UNIVERSITÀ DEGLI STUDI DI TRIESTE

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXXV CICLO DEL DOTTORATO DI RICERCA
IN INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Applications of deep learning-based methods in medical image analysis

Settore scientifico-disciplinare: SSD ING-INF/06 Bioingegneria Elettronica ed Informatica

DOTTORANDA
Teresa Pace

COORDINATORE
Prof. Alberto Tessarolo

SUPERVISORE DI TESI
Prof. Agostino Accardo

ANNO ACCADEMICO 2021/2022

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence

AMD Age-related macular degeneration

ANN Artificial Neural Network

ASUGI Azienda Sanitaria Universitaria Giuliano Isontina

AUC Area Under the ROC curve

BCVA Best-corrected visual acuity

BM Bruch membrane

CNN Convolutional Neural Network

CNV Choroidal neovascularization

CV Cross-validation

DL Deep Learning

DME Diabetic macular edema

DSM Dome-shaped macula

FN False negative

FP False positive

FPR False Positive Rate

FTMH Full thickness macular hole

GAP Global Average Pooling

H&E Hematoxylin and Eosin

HTL Health Telematics Laboratory

ILM Internal limiting membrane

I-LMH Inner lamellar macular hole

ILSVRC ImageNet Large Scale Visual Recognition Challenge

I-MS Inner macular schisis

IO-MS Inner and outer macular schisis

LRN Local Response Normalization

MB Macular buckle

MCT Macula choroidal thinning

MD Macular detachment

ML Machine Learning

MS Macular schisis

MSS MTM Staging System

MTM Myopic Traction Maculopathy

NAFLD Non-Alcoholic Fatty Liver Disease

NAS NAFLD Activity Score

NASH Non-Alcoholic Steatohepatitis

OCT Optical Coherence Tomography

O-LMH Outer lamellar macular hole

OMCNS Pathological myopic choroidal neovascularization

O-MS Outer macular schisis

OvR One-vs-Rest

PoC Proof of concept

PPV Pars plana vitrectomy

QA Quality Assessment

ReLU Rectifier Linear Unit

ROC curve Receiver Operating Characteristic curve

SGD Stochastic Gradient Descend

TN True negative

TP True positive

TPR True Positive Rate

WSI Whole Slide Image

XAI Explainable AI

Abstract

The focus of the research activity carried out during the Ph.D. course with higher education and research apprenticeship program between O3 Enterprise s.r.l. and the University of Trieste was the application of deep learning to the analysis of medical images. In this thesis, two novel methods for addressing tasks impacting clinical practice in two different fields of medical imaging are presented.

The first is the result of the study conducted in collaboration with the Surgical Pathology Unit of Cattinara Hospital and the Health Telematics Laboratory (HTL) of the Complex Structure of Informatics and Telecommunications at Azienda Sanitaria Universitaria Giuliana Isontina (ASUGI). In this study both classical imaging techniques and deep learning methods were applied to the quality assessment of histological images in liver biopsies, reproducing in a fully automatic and objective way what must be still manually done in clinical practice by pathologists before any histological image analysis performed by a computer-aided system. The automatic quality assessment of histological images represents a difficult task to perform due to the complexity and heterogeneity of the elements that may alter the quality of the images during the process of slide preparation. For this reason, as things stand, this issue has not been fully

addressed in literature. This study aims to investigate a method that can consider all elements that may influence the quality of histological images. The proposed method reached promising results in terms of accuracy (96,88%), sensitivity (95.31%), specificity (98.44%), and AUC (98,94%), especially when compared with the results currently offered by literature, and it could be assumed to have a useful application in speeding up the histological image analysis process.

The second study, born from the collaboration between O3Enterprise s.r.l. and Dr. Parolini, currently Director of the Vitreoretinal Service at Eyecare Clinic in Brescia, applied and compared different deep learning models for the development of an AI system able to stage the myopic traction maculopathy (MTM) – i.e. complex disease characterized by a wide spectrum of clinical pictures that may affect eyes with high myopia – according to the recently proposed MTM staging system. This study applied for the first time deep learning-based methods to the analysis of MTM, whose assessment still represented a diagnostic challenge even for experienced ophthalmologists, showing great potential in differentiating the various stages that characterize the evolution of this disease. The good performances achieved in this study (AUCs: 94.73% - 99.51%) demonstrated the feasibility of the project that aims to help and train ophthalmologists in the staging and management of such a complex disease.

Both studies, despite some limitations mainly due to the difficulty of collecting a large number of data, have demonstrated the effectiveness of applying deep learning-based methods even to complex tasks that have never been addressed before in research activities, laying the foundations for further developments and investigations and offering insights for further research.

Sommario

Il focus dell'attività di ricerca svolta durante il corso di dottorato con un programma di apprendistato di alta formazione e ricerca tra O3 Enterprise s.r.l. e l'Università di Trieste è stato l'applicazione del deep-learning all'analisi di immagini mediche. In questa tesi vengono presentati due nuovi metodi che sono stati utilizzati per svolgere compiti aventi un importante impatto sulla pratica clinica in due diversi campi dell'imaging medico.

Il primo è il risultato dello studio condotto in collaborazione con la Struttura Complessa di Anatomia e Istologia Patologica dell'Ospedale di Cattinara e il Laboratorio di Telematica Sanitaria (HTL) della Struttura Complessa di Informatica e Telecomunicazioni dell'Azienda Sanitaria Universitaria Giuliana Isontina (ASUGI). In questo studio sia le tecniche classiche di imaging che i metodi di deep learning sono stati applicati alla valutazione della qualità delle immagini istologiche delle biopsie epatiche, riproducendo in modo completamente automatico e oggettivo ciò che nella pratica clinica deve essere ancora fatto manualmente dai patologi prima di una qualsiasi analisi delle immagini istologiche eseguita da un sistema computerizzato. La valutazione automatica della qualità delle immagini istologiche rappresenta un compito difficile da svolgere a causa della complessità e dell'eterogeneità degli elementi che possono

alterare la qualità delle immagini durante il processo di preparazione dei vetrini. Per questo motivo, allo stato attuale, questo compito non è stato ancora completamente affrontato in letteratura. Questo studio si propone di valutare un metodo in grado di considerare tutti gli elementi che possono influenzare la qualità delle immagini istologiche. Il metodo proposto ha raggiunto risultati promettenti in termini di accuratezza (96.88%), sensibilità (95.31%), specificità (98.44%) e AUC (98.,94%), soprattutto se confrontato con i risultati attualmente offerti dalla letteratura, e si può ipotizzare possa avere un'utile applicazione nel velocizzare il processo di analisi delle immagini istologiche.

Il secondo studio, nato dalla collaborazione tra O3Enterprise s.r.l. e la Dr.ssa Parolini, attualmente Direttrice dell'Unità Vitreoretinica della Clinica Oculistica di Brescia, ha interessato l'applicazione e il confronto di diversi modelli di deep learning per lo sviluppo di un sistema di intelligenza artificiale in grado di effettuare la stadiazione della maculopatia miopica trattiva (MTM), una patologia complessa caratterizzata da un ampio spettro di quadri clinici che può colpire occhi con elevata miopia, secondo il sistema di stadiazione MTM recentemente proposto. Questo studio ha utilizzato per la prima volta dei metodi basati sul deep learning per l'analisi della MTM, la cui valutazione rappresenta ancora una sfida diagnostica anche per gli oftalmologi più esperti, mostrando un grande potenziale nel differenziare i vari stadi che caratterizzano l'evoluzione di questa malattia. Le buone prestazioni ottenute in questo studio (AUCs: 94,73% - 99,51%) hanno dimostrato la fattibilità del progetto che mira a formare ed aiutare gli oftalmologi nella stadiazione e nella gestione di una malattia così complessa.

Entrambi gli studi, nonostante alcune limitazioni dovute principalmente alla difficoltà di raccogliere un gran numero di dati, hanno dimostrato l'efficacia dell'applicazione di metodi basati sul deep learning anche a compiti complessi mai affrontati prima nelle attività di ricerca, ponendo le basi per ulteriori sviluppi e indagini e offrendo spunti per ulteriori ricerche.

Introduction

In the past few decades, the increased role of medical imaging within the diagnostic process, together with the progress in medical imaging technology, has given rise to the need for the development of novel computational methods able to analyze large volumes of data and improve efficiency, compliance, and interpretative accuracy. During this time, several image analysis systems have been developed, using methods based firstly on heuristics techniques, then on manual, handcrafted features extraction techniques, and finally on supervised deep learning techniques.

The widespread use of supervised deep learning-based methods in image recognition came about after the convolutional neural network (CNN) proposed by Krizhevsky *et al.* [1] won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Since then, CNNs have become the most used architecture for image analysis, achieving even higher performance than humans. The advantage of CNNs is represented by their ability to automatically and adaptively learn spatial hierarchies of features, from low-level patterns, such as lines or edges, to higher-level patterns, such as shape, mimicking the way animal visual cortices process visual information and recognize objects. Today, CNNs are considered to represent the state of the art in image analysis [2].

Comprehensive academic research is working on finding deep learning solutions that can be applicable to the medical world. This mainly concerns the field of radiology but is spreading also to other image-centric specialties, such as pathology and ophthalmology.

In this thesis, two novel applications of deep learning methods to medical images are presented as the results of the research activity, performed during the Ph.D. course with higher education and research apprenticeship program between O3 Enterprise s.r.l. and the University of Trieste.

The first application was carried out in collaboration with the Surgical Pathology Unit of Cattinara Hospital and the Health Telematics Laboratory (HTL) of the Complex Structure of Informatics and Telecommunications at Azienda Sanitaria Universitaria Giuliana Isontina (ASUGI). It had at its goal the development of a deep learning-based method to automatically assess the quality of histological images in liver biopsies as the first stage of a computer-aided system for the analysis of whole slide images (WSIs). There is a growing interest in the development of computer-aided systems for automated examinations of histological images. However, the robustness of these systems can only be ensured if the image quality is first verified. Currently, this task is manually performed by a pathologist who selects the regions of interest that meet the quality standard for further automated analysis. A system capable to select accurately and objectively good-quality image regions from histological images is therefore needed, although until now it has not yet been implemented mainly due to the high complexity and heterogeneity of the elements that might alter the quality of histological images.

The second application was carried out in collaboration with O3 Enterprise s.r.l. and Dr. Parolini, currently Director of the Vitreoretinal Service at Eyecare Clinic in Brescia and aims to study for the very first time the feasibility of deep learning methods in staging the myopic traction maculopathy (MTM) from optical coherence tomography (OCT) according to the recently proposed MTM

staging system (MSS). MTM is a complex disease affecting approximately 30% of eyes with pathological myopia whose pathogenesis, natural evolution, and prognosis are still not entirely known. An effort in this regard has been made by the MSS, which defined the changes seen in MTM not just as different types of clinical pictures, but as different stages of one evolving disease. The development of an AI system able to automatically and accurately analyze OCT images to assess the stage of an eye with MTM is thought to have important educational and clinical implications in spreading knowledge of this complex disease and supporting patient management.

This thesis is structured as follows: in the first chapter, an overview of the principles at the basis of CNNs and the most commonly used architectures for medical image analysis are provided, while the second and the third chapters describe the steps followed in the development of a deep-learning based method for the quality assessment of histological images and the development of an AI system for staging MTM according to MSS, respectively, and the results achieved.

Chapter 1
Convolutional Neural Networks

1.1 Artificial Neural Networks

Artificial Intelligence (AI) was born in the 1950s when some pioneers of the field of computer science started to wonder if computers could be made to "think". AI can be defined as the effort to automate intellectual tasks normally performed by humans. *Machine learning* (ML) is a subfield of Artificial Intelligence that focuses on the development of algorithms that use statistical methods to automatically learn and improve from experience without being explicitly programmed. *Deep learning* (DL) is a subset of machine learning in which algorithms with a brain-like logical structure learn from vast amounts of data. The main difference between classical ML and DL is that the former requires the manual selection of image features that best represent the data, while in the latter the features that are best for carrying out the computational task are learned automatically by the algorithms.

Most deep learning algorithms are based on *Artificial Neural Networks* (ANNs), computational processing systems inspired by the way biological

nervous systems operate. An ANN is a collection of connected units called *artificial neurons*, which model the neurons in a biological brain.

Biological neurons typically consist of three parts: dendrites, a cell body (or soma), and an axon. The dendrites receive input signals, the cell body performs a summation function and if the final sum is above a certain threshold, the neurons output an action potential sending a spike along their axon. In most synapses, signals flow from the axon of one neuron to a dendrite of another. An artificial neuron models this behavior receiving one or more inputs and computing its output by a non-linear function of the weighted sum of its inputs, plus a bias (Figure 1.1). Each input has a weight that adjusts during learning, increasing or decreasing the strength of a connection. A bias is used for shifting the activation function towards left or right.

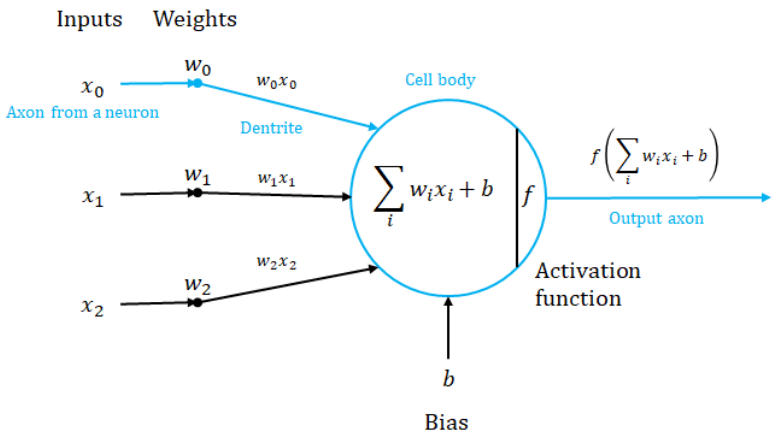


Figure 1.1 – Model of an artificial neuron.

In an ANN, artificial neurons are aggregated into layers: the signal goes from the input layer to the output layer, after traversing one or more hidden layers (Figure 1.2). Different layers may perform different kinds of transformations on their inputs.

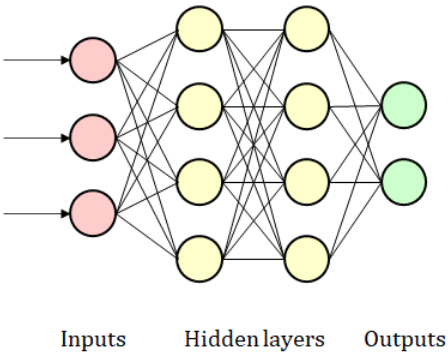


Figure 1.2 – Schematic representation of an ANN.

The most popular ANNs used for both regular and medical image analysis problems are Convolutional Neural Networks.

1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are artificial neural networks inspired by the organization of animal visual cortex which are specifically designed for image analysis to learn representations of relevant features automatically and adaptively.

Since the great results achieved in 2012 at the ImageNet Large Scale Visual Recognition Competition (ILSVRC), CNNs have been dominant in computer vision tasks, forming the basis for some of the most influential innovations in this field [1]. Medical imaging is no exception, as CNNs have obtained, or even exceeded, expert-level performances in various image-centric specialties, such as radiology, dermatology, pathology, and ophthalmology.

The typical CNN architecture is composed of three types of layers (also known as *building blocks*): convolutional, pooling and fully connected layers. More specifically, a CNN usually consists of a repetition of a stack of several convolutional layers and a pooling layer, followed by one or more fully connected layers.

1.2.1 Convolutional layers

The *convolutional layer* is the major building block of a CNN. It contains a set of filters, or *kernels*, whose parameters are automatically learned throughout the training process.

Each kernel slides over the input and computes the dot product between the weights of the kernel and the value of the input to generate a two-dimensional map, called a feature map (Figure 1.3). The output volume of the convolutional layers is a stack of these feature maps for all kernels along the depth dimension.

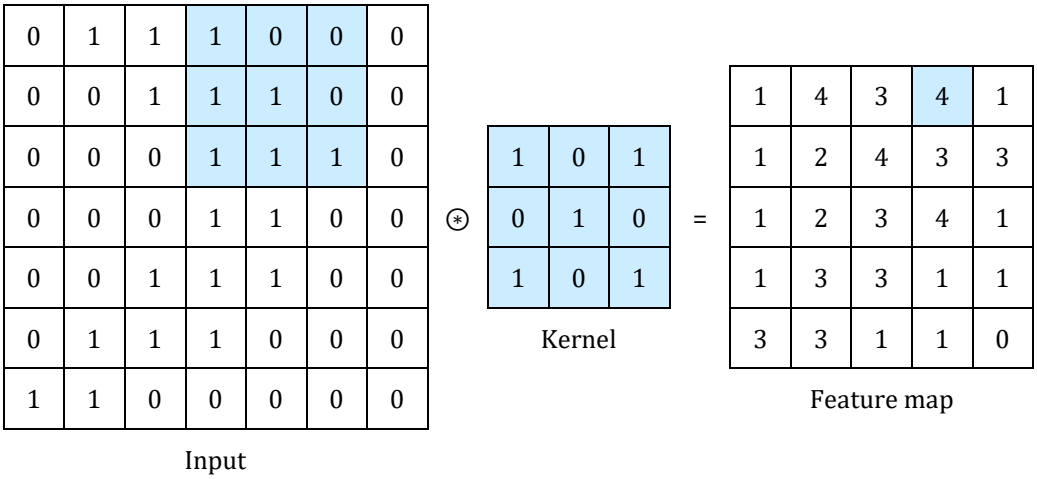


Figure 1.3 – Example of convolution operation with a kernel of size 3 × 3, stride 1, and no padding.

The convolution operation between the input image $I(i,j)$ and the kernel $K(i,j)$ to produce the feature map $F(i,j)$ can be expressed as:

$$F(i,j) = I(i,j) \circledast K(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \qquad 1.1$$

Kernels are designed to be smaller than the input so that the network learns filters which maximally respond to a local region of the input volume. The spatial extent of this local connectivity represents a hyperparameter also called receptive field. The extent of the connectivity along the depth axis is always equal

to the depth of the input volume. Kernels size is typically 3×3, 5×5 or 7×7 while the *depth* is arbitrary and corresponds to the number of kernels. Other hypermeters are the *stride*, which is the distance between two consecutive kernel positions and commonly is set to 1, and the *zero-padding*, that consists in padding the input volume with zeros around the edges (Table 1.1).

Layer	**Parameters**	**Hyperparameters**
Convolutional layer	Kernel weights	Number of kernels, kernels size, stride, padding, activation function
Pooling layer	None	Pooling method, filter size, stride, padding
Fully connected layer	Weights	Number of weights, activation function
Other		Model architecture, optimizer, learning rate, loss function, mini-batch size, epochs, regularization, weight initialization, dataset splitting

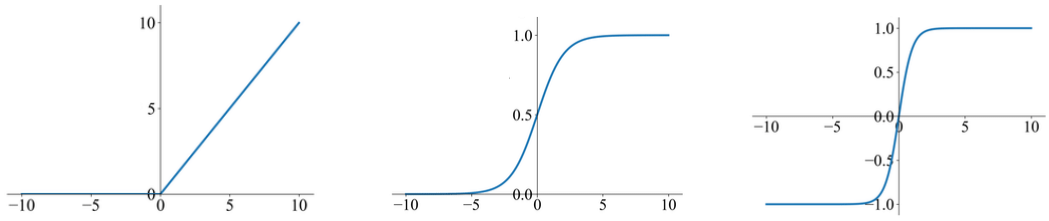
Table 1.1 – Parameters and hyperparameters in a CNN. A parameter is a variable that is automatically learned during the training process, whereas a hyperparameter is a variable that needs to be set before the training process.

The spatial size of the output volume is therefore a function of the input volume size (W), the size of the receptive files (F), the stride applied (S) and the amount of the zero-padding used (P) on the border:

$$OutputSize = \frac{W - F + 2P}{S} + 1 \qquad 1.2$$

The key feature of convolutional layers is weights sharing: kernels are shared across all the image positions, with the great advantage of reducing the number of learnable parameters and significantly improving the efficiency of training.

The output of a convolutional layer is passed through a non-linear activation function. The most commonly used activation functions are rectified linear unit (ReLU), sigmoid, and hyperbolic tangent (tanh) functions (Figure 1.4).



$f(x) = \max(0, x)$ (a)

$f(x) = \frac{1}{1 + e^{-x}}$ (b)

$f(x) = \tanh(x)$ (c)

Figure 1.4 – Activation functions commonly applied to CNNs: (a) rectified linear unit (ReLU), (b) sigmoid, (c) hyperolic tangent (tanh).

1.2.2 Pooling layers

The *pooling layer* down-samples feature maps to reduce the number of learnable parameters and introduce a translation invariance to small shifts and distortions. The down-sampling of feature maps creates a lower resolution version of the input that still contains the large or important structural elements, without the fine details that may not be useful to the task.

Several types of pooling methods are available. The most frequently utilized are max pooling, min pooling, average pooling, and global average pooling (GAP).

Max, min, and average pooling layers apply a filter to each input feature map and compute the maximum, minimum, and average value in each extracted patch, respectively. Pooling layers have no learnable parameters, whereas filter size, stride, and padding are hyperparameters to set before training. Max, min and average pooling commonly use a filter of size 2×2 and stride of 2, down-sampling height and width of the input feature maps by a factor of 2 (Figure 1.5).

Global average pooling, otherwise, performs an extreme type of down sampling computing the average of all the elements in each feature map. Height and width dimensions of the input feature maps are down-sampled into a 1×1 array while the depth is maintained. The GAP layer is typically used only once before the fully connected layers.

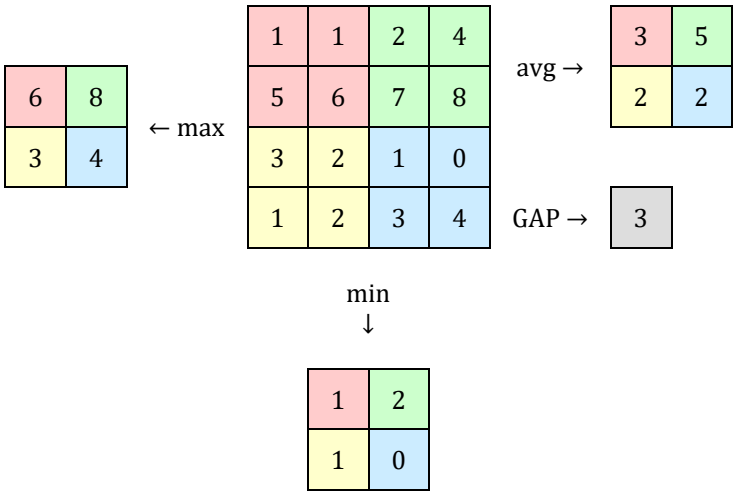


Figure 1.5 – Examples of max, min, and average pooling with a filter size of 2×2, no padding and stride 2, and global average pooling.

1.2.3 Fully Connected layers

The *fully connected layer*, also known as *dense layer*, is a layer in which every input is connected to every output by learnable parameters. In a CNN, the output feature maps of the final convolutional or pooling layer are flattened – i.e. transformed into a 1D vector – and passed to one or more fully connected layers. Like convolutional layers, each fully connected layer is followed by a nonlinear activation function, such as ReLU. The activation function of the last fully connected layer, which acts as the CNN classifier, is specific to the task to be performed (Table 1.2).

Task	**Last fully connected layer activation function**
Binary classification	Sigmoid
Single-label multiclass classification	Softmax
Multi-labels multiclass classification	Sigmoid
Regression to continuous values	Identity

Table 1.2 – Commonly activation functions applied to the last fully connected layer depending on the task.

