors including σ, the slide size and the polyp type. As common in similar datasets, the different classes are highly unbalanced in the dataset.

We make publicly available a total of 9536 patches, 8669 of which extracted at $\sigma = 800$ (1812×1812 pixels patches) and 867 at $\sigma = 7000$ (15 855×15 855 pixels patches). Tab. 4.1 provides a summary of the class distribution for the whole slides and the released patches. For each crop, the dataset maintains its slide index. This index is used to compose our subsets without intersection between patients. Each slide is associated with some metadata (stored in NanoZoomer Digital Pathology Annotations `.ndpa` file format), including a collection of Region of Interests (RoIs) associated with the corresponding class (see Sec. 4.4.2). Each RoI is determined by the pathologist and is defined by a free-hand contour, identifying the tissue area exhibiting histological findings. The number and the size of RoIs is highly variable and depends on both the tissue availability and the histological analysis. Such heterogeneity unfortu-

nately, leads to dataset unbalancing. Moreover, the collection and scanning of these slides cannot be homogeneous: As an example, the number of biopsy with Normal diagnosis is small compared to Tubular Adenoma cases, because biopsy is performed only on already suspicious areas. This difference is typical of the analysis process: if the presence of adenoma is not assumed, the biopsy is not prescribed. In our dataset, the distribution of the tissue data by classes is shown in Tab. 4.1. In the table the first rows represent the number of slides and the last one total tissue annotated area A for each class, respectively. We say we have R RoIs, which can be sliced into P patches. Therefore, annotated tissue area is $A_t = area(R_t)$, where t is the tissue class index. The *UniToPatho* patches P are cropped from R with a sliding window approach. A simple colour thresholding method discards the background patches.

4.4.1 Data collection

The data collection for the *Deephealth* use case colorectal polyps classification continued beyond publication of the dataset *UniToPatho*, by including new expert annotators for the biopsies samples. Tab. 4.2 shows the current data collection status. Moreover, it can be noted that the number of slides, reported in Tab. 4.2, denotes the frequency of appearance of the diseases, which is different for each one. The diagnostic hypothesis preceding the biopsy affects instead the amount of tissue taken from the patient, thus the amount of diagnostic tissue. Tab. 4.4 and Tab. 4.5 show different statistics in the Region of Interest analysis on the whole data collection. From these graphs it is easy to see how much the amount of relevant diagnostic tissue correlates with the type of colorectal polyp present in the biopsy. For example, slides containing tubular villous adenomous material (TVA) have much larger annotations than the other classes. This is due to the morphology of the pathology itself: the tubulo-villous adenoma presents macroscopic morphological characteristics, observable with a low magnification of the tissue, that differentiate it from the tubular adenoma.

	HP	NORM	TA		TVA		**Total**
			HG	LG	HG	LG	
Slides	108	30	126	534	129	104	**1031**
RoIs	269	112	414	1530	591	399	**3315**
$A\,[cm^2]$	18.74	18.38	58.29	154.59	265.65	109.63	**625.30**

Table 4.2: Class distribution for whole current image slides collection (top), the relative number annotated Region of Interest (RoI) (middle) and annotated underlying relevant tissue area.

In Fig. 4.4 is presented an overview of the diagnostic tissue that our pathologists annotate on the available slides. We can clearly see how the total data collection is unbalanced, due to the frequency of each patient disease. Fig. 4.5 shows the same data by looking the mean amplitude of the RoIs for each class. Both graphs show how different polyp types have relevant differences in available diagnostic tissue for a single slide.

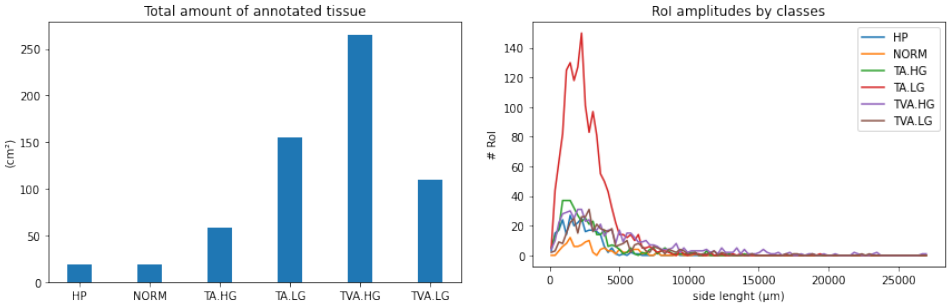


Figure 4.4: Visual report of unbalance in available diagnostic tissue. We take in consideration only the annotated tissue area as diagnostic.

4.4.2 Processing annotations

As we already mentioned in Sec. 4.4, the annotations are produced with the NanoZoomer Digital Pathology Annotations software and a single metadata file in `.ndpa` file format is produced for each slide. There is very poor documentation on

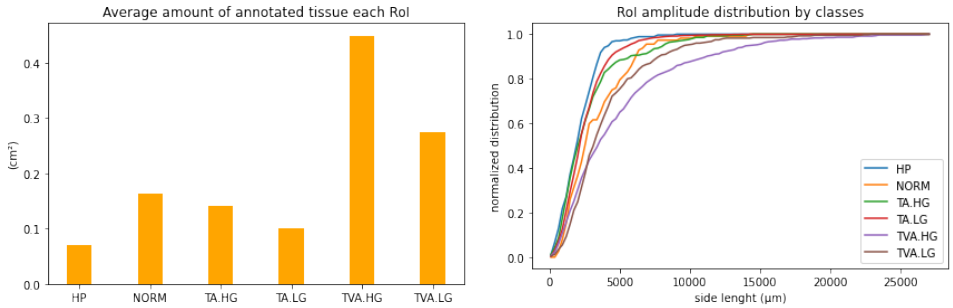


Figure 4.5: Diagnostic tissue distributions for each class. As in the Fig. 4.4, we consider the diagnostic tissue as the RoIs area.

how to handle this proprietary format without the proprietary Hamamatzu viewer software, so we want to suggest here some simple guidelines.

Basically, the `.ndpa` file can be read and processed as `.xml` file. It presents a list of `<annotations>` tags, each one describe a different annotation (called RoI in our case). Each `<annotation>` tag can have different `type` attribute, like `freehand,rectangle,circle`, and provides the coordinates of each perimeter point. Let A be an annotation and S the reference slide scan, you have to access to both to get the correct tissue slice:

- manage the coordinates format: A coordinates are expressed in nanometers (nm), while S express coordinates in pixels (px) and provide the acquisition resolution in µm/px. In order to read the slide patches correctly, this conversion needs to be handled.

- manage coordinate reference: A points coordinates can be found with positive or negative values: A reference is the center of S. Instead, the S pixel coordinate reference is in top-left corner. The A reference can be found by applying the coordinate transformations from S dimensions or, if they are available, by using S `XOffset` and `YOffset` properties (expressed in µm).

In Fig. 4.6 an example of multiples RoIs, in a single slide, is shown.

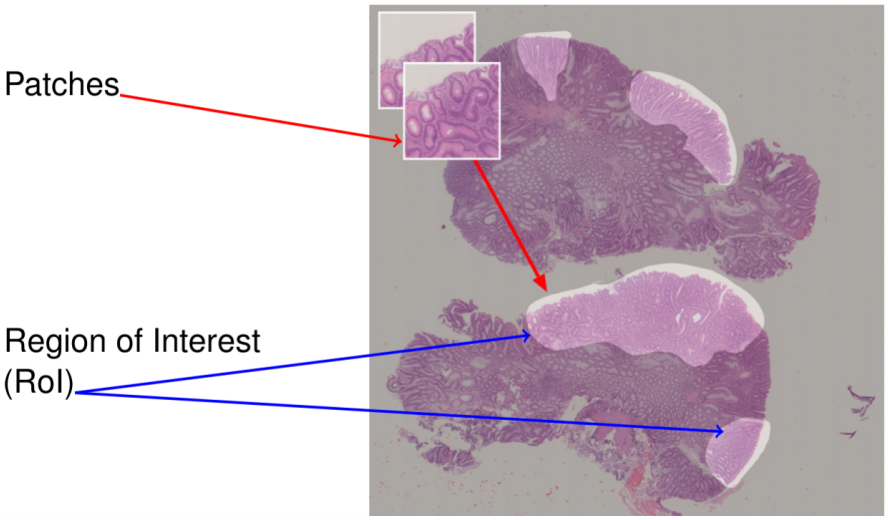


Figure 4.6: Example of annotated regions and patch extraction

Chapter 5

Automatic histopathologic diagnosis

In this chapter we are going to see how to take care of the healthcare task related to histopathological diagnosis presented in the previous chapter. First, we present an in-depth study on how the histopathologic tissues collected within *DeepHealth* are suitable to solve this task, then some consideration on how to take in account data unbalance in tissue image preprocessing. The descriptions, considerations and the images reported below are taken from our related work [5].

5.1 Study method pipeline

The usage of deep learning for classification already proved, in similar learning tasks, to be extremely effective and robust [47, 92]. How we have already seen in Sec. 4.4 direct classification on the (high resolution) whole slide, in our context, is unfeasible: the relevant features are local and can be detected at very low image scale. For this reason, the deep learning model is not trained on the full slides, but on some crops we refer to as *patches* (Sec. 4.4.2). High-level representation of our approach is reported in Fig. 5.1. The tissue outside the RoIs contains less

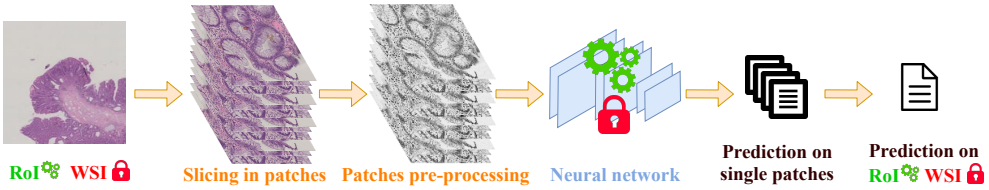


Figure 5.1: The neural network is trained on RoI images (gears symbol) and tested on WSI (lock symbol).

salient features for the overall classification task (or even different type tissue of organic tissue) and should be ignored. Hence, in our training pipeline, we use patches extracted from RoIs only. In our work [5], once the model is trained on patches' classification, we use the average score from all patches to infer a prediction for the starting whole slide in the test set.

The first operation we perform on RoIs (even before slicing them into patches) is re-scaling them, in order to fit the chosen model input resolution to some target physical resolution σ, by using the Lancos-3 filter (as it produce slightly sharper results than bicubic method) [64]. We chose to use a input image resolution compatible with current CNN models, typically working on images with hundreds of pixels per row/column.

In this study, we choose to slice the RoIs/WSIs tissue area A_t into 224×224 pixels large patches with some overlapping area $\hat{A}_t$, where t identifies the class. The inclusion of $\hat{A}_t$ in the patches extraction process wants to compensate the imbalance in $A_t \ \forall \ t$ in less represented classes [93]. This is also a type of regularization by data manipulation, as we seen in Sec. 3.1. If we consider P as the total set of tissue patches, we want all $|P_t|$ to have at least the same order of magnitude, by generating overlapping images. We address this by changing the sliding window stride for images belonging to less represented classes. By reducing the stride, we add tissue redundancy between patches, or $\hat{A}_t$.

$$\text{stride(t)} = 1 - 1/\max(1, (\hat{a}\hat{P}_t/A_t)^{\beta}) \tag{5.1}$$

where t identifies the class, $\hat{P}_t$ is the expected number of patches for class t and $\hat{a}$ is the single patch area. Since $\hat{P}_t$ represents the expected $|P_t|$, that is set to the average patch

number from most represented classes. β hyperparameter regulates the stride lower-bound. During training we augment data: we include vertical/horizontal flips and a random operation chosen between rotation, equalization, solarization, inversion and contrast enhancing, as proposed in [20]. The solarization function randomly transforms the image by inverting all pixel values above a given threshold. Data augmentation is crucial for training a neural network in this context. The variability of the dye, the quality of the scans and the shared features from tissues of the same patient, are obstacles to a good solution able to generalize well. Without data augmentation there is an increased risk of producing overfitting neural models.

In order to perform classification on the patches, we have used ResNet-18: it represents a good trade-off between complexity and performance and is one of the broadly-used to solve similar tasks [47, 92]. ResNet's architecture high performances come from its block compositions, called *residual blocks* (Fig. 5.2). It is a block of convolutional layers that is learned via back propagation as part of a larger network. The final layer of the residual block is called the merge layer. It uses the learned parameters to merge the outputs of the convolutional layers. It is also interesting that ResNet's initial layers are also designed to be "deep", meaning that they have a large depth. This is the opposite of the common strategy of using shallow architectures to train deep neural networks. Pre-trained deep neural networks (on the ImageNet classification task) can be effectively used as initialization for medical classification tasks, showing good performance [47]. [1]

5.1.1 Patches normalization studies

We perform a study at different RoI phisical resolutions: the goal here is to identify the best scale the deep model is able to extract the useful features to predict correctly the slide label from a single tissue patch.

[1] Pre-trained model used in all the experiments is available on *PyTorch* website.

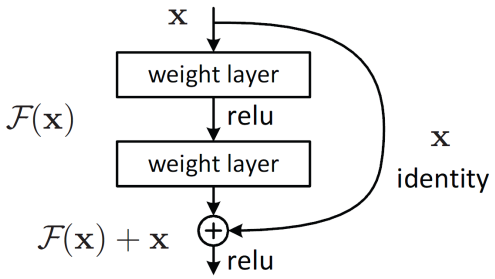


Figure 5.2: Scheme for a ResNet's *residual block*. It is an example of *shortcut* or skipping connections. Shortcut connections are those skipping one or more layers, by performing identity mapping, and their outputs are added to the outputs of the stacked layers. Their main advantage is to forward an high signal gradient to the previous layers in back propagation during the training phase. The image is taken from He *et al.* [35]

A common step for microscopy image analysis is color normalization. This process reduces the differences in tissue samples due to variation in staining and scanning conditions. Towards this end, we consider 8 possible patches resolutions $\sigma \in [300; 1000]$ µm, and 3 possible input preprocessing strategies: use of the original patches (RGB), conversion to gray-scale (gray) and the use of a standard slide normalization strategy (Macenko *et al.* [55]), resulting in 24 training possibilities, which are reported in Fig. 5.3. For our classification task, we

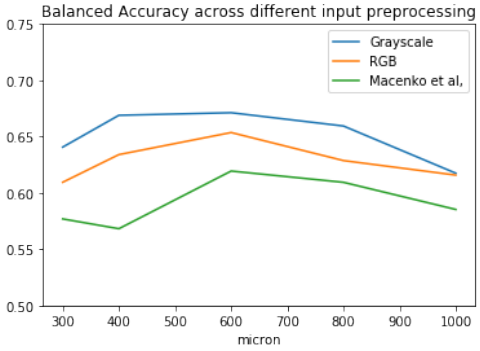


Figure 5.3: Classification performance at patches level. The source dataset was a slight different slide collection from *UniToPatho*. The dataset is divided into patches by following the same criterion for different physical resolutions.

found that the use of gray-scale images does not remove useful information (which might be embedded in the color) and, on the contrary, helps in removing the expected color bias [56, 69]. From our results we learn that, for the particular classification task we aim at solving, the the signal strength is one of the most relevant features in the image texture and, while the direct use of the RGB image does not compensate the color bias, or even standard slide normalization strategies, like [55], destroy some useful information which is not embedded in the color feature. For these reasons, we focus the analysis presented in [5] by using gray-scale patch images as input.

5.1.2 Preliminary patches resolution and WSI classification studies

Here we will inspect more in depth the study on WSI classification performance using gray-scaled 224×224 pixel input. In this study, for the WSI inference, we use a very simple heuristic: we perform a majority vote from patches prediction for a given slide. Fig. 5.4 provides a general overview of some metrics evaluated, where single patches at different resolution σ (ranging in $[300; 1000]$ µm), by take in account the dysplasia grade discrimination as a priority. There is not a clear choice regarding the optimal scale features have to be extracted. If our goal is to maximize the sensitivity for the HG class, we should choose 400 µm: inspecting the HP's specificity for the same scale, we observe a drop which, however, is overall tolerable. F1-score gives us a more global information: indeed, for the HG class, 400 µm is the best one. However, if we look at average performance on all classes (avg), focusing on F1-score and balanced accuracy, we can observe similar performance for 400 µm and 600-800 µm.

