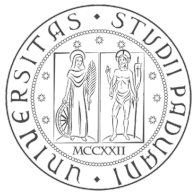

Università degli Studi di Padova

Facoltà di Ingegneria

An investigation into deep learning for locating the optic disc in scanning laser ophthalmoscope retinal images

Candidate

Stefano Gennari

Supervisor

Professor Andrea Facchinetti

Co-supervisor

Professor Emanuele Trucco

Academic year 2018/2019

Abstract

The optic disc (OD) is a key anatomical structure in the retina for monitoring the progression of glaucoma and plays a fundamental role as a landmark in automatic screening systems. Most of the work for the automatic detection and segmentation of the OD has been focusing on fundus camera images. In this work, we present a deep learning approach for OD detection and segmentation in scanning laser ophthalmoscope (SLO) images. The core of the method consists of a convolutional neural network (CNN) inspired by the U-net architecture which has been largely applied in the field of semantic segmentation of biomedical images. To tackle the limited availability of ground truth images for training the network we divided the learning process into two phases: a first phase where the net is trained on a data set of SLO images labeled by an automatic algorithm and a second training phase on a data set where medical annotations for ground truth were provided. We evaluate the performances of our method by comparing the automatic results with medical annotations on a test set of 20 SLO images. The algorithm reaches an accuracy, in terms of Dice-Søresen coefficient, of 0.91, achieving comparable results with the other methods proposed for solving the same task. Furthermore, we compare the resulting contours with those obtained by a validated OD algorithm on registered fundus camera images, and we discuss the ophthalmologists' consensus in indicating the OD contour both in SLO and fundus images.

Sommario

Il disco ottico è una struttura retinica chiave per il monitoraggio del glaucoma e svolge un ruolo fondamentale come punto di riferimento nei sistemi di screening automatico. La maggior parte della ricerca sulla locazione e segmentazione automatica del disco ottico si è concentrata su immagini retiniche acquisite con *fundus camera*. In questo lavoro, presentiamo un nuovo approccio, basato sul *deep learning*, per la locazione e segmentazione automatica del disco ottico in immagini del fondo oculare SLO (*Scanning Laser Ophthalmoscope*). Il nucleo del metodo consiste in una rete neurale convoluzionale (CNN) ispirata all'architettura U-net la quale è diffusamente usata nell'ambito di segmentazione automatica di immagini biomediche. Per ovviare alla limitata disponibilità di immagini di *ground truth* necessarie per allenare la rete neurale abbiamo diviso il processo di apprendimento in due fasi: una prima fase in cui la rete è allenata su un insieme di immagini SLO dove il disco ottico è indicato da un algoritmo automatico e una seconda fase di allenamento della rete effettuata utilizzando un insieme di immagini provviste di annotazioni mediche del disco ottico. La qualità dell'algoritmo viene valutata tramite il computo del coefficiente di Dice-Søresen medio tra risultati automatici e le annotazioni mediche, tale coefficiente calcolato su 20 immagini di test risulta uguale a 0.91; risultato comparabile con quelli ottenuti da altri metodi proposti per risolvere lo stesso compito. Inoltre, in questo lavoro, conduciamo un'analisi del consenso degli oftalmologi nell'indicare il contorno del disco ottico comparando le annotazioni mediche di immagini SLO e fundus. Infine, tramite la registrazione di immagini fundus rendiamo possibile il confronto tra immagini della stessa retina acquisite con le due diverse strumentazioni.

Contents

Chapter 1

Retinal Imaging

1.1 About this chapter

In this chapter, we will introduce the basic anatomical notions of the eye, particularly, we will focus on the retina and the structures that are possible to investigate with the modern imaging techniques. We will briefly describe such techniques and provide to the readers the motivations that have led to the developing of the method presented in this thesis.

1.2 Retina

The retina is the inner surface of the human eye, in an adult, it is approximately a sphere of 22mm diameter (Figure 1.1) which mainly consists of a series of tissue layers composed of neurons and supporting cells. In the retina, we can isolate different anatomical structures, such as the macula, the fovea, and the optic disc. The macula is a region with a diameter around 0.5cm which peculiarity is to host most of the photoreceptors, at its centre we find a small, approximately circular, depression (≈1,5mm) known as the fovea. The fovea is responsible for our sharpest vision and is where the density of the receptors reach its peak. All the receptors in the retina are connected to the innermost layer, the retinal nerve fibre layer (RNFL), which is linked to the optic nerve, the latter leaves the eye at the level of the optic disc (OD). Finally, all the retinal surface, except for the macula, is supplied by a rich vasculature system of venules and arterioles.

1.3 Retinal imaging techniques

Nowadays three principal non-invasive techniques allow to investigate the retina and its structures, namely: fundus camera imaging, scanning laser ophthalmoscopy (SLO) and Optical coherence tomography (OCT).

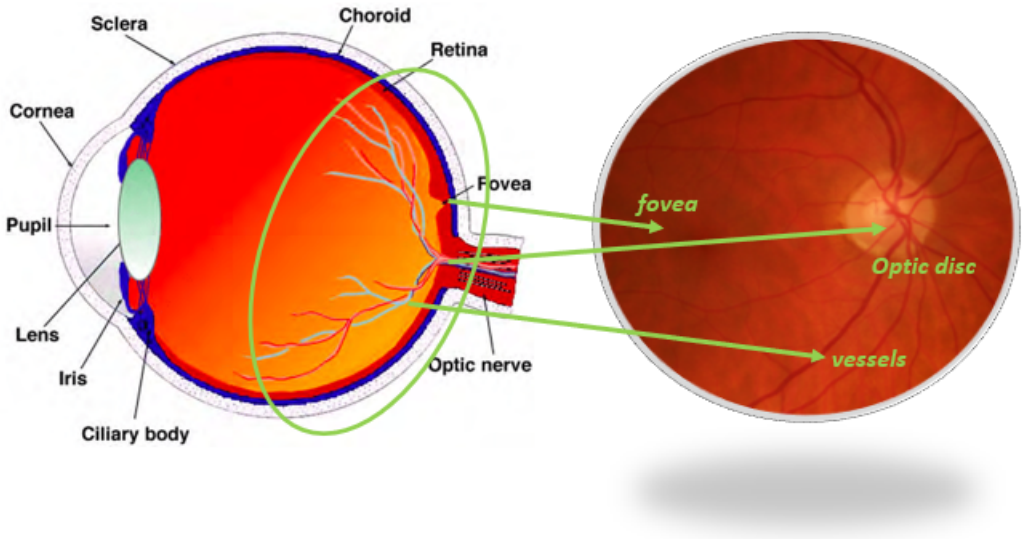

Figure 1.1: *Left, anatomy of the adult human eye, sagittal view showing the main structures (image taken from [20]). Right, fundus camera image centered on the optic disc.*

1.3.1 Fundus camera

Fundus imaging generates an RGB image of the retina through a system that consists of a specialized low-power microscope and an attached camera. The image is acquired while the patient sits with his/her chin in rest and forehead placed against a bar. The operator set the focus, align the camera and presses the shutter, then the fundus is illuminated by a flash and the image acquired. This image is a magnified picture of the retina with an angle of view that varies between 30°, 45° or 60°. A larger field of view (FOV) can be achieved by composing multiple images acquired at different fixation points. Also, images of higher quality can often be achieved by dilating the pupils with mydriatic eye drops to enlarge the FOV of the fundus. Current image resolutions are around 3000x3000 pixels. It is possible to enhance the contrast of the vessels or highlights damaged regions via fluorescein angiography (FA) or indocyanine green angiography. The receives an intravenous injection of a fluorescent dye while the retina is illuminated with light at a certain frequency that fluoresces light of another colour where the dye is present. Using this method is also possible to study the fluid dynamics of the blood in the vessels by looking to the dynamics of the dye in the vasculature. In Figure 1.2.a, an example of a retinal image acquired with this technique.

1.3.2 Scanning laser ophthalmoscopy (SLO)

SLO [28] exploits infra-red light for acquiring the image of the retinal fundus. SLO is a confocal imaging technique, a laser beam scans slice by slice the target area of the retina, the reflected light is collected by a confocal pinhole. The main advantage of SLO technique is that the FOV can vary between 15° to 200° degree (ultra wide field of view UWFOV).

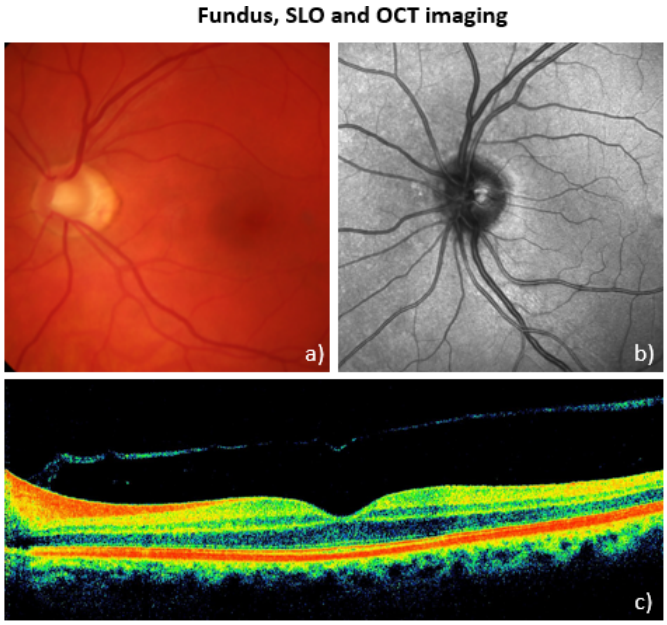

Figure 1.2: *a) Detail of a fundus camera image, where is possible to see the optic disc (bright circular shape in the centre), the vasculature and the fovea (dark smoothed spot on the right. b) Details of a SLO image. It is possible to see the OD (dark circular spot), the vasculature and part of the fovea, on the right. c) Detail of an OCT image, cross-section of the retinal fundus. OCT image taken from [26].*

SLO works well with fluorescein angiography, because of the narrow band of wavelength used by the laser beam the contrast of the acquired images is higher w.r.t. fundus images. The typical resolution of the acquisition is 3000x2800. This techniques is more expensive and often confined to the research field or ophthalmology clinics. SLO is easy to use less invasive for the patient due to the lower cost in time required for the acquisition (compared to the fundus camera technique) and the fact that the there is no need of the use of the flash. In addition, SLO can be used in combination with OCT to reduce the noise in the images acquired from the latter. In Figure 1.2.b, an example of a SLO image of the retina.

1.3.3 Optical coherence tomography (OCT)

Optical Coherence Tomography [30] generates cross-sectional images by analyzing the time delay and magnitude change of low coherence light as it is reflected by ocular tissues. Without entering the details this technique allows to acquire several cross-sectional images of the retinal tissues that combined together can result in a volumetric image. Cross-sectional visualization is an extremely powerful tool in the identification and assessment of retinal abnormalities. In Figure 1.2, an example of an image acquired with this technique.

1.4 Retinal biomarkers

The eye is the only part of the human body in which, thank to the aforementioned technologies, we can directly asses to and see a rich part of the circulatory system. With OCT we can investigate the retinal nerve fibre layer and the head of the optic nerve which is directly connected to the central nervous system. For this reason, the retina or parts of it can represents an important biomarker for many, both retinal and systemic, diseases. Some

links between retinal features and pathologies are well known. For example, systemic hypertension and hypertensive retinopathy are recognizable in fundus images, where are usually associated with visible retinal damage, increasing of the tortuosity in the vasculature and narrowing of the arteriole [20]. Moreover, associations between retinals quantitative measurements and strokes risk, cardiovascular diseases, diabetic retinopathy and many others have been reported [20],[21]. The work presented in this thesis has been conducted in the context of the VAMPIRE (Vascular Assessment and Measurement Platform for Images of the REtina) project which aim is to develop a software application for efficient automatic or semi-automatic quantification of retinal vessels properties in order to provide efficient and reliable detection of retinal landmarks (optic disc, retinal zones, main vasculature), and quantify key parameters used frequently in investigative studies.

1.4.1 Optic disc

The appearance of the optic disc (OD) is important for evaluating and monitoring the progression of glaucoma [22],[23]. In fact, in a healthy retina, the optic disc can be flat or can present slight cupping. In a retina affected by glaucoma, in most cases, the intra-ocular pressure increases producing a further cupping of the optic disc, for this reason, is useful to measure and monitoring the cup-to-disc-ratio that is the ratio between the optic cup and the optic disc diameters, in Figure 1.3.a an example. Another important biomarker for glaucoma is the Peripapillary atrophy (PPA), a form of outer retinal atrophy that abuts the optic disc and it is usually divided into two regions, (α) and beta (β). In fundus, it appears as a blurred region surrounding the OD boundaries while in SLO images appears as a white halo. in Figure 1.3.b an example of PPA in the fundus image.

These are some of the reasons why a reliable automatic detection and identification of the OD contour is useful. In fact, as will be shown in the chapter "Medical annotations" the inter-observer consensus in defining the OD contour can be very low leading to identifications of different contours between different ophthalmologists. Furthermore, is useful to get a reliable localization of the optic disc to have a landmark for many other image analysis tasks such as vessel zones identification, multi-modal images registration, vessels tracking or fovea localization.

Figure 1.3: *a) Images from [27], on the right: Cup/Disc ratio; on the left: Glaucomatous cupping, change of colour and contour. b) Example of optic disc with presence of PPA, alpha and beta zones underlined.*

1.5 Summary

Retinal imaging offers a non-invasive and inexpensive way to access a rich part of the microvasculature and to optic nerve which is directly wired to the central nervous system. For this reason, the retina assures an important source of biomarkers for many systemic diseases. There are three main techniques for acquiring image retinal images: fundus camera, SLO and OCT. Among the structures that we can observe in retinal images we find the optic disc (OD), which features and quantitative measurements represent important indicators for evaluating the progression or the risk of glaucoma. OD automatic detection and segmentation can be very useful for extracting reliable measurements of the latter and for detection, localization and measurement of the other retinal structures.

Chapter 2

Theoretical tools

2.1 About this chapter

In this chapter, all the theoretical notions required to understand the presented method will be explored. We will start introducing the basic concepts of machine learning that are necessary for understanding the fundamentals of deep learning which is in turn what our method is based on.
About deep learning, we will introduce the neural networks (NNs) and then move to the description of the convolutional neural networks (CNNs) mainly focusing on the details directly related to our work. We will proceed and present the idea of transfer learning and the different criteria used in this thesis for evaluating algorithms performances. We end the chapter introducing image registration and briefly illustrating two methods used for this thesis.

2.2 Machine learning

As the words suggest, we can speak of "machine learning" every time a machine learn something, but first we have to agree on what "to learn" means. For the goal of this work we will state that "to learn", in a general context, means to experience something and be able to use this experience to drive a judgment or a prediction. For example, if we see a dancing red fire and we touch it, we discover that fire is hot and to touch it wasn't really a good idea. Then, if, after the bad experience, we see Santa Claus dancing at the mall we will think that, because he is red and is dancing, probably it is hot. In this case, we would have learned something wrong, but we have learned something: do not touch red and dancing things.
When we move the context to the machines, things don't change very much. In fact, what is experience if not a collection of data?
And what is a judgment if not a function of the experience?

The aim of machine learning techniques then, is to emulate the learning process. Which means: starting from a set of data (the experience), build functions able to return valuable

judgments on it or on new data. Formally, we can talk about learning for a "computer program" if:

Definition 2.2.1. *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure L, if its performance at tasks in T, as measured by L, improves with experience E.* [1]

Machine learning tasks are usually separated in (at least) two main branches:

- **Supervised learning:** Given a labeled data set we want to find a function that maps the data into the corresponding labels, and we want this function to be useful to predict the label of novel unlabeled data.
- **Unsupervised learning:** the input given is not labeled and the goal of the algorithm is to infer a function to describe hidden structure or pattern in the input.

The tools utilized in this work are related to the only field of supervised learning, then, to better understand what a labeled data set is, what kind of functions we are looking for and how to evaluate whether those functions are useful or not we think that is worth to introduce some formalism[1].

2.2.1 Supervised learning, a formal model

We are in the context of supervised learning; we, our machine or our algorithm is the learner and the learner has access to:

- The domain set **X**. The set of all possible learnable/predictable objects. An instance $x \in X$ is usually represented by a vector of features.
- The label set Y. The set of all possible labels y that can be associated with an instance x.
- The training data $S = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$. That is a labeled set of instances. It is also called: *training set.*

The output of the learner is represented by:

- The predictor $h : X \longrightarrow Y$. A function that maps each element of X in Y. It can be also called: hypothesis or classifier. $h = A(S)$, the predictor is learned by a learning algorithm A when the training set S is given as input to A.

What is unknown to the learner is the data generation model, composed of:

- The distribution D over X. The distribution of probability according to which instances x are generated.
- The true labeling function $f : X \longrightarrow Y$.

[1]The formalism used in this work has been borrowed by [2]

Moreover, we need a measure of success in order to evaluate the performances of a predictor. This measure should corresponds to the probability that the label predicted by h for an instance x is equal to $f(x)$, in other words, the probability of $h(x) = f(x)$. Usually, instead of measuring the success of a predictor we measure the probability of its failure (1-probability of success). The function that assign an error (a value in $\mathbf{R}^+$) to a prediction is called: loss function $l(h, z)$, where z is an instance/label pair. The value $E_{z\sim D}[l(h, z)]$ is called *true error*[2] and represents the expected error that the predictor makes. When we compute the mean of loss function on the training set w.r.t. a predictor/hypothesis h we talk of empirical risk or training loss and we indicate it as $L_S(h)$

Summarizing and simplifying, the goal of the learner is to find a predictor h which is as similar as possible to the hidden labeling function f, knowing that each element of the training set S is a pair (x_i, y_i), where x_i has been drawn according to D and y_i is equal to $f(x_i)$.

2.2.2 Supervised learning framework

The general protocol that lead to the deployment of the final learned predictor is the following:

1. **Data splitting.** All the available labeled data are split in three different sets: training (S), validation (V) and test (T) set. Those three sets must be as independent as possible; to guarantee this independence usually the splitting rule is the outcome of a random process.