In a binary classification task, a CNN commonly ends with a fully connected layer with one unit and *sigmoid* activation function so that its output is a scalar between 0 and 1, encoding a probability.

In a single-label multiclass classification problem, the last fully connected layer has the same number of output nodes as the number of classes and *softmax* activation function, which outputs a probability distribution over the output classes. The softmax activation function is mathematically represented by Equation 1.3, where K is the number of output classes.

$$f(x_k) = \frac{e^{x_k}}{\sum_{i=1}^{K} e^{x_i}}, \quad k = 1, \dots, K \tag{1.3}$$

1.2.4 Training a CNN

The training process consists in finding parameters (i.e. kernels and weights for convolutional and fully connected layers, respectively) which minimize the differences between output predictions and ground truth labels on the training dataset. Typically, this is addressed by calculating a model's performance under certain parameters with a loss function through *forward propagation* on the training dataset and then updating parameters according to the loss value through *back-propagation* with a gradient descent optimization algorithm.

1.2.4.1 Loss function

The *loss function* is an hyperparameter that needs to be defined according to the given task before the training process. The most used loss function for classification tasks is the *cross-entropy loss*, also referred as the *log loss function*. The mathematical representation of cross-entropy loss for multiclass classification tasks is given in Equation 1.4:

$$L = -\sum_{i=1}^{K} t_i \log(p_i) \tag{1.4}$$

where K is the number of classes, t_i is the truth label probability distribution and p is the softmax probability distribution for the i^{th} class.

In binary classification problems, where $K = 2$, the cross-entropy loss can be defined also as in Equation 1.5:

$$\begin{aligned} L &= -\sum_{i=1}^{2} t_i \log(p_i) \\ &= -[t_1 \log(p_1) + t_1 \log(p_1)\ ] \\ &= -[t \log(p) + (1-t) \log(1-p)\ ] \end{aligned} \tag{1.5}$$

1.2.4.2 Optimizers

The algorithms used to iteratively update the learnable parameters of the network in order to minimize the loss function are called optimizers.

The most used optimizer is the *gradient descent*, which computes a partial first-order derivative of the loss function with respect to each learnable parameter and updates each parameter in the negative direction of the gradient with an arbitrary step defined by a hyperparameter called *learning rate*.

A single update of a parameter is formulated as in Equation 1.6:

$$w \coloneqq w - \alpha \frac{\partial L}{\partial w} \tag{1.6}$$

where w stands for each learnable parameter, α is the learning rate, and L it the loss function. The parameter updating process is performed through network back-propagation, in which the gradient is back-propagated to the preceding layer.

Many alternatives of the gradient descend algorithm have been proposed, including the following:

- *Batch Gradient Descent*: updates parameters after computing the gradient of the loss function with respect to the entire training set.
- *Stochastic Gradient Descent* (SGD): updates parameters after computing the gradient of the loss function with respect to a single training example.
- *Mini-batch Gradient Descent*: updates parameters after computing the gradient of the loss function with respect to a subset of the training set.
- *Adaptive Moment Estimation* (Adam) [3]: extension to SGD method that calculates adaptive learning rate for each parameter through adaptive estimation of first-order and second-order moments. It represents the latest trend in deep learning optimization.

1.2.4.3 Regularization

One of the central issues in CNN training process is *overfitting*, which occurs in the cases where the model executes especially well on the training set but does not succeed on a new dataset, i.e. it does not generalize to unseen data. The opposite issue is referred as to *underfitting* and occurs when the model performs poorly on both the training and validation sets.

The best solution to avoid overfitting is to collect more training data, although this is hardly obtainable in the field of medical imaging. In this case, it is common to resort to the use of *data augmentation* techniques, which aim to increase the amount of available data by introducing random transformations, such as flipping, translation, rotation, and cropping, so that the model will not see the same inputs during training iterations.

Other solutions to minimize overfitting include regularization with dropout or weight decay, and batch normalization, as well as reducing the complexity of the CNN architecture:

- *Dropout*: it consists in randomly ignoring some number of layers outputs during each training epoch so that the model becomes less sensitive to specific weights in the network [4].
- *Weight decay*: also known as *L2 regularization*, it adds a penalty term to the loss function of the network so that the weights were shrinked during backpropagation (Equation 1.7):

$$finalLoss = loss + \lambda \|w\|^2 \qquad 1.7$$

  where λ is a hyperparameter of the model known as the regularization term and w is each learnable parameter.
- *Batch Normalization*: it consists in a supplemental layer which affects the output of the previous layer by subtracting the batch mean and dividing by the batch standard deviation [5].

1.2.4.4 Transfer Learning

Transfer Learning is a common and efficient strategy used to address the lack of training data. It involves the application of a model pre-trained on an extremely large dataset (such as the ImageNet dataset, which consists of 1.4 million images with 1,000 classes) to the specific task of interest.

In practice, there are two methods to perform transfer learning (Figure 1.6):

- *Fixed feature extraction*: the fully connected layers from the pre-trained model are replaced with a new set of fully connected layers, while its convolutional base – i.e. the stack of convolutional and pooling layers – is maintained. During the training process, only the new set of fully connected layers is retrained on the dataset of interest. This approach is rarely used in medical imaging research because of the dissimilarity between the dataset used to pre-train the model and the dataset consisting of medical images.

- *Fine-tuning*: the fully connected layers from the pre-trained model are replaced with a new set of fully connected layers but, in this case, during the training process also all the layers in the convolutional base will be fine-tuned on the new dataset. Alternatively, some earlier layers, which extract more generic features, can be fixed while the rest of the layers are fine-tuned on the data of interest.

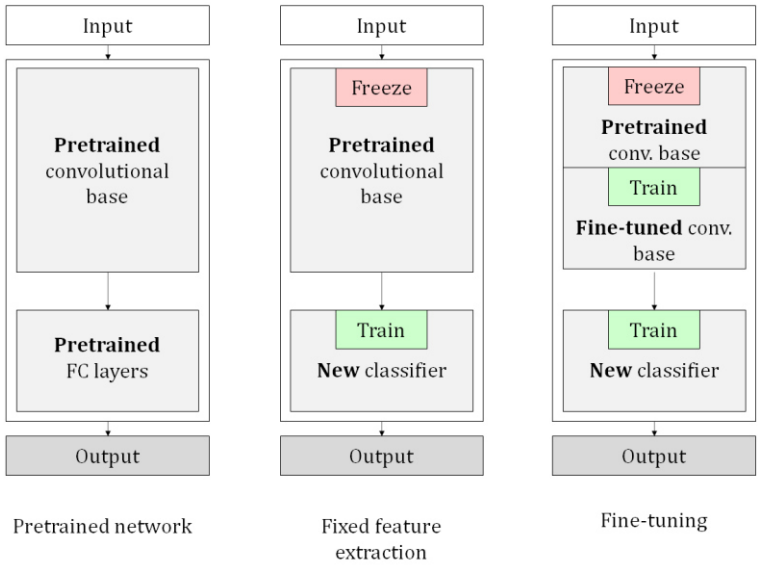


Figure 1.6 – Transfer learning methods.

1.2.4.5 Dataset splitting

Available data are typically split into three sets: a training, a validation, and a test sets (Figure 1.7a). The *training set* is used to train the network while the *validation set* is used to evaluate the model during the training process. The *test set* is used only once at the end of the project in order to evaluate the performance of the final model. The reason while separate validation and test sets are needed is to avoid information leakage. Since the hyperparameter tuning – i.e. the process of determining the right combination of hyperparameter that maximized the model performance – and the model selection are based on the performance

achieved on the validation set, some information about the validation set leaks into the model itself. Therefore, only the evaluation of the model performance on completely unseen dataset allows to verify actual generalizability of the model.

An issue that arises when the available data is split into three sets is the drastic reduction of the samples that can be used for training the model. This problem is particularly evident on small datasets, as is often the case in the field of medical imaging. Cross-validation (CV) is the approach commonly used to overcome to this problem. In k-*fold cross-validation* the dataset is split in a training and a test sets, and the training set is in turn partitioned into k subsets (Figure 1.7b). The model is trained using $k - 1$ of these subsets as training data and validated on the remaining subset. The process is iterated k times by rotating the training and validation subsets. Hyperparameter tuning and model selection are based on the average performance obtained in the validation subsets.

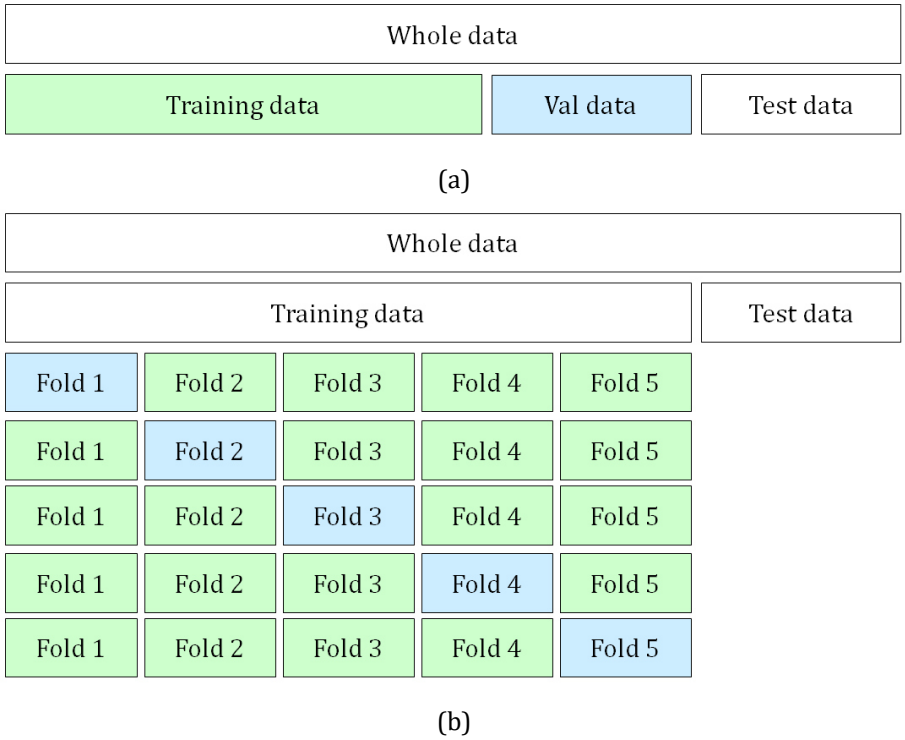


Figure 1.7 – Dataset splitting: (a) training, validation and test sets, (b) 5-fold cross-validation.

1.3 CNN architectures

Over the last 10 years, several CNN architectures have been presented, starting from the AlexNet model in 2012 and ending with the High-Resolution model in 2020.

Table 1.3 shows a brief overview of CNN architectures. CNN architectures differ for various factors but those that allowed significant improvements in performance were the reorganization of the processing unit, the development of novel building blocks, and the increment of the network depth.

Model	Main finding	Depth	Dataset	Error rate	Input size	Year
AlexNet	Utilizes Dropout and ReLU	8	ImageNet	16.4	227×227×3	2012
NIN	New layer, called mlpconv, utilizes GAP	3	CIFAR-10 CIFAR-100 MNIST	10.41 35.68 0.4	32×32×3	2013
ZfNet	Visualization idea of middle layers	8	ImageNet	11.7	224×224×3	2014
VGG	Increased depth, small filter size	16, 19	ImageNet	7.3	224×224×3	2014
GoogLeNet	Increased depth, block concept, different filter size, concatenation concept	22	ImageNet	6.7	224×224×3	2015
Inception-v3	Utilizes small filter size, better feature representation	48	ImageNet	3.5	229×229×3	2015
Highway	Presented the multipath concept	19, 32	CIFAR-10	7.76	32×32×3	2015
Inception-v4	Divided transform and integration concepts	70	ImageNet	3.08	229×229×3	2016
ResNet	Robust against overfitting due to symmetry mapping-based skip links	152	ImageNet	3.57	224×224×3	2016
Inception-ResNetv2	Introduced the concept of residual links	164	ImageNet	3.52	229×229×3	2016
FractalNet	Introduced the concept of Drop-Path as regularization	40, 80	CIFAR-10 CIFAR-100	4.60 18.85	32×32×3	2016
WideResNet	Decreased the depth and	28	CIFAR-10 CIFAR-100	3.89 18.85	32×32×3	2016

	increased the width					
Xception	A depthwise convolution followed by a pointwise convolution	71	ImageNet	0.055	229×229×3	2017
Residual attention neural network	Presented the attention technique	452	CIFAR-10 CIFAR-100	3.90 20.4	40×40×3	2017
Squeeze-and-excitation networks	Modeled interdependencies between channels	152	ImageNet	2.25	229×229×3 224×224×3 320×320×3	2017
DenseNet	Blocks of layers, layers connected to each other	201	CIFAR-10 CIFAR-100 ImageNet	3.46 17.18 5.54	224×224×3	2017
Competitive squeeze and excitation network	Both residual and identity mappings utilized to rescale the channel	152	CIFAR-10 CIFAR-100	3.58 18.47	32×32×3	2018
MobileNet-v2	Inverted residual structure	53	ImageNet	-	224×224×3	2018
CapsuleNet	Pays attention to special relationships between features	3	MNIST	0.00855	28×28×1	2018
HRNetV2	High-resolution representations	-	ImageNet	5.4	224×224×3	2020

Table 1.3 – Brief overview of CNN architectures [6]. The highlighted networks are the ones detailed in this paragraph.

The following subparagraphs present the CNN architectures evaluated in this research work, which are also the most widely used in literature for medical imaging analysis.

1.3.1 VGG

VGG is a convolutional neural network proposed by Karen Simonyan and Andrew Zisserman from the Visual Geometry Group of Oxford University [7]. It was applied to the ILSVRC-2014 competition achieving first and second places in the localization and classification tasks respectively.

The input of the network is a fixed size 224×244 RGB image. The image is pre-processed by subtracting the mean RGB value from each pixel and then is passed through a stack of convolutional layers with kernels of size 3×3 and

stride 1. Five max-pooling layers with 2 × 2 polling window and stride 2 follow some of the convolutional layers. The number of filters of convolutional layers ranges from 64 to 512, increasing by a factor of 2 after each max-pooling layer. Each convolutional layer has a ReLU activation function. The stack of convolutional layers is followed by three fully connected layers: the first two layers contain 4096 units each, and the third is a softmax layer and has 1000 output nodes, one for each class of the ImageNet dataset.

VGG-11	**VGG-11 LRN**	**VGG-13**	**VGG-16 conv1**	**VGG-16**	**VGG-19**
input (224×224 RGB image)					
3×3 conv, 64	3×3 conv, 64 LRN	3×3 conv, 64 3×3 conv, 64	3×3 conv, 64 3×3 conv, 64	3×3 conv, 64 3×3 conv, 64	3×3 conv, 64 3×3 conv, 64
2×2 max pool, stride 2					
3×3 conv, 128	3×3 conv, 128	3×3 conv, 128 3×3 conv, 128	3×3 conv, 128 3×3 conv, 128	3×3 conv, 128 3×3 conv, 128	3×3 conv, 128 3×3 conv, 128
2×2 max pool, stride 2					
3×3 conv, 256 3×3 conv, 256	3×3 conv, 256 3×3 conv, 256	3×3 conv, 256 3×3 conv, 256	3×3 conv, 256 3×3 conv, 256 1×1 conv, 256	3×3 conv, 256 3×3 conv, 256 3×3 conv, 256	3×3 conv, 256 3×3 conv, 256 3×3 conv, 256 3×3 conv, 256
2×2 max pool, stride 2					
3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512 1×1 conv, 512	3×3 conv, 512 3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512 3×3 conv, 512 3×3 conv, 512
2×2 max pool, stride 2					
3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512 1×1 conv, 512	3×3 conv, 512 3×3 conv, 512 3×3 conv, 512	3×3 conv, 512 3×3 conv, 512 3×3 conv, 512 3×3 conv, 512
max pool, stride 2					
fc, 4096					
fc, 4096					
fc, 1000, softmax					

Table 1.4 – VGG architectures.

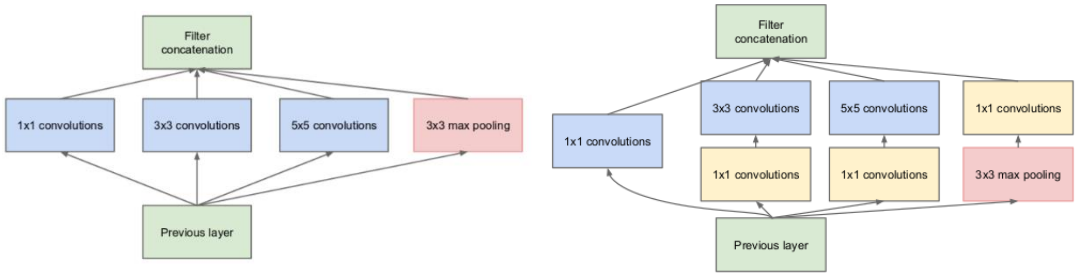
The Visual Geometry Group proposed different architectures for the network, varying the number of weighted layers from 11 (8 convolutional and 3 fully connected layers) to 19 (16 convolutional and 3 fully connected layers), as described in Table 1.4. Only one network (VGG-11 LRN) contains Local Response Normalization (LRN), a normalization layer introduced by Krizhevsky et al. [1] to

encourage lateral inhibition related to the biological phenomenon of an exited neuron inhibiting its neighbors. All the convolutional layers use 3×3 convolutional filters, except for one network (VGG-16 conv1) which also utilizes 1×1 convolutional filters as a linear transformation of the input. The stride of convolutional layers is 1.

1.3.2 Inception

In 2014, researchers at Google proposed a novel deep CNN architecture codenamed Inception [8] which achieved high-level accuracy with decreased computational cost becoming, at the time, the largest and most efficient deep neural network architecture. Under the name of GoogLeNet – but then referred also as Inception-v1 –, this network won first place in the classification task of the ILVRC-2014 competition.

The main idea behind the Inception network is substituting fully connected architectures with sparsely connected architectures, even inside the convolutional layers, to reduce the number of parameters and improve computational efficiency.



(a) (b)

Figure 1.8 – Inception module: (a) naïve version, (b) inception module with dimension reductions [8].

The core concept of sparely connected architecture is the i*nception module*, a building block in which the outputs of a 1×1 convolutional layer, a 3×3

convolutional layer, and a 5×5 convolutional layer are concatenated into a single output vector forming the input of the next layer. Figure 1.8a shows the "naïve" version of the inception module, then improved by the one represented in Figure 1.8b, in which a 1×1 convolutional layer is applied before each larger convolutional layer and after the max pooling layer to reduce dimensionality.

The overall architecture of GoogLeNet is detailed in Table 1.5.

GoogLeNet			
Layer	**Kernel size/Stride**	**Output Size**	**Depth**
convolution	7×7/2	112×112×64	1
max pool	3×3/2	56×56×64	0
convolution	3×3/1	56×56×192	2
max pool	3×3/2	28×28×192	0
inception		28×28×256	2
inception		28×28×480	2
max pool	3×3/2	14×14×480	0
inception		14×14×512	2
inception		14×14×512	2
inception		14×14×512	2
inception		14×14×528	2
inception		14×14×832	2
max pool	3×3/2	7×7×832	0
inception		7×7×832	2
inception		7×7×1024	2
avg pool		7×7×1024	0
dropout (40%)	7×7/1	7×7×1024	0
fully connected		1×1×1000	1
softmax		1×1×1000	0

Table 1.5 – Details of GoogLeNet architecture.

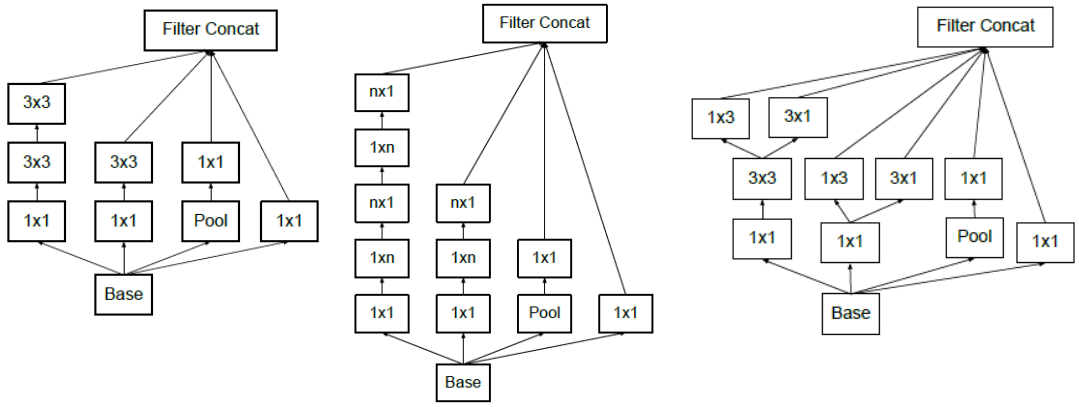
To improve the performance of this network, other versions of Inception architecture were proposed, first by the introduction of Batch Normalization (Inception-v2), then by additional factorization (Inception-v3).

Inception-v2 [5] introduces normalization – i.e. linear transformation that makes inputs have zero mean and unit variance – as part of the model architecture and performs the normalization for each training mini-batch. Applied to GoogLeNet architecture, Batch Normalization has been shown to significantly speed up the training process.

A further upgrade of the Inception networks has been made with the Inception-v3 architecture [9]. The main idea behind Inception-v3 is the factorization of convolutions to reduce the number of parameters without decreasing the network efficiency. The authors proposed two ways to achieve additional factorization:

- Factorization into smaller convolutions: convolutional layers with large kernel size (e.g. 5×5 or 7×7) can be reduced into a sequence of 3×3 convolutional layers (Figure 1.9a).
- Spatial factorization into asymmetric convolutions: a $n \times n$ convolution can be always replaced by a $1 \times n$ convolution followed by a $n \times 1$ convolution and the computational cost is reduced as n grows (Figure 1.9b and Figure 1.9c).

The schema of the Inception-v3 architecture is shown in Table 1.6.



(a) (b) (c)

Figure 1.9 – Updated inception module: (a) inception module where each 5×5 convolution layer is replaced by two 3×3 convolutional layers; (b) inception module after asymmetric factorization of the $n \times n$ convolutions (in the Inception-v3 implementation $n = 7$); (c) inception module after asymmetric factorization in coarsest grids to promote high dimensional sparse representations [9].

Inception-v3		
Layer	**Kernel size/Stride**	**Output Size**
convolution	3×3/2	149×149×32
convolution	3×3/1	147×147×32
convolution padded	3×3/1	147×147×64
pool	3×3/2	73×73×64
convolution	3×3/2	71×71×80
convolution	3×3/1	35×35×192
convolution	3×3/1	35×35×288
3×inception	As in Figure 1.9a	17×174×768
5×inception	As in Figure 1.9b	8×8×1280
2×inception	As in Figure 1.9c	8×8×2048
pool	8×8	1×1×2048
fully connected		1×1×1000
softmax		1×1×1000

Table 1.6 – Schema for the Inception-v3 architecture.

1.3.3 ResNet

The ILSVRC-2015 classification task challenge was won by a new architecture, called Residual Network (or ResNet), proposed by researchers at Microsoft Research [10]. In order to address the problem of accuracy degradation that occurs as network depth increases, the authors of ResNet introduced a deep residual learning framework, in which stacked layers are made to learn a residual mapping rather than fitting a desired underlying mapping. This is based on the hypothesis that optimizing residual mapping referenced to inputs is easier than optimizing the original unreferenced mapping.

