N° Ordre...../FT/UMBB/2023

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES

Faculté de Technologie

Thèse de Doctorat

Présentée par :

Mohamed Elssaleh Bachiri

En vue de l'obtention du diplôme de **DOCTORAT - LMD** en **:**

Filière : Génie Biomédical

Spécialité : Instrumentation Biomédicale

Application de deep learning dans les images médicales

Soutenue le 12 Novembre 2023, devant le jury composé de :

Mr	Ammar	Mohammed	Prof.	Univ. Boumerdes	Président
Mr	Rahmoune	Fayçal	Prof.	Univ. Boumerdes	Directeur
Mr	Rahmoune	Adel	MCA	Univ. Boumerdes	Co- Directeur
Mr	Messaoudi	Noureddine	MCA	Univ. Boumerdes	Examinateur
Mr	Benlatreche	Mohamed Salah	MCA	C. U. Mila	Examinateur
Mr	Omari	Tahar	MCA	Univ. Boumerdes	Examinateur

Année Universitaire 2022/2023

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

M'HAMED BOUGARA UNIVERSITY – BOUMERDES

Faculty of Technology

PhD Dissertation

Presented by:

Mohamed Elssaleh Bachiri

In order to obtain the degree of **PhD - LMD** in:

Field: Biomedical Engineering

Speciality: Biomedical Instrumentation

APPLICATION OF DEEP LEARNING IN MEDICAL IMAGES

Defended on November 12th 2023, before defense committee members:

Mr	Ammar	Mohammed	Prof.	Univ. Boumerdes	President
Mr	Rahmoune	Fayçal	Prof.	Univ. Boumerdes	Supervisor
Mr	Rahmoune	Adel	MCA	Univ. Boumerdes	Co- Supervisor
Mr	Messaoudi	Noureddine	MCA	Univ. Boumerdes	Examinator
Mr	Benlatreche	Mohamed Salah	MCA	C. U. Mila	Examinator
Mr	Omari	Tahar	MCA	Univ. Boumerdes	Examinator

Academic Year 2022/2023

Abstract

The segmentation of medical images and structures is a necessary step for image analysis. Image segmentation provides us with valuable information about abnormal distortions and changes, which allows us to process these this is as well as follow them. We are interested about the retinal vessels that are used in the diagnosis of diseases such as diabetes and hypertension, ophthalmological evaluation, and various medical fields. The process of segmentation of retinal blood vessels is a complex and arduous matter due to the physiological structure. Therefore, segmentation must be automatic and effective. Various contributions have been made to the segmentation of retinal vessels in the past decades. Due to the development of deep learning and its use in various fields, especially computer vision, different methods have been implemented to segment blood vessels by deep learning, such as convolutional neural networks (CNN), which can perform classification operations as well as segmentation, due to their structure that allows this through various layers and filters that help us to extract Features needed for retinal segmentation. In this thesis, we used two convolutional neural networks to segment the retina of vessels, the first convolutional network VGG-16, and the second convolutional network Resnet-34. Moreover, we have evaluated the two methods and compared them with different methods. The results obtained with both proposed models gave clearer results than the usual ones given by most of the methods commonly used in the field. Both models could be used to segment vessel-like structures in other medical applications. This paper is one of the first research projects that used VGG 16 and Resnet 34 with U-net to segment and identify blood vessels. This work is outperforming many contributions to this topic.

Keywords: retinal segmentation; convolution neuron network; U-Net; deep learning; VGG 16; Resnet 34.

Résumé

La segmentation des images et des structures médicales est une étape nécessaire à l'analyse d'images. La segmentation des images nous fournit des informations précieuses sur les distorsions et les changements anormaux, ce qui nous permet de traiter ces parasites ainsi que de les suivre. Nous nous intéressons aux vaisseaux rétiniens qui sont utilisés dans le diagnostic de maladies telles que le diabète et l'hypertension, l'évaluation ophtalmologique et divers domaines médicaux. Le processus de segmentation des vaisseaux sanguins rétiniens est une question complexe et ardue en raison de la structure physiologique. La segmentation doit donc être automatique et efficace. Diverses contributions ont été apportées à la segmentation des vaisseaux rétiniens au cours des dernières décennies. En raison du développement de l'apprentissage profond et de son utilisation dans divers domaines, notamment la vision par ordinateur, différentes méthodes ont été mises en œuvre pour segmenter les vaisseaux sanguins par apprentissage profond, comme les réseaux de neurones convolutifs (CNN), qui peuvent effectuer des opérations de classification ainsi que de segmentation, en raison de leur structure qui permet cela à travers diverses couches et filtres qui nous aident à extraire les caractéristiques nécessaires à la segmentation rétinienne. Dans cette thèse, nous avons utilisé deux réseaux de neurones convolutifs pour segmenter la rétine des vaisseaux, le premier réseau convolutif VGG-16, et le second réseau convolutif Resnet-34. De plus, nous avons évalué les deux méthodes et les avons comparées à des méthodes différentes. Les résultats obtenus avec les deux modèles proposés ont donné des résultats plus clairs que ceux habituellement donnés par la plupart des méthodes couramment utilisées dans le domaine. Les deux modèles pourraient être utilisés pour segmenter des structures de type vaisseau dans d'autres applications médicales. Cet article est l'un des premiers projets de recherche à utiliser VGG 16 et Resnet 34 avec U-net pour segmenter et identifier les vaisseaux sanguins. Ce travail surpasse de nombreuses contributions à ce sujet.

Mots Clé : segmentation rétinienne ; réseau de neurones à convolution ; U-Net ; l'apprentissage en profondeur ; VGG 16 ; Resnet 34.

الملخص

تعد تجزئة الصور ومختلف التركيبات في المجال الطبي خطوة ضرورية لتحليل الصور. يزودنا تجزئة الصور بمعلومات قيمة حول التشوهات والتغييرات غير الطبيعية، مما يسمح لنا بمعالجتها وكذلك متابعتها. نحن مهتمون بأوعية الشبكية المستخدمة في تشخيص الأمراض مثل السكري وارتفاع ضغط الدم وتقييم طب العيون والمجالات الطبية المختلفة. عملية تجزئة الأوعية الدموية في شبكية العين هي مسألة معقدة وشاقة بسبب التركيب الفسيولوجي. لذلك، يجب أن تكون التجزئة تلقائية وفعالة. تم تقديم مساهمات مختلفة لتجزئة الأوعية الشبكية في العقود الماضية. نظرًا لتطور التعلم العميق واستخدامه في مختلف المجالات، وخاصة رؤية الكمبيوتر، فقد تم تنفيذ طرق مختلفة لتقسيم الأوعية الدموية عن طريق التعلم العميق، مثل الشبكات العصبية التلافيفية(CNN) ، والتي يمكنها إجراء عمليات التصنيف وكذلك التجزئة، نظرًا لبنيتها التي تسمح بذلك من خلال طبقات ومرشحات مختلفة تساعدنا على استخراج الميزات اللازمة لتجزئة الشبكية. في هذه الرسالة، استخدمنا شبكتين عصبيتين تلافيفيتين لتقسيم شبكية العين، أول شبكة تلافيفيةVGG-16 ، والشبكة التلافيفية الثانية .Resnet-34 علاوة على ذلك، قمنا بتقييم الطريقتين ومقارنتهما بطرق مختلفة. النتائج التي تم الحصول عليها مع كلا النموذجين المقترحين أعطت نتائج أوضح من تلك المعتادة التي أعطتها معظم الطرق المستخدمة بشكل شائع في هذا المجال. يمكن استخدام كلا النموذجين لتقسيم الهياكل الشبيهة بالأوعية في التطبيقات الطبية الأخرى. هذه الورقة هي واحدة من أولى المشاريع البحثية التي استخدمت VGG 16 و Resnet 34مع U-net لتقسيم وتحديد الأوعية الدموية. هذا العمل يتفوق على العديد من المساهمات في هذا الموضوع.

الكلمات المفتاحية: تجزئة الشبكية، الشبكة العصبية التلافيفية، التعلم العميق.

Table of Contents

LIST OF TABLES

List of Figures

Abbreviations

ACC	Accuracy
AI	Artificial intelligence
ANN	Artificial Neural Network
CPU	Central processing unit
CNN	Convolutional Neural Network
DoG	Difference of Gaussians
DRIVE	Digital Retinal Images for Vessel Extraction (retinal image database)
STARE	Structured Analysis of the Retina (retinal image database)
GHz	Gigahertz
GPU	Graphical processing unit
KNN	K-nearest neighbors algorithm
PPM	Portable pixel map image format
RAM	Random access memory
RGB	Red, green and blue color space
Resnet	Residual Network
FP	False Positive
FN	False Negative
ReLu	Rectified linear unit
TF	Transfer Functions
TP	True Positive
TN	True Negative
VGG	Visual Geometry Group

Acknowledgements

Who does not thank people does not thank God, Firstly, I would like to express my great gratitude to my supervisor **Prof. Rahmoune Fayçal** for his assistance and facilitation of obstacles in front of me throughout the stages of my formation.