It is important to compare the model performance with the results obtained by human pathologists. Table 5.1 reports performance comparison for HP, LG and HG classes in terms of balanced accuracy, sensitivity and specificity. Here, human

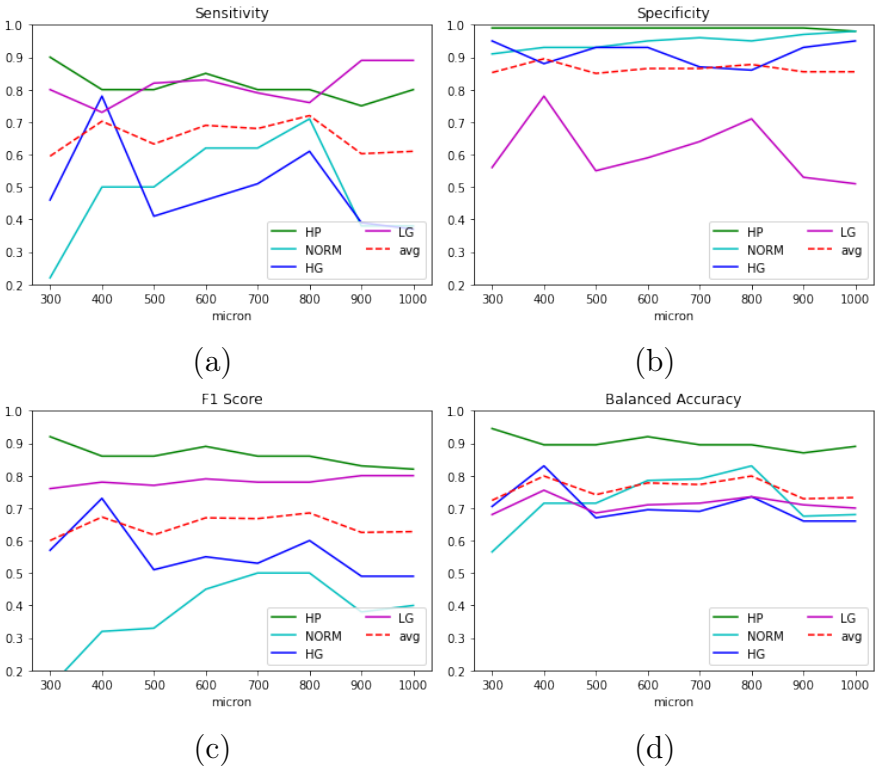


Figure 5.4: WSI inference performance comparison between different tissue categories at different patches resolutions: sensitivity (a), specificity (b), F1-score (c) and balanced accuracy *BA* (d). Red dashed line is the average performance (avg).

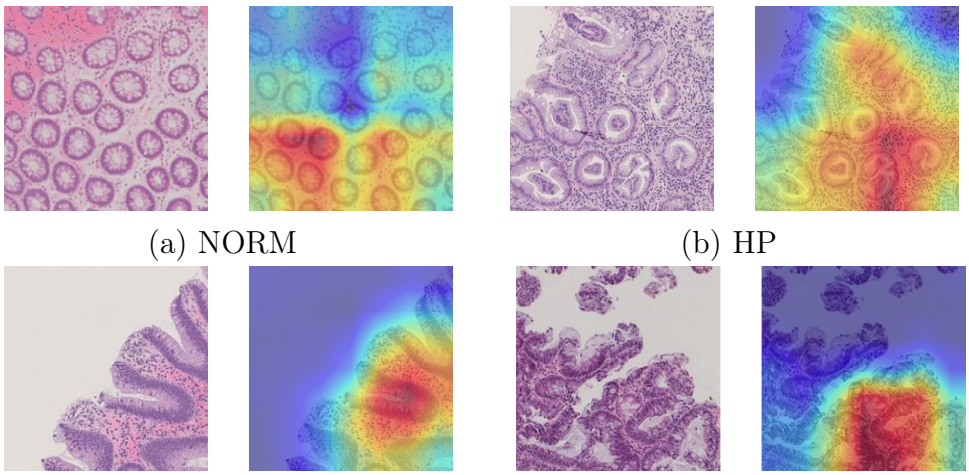
pathologist's average agreement is taken from Denis *et al.*'s work [23], evaluated on qualitatively similar data. The dataset used by Denis *et al.* is not publicly available, so the comparison has to be intended as no more than a guideline. As we observe, the performances in this study is very close to the pathologists'. In particular, HP classification increases of more than 10% in accuracy, showing a quite significant improvement in terms of sensitivity. LG classification improves as well up to 10% in balanced accuracy, yielding a significant improvement both in terms of sensitivity and specificity. HG classification score is in the same order than human pathologists (this finding is likely to be due to HG features that are known to be visually easier to detect).

Fig. 5.5 shows some example of the ANN focus on different tissue inputs. Our pathologists agree that the highlighted fea-

Table 5.1: Human hyperplastic and dysplasia diagnostic performance comparison. The original dataset is unavailable for accurate comparison, but the samples are qualitatively comparable between datasets. We report the Balanced Accuracy as BA.

		BA	Sensitivity	Specificity
HP	Our (400 µm)	0.90	0.80	**0.99**
	Our (600 µm)	**0.92**	**0.85**	**0.99**
	Pathologist [23]	0.79	0.30	0.97
LG	Our (400 µm)	**0.76**	0.73	**0.78**
	Our (600 µm)	0.71	**0.83**	0.59
	Pathologist [23]	0.66	0.57	0.69
HG	Our (400 µm)	**0.83**	0.78	0.88
	Our (600 µm)	0.70	0.46	**0.93**
	Pathologist [23]	**0.83**	**0.81**	0.84

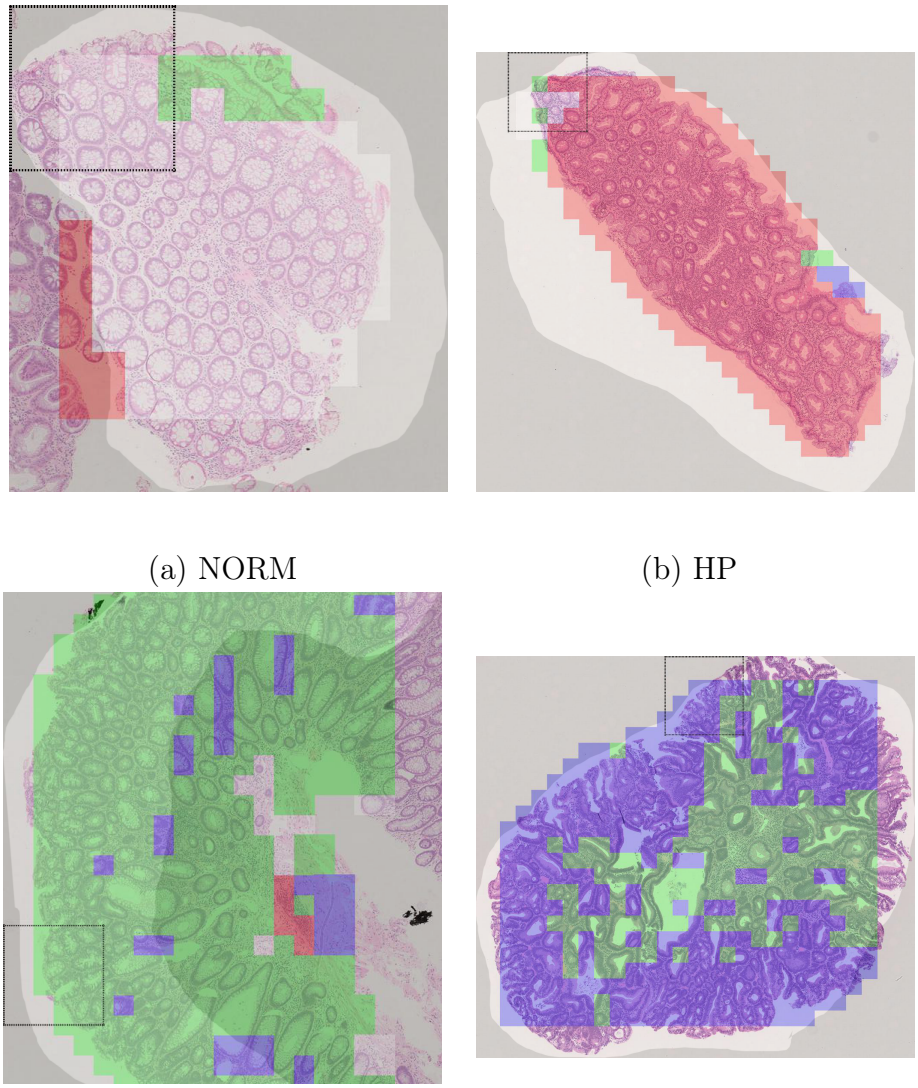
tures are significant for the type of tissue. Fig. 5.6 report some example of RoIs class inference from the patches prediction.



(a) NORM (b) HP

(c) LG (d) HG

Figure 5.5: Regions where the trained neural network model focuses on 600µm patches. Above images are obtained with Grad-CAM algorithm [74]. For example, the hot spot of the HP sample is on a serrated gland which is a characteristic features. Instead, in the HG sample, the focus is correctly placed on the pre-neoplastic lesion.



(a) NORM (b) HP

(c) LG (d) HG

Figure 5.6: Visualizing patch classification: Each box is located at the center of the corresponding patch with a color representing the predicted class: HP (red), NORM (white), LG (green), HG (blue). The black dashed square visually represents the patch scale, (in this case $\sigma = 600\mu m$). Lighter regions are the annotated area of interest.

5.2 Patch classification on *UniToPatho*

By increasing the number of slides at our disposal, we were able to investigate classification based on the patches approach in two different directions.

- approaching classification of full resolution images: given the µm/pixel ratio with which we know a slide is produced, patches are generated with the corresponding size, in pixels, with respect to the chosen σ measure.

- approaching classification by exploring patches with much higher σ sizes. The number of collected annotations, from which *UniToPatho* is derived, is sufficient to adopt *"data-greedy"* deep learning algorithms. Given the large size, the minimum number of images will be at least the number of annotations.

Below the results reported by these analyses and the related images are taken from our proposed work [1].

We perform our preliminary experiments on the *UniToPatho* dataset: this time, in order to expand Perlo *et. al* [5] that focus on dysplasia grade prediction, we focus first on the colorectal polyp type analysis. Thanks to the data collection for *DeepHealth* use-case advancement, we can investigate more methods and strategies in order to achieve better prediction results. The method which will be described in the next section builds upon the lessons learned during these experiments.

In Fig. 5.7 we report the same study performed previously, by making two important additions to the analysis:

- first we explore way wider resolution. In order to maintain a consistent overall number of patches even with high-resolution clippings, we set a lower-bound on the number of patches P:

$$|P(s)| \leq |RoI(s)| \forall s \in S \tag{5.2}$$

where S is the whole slide collection. We are now going extend the explored range form $\sigma \in [300, 1000]$µm range to $\sigma \in [100, 8000]$µm range. We repeat the whole training process for each one of the chosen σ. For each σ, the trained network is eventually tested on the patches extracted at the same scale from the test slides.

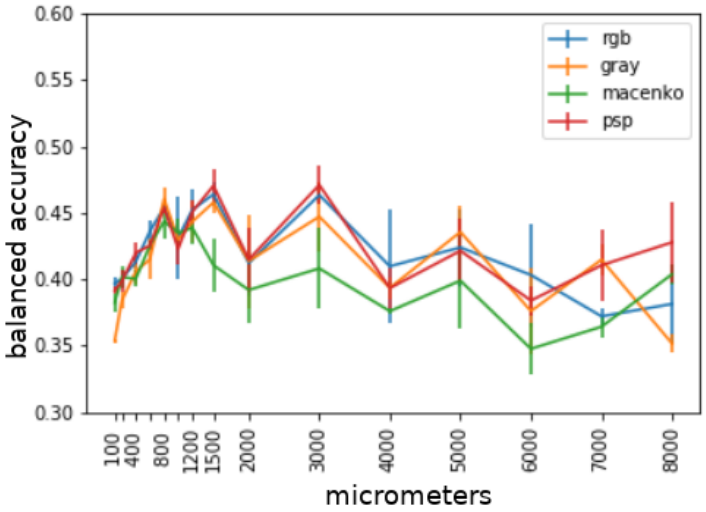
- we include our *PSP* regularization method, that is described in Sec. 3.2.1. How we described before, we want to investigate on *PSP* capabilities in a real-world task.

In order to obtain the results, we randomly color-jitter the patches described in the previous Sec. 4.4 to augment the dataset, as proposed in [92, 47]. As in [47, 92, 5], we train a deep convolutional neural network for image classification on the patches belonging to the train slides. Namely, we train an ImageNet-pretrained ResNet-18 [35] with SGD optimizer for 50 epochs, with an initial learning rate of 0.01 decayed by a factor of 0.1 every 20 epochs. Furthermore, we decide to change the dataset balancing policy. By strictly working with *UnitoPatho* collection, more experimental trainings show how including the overlap between patches (as described in Sec. 5.1) can lead to overfit models (Sec. 3.1). Our conclusion is that the model learns to recognise the shared tissue patterns in the images, rather than finding significant tissue features. Therefore, in the following experiments, we change the dataset balance policy by duplicating or by removing tissue images from the dataset pool, in order to have the same amount of images for each tissue class.

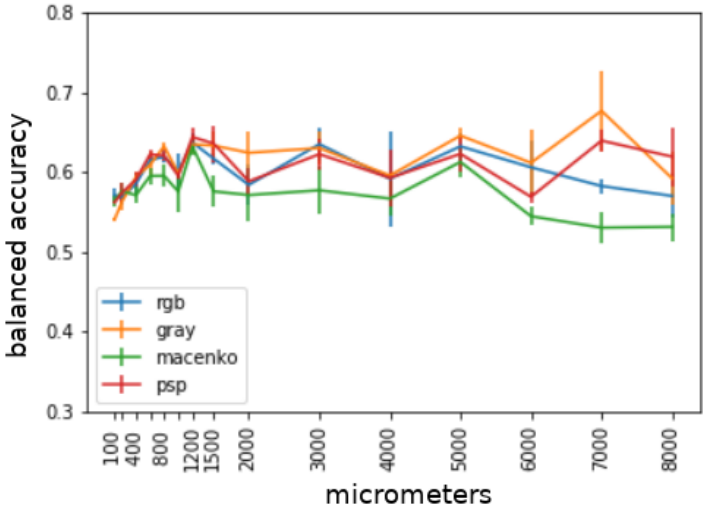
The output layer is reshaped to match the UniToPatho 6-class classification problem. Each patch is downsampled to the standard 224×224 pixels ImageNet size prior being fed into the ResNet.

In the following, the classification accuracy will be defined in terms of Balanced Accuracy (BA) to cope with the unbalanced samples in the dataset. By looking Fig. 5.7, we can get different conclusions:

- There isn't an absolute physical resolution that helps us



(a)



(b)

Figure 5.7: Classification performance at patches level. The source dataset was *UniToPatho*. The dataset is divided into patches by following the same criterion for different physical resolutions. (a) shows the BA for the 6 class classification problem; (b) shows the BA for the polyps type prediction only (HP,NORM,TA,TVA).

to achieve the best patch classification performance. On the other hand, we can see how overall performances, on the 6 classes Fig. 5.7a, start to decrease by going forward with bigger σ.

- *PSP* regularization gives the best results in most of the σ we take in consideration. This experiment shows its applicability to our custom dataset.

- Also for the 4 classes polyps classification problem, Fig. 5.7b doesn't show a clear indication on witch σ fits our performance needs. In this graph only, we can see how the gray scale preprocessing method seems to achieve generally better results.

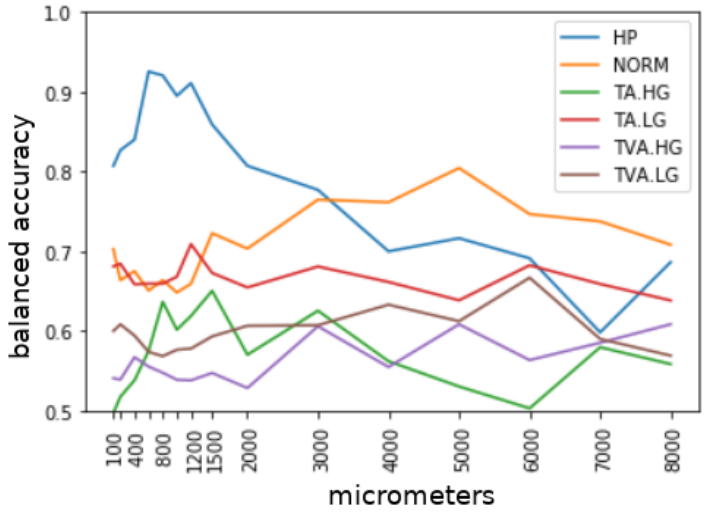
We need to delve further into the analysis, though, because using only global balance accuracy, we may have missed the behavior of the trained network with respect to the individual polyp's classes in the problem. Nonetheless, we conjecture that different polyp types can be better recognized at different scales: as a consequence, in Tab. 5.2 we also show the BA concerning the classification of each single polyp type HP, TA, TVA plus NORM. The TA and TVA classes encompass the respective low grade (LG) and high grade (HG) subclasses. In Tab. 5.2 (first row) we show the BA achieved when attempting to discriminate all 6 classes with the baseline approach as function of the patch scale σ: it can be noted that even the best accuracy achieved at σ=1500 is quite low.