Formally, considering $\mathcal{H}(x)$ as the underlying mapping to be fit by few stacked layers, with x denoting the inputs of the first of these layers, residual learning consists in explicitly letting the stacked layers to learn a residual function $\mathcal{F}(x) := \mathcal{H}(x) - x$ instead of directly fit $\mathcal{H}(x)$. The original mapping thus becomes $\mathcal{F}(x) + x$. Figure 1.10 shows the ResNet building block, where the operation $\mathcal{F}(x) + x$ is obtained by identity shortcut connection – i.e. a link between two distant layers without involving the set of layers between them – and element-wise addition.

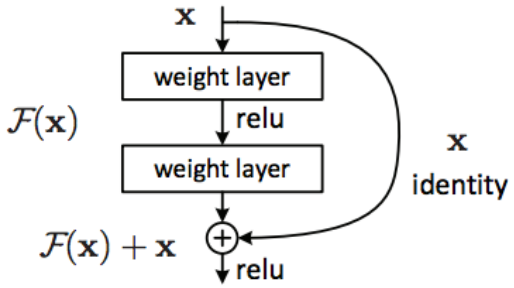


Figure 1.10 – Residual learning: a building block [10].

ResNet architectures use plain baseline networks inspired by VGG nets in which identity shortcut connections are added to perform residual learning (Figure 1.11). Details of the proposed architectures are described in Table 1.7.

ResNet-18	ResNet-34	ResNet-50	ResNet-101	ResNet-152	Output size
input (224×224 RGB image)					
7×7 conv, 64, stride 2					112×112
3×3 max pool, stride 2					112×112
$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	56×56
$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$	28×28
$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$	14×14
$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	7×7
average pool					1×1
fc, 1000, softmax					1×1

Table 1.7 – ResNet architectures. Building blocks are shown in brackets, with the number of blocks stacked. All the convolutional layers in the building blocks have stride 2.

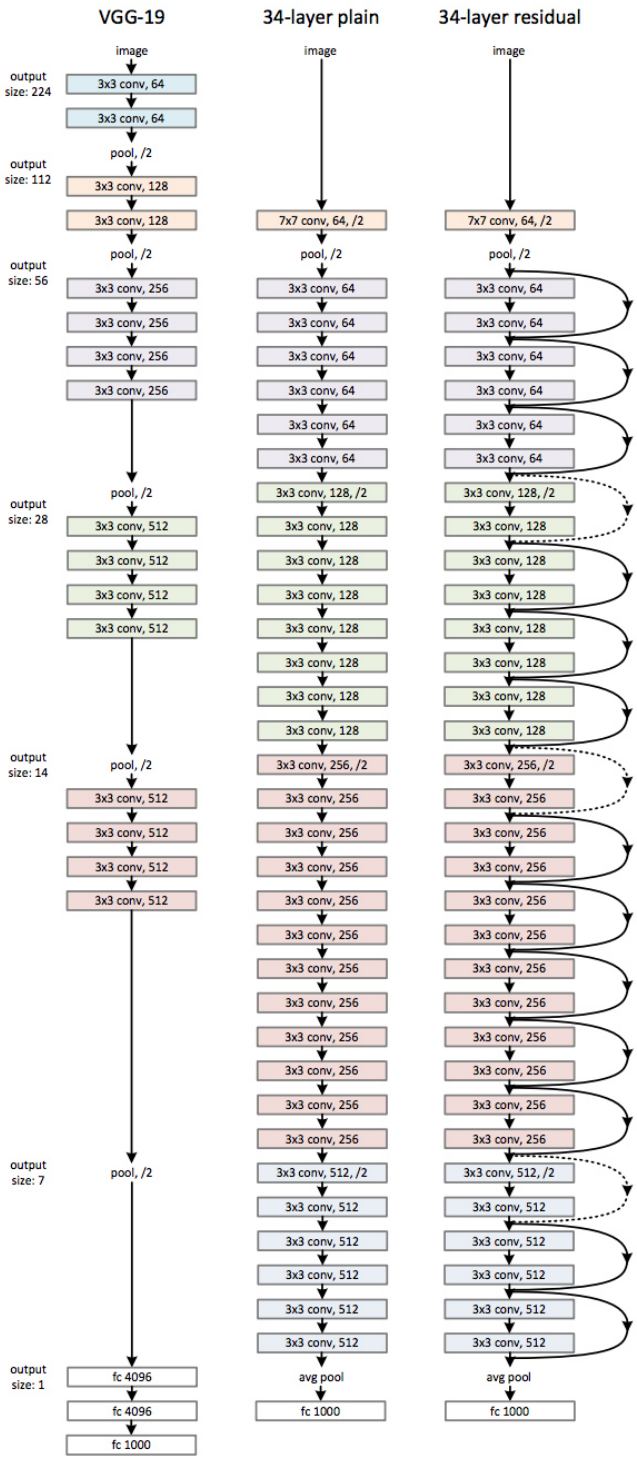


Figure 1.11 – ResNet-34. From left to right: the VGG-19 model used as a reference, a 34-layers plain network, and a residual network with 34 layers and shortcut connections [10].

1.3.4 Inception-ResNet

After the introduction of ResNet has yielded state-of-the-art performance in ILSVRC-2015 competition, the authors of Inception networks valuated the benefit in combining the Inception architecture with residual connections, demonstrating that residual connections dramatically speed up the training of Inception networks. In [11] three new architectures were proposed:

- Inception-v4: an upgraded version of Inception that has a more uniform simplified architecture and more inception modules than Inception-v3.
- Inception-ResNet-v1: a hybrid version that has a similar computational cost to Inception-v3.
- Inception-ResNet-v2: a hybrid version costlier than Inception-v3 but with significantly improved recognition performance.

The schema of those architectures is presented in Table 1.8.

Inception-v4		**Inception-ResNet-v1**		**Inception-ResNet-v2**	
Layer	**Output Size**	**Layer**	**Output Size**	**Layer**	**Output Size**
Input	299×299×3	Input	299×299×3	Input	299×299×3
Stem	35×35×384	Stem	35×35×256	Stem as Figure 1.12a	35×35×256
4×Inception-A	35×35×384	4×Inception-A	35×35×256	4×Inception-A as Figure 1.12b	35×35×384
Reduction-A	17×17×1024	Reduction-A	17×17×896	Reduction-A as Figure 1.12c	17×17×1152
7×Inception-B	17×17×1024	10×Inception-B	17×17×896	10×Inception-B as Figure 1.12d	17×17×1154
Reduction-B	8×8×1536	Reduction-B	8×8×1792	Reduction-B as Figure 1.12e	8×8×2146
3×Inception-C	8×8×1536	3×Inception-C	8×8×1792	3×Inception-C as Figure 1.12f	8×8×2048
Average Pooling	1×1×1536	Average Pooling	1×1×1792	Average Pooling	1×1×2048
Dropout (0.8)	1×1×1536	Dropout (0.8)	1×1×1792	Dropout (0.8)	1×1×1792
Softmax	1×1×1000	Softmax	1×1×1000	Softmax	1×1×1000

Table 1.8 – Schema for Inception-v4, Inception-ResNet-v1 and Inception-ResNet-v2 architectures.

Figure 1.12 shows the detailed schema of each Inception-ResNet-v2 block.

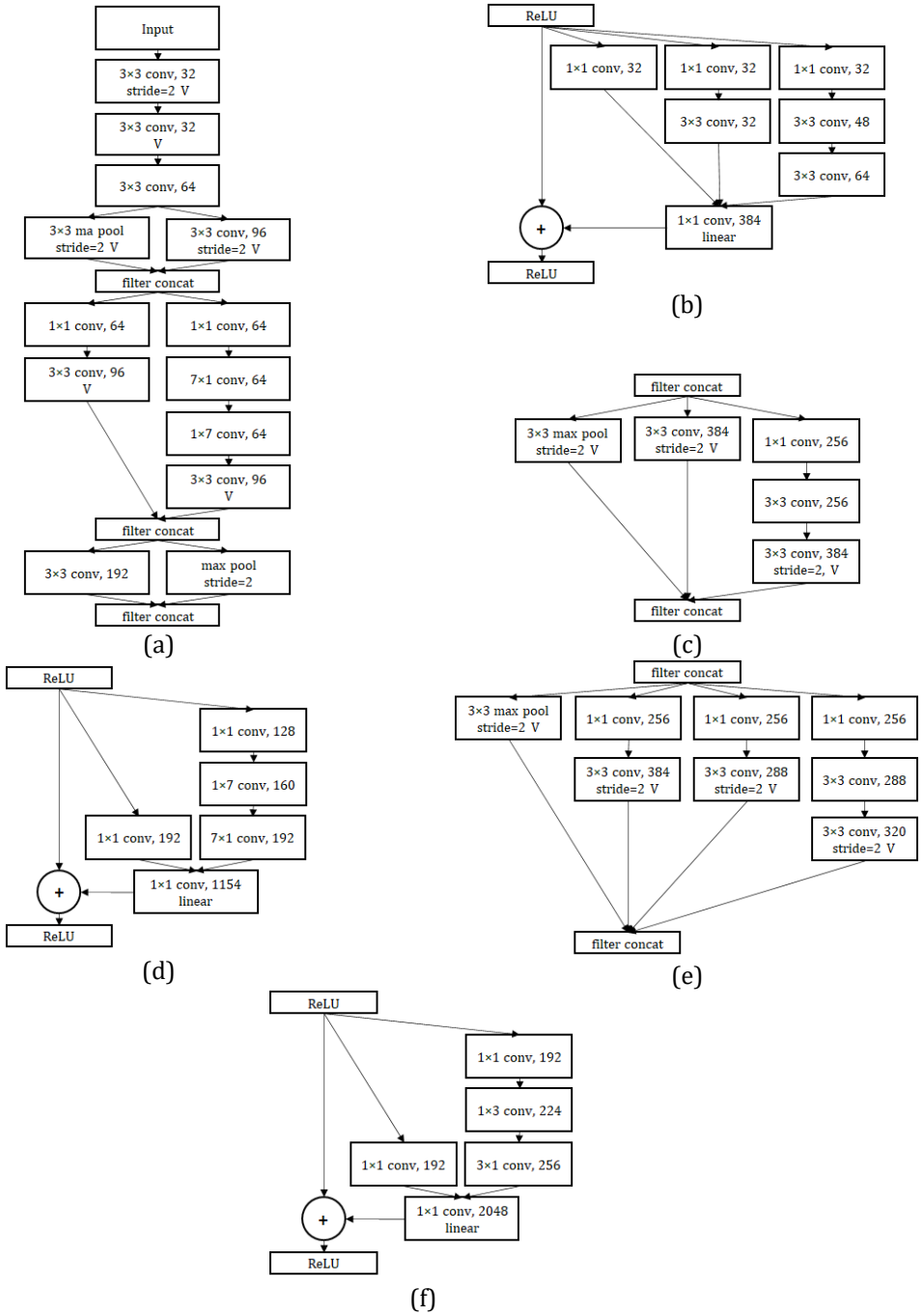


Figure 1.12 – Details of the blocks used by Inception-ResNet-v2: (a) input part; (b) Inception-A block; (c) Reduction-A block, which reduces the output size from 35×35 to 17×17, (d) Inception-B block; (e) Reduction-B block, which reduces the output size form 17×17 to 8×8; (f) Inception-C block. The overall schema is reported in Table 1.8. Convolutional layers that are not marked with "V" are zero padded so that their output has the same size of their input.

The residual versions of the Inception networks – i.e. Inception-ResNet-v1 and Inception-ResNet-v2 – have an Inception block cheaper than the original Inception module. Moreover, each inception block is followed by a 1×1 convolutional layer without activation function to scale up the dimensionality of the filter bank before the addition in order to match the depth of the input.

1.4 Model evaluation

Model evaluation is the process of using different evaluation metrics to understand the performance of a model. The most popular metrics for measuring classification performance include accuracy, sensitivity, specificity, precision, confusion matrix, and area under the ROC curve.

1.4.1 Accuracy, sensitivity, specificity, and precision

Accuracy measures the ratio between the number of correct predicted classes to the total number of predictions (Equation 1.8).

$$Accuracy = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ predictions} \quad 1.8$$

For binary classification, accuracy can also be expressed in terms of positives and negatives, as follows (Equation 1.9).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad 1.9$$

In binary classification *True Positive* (TP) and *True Negative (TN)* are defined as the number of positive and negative instances, respectively, which are correctly classified by the model while *False Positive* (FP) and *False Negative* (FN)

are defined as the number of negative and positive instances, respectively, incorrectly classified by the model.

Sensitivity, also known as *true positive rate* (*TPR*) or *recall*, measures the fraction of positive cases that are correctly classified and is defined as the number of true positives divided by the total number of positive cases (Equation 1.10).

$$Sensitivity = \frac{TP}{TP + FN} \tag{1.10}$$

Specificity calculates the fraction of negative cases that are correctly classified and is defined as the number of true negatives divided by the total number of negative cases (Equation 1.11).

$$Specificity = \frac{TN}{TN + FP} \tag{1.11}$$

Precision measures the proportion of positive cases that are actually correct and is defined as the number of true positives divided by the total number of positive predictions (Equation 1.12).

$$Precision = \frac{TP}{TP + FP} \tag{1.12}$$

1.4.2 Confusion matrix

A *confusion matrix* is a table commonly used to evaluate the performance of a classification model. It compares the actual target values, or *ground truths*, with those predicted by the model.

In binary classification, where target classes are labeled as positive or negative, a confusion matrix is a 2×2 matrix arranged as shown in Figure 1.13.

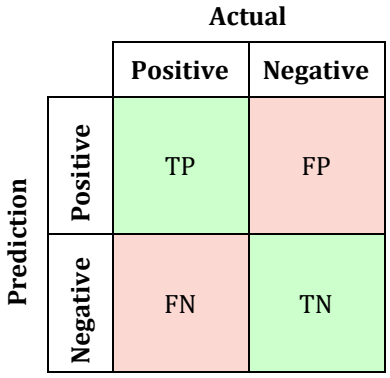


Figure 1.13 – Confusion matrix for binary classification.

In multiclass classification, the confusion matrix is a $N \times N$ matrix, with N the number of classes, where the columns represent the actual values and the rows represent the predicted ones (or *vice-versa*). Multiclass data are binarized under a *One-vs-Rest* (*OvR*) transformation, which splits the multiclass classification into one binary classification problem per class.

1.4.3 Area under the ROC curve

The *receiver operating characteristic curve (ROC curve)* is a graph that shows the performance of a classification model by plotting the true positive rate against the false positive rate at different classification thresholds.

The *false positive rate* (FPR), also known as *fall-out*, is defined as the number of false positives divided by the total number of negative cases (Equation 1.13).

$$FPR = \frac{FP}{FP + TN} = 1 - Specificity \tag{1.13}$$

ROC curves are typically used in binary classification. In order to extend ROC curves to multiclass tasks, it is necessary to binarize the output.

The *One-vs-Rest* (OvR) methodology reduces the multiclass classification output into a binary classification by comparing each class against all the others at the same time.

The *micro-average* method calculates the performance from the individual true positives, true negatives, false positives and false negatives of the model:

$$TPR_{micro} = \frac{TP_1 + \cdots + TP_k}{TP_1 + \cdots + TP_k + FN_1 + \cdots + FN_k} \quad 1.14$$

$$FPR_{micro} = \frac{FP_1 + \cdots + FP_k}{FP_1 + \cdots + FP_k + TN_1 + \cdots + TN_k} \quad 1.15$$

This is the preferable method in case of class imbalance.

The *macro-average* method averages the performances of each individual class:

$$TPR_{macro} = \frac{TPR_1 + \cdots + TPR_k}{k} \quad 1.16$$

$$FPR_{macro} = \frac{FPR_1 + \cdots + FPR_k}{k} \quad 1.17$$

The *area under the ROC curve (AUC)* measures the two-dimensional area underneath the ROC curve from (0,0) to (1,1). It ranges in value from 0.0 (all predictions wrong) to 1.0 (all predictions correct) and indicates how well the model is capable of distinguishing between classes.

Since AUC is scale-invariant and classification-threshold-invariant, it is a metric commonly used to evaluate the performance of a classification model and to conduct comparisons between models.

1.5 Explainable AI

One of the main stumbling blocks to the application of artificial intelligence to medical imaging is the fact that AI systems are often perceived by physicians as black boxes. This has led to a relatively low acceptance of AI among physicians, who complain that they are unable to understand the prediction process and verify the results given by the models.

To encourage the use of AI systems in the clinical field, some studies attempt to incorporate *explainable artificial intelligence* (XAI) into their proposed model. XAI is a technique that has been introduced to explain how the AI model derived its prediction, helping the user to understand the prediction process. In image classification, moreover, XAI can highlight which part of the image is most important for the purpose of the prediction.

One of the most common XAI techniques used to interpret DL models applied to medical images is the *Gradient-weighted Class Activation Mappin*g (Grad-CAM) [12], which aims to generate visual explanations for a large class of CNN-based network without requiring architectural changes or re-training.

GradCAM is typically applied at the final convolutional layer, which is known to have the best compromise between high-level semantics and detailed spatial information, using the gradient information flowing into it to assign an importance value to each neuron for a particular decision of interest. GradCAM output is a coarse localization heatmap that highlights the regions of the input image considered important for the final prediction.

In image classification tasks, GradCAM has been shown to be useful not only in improving model interpretability but also in diagnosing model failures, helping to identify biases in datasets, and to discern a stronger network from a weaker one.

Chapter 2
Automated quality assessment of histological images in liver biopsies

2.1 Background and objective

The introduction of digitalized whole slide images (WSIs) technology has brought an exponentially growing interest in the development of computer-aided systems for the analysis of histological slides. These systems aim to create new clinical tools capable to surpass current clinical approaches in terms of accuracy, objectivity, and reproducibility improving efficiency, safety, and quality of diagnosis. Promising results of the application of deep learning to histological images have given rise to the development of many computer-aided systems, most of them to support pathologists in tasks such as tumor detection and grading [13][14][15][16][17][18].

The research group born from the collaboration between the Surgical Pathology Unit of Cattinara Hospital and the Health Telematics Laboratory (HTL) of the Complex Structure of Informatics and Telecommunications at Azienda Sanitaria Universitaria Giuliana Isontina (ASUGI) is currently working at the

development of a computer-aided system that applies deep learning methods to support pathologists in the diagnosis of human non-alcoholic fatty liver disease (NAFLD).

NAFLD affects one-quarter of the general population [19] and is recognized as the leading cause of chronic liver disease across all age groups [20], constituting a serious and growing clinical problem. NAFLD encompasses a spectrum of liver conditions that start from simple steatosis and may progress to nonalcoholic steatohepatitis (NASH), fibrosis, cirrhosis, and hepatocellular carcinoma [21]. Early diagnosis of NAFLD is therefore fundamental to halting the progress of the disease. Although the diagnosis of NAFLD can be achieved by imaging procedures such as ultrasounds, computerized tomography, or magnetic resonance imaging, liver biopsy remains the gold standard for accurately staging the disease [22]. The most widely used histological staging system for NAFLD is the NAFLD activity score (NAS) proposed by the Pathology Committee of the NASH Clinical Research Network. The NAS system comprises 14 histological features and is shown in Table 2.1 [23].

Item	Definition	Score
Steatosis		
Grade	*Low-to medium-power evaluation of parenchymal involvement by steatosis*	
	<5%	0
	5%-33%	1
	33%-66%	2
	>66%	3
Location	Predominant distribution pattern	
	Zone 3	0
	Zone 1	1
	Azonar	2
	Panacinar	3
Macrovesicular steatosis	*Continuous patches*	
	Not present	0
	Present	1
Fibrosis		
Stage		
	None	0
	Perisinusoidal or periportal	1

	Mild, zone 3, perisinusoidal	1A
	Moderate, zone 3, perisinusoidal	1B
	Portal/periportal	1C
	Perisinusoidal and portal/periportal	2
	Bridging fibrosis	3
	Cirrhosis	4
Inflammation		
Lobular Inflammation	*Overall assessment of all inflammatory foci*	
	No foci	0
	<2 foci per 200× field	1
	2-4 foci per 200× field	2
	> 4 foci per 200× field	3
Microgranulomas	*Small aggregates of macrophages*	
	Absent	0
	Present	1
Large lipogranulomas	*Usually in portal areas or adjacent to central veins*	
	Absent	0
	Present	1
Portal inflammation	*Assessed from low magnification*	
	None to minimal	0
	Greater than minimal	1
Liver call injury		
Ballooning		
	None	0
	Few balloon cell	1
	Many cells/prominent ballooning	2
Acidophil bodies		
	None to rare	0
	Many	1
Pigmented macrophages		
	None to rare	0
	Many	1
Megamitochondria		
	None to rare	0
	Many	1
Other findings		
Mallory's hyaline	*Visible on routine stains*	
	None to rare	0
	Many	1
Glycogenated nuclei	*Contiguous patches*	
	None to rare	0
	Many	1

Table 2.1 – NASH Clinical Research Network Scoring System Definitions and Scores [23].

As the first stage of the overall project, the ASUGI group developed a method to automatically quantify the hepatic steatosis grade in liver biopsies, accordingly to the NAS scoring system.

Hepatic steatosis is characterized by the abnormal accumulation of triacylglycerol lipid droplets within the hepatocytes, the liver parenchymal cells performing many metabolic functions. Although hepatic steatosis is often considered a benign condition, excess lipid storage is a risk factor for liver metabolic dysfunction, inflammation, and advanced forms of NAFLD.

In clinical practices, pathologists determine the grade of hepatic steatosis by evaluating hematoxylin and eosin (H&E) stained liver histology specimens. However, this scoring methodology is prone to significant inter- and intra-observer variation due to sampling bias and poor reproducibility [24].

In recent years, some studies have been proposed for the automatic quantification of hepatic steatosis in liver biopsy using approaches ranging from classical image processing techniques, such as image segmentation [25][26], to more advanced methods based on machine learning [27] and deep learning [28][29][30].

The method proposed by the ASUGI group determines the grade of steatosis using a hierarchical network structure composed of three CNNs, inspired by VGG-16 architecture, which perform binary classification. A schema of the overall network is shown in Figure 2.1.

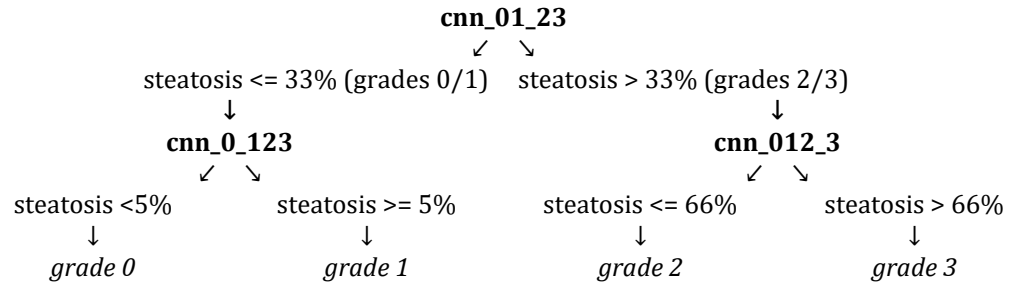


Figure 2.1 – Schema of the overall network proposed by the ASUGI group for hepatic steatosis grading.

The image is initially processed by *cnn_01_23* and then, depending on the obtained result, it is successively fed to *cnn_0_123* or *cnn_012_3* to perform the

final classification. The proposed model reached a classification accuracy higher than 95% and comparable to the one achieved by experienced pathologists.