I am very grateful to my co-supervisor **Prof. Rahmoune Adel** who accompanied me on all the details of the publications completed throughout the research period. I am fortunate to work with both professors, whom I consider to be big brothers to me and who continue to mentor me.

I offer my special appreciation to the president of the defense committee **Prof. Ammar Mohammed** and the members of the defense committee **Dr. Messaoudi Noureddine**, **Dr. Benlatreche Mohamed Salah and Dr. Omari Tahar** for taking the time to review and discuss the thesis.

I would like to thank my parents, especially my father, who supported and motivated me throughout my journey. This work is for you alone. I would also like to thank my sisters, my wife, and my brothers for encouraging me as well.

Dedication

For my parents, my son Sohail and my children in the future, my grandparents, my grandmothers...

General Introduction

Blood vessels are responsible for transporting blood from the heart to the eye. When blood does not flow to the retina properly, problems occur and various diseases appear. We can rely on the retinal blood vessels to diagnose diseases such as diabetes, hypertension, and cardiovascular diseases. Different vascular components aid in the diagnosis process. We can also identify other symptoms such as hemorrhage that affect the eye tissues [1]. The diseases we mentioned cause retinal deformities such as aneurysms, the emergence of new abnormal vessels that can rupture consequently serious bleeding [2], and arterial occlusion [3]. In addition, stiffening, narrowing or enlargement of the retina occurs, which leads to a risk for the patient. Some of these diseases are due to genetic causes and others may have a role for the patient, such as smoking or high blood pressure. Some studies indicate that there is a relationship between changes in blood vessels and strokes. Figure 1 shows us abnormal tortuosities of the retinal veins that cause vascular occlusion. Most diabetic patients suffer from retinopathy, which affects the entire retina due to the irregular levels of the blood sugar level, which causes dilation of small retinal blood vessels; this increase in dilation affects the blood flow causing bleeding.

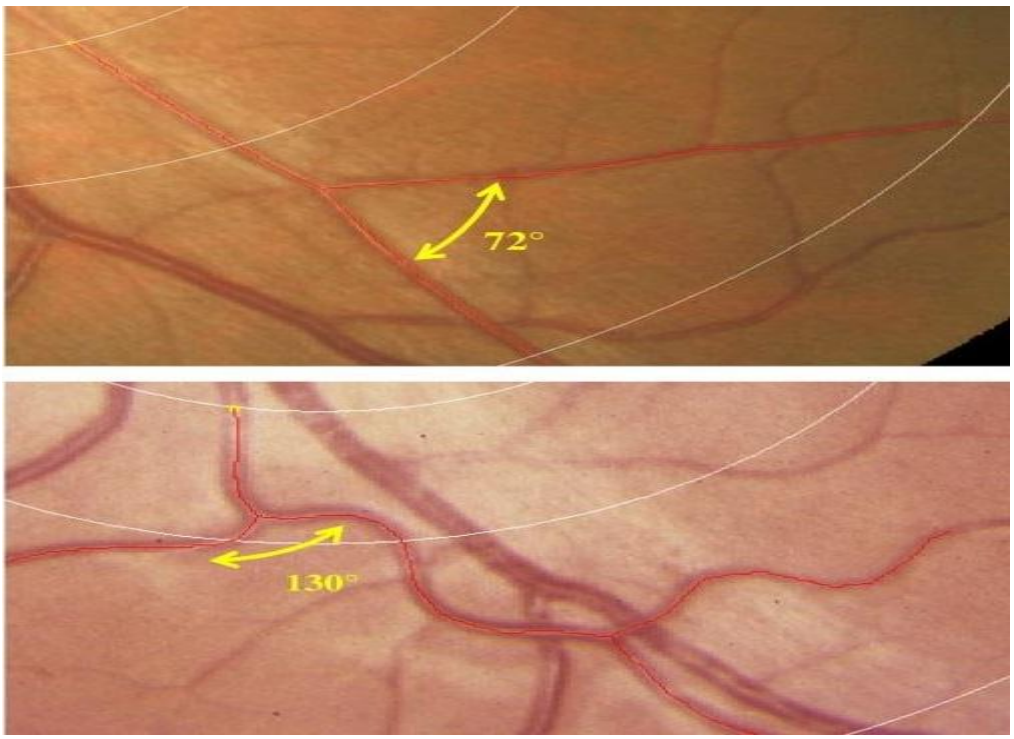


Figure 1: Example of retinal vessel tortuosity with different angles [4].

Because of the large number of people who suffer from diseases related to the vessels of the retina, the information and images are in huge quantities. Some retina trees have a complex structure so defining and segmenting them is a complex and repetitive task.

Several retinal imaging techniques have been proposed and developed. The first study was 1989 of retinal vascular segmentation. Technological development has allowed the optimization of retinal image database images, advanced image processing algorithms, and supervised automated deep learning all leading to better diagnosis.

Motivation

During the organic process inside the eye, oxygen and necessary elements are absorbed from the blood and then carbon dioxide is excreted. Some problems may occur during this organic process in the eye, causing blockage of the main blood vessels, bleeding, or the formation of new blood vessels, so we must find a way that enables us to detect these changes early to contain them.

There are still some challenges in the field of retinal vascular segmentation that confront us, as the discrepancy between vessel sizes appears, due to our primary dependence on the main vessels, narrow vessels also have an important role in the diagnostic process. Most algorithms detect major vessels in the retina, but there is still a critical need to improve the detection of delicate vessels.

The process of manual segmentation of the retina is difficult to process, tedious and errors can occur during the process, as it requires the competence of the expert to perform this process and therefore it is expensive [5]. Therefore, our need for automatic segmentation of blood vessels is essential, as it is characterized by high speed and accuracy, but the problem of low contrast in the images and lack of necessary data may make it difficult to identify and segment the main blood vessels, especially thin vessels whose pixel value is close to the background. In chapter 2, we will present various contributions and works in the field of network segmentation and then evaluate them.

Structure of the thesis

Chapter 1 contains the composition of the human eye, presenting each element separately. It also includes the type of data that was used to develop algorithms to perform the segmentation of the different retinal vessels. This is followed by the various methods that were used and the

contributions that were used for segmentation. We classified them into two categories, mentioning the methods that each category contains and the results that they get it.

Chapter 2 introduces the stages of development of deep learning.

Chapter 3 explains in detail the two methods we used to segment the retinal vessels.

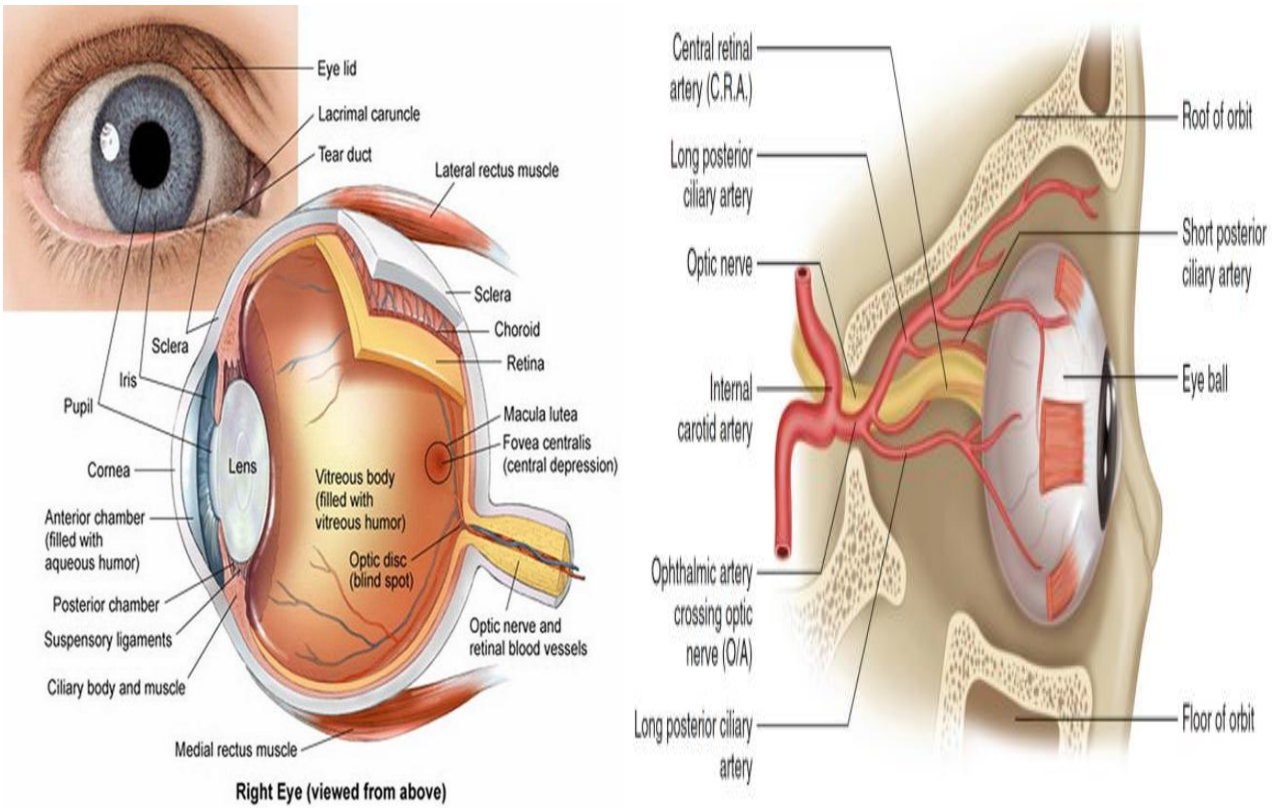
Chapter 4 presents the results obtained from the two proposed methods, as we compare our results with the results and measures used to know performance.