2. **Choice of the model.** Once the three data sets are settled, usually a class of hypothesis H is chosen. A class of hypothesis $\hat{H}$ could be represented by a model dependent on a set of parameters, in that case, to all the possible combinations of parameters corresponds all the possible hypothesis that one could pick within the class $\hat{H}$.

3. **Choice of the algorithm.** At this point we need an algorithm and a criterion to produce hypothesis candidates for our final predictor. There are many criteria that one can use to pick an hypothesis h^* within a class H, one of such criteria is called empirical risk minimization (ERM) and consists on choosing the h that minimize the value of the empirical risk $L_S(D)$. In other words, with ERM one try to minimizes the error of the predictor on the training set. Using this criterion, the higher is the complexity[3] of H the more it is likely to overfits the training data S. It means that, it is easy to find a predictor that works very well on S and very bad on novel data. Other criteria that allow to prevent overfitting exist, such as Structural Risk Minimization (SRM) and Regularized Loss Minimization (RLM). Without entering the details, the

[2] In this case we consider D as a joint distribution over X and Y, for example the conditional distribution $D((x, y)|x)$.

[3] We will not discuss about hypothesis class complexity in this work, to give an idea we suggest that the complexity of a class of function is related to its ability to divide a space following a complex pattern.

aim of these criteria is to find an hypothesis that represents a good trade-off between minimizing the training loss and find a hypothesis that works well in general.

4. **Training.** Run the algorithm or the algorithms on S to find $h_1, h_2, ..., h_n$ hypothesis candidates to be the final sought predictor.

5. **Validation.** Test the performances (computing $L_V(h)$) of the hypothesis candidate set $h_1, h_2, ..., h_n$ on the validation set in order to chose h^*, the best candidate among all. Eventually go back to point 4 and generate new hypothesis candidates. Practically, in this step what usually happen is the tuning of the parameters on which a chosen model depends on.

6. **Testing.** Test the performances of h^* on the testing set T. To test the performances we compute $L_T(h^*)$ which represents an estimation of the the true error for h^*. The larger is the test set the more reliable will be the estimation and the more we will be sure of the quality of the performances of our predictor.

2.3 Learning tasks

Depending on the nature of the label set Y, in the context of supervised learning, we can distinguish two different learning tasks: classification and regression. We are trying to solve a classification problem when the set of labels is discrete, a regression one when Y is continuous. In the previous section we have shown a simple binary classification problem, in fact the label set Y was yellow/purple, that is referable to a binary set $\{-1, 1\}$.

2.3.1 Computer vision tasks

In computer vision, machine learning find its place in numerous tasks, some of which are listed below:

- Image Classification: standard classification task where the instances of the Domain set are images. The algorithm has to return a classifier able to assign the right class to the input image.

- Object classification and localization: the task of not only correctly classify what the image is about but also to locate the element/object of the image that is the major responsible for the classification. It is usually set as a regression problem, in which the object centre coordinates (real numbers) are sought.

- Multiple objects detection and localization: when happen that the label to assign to an image is not unique. In other words, more than one object can be detected/localized in the image.

- Semantic segmentation: it is a classification task, where a label has to be assigned to every pixel of an image. It will later be referred also as pixel-wise classification.

2.3.2 Semantic segmentation

The method presented in this work is designed to solve a semantic segmentation problem where each pixel of the input image has to be classified as optic disc or not optic disc. Hence, for each pixel, we have to solve a binary classification problem. We chose to assign label 1 to pixels belonging to the OD and 0 to the others. The classifier's overall output will result in a binary image. We will later use as synonyms: binary map or segmentation map. In Figure 2.1 some visualization examples.

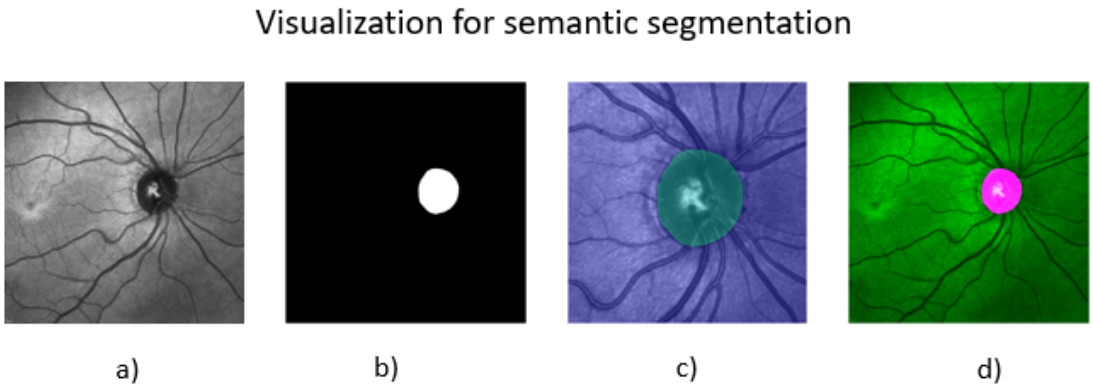

Figure 2.1: *a) input image. b) output binary map. c) A segmentation map and its corresponding image overlap. In cyan, OD pixels. In purple, not OD pixels. Visualization for details inspection. d) segmentation map and its corresponding image overlap. In pink, OD pixels. In green, not OD pixels. Visualization for a rough inspection.*

2.4 Deep learning

Deep learning is the branch of machine learning including all the techniques that, for solving a learning task, exploit a deep representation of the instance features. The concept of "deep" used in this context will become clear going through the next pages where Neural Networks (the leading actor in the field of deep learning) will be illustrated in details.
Before describing Neural Networks (NN) we will explain the Perceptron algorithm, that is the fundamental unit of which NN are composed of.

2.4.1 Perceptron

Assuming to be in the context in which we have a labeled data set of instances in $\mathbf{R}^d$ and we want to solve a binary classification problem. Moreover, we want to build a classifier represented by a hyperplane. In other words, we would like to get the best classifier h^*, for solving our classification problem, picking among the hypothesis class Hs_d of the halfspaces in $\mathbf{R}^d$. Where the Hs_d is defined as follow:

$$Hs_d = sign \circ L_d = \{x \to sign(h_{x,b}(x)) : h_{(v,b)} \in L_d\}$$

L_d here represents the class of the affine functions in $\mathbf{R}^d$. It is to say:

$$L_d = \{h_{v,b} : v \in R^d, b \in R\}$$

$$h_{v,b} : x \in \mathbf{R}^d \longrightarrow \mathbf{R}$$

$$h_{v,b}(x) = v \cdot x + b$$

In fact, L_d represents the class of functions in which each element $h_{v,b}$[4] depends on the two parameters v and b. Where v is a d-dimensional vector and b, usually referred as "bias", a real number.
We need an algorithm able to return h^* (best classifier) or at least an hypothesis as close as possible (in terms of true error) to h^*.
In this context, Perceptron is an algorithm for finding a classifier h^p that minimizes the empirical risk[5].
At this point we slightly change the notation merging the bias b and vector v in a unique vector w and adding the value 1 as first features of each instance x of our data set. We then define: $w = (b, v_1, v_2, ..., v_d)$ and we substitute the meaning of x with $x = (1, x_1, x_2, ..., x_d)$, resulting in $x \in \mathbf{R}^{d+1}$.
According to the new notation, we want to find an hypothesis $w \in Hs_d$.

Perceptron pseudo-code

Perceptron algorithm consists in a initialization (usually random or filled with zeros) of the classifier w, followed by an iterative update of the latter that lead step by step w to correctly classify the input data.

[4] the symbol $\cdot$ it is used to indicates the scalar product.
[5] we remind that using ERM rule for picking a classifier could lead to overfitting in case of data poor.

Data: training set: $(x_1, y_1), ..., (x_m, y_m)$, initialized $w^{(1)}$
Result: final predictor $w^{(f)}$;
for $t = 1, 2, ...$ **do**
  if $\exists i$ *s.t.* $y_i(w^{(t)} \cdot x_i) \leq 0$ **then**
    $w^{(t+1)} \leftarrow w^{(t)} + y_i x_i$
  else
    return $w^{(t)}$
  end
end

Algorithm 1: *Perceptron's pseudo-code*

Interpretation

As we have described in the previous sections, Perceptron is an algorithm for finding, within the class of the halfspaces Hs_d, a good classifier (according to ERM).
In fact, to give a geometrical idea we remind the reader that a vector, such as w, uniquely determines a perpendicular hyperplane passing through the origin. Every hyperplane divides the space into two sub-spaces (halfspaces) in which we would like to have instances with label 1 in the "upper" space and labels -1 in the "lower". In Figure 2.2 these concepts are clearly illustrated.
To check whether w is correctly classifying a labeled data x_i we simply compute $c = y_i(w \cdot x_i)$. In fact, the scalar product $w \cdot x_i$ return a positive value for a point "over" the plane and negative one otherwise. If this value and y_i are concordant, then c is ≥ 0 and the classification is correct.
Another cue, the updating rule in Perceptron, derives from the derivative of c. Every update is made to make the hyperplane tweaking in order to let a misclassified point becomes well-classified.

2.4.2 Neural Networks

A Neural Network (NN), also called Multi-layer Perceptron is a powerful model for learning and the core of deep learning. Informally, NN is a way to combine many Perceptron units (hyperplanes) to form complex functions for solving learning tasks. Formally[6], NN can be defined as a directed acyclic graph G = (V,E) organized in layers. Where V is the set of nodes and E of edges. To each edge e corresponds a weight $w(e)$ specified by $w : E \longrightarrow \mathbf{R}$. The first layer V_0 is the input layer, the last V_T is the output layer, layers in the middle are called hidden layers. An edge $e \in E$ can connect only nodes from a previous layer to nodes belonging to a subsequent layer. Network shown in Figure 2.3 is a fully-connected neural network because exists an edge linking each nodes from a previous layer to each nodes in the subsequent layer, other configurations are possible.
From the point of view of a generic node $v_{t+1,j}$:

- The input, when x is fed to the net, is $a_{t+1,j}(x)$. It consists in a linear combination, dependent on w, of the nodes output from the previous layers. Hence: $a_{t+1,j}(x) \in \mathbf{R}$.

[6]The formalism used in this work has been borrowed by [2]

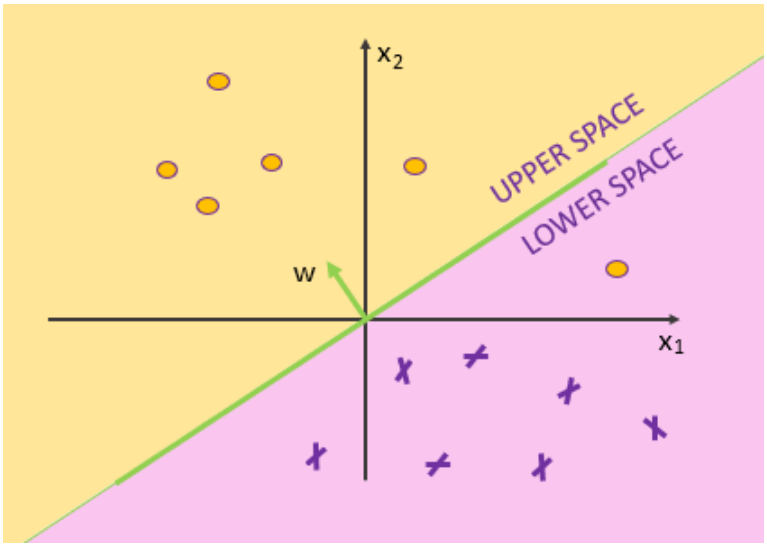

Figure 2.2: *Example of vector w, defining a hyperplane in* $\mathbf{R}^2$ *(a line).*

- The output is $v_{t+1,j}(x)$. The output of each node is a function of its input $\sigma : \mathbf{R} \longrightarrow \mathbf{R}$. It is called activation function.

In this work, as activation function we will use only Rectified linear unit [4] $\sigma_{ReLU}(z)$:

$$\sigma_{ReLU}(z) = \begin{cases} 0 & \text{for } z < 0 \\ z & \text{for } z \geq 0 \end{cases}$$

Many other functions can be used, the most common in literature are: binary step, identity function, tanh and leaky ReLU.

The architecture A of a NN is then defined by $A = (V, E, \sigma)$. The architecture defines the hypothesis class H_A of the all possible predictors that can be "built" with that architecture. Each predictor $h_{A,w}$ is defined in the moment we fix w (mapping between edges and weights):

$$h_{A,w} : \mathbf{R}^{|V_0|-1} \longrightarrow \mathbf{R}^{|V_T|}$$

2.4.3 Learning process

The general process that allow a NN to fit the input data, using a stochastic gradient descent (SGD) strategy is the following. An instance $x = (x_1, ..., x_d)$ is randomly drawn from the training set, the network compute the output y (prediction) by propagating linear combinations/activation functions through the layers of the net. The overall process is named forward propagation.
The output y is compared with the true label y_t and a loss function return the error. The error is then back-propagated (from the output layers to the input) and the weights in the net are tweaked in a way to reduce the error in case of re-computing the output y' (back-propagation algorithm). Another instance is randomly drawn and the process is repeated.

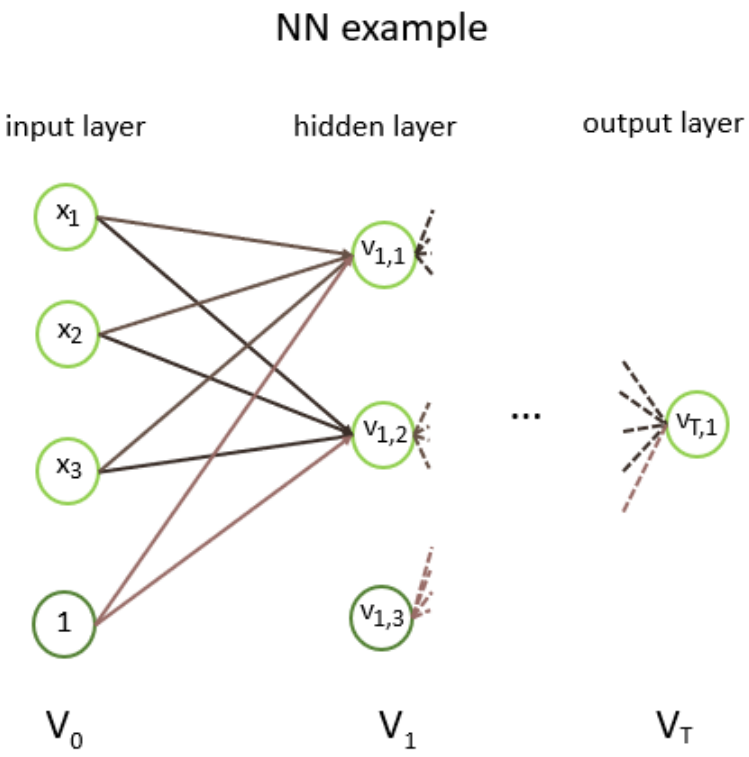

Figure 2.3: *Example of NN architecture, with an input layer of 4 features of which one is a constant. V_1 is an example of hidden layer with two nodes plus a node representing the bias. The number of nodes in the output layer depends on the dimension of the labels we want to learn.*

2.4.4 Matrix notation

The overall input to the layer V_t is $a^{(t)}$, given by:

$$a^{(t)} = (w^{(t)})^T v^{(t-1)}$$

where $w^{(t)}$ represents the set of weights of the edges linking layers V_{t-1} and V_t. Assuming $d^{(t)}$ to be the number of nodes at layer t, the output of such layer would be the array of size $d^{(t)} + 1$:

$$v^{(t)} = \begin{bmatrix} 1 \\ \sigma(a(t)) \end{bmatrix}$$

We can then write the matrix w as:

$$w^{(t)} = \begin{bmatrix} w_{01}^{(t)} & w_{02}^{(t)} & w_{03}^{(t)} & \dots & w_{0d^{(t)}}^{(t)} \\ w_{11}^{(t)} & w_{12}^{(t)} & w_{13}^{(t)} & \dots & w_{1d^{(t)}}^{(t)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{d^{(t-1)}1}^{(t)} & w_{d^{(t-1)}2}^{(t)} & w_{d^{(t-1)}3}^{(t)} & \dots & w_{d^{(t-1)}d^{(t)}}^{(t)} \end{bmatrix}$$

2.4.5 Forward propagation

As we mentioned before, we call "forward propagation" the overall computation process that leads the network to produce the output y when the instance x is given as input. As foolows, the pseudo-code of such algorithm:

Data: $x = (x_1, x_2, ..., x_d)^T$
Result: label predicted y
$v^0 \leftarrow (1, x^T)^T$;
for $t \leftarrow 1$ *to* T **do**
  $a^{(t)} \leftarrow (w^{(t)})^T v^{(t-1)}$;
  $v^t \leftarrow (1, \sigma(a^{(t)})^T$;
  $y \leftarrow v^{(T)}$;
end

Algorithm 2: *Forward propagation's pseudo-code*

2.4.6 Back propagation

Back-propagation algorithm is the method used for updating the weights of the net in order to fit the data. Before showing the algorithm in a pseudo-code form, some definitions and preliminaries are needed.
We would like to find the weights that minimize the empirical error which depends on the loss function L. To achieve this goal we are interested in knowing how the error depends on the weights for knowing how to tune such weights. Using a gradient descent as minimizer means that we need an update rule such that:

$$w^{(t)} \leftarrow w^{(t)} \eta \Delta L_S(w^{(t)})$$

Where $\Delta L_S(w^{(t)})$is the gradient of L and η is a scalar, called learning parameter. To compute the gradient we need to compute for each layer t, the derivative of L_S w.r.t. $w^{(t)}$. It is convenient at this point to define the sensitivity vector for layer t.

$$\delta^{(t)} = dL/da^{(t)} = \begin{bmatrix} dL/da_{t,1} \\ \vdots \\ dL/da_{t,d^{(t)}} \end{bmatrix}$$

Sensitivity vector represents how the input to layer t influences the changing of the error. We now jump directly to the conclusion, remanding the full algebraic illustrations to [2] and showing the equation:

$$\delta_j^{(t)} = \sigma'(a_{t,j}) \sum_{k=1}^{d^{(t+1)}} w_{j,k}^{(t+1)} \delta_k^{(t+1)}$$

It is interesting to notice in this equation that the sensitivity for t must be computed starting from the sensitivity of layer $t+1$. Hence, first we need $\delta^{(T)}$.
We separately show the pseudo-code of the backward propagation routine and then the pseudo-code for the entire back-propagation algorithm.