Fig. 5.8 help us to visualize such behavior:

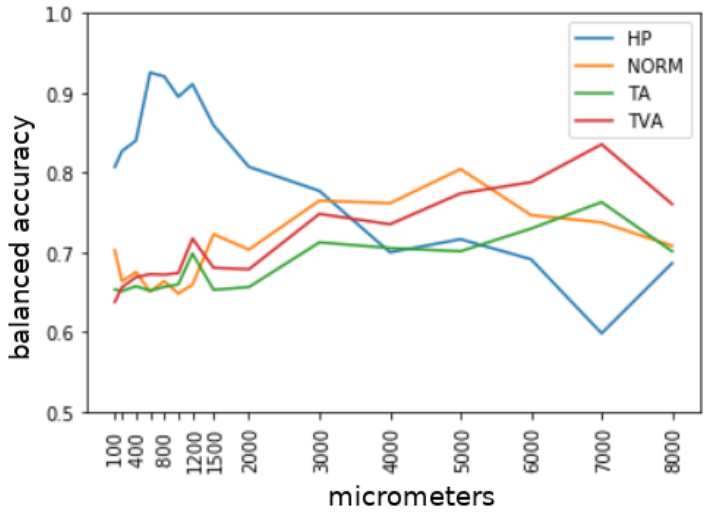
- Breaking down the accuracy on a per-class basis reveals that different types of polyps achieve top-classification accuracy at different scales.

- $\sigma \in [800, 1200]$µm scale is the best resolution in order to catch Hyperplastic Polyps (HP) tissues. We hypothesize these types of benign polyps are best discriminated by looking at smaller-scale details such as gland edges [85].

- Conversely, Tubular Adenomas (TA) and Tubulo-Villous Adenomas (TVA) are best classified at a coarser 7000µm scale: we hypothesize this type of polyps is best discriminated by looking at large-scale macro structures such as entire glands shapes [89].



(a)



(b)

Figure 5.8: Focus Per class, on classification performance at different patches level. (a) shows the BA achieved on 6 classes by using raw *RGB* images; (b) shows the BA for the polyps type prediction only.

- *RGB* image prediction performances drop at 7000µm scale only involves HP class mainly that is misinterpret as Normal tissue. Grayscale images model seems someway overcoming this issue, but, talking about *not-*

	Patch scale σ [µm]					
Type	100	800	1500	4000	7000	8000
BA (6-class)	0.40	0.45	**0.46**	0.41	0.37	0.38
NORM	0.70	0.66	0.72	0.76	0.78	0.71
HP	0.81	**0.92**	0.85	0.70	0.60	0.69
TA (HG+LG)	0.65	0.66	0.65	0.71	**0.76**	0.70
TVA (HG+LG)	0.64	0.67	0.68	0.74	**0.84**	0.76

Table 5.2: Overall BA for all of the six classes (first row) and BA for each polyp type, plus normal tissue.

adenoma tissues (NORM and HP), the two downsampled input types have very similar tissue structures.

Coming to the problem of predicting the grade (LG or HG) of TA and TVA adenomas, we investigate whether that could be best predicted at some particular scale σ. From our experiments, the adenoma grade classification accuracy does not appear being a function of the patch scale. In fact, visual inspection of the 224×224 pixels downsampled patches reveals that the downsampling trashes discriminatory details in the cells nuclei upon which pathologists are known to rely.

Concluding, our preliminary experiments show that *i)* different polyp types are best classified at different scales (in our case $\sigma = 800$ for HP and $\sigma = 7000$ for TA and TVA) and *ii)* adenoma grade prediction may be jeopardized if the details in cells nuclei are lost. The general improvement in accuracy compared to the 6-class classification across all of the patch sizes also hints that the grade prediction might be negatively influenced by the image downsampling.

5.2.1 Inference by ensembling classifiers

This section details our proposed approach towards classification of *UniToPatho* images: a multi-resolution ensemble of cascaded classifiers. The experiments in Sec. 5.2 showed that: *i)* 800µm patches should be used for HP detection; *ii)* 7000µm

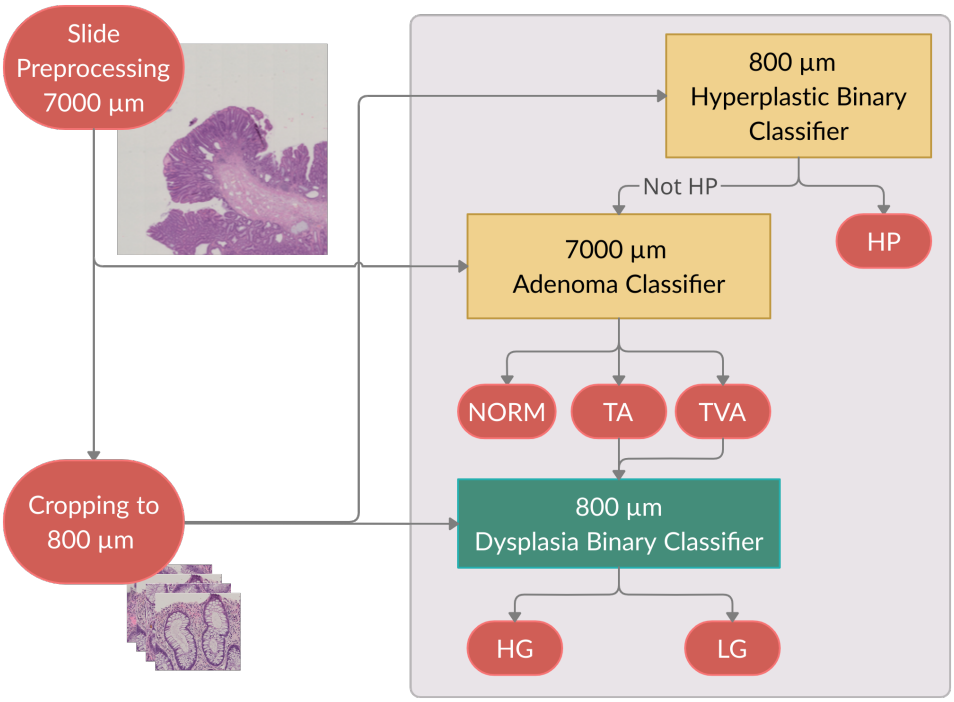


Figure 5.9: The proposed multi-resolution ensemble of cascaded classifiers. The Hyperplastic and Adenoma classifiers (yellow) are trained with inputs downsampled to 224×224 pixels, while the Dysplasia classifier (green) uses full resolution images.

patches for adenomas detection and classification; *iii)* grade prediction might be negatively influenced by the input image downsampling. The proposed main idea is to search for specific features detected at different σ, by develop a multi-resolution approach. The method presented in our work Barbano *et. al* [1] relies on three cascaded ResNet-18 classifiers, having the output layer specifically adapted for each classification task. The classifiers are trained on patches extracted at either $\sigma = 800$ or $\sigma = 7000$.

The overall process of inference is depicted in Fig. 5.9, with the input being a single 7000×7000µm patch which is used by the three ResNet-18 classifiers mentioned above. For classifiers working at $\sigma = 800$, we crop the input image into smaller 800×800µm sub-patches which are used to generate a prediction for the entire image. The ensemble works with 7000µm

input patches, as it is the largest chosen size among HP, TA and TVA. The ordering of the different classifiers in the ensemble follows the accuracy shown in Tab. 5.2: HP detection, followed by TA and TVA classification. Obviously, grade prediction must be posterior to TA and TVA classification, as it can only be applied on adenomatous tissue.

HP polyps detection

Starting from a full-resolution 7000µm input image, we first apply a binary HP classifier on sub-patches of size 800µm extracted from the input image. First, HP polyps are discriminated from adenomas and normal images via a binary classifier that takes as input sub-patches of size 800×800µm extracted from the larger input image. In fact, Tab. 5.2 shows that HP polyps are identified with top accuracy (0.92 BA) at scale $\sigma = 800$.

To infer the probability of predicting HP for entire image, we compute the average predicted probability of the sub-patches of being HP. In the case the patch content is not classified as an HP polyp, the second classifier in the cascade is invoked.

Adenoma detection

Second, TA adenomas are discriminated from TVA adenomas and from normal images via a ternary classifier taking as input the entire 7000×7000µm patch. In the case the patch content is classified as a TA or TVA polyp, the third and last classifier is invoked to infer its grade. Tab. 5.2 shows that TA and TVA adenomas are identified with top accuracy at this scale. To accomplish this, we define a "super" class *OTHER* by grouping NORM and HP. Therefore, e train a classifier on the classes OTHER, TA and TVA. This approach can easily be extended to include other tissue types by adding samples to the OTHER class, without changing the network architecture. Given that, in this case, the classifier is applied to negative HP-predicted samples only, images which are not classified as either TA or

TVA are labeled as normal (NORM). Thus, employing the super class OTHER, also easily allows for different ensemble hierarchies.

Dysplasia grade classification

Finally, a binary classifier is used to determine the dysplasia grade for TA and TVA adenomas. Sec. 5.2 suggested that downsampling the patches to 224×224 pixels might be detrimental towards inferring the dysplasia grade due to the loss of important features such as cells nuclei, as also observed by [80]. Thus, only for this specific classification task, we skip downsampling to prevent the loss of fine grained details. The dysplasia classifier input are full resolution tissue patches, so the patch edge measures $800 \times 0.4415\mu m/pixel = 1812$ pixels. To account for the increased size of the feature space, we add an adaptive average pooling layer [36, 95] before the fully connected layer of the ResNet-18 network. We repeat the experiment of Sec. 5.2 sweeping the patch scale σ this time without downsampling and we find that adenoma grade classification peaks (0.81 BA) for $\sigma = 800$. As a consequence, we apply the grade classifier on sub-patches extracted from the input image at the scale $\sigma = 800$. Finally, we infer the dysplasia grade for the entire input image according to a threshold T_d: if the ratio of high grade predictions on the sub-patches is higher than T_d, the image is labeled as HG, otherwise as LG. This choice is motivated by pathologists grading guidelines, where HG can be decided on the base of small portions of tissue; clearly, setting smaller values for T_d can mimic this behaviour [88].

5.2.2 Evaluation by multi-resolution ensemble

We evaluated a multi-resolution classifier ensemble on the 7000×7000µm test patches of *UniToPatho* on top of the previous analysis. By analyzing different T_d threshold effects on the ROC curve displayed in Fig. 5.10, we tune the classifier responsible for the dysplasia grade prediction. As expected,

the T_d allows one to reach a good compromise between true and false positive rates yielding a good AUC score of 0.83. In

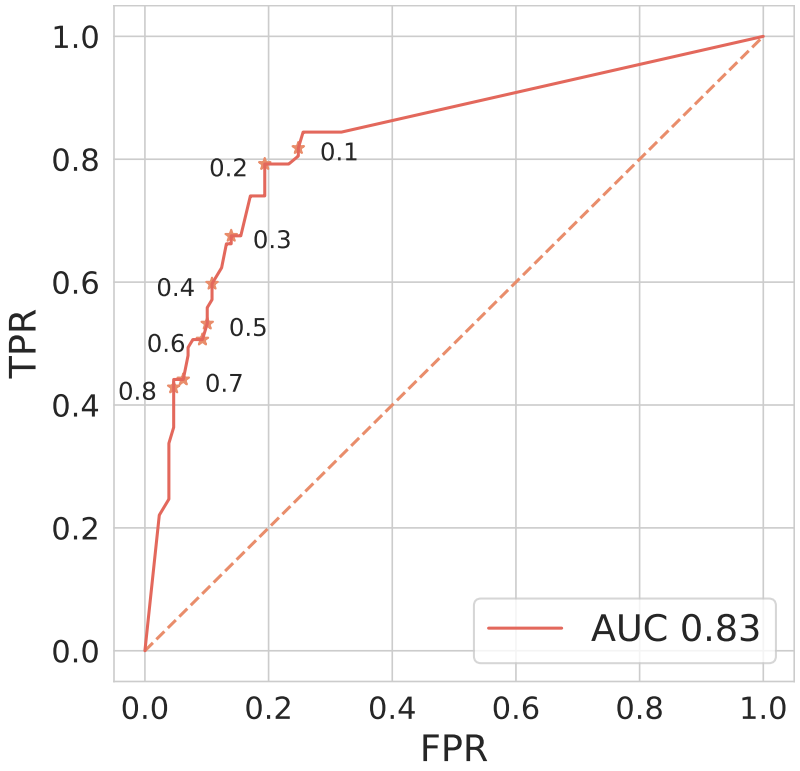
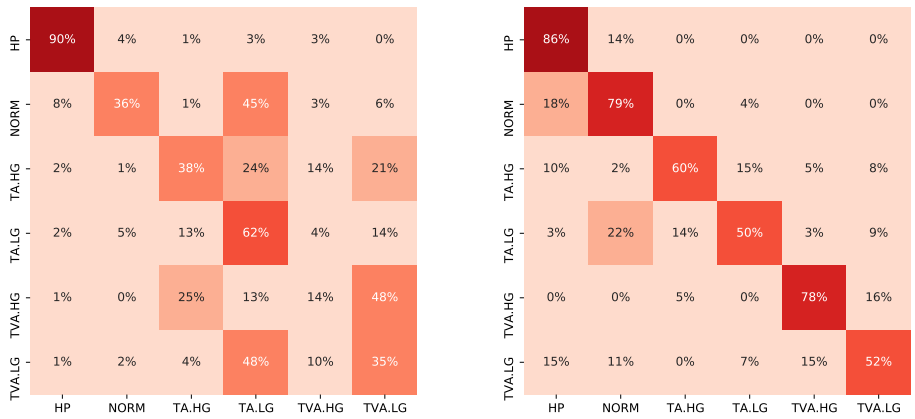


Figure 5.10: ROC curve for the adenoma grade predictor as a function of the threshold T_d.

the experiments reported below, we choose to use $T_d = 0.2$, as it strikes a favorable balance between false positives and sensitivity score. In which cases where minimizing the false negatives rate is more important, for example in clinical applications, lower thresholds may be preferred.

Fig. 5.11 shows the 6-class confusion matrix of the proposed method compared to the baseline approach described in Sec. 5.2. Looking at the diagonal, it is quite evident that the proposed method significantly improves in average accuracy, that leaps from 0.46 to 0.67 (50% relative increase). The baseline reference is taken by using $\sigma = 1500$ which, as already shown in 5.7, yields the best overall BA.

The multi-resolution approach shows remarkable improvements in assessing the correct grade. On the other hand, we noticed previously how the baseline model is biased towards the



(a) Baseline (b) Multi-resolution Ensemble

Figure 5.11: Confusion matrices for the (a) baseline at $\sigma = 1500$ (b) multi-resolution ensemble.

lower grade classes: this represents further proof that down-sampling the images results in the lack of useful features to distinguish high grade from lower grade and normal tissue.

Tab. 5.3 reports other metrics that are common in the related literature. The multi-resolution approach achieves quite high specificity for all classes, and we can claim, especially for the higher-risk TA.HG and TVA.HG classes, promising sensitivity values.

Finally, we also analyze the per-type performance, as done in Sec. 5.2. The results are shown Tab. 5.4. Notably, there is a great reduction in false positive adenoma predictions, and, most importantly, a more precise distinction between Tubular and Tubulo-Villous adenomas. Despite the small HP class accuracy drop we observe an increase for all of the other tissue classes.

	HP	NORM	TA		TVA	
			HG	LG	HG	LG
Sensitivity	0.86	0.79	0.60	0.50	0.78	0.52
Specificity	0.93	0.87	0.92	0.94	0.96	0.92
Positive Discovery Rate	0.43	0.44	0.85	0.60	0.76	0.44
Negative Predictive Value	0.99	0.97	0.73	0.92	0.96	0.94
False Positive Rate	0.07	0.13	0.06	0.08	0.04	0.08
False Negative Rate	0.14	0.21	0.50	0.40	0.22	0.48
False Discovery Rate	0.57	0.56	0.15	0.40	0.24	0.56
Accuracy	0.93	0.86	0.87	0.76	0.93	0.88
Balanced Accuracy	0.89	0.83	0.76	0.72	0.87	0.72
F1 Score	0.57	0.56	0.63	0.60	0.77	0.47

Table 5.3: Sensitivity, Specificity, BA and other common metrics per class: Balanced Accuracy, Sensitivity and Specificity levels are qualitatively comparable to the pathologists agreement [23]. Despite medium sensitivity values for LG adenomas, specificity values are stable for each class.