Chapter 1: Study of the human eye

1. Study of the human eye

The eye is the only organ in the body that enables us to see. Most people are born with eyes that give us, on average, a field of vision 200 degrees wide and 135 degrees high, and enable us to see in three dimensions and the ability to see and distinguish colors. The eye begins to work through light emanating from the outside world, as the eye allows it to pass through its components to bend and focus the light.

The eye contains light receptors, which in turn become a nerve message that is transmitted to the brain by the optic nerves. These signals are processed to embody our sense of vision [6]. Figure 1.1 shows the details of the human eye.



(a) (b)

Figure 1.1: Structure of the human eye: (a) Vascularisation of the retina, and (b) arterial supply [7].

The parts of the eye contain some elements such as:

Cornea: protects the inner part of the eye and bends light entering the eye. **Iris**: It contains the muscles that are the size of the pupil and are responsible for eye color. **Pupil:** is the dark circle inside the iris, flexible in shape, as it expands and narrows to control the amount of light entering

the eye. **Retina**: contains a thin layer of light-sensitive cells, which convert light into electrical signals [123].

The posterior ciliary arteries support the middle layers of the retina, and the inner retina is supplied by the central retinal artery.

Narrow arteries and wide veins emerge from the optic disc, which is visible from the optic nerve. It appears in the following Figure 1.2.

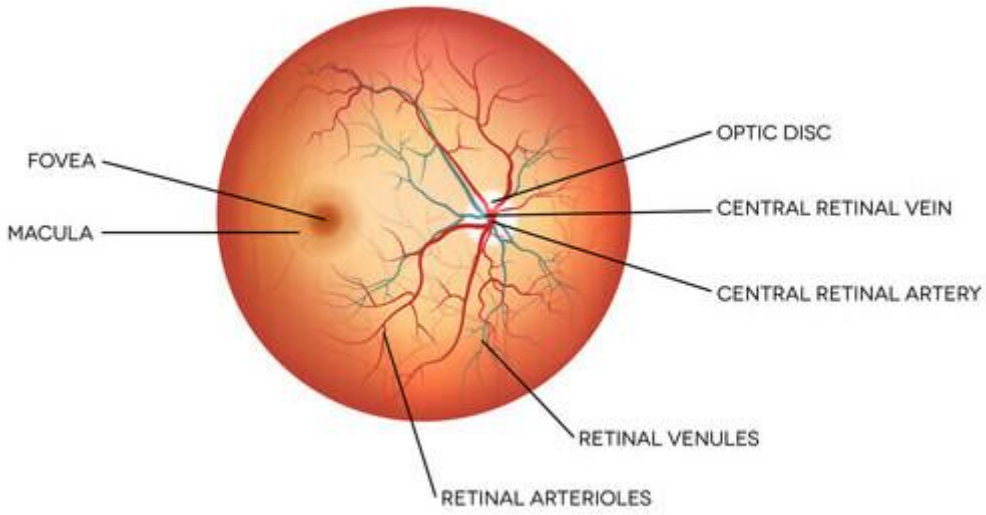


Figure 1.2: Bottom image of the retina [8].

1.2 Retinal blood vessel databases

The study of retinal vasculature depends on our available database to compare treatment methods with each other.

Most retinal vascular segmentation methods rely on general images of the retina that are colored and attached to its details in terms of size, color, and number of images. The blood vessels are visible and we can distinguish the pixel of the vessel from the pixel of the background. Based on this evidence, any researcher can perform his algorithm for retinal segmentation and measure its performance and quality.

Fortunately, for us, many databases of retinal blood vessels exist. Most of the contributions and methods of retinal vascular segmentation are evaluated on the DRIVE and STARE databases which we used in our thesis to measure the effectiveness of our contribution to retinal vascular segmentation.

1.2.1 DRIVE

Digital Retinal Images for Vessels Extraction, contains 40 color images RGB of the retina, the images were taken at random from 400 people with diabetes, 7 of these images belong to people suffering from retinopathy (images contain pathology) and the rest of the images are for healthy people. The images were taken with a Canon CR5 non-mydriatic3-CCD camera with a 45-degree field of view (FOV). These images are stored in Joint Photographic Experts Group (JPEG) format. Image dimensions are 768 x 584 pixels. The image base is divided into two groups: the training group contains 20 images with their masks and manually segmented, and the test group contains 20 images, with two sections for manual segmentation in binary (black and white), the images were segmented by experts with a high degree of certainty of the vessel and background pixel rating.

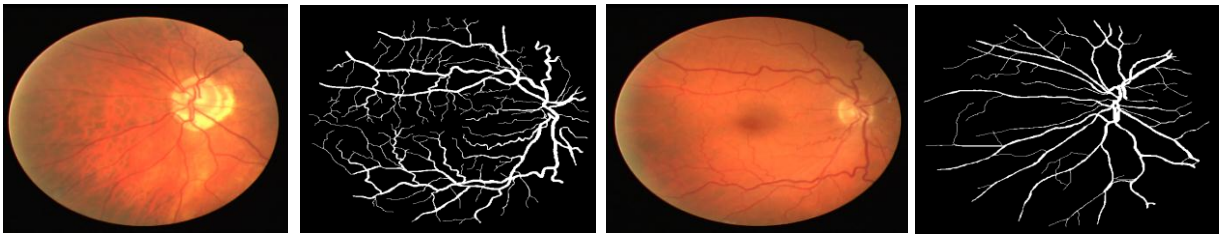


Figure 1.3: Two original retina images with Ground truth from DRIVE database.

1.2.2 STARE

Structuring Analysis of the Retina, this base contains 20 color images of the retina. The Topcon TRV-50 camera was used to capture images with a viewing angle of 35 degrees Field of View (FOV). The dimensions of each image are 700 x 600 pixels. These images are stored in portable pixmap (PPM) format. There is no set of training and test images like DRIVE. There are two sets manually segmented by two experts, the thinner blood vessels are more visible in the second set. There are only 9 images with this rule in which the retina is healthy, while the rest show the symptoms of diabetes and its effect on the vessels. STARE is among the bases with complex images.

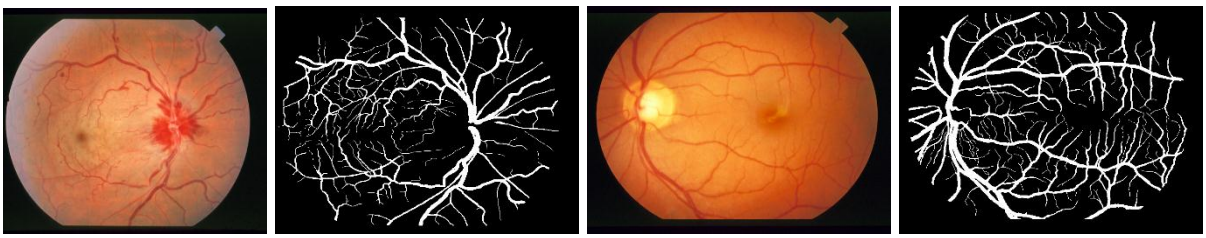


Figure 1.4: Two original retina images with Ground truth from STARE database.

1.3 Literature Review

Since 1989, several algorithms have been proposed for retinal vascular segmentation. There are techniques for vascular segmentation, which are: (i) kernel-based techniques; (ii) vessel-tracking [9]; (iii) mathematical morphology based [10]; (iv) multiscale approaches; (v) model-based approaches; (vi) adaptive local thresholding [11]; pattern recognition techniques; matched filtering and parallel/hardware-based approaches.

There are three types of learning, supervised learning, unsupervised learning, and semi-supervised learning.

Our thesis focuses on the supervised learning that is involved in deep learning.