Despite the performance obtained, the major limit to the reliability of that method was represented by the need for the intervention of a pathologist to confirm the quality of the images before they are processed by the network.

If on one hand performing analysis on low-quality images hampers the validity of the system, on the other hand, the manual image quality assessment precludes the development of a fully automated and objective system, is time-consuming and inefficient, particularly in the context of a lack of pathologist resource. This issue unites any computer-aided system for images evaluation: as a matter of fact, before carrying out any type of image analysis it is essential to verify that the images meet a certain standard of quality [31].

In the specific case of histopathology, the main cause of low-quality images is given by various types of artifacts that can be introduced throughout the entire slide preparation workflow, including surgical removal, fixation, tissue processing, embedding and microtomy, staining, and mounting procedures [32], as well as during slide digitization. The presence of artifacts, which can be defined as structure or tissue alteration on a prepared microscopic slide as a result of an extraneous factor, may lead to an incorrect or inconclusive interpretation of tissue sections. While trained pathologists are able to identify and ignore artifacts during histopathology examination, a computer-aided system may misinterpret them, giving rise to erroneous results. Discarding portions of an image presenting artifacts before its processing is therefore an important step in improving the accuracy of the analysis. However, the heterogeneity and the high number of artifact types that may cause tissue alterations make the quality assessment of histological images a very complex task to achieve automatically.

In this context, therefore, it is this study that has as its objective the development of a method for the automated quality assessment (QA) of histological images to be applied to the case of the hepatic steatosis.

2.2 Related works

As mentioned above, the QA of histological images is complex. This is due to the intrinsic complexity of tissue sections, as well as to the variety of artifact types that may impact tissues in several ways during slide preparation and digitalization. This is further complicated by the fact that the concept of image quality concerning histological images is strictly dependent on the diagnostic question and requires evaluation by an experienced pathologist. For these reasons, it would be advisable that methods for image QA make use of learnable features rather than hand-crafted ones.

The current literature lacks studies on the automatic QA of histological images. The few studies present tend to be limited to the identification of only one particular type of artifact, using hand-crafted features and without considering a certain level of acceptability that only an expert pathologist can provide.

Ameisen *et al.* [33] developed a method that automatically assesses WSIs during image acquisition by evaluating blurriness, contrast, brightness, and color. Although this method allows to quickly determine if WSIs obtained from an acquisition system have sufficient quality to be accepted or need to be re-acquired, speeding up laboratory workflow, it takes only into account artifacts that may be introduced during digitization.

Kothari *et al.* [34] used handcrafted features and image statistics to detect tissue-fold artifacts, a specific type of artifact that occurs when a thin tissue slice folds on itself. In the proposed method tissue-fold artifacts are detected by estimating adaptive soft and hard thresholds based on tissue connectivity in the saturation and intensity color space. The authors demonstrate that applying this method on ovarian serous adenocarcinoma (OvCa) and kidney clear carcinoma (KiCa) WSIs, available from The Cancer Genome Atlas, to detect and eliminate

tissue-fold artifacts improves the performance of cancer-grade prediction by 5% and 1% in OvCa and KiCa respectively.

Vanderbeck *et al.* [35] automatically assessed hepatic steatosis in NAFLD using supervised machine learning classifiers to differentiate the various types of white regions that may be present in liver biopsies (e.g. macrosteatosis, central veins, portal veins, portal arteries, sinusoids and bide ducts) through the evaluation of morphological features and the examination of texture and statistical properties of adjacent regions. The authors obtained good performance in classifying white regions as either steatosis or not-steatosis achieving an overall accuracy of 89%.

The most noteworthy study in that field was proposed by Foucart *et al.* [36], which for the first time applied a method based on deep learning to detect various types of artifacts on WSIs. In that study, the authors used a weak and noisy dataset – i.e. a dataset less precise than the desired output with uncertain labels – to train different residual network architectures, whose details are shown in Figure 2.2.

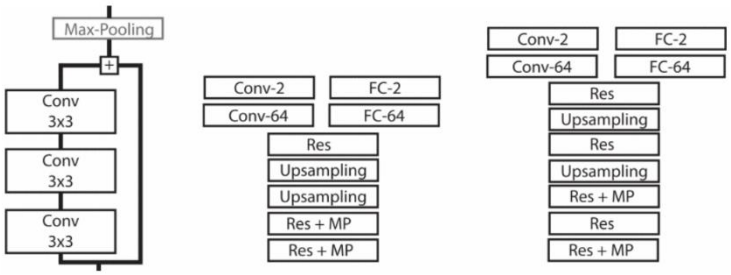


Figure 2.2 – Residual network architectures proposed in [36]. From left to right: residual unit, Residual-3 architecture (also referred as to Res3), and Residual-5 architecture (also referred as to Res5).

The core architecture is based on residual units and up-sampling layers to resize the last feature map into the original image size and is used to perform both detection – i.e. tile-based classification – and segmentation – i.e. pixel-based

classification – tasks. The training dataset was balanced by forcing every batch to contain a certain portion of tiles with artifacts.

	Architecture	Task	Balance	Accuracy	TPR	TNR
A	Res3	Detection	50%	84.10%	87.21%	83.97%
B	Res5	Detection	50%	81.22%	87.36%	80.98%
C	Res5	Segmentation	50%	94.67%	65.88%	95.83%
D	Res5	Segmentation	100%	95.62%	69.95%	96.68%
B+C				86.14%	87.34%	86.10%

Table 2.2 - Results of the best performing networks in [36].

Results in Table 2.2 show that the detection networks are inclined to overestimate artifacts, while the segmentation networks tend to underestimate them. However, both networks were able to successfully find areas of tissue without artifacts for further processing.

2.3 Material and Methods

2.3.1 Original dataset

The research related to the development of a method to automatically assess image quality in histological images was conducted starting from the dataset previously collected by the ASUGI group for the hepatic steatosis staging networks.

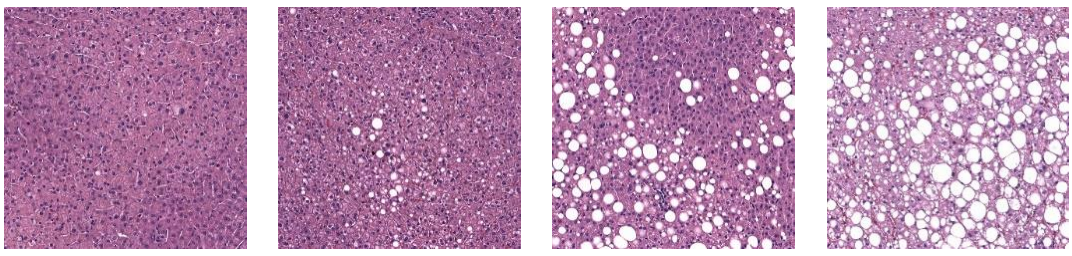
The original dataset consists of 41 liver biopsies obtained from subjects undergoing bariatric surgery at the Cattinara Hospital (Azienda Sanitaria Universitaria Giuliana Isontina - ASUGI, Trieste, Italy) digitally scanned into WSIs. A multidisciplinary team composed of surgeons, dieticians, hepatologists, and psychiatrists prospectively and consecutively enrolled subjects undergoing bariatric surgery.

Inclusion criteria were based on international guidelines: age 18-65 years, body mass index (BMI) > 40 kg/m^2 or > 35 kg/m^2 if obesity-related comorbidity is already present, acceptable operative risks, absence of surgical treatments, declared compliance to follow lifelong medical surveillance [37].

Liver biopsies were fixed in 10% formalin, embedded in paraffin, and stained with hematoxylin and eosin. Slides were scanned using D-Sight Menarini® microscope and stored in JPEG 2000 format. Patient records were processed accordingly to protect their sensitive data by replacing them with anonymous codes. Tiled (i.e. non-overlapping) images of 1024×1024 pixels were extracted from each WSI and examined by trained pathologists with more than 5 years of experience.

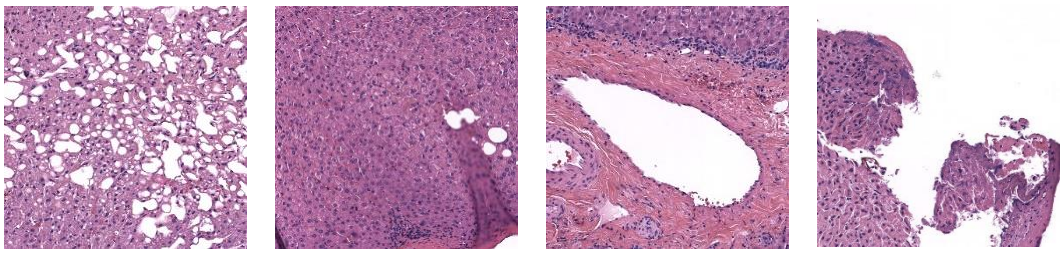
The grade of hepatic steatosis was assessed with respect to the NAS score. Absence/minimal steatosis (grade 0) was classified as less than 5% of macrovesicular steatosis (Figure 2.3a), mild steatosis (grade 1) between 5% and 33% (Figure 2.3b), moderate steatosis (grade 2) between 33% and 66% (Figure 2.3c) and severe steatosis (grade 3) more than 66% (Figure 2.3d).

Tiled images that did not meet the quality standard required for further processing were labeled as *not usable*. The threshold of usability was determined by trained pathologists according to the absence of artifacts (Figure 2.4a-b) and uninformative data, i.e. large white gaps in the tissue that differ from lipid droplets, such as big vessels and background (Figure 2.4c-d).



(a) (b) (c) (d)

Figure 2.3 – Examples of hepatic steatosis grades accordingly to the NAS scoring system: (a) grade 0, (b) grade 1, (c) grade 2, (d) grade 3.



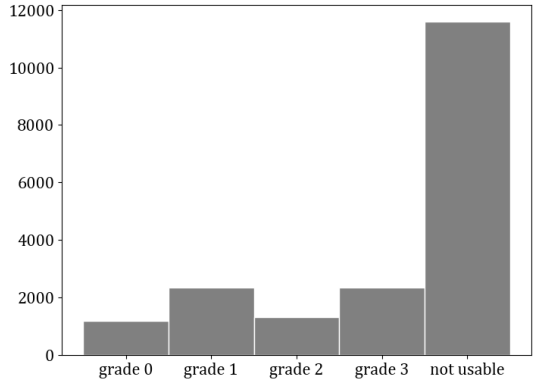
(a) (b) (c) (d)

Figure 2.4 – Examples of images that do not meet the quality standard due to the presence of various types of artifacts (a -b) and uninformative data (c-d).

The use of tiled images rather than WSIs is the most common choice when deep learning methods are applied to the analysis of histological images. In addition, most successful deep learning-based approaches of WSIs analysis extract only a small number of patches instead of using the whole image [38].

WSIs are images with high resolution (typically 100,000×100,000 pixels) and contain a vast amount of data that may not only be unfeasible to process in their entirety, but also partially uninformative or prone to incorrect interpretation. Efforts to limit the analysis to regions of interest can be therefore necessary for accurate and computationally feasible analysis. This is supported by the fact that also in clinical practice pathologists focus their examination on only the slide portions of greatest interest. Moreover, patch-level annotations, which imply that each training image has its own class label, allow more accurate ground truth labeling and enable strong supervision.

The distribution of tiled images labels into the dataset is shown in Figure 2.5. Patches labeled as *not usable* represent the majority of the images, and this is justified by the fact that WSIs contain a large number of white pixels corresponding to the background of the slide.



Classes	Samples
grade 0	1,190
grade 1	2,348
grade 2	1,311
grade 3	2,337
not usable	11,598
Total	**18,784**

Figure 2.5 – Classes distribution of the original dataset.

2.3.2 Methodology

Since in the original dataset the tiled images were labeled as *not usable* when including not only artifacts but also uninformative data – i.e. images composed mainly of background or large vessels –, the developed QA method was divided into two consecutive stages (Figure 2.6).

In the first stage, from now on referred as to the *pre-processing stage*, classical digital image processing methods are applied to the original dataset with the aim of identifying and discarding in a simple and objective way uninformative data.

In the next one, the a*rtifact detection stage*, the remaining images are fed into a CNN to detect various types of artifacts. As discussed before, artifacts are complex elements that should be better identified with learnable features rather than hand-crafted features. CNNs, which automatically and adaptively learn important features directly from images, perfectly meet this need.

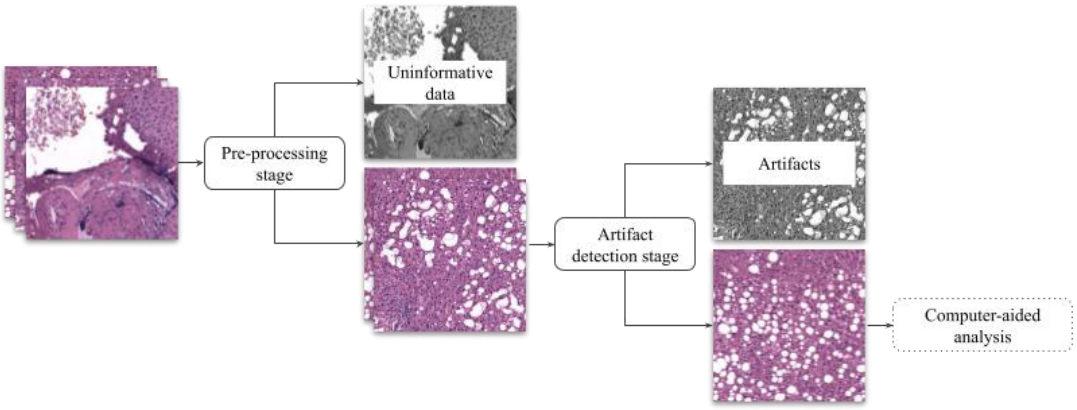


Figure 2.6 – Workflow of the proposed method: image patches are pre-processed in order to discard those with a high percentage of uninformative white regions and then fed into a CNN which allows separating good-quality images from those with various types of artifacts.

2.3.2.1 Pre-processing stage

The pre-processing stage is intended to identify and discard image patches that do not contain informative data.

WSIs usually contain large white areas representing the slide background. When a WSI is divided into tiled patches, a vast amount of them will be empty or made of an insufficient portion of tissue for further examination. Moreover, large white areas in image patches may also be related to big vessels and, in the specific case of hepatic steatosis, should be discarded not only because they represent uninformative data but also because a computer-aided system may misinterpret them. Hepatic steatosis is visible as fat droplets in the cytoplasm of hepatocytes. During histological processing, the fat dissolves and droplets appear as white gaps in the hepatic tissue. However, fat droplets can most often be distinguished from other empty spaces, such as vessels, by their size (Figure 2.7).

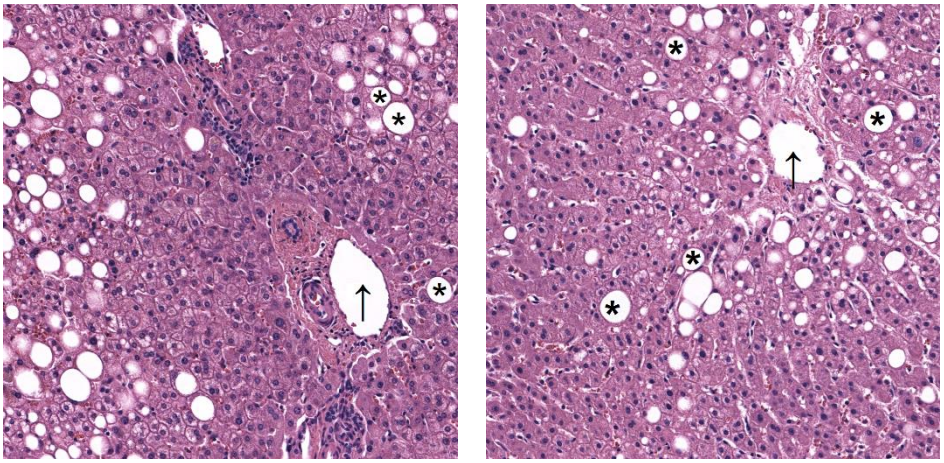
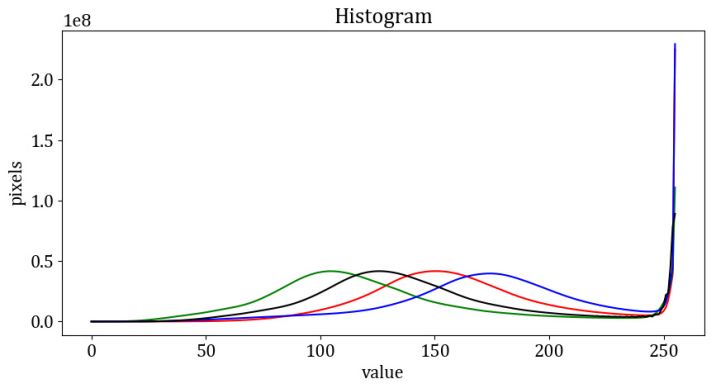


Figure 2.7 – Hepatic tissue with steatosis: fat droplets are marked with asterisks while big vessels are marked with black arrows.

Given these premises, the simplest but still effective approach to identifying large non-informative white areas was the use of classical image processing techniques to analyze the ratio between the size of these areas with respect to the size of the whole tiled image.

Initially, the method was applied only to the images labeled as *not usable* in order to find the threshold to be used to discriminate small droplets of fat from other tissue gaps. Then the same method was then applied to the whole dataset using the found threshold to automatically identify uninformative data.



(a)

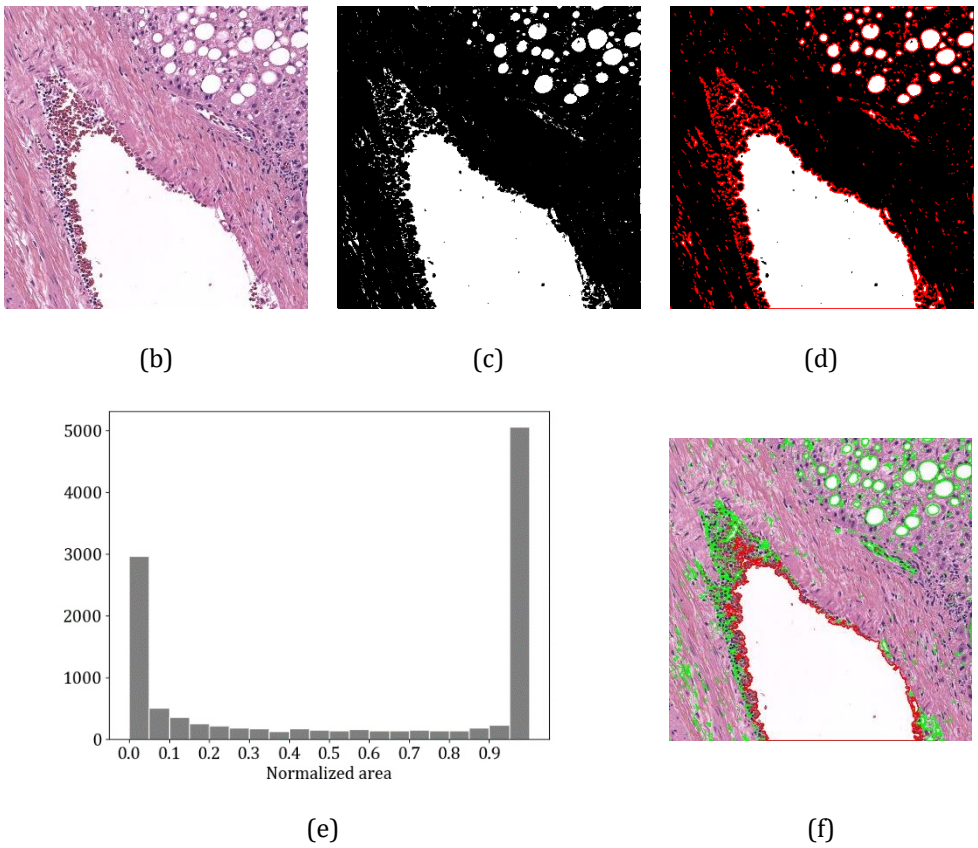


Figure 2.8 – Steps of the pre-processing stage method: (a) intensity histogram, (b) original image, (c) binarized image, (d) contours of white areas superimposed on the binarized image (red lines), (e) histogram of the normalized maximum area within each *not usable* image, (f) contours of white areas superimposed on the original image (green lines); contours of areas over the threshold are highlighted in red.

RGB images were converted to grayscale images and binary thresholding was performed to separate white pixels from pixels representing tissue. The threshold value was set to 245 accordingly to the intensity histogram evaluation (Figure 2.8a-c).

The contour – i.e. the curves joining continuous points with same intensity along their boundary – of each white space was computed and its area was calculated (Figure 2.8d). The histogram of the normalized maximum area within each *non usable* image was then plotted in order to find the cutoff between steatosis droplets and other white areas (Figure 2.8e). From the histogram

evaluation, the threshold was set to 10% of the whole area of the tiled image (Figure 2.8f).

The threshold was then applied to the whole dataset in order to objectively discard non-informative data: image patches containing a white area greater than 10% of the whole area were automatically classified as uninformative and excluded from the dataset.

2.3.2.2 Artifact detection stage

Images that were not discarded from the previous stage were used to populate the dataset on which to train, validate and test the DL model for artifact detection. The remaining image patches were randomly selected to create a balanced dataset: 3,200 images were taken from the *not usable* class while 800 images were taken from each steatosis class, for a total of 3,200 *good-quality* images. The final dataset, containing 6,400 images, was then randomly split into a training set (80%), used to train the network, a validation set (10%), used to evaluate the model during the training process, and a test set (10%), used to evaluate the performance of the final model, keeping class balance. It is worth to be noted that the study aims to evaluate the quality of the images at patch level rather than at WSIs level. For this reason, the dataset was split without considering a division in terms of patients but only in terms of image patches. The repartition of the final dataset is shown in Table 2.3.

Classes		**Training set**	**Validation set**	**Test set**
good-quality	grade 0	640 (10%)	80 (1.25%)	80 (1.25%)
	grade 1	640 (10%)	80 (1.25%)	80 (1.25%)
	grade 2	640 (10%)	80 (1.25%)	80 (1.25%)
	grade 3	640 (10%)	80 (1.25%)	80 (1.25%)
artifacts		2,560 (40%)	320 (5%)	320 (5%)
Total		**5,120** (80%)	**640** (10%)	**640** (10%)

Table 2.3 – Dataset splitting in training, validation, and test sets for class balancing. The class named *artifacts* refers to the samples originally labeled as *not usable* that were not discarded after the pre-processing stage.

Although VGG-16 [7] is one of the earliest CNN models used for image recognition, it was the architecture initially examined to perform artifact detection. This choice was motivated by the fact that this network reflects the base model chosen by the ASUGI group to perform steatosis grading and it is still one of the simplest and most widespread CNN networks used for medical image classification thanks to its easy implementation and its good accuracy. For completeness, it should be noted that also a custom network, having the same architecture as the base model of the ASUGI group, as well as deeper VGG nets and different architecture of residual networks were examined, but the performance obtained was not worthy of being considered.

In this study, the network was used as a binary classifier. The VGG-16 architecture was pre-trained on ImageNet dataset [39] and fine-tuned. A single unit layer with a sigmoid activation function was added on the top of the network to encode the probability that the input image has artifacts. To prevent overfitting, dropout regularization with a dropout ratio of 0.5 was added before the last layer.

Before feeding the image into the network, the data was preprocessed by downsizing it from 1024 × 1024 pixels to 300 × 300 pixels, due to memory limitations, and scaling it so that all values are in the [0, 1] range.