	σ	HP	NORM	TA	TVA
Baseline	800	**0.92**	0.66	0.66	0.67
Baseline	1500	0.85	0.72	0.65	0.68
Baseline	7000	0.60	0.78	0.76	0.84
Multi-resolution	-	0.89	**0.83**	**0.81**	**0.87**

Table 5.4: Comparison of the class BA between the baseline and the proposed multi-resolution approach.

Chapter 6

Perfusion brain maps prediction

In this chapter we present another case study, which is completely different from what we've seen above. In this study the subject of analysis changes and the focus is on the analysis of images of the brain. Specifically, we describe our experimental contribution by collecting and analyzing Computed Tomography (CT) scans of the brain, with the goal of identifying cerebral vessel occlusions in ischemic stroke cases. This study has been included as a use-case in *DeepHealth* project, and the collected images are composing a new dataset, named as *UniToBrain*[3], within the perimeter of the project.

6.1 Overview

The occlusion of a cerebral vessel causes a sudden decrease in blood flow in the surrounding vascular territory, in comparison to its centre. The identification of such an occlusion reliably, quickly and accurately is crucial in many emergency scenarios like ischemic strokes [26]. The CT perfusion (CTP) is a medical exam for measuring the passage of a bolus of contrast solution through the brain on a pixel-by-pixel basis. CT perfusion is performed with a sampling time of approximately 1 Hz. Based on these measurements, low-dose serial scans are acquired and blood time-density curves, corresponding to the passage of the

contrast agent into the brain tissue, and several other parametric maps are calculated. The most relevant parameters used in clinical practice are maps representing cerebral blood volume, cerebral blood flow and time to peak (called CBF, CBV, TTP respectively) [7]. In particular, ischemic lesions develop very rapidly, originating from the central area of the occluded vascular region and progressively expanding to increasingly peripheral regions. From the onset of symptoms, two different regions can be identified in the problematic ischemic area of the brain: a central *"core"*, corresponding to the area of irreversible damage, and a peripheral *"penumbra"*, corresponding to the area of potential recovery and possible recanalization, provided by the occluded vessel, are visible. Generally, CBV maps are used for core segmentation, while CBF and TTP for penumbra areas. Therefore, the identification of the core and of the penumbra can predict the fate of the brain tissue itself and guide physicians in reperfusion treatments [26, 91]. An example of colored perfusion maps and core-penumbra region is shown in Fig 6.1. The extent of the core and its penumbra area can be estimated clinically based on symptoms and their time of onset, and using common perfusion techniques such as CTP. A time-intensity curve is the result of processing the signal intensity values over time. Several algorithms are used to perform deconvolution of time-intensity curves, but some of them are not public and may produce maps with divergences [50]. If we place ourselves in an ideal environment, where we find limited noise, limited variance and no artifacts created by patient motion during scanning, the optimal choice for obtaining realistic, accessible and reproducible maps is an analysis performed pixel by pixel, such as the one performed by deconvolution-based algorithms. Unfortunately, in the real clinical case, the information must be redundant in order to overcome the problems caused by the presence of noise, large variances and motion artifacts. In practice, this results in obtaining more brain slices, requiring more patient X-ray exposition, more acquisitions, estimation of an Arterial Input Function (AIF), and a series of spatial pre-processing steps for noise and variance reduction. The following method

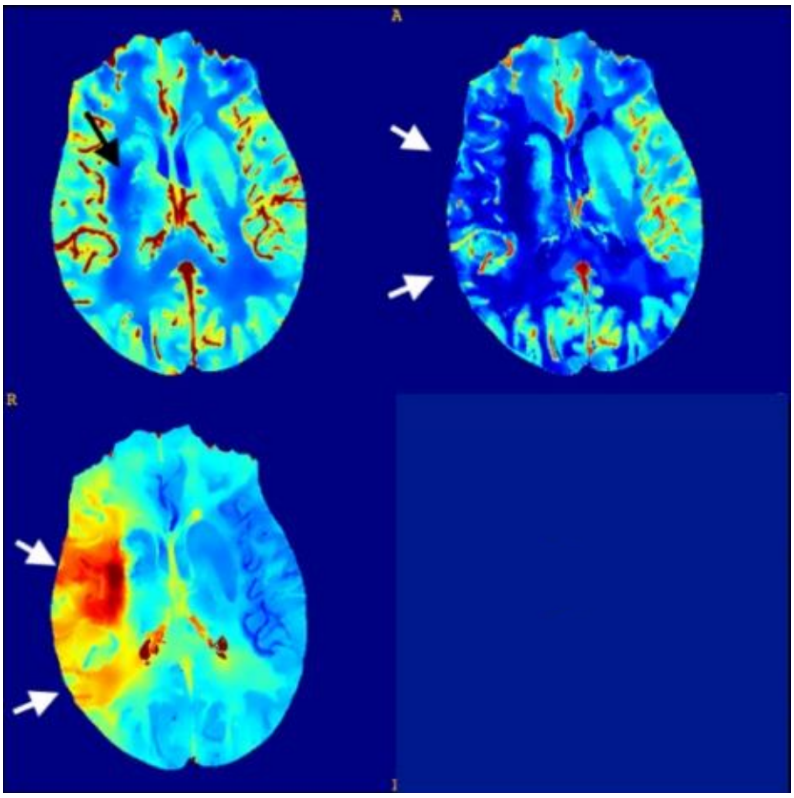


Figure 6.1: Examples of perfusion maps, in which the core is highlighted by the black arrow in the CBV map (top-right image) and the penumbra is is indicated by white arrows in CBF and TTP maps. The image is taken from Gava *et al.* [30]

and more detailed results are presented in Gava *et al.* [30].

6.2 Data collection

The Neuroradiology Division in Molinette Hospital, Turin (Italy), collected perfusion data from 127 patients, retrieving them from the hospital's PACS system retrospectively. The CTP acquisitions were performed using a GE 64 Scanner, and the parameters for the examination were defined as follows, the same for each case: 80 kV, 150 mAs, 44.5 sec duration, 89 volumes (40 mm axial coverage), injection of iodinated contrast agent for 40 ml (300 mg/ml) at 4 ml/s speed. Perfusion maps, including CBF, CBV, TTP, were calculated using a standard spatial pre-processing pipeline followed by a

nonlinear regression (NLR) method based on a state-of-the-art fast model developed and described by Bennink *et al.* [10]. A motion correction is required and was performed using a rigid registration method. Next, all images were pre-processed by implementing a bilateral filter [46]. Estimates of the AIF and vein output function (VOF) were made automatically using a sample of 100 voxels. The impulse response function (IRF) of perfused tissue is described by the model developed by Bennink *et al.*, in terms of CBV and tracer delay, which is fundamental in the clinical context of ischemic stroke. The tissue temporal attenuation curve and associated maps (CBV, CBF, and TTP) are then estimated using the AIF and IRF calculated by that method [10].

In this collection, 12 of the total cases under analysis were excluded due to unreliable data: no contrast seen, excess of motion, or premature termination of the acquisition procedure. The remaining 115 were divided into a subset of 100 patients, for AI training described later, and 15 to validate the results and the performances.

6.3 Study and method

Employing Deep Learning approaches to the problem of deconvolving time-intensity curves, offers several potential advantages over canonical algorithms. Indeed, DL allows the extraction of information and features that are relatively insensitive to noise, misalignments and variance. Gava *et al.*[30] shows whether a properly trained Convolutional Neural Network (CNN), based on a U-Net-like structure, on a pre-processed dataset of CTP images, can generate clinically relevant parametric maps of CBV, CBF and time to peak TTP.

Since this specific model has been originally thought for medical images segmentation [67], and in this case the goal is to generate parametric images, some changes to the standard model are introduced. There is evidence that this CNN model is effective in other tasks, other than image segmentation, as well [29]. In contrast to similar state-of-the-art U-Net based methods, no extra information, like the above mentioned arterial input function AIF, has not been provided to the U-Net model.

We provide a short description on how to train an ANN model by using perfusion CT scans to generate CBV, CBF, and TTP maps. The raw inputs are a collection of multiple CT scans and 8 target CBV, CBF, and TTP maps for each patient, in *DICOM* format. *DICOM* format is an acronym for *Digital Imaging and Communications in Medicine* and it is the standard for the communication and management of medical imaging information and related metadata. Each target map corresponds to the output of the perfusion scan at a certain scan height of the patient's brain. Thus, given a target t generated at a specific height h for a patient p, we need to search between his CT scans for all images i^p (in grayscale format) acquired at the same height h. In our case $|i^p_h| = 89$, $\forall t^p_h$: the same brain portion appears 89 times at different scan timing. The input pre-processing consists in finding the images correlated by h and grouping them in a single image tensor, whose depth is equal to 89, for each t_h. Fig. 6.1 summarizes this

input preprocessing step. Therefore CT images can be viewed

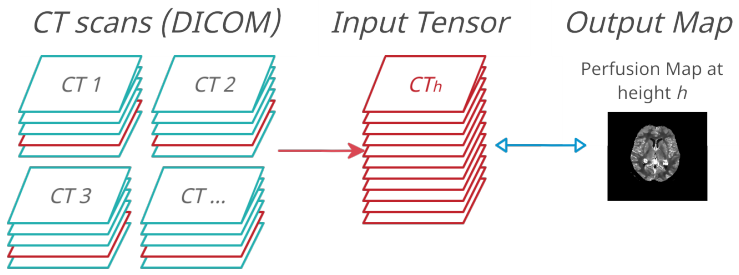


Figure 6.2: A representation of the *UniToBrain* input preprocessing step.

as 3D tensors, with the third dimension being the time axis. The pre-processed images are the only input for the trained U-Net network architecture [67]. Here, we consider using it for inference on perfusion maps. The overall model can be found in Fig. 6.3. We slightly change the original architecture design

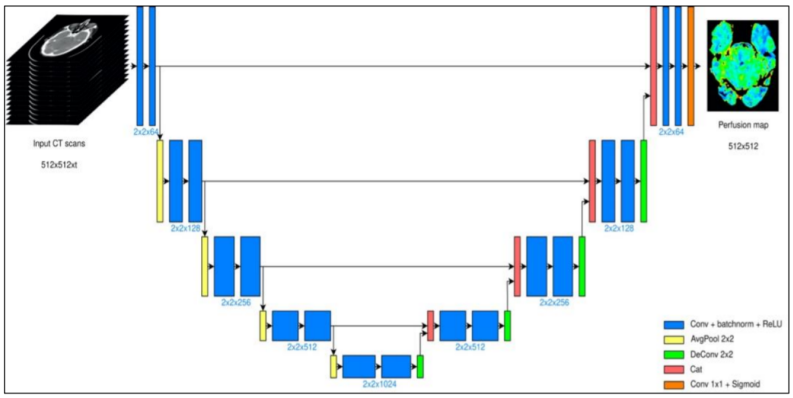


Figure 6.3: A representation of the U-Net architecture used to generate perfusion maps. The model takes as input the scans acquired in different time instants. The images resolution is 512×512. After four encoding stages, the decoder stages produce a map with the same input resolution.

due to empirical quality observation of the generated maps: Max-pooling layers have been replaced by Averaging-pooling layers. Using the traditional Max-pooling layer here is sub-optimal, as there isn't any expectation of sparse feature extraction. As described in Sec. 2.3, we use a common loss function

definition for image generation tasks. We minimize the mean squared error (MSE) loss between the required ground-truth and the U-Net output. Additional information or data (e.g. AIF) are not provided to the CNN: all information is found in the pre-processed input CT images. The model is pre-trained to produce a 128×128 output map for 250 epochs, after that it is trained and tuned for 50 epochs to produce 512×512 maps, in which all pixel values fall into the range $[0; 1]$. The choice to pretrain the U-Net model with downsampled images comes mainly from practical factors. Training on full resolution images requires a high consumption of GPU memory and the time needed to reach model convergence becomes very high. Training on downsampled images allows to reach a solution in a much shorter time, since the task is simplified. The pretraining step produces a suitable set of starting parameters for the model: this type of approach is often used for training generative networks, and the results are comparable, and smoother, than those produced with very high dimensional image training. The training of the entire model is done using the Adam optimization strategy.

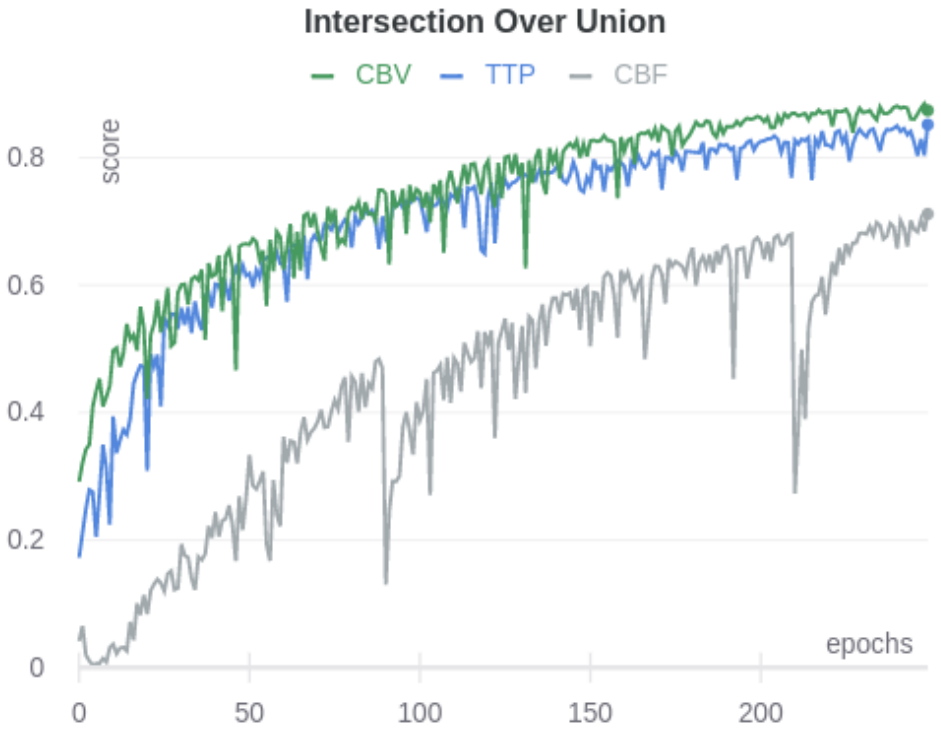
6.4 Results

The following are the results of training the U-Net architecture for the three identified target perfusions, CBV,CBF and TTP. The evaluation of the resulting images is performed by the value of the loss Mean Square Error (MSE) function, and a metric called *Intersection over Union*(IoU). While MSE clearly indicates how much the predicted maps deviate from the target, the IoU metric just indicate how many pixels reach a significant activation with respect to the target image. The IoU have not to be intended to have any medical meaning, because it takes in account also target meaningless regions for diagnosis. IoU is defined as the ratio between the intersection of the prediction and target areas, and their union.