We will mention some of the methods that enabled the identification and segmentation of retinal blood vessels.

1.3.1 Unsupervised method

Called sometimes self-taught learning, one of the most important characteristics of unsupervised methods is that they do not require a training process and are usually quick to implement. In an unsupervised method, a model-based approaches technique is usually used to extract blood vessels, various techniques are used in this method, such as Matched filter, vessel tracking, and model-based methods.

In the matching filter, a convolution filter was used to scan the image and then evaluate the response of the corresponding filter to detect the presence of vessels in the image [12, 13, 14]. Chaudhuri et al. [15] proposed a linear kernel using a Gaussian profile to identify blood vessels, the kernel is rotated by 15 degrees each time to fit vessels in different directions, after this it takes the highest values of the filters for each pixel where the threshold is applied to obtain a binary image.

Hoover et al. [16] Use the thresholding technique on a matched filter response image, the threshold is applied to each pixel iteratively and based on these values and the obtained local and region-based properties, the pixels are classified.

In [17] The width of the vessels was measured with a linear relationship with the factor of the matched Gaussian filter. Fig 1.5 provides us with the vessels that have been identified.

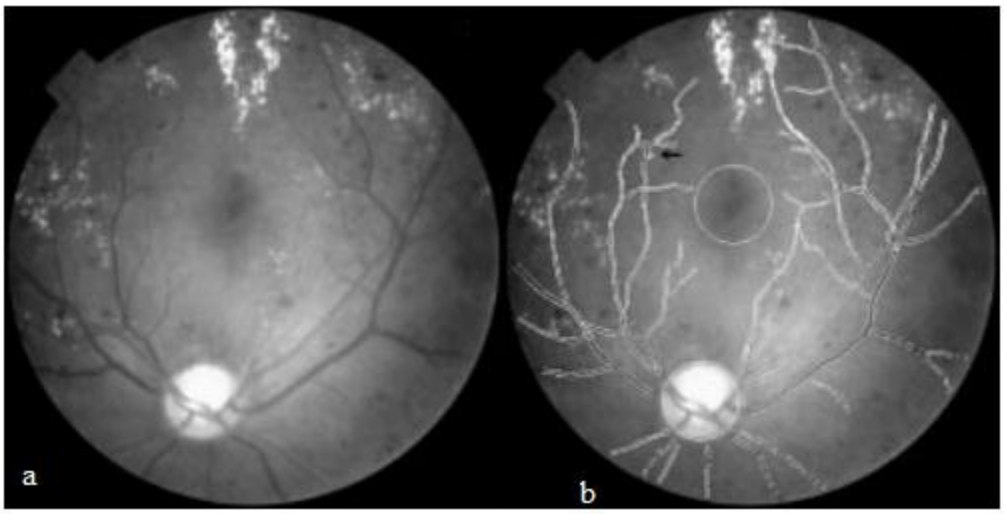


Figure 1.5: Passing a Gaussian filter to identify retinal vessels: (a) the green channel from the color image of the retina (b) detection results.

In [18] An automated algorithm was used to extract blood vessels by applying a statistically optimized LOG edge detection filter. This method was evaluated on 100 images and obtained sensitivity and specificity of 0.9879 and 0.8934, respectively.

Yao and Chen [19] applied a 2-D Gaussian matched filter for vascular improvement and then used neighborhood neurons to segment and obtain a complete vascular tree, the results for the STARE database were as follows: 0.8035 true positive rates and a 0.0280 false positive rate.

A speed method for extracting blood vessels using phase congruency was proposed, as Amin and Yan [20] worked on. First, congruency of the retinal image was performed by applying log-Gabor filters to obtain a binary pattern vascular tree after applying thresholding. Blood vessels were determined from the DRIVE and STARE databases in 10 s, the accuracy was 0.92 for DRIVE and 0.91 for STARE, respectively. Fig.1.6 test results provided to the Drive base.

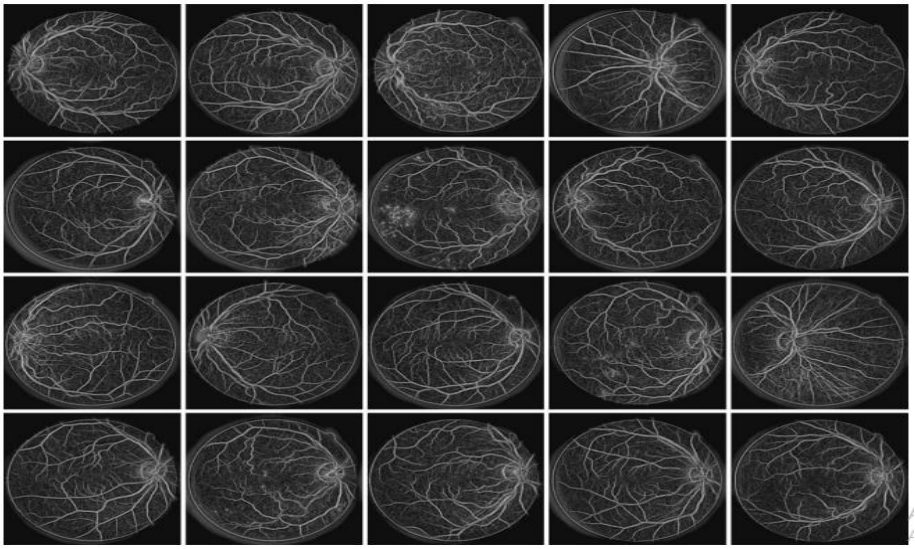


Figure 1.6: Match results on 20 test images for the DRIVE database.

Subhasis et al. [21] combined the ant colony algorithm with matching filtration and the implementation was carried out simultaneously on retinal images and the vessel network was obtained from both. For mathematical morphology, they used erosion and dilution to represent shapes like features. It is considered one of the most important approaches used for image segmentation because of its speed and resistance to noise

Jiang and Mojon [22] used an adaptive local threshold which allowed setting the threshold for each pixel instead of the overall image threshold then determined blood vessels. Luo et al. [23] estimated vessels by modifying the amplitude second-order Gaussian filter. Bob et al. [24] proposed a Matched Filter with a First-Order Derivative of Gaussian to detect vascular features. Model-based methods extract the vessels based on models such as vascular perimeter and deformation models. multi-concavity modeling method [25] and active contour model [26] are some of the methods that used this approach.

Frédéric and Jean-Claude [27] segmented blood vessels using curvature evaluation and mathematical morphology. Lucia et al. [28] used an anguine that was improved by a deformable contour model to segment retinal blood vessels. O. Chutatape et al. [29] used a derivative Gaussian matched filter to locate the center point and employed a Kalman filter for the optimal linear estimation together to detect blood vessel features. Uyen.T.Nguyen et al. [30] extracted blood vessels from color images of the retina by changing the length of the basic line detector. In the case of the Vessel tracking method, only vessel centerlines are determined by using vessel edges and local information [31-32]

Morphological processing is applied to extract image components to describe them such as boundaries, skeletons, and convex hulls. Two of the operators used in this pattern are operators are dilation and erosion

An array of morphological filters was used to segment the vessels by Zana et al. [33, 27]. The difference of the Offset Gaussian (DoOG) filter with morphological reconstruction [58-5] was used to define ocular vasculature. Firstly the central vessels are extracted by a DoOG filter, then they are optimized by using a top hat operator with variable-size circular structures to reinforce the vessels of different widths. Fig 1.7 The results of the obtained segmentations appear.

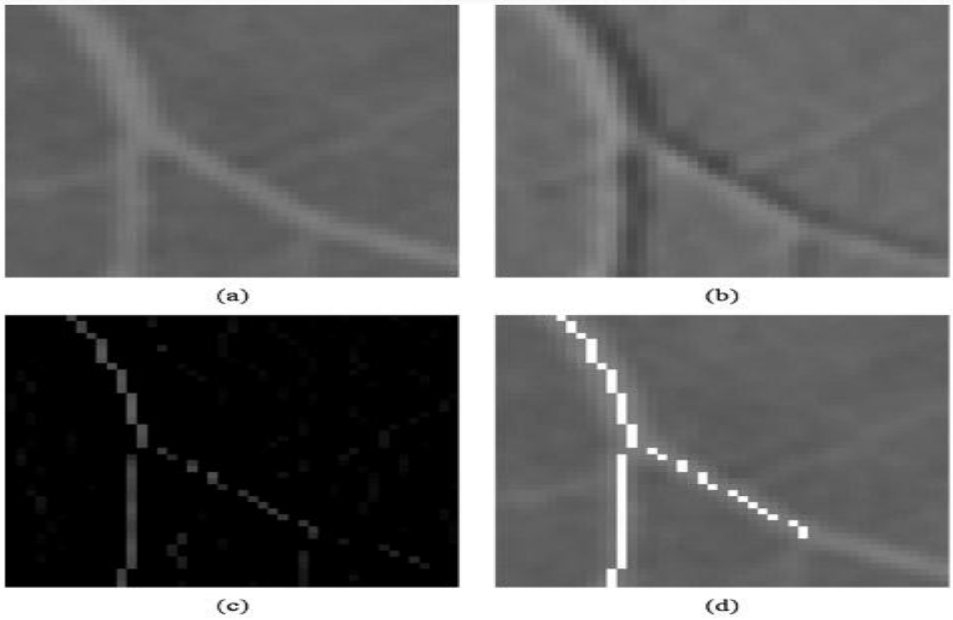


Figure 1.7: Segmentation results (a) Background containing vertical blood vessels. (b) Results of applying the DoOG filter (c) Image with midfield candidates. (d) Thresholding was applied to the image to obtain the completely segmented image.