Data: instance $x = (x_i, y_i)$
Result: sensitivity $\delta(t)$ for each t;
forward propagation to compute $a^{(t)}, v^{(t)}$ for each t;
$\delta^{(T)} \leftarrow dL/da^{(T)}$;
for $t = T-1$ *to* 1 **do**
  $\delta_j^{(t)} = \sigma'(a_{t,j}) \sum_{k=1}^{d^{(t+1)}} w_{j,k}^{(t+1)} \delta_k^{(t+1)}$ for all $j = 1, 2, ..., d^{(t)}$
end

Algorithm 3: *Backward propagation routine pseudo-code*

Data: training set S, NN with initialized weights $w_{i,j}^{(t)}$ for all i, j, t
Result: NN with updated weights
for $t = 0, 1, 2, ...$ *until convergence* **do**
  choose a random data point (x_k, y_k);
  forward routine;
  backward routine;
  $w_{ij}^{(t)} \leftarrow w_{ij}^{(t)} \eta \delta_j(t) v_i^{t-1}$ for all i,j;
end

Algorithm 4: *Back-propagation routine pseudo-code*

Using this algorithm we update the weights in order to try to minimize the empirical error. We are using a SGD strategy, in fact, at each iteration a single data point is randomly picked from the training data giving in this way randomicity to the update directions for w. A common variant is to, at each iteration, instead of picking a single data point, to pick a mini-batch of two or more data points, then compute the mean error among all the mini-batch and consequently update the weights.
During the training phase, the back-propagation algorithm runs for many epochs. An epoch ends when all the data points/mini-batches have been fed to the net. At the beginning of each epoch, the data points are (usually) shuffled and, in case, divided in mini-batches. The training ends when the training error or the improvement in accuracy per iteration are low.

2.4.7 Meaning of *deep*

We use the adjective "deep" when we have a NN with at least one hidden layer. In fact, let's take as an example the net NN_s composed of three layers: V_0, V_1, V_T, where V_0 is the input layer and V_1, V_T have respectively 2 and 1 nodes. In addition, all the activation functions in NN_s are the identity function. In such a network, an input instance x is fed to the network in layer V_0, then given to V_1. In V_1 two hyperplanes, h_1 and h_2 (two nodes in V_1), project x into x', a two dimensional representation of x (the activation function is the identity, then no ulterior transformation is applied to x'). The two features of x' are related to the distances from the two hyperplanes. We call x' a "deep representation of x", that can be seen in this case as the representation of x in the space defined by h_1 and h_2.
The higher is the number of layer of a NN the deeper is the network.

2.4.8 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are not-fully-connected networks. In other words, CNNs are NNs in which it is not required that for each node in a previous layer, exists an edge linking to each node in the subsequent layer. This property let CNN be useful in the field of image analysis and elaboration. In fact, in images, the information is usually highly correlated only within small regions compare to the entire image. In CNN, nodes are usually called "filters". Filters are frequently squared shaped and can be seen as masks of weights. We remind that when a one-channel image is filtered with linear filters, the output is another one-channel image, in which each pixel value is the results of the scalar product between the filter and the input image's patch that is centred on that pixel. If the input image has more than one channel, the input of the filter would be the volume defined by the pixels belonging to the same patch among all the channels; the output would results again in a single channel image. If we want the output image to have the same size of the input we must apply an adequate padding to the input image and compute the filtering for every possible patch of the input image. When a CNN is fed with an image x, the output y is given by the forward propagation algorithm, the intermediate images resulting from each step (after each layer) are called features maps. It is common to use CNN architectures in which, as the network goes deeper the size of the features maps gradually decreases. To achieve such reduction in size, two method are commonly used: via pooling layers (lately discuss) or, when filtering in a certain layer, do not apply the filter to every single patch, but skipping some by setting a stride factor. Among the advantages of decreasing the size of the features we find: to gradually relate the information coming from different regions of the input image and to decrease the computational cost when training the network.

Like for NN also for CNN is possible to use specialized layers to perform different kinds of operations. In the networks developed for this work, the following specialized layers have been used: max-pooling, batch-normalization, softmax, concatenation, transposed-convolutional and ReLU. Moreover, one can use different types of output layers, that is to say, layers in which the final output function is computed and, while training, the error assigned.

2.5 Specialized layers

In this Section we briefly describe the layers used in the architectures presented in this thesis. Each of the following layers is used to insert particular functions at different levels of the networks. The layers have to be adequate built for being able to work both during the forward and backward propagation phases.

Max-pooling layer

Max-pooling layer is a non-linear filter that takes in input a patch of an image/feature map and return as output the value of the pixel within the patch with maximum value. For example, networks implemented in this work utilize max-pooling layers with filter size of 2x2, no padding and stride equal to 2. The output image of such filters has half of the dimension of the input. In fact, the image is divided in a grid of 2x2 pixels squares, for each square only the maximum value is kept as output.

Batch-normalization layer

Batch-normalization layers are used to make feature maps having values varying in a restrained range. The main benefits, according to [5] of using batch-normalization layers are: to speed-up training, to make the learning more stable and to produce some sort of regularization effect. Practically, the layer computes mean value and variance of a feature among the mini-batch and normalizes according to such values (by subtracting the mean and dividing for the variance). Afterwards, the layer multiplies each feature in the mini-batch for a scale factor γ and add an offset β. Where γ and β are learnable parameters.

ReLU and Softmax layer

ReLU layer is a layer that computes the ReLU activation function for each pixel of a feature map.

The softmax layer, differently from the previous layers, takes in input one or more features maps ($\{f^{(1)}, f^{(2)}, ..., f^{(k)}\}$) of the same size. The layer computes the softmax function with respect to each sequence of corresponding pixels among the input maps; pixels belonging to different maps, at the same coordinates: $z_{i,j} = (z_{i,j}^{(1)}, \ldots, z_{i,j}^{(k)})$). Softmax function $\sigma : \mathbf{R}^k \to \mathbf{R}^k$ is defined by the formula:

$$\sigma(z_{i,j}) = \begin{bmatrix} \frac{e^{z_{i,j}^{(1)}}}{\sum_{f=1}^{k} e^{z_{i,j}^{(f)}}} \\ \vdots \\ \frac{e^{z_{i,j}^{(k)}}}{\sum_{f=1}^{k} e^{z_{i,j}^{(f)}}} \end{bmatrix}$$

The softmax layer is usually used before the output layer to give to the latter batches of features maps normalized between 0 and 1, according to the softmax function.

Transposed convolutional layer

Also called "deconvolution layer", this layer has been proposed first in [6]. Without entering the details, this layer allows the network to increase the resolution of the features maps, learning a sort of optimum up-sampling of the input image. The input-output relation is given by the formula: $O = (I - 1)s + k$ Where O is the output size, I the input size, s he stride and k the kernel size.

Concatenation layer

Concatenation layer allows to stack feature maps coming from different path in a CNN. The stacking is performed channel-wise, the output of the concatenation layer can be given as input to following convolutional or specialized layers.

2.6 Output layers

In this section we briefly describe the output layers used in the architectures presented in this thesis. Output layers are layers in which the error to be back-propagated is computed (loss function).

2.6.1 Pixel-wise classification layer with cross-entropy loss function

Pixel-wise classification layer, with cross-entropy loss function, is a type of output layer.In a binary classification framework, given a training set S of m images, given an image X in S the cross-entropy loss function is defined as follows:

$$L(h_{A,w}) = -(\beta \sum_{j \in Y_1} \log P(y_j = 1|\ X;\ h_{A,w}) + (1 - \beta) \sum_{j \in Y_0} \log P(y_j = 0|\ X;\ h_{A,w})).$$

Where:

- the function assigns an error to the hypothesis $h_{A,w}$ defined by the network architecture and the weights w.
- The value y_j is the label that can be assigned to pixel j of X.
- β is a parameter for taking into account eventually class unbalances. It is settled to be equal to the ratio of "zero labeled pixels" and "one labeled pixels" in S.
- Y_1 and Y_0 represent the set of pixels having respectively label 1 and 0 in the ground truth.

2.6.2 Pixel-wise classification layer with generalized Dice loss function

The Dice loss is based on the Sørensen-Dice similarity coefficient for measuring overlap between two segmented images (see Section 1.7.1). The generalized Dice loss ([10], [9]), L, for between one image X and the corresponding ground truth Y is given by:

$$L = 1\frac{2\sum_{k=1}^{K} w_k \sum_{m=1}^{M} X_{k,m}Y_{k,m}}{\sum_{k=1}^{K} w_k \sum_{m=1}^{M} X_{k,m}^2 + Y_{k,m}^2}$$

where K is the number of classes, M is the number of elements along the first two dimensions of X, and w_k is a class specific weighting factor that controls the contribution each class makes to the loss. w_k is typically the inverse area of the expected region:

$$w_k = \frac{1}{(\sum_{m=1}^{M} T_{k,m})^2}$$

This weighting helps counter the influence of larger regions on the Dice score making it easier for the network to learn how to segment smaller regions.

2.7 Transfer Learning

In machine learning, transfer learning is when the knowledge gained while solving a particular problem it is used to solve another problem related to the first. To give an example we illustrate, how this concept has been applied in our work.
We wanted to find a good classifier, using a CNN, for solving the semantic segmentation problem explained in Section 2.3.2. Our data set was composed of an overall of 120 SLO images of which 50 annotated by ophthalmologists and 70 with annotations produced by ALG_1, an automatic algorithm (less accurate than doctors). We decided to split the training into two phases, a first phase where our CNN is trained on the 70 images batch and the second phase of training using the "doctor's" batch. We applied the transfer learning concept in the sense of learning the optimum weights from the first "roughly annotated" batch and use the resulting network as initialization for the second learning phase on the "fine" batch.

2.8 Performances evaluation criteria

There are many criteria that one can choose for evaluating the performances of an algorithm for image segmentation. In this work, we used the following coefficients/metrics: Dice-coefficient, Jaccard, contour distance and accuracy.

2.8.1 Sørensen-Dice and Jaccard coefficients

The Sørensen-Dice coefficient (frequently addressed simply as Dice coefficient or F1 score) is defined as follow:

$$Dice(X,Y) = \frac{2|X \cap Y|}{|X| \cup |Y|} = \frac{2TP}{2TP + FP + FN}$$

Where for the first expression, X, Y are sets of elements and |.| indicates the cardinality of a set. The second definition, instead, explicits the coefficient as function of true positive (TP), false positive (FP) and false negative (FN) values. Jaccard coefficient is very similar to the Dice coefficient, in fact is defined by the formula:

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{TP}{TP + FP + FN}$$

We use these coefficients for comparing two binary maps, for example the map returned by our algorithm and the map deriving from the doctor's annotation. In such context, we can compute the two indexes by applying to the binary maps, pixel-wise, the logical definitions.

2.8.2 Contour distance metrics

The information returned by Dice and Jaccard coefficients is related to area of the two binary maps overlap region, the shape of the contour of the segmented object is not taken into account. For evaluate the contour matching between two contours we used the Hausdorff distance, and the mean contour distance, these indexes will be lately indicated respectively as δ_{max} and δ_{mean}.
We can define these indexes as follows; given two sets of contour points C_1, C_2, for each point $p_i \in C_1$ we find the point $q_j \in C_2$ that has, among all the points in set C_2, the lowest Euclidean distance form p_i, we name this distance d_i.

$$d_i = \min_j \{ \ ||p_i - q_j|| \quad : \ p_i \in C_1, q_j \in C_2 \}$$

Hence, we define the contour distance $\delta_{max}(C_1, C_2)$ as the highest distance among all the distances d_i $(i = 1, ..., |C_1|)$ and $\delta_{mean}(C_1, C_2)$ as the mean.

2.9 Registration

In the context of computer vision, image registration is the task of bringing data from different spaces, in particular, images represented in different systems of coordinates, to data represented in the same space. Image registration very useful in the field of medical imaging, for example, for comparing images of the same organ or tissue acquired in different periods or by diverse instrumentations. In this work, image registration have been used for comparing retinal images of the same eye, acquired with different cameras (SLO and fundus).

Usually, for solving the registration problem between two images, one is taken as reference (fixed image) and its coordinates are taken as the coordinates of the final common space. The other image is called the moving image and is the image that have to be modified and represented in the new coordinates (the fixed image coordinates). In order to do so, we have to find a transformation function able to map each point of the moving image to the corresponding point of the fixed image. Many approaches for finding such function have been proposed, in this work, we used the method proposed in [7] and [8], respectively named: piece-wise linear mapping function and local weighted mean.

2.9.1 Piece-wise linear mapping function

The idea behind this method is to find the global transformation function by merging local transformation function defined on small regions of the two images (piece-wise). The procedure can be summarized as follows:

1. Sample n control points in two images, points (X_i, Y_i) in the fixed image corresponding to points (x_i, y_i) in the moving image.

2. Find, among the moving image control points, the optimal triangulation, that is the triangulation formed by the triangles such that any point in a triangle is closer to the three vertices that make the triangle than to vertices of any other triangles. Such triangulation is used to divide the images in regions where is possible to define the sub-transformation functions.

3. To each moving control point (x_i, y_i) the x-coordinate of the corresponding fixed control point (X_i, Y_i) is associated, the results is a 3-dimensional point (x_i, y_i, X). At this point, the sub-transformation functions are found by interpolating with a plane the three vertices of each triangle (vertices are 3-D points).

4. Because the optimal triangulation led to the definition of a convex region in the moving image, some part of the image will remain outside the triangulation, to menage those excluded regions the method assigns to points belonging to such regions the transformation functions defined on the closest triangle.

5. Now we have a function $X = f(x, y)$ defined for all the pixels in the moving image that has been obtained by merging the linear interpolations of several triangular regions. In order to get $Y = f(x, y)$ we can redo the procedure starting from point 2, by forming the triplets (x_i, y_i, Y).

2.9.2 Local weighted mean

As in the previous method, n pairs of control points are taken from the two images. For each of the control points, the set of n-1 nearest neighbours is defined and a polynomial[7] over the point and its neighbourhood set is fitted by a weighted fitting algorithm. In this way, we obtain n polynomials and each polynomial will fit the most the point used as a

[7]The method works independently on the polynomial degree.

reference.
The method defines the transformation function $X = f(x, y)$, where (x, y) is a general point in the moving image, as the function resulting by a weighted sum of the n polynomials. The weighting function $W(x, y)$ used for the summation gives a higher weight to the polynomials fitted using as reference points closer to (x, y). It can be proven that the registration function resulting from this process is smooth everywhere.

2.10 Summary

The method presented in this thesis is based on deep learning, a branch of machine learning (Section 2.2). It is a supervised learning approach for solving a pixel classification problem. Examples of input data and corresponding outputs are shown to a system which tries to learn the hidden input-output linking function. The "system" in our case is represented by a CNN, in Section 2.4.2 we explained how the learning process for NN works. In Sections 2.8, we illustrated the metrics that we used for evaluating the performances of our system. Finally, we reported the two methods that we exploited for multi-modal image registration of same retina.

Chapter 3

Literature methods for OD detection and segmentation

3.1 About this chapter

In this chapter, we are going through a brief overview of a small sample of methods proposed for the automatic segmentation of the optic disc in fundus camera images. We will see an example of an algorithm based on the optic disc appearance (non-learning technique) and others based on machine learning/deep learning. Moreover, we will describe ALG_1, an algorithm for OD segmentation in SLO images developed in the context of the same project of this thesis.

3.2 Optic disc segmentation in fundus images

Most of the work for OD automatic detection and segmentation have been focusing on fundus images. In recent years the research moved from non-learning techniques to the development of algorithms based on machine learning and in particular deep learning. Belonging to the first category (non-learning) we will show the method proposed in [11], this method is currently implemented in the tool released by the VAMPIRE project (link at [29]). As a representative of the second category (deep learning), we will show the method proposed in [14].

3.2.1 A non-learning approach, Giachetti et al.

The idea of the method presented in [11] is based on the following observations regarding the general OD appearance in fundus images:

- *"Its shape is approximately elliptic. It is not always the brightest part of the retina, but, even in many anomalous cases, it is the bright part with the highest radial (circular) symmetry."*

- *"There is a high vessel density inside its contour.The structure of the vasculature may not be easy to model, but vessels can always be seen near/inside the OD and a rough segmentation of them can be used to estimate a local density."*

According to the original paper, these OD features appear to be stable among different data sets.

The pipeline of this method is subdivided between detection and subsequent segmentation of the OD. For the detection (finding of the centre coordinates) the algorithm uses a combination of two weak detectors: one seeks for bright regions with high radial symmetry, one seeks for high-density vessels regions. The final output of these two detectors are two probability maps that combined by a pixel-wise multiplication (with some adjustments) provide a probability map for the OD location (the region with the highest probability/score is chosen as location).

Once the centre has been located, the method proceeds to seek for the contour by a coarse-to-fine strategy in which both the resolution [1] of the input image and the complexity of the model used for fitting the contour are gradually increased.
At low resolution the OD contour is fitted with a circle (circular sampling of points) by finding the circle that maximizes its inner/outer contrast through an optimization procedure. Hence, for two times the resolution increases and the contour is fitted with an elliptical shape. At this point, the original resolution is reached and a snake technique is used for finding a free form contour. The final result is an elliptical fitting of the previous points determined by the snake. In Figure 3.1 example of results obtained at the different stages by the algorithm.