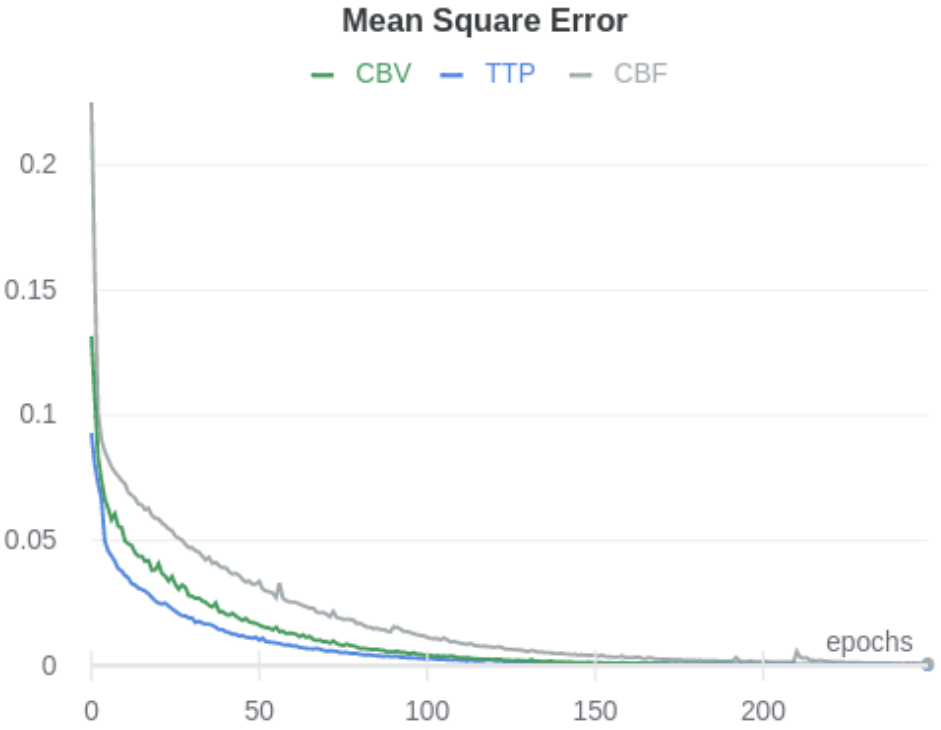
$$IoU(y, \hat{y}) = \frac{y \wedge \hat{y}}{y \vee \hat{y}} \tag{6.1}$$

Eq. 6.1 shows its definition, in witch y is the CNN output and $\hat{y}$ the ground truth. Since both y and $\hat{y}$ values fall in the range $[0; 1]$, a threshold is set to 0.5 and IoU is evaluated on resulting binary masks.

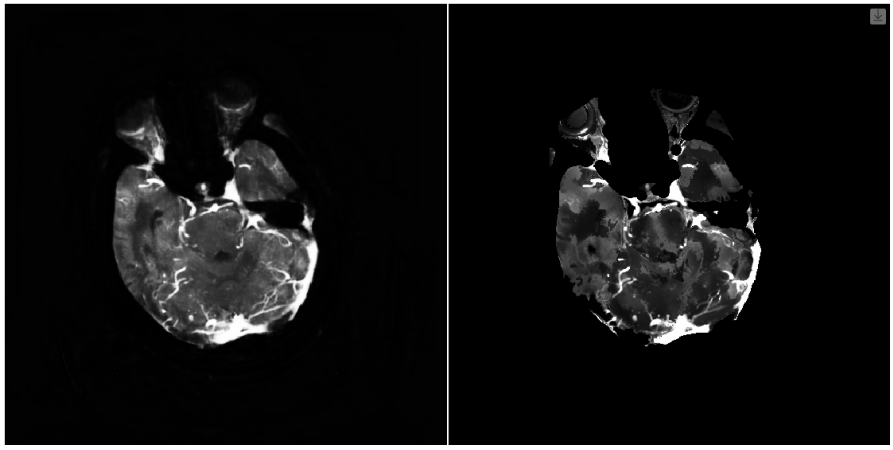
Fig. 6.4a shows how IoU increments during the pretrain phase, while Fig. 6.4b reports that MSE reach promising values below 0.01. While CBV and TTP experiments show similarities in both measures, CBF reaches lower IoU scores. From the trends of CBF curves, the map generation seams more difficult to be learned rather than CBV and TTP. Also MSE shows a slightest decrease for CBF, that confirm the major task complexity. Fig. 6.5 show some examples of generated perfusion maps from the U-Net training. The U-Net CNN can accurately evaluate the perfusion parameters by only using CT scans images only, without any other input information. On the other hand, decomposition-based CT and MRI methods for analyzing cerebral perfusion require additional input data, such as AIF curves measured in the saphenous vein. This suggests that the CNN solution can combine the information associated with both tissue and arterial signals to estimate CBV, CBF, and TTP maps. On the other hand, other recent automated analysis methods use AIF to estimate ischemic core before calculating perfusion parameters [14]. The U-Net CNN approximates sufficient perfusion mismatch in brain tissues to create an information-rich perfusion map for patients with ischemic lesions. In addition, the growing potential of machine-learning-based perfusion analysis methods could lead to new and improved standards in acquisition protocols and administered radiation doses.



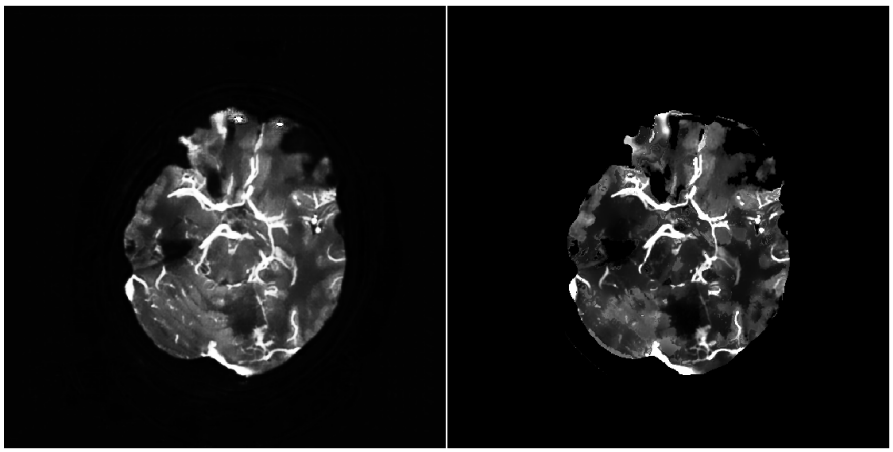
(a) Growing IoU metric on the test set during the pretraining phase.



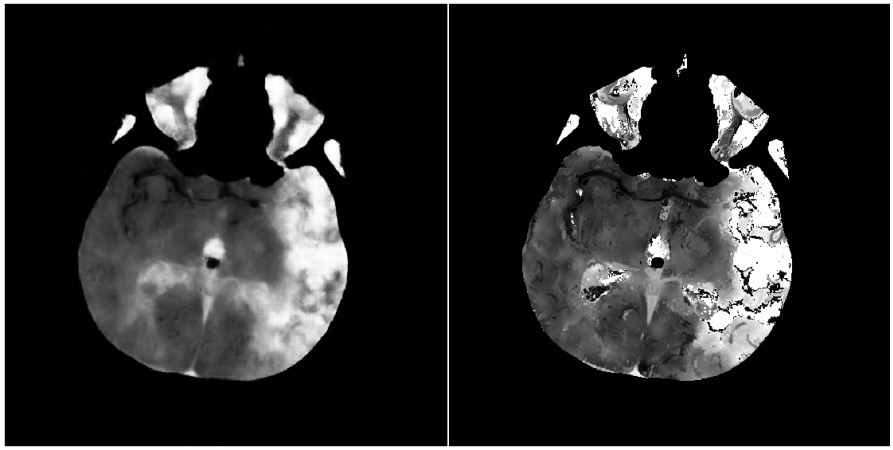
(b) Decreasing MSE metric in the same train configuration. Low is better, it indicates that the generated maps are numerically very close to the targets. MSE does not require image binarization to be evaluated.



(a) CBF



(b) CBV



(c) TTP

Figure 6.5: Examples of map predictions for each target type. The image on the left represent the U-Net output, while the ground-truth is on the right.

Chapter 7

DeepHealth European Libraries

The European Union is spending significant amount of resources properly leading the AI development and research to maintain its proper position in this emergent and challenging context. EU funds projects to deploy pilots and large-scale experimental research solutions by combining the latest discovered technologies in artificial intelligence. The same effort is spent to support AI technologies like distributed high-performance computing, cloud computing and big data support. *DeepHealth*, as described in the first chapter, is an example of this effort, that is thought (and developed) as health-focused project. The European framework of *DeepHealth*, called *DeepHealth Toolkit*, is a single entity that aim to use heterogeneous architectures, such as high-performance computing, big data and cloud computing, to provide deep learning capabilities and computer vision off-the-shelf services, that can be easily integrated into any software project and deployed on any computing platform. *DeepHealth Toolkit*'s aim is to make it easier to develop and deploy new applications that solve specific problems regardless of the type of context or application field. This chapter mainly describes our experimental contribution to the *DeepHealth Toolkit*, by taking into account some of its main components, in particular *EDDL* and *ECVL* libraries, and performing performance testing and

comparisons. The performance reports are comprehensive of model training time, CPU/GPU usage, GPU allocation, memory allocation, and GPU temperature, by showing multiple distributed training scenarios. We first present the idea of how *DeepHealth Toolkit* is indented to use to facilitate its usage in the European community by developers and data scientists working on AI-based solutions in health care and other fields. A visual representation is shown in Fig. 7.1.

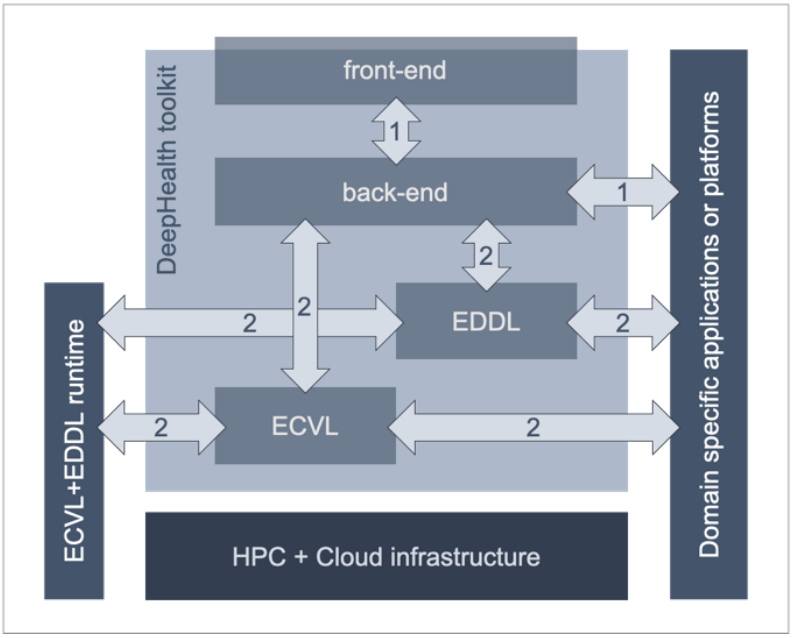


Figure 7.1: *DeepHealth Toolkit* schema that shows service exchanges between all its inner components. This schema is taken from book chapter dedicated to *DeepHealth* [4]

The *DeepHealth Toolkit* is a versatile deep learning system with fully integrated image processing and computer vision (VC) capabilities that uses HPC and cloud infrastructure to perform parallel and decentralized AI inference and learning processes. Under the MIT license, each component of this EU instrument is freely available as open source software. The toolkit is designed to handle large and growing datasets, by matching the nature of medical image datasets. For those large datasets, more and more sophisticated neural networks and learning strategies can be used to improve the accuracy and reliability of AI-based predictive models, but at the expense of

much higher computing power requirements. To that end, the *DeepHealth Toolkit* solution aims to transparently integrate the latest parallel programming technologies to leverage the parallel performance capabilities of HPC and cloud infrastructures, by including symmetric multiprocessors (SMPs), graphics processing units (GPUs) and various other computing acceleration technologies. It also includes external frameworks to generalize the toolkit to specialized infrastructures and allow different levels of parallelization mechanisms. The toolkit also includes features that can be used for training and inference by abstracting the complexity of different computational resources and facilitating architecture developments adopted in the learning and inference phases. AI specialists will train AI solutions on dedicated HPC architectures equipped with the latest GPU components. The goal is to maximize the number of the data samples processed each second, while the overall accuracy remains stable. The AI inference (or test) phases are made in a future production environment, by including small dedicated devices, by using trained models, where the response time required to predict a single data sample, or image, is critical. In order to support AI-based application developers, data scientists, and ML professionals in general, DeepHealth's unique toolkit is designed to provide the following targets:

- Increasing the productivity of both data and computer scientists, by reducing the time required to the training processes by parallelization on HPC and cloud infrastructures, and also providing a tool to speed up the design and test of predictive solutions in a unique tool.

- Facilitating the development and deployment of AI-based applications and models, by providing the most common DL and VC capabilities in a single, integratable toolkit that supports multiple operating systems. It also allows DL trainings outside of the application/platform installed by installing locally those *DeepHealth Toolkit* libraries the developer needs.

- Freeing the needs for highly professionals HPC and cloud

experts. The AI solution learning processes can run transparently to the data scientist. Therefore, data scientists and developers can work with less in-depth knowledge of HPC, DL, VC, big data or cloud computing. Furthermore, a production application does not need to be adapted to running multiple distributed processes on HPC infrastructures.

A more detailed overview can be found in the publication dedicated to *DeepHealth* [22].

7.1 EDDL

EDDL is a deep learning library that was originally developed to be general-purpose and meet the AI development needs inside the DeepHealth context, therefore with the aim to be used in healthcare applications. *EDDL* is a free and open source software library, available on online repository as a core part of the *DeepHealth Toolkit*, as we can see in Fig. 7.1. At the moment, it integrates and supports the most popular and well known deep neural network topologies, by including both convolutional and recurrent models. These models can be used for a variety of already discussed tasks such classification, time series prediction, machine translation, semantic image segmentation, and also image description. *EDDL* uses the *Open Neural Network eXchange* format (*ONNX* [62]) in order to interface itself with other DL libraries, enabling the import and export capabilities of neural networks models, by including weights (if already trained) and topology. It is essential for maintaining compatibility and exchanging solutions with existing and widely used development tools and other deep learning tools, like *PyTorch* or *TensorFlow*. *EDDL* aims to simplify the development of hardware-specific accelerated deep learning mechanisms by providing hardware-independent tensor operations and ANN topologies components like activation functions, regularization functions, optimization methods, and multiple types of different layers. Those components fall into

two main categories: ANN operations, as models, layers, regularization functions, initializers, and tensor specific operations like serialization, I/O, and mathematical transformations. The tensor is the mathematically manipulable data format which allows to perform all the needed calculation in a DL application. The neural network library contains both high-level tools, such as training and evaluation of a model, and low-level tools that allow for finer control of each step in the training or inference loop, like individual epochs, batches, or gradient manipulation. The core implementation of *EDDL* is done in *C++* programming language. In order to rise the attractiveness of *EDDL* to the whole scientific community, A *Python application programming interface* (*API*) has been developed, with the name *PyEDDL*. *PyEDDL* is a Python library that provides *EDDL* functionalities and interfaces them to a lot of popular scientific programming and data manipulation tools, like *NumPy/SciPy* and *Pandas*. In particular, *PyEDDL* supports bidirectional conversions between internal *EDDL* tensors and *NumPy* tensors, which is a key integration step to achieve compatibility with other existing *Python* libraries. *PyEDDL* is born as a native extension of the *EDDL* by wrapping its *C++* core. It allows users to take advantage of all the benefits of using *Python* language for development and by using *Python* as a high-level abstraction to combine computationally intensive procedures. Like other elements of the *DeepHealth Toolkit*, also *PyEDDL* is released as free and open source software, and the source code is available. Neural network build functions are already included in *PyEDDL/EDDL* framework. These functions create all needed data structures, based on the network topology itself, and allocate the necessary memory. Via these build functions, it is possible to describe the available hardware that *EDDL* will use to perform the training and other processing. Each hardware description is called *Computing Service*. At the moment, there are three types of *computing services* defined: CPU, GPU, and FPGA. computational services. *EDDL* GPU support is provided by a specific *CUDA kernel* developed as a component of the *EDDL* core. As mentioned earlier, the use of different hardware and architec-

ture wants to be completely transparent to AI developers and programmers. By simply creating the appropriate *Computing service* to use all or a fraction of the available computing resources. The *DeepHealth Toolkit* also provides the computing GPU acceleration *NVIDIA cuDNN library* as integration into *EDDL* as an alternative to the native *CUDA kernel*.

EDDL represents internally the topology of a neural network by two different directed acyclic graphs *DAGs*, for forward and backward training stages. Each *DAG* represents the sequence of mathematical operations to be performed on tensors, in order to make the computation described in the network model definition. Each layer computation is done when all dependencies for that level are satisfied, according to the *DAG*. Therefore a layer computation is performed when all the computation for its inputs are done. By working on GPUs, *EDDL* has three memory management modes to manage possible memory shortages. It helps *EDDL* to scale over different hardware components. In the most performing mode, *EDDL* tries to minimize as possible the number of memory transfers by allocating the most of the graph tensors to the GPU memory. On the other hand, less efficient modes handle larger graphs by increasing the number of memory transfers for forward and backward stages. In distributed learning in HPC and cloud architectures, *EDDL* includes special features to simplify the batch distribution for training and inference processing by using well known HPC frameworks. Those frameworks' purpose is to accelerate DL learning by partitioning the training dataset into a large number of compute nodes to perform partial learning processes. *EDDL* uses *ONNX* to serialize the network and manage weights and gradient transfers between a master node and worker nodes. In order to meet different needs in distributed learning applications, the serialization feature implemented in *EDDL* allows the user to choose whether to include weights or gradients. The *PyEDDL/EDDL* code is tested by a full test suite, by covering a number of use cases in *Python* and *C++*, by including various models training and evaluating scripts, *ONNX* serialization, *NumPy* compatibility, etc. *PyEDDL/EDDL* provide the documentation and ex-

amples in order to facilitate the library installation and the applications' development.