The fuzzy clustering algorithm was combined with mathematical morphology by Yang et al. [34]. Firstly, the blood vessels were enhanced, then the background was removed by morphological top-hat operation, and finally, the vessels were extracted by fuzzy clustering. This method is applied to the STARE database. Figure 1.8 provides step details.

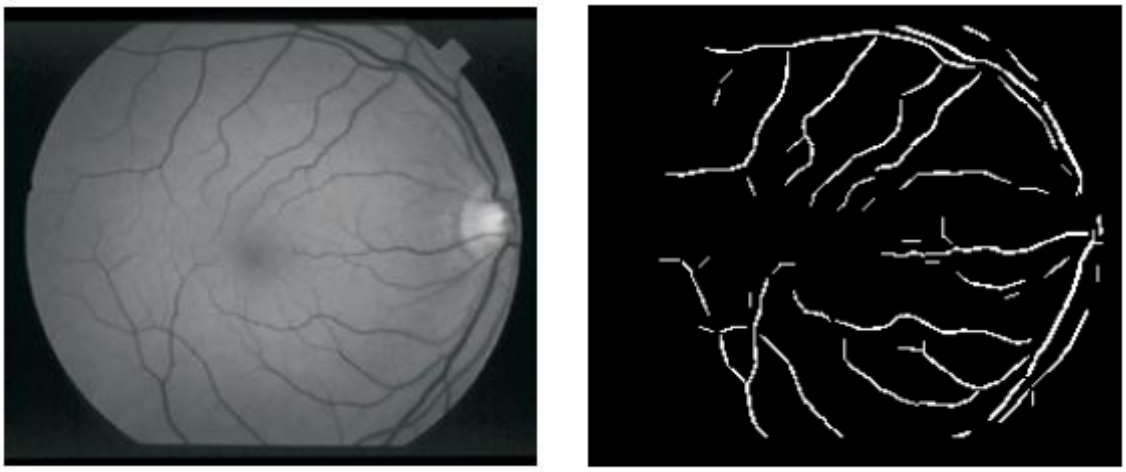


Figure 1.8: The obtained results, (a) original image, (f) segmentation result by Otsu method.

Sun et al. [35] used three algorithms for vascular segmentation, morphological multiscale, fuzzy filter, and watershed transformation. The background is extracted by non-linear multiscale morphology which is applied to each pixel and then subtracted from the image, the image is processed by a fuzzy morphological operation that rotates every 15◦ between zero and 180◦, the vessel boundaries are extracted using watershed techniques with centerline vessels. The method was evaluated on seven patients.

The blood vessels were segmented using multi-structure mathematical morphology [36] and fast discrete curvelet transform (FDCT), first the image contrast was enhanced by FDCT and then the edges of the vessels were mapped by applying a multi-structure morphological transformation, false edges were eliminated by morphological opening by reconstruction. The length of the image was scanned with density by a filter to detect vascular structures and obtain a vascular tree. The method achieved an accuracy of 0.9458 on the DRIVE database. Figure 1.9 shows the results of the segmentations obtained.

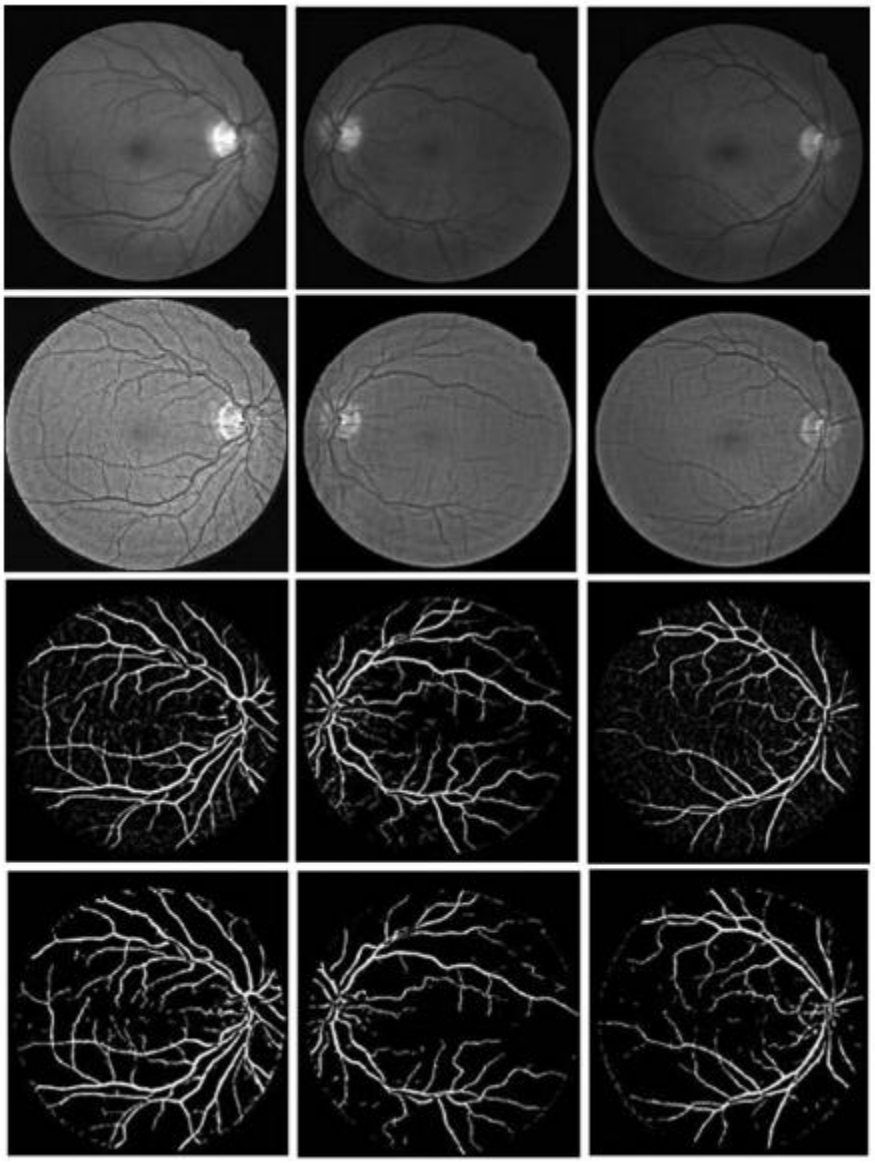


Figure 1.9: The results obtained from the DRIVE database, from the top is the green image, then the image after it has been improved by FDCT, followed by edge selection, and then in the last row it represents the segmentation results after morphology operators are applied to them.

Vessel tracking algorithms use local information to segment vessels between two points, segmenting a single vessel rather than a group of vessels. Blood vessels are identified by different characteristics such as average width, gray level intensity, and tortuosity, which is calculated during tracking. Tracking relies on central seed points featuring local information, which attempts to find its path by finding the best match while searching for the blood vessel. This method allows tracking a complete tree of vessels without wasting time with the rest of

the image content that does not contain blood vessels. One of the drawbacks of this method is its inability to track vessels that do not contain seed points, and thus the inability to discover important sub-trees.

Marios and Evangelos [37] proposed a Multi-scale algorithm that depended on line tracking to extract the vascular network and this method allows the extraction of micro-vessels but cannot identify vessels that do not contain seed points.

Adaptive tracking [38] first selects the seed critics of the blood vessels manually, then the track is traced to the blood vessel, which is then deleted by growing the deletion intensity value, this process is repeated until stability occurs. Liang et al. [39] developed the tracking algorithm by increasing the search in another direction and with a fixed and specific length, this algorithm ignores the small branches of blood vessels and also needs a manual setting to determine the starting and ending points and the tracking direction. Ali et al. [40] used an algorithm based on recursively tracking from the seed points as each time for this approach using a set of templates, each template gives maximum response to each edge, we define the edges through the points that give the highest response of the templates, the algorithm then progresses to the maximum response and the process is repeated at the new point. Branching and cross-over points can be estimated by this algorithm and it is very fast in implementation as it was used in real-time. Kelvin et al. [41] proposed a semi-automated method by integrating incorporates the multiscale vesselness filtering [42] with the conventional Livewire framework [43], first the original seed points are identified along the vessel and then contours connecting these are found points. This method was applied to the DRIVE database and reduced the execution time by 68.2%.