Horizontal and vertical flipping was adopted for data augmentation in order to reduce overfitting.

The hyperparameters were optimized on the basis of the performances obtained on the validation set. The network was trained for 100 epochs in mini-batches of 16 samples using Adam optimizer and binary cross entropy loss. The learning rate was initially set to 10^{-4} and then reduced by a factor of 2 when the validation set accuracy stopped improving. During training, only the weights of the model that achieved maximum accuracy on the validation set were maintained, regardless of the epoch number reached.

The summary of the overall architecture of the network and its training configuration are reported in Table 2.4.

Layer		Output size	Params
input		300×300×3	0
block 1	conv 1	300×300×64	1,792
	conv 2	300×300×64	36,928
	max pooling	150×150×64	0
block 2	conv 1	150×150×128	73,856
	conv 2	150×150×128	147,584
	max pooling	75×75×128	0
block 3	conv 1	75×75×256	295,168
	conv 2	75×75×256	590,080
	conv 3	75×75×256	590,080
	max pooling	37×37×256	0
block4	conv 1	37×37×512	1,180,160
	conv 2	37×37×512	2,359,808
	conv 3	37×37×512	2,359,808
	max pooling	18×18×512	0
block 5	conv 1	18×18×512	2,359,808
	conv 2	18×18×512	2,359,808
	conv 3	18×18×512	2,359,808
	max pooling	9×9×512	0
GAP		512	0
dropout (0.5)		512	0
dense (sigmoid)		1	513
Total number of trainable parameters:			**14,715,201**

Hyperparameters	
Optimizer	Adam
Learning rate	10^{-4}
Loss function	Binary cross entropy
Batch size	16
Epochs	100
Regularization	Dropout
Weight initialization	ImageNet
Dataset splitting	Train-test

Table 2.4 – Summary of the VGG-16 model used to perform artifacts detection and initial training hyperparameters configuration.

2.3.2.3 Framework

All the code was written in Python. Images were pre-processed using the OpenCV Python library. DL experiments were implemented using TensorFlow and Keras. Training and evaluation were performed using an NVIDIA GeForce GTX 1070 graphic processing unit (GPU) on a 64-bit computer processor with an Intel(R) Core (TM) i7-8700 CPU @ 3.20GHz.

2.4 Results

2.4.1 Pre-processing stage

In order to discard uninformative data, a total of 18,784 tiled images, belonging to the original dataset, were pre-processed using the method proposed in section 2.3.2.1 The results of applying the method on the original dataset are reported in Table 2.5. 8,249 images – i.e. 43.9% of the total number – were automatically identified as uninformative data. Of these, 98.4% come from the *not usable* class while only a minimal part comes from the classes initially considered as informative and labeled according to steatosis grades. All the discarded samples were reviewed by a trained pathologist and approved.

Classes	Original dataset	Maintained samples	Discarded samples
grade 0	1,190	1,186	4
grade 1	2,348	2,342	6
grade 2	1,311	1,308	3
grade 3	2,337	2,221	116
not usable	11,598	3,478	8,120
Total	**18,784**	**10,535**	**8,249**

Table 2.5 – Repartition of the original dataset after the pre-processing stage.

Figure 2.9 shows some representative images for each class and output type (maintained/discarded). White tissue gaps having an area minor than the threshold – i.e. <= 10% of the tiled image area – are contoured by a green line while empty spaces having an area major than the threshold are highlighted in red. In the latter case, images are excluded from the original dataset.

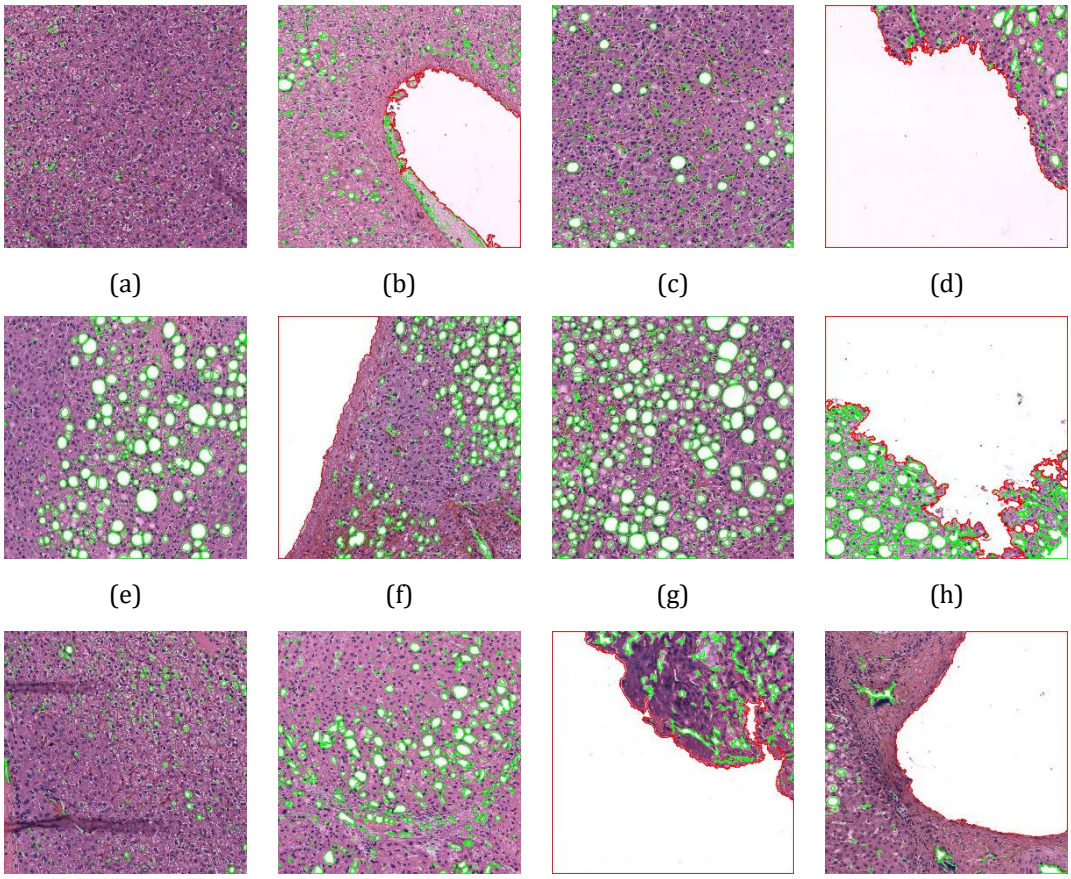


Figure 2.9 – Examples of image patches underwent the pre-processing stage for each class and output type: (a) *grade 0*-maintained, (b) *grade 0*-discarded, (c) *grade1*-maintained, (d) *grade1*-discarded, (e) *grade 2*-maintained, (f) *grade2*-discarded, (g) *grade3*-maintained, (h) *grade3*-discarded, (i) *not-usable*-maintained (then labeled as *artifacts*), (j) *not-usable*-discarded.

2.4.2 Artifacts detection stage

The VGG-16 network was trained on 5,120 images belonging to the training set and fine-tuned according to its performance on 640 images belonging to the validation set. Many experiments were done and only the model with the best performance on the validation set was selected.

The optimal binary classification threshold – i.e. the value providing the best tradeoff between sensitivity and specificity – was computed through the ROC

curve analysis on the validation set. It is the value t for which the Youden Index (or Youden's J statistic) [40] is maximal:

$$J_{max} = max_t(Sensitivity_t + Specificity_t - 1) \quad 2.1$$

In ROC diagrams, the Youden Index is the vertical distance between the ROC curve of the classifier and the ROC curve of a random classifier represented by the diagonal line (Figure 2.10). On the validation set, the threshold that maximizes the Youden Index is $t = 0.0788$; for which sensitivity and specificity are 0.9813 and 0.9594, respectively, and $J_{max} = 0.9407$.

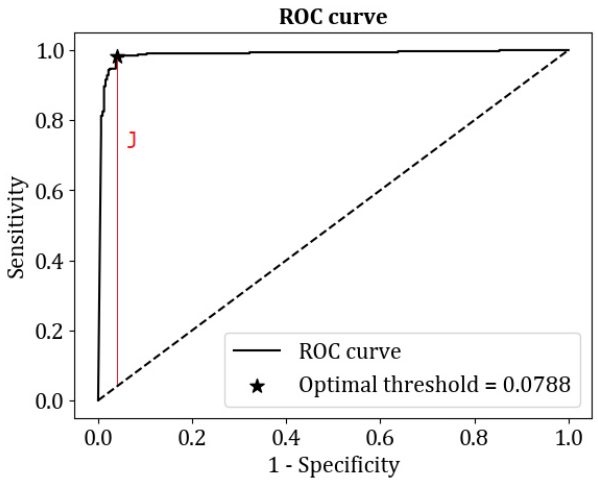


Figure 2.10 – ROC curve on the validation set. In red the Youden Index at the optimal threshold.

The performance of the final model was evaluated on the test set, composed of 640 images independent of training and validation sets. The ROC curve and the confusion matrix are shown in Figure 2.11 and Figure 2.12, respectively. The metrics on both validation and test sets are summarized in Table 2.6.

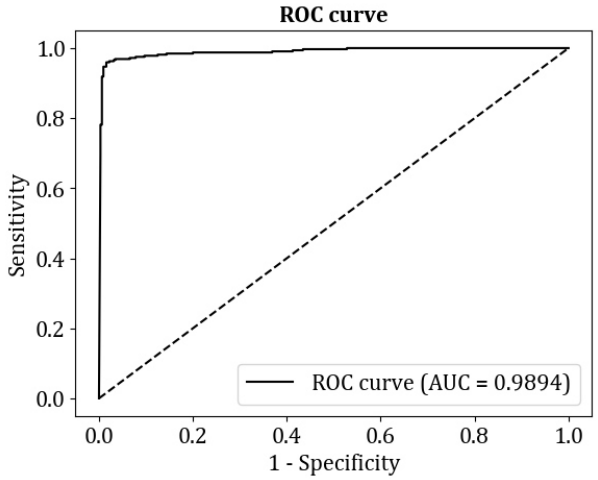


Figure 2.11 – ROC curve of the final model.

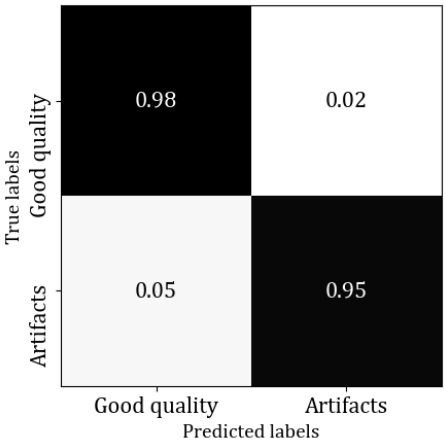


Figure 2.12 – Confusion matrix of the final model (optimal threshold $t = 0.0788$).

	Accuracy	TPR	TNR	AUC
Validation set	97.97%	98.13%	95.94%	-
Test set	96.88%	95.31%	98.44%	98.94%

Table 2.6 – Final model performance on validation and test sets.

Model	Task	Accuracy	TPR	TNR
Res3 [36]	Artifacts detection	84.10%	87.21%	83.97%
Res5 [36]	Artifacts detection	81.22%	87.36%	80.98%
VGG-16 (this study)	**Artifacts detection**	**96.88%**	**95.31%**	**98.44%**

Table 2.7 – Comparison between the results presented in the literature [36] and the ones achieved in this study (in bold).

Gradient-weighted class activation mapping (Grad-CAM) was used to generate heatmaps for the images of the test set that the model classified as artifacts. Figure 2.13a and Figure 2.13b shows hemorrhage occurring during surgery which may be misinterpreted as pathological change. In Figure 2.13c, Figure 2.13d, and Figure 2.13e red areas of the heatmap correspond to prefixation artifacts, the most frequent form of artifact. This form of tissue distortion results from even minimal compression of the tissue by forceps or

other surgical instruments. It includes crush, hemorrhage, splits, and fragmentation. Microscopically, in crush artifacts, the cellular details are not recognizable, and nuclei appear darkly stained and distorted. Figure 2.13f shows tissue fold. Folding of tissue section occur when very thin paraffin sections are forced to stretch unevenly around other structures which have different consistencies. These artifacts appear as darker-stained strands.

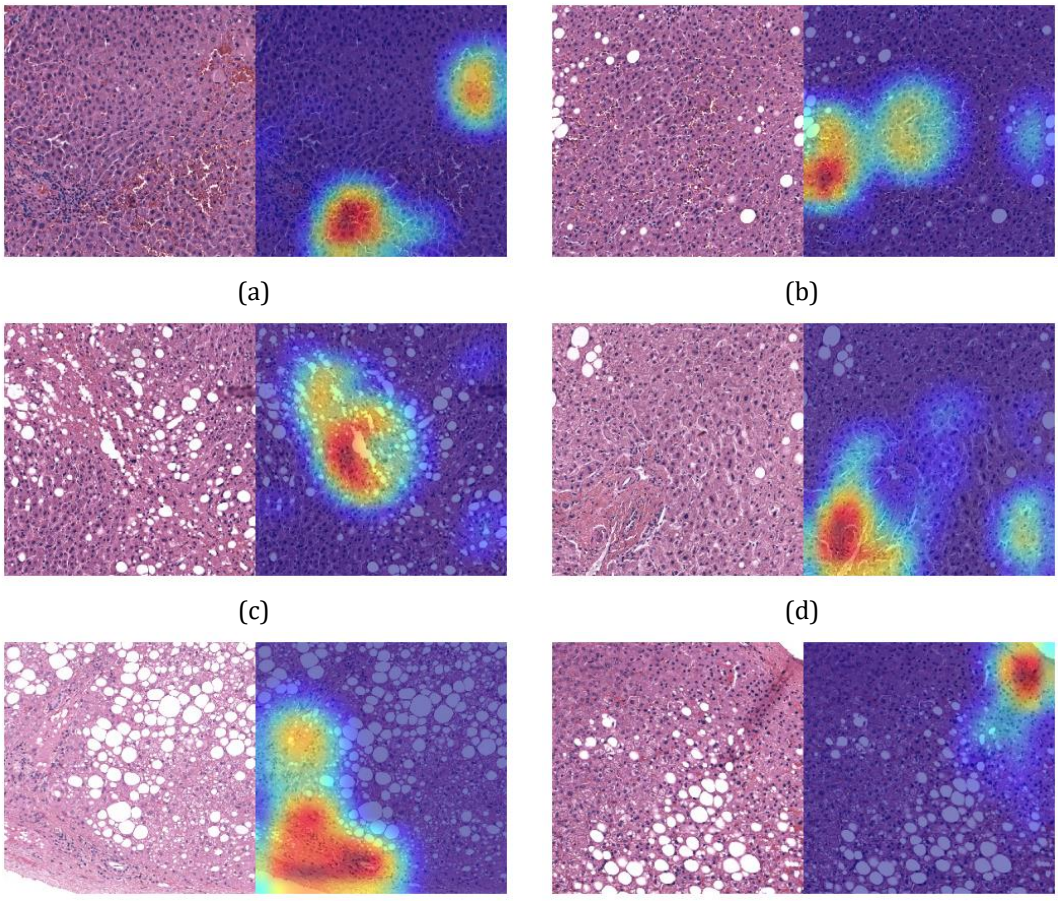


Figure 2.13 – Examples of visual explanation generated by Grad-CAM on patches classified as artifacts. Red areas of heatmaps reflect image regions where the final model searched for artifacts: (a)-(b) surgical hemorrhage, (c)-(e) prefixation artifacts, (f) tissue-fold artifact.

2.5 Discussion

There is a growing interest in the development of computer-aided systems for an automated and objective examination of histological WSIs, as in the case of the hepatic steatosis assessment. However, the robustness of these systems can only be ensured if the image quality is first verified. Currently, this time-consuming task should be performed by a trained pathologist who must manually select the regions of interest that meet the quality standard for further automated analysis.

As things stand at present, the automatic QA of histological images is an issue little or not at all addressed in literature mainly due to the intrinsic complexity and heterogeneity of the elements that can alter the quality of the image in the whole process of slide preparation.

In this study, a method for the automated QA of histological images was developed and applied to the liver biopsy images collected by the ASUGI group for the hepatic steatosis grading. This method is made of two consecutive stages that use different techniques to perform two distinct tasks of different complexity.

The pre-processing stage aims to automatically identify uninformative data within a WSI. These uninformative data are mainly represented by the slide background that may occupy the largest part of an image. This leads to fully or partially empty images when tiles are extracted from WSIs to perform patch-level analysis. Big empty spaces in tiled images may be also represented by large vessels. In the specific case of hepatic steatosis, large vessels should be discarded since they are uninformative and can be misinterpreted by the system.

This relatively simple task was addressed using classical image processing techniques. Results obtained at this stage, reviewed by an experienced pathologist, have demonstrated the effectiveness of the proposed method in automatically identifying uninformative data. The major advantage of this

method is the objectivity with which images lacking sufficient informative data for further analysis are discarded from the dataset.

The artifacts detection stage aims to perform the very complex task of the automatic identification of various types of artifacts. Artifacts are alterations that may be introduced during any stage of slide preparation. Their characteristics are very heterogeneous and their influence on the interpretation of tissue sections depends on the diagnostic question. These factors have led to the choice, if not the need, to use a deep learning-based method, and in particular a CNN, to learn the features rather than handcrafting them.

Despite the complexity of the task, a simple CNN network – the VGG-16 – has proven to be an excellent tool for detecting artifacts. The network, trained after the pre-processing stage, has shown very good performance in terms of accuracy, sensitivity, and specificity compared to the one of trained pathologists. Furthermore, results significantly outperform the performance reached by the model developed in [36] (Table 2.7).

The heatmaps generated by GradCAM confirmed the accuracy of the model in identifying artifacts. They demonstrate how the network actually bases its decision in producing the classification on the presence or not of different types of artifacts. For example, Figure 2.13 shows the ability of the network to correctly recognize surgical hemorrhages, prefixation artifacts, and tissue fold distortions, even when such artifacts are not perceptible to an inexperienced eye.

For the sake of completeness, experiments were also done by training the model directly on the original dataset, without the pre-processing stage, but they did not produce sufficient accuracy and are therefore not addressed here. However, the poor results obtained from training the network on the original dataset demonstrate the validity of the chosen two-stages approach.

The overall method developed in this study allows to perform automatic QA of histological images, selecting accurately and objectively good-quality image tiles from WSIs. Although the method was applied to liver biopsies, there are

prerequisites to extending the proposed methodology to other types of histological images. The promising results obtained demonstrate that classical techniques allow an initial simple “skimming”, while methods based on deep learning can be effectively used for complex tasks such as artifact detection.

This methodology could easily be proposed as the first stage of any computer-aided system for histological image examination in order to automate the entire analysis process and improve accuracy and reproducibility. This could speed up the process and be particularly useful when, for a variety of reasons, a pathologist is not available.

Chapter 3
Development of an AI system for staging the Myopic Traction Maculopathy

3.1 Background and objective

Myopic Traction Maculopathy (MTM) is a complex disease characterized by a wide spectrum of clinical pictures that affects 9% to 34% of eyes with high myopia (refractive error > 6.00 diopters and/or axial length > 26.5 mm) [41].

The term “myopic traction maculopathy” was first coined by Panozzo *et al.* [42] to describe the combination of different tractional forces that act on the retina and fovea in highly myopic eyes leading to macular damages such as macular schisis, macular detachment, and macular holes.

Macular schisis (MS) is an increased thickness of the neurosensory retina in a column-like structure. It can be distinguished in inner macular schisis (I-MS) and outer macular schisis (O-MS). The former starts from the inner nuclear layer to the internal limiting membrane (ILM). The ILM can be detached from the nerve fibers layers and connected to it with a column-like structure. The latter starts

from the outer plexiform layer, that changes in a column-like structure, to the outer nuclear layer.

MS can be associated to macular detachment (MD), a neurosensory detachment of the macula with separation of the photoreceptors from the retinal pigment epithelium. MD can also involve the whole macula with no areas of schisis visible and occasionally expand beyond the macula.

Macular holes can be present as inner lamellar macular hole (I-LMH), a splitting of the inner foveal layers, developing from the ILM and not reaching the photoreceptors, outer lamellar macular hole (O-LMH), a splitting and an interruption of the photoreceptors, and full-thickness macular hole (FTMH), an interruption of all the neuroretinal layers.

For more clarity, the image in Figure 3.1 shows the nomenclature of the layers that can be identified in a healthy retina, some of which were previously named, while the images in Figure 3.2 show some examples of the macular damages that may involve an eye with MTM.

More complete information on the nomenclature, pathogenesis, natural evolution, the prognosis of MTM was offered by the research conducted by Dr. Parolini, currently Director of the Vitreoretinal Service at Eyecare Clinic in Brescia.

After reviewing optical coherence tomography (OCT) images of 281 eyes with MTM that had been operated between 2006 and 2018, Parolini *et al*. introduced the new MTM staging system [44], shorten by the acronym MSS, a comprehensive and practical OCT-based staging system of MTM that has the purpose to help ophthalmologists to understand the progression of this complex disease and better manage their patients.

The system defines the evolution of the disease in a direction perpendicular to the retina and tangential to the retina and the fovea, identifying four retinal patterns, due to centrifugal forces perpendicular to the retinal plane, and three foveal patterns, due to centrifugal forces tangential to the retinal and foveal plane.

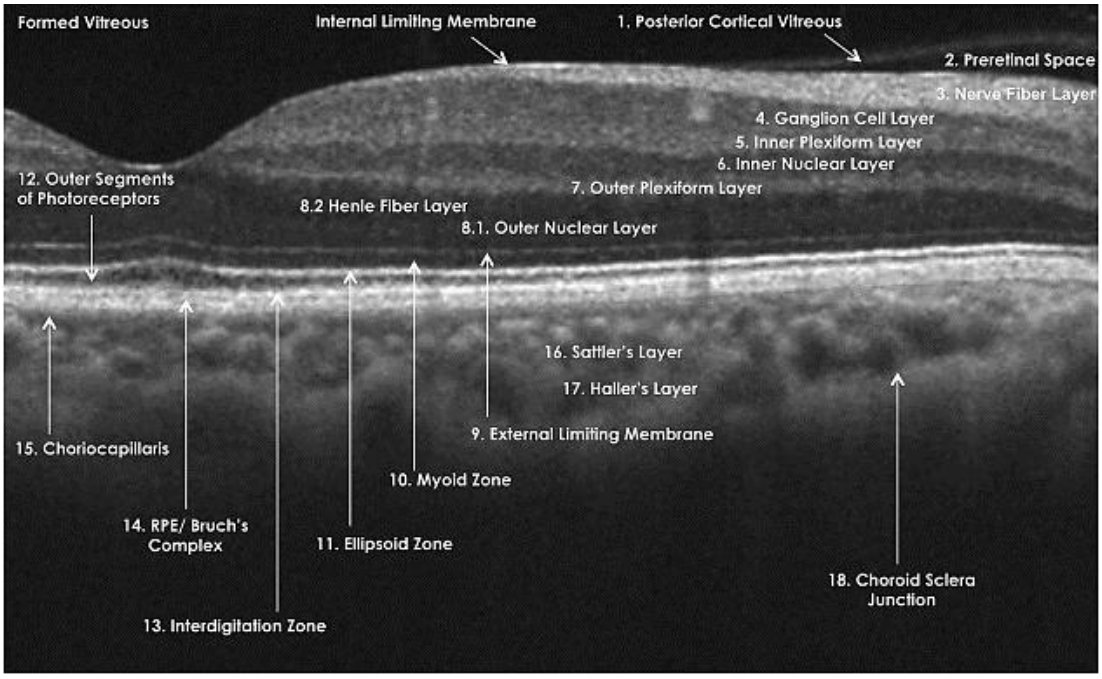


Figure 3.1 – Nomenclature for normal retinal layers seen on spectral domain coherence tomography (SD-OCT) images proposed and adopted by the International Nomenclature for Optical Coherence Tomography Panel [43].