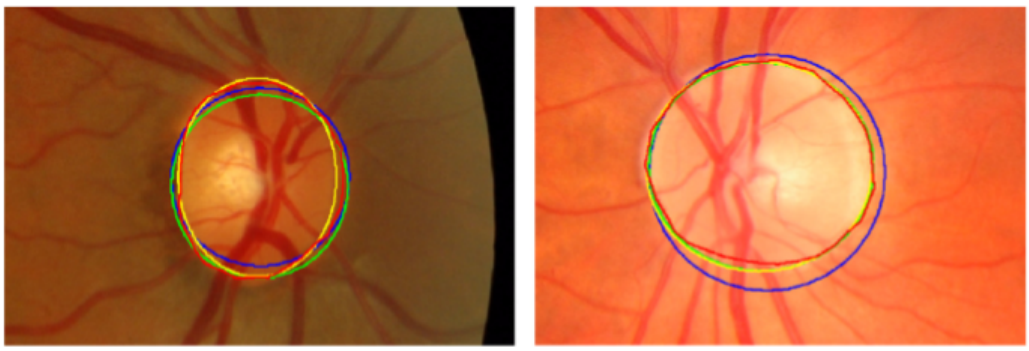

Figure 3.1: *Example of contours detected at multiple scales. Blue line: initial circular shape. Green line: intermediate elliptic contour. Yellow: final ellipse fitting result. Red: free-form contour computed with the snake-based refinement algorithm. Image taken from [11].*

[1] The segmentation pipeline starts from downsized images and end with images at the original resolution

3.3 Deep learning approaches on fundus camera images

In recent years, deep learning has outperformed other techniques in many computer vision tasks. Deep learning has been successfully applied in the field of retinal image understanding and many algorithms for vessels and optic disc segmentation/detection have been proposed.

In general, solving a segmentation problem via deep learning, means, to find the right CNN architecture, the one that achieves the best results among a testing set. Many of the proposed architectures exploit or take inspiration from the architectures VGG and U-net presented respectively in [12], [13]. The system developed in [14] takes advantage of VGG architecture for both vessels and OD segmentation while in [16] a modified version of U-net architecture is used for optic cup and disc segmentation. In the following sections, we are going to illustrate such architectures and we will enter the details of the method proposed in [14].

3.3.1 VGG architecture

VGG architecture has been proposed in [12] after winning the ImageNet competition in 2014 ([24]). The network has been designed for solving a multi-class (1000 classes) classification problem taking as input 224x224 RGB images[2]. The architecture consists of a stack of convolutional layers with kernels size of 3x3, stride 1 and zero-padding 1. Five max-pooling layers (2×2 pixels window, with stride 2 each) reduces the size of the features maps as the net goes deeper (resolutions: 224, 112, 56, 28, 14). At the end of the stack of convolutional layers, 3 fully-connected layers: the first two have 4096 channels each, the third contains 1000 channels (one for each of the 1000 classes). The nal layer is the soft-max layer. All hidden layers are followed by ReLU layers.

3.3.2 U-net architecture

The network architecture is illustrated in Figure 3.2. It consists of an encoding path (left side) and a decoding path (right side). The encoder consists of the repetition of the pattern: convolutional layers (filter size 3x3, stride 1, no-padding), ReLU layer, max pooling (2×2 pixels window, with stride 2 each). After each pooling number of feature maps are doubled. Every step in the decoding path consists of an up-sampling of the feature maps followed by a 2x2 convolution (transposed-convolutional layer) that halves the number of feature channels, a concatenation with the correspondingly cropped feature maps from the encoding path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer, a 1x1 convolution is used to map each 64 component feature vector to the desired number of classes. In total the network has 23 convolutional layers.

[2]Images centred by subtracting the mean RGB value for each pixel

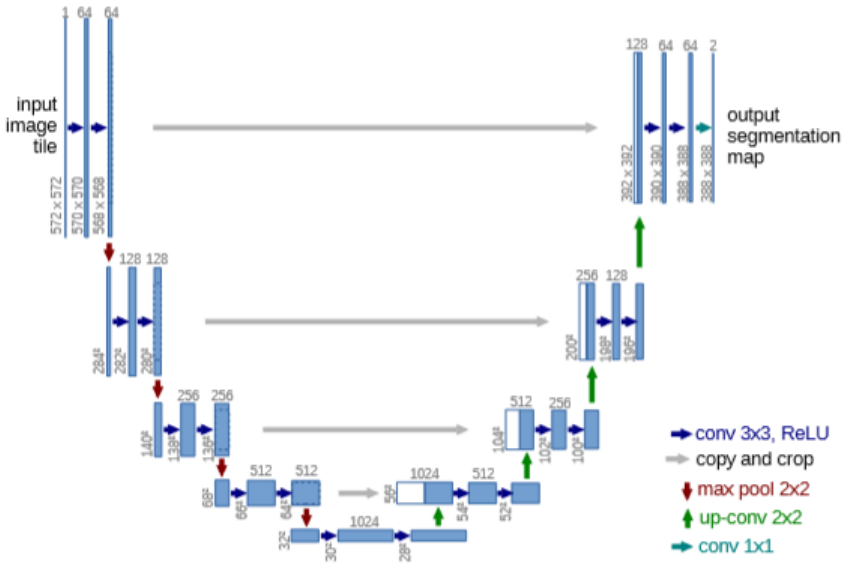

Figure 3.2: *U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the dierent operations. Figure and description from the original paper [13].*

3.3.3 Deep learning approach, Maninis et al.

The approach presented in [14] relies on the concept of transfer learning. In fact, a pre-trained model of the network VGG described in Section 3.3.1 is used as a starting point for both vessels and optic disc segmentation in retinal fundus images. VGG has been designed to be trained on millions of natural images and for solving a multi-classification problem, that is why it can't be directly used and applied for a segmentation task. The system (DRIU) has been developed to exploit the "knowledge" learned by VGG. DRIU consists of a CNN that is the assembling between a base network and some specialized layers. The base network is a pre-trained net VGG where the last fully-connected layers have been removed (used for classification). The specialized layers, in this context, are newly initialized convolutional layers that take as input a stack of base network's features maps taken from different levels of depth. The output layer is placed at the end of such specialized layers and the loss utilized is pixel-wise cross-entropy loss function (Section 2.5.2). The scheme in Figure 3.3 is clearly illustrating the final result. It is worth to add to the description the following:

- the paths leading to vessels segmentation and OD segmentation are separately trained.
- To merge the features maps coming from different parts of the base network, those have to be resized to the same resolution.

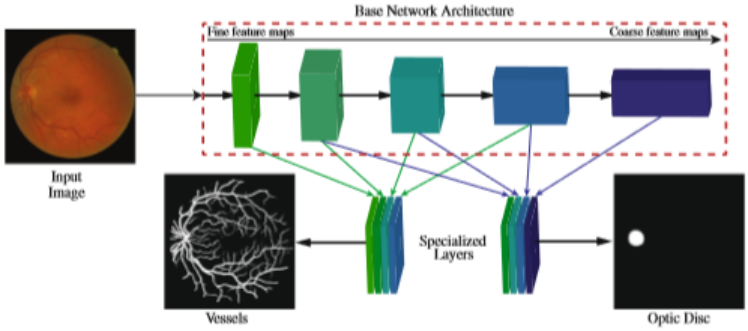

Figure 3.3: *Given a base CNN, we extract side feature maps and design specialized layers to perform blood vessel segmentation (left) and optic disc segmentation (right). Figure and description from the original paper [14].*

3.4 Optic disc segmentation in SLO images, ALG_1

AlG_1 is an algorithm for OD segmentation in SLO images developed in the context of the same project of this thesis, it will be presented in [15]. It consists in a non-learning approach partially inspired by the algorithm described in Section 3.2.1, like the latter, it is based on the optic disc appearance. The main characteristic of the OD in SLO images are:

1. the OD is usually dark but it might have a lighter spot inside.
2. The contrast between vessels and OD is usually low.
3. The contrast between OD and background is usually high.
4. The shape is approximately elliptical.

The pipeline can be divided in the following steps: detection, circle fitting and sampling, features extraction, contour selection, refinement.

Detection

For localizing the OD, the method takes advantage of the previous observations and via morphological operations obtains a reliable detection of the OD. Summarizing, when an SLO retinal image is given to the algorithm, the procedure for the localization is the following:

1. run of an iterative routine of adaptive thresholding that stops when a certain ratio between foreground/background[3] pixels is reached. The result is a binary map, that represents an initial segmentation of both vessels and OD.
2. An opening operation performed over the segmentation map helps to separate vessels from OD. The result is a binary map with several connected components.

[3] Are considered foreground the dark pixels.

3. Only the largest connected component is kept, such component it is very likely to include the OD. In fact, not only the OD is usually large and dark but also the vessels within the OD, and close to, are the darkest and the thickest among the ones appearing in a general SLO image.

4. Closing of the resulting binary image to fill holes generated during phase 2 and computation of the center.

Circle fitting and sampling

The computed centre is used as the starting point for a circle fitting of the OD contour. The circle fitting is obtained using the same optimization procedure of [11]. Hence the result is a circle, sampled in N points, that maximize its outer/ inner contrast. Those N circle points are used as a landmark for building a radial sampling grid; for each circle point, a set of n points are sampled along the radial direction. The overall sampling grid can be represented as the matrix S of nxN intensity values. The bottom row of S represents points sampled near the centre of the OD, points in the central row correspond to the N circle points, points in the first row correspond to points sampled in the periphery.

features extraction

Hand-crafted features are extracted over each sampled point by filtering S with a set of selected filters. The results after this phase, are m features maps (each obtained by applying a different operation on S) that can be organized as a volume of size $nxNxm$ or as a set X of nxN m-dimensional points.

Contour selection

Over the set X the method seeks for the points that better match the typical appearance of an OD contour by choosing the sequence of points C that minimizes an *ad-hoc* built cost function.

Refinement

The sequence C is smoothed to obtain a more regular contour candidate, hence C is used as a landmark for building a new and finer sampling grid where extract new feature and compute again the contour selection phase. In Figure 3.4 a concise representation of the pipeline of this method.

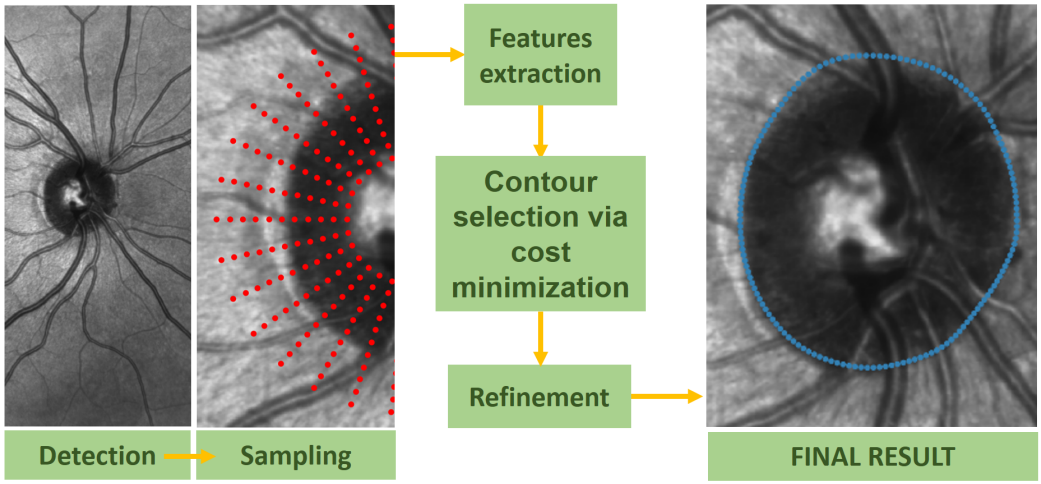

Figure 3.4: *Pipeline of ALG_1. Figure from the conference poster presented in [15].*

3.5 Performances assessment

Comparing the results obtained by methods such as the ones described in the previous chapters is not always an easy task due to differences in the evaluation criteria used in different works or due to the fact that the data sets used for the evaluation are not the same. In the recent work of Qin et al. [19] a comparison in terms of mean Dice and Jaccard coefficients is conducted between their proposed method (deep learning) and some of the aforementioned systems. The data sets used for comparing the methods are two glaucoma screening data sets. The first one is the REFUGE data set, which consists of 400 images with 40 glaucoma cases. The second one is the data set from the Second Affiliated Hospital of Zhejiang University School of Medicine, which contains 697 fundus images with manual ground truth of optic disc segmentation, including 230 glaucoma cases and 467 normal cases. The results are summarized in table 3.1.

The results achieved by the method described in Section 3.2.1 and 3.4 will be evaluated in Chapter 6 and directly compared with the method proposed in this work.

Methods	Dice	Jaccard	Ref.
Maninis et al.	0.96	0.89	[14]
Qin et al.	0.95	0.92	[19]
Sevastopolsky	0.94	0.91	[16]
Zilly et al.	0.94	0.89	[18]

Table 3.1: *Results as reported in [19].*

3.6 Summary

In recent years, methods based on deep learning have outperformed the other approaches in many fields of computer vision. This is true also for retinal imaging. Deep learning has been used for the automatic segmentation of the optic disc in fundus images achieving the ophthalmologists' accuracy in solving the same task. Most of these methods exploits CNNs inspired by the architectures U-net and VGG.
To the best of our knowledge, until today, no other method has been proposed for OD segmentation in SLO images except for ALG_1 that is based on a non-learning approach.

Chapter 4

Medical annotations

4.1 About this chapter

In this chapter, we are going to discuss the protocol that has been established for acquiring the medical annotations of the optic disc in SLO and fundus images. In fact, such annotations have been necessary for both the development of the core method described in this thesis and the evaluation of the latter. Furthermore, in the chapter, a statistical analysis of the collected annotations is conducted for defining parameters, such as the annotators intra and inter-agreement, that are fundamental for assessing the quality of the results proposed by an automatic algorithm.

4.2 Ophtalmologists

Four ophthalmologists have partecipated to the project, giving their availability for producing the annotations. The doctors have different grades of experience: one consultant eye surgeon, two ophthalmology registrars and ophthalmic Specialist (Trainee, year 2).
In the following sections/chapters we will use the abbreviations: A_1, A_2, A_3, A_4 for referring both to the four annotators and to the related set of annotations.

4.3 Images

The overall set of images to be annotated consists of 50 pairs of SLO (1536x1536) and fundus camera images (2048x3072). The images were obtained in the context of the PREVENT Dementia study [25] from Edinburgh Imaging and the Edinburgh Clinical Research Facility. The SLO images were acquired with a Heildelberg SPECTRALIS SLO camera, the fundus images with a Canon non-mydriatic camera.

4.4 Annotation protocol

In this section, we report the SOP (standard operation procedure) that has been given to the ophthalmologists for standardizing the annotation procedure. The SOP was divided into 6 Sections: purpose and context (1), methods (2), loading of the images (3), annotation protocol (4), images for repeatability (5), returning the annotated images (6). As follows, we directly report Sections: 2, 4, and 5.

4.4.1 Methods

For simplicity, we recommended to use Microsoft Paint (Our guidelines refer to Microsoft Paint for Windows 10, version 1803). We suggested to annotate two contours per OD: one showing the most likely contour in your opinion, and a second, where needed, showing a plausible alternative for possible, uncertain parts of the contour. In case of visible peripapillary atrophy (PPA) in the image, we asked to the doctors to annotate it with a third contour. A contour is annotated by placing (clicking) points along it. For guidance, e.g. frequency of points, we provided a few examples of annotated contours. For placing the points, we allowed the use of the zoom.

4.4.2 Annotation protocol

We provided the doctors with the instructions and recommendations to follow for the annotating procedure which are reported below:

a. Before starting look at the examples.

b. Try to annotate the whole set of images consistently, e.g. in similar conditions, and dedicating comparable amounts of time and attention to each image.

c. It is required to annotate the SLO and fundus images independently.

   1. Open the image in Paint.
   2. Place red dots along your first choice of contour. To do this, select 'brush' and 'size' as shown in Figure 4.1. For number / frequency, please follow the examples.
   3. If you are uncertain about some parts of the contour, place yellow dots along your second choice of contour.
   4. If PPA is visible, place green dots along the contour.
   5. Save the annotated image as <name>ann.tif; for instance, the annotated version of image 05.tif would be 05ann.tif.

4.4.3 Images for intra-observer agreement

Once the doctor completed the annotations, we asked him to produce a second annotation for a batch of 30 images (15 SLO, 15 fundus) randomly chosen. We will use In Figure 4.3 an example of annotations made by the same annotator on images fundus and SLO of the same retina.

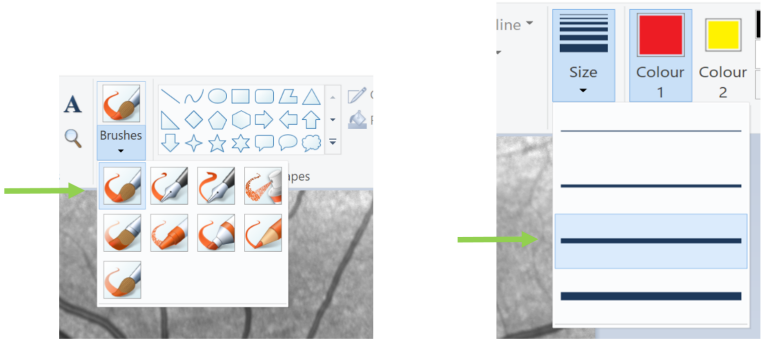

Figure 4.1: *Paint menus. Left: please select the brush indicated by the arrow in the Brushes menu. Right: please select the thickness indicated by the arrow in the Size menu. Colour (red for first-choice contour, yellow for second-choice one) can be selected using the adjacent boxes.*

4.5 From annotated images to ground truth binary images

On each annotated image from one to three contours have been indicated by doctors: the red (more likely the OD contour), the yellow (second choice) and the PPA. For the red and yellow contours, we needed the corresponding segmentation maps. The procedure used for extrapolating the segmentation map from the red contours consists of the extraction of the red dots followed by linear interpolation of such dots. The result is a black image with a white polygon representing the approximation of the area covered by the OD in the original image. The procedure for the yellow contour is very similar. The yellow dots can define alone totally different contour w.r.t. the red one or can represent only a variation in a limited part of the OD contour (more frequently).