7.2 ECVL

The *European Computer Vision Library (ECVL)* is the *DeepHealth* computer vision library, developed to be general-purpose, with a special focus on supporting healthcare applications. *ECVL* goal is to facilitate the integration between applications and existing libraries, such as *OpenCV* [61]. It provides high-level computer vision capabilities, by providing also specialized and hardware accelerated implementations of commonly used image processing algorithms in deep learning. Of course, these capabilities are useful for image processing in other application areas and contexts other than strictly in healthcare. *ECVL* is designed to provide its services on top of a *Hardware Abstraction Layer* (HAL). HAL allows, while maintaining the same interface, to maintain a flexible differentiation of hardware for accelerated features. The hardware-independent API simplifies the libraries usage of and facilitates the development of image processing components in distributed and parallel computing environments. *ECVL*'s design revolves around the pivotal concept of *Image* as the centerpiece of the entire library. *ECVL* gives support to many scientific image data types and file formats, like *jpeg, png, bpm, ppm, pgm.* Special features are included in the library for medical image data manipulation. These images are often retrieved with proprietary or multi-scan formats, such as DICOM, NIfTI and others virtual slide formats. In some of these complex image formats, the *Image* object representation allows to highlight a specific area and magnification, in order to extract the interested image portion from the file. *Image* component is designed to capture and process a wide range of different types of images. A main feature is the capacity to work across different image channel configurations, by providing read and write operations for all the already mentioned data formats. *Image* provides the arithmetic instruments

in order to apply mathematical operations between images and scalars. More common affine image transformations, such as rotating, scaling, mirroring, and color space changes are available. *ECVL* image transformations implementations are designed to run during training of deep neural networks as data augmentation technique support. *ECVL* provides a simple *domain-specific language* (DSL) [15] in order to extend the definition of applicable transformations and their input parameters. This allows customization of sets of options to be defined for each data subset, like training, validation, and test. Additional modules are included in the library and, when enabled, provide additional functionality, such as the *3D Volume Viewer*, which allows users to easily visualize images of *ECVL* imaging objects and view different *CT* slices from different angles. *ECVL* defines *DeepHealth Dataset Format (DDF)* as a simple mechanism for describing an image dataset that will be used in training distributed models. DDF consists of a YAML-based syntax document: a DDF file contains all the information needed for dataset characterization, data loading, image pre and post-processing. In addition, special *Dataset* modules for reading and parsing DDF files have been implemented and are available through the library interface. *ECVL* is accompanied by a *Python API* called *PyECVL*, just like *PyEDDL/EDDL*, whose main advantage is that it not only speed up application developments but also integrates with the rich *Python* ecosystem of scientific of programming tools. *PyECVL* is also based on a wrapper that redirects calls to the *C++* code, allowing users to take advantage of *Python* development without sacrificing performance. *PyECVL* provides all *ECVL* services, including *Image* objects, different data types and colors spaces, mathematical operations, image processing, input/output functions and DDF parser extension. As already said, support for *NumPy* allows developers to process data with many other scientific tools. *ECVL* born to support the use of GPUs to run the computer vision algorithms needed for Deep Learning. The GPU implementation also uses *CUDA kernels* and this makes the AI solutions more power-efficient and cost-effective. *PyECVL/ECVL*, as

well as other toolkits, are available as open-source software. Each release includes documentation, extensive testing suites, and several use cases with both interfaces. Examples include *DeepHealth* dataset manipulation, *EDDL/PyEDDL* handling, image processing, and I/O. Therefore, the documentation includes detailed instructions for installing each library independently.

7.3 Comparison with PyTorch

In this section we want to try to give an idea of comparison of performance and DL application development paradigms between the *EDDL/ECVL* libraries and the widely used *PyTorch*. The comparison is made, specifically, using the *DeepHealth Toolkit* with the respective *Python* portings, *PyEDDL* and *PyECVL*. First, we want to give an overview of differences in AI application development paradigms between the AI *Python* engines. After, we set up multiple run configurations, for both the use cases seen before, in order to show and discuss performance benchmark results. We perform all the simulations on the *HPC4AI* [8] infrastructure. *HPC4AI* is a federated cloud platform for high performance computing developed by *University of Turin*, in collaboration with other local institutes, that support the AI application development and distribution for research and industries. We use a remote container manager, called *Kubernetes*, in order to access the HPC4AI computational resources. All the distributed learning setups are made by accessing the same resource instance that consists of four 16 GB NVIDIA GPUs Tesla T4 and CUDNN support.

AI model definition

Starting from the definition of the AI model, the architecture is defined *PyEDDL* as a concatenation of layers that evolves from a basic entity defined as `eddl.Input`, also belonging to the *EDDL* definition of Layer. Every subsequent operation, that is

every additional layer of the model, is defined taking a previously defined Layer entity as input. Basically, a neural network architecture is represented as a Layers chain. Also weight initialization algorithms are defined as Layers in the chain. Here we notice the first difference in the paradigm of development in *EDDL* and the paradigm used by *PyTorch*. In the latter, the model is a set of Layer components, while the chain of actions is defined separately in a forward function. This combines the outputs of the defined layers, not the layers themselves, directly from a more generic input tensor. The different design choice for *PyEDDL* comes from the ability of the *EDDL* engine to calculate the dimensionality of the tensors involved at each step of the chain in a transparent way. This leads to simplification in the development of architectures, since, differently from the *PyTorch* engine, the developer has to worry about the correct dimensioning of the tensors. The interchange of architecture definitions between the two development paradigms is possible through the *ONNX* tool, which defines a common and standard encoding.

The definition of AI Model in *EDDL* is not limited to its architecture. In *EDDL*, a model is thought of as the whole chain of layers plus the "building" of necessary components to train the architecture. Consequently, the whole model includes the definition of the optimization algorithm, e.g. *SGD* or *Adam*, the declaration of the loss function, an evaluation metric, and most important the definition of the hardware on which to train the model. This last component is necessary to define the abstraction level for distributed computing described in the previous section.

Dataset management

PyTorch exposes the generic primitives necessary and expandable for dataset handling, be it image data, tensors, audio or other formats. Unlike *PyTorch*, *PyEDDL* and *EDDL* do not have proper internal dataset handling. The dataset management is delegated to *PyECVL/ECVL*, which uses the description of the data in the *DeepHealth Dataset Format*, already

described in Sec. 7.2. Since *ECVL* is a library dedicated to computer vision, it is not possible, at the moment, to use the *PyECVL* API to work with different data formats. For example, the *UniToBrain* use case uses gray-scale images only for the target maps, while the input images are stacked in tensors with high depth as single input. In addition, at present time, the management options for the image dataset itself are still limited. There are no native APIs that allow balancing and sampling options, which are features present in other widely used libraries.

Of course it is possible to describe the dataset in the *DeepHealth Dataset Format*, but at the moment the API provided does not allow handling non standard data format. In our opinion, this is a current limitation that weakens the incentive to use the DDF. It must be said that *EDDL/ECVL* are now in their first official release, so there will surely be room for expansion to embrace the needs of AI applications in more complex contexts.

Data augmentation

As with dataset management, data augmentation options are delegated to *ECVL*. As we saw in Sect. 3.1, data augmentation is a very effective regularization technique to avoid overfit of neural network-based AI models, as they present "variants" of the input images to improve generalization. These techniques are widely used especially in case of poor and unbalanced datasets. Currently, *ECVL* does not expose a high supply of effective data augmentation techniques. For example, unlike PyTorch, many useful transformation functions, such as color jittering, saturation, solarization and hue adjustment, are not yet available. Above all, the ability to define a custom image transformation function in the data augmentation pipeline, equivalent to *PyTorch*'s `tranforms.Lambda`, is missing.

These gaps will surely be filled in the future by *ECVL* developers in the future releases, so as to encourage the usage of European libraries for AI applications without having to worry about losing in capabilities and expressiveness.

Training and inference loops

If we compare the construction of the training loops in the two libraries, *EDDL* tries to offer more primitives for training management at different abstraction levels. *PyTorch* offers a unique set of APIs for a low level management of all the steps of the training loop: choice of the data batch, the model prediction step, followed by the calculation of the loss function, the optimization step, update of the neural network weights and eventually the learning rate management step, called *Scheduler*. This low level management is also offered by *EDDL*, to which are added also more generic APIs that come to group all these elements. Certainly, for more complex or peculiar tasks, their usage is not recommended, as the training loop is used as a black-box, limiting the control on it. In its first release, *EDDL* is still missing some important management options, like the *PyTorch Scheduler*, in order to offer a complete set of tools to replicate the current state-of-the-art DL applications. Some of these features will be added in the future to spread *EDDL* usage.

Distributed training

EDDL has been designed to simplify the use of distributed learning. Distributed learning identifies the possibility to split the computation on multiple dedicated hardware, accelerating, in the case of AI applications, the training time. Distributed learning requires that copies of the trained model be deployed on different hardware, such as multiple GPUs, and that the data to be split between them. During the optimization step, parameter updates are averaged over the results of each copy, and all copies are updated equally. This feature is essential to achieve high performance in high-performance computing environments, taking advantage of the full resource capacity provided. Setting up distributed training with *EDDL* is very simple. During the "building" phase of the model, already mentioned above, it is possible to indicate the references to the hardware resources, or devices, to be used and the frequency

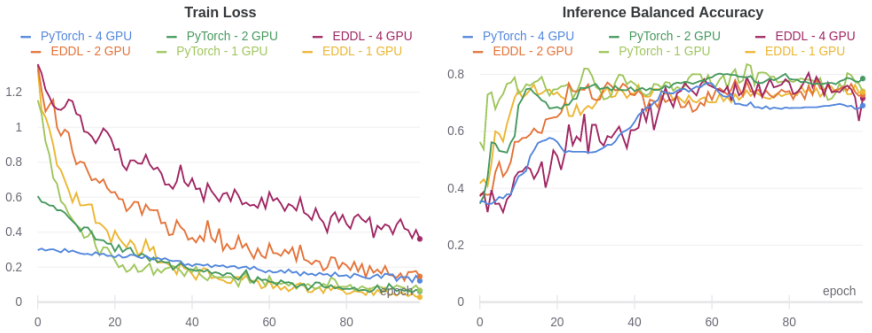
of synchronization of the copies of the model. The training is so distributed, in a completely transparent way to the developer. *EDDL* internally takes into account the devices used and synchronizes data between them. Compared to a widely used solution like *PyTorch*, this is a remarkable simplification. *PyTorch* requires, in a distributed computation application, wrappers for the dataset and the model, as well as a manual (and error prone) management of the processes and their concurrency for each GPU.

7.3.1 Evaluation on UniToPatho

In this section, our purpose is to evaluate how much the European libraries can be considered competitive, at the current state of development, with respect to the tool that currently dominates the landscape in the development of DL applications. We evaluated the different implementations, in *PyTorch* and in *EDDL* on HPC4AI environment, by accessing exclusively to 4 NVIDIA GPUs, and we use distributed training to compare the libraries. In this subsection we are going to compare the performances by reproducing the training of one of the necessary components of the inference pipeline on histo-pathological images, previously presented in Sec. 5.2.1. In particular, the training of the intermediate classifier, dedicated to the detection of adenomas, is proposed [1]. The training is carried out for a total of 100 epochs using the images available in *UniToPatho* at 7000×7000µm resolution resized to 2224×22424 pixels. For each simulation we used the same parametrization: we use the SGD optimizer applying a learning rate of $1e^{-4}$, momentum equals to 0.99, and weight decay set to $5e^{-4}$. In order to have a fair comparison, we use just the image augmentation proposed in both libraries: image flipping on both vertical and horizontal axis, random rotation, contrast and brightness adjustment. Other image augmentation, such as saturation, solarization and hue adjustment have been omitted.

We experimented and benchmarked how the two libraries perform during distributed training with dedicated hardware.

The trainings, are performed in different setups: 1, 2 or 4 dedicated GPUs. For each training, we increase the batch size accordingly with the number of GPUs, in order to have the same amount of data sent to each GPU across all the runs. In these experiments we use a batch size of 32,64 and 128 respectively.



(a) (b)

Figure 7.2: Performance summary on different use case setups. It is possible to see how the convergence of the model to the solution is caught up for every proposed setup. *EDDL* runs are highlighted in purple, orange and yellow, while *PyTorch* runs are in the blue to green range color.

Fig. 7.2 shows that each implementation can converge to a solution, which have similar prediction performances for each distributed training setup. Fig. 7.2a shows dissimilarity in reaching low error values during the training phase. This behaviour comes from the different policy used by *EDDL* and *PyTorch* to synchronize the training model across multiple GPU. *EDDL* synchronization is done by averaging the model's weights after each iteration, so the overall model is the average of each GPU-local set of weights. Instead, *PyTorch* synchronization is designed to average the computed gradients for each iteration, and then update each GPU-local model. In other words, the overall model is produced by updating its weights after averaging local update's directions.

Fig. 7.3 shows benchmark results on resource usage in comparison between different runs. Fig. 7.3a shows how the duration of each training epoch is higher for all the setups with

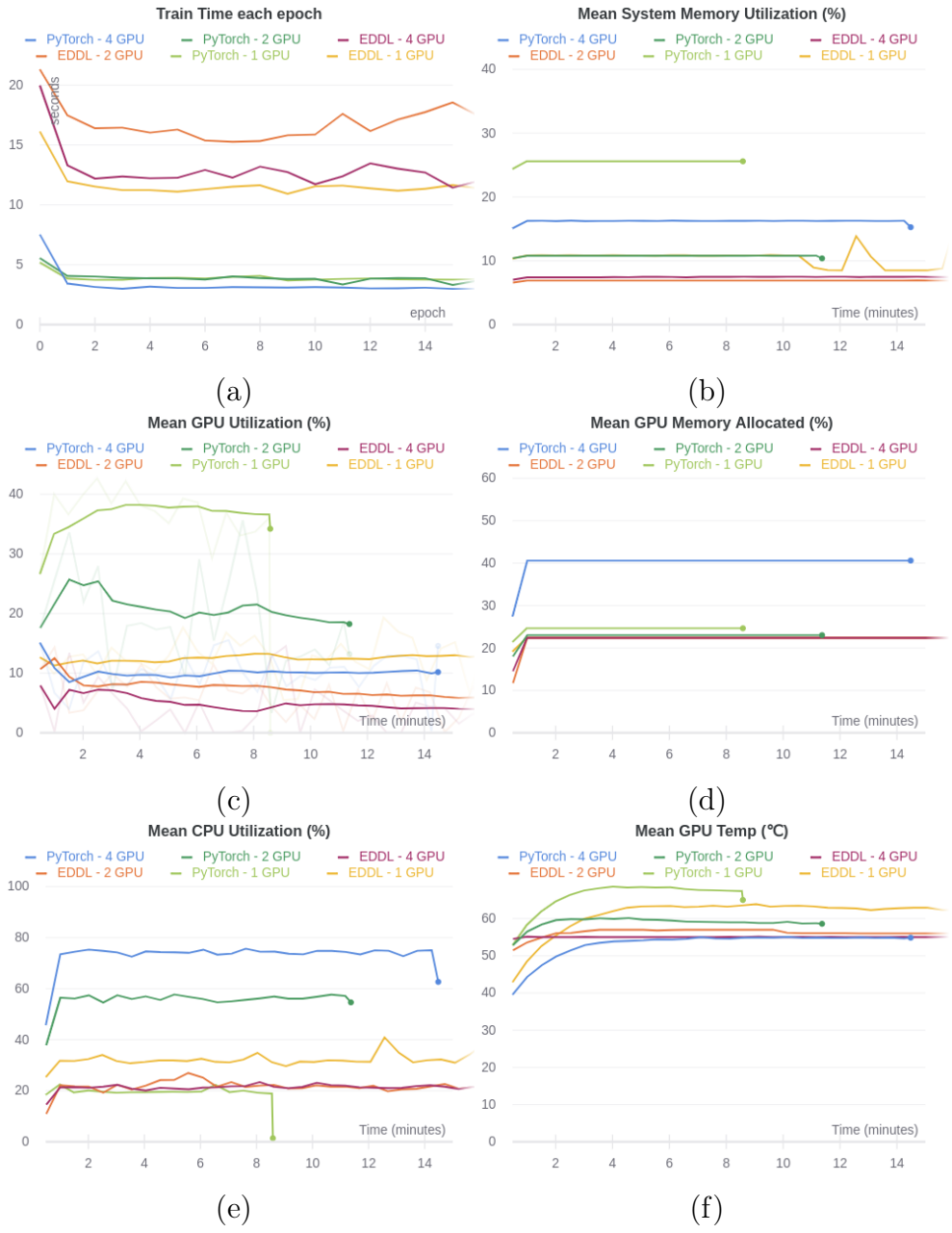


Figure 7.3: Resources usage summary on different setups for *UniToPatho* training.

the *EDDL* implementation. In this case we can not see a direct link between the training time and the number of GPU in *EDDL* runs.