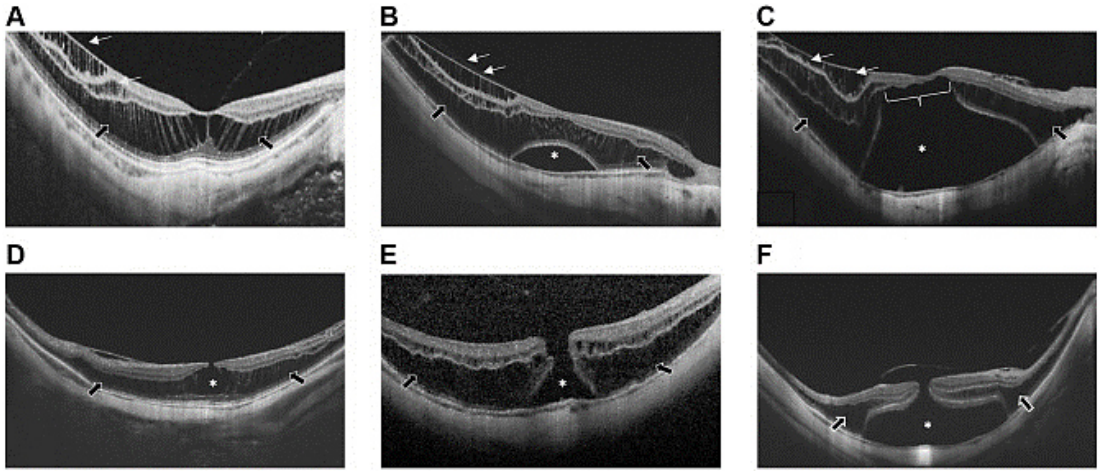


Figure 3.2 – (A) Macular schisis (MS). A separation of retinal layers, which remain connected by cells stretched in multiple columnar structures, appears in both inner layers (white arrows, I-MS) and outer layers (black arrows, O-MS). (B) Macular detachment (asterisk, MD). White arrows show I-MS, black arrows O-MS. (C) MD (asterisk) associated with I-MS (white arrows) and O-MS (black arrows). White line indicates outer lamellar macular hole (O-LMH). (D) Lamellar macular hole (asterisk, LMH) associated with O-MS (black arrows). (E) Full-thickness macular hole (asterisk, FTMH) associated with O-MS (black arrows) (F) FTMH associated with MD (asterisk) and O-MS (black arrows) [41].

Retinal patterns evolve perpendicularly to the retina from I-MS or inner and outer MS (IO-MS) in Stage 1, to predominantly O-MS in stage 2, to MS with MD in stage 3, and to MD in stage 4.

Foveal patterns evolve tangentially to the retina and the fovea from normal foveal profile in stage a, to I-LMH in stage b, and to FTMH in stage c.

O-LMHs may occur in stages 2, 3, and 4 while epiretinal abnormalities may be associated with every stage and contribute to the disease progression.

The retina can evolve from stage 1 to stage 4 and from stage a to stage c simultaneously or separately. The mean time taken to evolve from one stage to the next ranges from weeks to 18 months and is correlated with a decreasing of the best-corrected-visual-acuity (BCVA).

The MSS table is shown in Figure 3.3.

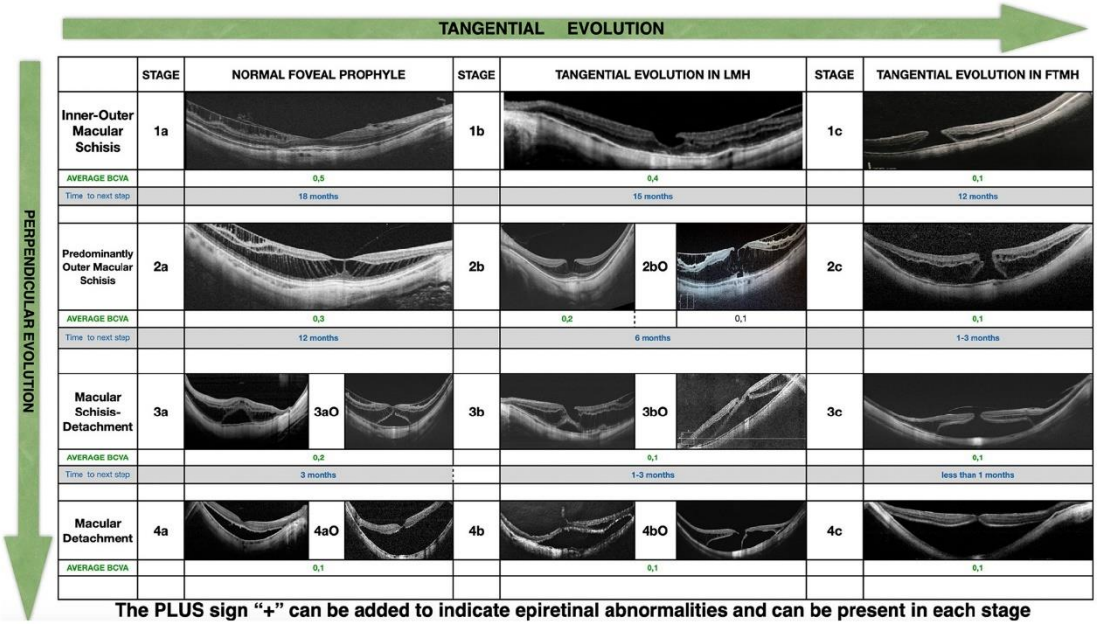


Figure 3.3 – The MTM staging system (MSS) table. The four rows represent the evolution perpendicular to the retina (stages 1–4). The three columns represent the evolution tangential to the retina and the fovea (stages a–c). The presence of O-LMH is marked as O while the presence of epiretinal abnormalities is marked as +.

Details of each MSS stage are given below:

- *Stage 1a*: presence of I-MS or inner-outer macular schisis (IO-MS) and absence of splitting in the fovea. It might be associated with epiretinal abnormalities.
- *Stage 1b*: presence of I-MS or IO-MS and presence of an I-LMH in the fovea. It might be associated with epiretinal abnormalities.
- *Stage 1c*: presence of I-MS or IO-MS and presence of a FTMH.
- *Stage 2a*: presence of predominantly O-MS and absence of splitting in the fovea. It might be associated with an O-LMH and epiretinal abnormalities.
- *Stage 2b*: presence of predominantly O-MS and presence of an I-LMH in the fovea. It might be associated with an O-LMH.
- *Stage 2c*: presence of predominantly O-MS and presence of a FTMH. It might be associated with an O-LMH and epiretinal abnormalities.
- *Stage 3a*: presence of an association of MS and MD and absence of splitting in the fovea. It might be associated with an O-LMH and epiretinal abnormalities.
- *Stage 3b*: presence of an association MS and MD and presence of an I-LMH in the fovea. It might be associated with an O-LMH and epiretinal abnormalities.
- *Stage 3c*: presence of an association of MS and MD and presence of a FTMH. It might be associated with an O-LMH and epiretinal abnormalities.
- *Stage 4a*: presence of a complete MD and absence of splitting in the fovea. It might be associated with an O-LMH and epiretinal abnormalities.
- *Stage 4b*: presence of a complete MD and presence of an I-LMH splitting the fovea. It might be associated with an O-LMH and epiretinal abnormalities.

- *Stage 4c*: presence of a complete MD and presence of a FTMH. It might be associated with an O-LMH and epiretinal abnormalities.

MSS not only serves as a base to define all the distinct types of MTM and their prognosis but also to propose guidelines for treatment. Knowing how forces act on the retina and the fovea allows one to make choices about the surgical treatment to be performed, which should counteract these forces. Therefore, retinal patterns from schisis to detachment could be better solved with a macular buckle (MB) while foveal patterns with splitting into the fovea should be treated with pars plana vitrectomy (PPV) and ILM management. When schisis and/or detachment are combined with a macular hole, MB can be combined with PPV and ILM management [45].

In light of the important clinical implications offered by this research, the idea to develop an AI system to support ophthalmologists in the assessment of eyes with MTM according to the MSS was born. At this early stage of MSS deployment, such a system could assume an important educational value for the assessment of a disease that still represents a diagnostic challenge, encouraging at the same time the use of the MSS as a universally accepted classification system.

This idea took place in the project named MTM AI, born at the beginning of 2022 from the collaboration between Dr. Parolini and O3Enterprise s.r.l. The MTM AI project aims to exploit the great potential of AI, and in particular DL – which in the last few years was increasingly applied to the analysis of ophthalmological images [46][47][48][49][50][51][52][53] –, to develop a system that can automatically process OCT scans of an MTM eye to provide its stage according to the new MTM staging system.

At present, the project is in the phase of Proof of Concept (PoC) and the evaluation of the feasibility of the application of DL methods to the staging of MTM, not yet covered in the literature, is the objective of this research.

3.2 Related works

Although this is the first study that aims to develop an AI system for staging MTM, even more so according to the new staging system, in literature there are several pieces of research that propose DL systems for the identification of different vision-threatening diseases (e.g. age-related macular degeneration, high myopia) and retinal abnormalities (e.g. cystoid macular edema, macular hole, retinoschisis, retinal detachment) from OCT images, which represents the most commonly used imaging modality in ophthalmology. These studies adopt common CNN architectures, such as VGG, ResNet, and Inception, to develop binary or multi-class classifiers for the analysis of OCT images, demonstrating very promising results.

Table 3.1 shows a brief overview of these studies, summarizing models adopted, datasets used, and results reached for the tasks addressed. It is noteworthy that the research proposed in [59] is the only one that applies DL to MTM analysis, although the task is considerably different from the one in this study.

Lee *et al.* [54] demonstrated that DL can be successfully used to distinguish normal OCT images from patients with Age-related Macular Degeneration (AMD). The authors extracted 2.6 million OCT images from two cohorts of patients: normal and AMD. Of those, 80,839 images were selected to train a CNN model, while 20,163 images were used to validate it. The architecture chosen was a modified version of the VGG-16 network. ROC curves were created at image level, macular level and patient level, and the AUCs achieved were 92.78%, 93.83%, and 97.45%, respectively.

Lu *et al.* [55] used four independent ResNet-101 binary classifiers to detect four retinal abnormalities: cystoid macular edema, macular hole, epiretinal membrane and serous macular detachment. The system was trained on 22,017 OCT images with 10-fold cross-validation and tested on 3,317 OCT images,

achieving an AUC of 98.4% when discriminating diseases from normal control. The AUCs of the four binary classifiers were 99.6% for cystoid macular edema, 99.9% for macular hole, 99.8% for epiretinal membrane, and 97.7% for serous macula detachment.

	Year	Task	Model	Dataset	AUCs
[54]	2017	Classification: - Normal - AMD	Modified VGG-16	Splitting: - Train: 80,839 - Val: 20,163 Image type: 11 central OCT slices	92.78% - 97.45%
[55]	2018	Classification: - Normal - Cystoid macular edema - Serous macular detachment - Epiretinal membrane - Macular hole	ResNet -101 (4 binary classifiers)	Splitting: - 10-fold CV: 22,017 - Test: 3,317 Image type: Macular Cube and Optic Disc Cube protocols (Zeiss Cirrus HD-OCT 4000)	97.7% - 99,9%
[56]	2018	Classification: - Normal - Drusen - CNV - DME	Inception-v3	Splitting: - Train: 198,312 (1,000) - Val: 1,000 Image type: horizontal foveal OCT slices	99.87% - 100.0%
[57]	2020	Classification: - Retinoschisis - Macular hole - Retinal detachment - PMCNV - CNV - DME	Inception-ResNet-v2 (4 binary classifiers)	Splitting: - Train: 4,338 - Val: 1,167 - Test: 412 Image type: vertical and horizontal foveal OCT slices (6 mm)	96.1% - 99.9%
[58]	2021	Classification: - Normal - High myopia - Other retinal disease	VGG-16 ResNet-50 Inception-v3	Splitting: - 5-fold CV: 1,200 - Test: 180 Image type: vertical and horizontal foveal OCT slices (9mm)	86.19%, 99.99%, 97.28%
[59]	2021	Classification: - Choroidal thinning - BM defects - SHRM - MTM - DSM	ResNet-101 (5 binary classifiers)	Splitting: - Train: 1874 - Val: 468 - Test: 450 Image type: vertical and horizontal foveal slices (6-8 mm)	92.7% - 97.4%

Table 3.1 – Overview of most interesting studies in the literature that apply DL methods to the identification of different vision-threatening conditions on OCT images.

Kermany *et al.* [56] proposed a deep-learning system for the diagnosis of the most common treatable blinding retinal diseases. Using transfer learning, the authors developed a system able to distinguish normal images from images with choroidal neovascularization (CNV), diabetic macular edema (DME), and drusen with performance comparable to that of human experts. The model used was an Inception-v3 architecture pretrained on the ImageNet dataset and fine-tuned on 198,312 OCT images. The model was evaluated on 1,000 images achieving an AUC of 99.9% at distinguishing urgent referrals (CNV or DME) from drusen and normal exams. To compare transfer learning performance on limited data, the authors also trained the mode with only 1,000 images randomly selected from each class. Under this condition, the ROC curves distinguishing urgent referrals from drusen and normal images had an AUC of 98.8%. When comparing CNV, DME or drusen from normal exams the system reached AUCs of 100.0%, 99.87%, and 99.96%, respectively.

Li *et al.* [57] adopted the Inception-ResNet-v2 architecture to train four independent binary classifiers in order to identify four vision-threatening conditions in high-myopia: retinoschisis, macular hole, retinal detachment and pathological myopic choroidal neovascularization (PMCNV). The dataset consisted in 5,505 images obtained from 1,048 patients, divided into training set (4,338 images, 80%) and validation set (1,167 images, 20%). An independent test set composed by 412 images was used to evaluate the model. For this dataset, ROC curves created at eye level had AUCs of 96.1% for retinoschisis, 99.9% for macular hole, 98.6% for retinal detachment, and 99.4% for PMCNV.

Choi *et al.* [58] trained and validated three DL models to classify normal, high myopia and other retinal diseases groups based on horizontal and vertical OCT images. The authors adopted VGG-16, ResNet-50, and Inception-v3 architectures as a backbone and developed a single-column and a multiple-column models to perform image classification. The single-column model used a single OCT image, processing separately vertical and horizontal images, while the multiple-column

model considered both vertical and horizontal OCT images simultaneously concatenating features from each backbone. The models were trained and validated through five-fold cross-validation on 1,200 images from 436 patients (600 eyes). The diagnostic performance was evaluated on the test dataset composed by other 180 images (90 eyes). The best performance was achieved by the ResNet-50 single-column model. The AUCs of the three DL single-column models were 99.93% for vertical VGG-16, 99.65% for horizontal VGG-16 models, 100.00% for vertical ResNet-50, 99.95% for horizontal ResNet-50, 96.08% for vertical Inception-v3, and 88.88% for horizontal Inception-v3. The AUCs of multi-column models were 86.19%, 99.99%, 97.28% for VGG-16, ResNet-50 and Inception-v3, respectively.

Ye *et al.* [59] adopted an ResNet-101 architecture to train five independent binary classifiers in order to identify five myopic maculopathies: macular choroidal thinning (MCT), macular Bruch membrane (BM) defects, subretinal hyper-reflective material (SHRM), myopic traction maculopathy, and dome-shaped macula (DSM). The models were trained and validated on 2,342 OCT images from 1,041 patients with pathologic myopia and tested on an independent test dataset composed of 450 images from 297 patients. On the test dataset, the AUCs were 92.7% for MCT, 93.8% for BM defects, 92.7% for SHRM, 97.4% for MTM, and 95.5% for DSM.

3.3 Materials and Methods

3.3.1 Dataset

In this study, OCT images from all the patient affected by MTM that were seen between July 2020 and September 2022 by Dr. Parolini were retrospectively collected. Signed informed consent documentation was obtained from all participants. OCT was performed using the Canon Xephilio OCT-A1 and the Canon Xephilio OCT-S1 instruments. Only OCT images acquired with the OCT scanning

protocol defined by twelve radial slices of the posterior pole centered on the fovea with scan length from 6 mm to 23 mm were selected.

1,668 OCT images from 139 eyes of 88 patients with MTM were initially screened. Among these, 466 images were excluded for poor acquisition quality or other factors hampering a proper assessment.

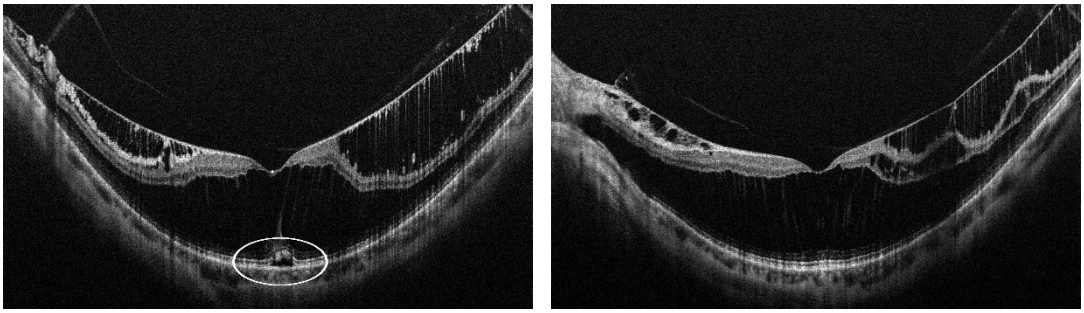
The remaining 1,202 images were labeled by two clinical experts according to the MSS table. Each OCT image was considered independently and classified through the observation of the retinal patterns (stages 1-4) and the foveal patterns (stages a-c). In this initial phase, also the presence of O-LMH (O) and epiretinal abnormalities (+) was considered. Table 3.2 shows the summary of the images collected for AI development.

Retinal pattern	**Foveal pattern (O-LMH/+)**			**Total (O-LMH/+)**
	a	**b**	**c**	
1	211 (0/26)	115 (0/16)	12 (0/0)	338 (0/42)
2	408 (0/82)	139 (0/56)	8 (0/0)	555 (0/132)
3	211 (56/53)	34 (0/3)	9 (0/0)	254 (56/56)
4	12 (0/2)	8 (0/2)	35 (0/0)	55 (0/4)
Total	842 (56/163)	296 (0/71)	64 (0/0)	1202 (56/234)

Table 3.2 – Characteristics of the collected dataset.

It is worth to be noted that different slices of the same OCT scan, and thus of the same eye, may show different foveal and retinal patterns. The final MTM stage of an eye is given by the highest stage in the perpendicular and tangential evolution of the disease identified through the observation of the whole OCT scan.

An example is given in Figure 3.4, which shows two different slices of the same eye with MTM. The slice in Figure 3.4a shows macular schisis and detachment (white circle) with normal fovea (stage 3a) while the slice in Figure 3.4a shows O-MS in the macula with normal fovea (stage 2a). The final MSS stage of the eye was 3a.



(a) (b)

Figure 3.4 – Different OCT slices of the same eye with MTM. (a) Slice 10/12: MS associated with MD and intact fovea (stage 3a), (b) Slice 12/12: O-MS and intact fovea (stage 2a). The final MSS stage was 3a.

This is the reason that dictated the choice to include in the dataset all the twelve slices acquired during a radial OCT scan rather than selecting only the vertical and horizontal foveal OCT slices as often proposed in the literature [56]-[59]. However, this choice exactly reflects what is also done in clinical practice during the staging of an eye with MTM.

The OCT scans included in this study were sent from the OCT instruments to a PACS reserved specifically for the research. All the DICOM images stored were fully anonymized. To be processed by the AI system, the images were retrieved through WADO-RS requests and converted into JPEG Lossless format.

3.3.1.1 CNN models

Since in MTM the evolution perpendicular to the retina (stages 1-4) occurs separately from the evolution tangential to the retina and the fovea (stages a-c), two independent CNN models were proposed: the former, hereinafter referred as to the *MTMrp* model, classifies retinal patterns (stages 1-4), the latter, hereinafter referred as to the *MTMfp* model, classifies foveal patterns (stages a-c). O-LMHs and epiretinal abnormalities, which for the purpose of the MTM staging are of lesser importance, were not taken into account at this phase of the study.

In the proposed AI system, each network processes a single OCT image and performs a multi-class classification task. The process is iterated until each slide

of an OCT scan is processed. All the classification outputs are collected and the highest stage is selected, providing a classification at eye level. The final MSS stage is given by the association of the eye-level classifications for retinal patterns and foveal patterns respectively, as in Figure 3.5.

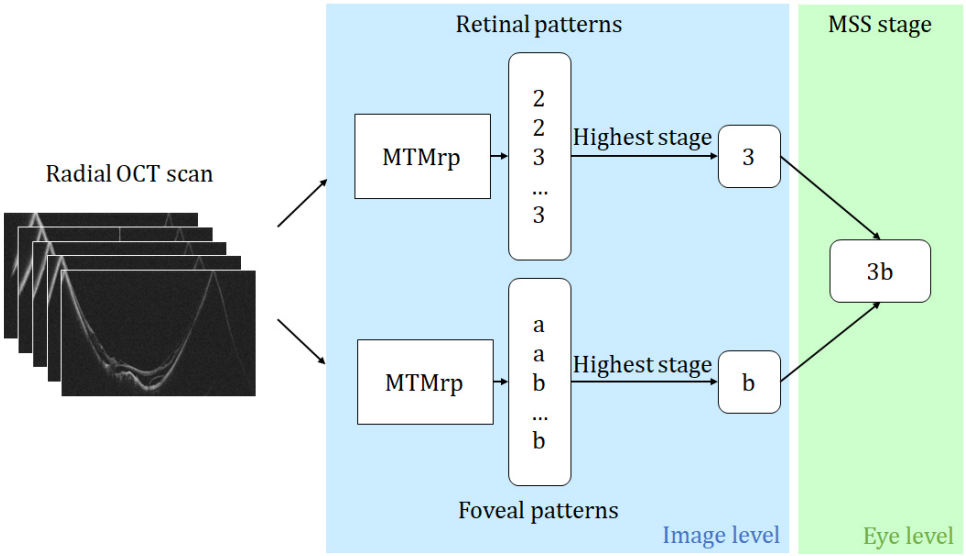


Figure 3.5 – Overview of the AI system for staging MTM.

In light of the promising results given by literature on the use of VGG, ResNet, and Inception architectures for OCT images classification, VGG-16, ResNet-50, ResNet-101, Inception-v3, and InceptionResNet-v2 were considered as the CNN backbone for both the MTMrp and MTMfp models. All the networks were pretrained on ImageNet dataset and fine-tuned. During transfer learning, initialized weights were not frozen, i.e. all the networks hidden layers were kept learnable. A single fully-connected layer with Softmax activation function was added after the original feature extracting layers of each CCN backbone to perform the final multiclass classification.

All the images were resized to 512 × 512 pixels, due to memory limitations, and pre-processed according to the specific type of input pre-processing expected by each model. Images passed as input data to VGG-16, ResNet50 and

ResNet101 were converted from RGB to BGR zero-centering each color channel with respect to the ImageNet dataset while images fed to Inception-v3 and Inception-ResNet-v2 were scaled so that all the values were in the [-1, 1] range.