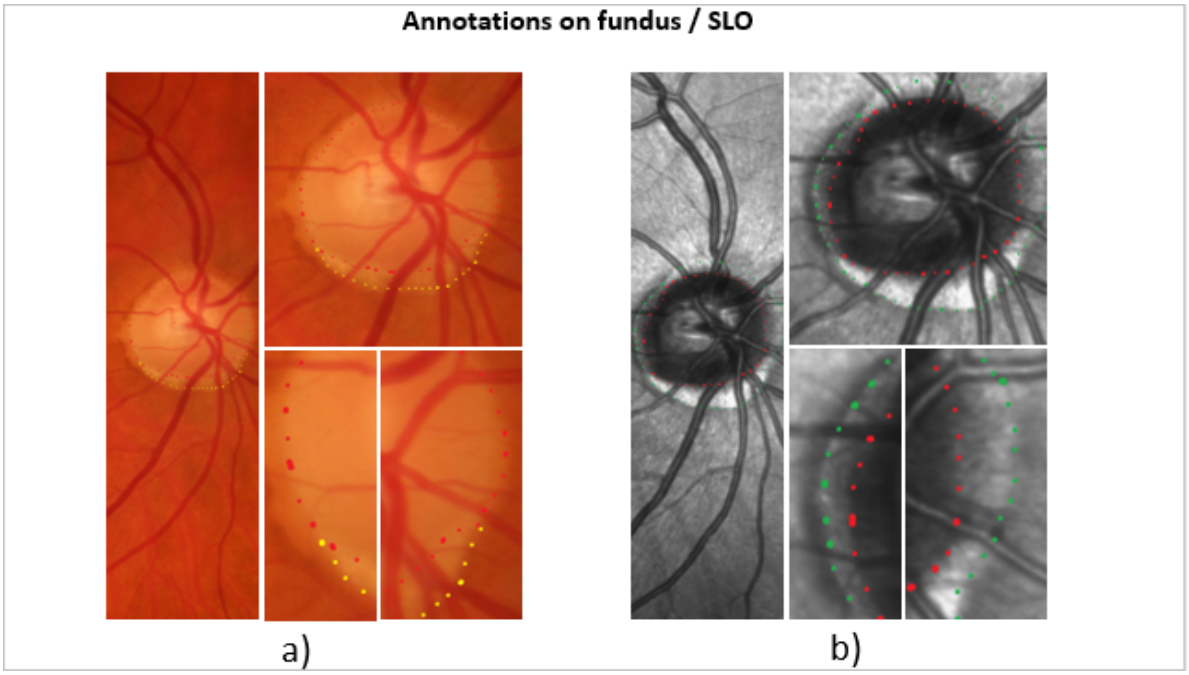

Figure 4.2: *a) Details of an annotated (according to the SOP) fundus image. b) Details of an annotated (according to the SOP) SLO image.*

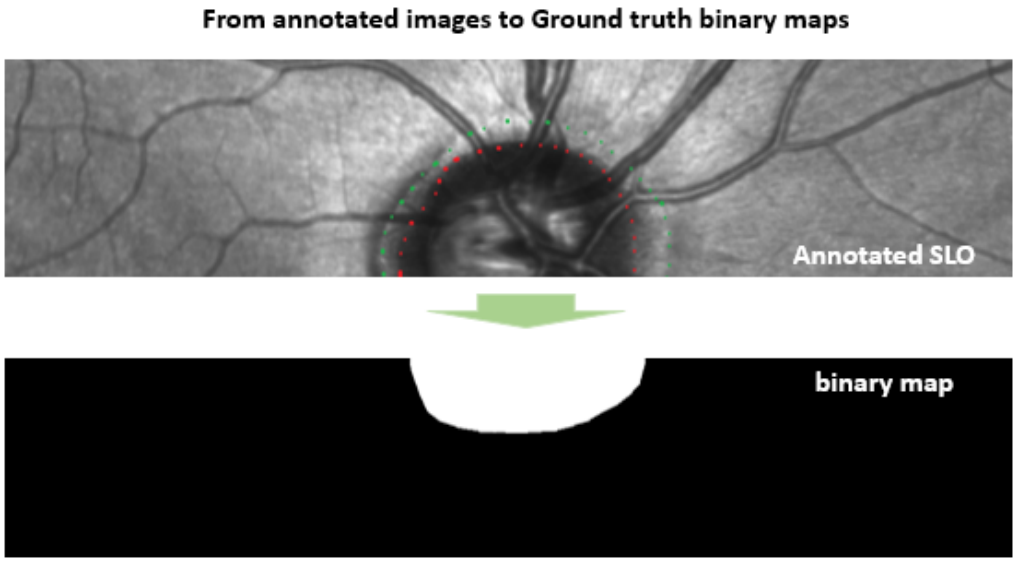

Figure 4.3: *Top, annotated SLO. Bottom, binary map used as ground truth.*

4.6 Statistical analysis

In this section, we are going to analyze the data provided by the ophthalmologists. The overall information consists of annotations of the optic disc in fundus and in SLO images, moreover for each image we have from one to three indications. We have to use this information to extract a statistical description of our annotators, in particular, it is interesting to study the doctors' agreement on both types of images and the "cross-agreement" between annotations on fundus and on SLO. In addition, we will make some, more qualitative, observations about the agreement regarding the PPA annotations.

4.6.1 Data description

The overall dataset consists of 50 pairs of annotated images for each of the 4 doctors (A_1, A_2, A_3, A_4) plus 15 SLO and 15 fundus for repeatability (per doctor). For future comodity we define the following:

- $\boldsymbol{S}$ **:** set of all the annotated SLO images.
  - $\boldsymbol{SR}$ **:** Set of all binary maps related to the red contour.
  - $\boldsymbol{SY}$ **:** Set of all binary maps related to the red contour.
  - $\boldsymbol{SP}$ **:** Set of all annotated SLO images where PPA has been indicated.
- $\boldsymbol{F}$ **:** set of all the annotated fundus images.
  - $\boldsymbol{FR}$ **:** Set of all binary maps related to the red contour.
  - $\boldsymbol{FY}$ **:** Set of all binary maps related to the red contour.
  - $\boldsymbol{FP}$ **:** Set of all annotated SLO images where PPA has been indicated.
- $\boldsymbol{A_1SR}$ **:** indicates the set of binary maps deriving from the red annotations by A_1, on SLO images. With the same rationale, A_1FY indicates the set of binary maps deriving from the yellow annotations by A_1, on fundus images. And so on.
- $\boldsymbol{A_1SP}$ **:** indicates the set of annotated SLO where A_1 has indicated a PPA contour.

In Figure 4.4 a chart showing the number of the different contours indicated by the annotators. From the chart we can notice that both in SLO and fundus in the large majority of the images the doctors have annotated a second contour (the yellow one) suggesting that there is an high intrinsic uncertainty in the definition of the OD border.
Another possible first inference is that doctors are more likely to mark a PPA contour in SLO images (in average ≈ 22 PPAs marked) rather than in fundus (≈ 16).

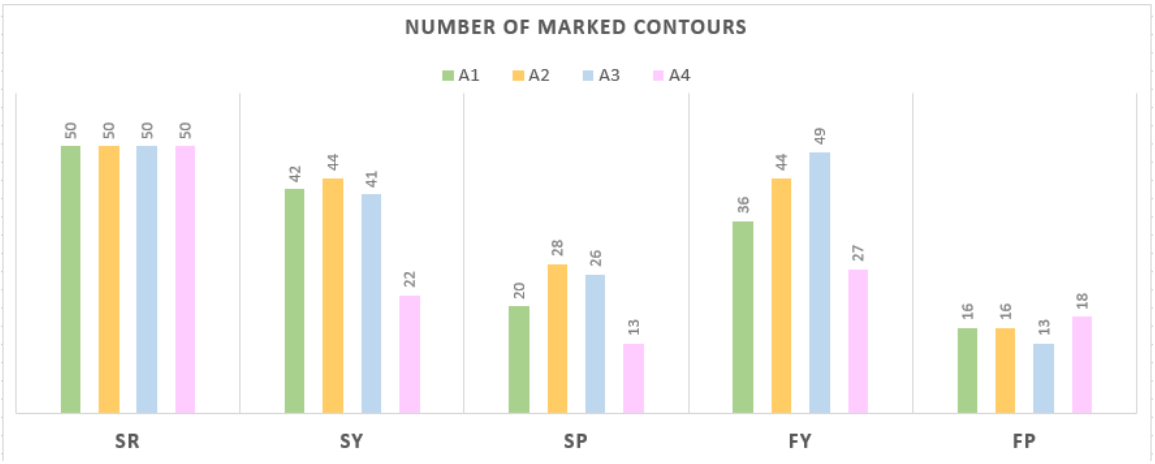

Figure 4.4: *Histogram representing the cardinality of the sets described above. Cardinalities of sets S and F are omitted, in fact, $|S| = |F| = |SR| = |FR| = 50$.*

4.6.2 Annotators' agreement

We would like to have a measure of the annotator's agreement; among the multiple choices available we chose to use the following metrics: Dice coefficient, Jaccard coefficient and mean contour distance (illustrated in Section 2.8). In particular, we will use the Dice coefficient for most of the evaluations and the others as a complement when necessary. In Figure 4.5 an example comparisons between pairs of annotations, we notice that: to two annotations indicating contours substantially different (comparison on the left) corresponds a dice coefficient equals to 0.68; while, to annotations indicating substantially the same contours corresponds a dice coefficient of 0.96. For the interpretation of the Dice coefficient, when comparing two OD segmentation, we can use as a reference the following qualitative grid of grades (from D to A+):

- **D** $[0 - 0.5]$: Error, likely different locations.
- **C** $[0.5 - 0.85]$: Different contours.
- **B** $[0.85 - 0.90]$: Partially matching contours.
- **A** $[0.90 - 0.95]$: Slightly mismatching contours.
- $\mathbf{A}^+$ $[0.95 - 1]$: Same contours.

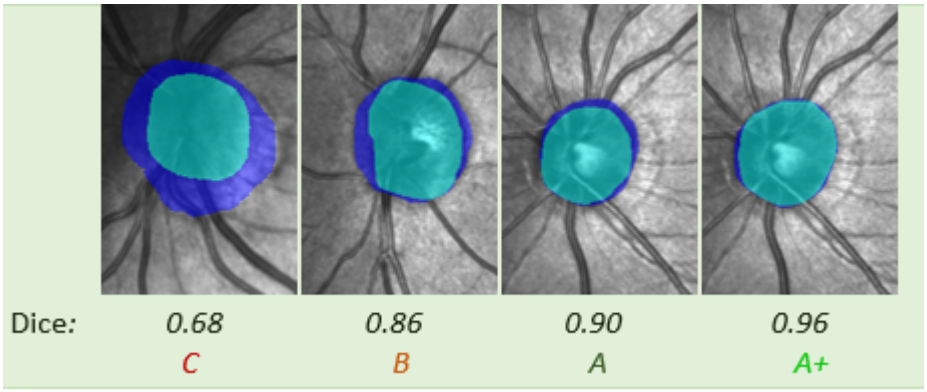

Figure 4.5: *Visualization of the overlap between pairs of annotation, corresponding dice coefficient and grade. The grades are: D for OD with different locations. C for different contours. B for OD with partially matching contours. A for slightly mismatching contours.* A^+ *for same contours.*

4.6.3 Agreement in SLO

In Figure 4.6, the count of the comparisons between annotations (red contours) divided by grades. More in details, for all the possibles coupling of annotators 50 dice coefficients, one per image, have been computed. From the chart, we can make the following observations. Picking randomly two annotations the most likely outcome would result in a grade of A^+. In fact, $P(A^+) = 0.56$. The probability of having two annotators indicating as OD contours two contours totally different (C) is not negligible, $P(C) = 0.05$. In particular, if we look at comparisons between A_1 and A_2 the probability increases to $P(C|A_1/A_2) = 0.12$. The probability of having a good matching $(A \cup A^+)$ is: $P(A \cup A^+) = 0.86$. The probability of having a mismatch $(D \cup C)$ is: $P(D \cup C) = 0.05$. Annotator A_1 is the one which produces the annotations that differs the most from the others (interestingly, A_1 is also the annotator with the highest degree of expertise). Results reported in the table below, 4.1. In Section 4.7 we will report the mean values and related standard deviations.

Event	**Probability**	**Description**
A^+	0.56	perfect match
$A \cup A^+$	0.86	good match
$D \cup C = C$	0.05	mismatch
$C \mid A_1/A_2$	0.12	mismatch between A_1 and A_2

Table 4.1: *Agreement in SlO.*

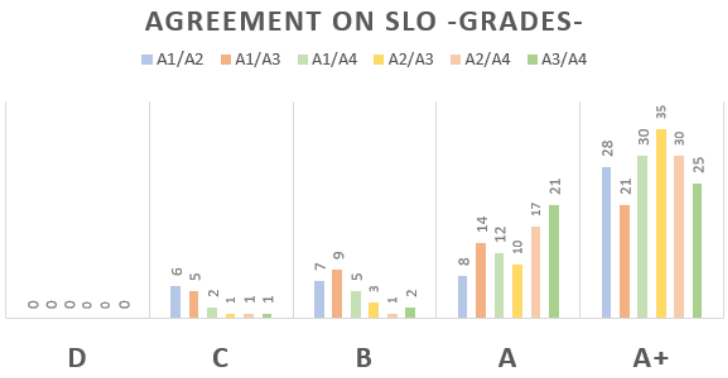

Figure 4.6: *Comparison between couple of annotators, organized by grades (SLO).*

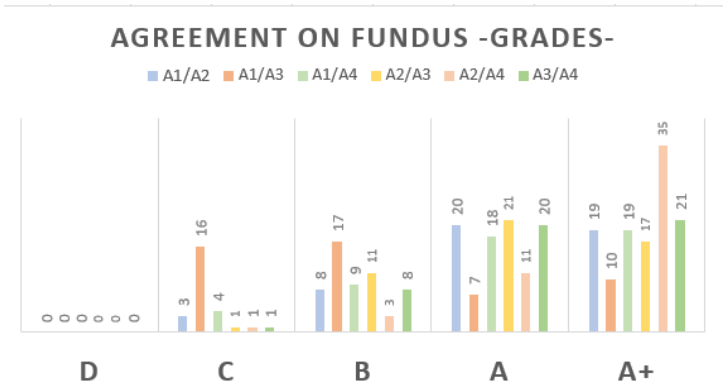

Figure 4.7: *Comparison between couple of annotators, organized by grades (fundus).*

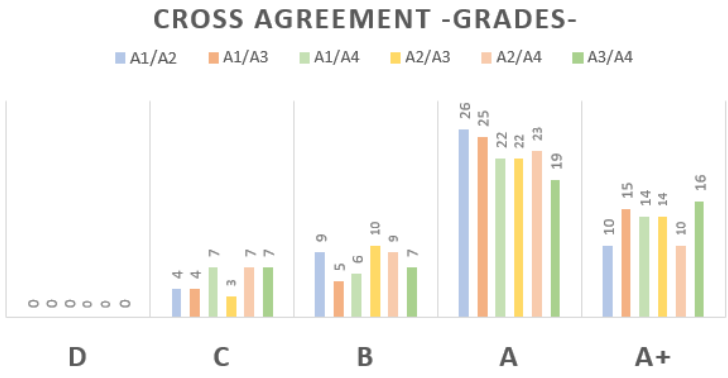

Figure 4.8: *Comparison between couple of annotators, organized by grades (registered fundus/SLO).*

4.6.4 Agreement in fundus

In Figure 4.7, as done for SLO, the counts of comparisons of pairs of fundus annotations divided by grades. Comparing Charts 4.7 and 4.6 we notice that in fundus the probability of getting a perfect matching between two annotations is significantly lower than in SLO. It is also worth to underline the surprisingly high values in the counts related to the events $C \mid A_1/A_3$ and $(B \cup C) \mid A_1/A_3$ that suggests a systematic difference in the definition of the OD contour followed by the two annotators. Results reported in the table below, 4.2.

Event	**Probability**	**Description**
A^+	0.40	perfect match
$A \cup A^+$	0.73	good match
$D \cup C = C$	0.09	mismatch
$C \mid A_1/A_3$	0.32	mismatch between A_1 and A_3
$(C \cup B) \mid A_1/A_3$	0.66	bad matching between A_1 and A_3

Table 4.2: *Agreement in fundus.*

4.6.5 Cross-agreement fundus-SLO

We set up a framework for comparing the annotations made on fundus and the ones on SLO. The framework consists of a multi-modal registration through which we lead the fundus images to match the corresponding SLO. The registration follow a semi-automatic procedure based on methods reported in 2.9.1 and 2.9.2. The set of control points required by such methods are chosen manually. Once the control points are set, using both the methods we compute the transformations functions and by visual inspection (using a checkerboard) we establish which gave the best result. In case none of the results is sufficiently accurate, we choose a different set of control points from which compute the new transformations functions (and so on until the desired accuracy is reached).

From Figure 4.8 we notice, w.r.t. the previous charts, how the counts "shift" to the left. The probability of having a perfect match drop to 0.27, while the probability of a good matching remains on the same level obtained on fundus images ($P(A \cup A^+) = 0.73$).

4.6.6 Intra-annotator agreement

We will refer as "intra-annotator agreement on SLO" to the similarity between the annotations taken by one annotator on the same SLO images (among the set of 15 images for repeatability, Section 4.4.3). While we will consider as intra-annotator-cross-agreement the similarity computed when comparing the SLO and registered fundus annotations of one doctor.
The quantitative results for the intra-annotator agreement will be illustrated in the next

Event	Probability	Description
A^+	0.27	perfect match
$A \cup A^+$	0.73	good match
$D \cup C = C$	0.11	mismatch
$(C \cup B)$	0.27	bad matching

Table 4.3: *Cross-agreement.*

section. Qualitatively we can say that the intra-annotator agreement in SLO and fundus is very high, in general, better than the agreement between different annotators. For as concern the cross-intra-annotator agreement, instead, we can notice that the similarity between annotations produced by the same doctor on the two types of images is not significantly higher than the one measured between annotations from different annotators.