While *PyTorch* implementation shows, for each setup, an adaptive resources consumption, we can see a more constant behaviour by using the European library. For example

Fig. 7.3b shows how how the system memory requested by *PyTorch* changes with the number of involved GPUs. Instead, *EDDL* doesn't not show similar memory usage cross different setups. The same behaviour can be seen also in Fig 7.3c and Fig 7.3e in GPU and CPU utilization respectively. Furthermore we notice in Fig. 7.3c how the mean GPU usage decreases if we enlarge the GPU pool for *EDDL*. This behaviour can be brought back to what we said in Sec. 7.3: *EDDL* is currently missing a dedicated dataset management system, or data-loader, in order to supply the data efficiently on each GPU. A less efficient data loading resolves to latency and under usage of multiple GPUs capabilities. However the development of *EDDL* is at its first release, so there are incoming improvements and integration.

Inference Model [1]	Time To Inference [s]		
	HP	Adenoma	Dysplasia Grade
EDDL/ECVL	3400	11.9	5330
PyTorch/OpenCV	1374	5.3	1381

Table 7.1: Timing differences in the *UniToPatho* inference pipeline between both implementations. Adenoma classification uses subsampled images at 224×224 pixels resolution

As a counter-evidence, the entire inference pipeline for the *UnitoPatho* dataset was implemented in *EDDL*. Tab. shows the execution times, by using a single GPU, for inference on the test set of the dataset, including all the steps required to process the 800×800µm images at full resolution. The implemented pipeline uses the same pre-trained networks, in order to measure the impact of using *EDDL* as inference engine and *ECVL* as image processing engine. The results show that a gap still exists, again due to *EDDL*'s lack of dataset management.

7.3.2 Evaluation on UniToBrain

We evaluate again the concurrent implementations on the same HPC environment with four 16 GB GPUs NVIDIA Tesla T4. Only comparisons for CBV perfusion synthesis are proposed

below, as there are no notable variations in performances or resource consumption compared with the other task targets, TTP and CBF maps. As before, we setup the same environment and the same experiment conditions to compare the two libraries during distributed network training on dedicated hardware. As previously, the setups are designed with 1, 2 or 4 dedicated GPUs. The refinement steps to 512×512 images, for 50 epochs, of the pre-trained model at 128×128 resolution are considered. All simulations with *EDDL* were performed using the "full memory" option available on the neural model building phase. A 4 GPU run with "low memory" setup is shown as a comparison. As before, we increase the batch size each time we scale the number of GPUs, in order to fit the same number of images processed on each GPU in each setup. Due to the size of the inputs, 512×512×89 wide tensors, we use a batch size equals to 1 for each GPU, and small learning rate of $1e^{-7}$.

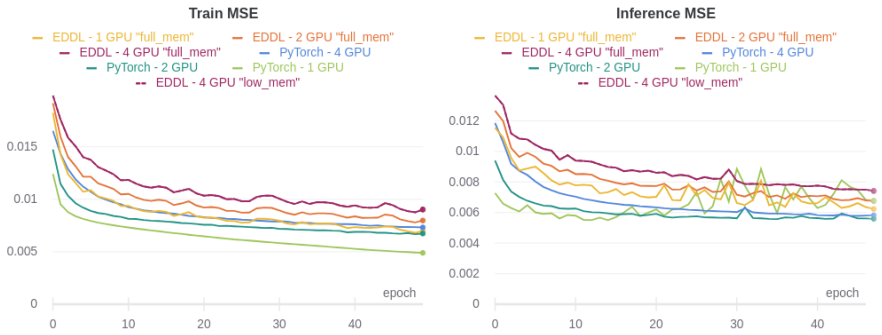


Figure 7.4: Performance summary on *UniToBrain* use case. As for *UniToPatho*, we maintain *EDDL* runs in purple to yellow range color, while *PyTorch* runs in the blue to green range color.

Fig. 7.4 shows the convergence to a solution is reached for each training and for both libraries. The small differences in the validation MSE metric is caused by the increments in the batch size.

Fig. 7.5 propose the benchmark results also for *UniToBrain* trainings across different implementation setups. Also Fig. 7.5a shows how of each training epoch duration increases in all the *EDDL* implementation setups. In this scenario, the

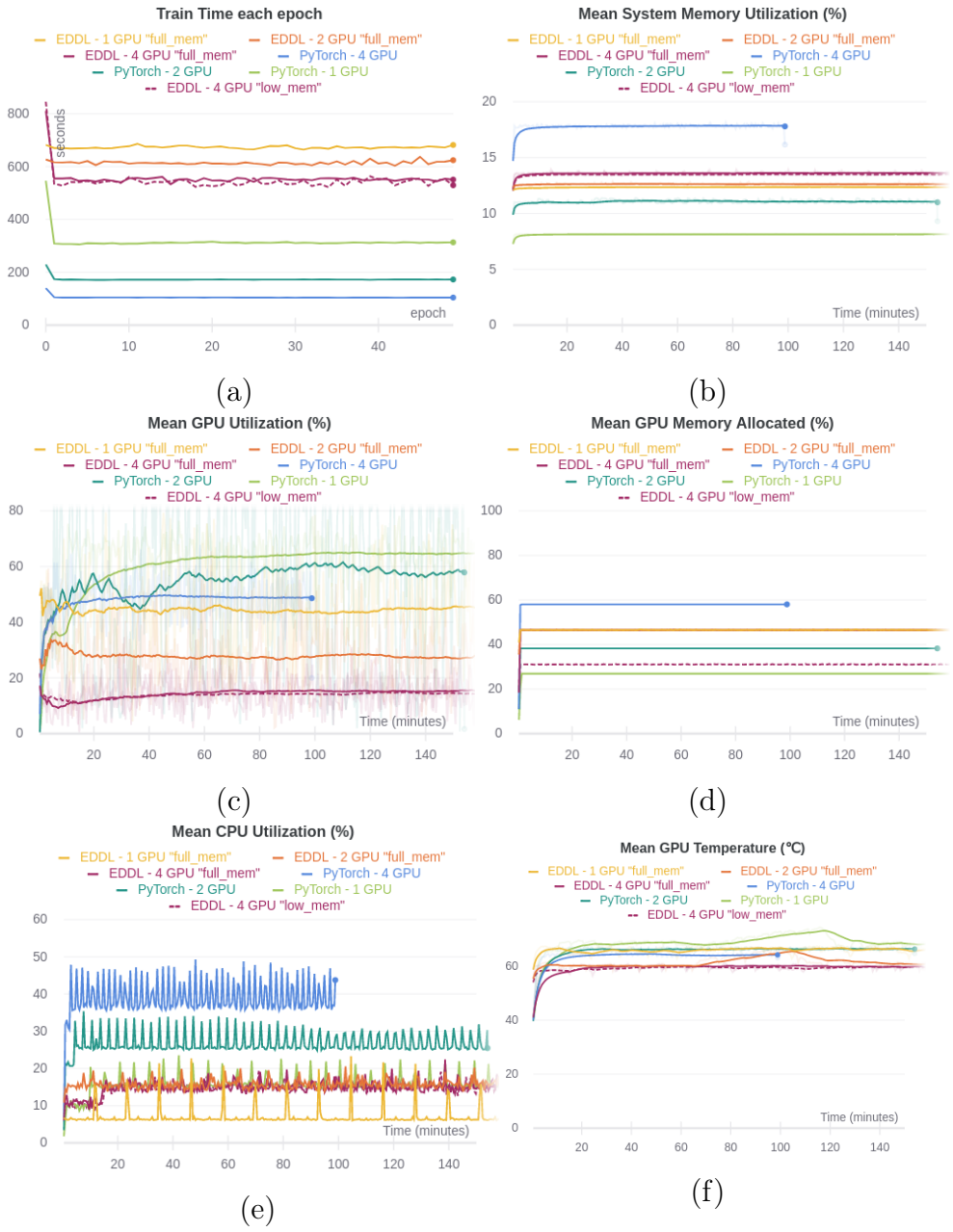


Figure 7.5: Resources usage summary on different library and environment setups.

gap between the execution time, across different implementations, is more evident: time decreases when the number of GPUs is increased, but not enough to have a competitive comparisons with *PyTorch*-based solution. Again, the whole resources consumption, shown by *PyTorch* implementation setups, is adaptive on the number of available GPUs. The con-

stant resource consuming behaviour by *EDDL* does not change. As before, as we can see in Fig. 7.5c, the mean GPU usage decreases by increasing the dedicated hardware. This behaviour support the previous consideration on the *EDDL* dataset management. Fig. 7.5e also shows, in terms of CPU usage, the minor support to the data transfers when the training is dealing with low batch sizes. We can see, as expected, the only significant differences between *EDDL* "full" and "low" memory setups in GPU memory allocation graph in in Fig. 7.5d.

In conclusion, *EDDL* is not yet at a sufficient stage of development to replace, in terms of performance, solutions that currently dominate the AI application development landscape. Since the first release, *EDDL* already gives the possibility to simplify the development of distributed training algorithms, laying the foundations for a simplification in the use of hardware acceleration components transparent to the developer. With the addition of still missing components, such as an internal data-loader manager, the performance gaps will be significantly reduced.

Chapter 8

Conclusions

We see how the entire landscape for biomedical research, and healthcare in general, is growing in the direction of including artificial intelligence applications in their processes chains. The research in advanced clinical decision support have been developed since the artificial intelligence concept began to find its first application to solve real and complex problems. For this reason, AI research finds an highly challenging application field in medicine. We believe that the key role of AI integration into the medical and diagnosis pipeline will be enhanced with machine learning development of accurate medical predictions, because healthcare needs to constantly improve the quality of its provided care. We show how, in reality, healthcare constitutes a particularly complex context for the application of AI solutions, and each diagnosis support challenge, or task, must be addressed starting from the assessment and analysis of its related medical data. We explain how complex feature extraction from medical data is, and how much this is related to the collection and processing of the data itself. The research results presented in this thesis have contributed to the European Community efforts towards the development of a common infrastructure to hold-up healthcare AI applications. Such an effort consists in the presented *DeepHealth* project. We see how the AI for medical application evolved through time, from simplest rule-based prediction systems, defined by medical expert users, to Deep Neural Networks applications, based on automatic features extraction on the medical data.

By using this paradigm, we see how different tasks can be addressed to solve real-world problems. One of the challenges in training a deep neural network is to define a model with high generalization results. Generalization improvements in predictions are an active research area in training a neural predictive model. We see how we can avoid a model to overfit, and therefore retrain the whole dataset by including different types of model regularizations. We also propose *PSP* as a practical solution, and we see how such a technique can be applied also in the automatic treatment of medical tasks. We see a practical use case in treating histopathological tissue analysis by studying tissue fragments associated with a particular disease. This thesis investigates colorectal polyps classification in-depth, through applying Deep Learning techniques to image processing. We see how in the medical image processing field, the collected raw data, e.g. 2D scans like tissues Hematoxylin & Eosin histology biopsies, presents dimensions for each sample and high variance compared to their diagnosed class of belonging. By focusing on colorectal polyps classification case, the tissue scan data collection for the *Deephealth* project continued beyond our publication of the first version of the dataset *UniToPatho*. The collection is growing by including new expert annotators for the biopsies samples. We present an in-depth study on how the histopathologic tissues samples collected within *DeepHealth* are suitable to investigate and solve this task, by taking in consideration data unbalancing treatment in tissue image preprocessing. We see how to approach classification of *UniToPatho* images in multiple ways, especially by developing a multi-resolution ensemble solution of cascaded classifiers. This method helps us to reach performance qualitatively comparable to the pathologist diagnosis. The study on image processing in the medical field was enlarged by taking in consideration another use case, related to brain image analysis. Specifically, we focus on collecting and analyzing Computed Tomography scans of the brain, and we use AI deep learning algorithms to identify cerebral vessel occlusions in ischemic stroke cases. This study has been included also as a use-case in *DeepHealth* project, by taking

the name of *UniToBrain*. We show how is possible to reach an Mean Square Error score below 0.01, that is a powerful indicator on how the generated perfusion maps from CT scans can be taken into account in order have an early and helpful instrument against ischemia stroke cases. The European Union effort to maintain its proper position in this emergent and challenging AI context is realized in the inclusion of our use cases and their solution development by using a common API interface, the *DeepHealth Toolkit*. By using this solution, try to give you an overview of development paradigmas and performance comparison on deep learning application development paradigms between its *EDDL/ECVL* components and the widely known and used *PyTorch*.

8.1 Future work

There are many directions in which the completed research can be further explored and deepened. Regarding the automated analysis of histopathological tissues, some types of polyps have been excluded from the collection, such as sessile serrated polyp (SSA) and traditional serrated (TSA) due to the impossibility of collection in a systematic way by the medical institute. For this reason, it would be interesting to make an analysis on the optimal resolutions to be taken into account in order to make a more complete diagnosis. In addition, an optimal method for inferring predictions from tissue clippings to the whole biopsy result has not yet been defined, since in the same biopsy there can be both areas of poor diagnostic interest or multiple conformations of features for several types of polyps, whose diagnosis has, however, different impacts on the routine colonoscopic surveillance of patients. This part of the research must be done in close contact with teams of expert pathologists, since the sensitivity of the automatic diagnostic tools must be such as to be a useful support in the diagnostic pipeline in the field. In addition, AI models are sensitive to the bias introduced by the scan acquisition tool. A lot could be added to the research with the acquisition of biopsies with

different tools, including various institutes, to verify the AI applications generalization, based on deep learning, on raw data from different medical organizations. Regarding research on the inference of cerebral perfusion maps for patients with ischemia strokes, there are many prospects for expansion. As for all medical tasks in general, also in this case the AI models are sensitive to the bias introduced by the image acquisition tools, in this case CT scans. In perspective, it would be interesting to investigate a solution based on data from multiple institutions. Moreover, the performance obtained can inspire to face other types of representations, or maps, even related to different categories of brain classes, putting this as a tool of prior prediction with respect to the following examinations that the patient will have to face.

List of Figures

List of Tables

Contributions

[1] Carlo Alberto Barbano et al. "Unitopatho, A Labeled Histopathological Dataset for Colorectal Polyps Classification and Adenoma Dysplasia Grading". In: *2021 IEEE International Conference on Image Processing (ICIP)*. 2021, pp. 76–80. DOI: 10.1109/ICIP42928.2021.9506198.

[2] Luca Bertero et al. *UNITOPATHO*. IEEE Dataport, 2021. DOI: 10.21227/9fsv-tm25.

[3] Umberto Gava et al. *UniToBrain Dataset*. Version V1.4. Zenodo, June 2021. DOI: 10.5281/zenodo.5109415.

[4] Dana Oniga et al. "Applications of AI and HPC in Health Domain". In: *HPC, Big Data, AI Convergence Toward Exascale: Challenge and Vision*. CRC Press, Taylor & Francis Group, 2021. Chap. 11. ISBN: 9781032009841.

[5] Daniele Perlo et al. "Dysplasia Grading of Colorectal Polyps through Convolutional Neural Network Analysis of Whole Slide Images". In: *Proceedings of 2021 International Conference on Medical Imaging and Computer-Aided Diagnosis (MICAD 2021)*. 2021. DOI: 10.1007/978-981-16-3880-0_34.

[6] Enzo Tartaglione, Daniele Perlo, and Marco Grangetto. "Post-synaptic Potential Regularization Has Potential". In: *Artificial Neural Networks and Machine Learning - ICANN 2019*. Vol. 11728. Springer, 2019, pp. 187–200. DOI: 10.1007/978-3-030-30484-3_16.

Bibliography

[7] Gregory W. Albers et al. "Thrombectomy for Stroke at 6 to 16 Hours with Selection by Perfusion Imaging". In: *New England Journal of Medicine* 378.8 (Feb. 2018), pp. 708–718. DOI: 10.1056/nejmoa1713973.