During each training epoch, the same number of new variations of original images were produced through data augmentation techniques. These variations included horizontal flipping and rotation.

The network was trained for 25 epochs in mini-batches of 8 samples. The Adam optimizer was used with an initial learning rate of 10^{-4} (Table 3.3).

Trainable parameters	
VGG-16	14,847,044
ResNet-50	24,060,164
ResNet-101	43,078,404
Inception-v3	22,293,924
Inception-ResNet-v2	54,670,692

Hyperparameters	
Optimizer	Adam
Learning rate	10^{-4}
Loss function	Categorical cross-entropy
Batch size	8
Epochs	25
Regularization	-
Weight initialization	ImageNet
Dataset splitting	5-fold cross-validation

Table 3.3 – Trainable parameters and hyperparameters of the networks.

Five-fold cross-validation was used to train and validate the models on the whole dataset. The dataset was randomly divided into five subsets, taking care to preserve the percentage of samples for each class and, when possible, including data from each patient in a single subset. Five-fold cross-validation was performed by training the models with four of these subsets and validating them with the remaining data. The process was iterated five times by changing the training and validation subsets. The dataset splitting for the five-fold cross-validation is summarized in Table 3.4.

The extremely limited number of images for each class, particularly for stage 4 (i.e. complete MD) and stage c (i.e. FTMH), did not allow the creation of a test

set of independent data for the evaluation of the performance of a final model. It was therefore necessary to limit the study to a comparison of the different models on cross-validation results, rather than selecting the model that performs better at this level and evaluate it on independent data. In addition, to avoid information leakages and strongly biased results, special care was taken to not maximize performance through hyperparameters tuning on cross-validation results. Precisely, the choice of the hyperparameters used in this study was based on previous independent research conducted on OCT images obtained from public datasets [60].

MTMrp								
Fold	**training**				**validation**			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
1	270 (28%)	444 (46%)	203 (21%)	44 (4%)	68 (28%)	111 (46%)	51 (21%)	11 (4%)
2	270 (28%)	444 (46%)	203 (21%)	44(4%)	68 (28%)	111 (46%)	51 (21%)	11 (4%)
3	270 (28%)	444 (46%)	204 (21%)	44(4%)	68 (28%)	111 (46%)	50 (21%)	11 (4%)
4	270 (28%)	444 (46%)	203 (21%)	44(4%)	67 (28%)	111 (46%)	51 (21%)	11 (4%)
5	271 (28%)	444 (46%)	203 (21%)	44(4%)	67 (28%)	111 (46%)	51 (21%)	11 (4%)
MTMfp								
Fold	**training**			**validation**				
	a	**b**	**c**	**a**	**b**	**c**		
1	673 (70%)	237 (24%)	51 (5%)	169 (70%)	59 (24%)	13 (5%)		
2	673 (70%)	237 (24%)	51 (5%)	169 (70%)	59 (24%)	13 (5%)		
3	674 (70%)	236 (24%)	51 (5%)	168 (70%)	60 (24%)	12 (5%)		
4	674 (70%)	237 (24%)	51 (5%)	168 (70%)	59 (24%)	13 (5%)		
5	674 (70%)	237 (24%)	51 (5%)	168 (70%)	59 (24%)	13 (5%)		

Table 3.4 – Summary of the class distribution over the five subsets used to perform five-fold cross-validation for both the MTMrp and MTMfp models.

3.3.1.2 Framework

All the code was written in Python, using TensorFlow as DL framework. Training and evaluation were performed using an NVIDIA GeForce GTX 1070 graphic processing unit (GPU) on a 64-bit computer processor with an Intel(R) Core (TM) i7-8700 CPU @ 3.20GHz.

Training execution times are reported in Table 3.5:

CNN Backbone	MTMrp	MTMfp
VGG-16	02:25:13	02:23:07
ResNet-50	02:11:52	02:09:29
ResNet-101	03:27:53	03:26:19
Inception-v3	02:05:14	02:05:19
Inception-ResNet-v2	04:06:02	04:05:25

Table 3.5 – Training execution time for each model. Time is in the *hh:mm:ss* format.

3.4 Results

Five-fold cross-validation was used to compare the performance of the different CNN backbones adopted for both the classification of retinal patterns, i.e. stages 1-4 (MTMrp model), and the classification of the foveal patterns, i.e. stages a-c (MTMfp model).

As mentioned before, the choice of comparing different models on cross-validation results rather than selecting a final model and evaluating it on a test set was dictated by the impossibility of creating an additional independent data set containing representative images of each MTM stage. Despite this serious issue, the study still allowed the evaluation of the feasibility of different CNN architectures in staging MTM from OCT images, which was the main goal of the PoC stage of the project.

Table 3.6 reports the five-fold cross-validation results. Micro-average ROC curves and per-class ROC curves were created for each CNN backbone of the MTMrp and MTMfp models and the averages of the AUC values obtained through the five cross-validation iterations were computed.

The micro-average AUCs for the MTMrp model ranged from 94.73%, for the VGG-16 CNN backbone, to 99.51% for the Inception-ResNet-v2 CNN backbone, while the micro-average AUCs for the MTMfp model varied from 95.55%, for the VGG-16, to 98.97%, for the Inception-ResNet-v2.

MTMrp					
CNN backbone	**Micro average AUC**	**Stage 1 AUC**	**Stage 2 AUC**	**Stage 3 AUC**	**Stage 4 AUC**
VGG-16	0.947±0.023 (0.916–0.979)	0.959±0.028 (0.920–0.997)	0.902±0.037 (0.849–0.956)	0.915±0.033 (0.870–0.961)	0.982±0.018 (0.957–1.000)
ResNet-50	0.994±0.005 (0.987–1.000)	0.997±0.002 (0.994–1.000)	0.987±0.008 (0.976–0.998)	0.987±0.009 (0.975–0.999)	1.000±0.000 (1.000-1.000)
ResNet-101	0.994±0.005 (0.989–1.000)	0.998±0.003 (0.994–1.000)	0.988±0.010 (0.974–1.000)	0.986±0.010 (0.972–1.000)	1.000±0.000 (1.000-1.000)
Inception-v3	0.995±0.004 (0.989–0.999)	0.997±0.0039 (0.991–1.000)	0.992±0.007 (0.982–1.000)	0.988±0.012 (0.971–1.000)	1.000±0.001 (0.999–1.000)
Inception-ResNet-v2	0.995±0.002 (0.992–0.998)	0.997±0.003 (0.994–1.000)	0.989±0.004 (0.984–0.995)	0.991±0.005 (0.984–0.998)	1.000±0.000 (1.000–1.000)

MTMfp				
CNN backbone	**Micro average AUC**	**Stage 1 AUC**	**Stage 2 AUC**	**Stage 3 AUC**
VGG-16	0.956±0.013 (0.935–0.953)	0.921±0.028 (0.883–0.959)	0.919±0.024 (0.885–0.953)	0.905±0.101 (0.765–1.000)
ResNet-50	0.983±0.007 (0.972–0.993)	0.964±0.016 (0.942–0.987)	0.962±0.019 (0.9364–0.988)	0.976±0.022 (0.946–1.000)
ResNet-101	0.985±0.003 (0.966–0.985)	0.9738±0.006 (0.965–0.982)	0.967±0.009 (0.956–0.982)	0.979±0.019 (0.952–1.000)
Inception-v3	0.987±0.005 (0.980–0.995)	0.975±0.012 (0.958–0.991)	0.973±0.012 (0.956–0.989)	0.976±0.019 (0.950–1.000)
Inception-ResNet-v2	0.990±0.005 (0.982–0.997)	0.980±0.011 (0.965–0.996)	0.978±0.009 (0.965–0.991)	0.988±0.011 (0.973–1.000)

Table 3.6 – Five-fold cross-validation results for each of the CNN backbone, both for the MTMrp and the MTMfp models. Data are given as mean ± standard deviation (95% CI).

Per-class AUCs were in the range of 95.87%-99.71% for stage1, 90.21%-99.17% for stage2, 91.52%-99.09% for stage3, 98.18%-100.00% for stage4, 92.08%-98.01% for stage a, 91.89%-97.81% for stage b, and 90.54%-98.80% for stage c. The VGG-16 CNN backbone revealed the worst performance, especially in the classification of foveal patterns. All the remaining CNN backbones achieved an average AUC greater than 99% and 98% for the MTMrp and the MTMfp models, respectively. The CNN backbone which performed better in the

classification of both retinal patterns and foveal patterns was the Inception-ResNet-v2.

Micro-average ROC curves for the MTMrp and the MTMfp models are shown in Figure 3.6 and Figure 3.8, respectively. Confusion matrices for the MTMrp and the MTMfp models were also computed and shown in Figure 3.7 and Figure 3.9, respectively.

Grad-CAM visualization heatmaps were provided in order to get visual feedback on how the models identified the features to differentiate retinal patterns (stages 1-4) and foveal patterns (stages a-c).

By way of example, Figure 3.10 shows some Grad-CAM visualization heatmaps produced passing OCT images to the MTMrp model (on the left) and the MTMfp model (on the right) with Inception-ResNet-v2 CNN backbone. Figure 3.10a depicts an eye with MTM at stage 1b: the MTMrp model correctly identified the presence of I-MS while the MTMfp model detected the presence of an I-LMH in the fovea. Figure 3.10b represents an eye with MTM at stage 2b: the MTMrp model searched for O-MS while the MTMfp model identified an I-LMH. The eyes in Figure 3.10c and Figure 3.10d have MTM at stage 3a and 3b, respectively. The red areas in the heatmaps generated by the MTMrp model correspond to MS associated with MD, while the red areas in the heatmaps generated by the MTMfp model highlight intact fovea and I-LMH, respectively. Figure 3.10e depicts an eye at stage 4c: the MTMrp model identified the presence of a complete MD while the MTMfp model detected FTMH.

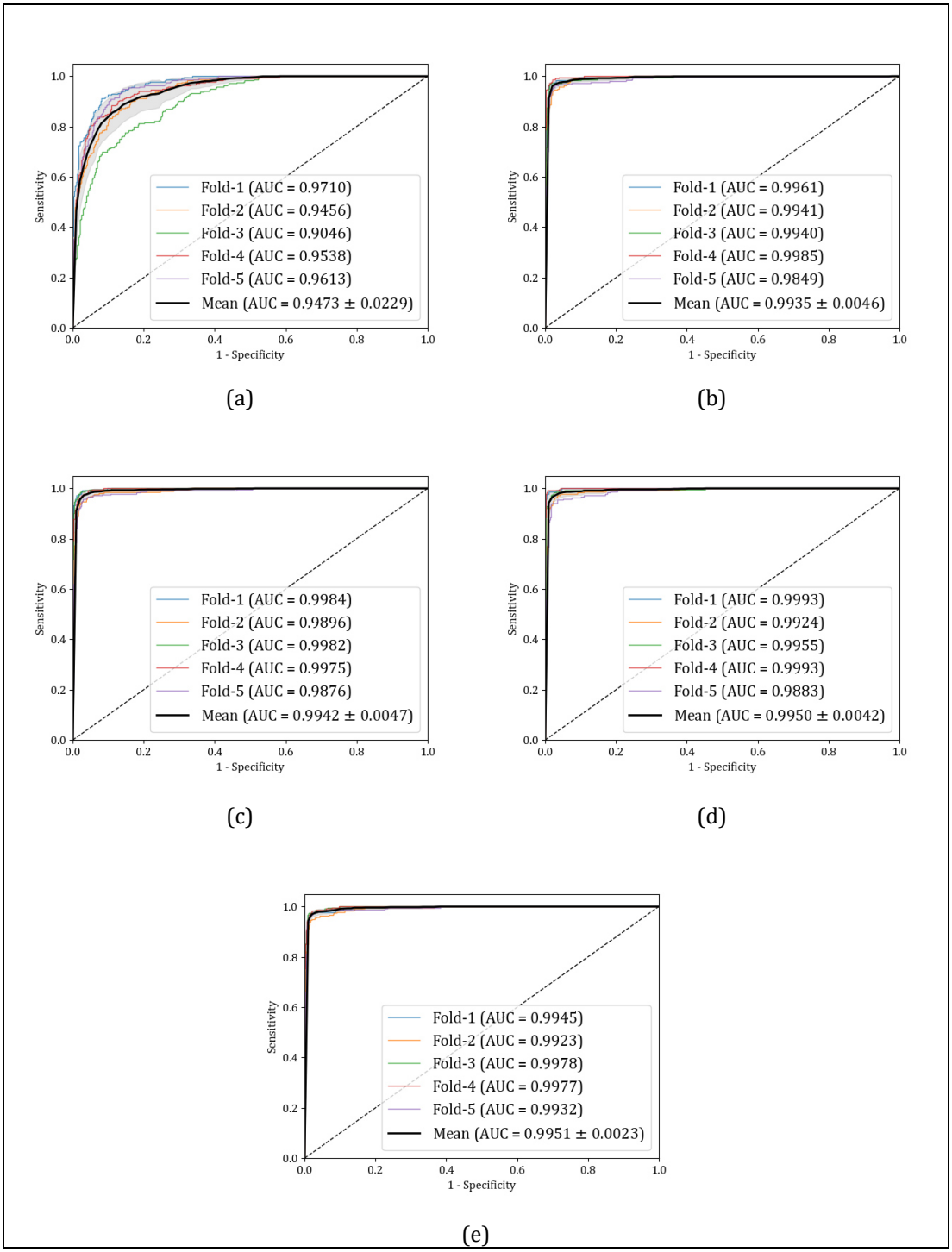


Figure 3.6 – Micro-Average ROC curves for the MTMrp model (stages 1-4): (a) VGG-16, (b) ResNet-50, (c) ResNet-101, (d) Inception-v3, (e) ResNet-Inception-v2.

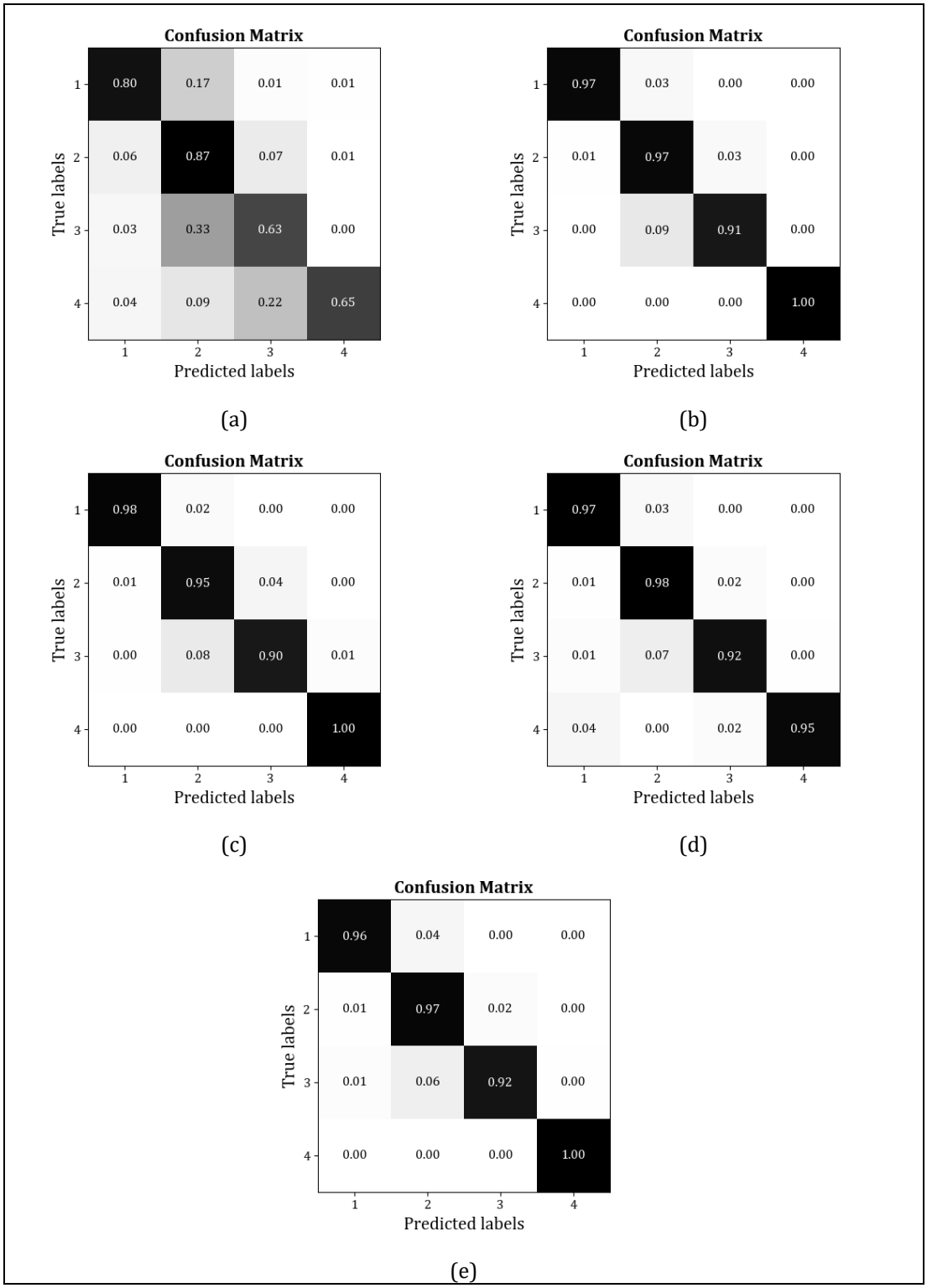


Figure 3.7 – Confusion matrices for the MTMrp model (stages 1-4): (a) VGG-16, (b) ResNet-50, (c) ResNet-101, (d) Inception-v3, (e) ResNet-Inception-v2.

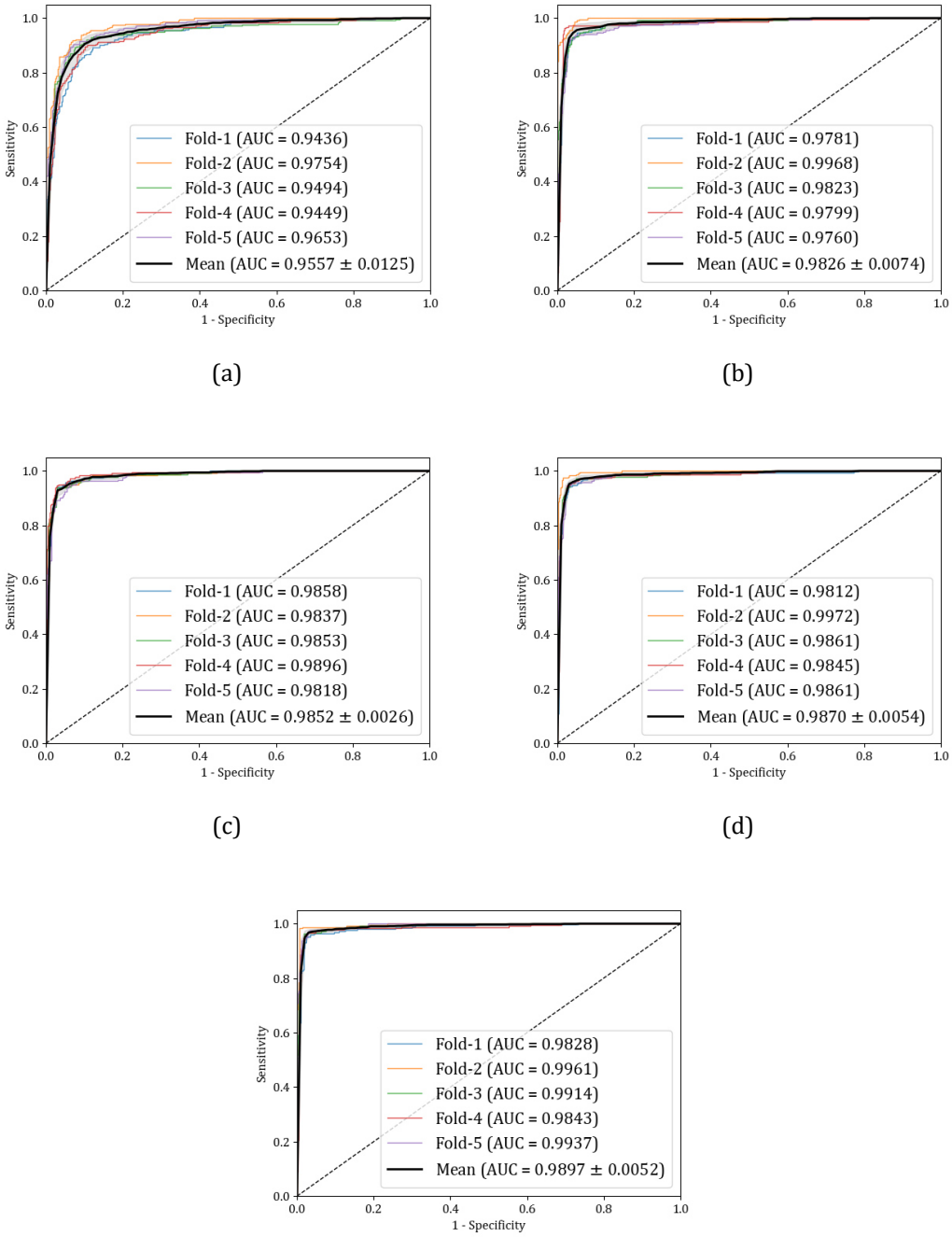


Figure 3.8 – Micro-Average ROC curves for the MTMfp model (stages a-c): (a) VGG-16, (b) ResNet-50, (c) ResNet-101, (d) Inception-v3, (e) ResNet-Inception-v2.