4.7 Agreement: summary

Summarizing the data exposed in the previous sections, we can state that:

1. the definition of OD borders is a difficult task with an intrinsic uncertainty that has to be taken into account. In most of the images, the doctors proposed two different solutions for indicating the contour (red and yellow contours).
2. In SLO images, w.r.t. fundus images, it is more likely for two doctors to annotate the same OD contour ($P_{SLO}(A^+) = 0.56$, $P_{fundus}(A^+) = 0.40$).
3. The intra-annotator agreement in both fundus and SLO is very good.
4. The similarity between annotations on fundus and SLO related to the same retina is relatively low, even between annotations from the same doctor.

In the next page, in Figure 1.9 the agreements and cross-agreement values, in terms of mean Dice coefficient (and related standard deviation), computed between each possible pair of annotators. In Figure 1.10 the corresponding grades. As finally aggregate indexes we report in Table 1.4 the mean agreements coefficients (calculated among all the possible pairs of annotations).

Images	Mean Dice	Std. dev.	Grade
SLO	0.93	0.06	A
fundus	0.92	0.06	A
SLO/fundus	0.89	0.07	B

Table 4.4: *Mean agreement in SLO, fundus and fundus/SLO.*

ANNOTATORS' AGREEMENT

DICE		SLO				fundus			
		A1	A2	A3	A4	A1	A2	A3	A4
SLO	A1	0.94 *0.04*							
	A2	0.92 *0.07*	0.95 *0.03*						
	A3	0.91 *0.07*	0.94 *0.03*	0.95 *0.03*					
	A4	0.94 *0.05*	0.94 *0.03*	0.94 *0.04*	0.97 *0.02*				
fundus	A1	0.88 *0.06*	0.91 *0.06*	0.91 *0.06*	0.89 *0.08*	~			
	A2	0.83 *0.08*	0.88 *0.07*	0.91 *0.06*	0.89 *0.08*	0.92 *0.07*	0.96 *0.02*		
	A3	0.83 *0.07*	0.89 *0.05*	0.92 *0.05*	0.90 *0.08*	0.87 *0.07*	0.93 *0.04*	0.95 *0.03*	
	A4	0.86 *0.07*	0.90 *0.05*	0.92 *0.05*	0.90 *0.08*	0.92 *0.06*	0.95 *0.03*	0.93 *0.04*	0.97 *0.01*

Figure 4.9: *In each element $e_{i,j}$ of the table the mean dice coefficient and its std.*
In the table is possible to distinguish 3 quadrants:
the quadrant related to the agreement in SLO (top-left), in fundus (bottom-right) and the cross-agreement (bottom-left).
In the diagonals of each quadrant the annotators' intra-annotator agreements.
We do not have the data for computing the missing value indicated with the symbol tilde.

ANNOTATORS' AGREEMENT

DICE		SLO				fundus			
		A1	A2	A3	A4	A1	A2	A3	A4
SLO	A1	A							
	A2	A	A+						
	A3	A	A+	A+					
	A4	A	A	A	A+				
fundus	A1	B	A	A	B	~			
	A2	C	B	A	B	A	A+		
	A3	C	B	A	A	B	A	A+	
	A4	B	A	A	A	A	A+	A	A+

Figure 4.10: *In each element $e_{i,j}$ of the table the mean dice coefficient and its std.*
In the table is possible to distinguish 3 quadrants:
the quadrant related to the agreement in SLO (top-left), in fundus (bottom-right) and the cross-agreement (bottom-left).
In the diagonals of each quadrant the annotators' intra-annotator agreements.
We do not have the data for computing the missing value indicated with the symbol tilde.

4.8 PPA analysis

In this section, we analyze and compare the medical annotations of PPA we have collected. We want to have a measure for the doctors' consensus in indicating the PPA contour (green contour) in both fundus and SLO. As we discussed in the first chapter, the PPA is an important biomarker for glaucoma, it can be used for assessing the risk of glaucoma or monitoring the progression of the disease. Hence it is important to know which is the intra and inter-observer agreement in indicating it.

Only by looking at the counts in Figure 4.4 we can notice that in SLO images is more likely for an annotator to indicate the presence of PPA; in fact, $|SP| = 87$ while $|FP| = 61$. Moreover, the variability is higher in SLO, A_2 indicated 28 PPAs, A_4 only 13. In fundus the counts range is from 13 to 18.

A further analysis as been conducted among the sets of annotations A_1, A_2, A_3 provided by the three ophthalmologists[1]. We counted the times in which doctors indicated the presence of the PPA, and we compared[2] the contours they indicated. From such counts we estimated that, given a retinal image, with probability 0.55 one of the doctors will not agree with the others about the presence or absence of the PPA if the image is SLO, with probability 0.43 if the image is a fundus. When the three doctors agree on the presence of PPA in SLO it is likely (prob. equals to 0.83) that they will indicate the same PPA, while in fundus they will disagree annotating the contour with probability 0.83.
From another point of view, given an SLO image where a doctor indicates the contour of a PPA, with probability 0.56 the same contour will be indicated by (at least) another of two doctors as OD contour. Similar result in fundus.

Furthermore, using the registered fundus images, we compared, annotator by annotator, their consensus in indicating the presence of a PPA in SLO and in the corresponding fundus images. When A_1 indicates the presence of a PPA in a SLO image, with probability 0.73 he will be consistent in indicating the presence of the PPA in the corresponding fundus image. For A_2 and A_3 this value decreases respectively to 0.59 and 0.55.

[1] At the time the set of annotations A_4 wasn't available.
[2] Qualitative comparisons, by visual inspection.

4.9 Summary

The doctors' agreement in annotating the OD border corresponds, according to the scale introduced in this chapter, to the grade A (both in fundus and SLO). The intra-observer agreement instead achieves, in average, the grade A^+, that is assigned to annotations indicating exactly the same contours. When comparing the annotations made independently on the two types of images of the same retina, the consensus decreases significantly reaching in average the grade B. Moreover, in most of the images the doctors indicated two possible contours. The agreement in indicating the PPA, but only looking at one type of image at the time, is very low, hardly three doctors will agree on the only presence or absence of the PPA in one image. In fundus, even when they agree on the presence of PPA it is likely that they will indicate different PPA borders. Finally, in both images it is very common (more than 50% of the times) that the contour indicated as PPA by a doctor will be indicated as OD contour by one of the others.

Chapter 5

A deep learning approach to OD segmentation

5.1 About this chapter

The following method exploits deep learning techniques for locating and segmenting the OD in SLO images. The idea of trying a deep learning approach comes by the fact that, as seen in Chapter 3, most of the best results obtained in similar tasks, such as OD segmentation in fundus images, were obtained via CNNs. Although this observation, the choice of this path wasn't an obvious one due to at least two factors:

1. the data set we had access to is relatively small and this can easily lead to overfitting when the model used for learning is very complex.

2. the hardware available was limited in terms of computational power and parallelization possibilities.

To tackle the limitation explicit in point 2 we chose to split the method into two main steps characterized by the use of the images at different levels of resolution. In fact, setting the problem in a straight-forward shape for learning, it is to say to learn a model from the original annotated images, would have been unaffordable in terms computational cost and possibly would have led to a less accurate solution. As said before, we decided to split the procedure into two phases where for each phase a classifier has been built, one for locating the OD and one for segmenting. This is also the choice of other related works ([16],[18]) and it is intuitively guided by the idea that the "amount of information" required for the only localization of the OD is less than the amount needed for an accurate segmentation. According to this idea we notice that 64x64 is a reasonable resolution for properly locate the OD and 512x512 is enough for distinguish clearly the finest features along the OD contour. The entire pipeline followed by the algorithm during the prediction is shown in 5.1 and below described.
Given an SLO image in input, the algorithm resizes the image to 64x64 pixels, then a classifier produces a rough segmentation map of the optic disc. This map is used to locate the OD computing the coordinates of the centre. At this point, the original image (1536x1536)

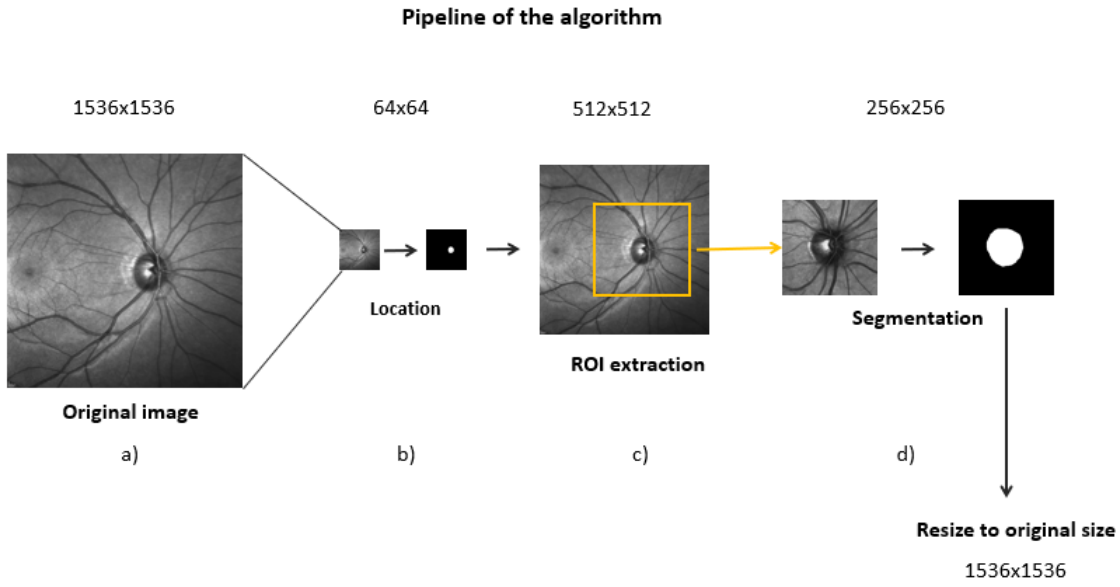

Figure 5.1: *The image shows the pipeline used for the prediction in four key points.*
a) Downsize of the input image to 64x64.
b) Location phase: a classifier produce a binary map which is used to locate the OD.
c) Resize of the original image and cutting of the ROI.
d) A second classifier trained on ROI images computes a new segmentation map. Upsampling to he original resolution, end.

is resized to a resolution of 512x512 and a squared ROI of dimension 256x256 is cropped around the centre coordinates which are accessible from the previous step. The ROI is then given as input to a second classifier that computes a finer segmentation map that, after being resized to the original resolution, represents the final prediction of the algorithm.

5.2 Location/detection

The goal is to build a robust classifier, using CNNs, for locating the OD in low-resolution images (64x64). The location, represented by the OD centre is obtained simply computing the barycentre of the binary map forwarded by the classifier. At this stage there is no need to have a predicted centre perfectly matching the real one, what is really fundamental is that the ROI extracted starting from the predicted centre includes within all the optic disc. In fact, a slight mismatch will make no difference at the next step while an OD not included in the ROI will lead to a failure in the final segmentation.

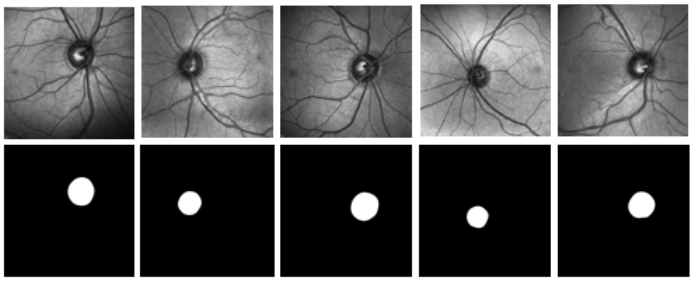

Figure 5.2: *Sample of five pairs of images and related ground truth. The pixels in ground truth images have value 1 where the*

5.2.1 Data set

The data set used for designing the classifier is composed of 120 SLO retinal images. For each image a binary map, representing the ground truth, is generated using ALG_1[1]
We split the data set in: 95 images for the training set, 10 for the validation and 15 for testing. In order to enhance the data, we used standard data augmentation methods such as X/Y-wise reflections, rotation (in a range of [-10, 10] rotation degree) and scaling (in a range of [0.8, 1.2] scaling factors). In Figure 5.2 a sample of five images and corresponding annotations (produced by ALG_1).

5.2.2 Classifier for location (description)

The classifier consists of four different CNNs (A, B, C, D) that are combined for obtaining a single binary map as output. In particular, at training time the four networks are trained independently while at the testing time the output map of each network is merged with the others to get the final result. The merging consists of the AND (pixel-wise) function applied to the four binary maps.
Following, the main features of the networks are listed:

1. A, B, C are designed for a pixel-wise classification task where the size of the output is equal to the input size.

2. A, B, and C share the same architecture and the same training set. The three networks only differ in initialization the weights.

3. D is designed for region-wise classification and the size of the output is 4x4. To set the problem as a region-wise classification problem the ground truth images have been

[1]At this stage, we are looking for the only location of the optic disc. For this reason we consider legit to use annotations/ground truth images that are not provided by doctors (in order to use images that otherwise would not have ground truth).

generated as written in the note[2].

4. The parameters of all the networks have been initialized by independently sampling from a normal distribution with zero mean and standard deviation 0.01.

5. The loss function used to train the networks is class balanced cross-entropy (the average ratio between OD pixels and not OD is: 0.004).

In Figure 5.3 and 5.4 the two architectures used are shown in details, it is interesting to observe that the models are very "light" as the number of nodes per network is relatively small and the overall amount of learnable parameters is: 3.135.

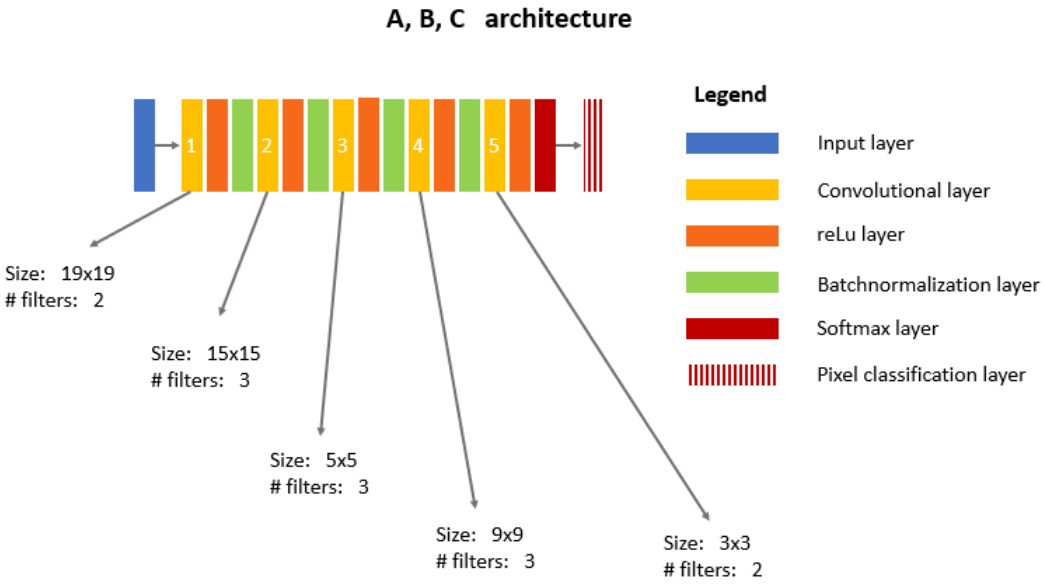

Figure 5.3: *Networks A, B, C. In all the convolutional layers the input has been zero-padded in order to keep the output at size of 64x64. The stride is set to 1.*

The networks have been trained with stochastic gradient descend as solver and the following set of parameters: mini-batch size equals to 3, 30 epochs, momentum set to 0.7, L2 regularization to 0.005 and learning rate equals to 10^{-3}.

It is worth noticing that, despite A, B, and C share the same architecture, the outputs are different because of the two stochastic processes that take place during the training: the initialization of the weights and the generation of the mini-batches (the seed for random number generation has been changed before each training).

[2]For each ground truth image at 64x64 resolution (i_{64}) The ground truth images (i_4) used for training network D has been derived as follows: i_{64} is divided with a grid 4x4, each square of the grid corresponds to a pixel of i_4. Each pixel of i_4 is set to 1 if at least one of the pixels in the related grid-square is equal to 1, to 0 otherwise.

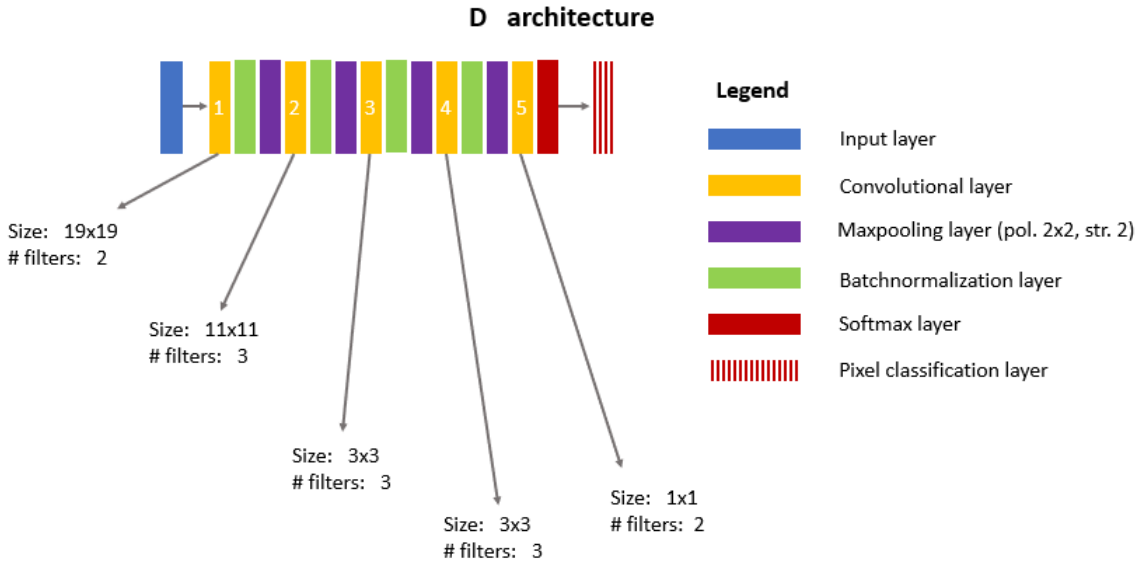

Figure 5.4: *Network D.*

5.2.3 Classifier design and motivation

The original idea was to build a single CNN for locating the OD. To find the right model we tried very different architectures following a simple-to-complex strategy, we started from basic networks of one layer and we proceeded gradually adding complexity to the network. Moreover, we tried many architectures inspired by the U-net, VGG and Inception models. After many attempts, we found out very hard to obtain a satisfiable solution to this particular problem using a single network. Despite the fact that it is easy to train a network that works very well on the average-looking SLO image, it is difficult to get one that provides a useful result for locating the OD in all the validation images. Hence we chose to move to the described set up of four CNNs. The rationale behind this choice is clear when looking at the results of network A illustrated in Figure 5.5.