Vermeer et al. [44] used Laplacian for blood vessel modeling, by convolving the image with a 2-D Laplace kernel, The remaining parts of the blood vessel were determined by applying morphological closing, the value of sensitivity of 92.4% and specificity of 92.1%.

Mahadevan et al. [45] gave an algorithm for modeling the framework for the vessel through three models, first Huber's censored likelihood ratio [46], then the ranked ordered test [47] finally nonlinear least squares fitting [48]. Lam and Hong [49] proceed from the idea of divergence of vector fields to segment the retina, the centerlines are defined by a normalized gradient vector field

Despite the development of contributions regarding retinal segmentation based on unsupervised methods, it still needs further improvement, especially for images with low contrast as well as those containing micro-vessels.

1.3.2 Supervised method

On the other hand, supervised methods are simple in concept, they depend on the experts to segment the image, in the beginning, to be more efficient when compared with unsupervised methods, but they have a common obstacle, which is manual segmentation done by the expert which is slow, expensive, and sensitive, it is considered the important stage that allows the identification of vessels. In supervised methods, pixels are categorized as if they belong to the vessels or the background. Feature vectors are often used in this approach to identify blood vessels.

There are examples of supervised algorithms, such as Orlando et al. [50] where they segmented the vessels with the least energy in a conditional random field (CRF). In [51] retinal segmentation was done by cross-modality learning where the color image was the first modality and the second modality was the corresponding vessel map. M. Al-Rawi and H. Karajeh [52] detected blood vessels by using a Genetic algorithm (GA) and found the optimal parameters for filtering (MF) response. Kundu and Chatterjee [53] used Morphological angular scale space for retinal vessel segmentation. In [54] global thresholding which depends on morphological operations was used. In [55] Fuzzy C-Means (FCM) clustering method was used for separating the pixels while using Artificial Bee-colony (ABC) for the optimization of the clustering. Finally, the images were classified using a support vector machine (SVM). In [56] the ideal selection of features was the most important stage for the success of the classification because they selected six features that were also related to the ant algorithm which reduced the time and complexity during the segmentation process. In [57] a technique that uses a flower pollination search algorithm (FPSA) was proposed along with pattern search (PS), the first algorithm found all the vessels and the second one found the vessels that the first algorithm could not identify. In [58] they used a kNN-classifier to Classify vascular features. The work of [59] used a Gabor wavelet transform with grayscale intensity in a unified model. The work of [60] classified pixels using neural network and feature vector computation.

Traditional machine learning, which has a significant impact at present, which involves supervised learning that relies on a training data set that helps us make the right decision. Use

Niemeijer et al. [61] k-Nearest Neighbors (kNN) technique for segmentation of blood vessels, knowing that the value of k = 30 and using the pixel intensity in the green channel. Staal et al. [62] extracted the blood vessels by sequential forward selection and a kNN classifier where the intensity extremum was used to determine the line primitives that make up the vessels. Soares et al. [63] used green channel intensity with a 2D Gabor wavelet and taking into account a few features, the wavelet is used for its ability to identify the appropriate details that represent to us the vessels.

Roychowdhuri et al. [64] detected the vessels through the result of the intersection of segmentations that occurred in two different ways, the first is to pass a high filter on the image of the retina and the second is to use top-hat reconstruction, the intersection between them provides us with an initial picture of the shape of the blood vessels. In [65] a GMM classifier was used to distinguish micro-vessels from the background. Strisciuglio et al. [66] used 42 filters (21 filters for the end of the blood vessel, and 21 special filters for the segmentation of the blood vessel). The sum of the filters is applied to the green-channel intensity value, and then the SVM classifier is used to classify the pixels if it is a blood vessel or belongs to the background. Ricci et al. [67] proposed a way that depends on line operators and Support Vector Machine to do the segmentation, first, the feature vectors are created by line operators, using a line detector (which depends on the average gray level that passes over the pixel lines) which is applied to the grayscale image, the contrast is high between the background pixels and the pixels of the blood vessel. The output of the pixel classification allows for segmentation. This model is characterized by the speed in execution.

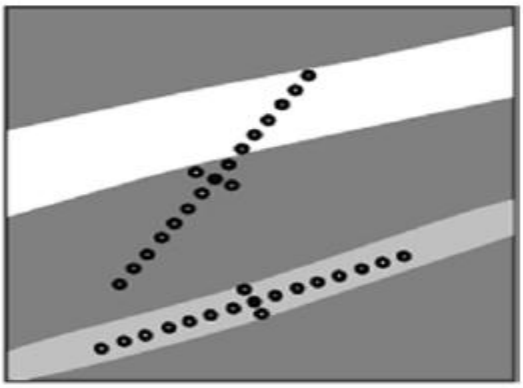


Figure 1.10: An example showing the working principle of a line detector by Ricci *et al.* [67].

Marin et al. [68] gave a method based on a neural network, pixels are classified by a neural network and a vector is computed with a set of features that are used to represent pixels, rely

on MLP to do training and classification, inputs consist of 7 neurons followed by 3 hidden layers (each layer it has 15 neurons) while the output layer consists of one neuron, the non-linear sigmoid activation function was used in all neurons. This method has proven the effectiveness and power of neural networks.

In the past 15 years, with the development of hardware and the evolution of convolutional networks, most researchers have turned to deep learning to address the various challenges they face.

Competitive results were obtained after applying deep learning for semantic segmentation. Many achievements have been based on deep learning for retinal blood vessel segmentation. The form of data representation differs in algorithms based on deep learning that segments the retina when compared to traditional machine learning algorithms, where we must think carefully before designing features and the classification, which makes it tedious and takes longer, in addition to its difficulty for non-experts, on the other hand, CNN takes the place of human beings in extracting features. It has been proven that the structure of deep learning has strong efficiency in learning-rich hierarchical representations. the convolutional neural network extracts features through convolutional operation and the Pooling layers, as it is not affected by changing edges and angles to extract features, which is an important and useful point as well, as these operations are employed in a set of sequential layers to determine the features that we use for classification and segmentation.

Melinšcak et al. [69] Melinscak et al. [70] presented a model for the segmentation of retinal blood vessels based on CNN. The model consists of 10 layers as shown in Table 3.4, where the first layer represents the input layer, after which a set of convolutional layers is applied to extract maps of features as it follows Max-Pooling to reduce output from each layer. The last layer, in short, contains two neurons to perform binary classification (vessel or non-vessel).

The model was trained and tested on the DRIVE database. The results are presented in Figure 1.11.

Table 1.1: architecture of the implemented convolutional neural network by Melinscak et al [70]

Layer	Type	Maps and neurons	Kernel size	Stride	Padding
0	Input	1 Map of 65 x 65 neurons	-	-	-
1	Convolutional	48 Maps of 60x60 neurons	6x6	1	0
2	Maxpooling	48 Maps of 30x30 neurons	2x2	1	0
3	Convolutional	48 Maps of 26x26 neurons	5x5	2	2
4	Maxpooling	48 Maps of 13x13 neurons	2x2	1	0
5	Convolutional	48 Maps of 10x10 neurons	4x4	2	0
6	Maxpooling	48 Maps of 5x5 neurons	2x2	1	0
7	Convolutional	48 Maps of 4x4 neurons	2x2	2	0
8	Maxpooling	48 Maps of 2x2 neurons	2x2	1	0
9	Fully connected	100 neurons	1x1	-	-
10	Fully connected	2neurons	1x1	-	-

Figure 1.11: Results of retinal vasculature identification obtained by Melinscak et al. [70].

Fu et al. [71] gave a segmentation method similar to that proposed by Melinscak et al. [70]. Fu introduced an image-to-image training method that produces a multi-scale and multi-level response. Use a convolutional network for feature extraction and keep using Conditional Random Field (CRF) to produce binary vessel segmentation results. Figure 1.12 illustrates the presented approach.