[8] Marco Aldinucci et al. "HPC4AI". In: *Proceedings of the 15th ACM International Conference on Computing Frontiers*. ACM, May 2018. DOI: 10.1145/3203217.3205340.

[9] M. A. Del Beccaro et al. "Computerized Provider Order Entry Implementation: No Association With Increased Mortality Rates in an Intensive Care Unit". In: *PEDIATRICS* 118.1 (July 2006), pp. 290–295. DOI: 10.1542/peds.2006-0367.

[10] Edwin Bennink et al. "Fast nonlinear regression method for CT brain perfusion analysis". In: *Journal of Medical Imaging* 3.2 (June 2016), p. 026003. DOI: 10.1117/1.jmi.3.2.026003.

[11] Jason Brownlee. *How to Avoid Overfitting in Deep Learning Neural Networks*. 2018. URL: https://machinelearningmastery.com.

[12] Dmitrii Bychkov et al. "Deep learning based tissue analysis predicts outcome in colorectal cancer". In: *Scientific Reports - Nature* 8 (2018). DOI: 10.1038/s41598-018-21758-3.

[13] Michael F Byrne et al. "Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of stan-

dard colonoscopy using a deep learning model". In: *Gut* 68.1 (2019), pp. 94–100. ISSN: 0017-5749. DOI: 10.1136/gutjnl-2017-314547.

[14] Bruce C.V. Campbell et al. "Imaging Selection in Ischemic Stroke: Feasibility of Automated CT-Perfusion Analysis". In: *International Journal of Stroke* 10.1 (Oct. 2014), pp. 51–54. DOI: 10.1111/ijs.12381.

[15] Michele Cancilla et al. "The DeepHealth Toolkit: A Unified Framework to Boost Biomedical Applications". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2021, pp. 9881–9888. DOI: 10.1109/ICPR48806.2021.9411954.

[16] Pratik Chaudhari et al. "Entropy-sgd: Biasing gradient descent into wide valleys". In: (2016). arXiv: 1611.01838.

[17] European Commission. *Communication Artificial Intelligence for Europe*. 2018. URL: https://ec.europa.eu/digital-single-market/en/news/communication-artificial-intelligence-europe.

[18] European Commission. *Redesigning health in Europe for 2020*. 2012. URL: https://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=2650.

[19] Angel Alfonso Cruz-Roa et al. "A Deep Learning Architecture for Image Representation, Visual Interpretability and Automated Basal-Cell Carcinoma Cancer Detection". In: *MICCAI (2)*. Vol. 8150. Lecture Notes in Computer Science. Springer, 2013, pp. 403–410. DOI: 10.1007/978-3-642-40763-5_50.

[20] E. D. Cubuk et al. "AutoAugment: Learning Augmentation Strategies From Data". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 113–123.

[21] Thomas Davenport and Ravi Kalakota. "The potential for artificial intelligence in healthcare". In: *Future healthcare journal* 6.2 (June 2019), pp. 94–98. DOI: 10.7861/futurehosp.6-2-94.

[22] DeepHealth. *Deep-Learning and HPC to Boost Biomedical Applications for Health*. 2019. URL: https://deephealth-project.eu/.

[23] Bernard Denis et al. "Diagnostic accuracy of community pathologists in the interpretation of colorectal polyps". In: *European journal of gastroenterology & hepatology* 21.10 (2009), pp. 1153–1160.

[24] R. C. Deo. "Machine learning in medicine." In: *Circulation* 132 (2015), pp. 1920–1930.

[25] David Deutsch. *Philosophy will be the key that unlocks artificial intelligence*. URL: https://www.theguardian.com/science/2012/oct/03/philosophy-artificial-intelligence.

[26] Joseph Donahue and Max Wintermark. "Perfusion CT and acute stroke imaging: Foundations, applications, and literature review". In: *Journal of Neuroradiology* 42.1 (Feb. 2015), pp. 21–29. DOI: 10.1016/j.neurad.2014.11.003.

[27] Mehmet Günhan Ertosun and Daniel L. Rubin. "Automated Grading of Gliomas using Deep Learning in Digital Pathology Images: A modular approach with ensemble of convolutional neural networks". In: *AMIA*. AMIA, 2015.

[28] European Community. *European Community Health-Care Statistics*. 2020. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php/Healthcare_expenditure_statistics#Healthcare_expenditure.

[29] Thorsten Falk et al. "U-Net: deep learning for cell counting, detection, and morphometry". In: *Nature Methods* 16.1 (Dec. 2018), pp. 67–70. DOI: 10.1038/s41592-018-0261-2.

[30] Umberto A. Gava et al. *Neural Network-derived perfusion maps: a Model-free approach to computed tomography perfusion in patients with acute ischemic stroke.* 2021. DOI: https://doi.org/10.1101/2021.01.13.21249757.

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[32] Varun Gulshan et al. "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs". In: *JAMA* (2016).

[33] M.N. Gurcan et al. "Histopathological Image Analysis: A Review". In: *IEEE Reviews in Biomedical Engineering* 2 (2009), pp. 147–171. DOI: 10.1109/rbme.2009.2034865.

[34] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CVPR*. IEEE Computer Society, 2016, pp. 770–778.

[35] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[36] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[37] Syed A. Hoda and Rana S. Hoda. "Rubin's Pathology: Clinicopathologic Foundations of Medicine, 5th Edition". In: *JAMA* 298.17 (Nov. 2007), pp. 2070–2075. ISSN: 0098-7484.

[38] National Cancer Institute. *NCI Dictionary of Cancer Terms - Dysplasia*. URL: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/dysplasia.

[39] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: (2015). arXiv: 1502.03167.

[40] Madabhushi A Janowczyk A. "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases". In: *Journal of pathology informatics* (2016), pp. 7–29. DOI: 10.4103/2153-3539.186902.

[41] Saurabh Jha and Eric J. Topol. "Adapting to Artificial Intelligence: Radiologists and Pathologists as Information Specialists". In: *JAMA* 316.22 (2016), pp. 2353–2354. DOI: 10.1001/jama.2016.17438.

[42] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. *100,000 histological images of human colorectal cancer and healthy tissue*. Version v0.1. Apr. 2018. DOI: 10.5281/zenodo.1214456.

[43] Jakob Nikolas Kather et al. "Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study". In: *PLoS Medicine* 16 (2019), pp. 445–54. DOI: 10.1371/journal.pmed.1002730.

[44] Kenji Kawaguchi. "Deep learning without poor local minima". In: *Advances in neural information processing systems*. 2016, pp. 586–594. DOI: 10.5555/3157096.3157162.

[45] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: (2014). arXiv: 1412.6980.

[46] S. Klein et al. “elastix: A Toolbox for Intensity-Based Medical Image Registration”. In: *IEEE Transactions on Medical Imaging* 29.1 (Jan. 2010), pp. 196–205. DOI: 10.1109/tmi.2009.2035616.

[47] Bruno Korbar et al. “Deep learning for classification of colorectal polyps on whole-slide images”. In: *Journal of pathology informatics* 8 (2017).

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *NIPS*. 2012, pp. 1106–1114.

[49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Commun. ACM* 60.6 (2017), pp. 84–90.

[50] Kohsuke Kudo et al. “Differences in CT Perfusion Maps Generated by Different Commercial Software: Quantitative Analysis by Using Identical Source Data of Acute Stroke Patients”. In: *Radiology* 254.1 (Jan. 2010), pp. 200–209. DOI: 10.1148/radiol.254082000.

[51] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[52] Henry W Lin, Max Tegmark, and David Rolnick. “Why does deep and cheap learning work so well?” In: *Journal of Statistical Physics* 168.6 (2017), pp. 1223–1247. DOI: 10.1007/s10955-017-1836-5.

[53] C. A. Longhurst et al. “Decrease in Hospital-wide Mortality Rate After Implementation of a Commercially Sold Computerized Physician Order Entry System”. In: *PEDIATRICS* 126.1 (May 2010), pp. 14–21. DOI: 10.1542/peds.2009-3271.

[54] Thomas KL Lui, Chuan-Guo Guo, and Wai K Leung. “Accuracy of artificial intelligence on histology prediction and detection of colorectal polyps: a systematic

review and meta-analysis". In: *Gastrointestinal Endoscopy* (2020). DOI: 10.1016/j.gie.2020.02.033.

[55] Marc Macenko et al. "A method for normalizing histology slides for quantitative analysis". In: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2009, pp. 1107–1110. DOI: 10.1109/ISBI.2009.5193250.

[56] Dwarikanath Mahapatra et al. "Structure Preserving Stain Normalization of Histopathology Images Using Self Supervised Semantic Guidance". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Springer International Publishing, 2020, pp. 309–319. DOI: 10.1007/978-3-030-59722-1_30.

[57] Christopher Malon and Eric Cosatto. "Classification of mitotic figures with convolutional neural networks and seeded blob features". In: *Journal of pathology informatics* (2013). DOI: 10.4103/2153-3539.112694.

[58] Reza Moradi, Reza Berangi, and Behrouz Minaei. "A survey of regularization strategies for deep models". In: *Artif. Intell. Rev.* 53.6 (2020), pp. 3947–3986.

[59] Nikhil Naik et al. "Deep learning-enabled breast cancer hormonal receptor status determination from base-level H&E stains". In: *Nature Communications* 11.1 (Nov. 2020). DOI: 10.1038/s41467-020-19334-3.

[60] Ziad Obermeyer and Ezekiel J. Emanuel. "Predicting the Future — Big Data, Machine Learning, and Clinical Medicine". In: *New England Journal of Medicine* 375.13 (Sept. 2016), pp. 1216–1219. DOI: 10.1056/nejmp1606181.

[61] *Open Computer Vision Library*. URL: https://opencv.org/.

[62] *Open Neural Network Exchange. The open standard for machine learning interoperability*. URL: https://onnx.ai/.

[63] R. D. Reed and R. J. Marks II. *Neural Smithing. Supervised Learning in Feedforward Artificial Neural Networks (A Bradford Book)*. MIT Press, 1999.

[64] Lancos Resampling. In: URL: https://en.wikipedia.org/wiki/Lanczos_resampling.

[65] Constantino Carlos Reyes-Aldasoro et al., eds. *Digital Pathology - 15th European Congress, ECDP*. Vol. 11435. Lecture Notes in Computer Science. Springer, 2019. ISBN: 978-3-030-23936-7. DOI: 10.1007/978-3-030-23937-4.

[66] Stanley J. Robboy et al. "Pathologist Workforce in the United States: I. Development of a Predictive Model to Examine Factors Influencing Supply". In: *Archives of Pathology & Laboratory Medicine* (Dec. 2013). DOI: 10.5858/arpa.2013-0200-OA.

[67] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[68] Swetlitz I. Ross C. *IBM pitched its Watson supercomputer as a revolution in cancer care. It's nowhere close.* 2017. URL: www.statnews.com/2017/09/05/watson-ibm-cancer.

[69] Santanu Roy et al. "A study about color normalization methods for histopathology images". In: *Micron* 114 (2018), pp. 42–61. ISSN: 0968-4328. DOI: 10.1016/j.micron.2018.07.005.

[70] Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach, Third International Edition*. Pearson Education, 2010. ISBN: 978-0-13-207148-2.

[71] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4510–4520. DOI: 10.1109/cvpr.2018.00474.

[72] Kenneth A. Schafer et al. "Use of Severity Grades to Characterize Histopathologic Changes". In: *Toxicologic Pathology* 46.3 (Mar. 2018), pp. 256–265. DOI: 10.1177/0192623318761348.

[73] John R. Searle. "Minds, brains, and programs". In: *Behavioral and Brain Sciences* 3.3 (1980), pp. 417–424. DOI: 10.1017/S0140525X00005756.

[74] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *ICCV*. IEEE Computer Society, 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

[75] Younghak Shin and Ilangko Balasingham. "Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification". In: *IEEE Engineering in Medicine and Biology Society (EMBC)*. 2017, pp. 3277–3280. DOI: 10.1109/EMBC.2017.8037556.

[76] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: (2014). arXiv: 1409.1556.

[77] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: (2014). arXiv: 1409.1556.

[78] Korsuk Sirinukunwattana, David R. J. Snead, and Nasir M. Rajpoot. "A Stochastic Polygons Model for Glandular Structures in Colon Histology Images". In: *IEEE Trans. Med. Imaging* 34.11 (2015), pp. 2366–2378.

[79] Korsuk Sirinukunwattana et al. "Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images". In: *IEEE Trans. Med. Imaging* 35.5 (2016), pp. 1196–1206.

[80] Zhigang Song et al. "Automatic deep learning-based colorectal adenoma detection system and its similarities with pathologists". In: *BMJ open* 10.9 (2020), e036423. DOI: 10.1136/bmjopen-2019-036423.

[81] Jost Tobias Springenberg et al. "Striving for simplicity: The all convolutional net". In: (2014). arXiv: 1412.6806.

[82] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[83] Andreas Stang et al. "Diagnostic agreement in the histopathological evaluation of lung cancer tissue in a population-based case-control study". In: *Lung Cancer* (Apr. 2006). DOI: 10.1016/j.lungcan.2005.11.012.

[84] Manuel Stritt et al. "Supervised Machine Learning Methods for Quantification of Pulmonary Fibrosis". In: *MDA*. ibai-publishing, 2011, pp. 24–37.

[85] Mehran Taherian et al. *Tubular Adenoma*. StatPearls Publishing, Treasure Island (FL), 2020. URL: http://europepmc.org/books/NBK553180.

[86] Enzo Tartaglione et al. "Learning sparse neural networks via sensitivity-driven regularization". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. 2018, pp. 3882–3892. DOI: 10.5555/3327144.3327303.

[87] Enzo Tartaglione et al. "SeReNe: Sensitivity-Based Regularization of Neurons for Structured Sparsity in Neural Networks". In: *IEEE Transactions on Neural*

Networks and Learning Systems (2021), pp. 1–14. DOI: 10.1109/tnnls.2021.3084527.

[88] J. Tseung. "Robbins and Cotran Pathologic Basis of Disease: 7th Edition". In: *Pathology* 37 (2005), p. 190.

[89] Mehtat Unlu et al. "Molecular Characteristics of Colorectal Hyperplastic Polyp Subgroups". In: *The Turkish Journal of Gastroenterology* 31 (May 2020). DOI: 10.5152/tjg.2020.19322.

[90] Stefan Wager, Sida Wang, and Percy S Liang. "Dropout training as adaptive regularization". In: *Advances in neural information processing systems*. 2013, pp. 351–359.

[91] Robert Wannamaker et al. "Computed Tomographic Perfusion Predicts Poor Outcomes in a Randomized Trial of Endovascular Therapy". In: *Stroke* 49.6 (June 2018), pp. 1426–1433. DOI: 10.1161/strokeaha.117.019806.

[92] Jason W Wei et al. "Evaluation of a Deep Neural Network for Automated Classification of Colorectal Polyps on Histopathologic Slides". In: *JAMA Network Open* 3.4 (2020). DOI: 10.1001/jamanetworkopen.2020.3398.

[93] Jason W. Wei et al. "Pathologist-level classification of histologic patterns on resected lung adenocarcinoma slides with deep neural networks". In: (2019). arXiv: 1901.11489.

[94] Kristoffer Wickstrøm, Michael Kampffmeyer, and Robert Jenssen. "Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps". In: *Medical Image Anal.* 60 (2020). DOI: 10.1016/j.media.2019.101619.

[95] Gerard Jacques van Wyk and Anna Sergeevna Bosman. "Evolutionary neural architecture search for image restoration". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[96] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: cs.LG/1708.07747 [cs.LG].

[97] Yuanpu Xie et al. "Deep Voting: A Robust Approach Toward Nucleus Localization in Microscopy Images". In: *MICCAI (3)*. Vol. 9351. Lecture Notes in Computer Science. Springer, 2015, pp. 374–382.

[98] Kun-Hsing Yu, Andrew L. Beam, and Isaac S. Kohane. "Artificial intelligence in healthcare". In: *Nature Biomedical Engineering* (2018), pp. 719–731.

[99] Kun-Hsing Yu et al. "Association of Omics Features with Histopathology Patterns in Lung Adenocarcinoma". In: *Cell Systems* (Dec. 2017).

[100] Kun-Hsing Yu et al. "Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features". In: *Nature Communications* 7.1 (Aug. 2016). DOI: 10.1038/ncomms12474.

[101] Ruikai Zhang et al. "Automatic Detection and Classification of Colorectal Polyps by Transferring Low-Level CNN Features From Nonmedical Domain". In: *IEEE J. Biomed. Health Informatics* 21.1 (2017), pp. 41–47. DOI: 10.1109/JBHI.2016.2635662.