As it is possible to infer from Figure 5.5, the false negative rate of A is equal to zero. In other words, net A is always (at least in our experiments) able to find all the pixels belonging to the optic disc. This observation led us to the idea that the intersection of the output maps of different networks, presenting this characteristic (OD accuracy = 100%), would be more accurate than the single output maps of each network. In particular, the higher the number of such networks the better would be the final results. We empirically proved this hypothesis, in Figure 5.6 it is shown the output of A, B, C and the resulting intersection on a subset of testing images.

As explained in the previous Section we added a fourth network, net D, to the pool. Net D is a region-wise classifier that, given an image as input returns as output a 4x4 binary matrix in which each element represents the prediction of the optic disc absence/presence

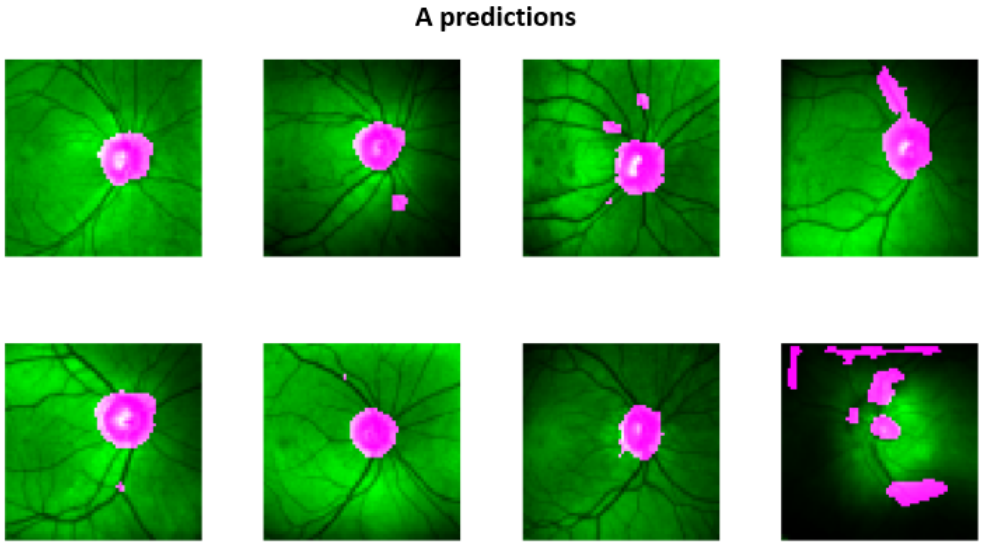

Figure 5.5: *Subset of 8 testing images; In pink, pixels of the input image classified by A as OD, in green the pixels classified as not OD.*

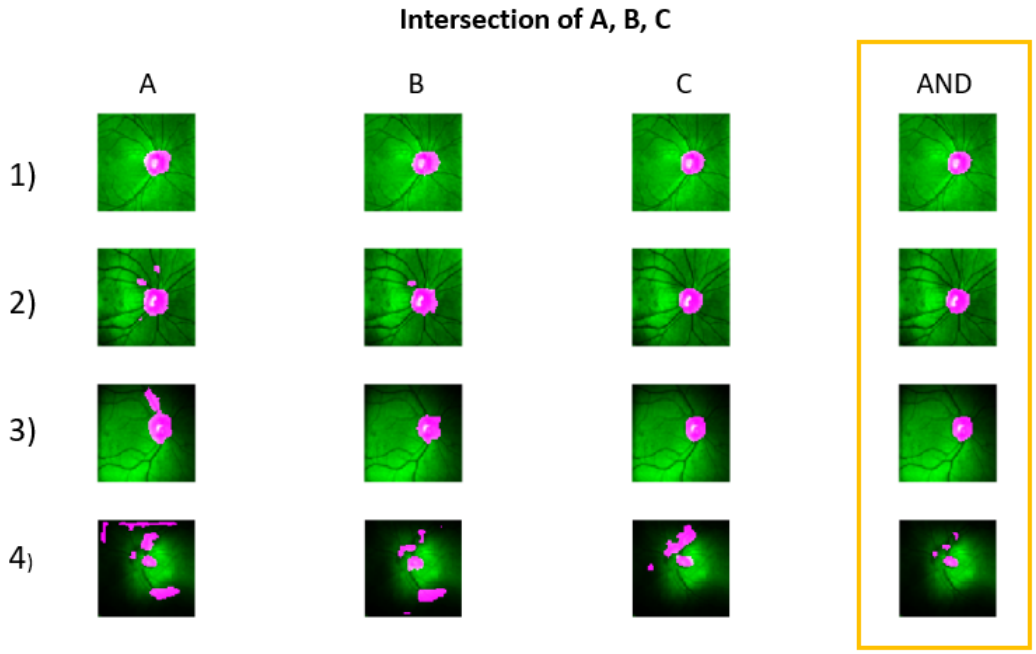

Figure 5.6: *Subset of 4 testing images; In each row the results obtained with nets A, B, C and their intersection. In pink, pixels of the input image classified by A as OD, in green the pixels classified as not OD.*

within the corresponding 16x16 pixels square in the input image. The main characteristic of D is the sequence of convolutional and max-pooling layers that gradually decrease the number of features in the net. The addition of this network do not improve substantially the results in our data set, nevertheless we consider adding this network useful in the sense of improving the robustness of the system in case of hard-looking images. In fact, due to the intrinsic nature[3] of this net and the peculiar task for which has been designed we consider it more capable, w.r.t. to nets A, B, and C, to identify the correct ROI in challenging images. In Figure 5.7 predictions of D among a subset of testing images.

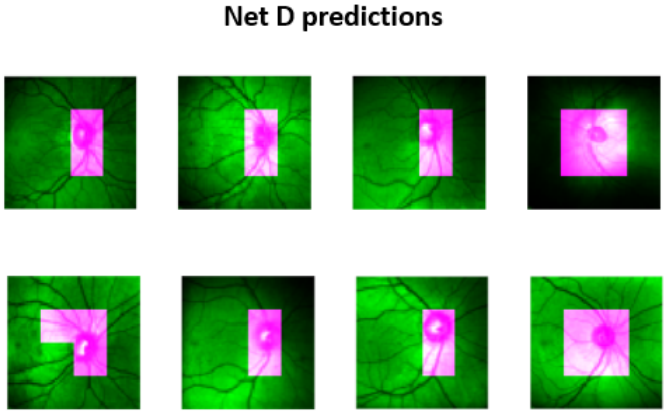

Figure 5.7: *Subset of 8 testing images; In pink, pixels of the input image classified by A as OD, in green the pixels classified as not OD.*

5.3 ROI extraction

The final classifier for locating the optic disc return a segmentation map that is the combination of the maps returned by the four sub-classifiers. We use this map to estimate the centre of the optic disc by computing the barycentre of the point classified as OD. With the centre is possible to extract the ROI from the original images. For the next step that consists of building a CNN to get the final segmentation of the optic disc we extract the ROI from the original SLO images downsized to the resolution of 512x512. The chosen ROI shape is a square of 256x256 centered in the predicted optic disc centre.

5.4 Segmentation

The goal is to build a classifier capable to get an accurate segmentation of the optic disc. To achieve that, we made some preliminary choice and observation in order to simplify the problem to our best.

[3]In fact, thank to the sequence of pooling layers the final classification of net D (for each region) strongly depends on all the regions of the input image.

1. We noticed that wasn't really necessary to use the images at the original resolution (1536x1536) and we decided to downsize to 512x512, resolution that still allows to clearly discriminate the OD edges and borders from the background.

2. We chose to work only on the extracted ROIs (256x256) to decrease the computational cost of the training process.

3. We chose to use transfer learning to exploit as much information as possible.

The idea of using transfer learning (point 3) derives by the fact that we had available 120 SLO images, of which only 50 annotated by doctors. In order to try to not waste available data, we decided to split the training process in two phases. In the first phase we train a network on the 70 images with no annotation using as ground truth the segmentation maps produced by ALG_1 (described in Chapter 3). ALG_1, as we will lately discuss, is very accurate but it is not a doctor and the quality of the features learned by the network at this phase is somehow dependent on the reliability of ALG_1. Hence, in the second phase the trained network is re-trained on the annotated images.

5.4.1 Data set

As explained in the previous Section the the data set is composed of two subsets of SLO ROI images (resolution: 256x256): annotated by annotators (DAnn) and by ALG_1 (Dalg1).

- **DAlg1:** We split this data set in 65 images for training and 5 for validation.

- **DAnn:** each of the 50 images has been independently annotated by three doctors, we chose to split into 25 images for training, 5 for validation and 20 for testing. More in details, we decided to train the classifier using as ground truth all the annotators. Practically, for each SLO image, two copies have been added to the data set and then each triplet of the same image has been associated with three different annotations. Different choices were considered such as training with only one annotator, the most "similar" to the others or the one with the highest repeatability score. We took this decision to try to not fit any annotator but a sort of intersection of them.

In order to enhance the data at each training phase we used standard data augmentation methods such as X/Y-wise reflections, rotation (in a range of [-10, 10] rotation degree) and scaling (in a range of [0.8, 1.2] scaling factors). In Figure 5.8 and 5.9 a sample of images from the two data sets with the corresponding ground truth images.

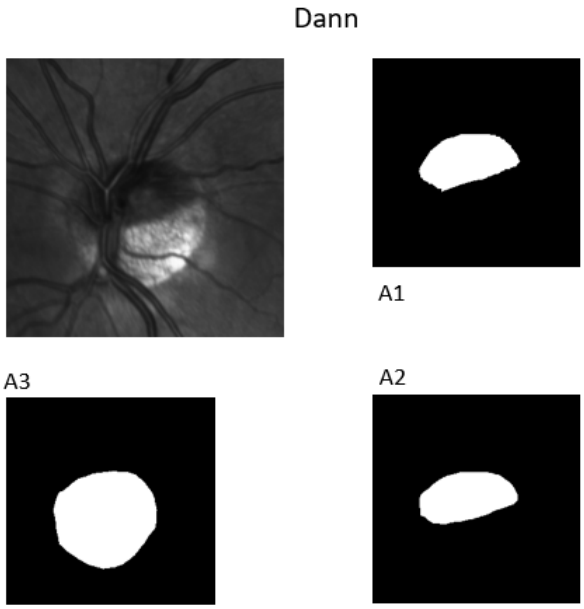

Figure 5.8: *A image from Dann with the corresponding 3 ground truth images, one per annotator (A1, A2, A3).*

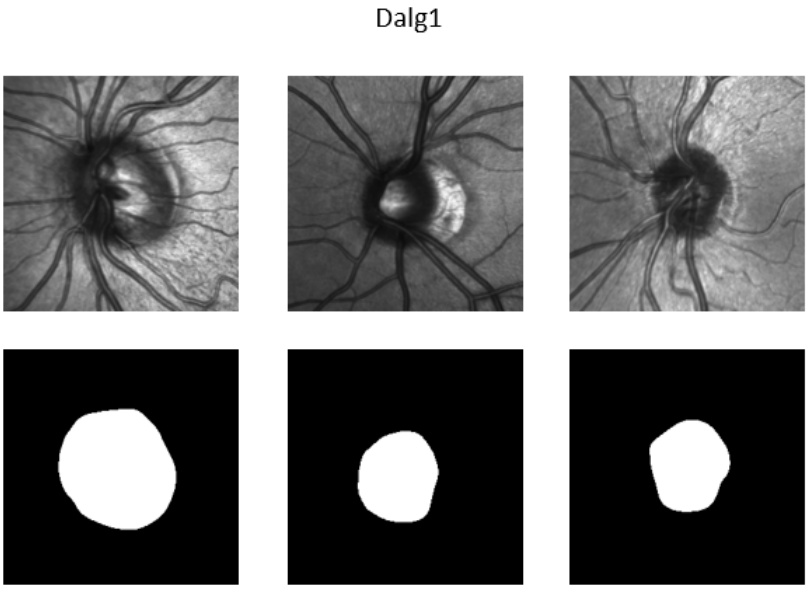

Figure 5.9: *sample of images from Dalg1 with the corresponding ground truth images.*

5.4.2 Classifier architecture

In Figure 5.10 the architecture of the network that, among several experiments works better on the validation set (Dalg1). The model is inspired by the U-net architecture, it is to say a net featured by a first encoder stage and a consequent decoder phase. The main features of the net are:

- The network is composed of 52 layers, and the total amount of learnable parameters is: 40.523.
- The output layer is the "dice-pixel-classification" layer, in fact we noticed a relevant improvement of the results using the dice-loss function instead of cross-entropy for training.
- The encoder stage consists of a repetition of the layers pattern: convolutional, batch-normalization, convolutional, batch-normalization, max-pooling.
- The up-sampling in the decoding phase is obtained via transposed-convolutional layers.

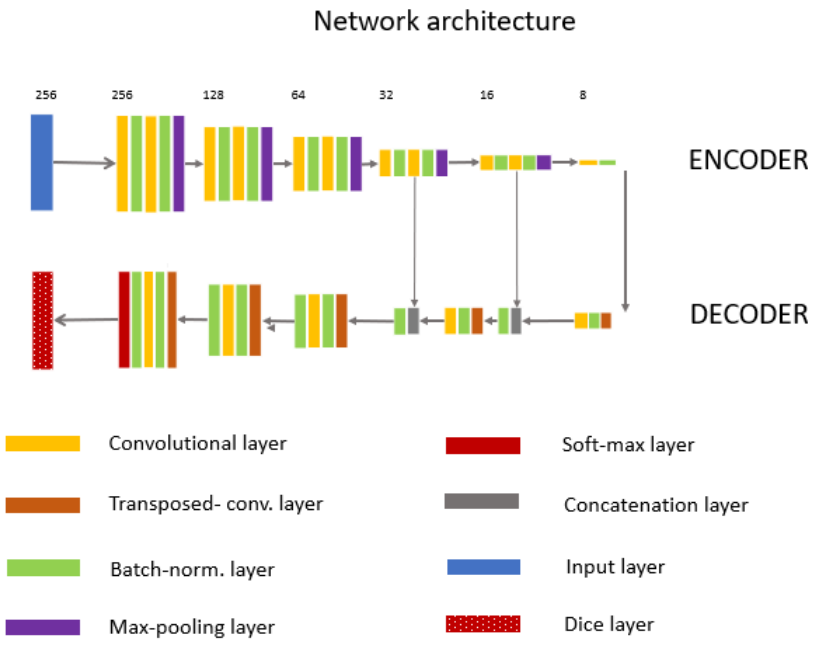

Figure 5.10: *Network architecture, in table A of the appendix all the details regarding number and size of the filters for each layer.*

5.5 Training

During the first phase (on Dalg1), the network has been trained using stochastic gradient descend with momentum as solver, mini-batch size 5, L2 regularization weight equals to 0.005, learning rate 10^{-3} for 40 epochs and 10^{-4} for other 10 epochs. In Figure 5.11 a sample of the output maps produced by the classifier obtained at this stage.

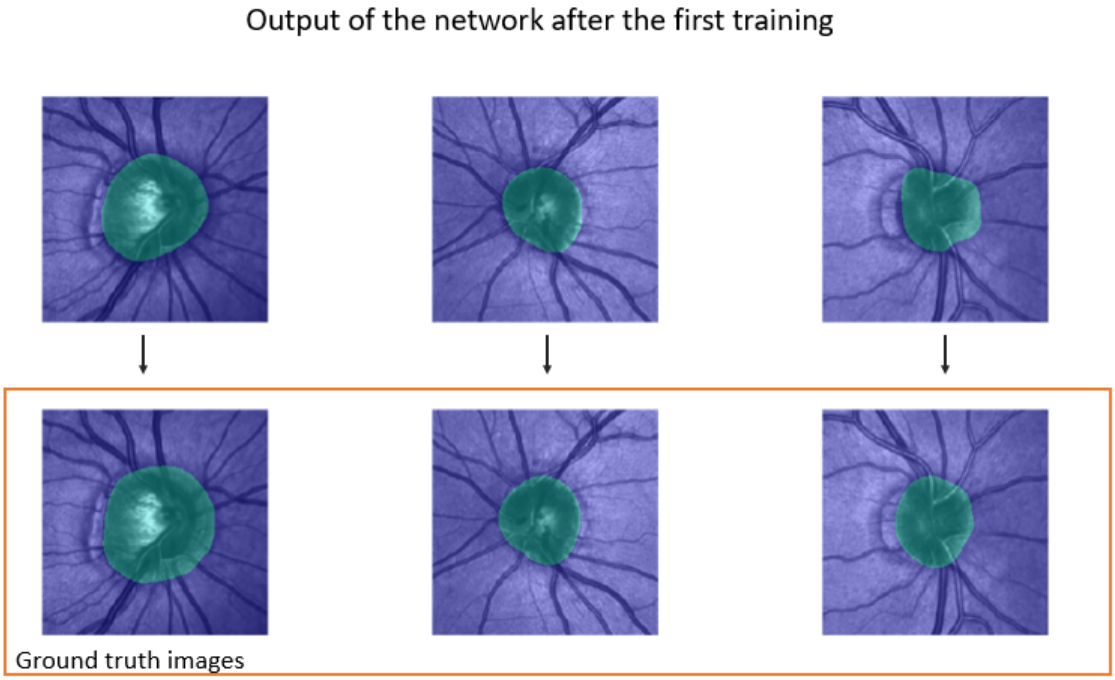

Figure 5.11: *Output of the network after the training on Dalg1. In the first row, the output maps are shown overlapped to the input SLO image; in purple, pixels classified as "not OD" in cyan as "OD". In the second row the ground truth images (obtained with Alg1).*

The trained network is then re-trained on Dann using: stochastic gradient descend with momentum as solver, mini-batch size 5, L2 regularization weight equals to 0.005, learning rate 10^{-2} for 10 epochs and then dropping the learning rate of a factor of 0.5 every 5 epochs until reaching 35 epochs.