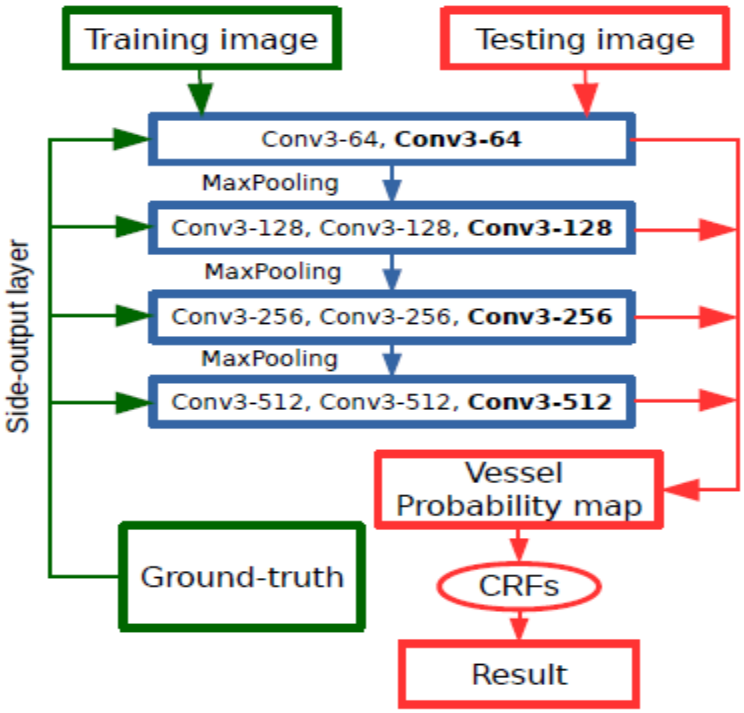


Figure 1.12: The general outline of the method proposed by Fu et al [49].

If we compare previous methods such as Melinscak et al. [70] which is based on the principle of pixel-to-pixel classification, the detection of blood vessels was dealt with by contour detection, where a model of Fully Convolutional Neural Network was built by holistically-nested edge detection (HED) [72]. HED allows learning of hierarchical representations to address the problem we face in detecting edges and objects, figure 1.13 [72] provides an example of HED.

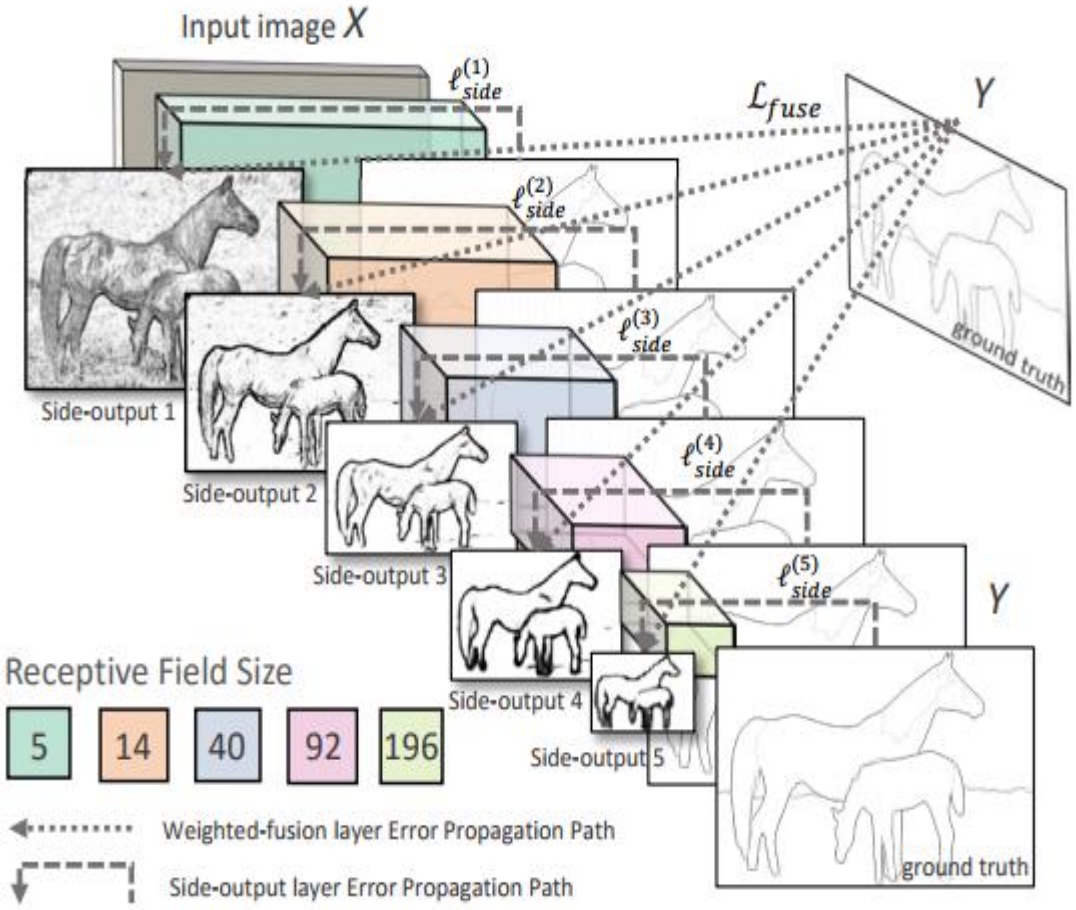


Figure 1.13: The general structure of the network responsible for edge identification using HED [72].

Wang et al. [73] The Convolutional Neural Network (CNN) was used to segment the vasculature as it was combined with the ensemble of Random Forests to classify the output of the features from the convolutional model. They used 6 layers as shown in figure 1.14. Table 1.2 also contains details.

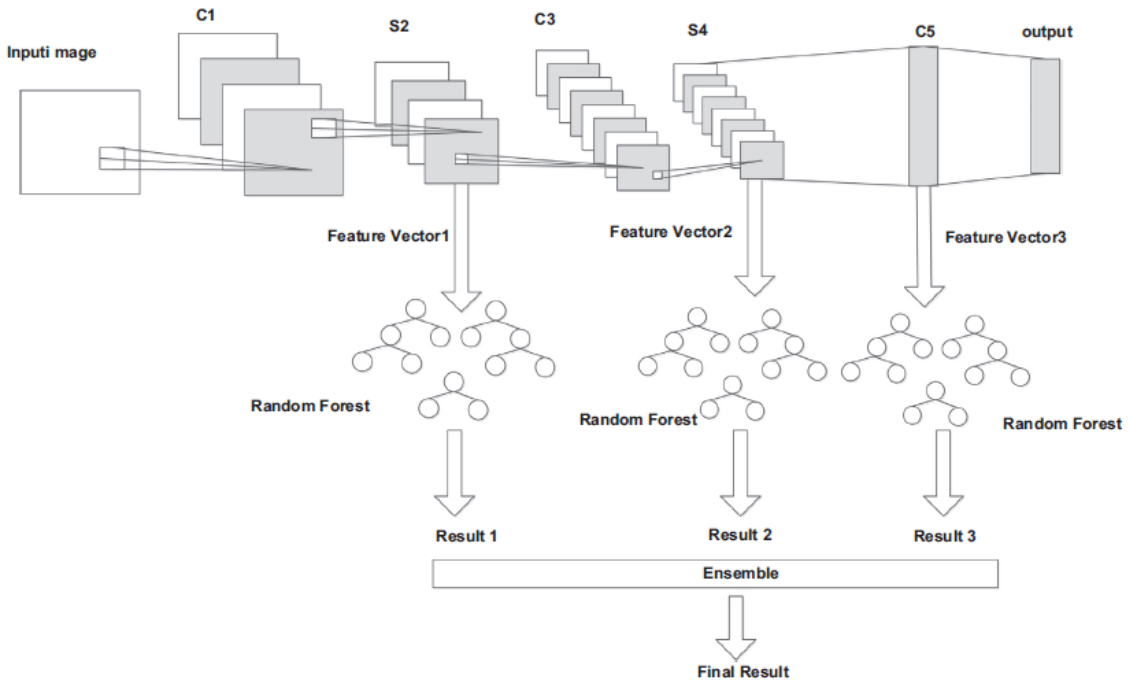


Figure 1.14: The general outline of the method proposed by Wang et al. [73].

Table 1.2: Architecture of the implemented Convolutional Neural Network by Wang et al. [73].

Layer	Type	Maps and Neurons	Kernel Size	Stride	Padding	Reference
0	Input	1 Map of 25 x 25 neurons				
1	Convolutional	12 Maps of 22 x 22 neurons	4x4	1	1	C1
2	Subsampling	12 Maps of 11 x 11 neurons	2x2	2	0	S2
3	Convolutional	12 Maps of 8 x 8 neurons	4x4	1	1	C3
4	Subsampling	12 Maps of 4 x 4 neurons	2x2	2	0	S4
5	Convolutional	100 Neurons	4x4	1	1	C5
6	Output	1 neurons				

Figure 1.15 presents the accuracy obtained.