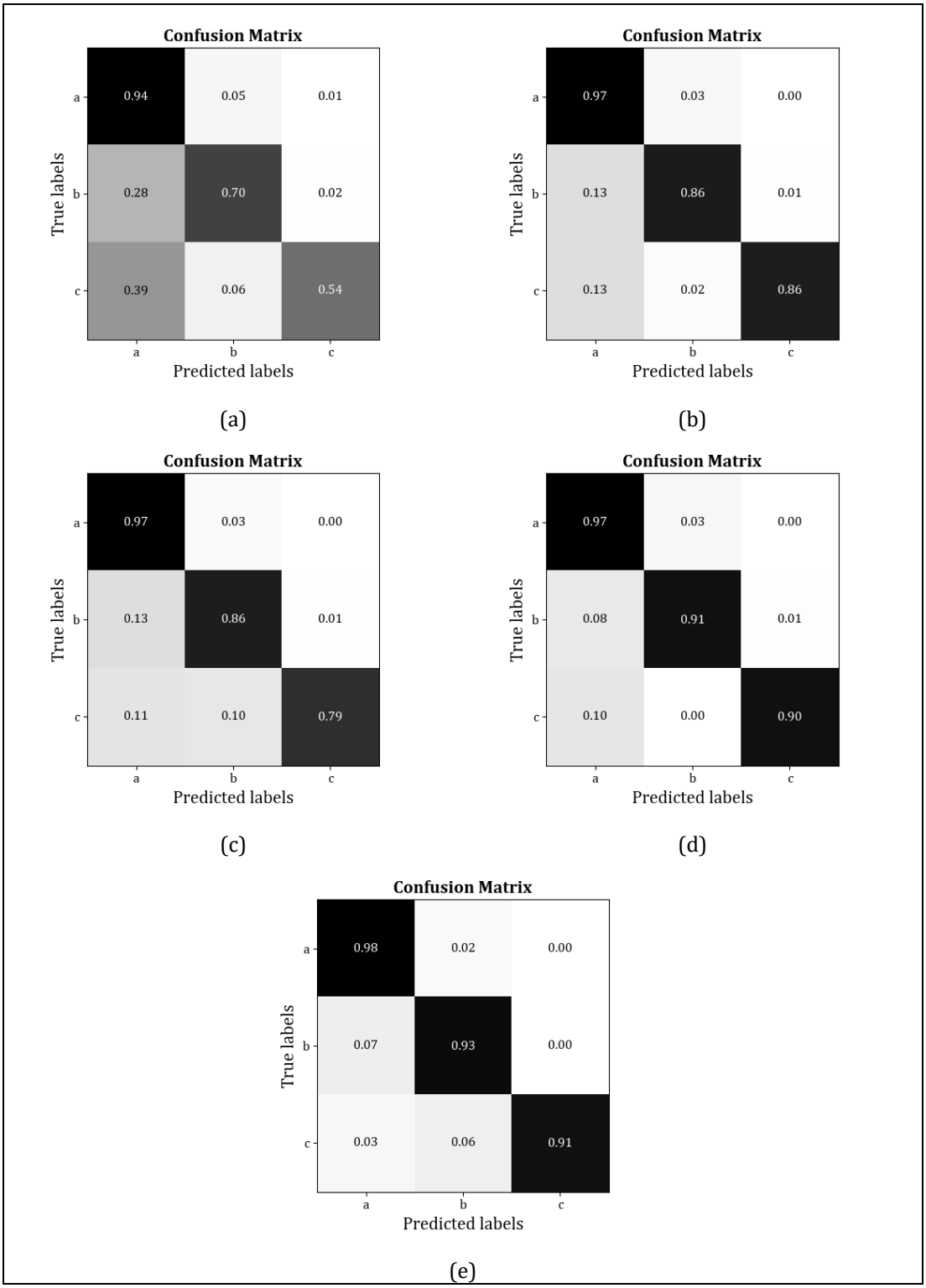


Figure 3.9 – Confusion matrices for the MTMfp model (stages a-c): (a) VGG-16, (b) ResNet-50, (c) ResNet-101, (d) Inception-v3, (e) ResNet-Inception-v2.

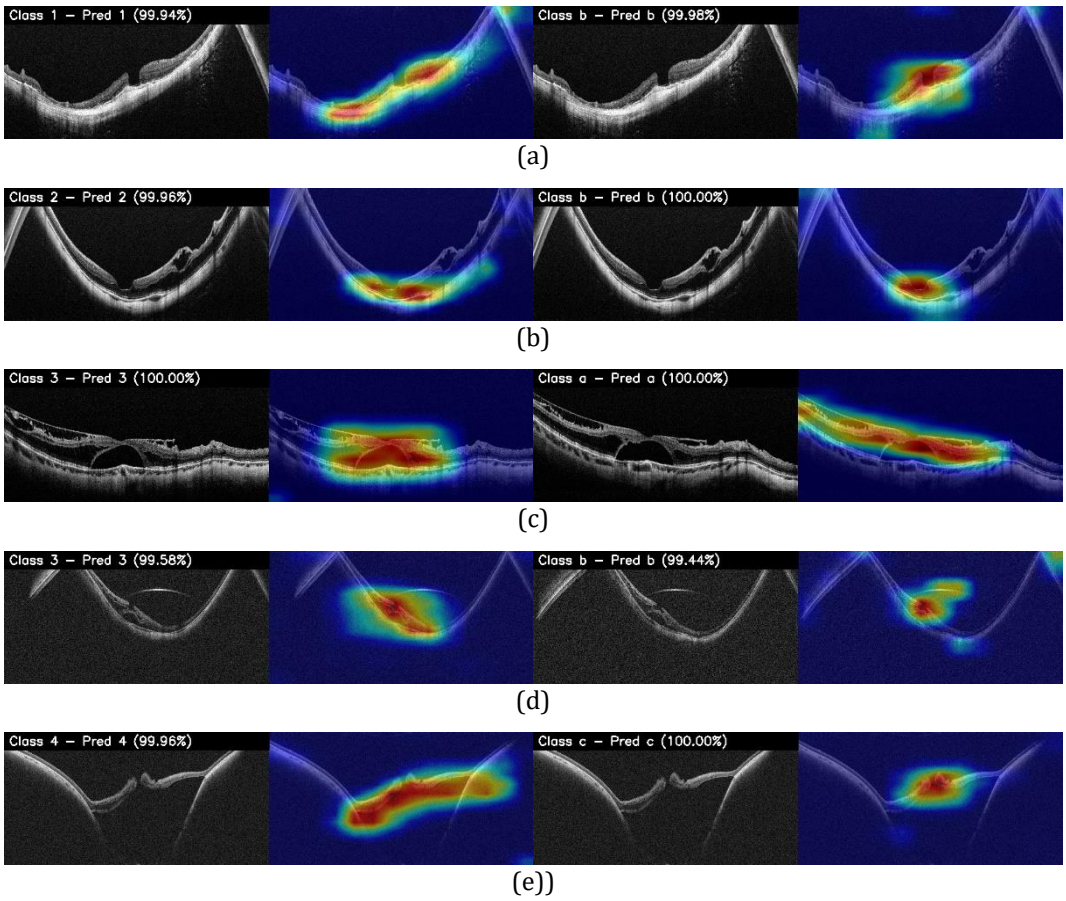


Figure 3.10 – Examples of visual explanation heatmaps generated by Grad-CAM on OCT images (Inception-ResNet-v2). Red areas of heatmaps reflect image regions where the final model searched for retinal (left) and foveal (right) patterns. (a) MTM eye with stage 1b, (b) MTM eye with stage 2b, (c) MTM eye with stage 3a, (d) MTM eye with stage 3b, (e) MTM eye with stage 4c.

3.5 Discussion

AI systems, in particular those based on deep CNNs, have been shown to achieve very good performance in the analysis of OCT images, being able to accurately distinguish various vision-threatening diseases [54]-[59]. This study demonstrated for the first time that two DL models based on OCT images can effectively differentiate the stages of evolution of a complex disease such as myopic traction maculopathy (MTM).

MTM is characterized by a wide spectrum of clinical pictures that may affect eyes with high myopia. These pictures include macular schisis (MS), macular detachment (MD), and macular holes (MH) at various levels of severity. Detailed information on the evolution of MTM was provided by the new MTM staging system (MSS) [44]. MSS describes four retinal patterns and three foveal patterns as the evolution of the disease in a direction perpendicular and tangential to the retina and the fovea, respectively. Moreover, this staging system offers information on the prognosis of MTM and sets the foundation for surgical treatments.

An AI system that can accurately and automatically identify each stage of MTM evolution according to the new MSS is thought to be a useful tool to support all those ophthalmologists who are faced with such a complex disease, improving patient management. This system is also expected to assume an important educational value, especially at the current time, when the pathogenesis, the natural evolution, and the prognosis of this disease are still scarcely known even among retinal experts.

In this study, OCT images of eyes with MTM were retrospectively collected and annotated by experts according to the MSS. The obtained dataset was used to train two models, one for staging retinal patterns and one for staging foveal patterns. Five CNN architectures were used as a backbone for each model and their performance was compared. Among them, the Inception-ResNet-v2 architecture demonstrated to achieve greater performance on both tasks.

As revealed in the ROC curves in Figure 3.6 and Figure 3.8, the Inception-ResNet-v2 backbone showed very high AUCs in differentiating inner MS from outer MS, MS associated with MD, and MD (average AUC=99.51%), and intact fovea from inner lamellar MH and full-thickness MH (average AUC=98,97%). This performance is comparable, or even better, to that currently reported in the literature, although the tasks were different. Is worth mentioning the binary classifiers proposed in [57], which achieved AUCs of 96.1%, 99.9%, and 98.6% in

identifying generical retinoschisis, macular hole, and retinal detachments, respectively.

This study, in addition, confirmed that GradCAM heatmaps allow the visualization of both retinal and foveal patterns, distinguishing among them. As illustrated in Figure 3.10, the heatmaps generated by the model for retinal patterns classification showed red areas in correspondence with inner and outer retinal layers, while red areas in the heatmaps generated by the model for foveal pattern classification are correctly concentrated on the fovea.

Despite these promising results, this study had limitations. First, the final evaluation of the selected model can be performed only after collecting a new independent data set (i.e., the test set). Second, a larger dataset which consider also images from other OCT instruments should be collected and used for training and testing. Third, in this study only retinal and foveal patterns were considered. Outer lamellar macular hole and epiretinal abnormalities, which might be associated with each stage of MTM, were excluded from the analysis due to dataset limitation and should be taken into account in feature research.

However, considering that this study represented a PoC and its primary objective was the evaluation of the feasibility of DL methods in addressing the MTM staging task, the models here proposed could be considered to have the requirements to form the basis for the development of an AI system with high clinical utility.

Conclusion

The aim of the research activity carried out during the Ph.D. course with higher education and research apprenticeship program between O3 Enterprise s.r.l. and the University of Trieste was the application of deep-learning based methods to the analysis of medical images in problems impacting clinical practice.

Deep learning, and in particular the class of deep neural networks that goes under the name of Convolutional Neural Networks (CNNs), has shown promising results in various fields of medical imaging, becoming dominant in the analysis of radiological images and spreading to other image-centric specialties such as pathology and ophthalmology.

In the field of pathology, various deep learning systems for the analysis of histological images were developed with the purpose to support pathologists in tasks such as the detection and grading of tumors and other diseases. The limit of these systems, however, is often represented by the need for the manual intervention of a pathologist to confirm the quality of the images before they are processed, according to criteria dependent on the diagnostic question. A novel method to overcome this time-consuming task through the automatic and objective quality assessment of histological images is proposed in Chapter 2. This

method is made of two consecutive stages that use different techniques to perform tasks of increasing complexity. While classical image processing techniques were demonstrated to be a simple but effective tool for the automatic and objective identification of uninformative data, such as slide background, a CNN model proved to be an excellent tool for addressing the complex task of the identification of artifacts, showing good results in terms of accuracy. The methodology proposed was applied to liver biopsies, but it is expected to be easily extended to other histological images to be used as the first stage of any computer-aided system for histological image examination, speeding up the analysis process and improving reproducibility.

In ophthalmology, the adoption of deep learning-based methods to Optical Coherence Tomography (OCT) has shown promising results for the identification of different retinal diseases. In Chapter 3 deep learning-based methods were for the first time investigated with the purpose of developing an AI system for the analysis of OCT images of eyes with myopic traction maculopathy (MTM) in order to distinguish the different retinal and foveal patterns that characterize the natural evolution of this complex diseases. These retinal and foveal patterns are described by the recently proposed new MTM staging system (MSS) which offers insights into the pathogenesis and natural evolution of MTM and provides guidelines for better management of the patients. In the research here presented, different CNN architectures were studied and compared among them, achieving very promising results on the classification of both retinal and foveal patterns. Although this research is only at an early stage, the results obtained laid the foundations for the development of an AI system for staging MTM according to MSS that is expected to have important clinical and educational implications in practice.

Both these studies demonstrated that deep learning-based methods, and in particular CNN architectures, are effective in accurately distinguishing various and complex features in different fields of medical imaging, even when the tasks

are extremely difficult to interpret even by experts. Moreover, the more advanced application of Explainable AI, and in particular of the GradCAM technique tailored to work with this kind of architecture, proved the ability of these methods in performing the task for which they were trained. No less important, GradCAM heatmaps provided physicians with a reference and this is expected to help the understanding of the mechanisms behind the predictions performed by deep learning models and reduce distrust of them.

Bibliography

[1] Krizhevsky, A., Sutskever, I., & Hinton, G .E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger (ed.), *Advances in Neural Information Processing Systems 25* (pp. 1097--1105). Curran Associates, Inc..

[2] Soffer, S., Ben-Cohen, A., Shimon, O., Amitai, M. M., Greenspan, H., & Klang, E. (2019). Convolutional Neural Networks for Radiologic Images: A Radiologist's Guide. *Radiology*, *290*(3), 590–606.

[3] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[4] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[5] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.

[6] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021).

Review of deep learning: concepts, CNN architectures, challenges, applications, future directions.*Journal of Big Data, 8*.

[7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[9] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[11] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

[12] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).

[13] Dimitriou, N., Arandjelović, O., & Caie, P. D. (2019). Deep learning for whole slide image analysis: an overview. *Frontiers in medicine6*, 264.

[14] Niazi, M. K., Parwani, A. V., & Gurcan, M.N. (2019). Digital pathology and artificial intelligence. *The lancet oncology*, *20*(5), e253-e261.

[15] Van der Laak, J., Litjens, G., & Ciompi, F. (2021). Deep learning in histopathology: the path to the clinic.*Nature medicine*,*27*(5), 775-784.

[16] Chen, Wei-Ming, et al. "Deep Learning-Based Universal Expert-Level Recognizing Pathological Images of Hepatocellular Carcinoma and beyond." *Frontiers in medicine* 9 (2022).

[17] Khened, Mahendra, et al. "A generalized deep learning framework for whole-slide image segmentation and analysis." *Scientific reports* 11.1 (2021): 1-14.

[18] Nam, David, et al. "Artificial intelligence in liver diseases: improving diagnostics, prognostics and response prediction." *JHEP Reports* (2022): 100443.

[19] Castera, L., Friedrich-Rust, M., & Loomba, R. (2019). Noninvasive assessment of liver disease in patients with nonalcoholic fatty liver disease. *Gastroenterology*, *156*(5), 1264-1281.

[20] Tiniakos, D. G., Vos, M. B., & Brunt, E. M. (2010). Nonalcoholic fatty liver disease: pathology and pathogenesis. *Annual Review of Pathology: Mechanisms of Disease*, *5*, 145-171.

[21] Pierantonelli, I., & Svegliati-Baroni, G. (2019). Nonalcoholic fatty liver disease: basic pathogenetic mechanisms in the progression from NAFLD to NASH. *Transplantation*, *103*(1), e1-e13.

[22] Wilkins, T., Tadkod, A., Hepburn, I., & Schade, R. R. (2013). Nonalcoholic fatty liver disease: diagnosis and management. *American family physician*, *88*(1), 35-42.

[23] Kleiner, D. E., Brunt, E. M., Van Natta, M., Behling, C., Contos, M. J., Cummings, O. W., Ferrell, L. D., Liu, Y. C., Torbenson, M. S., Unalp-Arida, A., Yeh, M., McCullough, A. J., Sanyal, A. J., & Nonalcoholic Steatohepatitis Clinical Research Network. (2005). Design and validation of a histological scoring system for nonalcoholic fatty liver disease. *Hepatology*, *41*(6), 1313-1321.

[24] French METAVIR Cooperative Study Group, & Bedossa, P. (1994). Intraobserver and interobserver variations in liver biopsy

interpretation in patients with chronic hepatitis C. *Hepatology*, *20*(1), 15-20.

[25] Nativ, N. I., Chen, A. I., Yarmush, G., Henry, S. D., Lefkowitch, J. H., Klein, K. M., Maguire, T. J., Schloss, R., Guarrera, J.V., Berthiaume, F., & Yarmush, M. L. (2014). Automated image analysis method for detecting and quantifying macrovesicular steatosis in hematoxylin and eosin–stained histology images of human livers. *Liver Transplantation*, *20*(2), 228-236.

[26] Munsterman, I. D., van Erp, M., Weijers, G., Bronkhorst, C., de Korte, C. L., Drenth, J. P., van der Laak J. A., & Tjwa, E. T. (2019). A novel automatic digital algorithm that accurately quantifies steatosis in NAFLD on histopathological whole-slide images. *Cytometry Part B: Clinical Cytometry*, *96*(6), 521-528.

[27] Forlano, R., Mullish, B. H., Giannakeas, N., Maurice, J. B., Angkathunyakul, N., Lloyd, J., Tzallas, A. T., Tsipouras, M., Yee, M., Thursz, M. R., Goldin, R. D., & Manousou, P. (2020). High-throughput, machine learning–based quantification of steatosis, inflammation, ballooning, and fibrosis in biopsies from patients with nonalcoholic fatty liver disease. *Clinical Gastroenterology and Hepatology*, *18*(9), 2081-2090.

[28] Roy, M., Wang, F., Vo, H., Teng, D., Teodoro, G., Farris, A. B., Castillo-Leon, E., Vos, M. B., & Kong, J. (2020). Deep-learning-based accurate hepatic steatosis quantification for histological assessment of liver biopsies. *Laboratory Investigation*, *100*(10), 1367-1383.

[29] Salvi, M., Molinaro, L., Metovic, J., Patrono, D., Romagnoli, R., Papotti, M., & Molinari, F. (2020). Fully automated quantitative assessment of hepatic steatosis in liver transplants. *Computers in Biology and Medicine*, *123*, 103836.

[30] Heinemann, Fabian, et al. "Deep learning-based quantification of NAFLD/NASH progression in human liver biopsies." *Scientific Reports* 12.1 (2022): 19236.

[31] Aeffner, F., Zarella, M. D., Buchbinder, N., Bui, M. M., Goodman, M. R., Hartman, D. J., Lujan, G. M., Molani, M. A., Parwani, A. V., Lillard, K., Turner, O. C., Vemuri, V. N., Yuil-Valdes, A. G., & Bowman, D. (2019). Introduction to digital image analysis in whole-slide imaging: a white paper from the digital pathology association. *Journal of pathology informatics*, *10*(1), 9.

[32] Taqi, S. A., Sami, S. A., Sami, L. B., & Zaki, S. A. (2018). A review of artifacts in histopathology. *Journal of oral and maxillofacial pathology: JOMFP*, *22*(2), 279.

[33] Ameisen, D., Deroulers, C., Perrier, V., Bouhidel, F., Battistella, M., Legrès, L., Janin, A., Bertheau, P., & Yunès, J. B. (2014, April). Automatic image quality assessment in digital pathology: from idea to implementation. In *IWBBIO* (pp. 148-157).

[34] Kothari, S., Phan, J. H., & Wang, M. D. (2013). Eliminating tissue-fold artifacts in histopathological whole-slide images for improved image-based prediction of cancer grade. *Journal of pathology informatics*, *4*(1), 22.

[35] Vanderbeck, S., Bockhorst, J., Komorowski, R., Kleiner, D. E., & Gawrieh, S. (2014). Automatic classification of white regions in liver biopsies by supervised machine learning. *Human pathology*, *45*(4), 785-792.

[36] Foucart, A., Debeir, O., & Decaestecker, C. (2018, November). Artifact identification in digital pathology from weak and noisy supervision with deep residual networks. In *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)* (pp. 1-6). IEEE.

[37] French METAVIR Cooperative Study Group, & Bedossa, P. (1994). Intraobserver and interobserver variations in liver biopsy interpretation in patients with chronic hepatitis C. *Hepatology*, *20*(1), 15-20.

[38] Dimitriou, N., Arandjelović, O., & Caie, P. D. (2019). Deep learning for whole slide image analysis: an overview. *Frontiers in medicine*, *6*, 264.

[39] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[40] Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, *3*(1), 32-35.

[41] Frisina, R., Gius, I., Palmieri, M., Finzi, A., Tozzi, L., & Parolini, B. (2020). Myopic traction maculopathy: diagnostic and management strategies. *Clinical Ophthalmology (Auckland, NZ)*, *14*, 3699.

[42] Panozzo, G., & Mercanti, A. (2004). Optical coherence tomography findings in myopic traction maculopathy. *Archives of ophthalmology*, *122*(10), 1455-1460.

[43] Staurenghi, G., Sadda, S., Chakravarthy, U., & Spaide, R. F. (2014). Proposed lexicon for anatomic landmarks in normal posterior segment spectral-domain optical coherence tomography: the IN•OCT consensus. *Ophthalmology*, *121*(8), 1572-1578.

[44] Parolini, B., Palmieri, M., Finzi, A., Besozzi, G., Lucente, A., Nava, U., ... & Frisina, R. (2021). The new myopic traction maculopathy staging system. *European journal of ophthalmology*, *31*(3), 1299-1312.

[45] Parolini, B., Palmieri, M., Finzi, A., & Frisina, R. (2021). Proposal for the management of myopic traction maculopathy based on the new MTM staging system. *European journal of ophthalmology*, *31*(6), 3265-3276.

[46] Sogawa, Takahiro, et al. "Accuracy of a deep convolutional neural network in the detection of myopic macular diseases using swept-source optical coherence tomography." *Plos one* 15.4 (2020): e0227240.

[47] Zhang, Chenchen, et al. "Applications of Artificial Intelligence in Myopia: Current and Future Directions." *Frontiers in Medicine* 9 (2022).

[48] Du, Ran, and Kyoko Ohno-Matsui. "Novel Uses and Challenges of Artificial Intelligence in Diagnosing and Managing Eyes with High Myopia and Pathologic Myopia." *Diagnostics* 12.5 (2022): 1210.

[49] Wu, Zhenquan, et al. "Predicting Optical Coherence Tomography-Derived High Myopia Grades From Fundus Photographs Using Deep Learning." *Frontiers in Medicine* 9 (2022).

[50] Wang, Ruonan, et al. "Efficacy of a Deep Learning System for Screening Myopic Maculopathy Based on Color Fundus Photographs." *Ophthalmology and Therapy* (2022): 1-16.

[51] Lu, Li, et al. "AI-model for identifying pathologic myopia based on deep learning algorithms of myopic maculopathy classification and “plus” lesion detection in fundus images." *Frontiers in cell and developmental biology* (2021): 2841.

[52] Du, Ran, et al. "Validation of soft labels in developing deep learning algorithms for detecting lesions of myopic maculopathy from optical coherence tomographic images." *The Asia-Pacific Journal of Ophthalmology* 11.3 (2022): 227-236.

[53] He, Xiaoying, et al. "Development of a deep learning algorithm for myopic maculopathy classification based on OCT images using transfer learning." *Frontiers in Public Health* 10 (2022).

[54] Lee, C. S., Baughman, D. M., & Lee, A. Y. (2017). Deep Learning Is Effective for the Classification of OCT Images of Normal Versus Age-Related Macular Degeneration. Ophthalmology Retina, 1, 322-327.

[55] Lu, W., Tong, Y., Yu, Y., Xing, Y., Chen, C., & Shen, Y. (2018). Deep learning-based automated classification of multi-categorical abnormalities from optical coherence tomography images. *Translational vision science & technology*, *7*(6), 41-41.

[56] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., ... & Zhang, K. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, *172*(5), 1122-1131.

[57] Li, Y., Feng, W., Zhao, X., Liu, B., Zhang, Y., Chi, W., ... & Lin, H. (2022). Development and validation of a deep learning system to screen vision-threatening conditions in high myopia using optical coherence tomography images. *British Journal of Ophthalmology*, *106*(5), 633-639.

[58] Choi, K. J., Choi, J. E., Roh, H. C., Eun, J. S., Kim, J. M., Shin, Y. K., ... & Kim, S. J. (2021). Deep learning models for screening of high myopia using optical coherence tomography. *Scientific reports*, *11*(1), 1-11.

[59] Ye, X., Wang, J., Chen, Y., Lv, Z., He, S., Mao, J., ... & Shen, L. (2021). Automatic screening and identifying myopic maculopathy on optical coherence tomography images using deep learning. *Translational vision science & technology*, *10*(13), 10-10.

[60] Pace, T., Degan, N., Giglio R., Tognetto D., Accardo A. (2022) A Deep Learning Method for Automatic Identification of Drusen and Macular Hole from Optical Coherence Tomography. In B. Séroussi et al. (Eds.) *Studies in Health Technology and Informatics Vol. 294.Challenges of Trustable AI and Added-Value on Health* (pp. 565-566) European Federation for Medical Informatics (EFMI) and IOS Press.