5.6 Summary

The presented method consists of a deep learning approach for the automatic segmentation of the OD in fundus images. The method exploits two classifiers, the first classifier is composed of four CNNs and provides a reliable localization of the OD in downsized SLO images, the second classifier is a single CNN which produces the finer and final segmentation. The second classifier has been trained in two phases; at first, on a dataset of 65 images using as ground truth the segmentation maps obtained by ALG_1, secondly on a smaller data set where medical annotations for ground truth were provided.

Chapter 6

Experimental results

6.1 About this chapter

In this chapter, we will discuss the quality of the optic disc segmentations produced by the presented method by comparing with medical annotations. Moreover we will make a comparisons with the algorithm for OD segmentation in SLO images ALG_1, presented in [15] and the VAMPIRE algorithm [11], designed for fundus images. Furthermore, we will investigate the effectiveness of the key choices that have led the pipeline of the algorithm to be the one presented. As done in Chapter "Medical annotations", most of the comparisons will consist of a measure of similarity through Dice-Søresen coefficient between segmentations maps from different sources and the assignment of the corresponding grades: D, B, A, A^+.

6.2 Method and annotators' agreement

Because we have used 30[1] annotated images for training the core classifier of the method M, we have 20 images left for testing it. We can assess the agreement between M and the annotators $A_s = \{A_{s,1}, A_{s,2}, A_{s,3}, A_{s,4}\}$ by counting the number of comparisons resulting in a good/bad matching. In Figure 6.1 the comparisons of the method and the annotators while in Figure 6.2, comparison between the annotators (on the 20 testing images). From the charts, we can infer the probabilities in Table 6.1. We notice, how in general it is more likely to have a perfect matching between two manual annotations $P_{A_s/A_s}(A^+) = 0.56$ w.r.t. one automatic and one manual ($P_{M/A_s}(A^+) = 0.21$). However, the gap significantly decreases when we take into account the probabilities of having a good matching ($A \cup A^+$); in fact $P_{A_s/A_s}(A \cup A^+) = 0.85$ and $P_{M/A_s}(A \cup A^+) = 0.75$. From Table 6.1 we can also observe that the agreement between the method and some of the annotators can be better than the agreement between two annotators (for example comparing $M/A_{s,4}$ with $A_{s,3}/A_{s,4}$). The comparison $A_{s,2}/A_{s,4}$ is the one that achieves the highest number of comparisons resulting in grade A^+.

[1] 25 for training, 5 for validation.

Event	A_s/A_s	M/A_s	$M/A_{s,4}$	$A_{s,3}/A_{s,4}$	$A_{s,2}/A_{s,4}$	**Description**
A^+	0.56	0.21	0.25	0.25	0.65	perfect match
$A \cup A^+$	0.85	0.75	0.90	0.75	0.95	good match
$C \cup B$	0.15	0.25	0.10	0.25	0.05	bad match
C	0.07	0.05	0.05	0.05	0.05	mismatch

Table 6.1: *Estimation of the significant probabilities in SLO.*

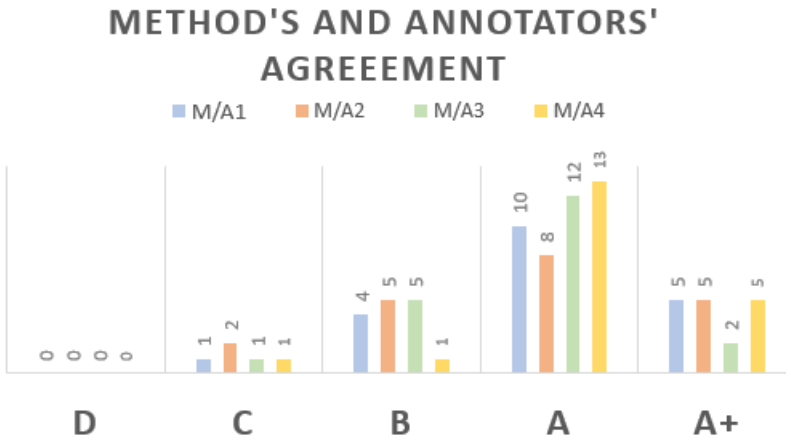

Figure 6.1: *Counts of comparisons between M and each annotator, organized by grades.*

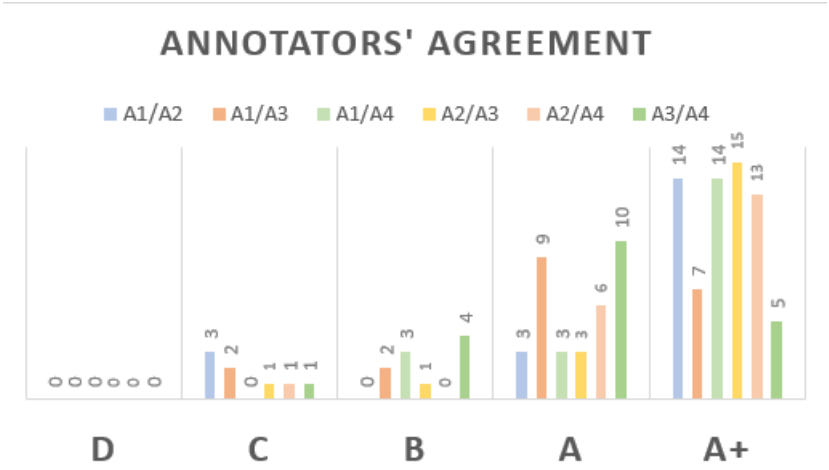

Figure 6.2: *Counts of comparisons between each possible pairs of annotators A_i/A_j, organized by grades. Data set of 20 SLO images.*

6.3 Comparisons with ALG1 and VAMPIRE

ALG_1 and the VAMPIRE algorithm V are both non-learning methods, then we can test the performances among all the data Set, in particular, the binary maps obtained with V are related to the fundus images and then have to be compared with the corresponding medical annotations. We summarize the results in Table 6.2. From Table 6.2 we can make the following observations:

1. The presented method M seems to work generally better than ALG_1, in fact, it is more likely for M to produce OD segmentations that achieve a good matching score with the ground truth.

2. M seems to work similarly to V when comparing with the corresponding ground truths. Differently from V, M is less likely to get a mismatch.

3. The pairs of type $M/A_{s,i}$ that produce the higher agreements are comparable with the pairs of the type $A_{s,i}/A_{s,j}$ the produce the lower (6.1).

Event	A_s/A_s	A_f/A_f	V/A_f	ALG_1/A_s	ALG_1/A_s	M/A_s	**Description**
A^+	0.56	0.40	0.18	0.20	0.14	0.21	perfect match
$A \cup A^+$	0.85	0.73	0.63	0.65	0.64	0.75	good match
$C \cup B$	0.15	0.27	0.37	0.35	0.36	0.25	bad match
C	0.07	0.09	0.20	0.16	0.12	0.05	mismatch
N. images	50	50	50	50	20	20	-

Table 6.2: *Estimation of significant probabilities for: annotations in SLO and fundus* $(A_s/A_s, A_f/A_f)$*, the method presented in [15]* (ALG_1)*, the method presented* (M) *and VAMPIRE (V).*

In Figure 6.3 a further comparison between the annotations and the algorithms is illustrated. For each set of segmentation maps (produced manually or automatically) and for each of the 20 testing images we computed the mean Dice coefficient. Hence, we order (ascending) those results obtaining a crescent curve for each method or set of annotations. Ideally, we would like to have as a result a line constantly equals to 1. The comparisons are made between: annotators in SLO (A_s), the presented method M and A_s, ALG_1 and A_s, annotators in fundus A_f, V and A_f[2]. From this comparison we can observe that M do not reach the performances achieved by the annotators, but, between the automatic methods compared is the one which produces results with highest similarity with the corresponding medical annotation.

[2]The annotations A_f are made on the corresponding set, of 20 fundus images, to the SLO testing set. The difficulty and the appearance of the OD in this is not necessarily correlated to the SLO set. Nevertheless, it is interesting to compare A_f with V.

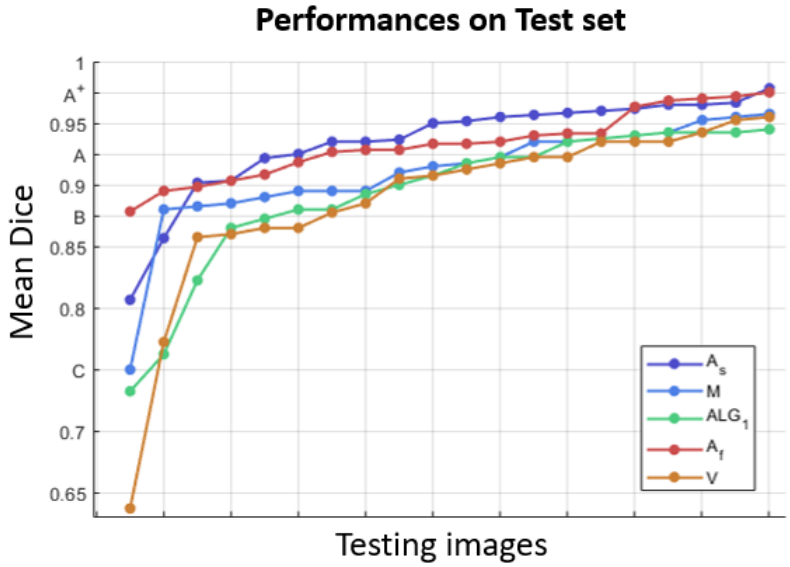

Figure 6.3: *The curve A_s is obtained by computing the mean agreement, between annotators in SLO, per image and ordering from the lowest to the highest. The other curves are the result of the same procedure. It is important to notice that because each curve has been order independently, the order of the related images on x-axes is not necessarily the same.*

6.4 Performances summary

In Table 6.4 we report the aggregate indexes for representing the agreement between the automatic algorithms and the annotators. It is worth noticing that in [11] where V has been presented it is claimed a Jaccard index, between V and the reference annotations[3] of 0.88. Moreover this value results to be higher than the same index computed between the annotators. In our experiments the mean Jaccard index is 0.82 and, more important, is significantly lower than the index computed for the annotators (0.88 for A_s) outlining how the assessment of the performance is strongly dependent on the testing data.

Segmentations	Dice	Jaccard	δ_{mean}	Description	Test images
A_s	0.93 (0.06)	0.88 (0.09)	15 (12.98)	manual	50 SLO
A_f	0.92 (0.06)	0.88 (0.09)	20 (14.96)	manual	50 fundus
M	0.91 (0.05)	0.84 (0.08)	21 (11.00)	automatic	20 SLO
ALG_1	0.90 (0.07)	0.82 (0.10)	24 (16.00)	automatic	50 SLO
V	0.89 (0.08)	0.82 (0.08)	28 (20.81)	automatic	50 fundus

Table 6.3: *Numeric values indicated as mean value and standard deviation in brackets.*

[3]The algorithm was tested on the MESSIDOR data set, a public set of annotated fundus camera images, link at http://www.adcis.net/en/third-party/messidor/.

Conclusions

We presented a novel method, based on deep learning, for the automatic localization and segmentation of the optic disc (OD) in scanning laser ophthalmoscope (SLO) images.
The algorithm has been tested on 20 SLO images, where, compared with the reference medical annotations of the OD, achieves the mean similarity index (Dice-Søresen coefficient) of 0.91. The method performs slightly better than ALG_1 (the only other method that has been proposed so far for solving the same task). The performances of our approach are comparable with the performances of V (algorithm designed for OD segmentation in fundus images). Although the encouraging results, the similarity index computed by testing each ophthalmologist against the others is higher and equals to 0.93 suggesting that there is still margin for further improvements. In fact, we would like to have the solutions provided by the automatic method to be indistinguishable from the solutions provided by specialized doctors. Because the proposed method is based on a learning approach we can state that the quality of the results is strongly dependent on the richness of the training data. We believe that by increasing the number of training examples this method it is very likely to reach the target accuracy.

A relevant finding of this work comes by the analysis of the collected medical annotations from which we can outline that, despite the mean annotators' agreement is relatively high, it is not negligible the probability of having doctors indicating contours that are substantially different. In particular, this is true for 5% of the comparisons between pairs of SLO annotations, 9% in fundus images and 11% when comparing SLO and corresponding fundus annotations. We believe that further research, from a medical perspective, on the definition of the OD border and its features related to the different imaging techniques is needed.

Acknowledgements

This work was supported by NHS Lothian R&D, Edinburgh Imaging and the Edinburgh Clinical Research Facility at the University of Edinburgh. This work has been funded by the ERASMUS Mobility Studentships.

Special acknowledgements to the enthusiastic supervision of Professor E. Trucco, to the availability of Prof A. Facchinetti and to the patience of my "colleague" S. Mattera.

Bibliography

[1] T. Mitchell, Machine Learning. McGraw-Hill, (1997).

[2] Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, (2014).

[3] Rosenblatt, Frank. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, (1957).

[4] Nair, Vinod; Hinton, Geoffrey E., "Rectified Linear Units Improve Restricted Boltzmann Machines", 27th International Conference on International Conference on Machine Learning, ICML'10, USA: Omnipress, pp. 807–814, ISBN 9781605589077 (2010).

[5] Sergey Ioffe, Christian Szegedy Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift ICML, Google Inc., (2015).

[6] Zeiler, Matthew D., et al. "Deconvolutional networks." Computer Vision and Pattern Recognition (CVPR), IEEE, (2010).

[7] Goshtasby, Ardeshir, "Piecewise linear mapping functions for image registration," Pattern Recognition, Vol. 19, pp. 459-466, (1986).

[8] Goshtasby, Ardeshir, "Image registration by local approximation methods," Image and Vision Computing, Vol. 6, pp. 255-261, (1988)

[9] Crum, William R., Oscar Camara, and Derek LG Hill. "Generalized overlap measures for evaluation and validation in medical image analysis." IEEE transactions on medical imaging 25.11, pp. 1451-1461, (2006):

[10] Sudre, Carole H., et al. "Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations." Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, Cham, pp. 240-248, (2017).

[11] A. Giachetti, L. Ballerini, and E. Trucco, "Accurate and reliable segmentation of the optic disc in digital fundus images," Journal of Medical Imaging 1, 024001 (2014).

[12] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, published at ICLR, University of Oxford, (2015).

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, University of Freiburg, Germany, (2015)

[14] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbel´aez, and Luc Van Gool, Deep Retinal Image Understanding, Springer International Publishing, MICCAI 2016, Part II, LNCS 9901, pp. 140–148, (2016).

[15] Sabrina Mattera, Stefano Gennari, Andrew Tatham, Sirjhun Patel, Fraser Peck, Obaid Kousha, Tom Macgillivray, Emanuele Trucco, On optic disc contour detection in scanning laser ophthalmoscope, 11th SINAPSE Annual Scientific Meeting, Dundee, (2019).

[16] A. Sevastopolsky, Optic Disc and Cup Segmentation Methods for Glaucoma Detection with Modification of U-Net Convolutional Neural Network,ISSN 1054-6618, Pattern Recognition and Image Analysis, Vol. 27, No. 3, pp. 618–624. © Pleiades Publishing, Ltd., (2017).

[17] Zilly J, Buhmann J M, Mahapatra D. Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation[J]. Comput Med Imaging Graph, 55:28-41, (2017).

[18] Zilly J, Buhmann J M, Mahapatra D. Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation[J]. Comput Med Imaging Graph, 55:28-41, (2017).

[19] Pengzhi Qin1, Linyan Wang2, Hongbing Lv, Optic disc and Cup Segmentation Based on Deep Learning, IEEE 3rd Information Technology,Networking,Electronic and Automation Control Conference (ITNEC), (2019).

[20] MacGillivray TJ, Trucco E, Cameron JR, Dhillon B, Houston JG, van Beek EJR, Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions. Br J Radiol, 87:20130832, (2014).

[21] Sarah McGrory, James R. Cameron, Enrico Pellegrini, Claire Warren, Fergus N. Doubal, Ian J. Deary, Baljean Dhillon, Joanna M. Wardlaw, Emanuele Trucco, Thomas J. MacGillivray, The application of retinal fundus camera imaging in dementia: A systematic review, Published by Elsevier Inc. (2016).

[22] Patton N, Aslam T, MacGillivray T., Deary I., Dhillon B., Eikelboom R., Yogesan K., Constable I. Retinal image analysis: concepts, applications and potential. Published by Elsevier Inc.(2005)

[23] Shankaranarayana S.M., Ram K., Mitra K., Sivaprakasam M. (2017) Joint Optic Disc and Cup Segmentation Using Fully Convolutional and Adversarial Networks. In: Cardoso M. et al. (eds) Fetal, Infant and Ophthalmic Medical Image Analysis. OMIA, FIFI 2017. Lecture Notes in Computer Science, vol 10554. Springer, Cham, (2017).

[24] http://image-net.org/

[25] https://preventdementia.co.uk/.

[26] http://www.oct-optovue.com/oct-retina/oct-retina.html.

[27] https://www.intechopen.com/books/the-mystery-of-glaucoma/the-optic-nerve-in-glaucoma

[28] https://www.opsweb.org/page/SLO

[29] https://vampire.computing.dundee.ac.uk/tools.html

[30] https://www.opsweb.org/page/octimaging