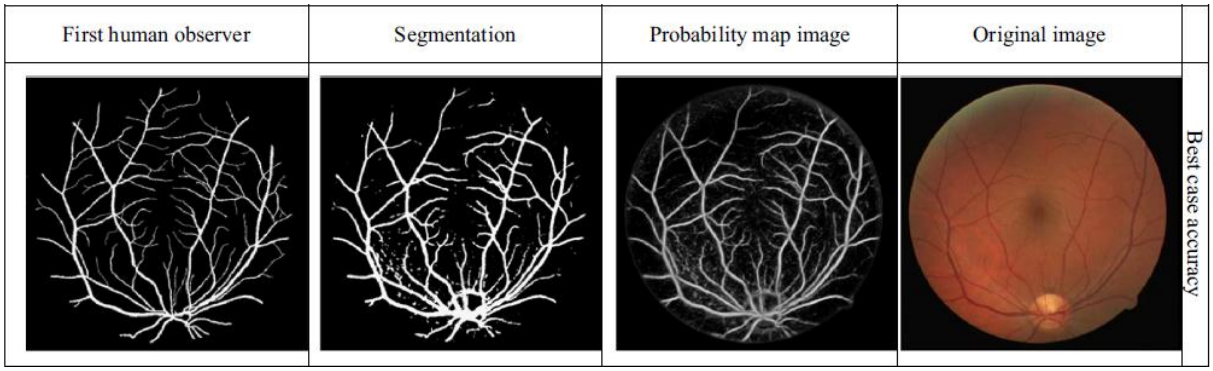
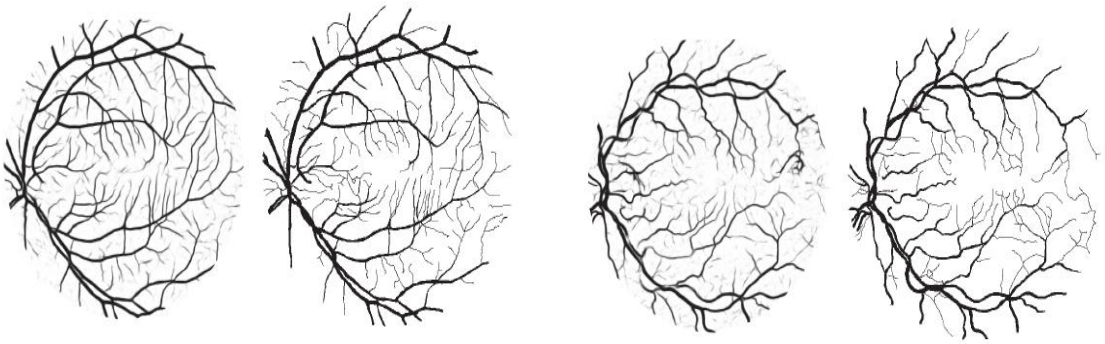


Figure 1.15: The optimal results obtained by Wang et al. [73] applied to the DRIVE database.

Liskowski and Krawiec [74] presented one of the most important methods for retinal segmentation by CNN, pre-processing of samples was performed before the network training stage, relying on global contrast normalization, and zero phase whitening, in addition to increasing the data by doing geometric transformations and gamma correction. Experiments have shown that pre-processing and data augmentation allows for an increase in performance. 400 000 patches have been created from the DRIVE database and 380 000 from STARE. Table 3.5 displays the details, Figure 1.16 shows the best segmentations for the DRIVE (a) and STARE (b) databases.



(a) (b)

Figure 1.16: Prediction results (left) and Ground truth (right): (a) DRIVE and (b) STARE [74].

Table 1.3: Two most relevant architecture proposed by Liskowski and Krawiec [74]

Implementation	Layer	Maps	Kernel size	Stride	Padding
Plain	Convolutional	64	4x4	1	0
	Convolutional	64	3x3	1	1
	Maxpooling	-	2x2	2	0
	Convolutional	128	3x3	1	1
	Convolutional	128	3x3	1	1
	Maxpooling	-	2x2	2	0
	Fully connected	512	-	-	-
	Fully connected	512	-	-	-
	Fully connected	2	-	-	-
No-Pool	Convolutional	64	3x3	1	1
	Convolutional	64	3x3	1	1
	Convolutional	128	3x3	1	1
	Convolutional	128	3x3	1	1
	Fully connected	512	-	-	-
	Fully connected	512	-	-	-
	Fully connected	2	-	-	-

Table 3.5 presents the difference in spatial pooling, where the results showed that the models that do not contain Max-pooling are better in terms of performance compared to those that have Max-pooling.

Li et al. [75] changed the segmentation of blood vessels into a cross-modality data transformation, the images were divided into 16×16 patches and they used a 5-layer neural network where each pixel is predicted from the patch.

Huazhu [76] formulated the vessel segmentation into a boundary detection problem; They used fully convolutional neural networks (CNNs) to create a vessel probability map. Moreover, fully-connected Conditional Random Fields (CRFs) had been employed to combine the characteristic vessel probability map and the big differences between the pixels. Finally, the binary vessel segmentation accuracy and sensitivity reached were 94.72% and 77.61% respectively. Ming et al. [77] combined a dense network with U-net to predict vessels and got an accuracy of 96.74%, a sensitivity of 81.50, and a specificity of 98.20%. Gao et al. [78] combined Gaussian matched filter with U-net. Their model was able to distinguish the vessels from the background despite the problems related to the pathological regions. Liang et al. [79] merged dense network and

expansion convolution into the U-shaped network where they treated the problem of ignored important information during the pooling layer process. They also solved the weakness of incorrect microvascular, but still, they were unable to solve a small amount of microvascular at the end of the optic disc and not adapting scale information as well. Jin et al. [80] Use U-Net architecture with convolutional deformable kernels.

[81] proposed two U-Net architectures with pre-processing done by Gabor Filtering, Gaussian Blur, and Sobel, applied to the DRIVE database, and the precision value was 0.9671 and 0.9659 for both proposed architectures. [82] They made a comparison on the DRIVE database using different U-net models (U-Net, Attention U-Net, and Nested UNet) and loss functions (Binary Cross Entropy, Dice, Tversky, and Combo).

Changlu [83] proposed a method inspired by U-Net through feature map reuse called DRNet, which was highly efficient in learning as well as stable. Changlu [84] They came back again with a new way to segment the retina, they used Spatial Attention U-Net (SA-UNet) as it does not require much training data.

Juntang Zhuang [85] proposed the LadderNet model, which belongs to the U-net family, but this model is characterized by having multiple pairs of encoder-decoder, as shown in the f figure 1.17.

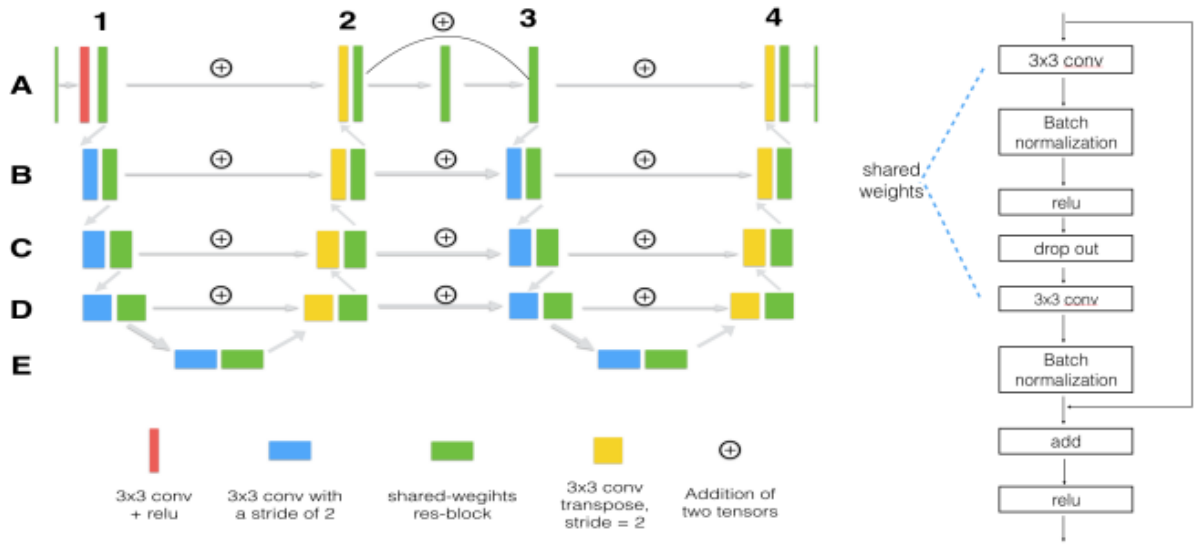


Figure 1.17: Left: Backbone of LadderNet and right residual block.

The accuracy of the model in DRIVE and CHASE was 0.9561 and 0.9656, respectively. [86] Md Zahangir Alom proposes a Recurrent Convolutional Neural Network (RCNN) This

technique is based on U-Net in addition to Recurrent Residual Convolutional Neural Network (RRCNN), this method allows training deep networks and gives better results than the original U-Net, It also ensures better representation of features through recurrent residual convolutional layers.

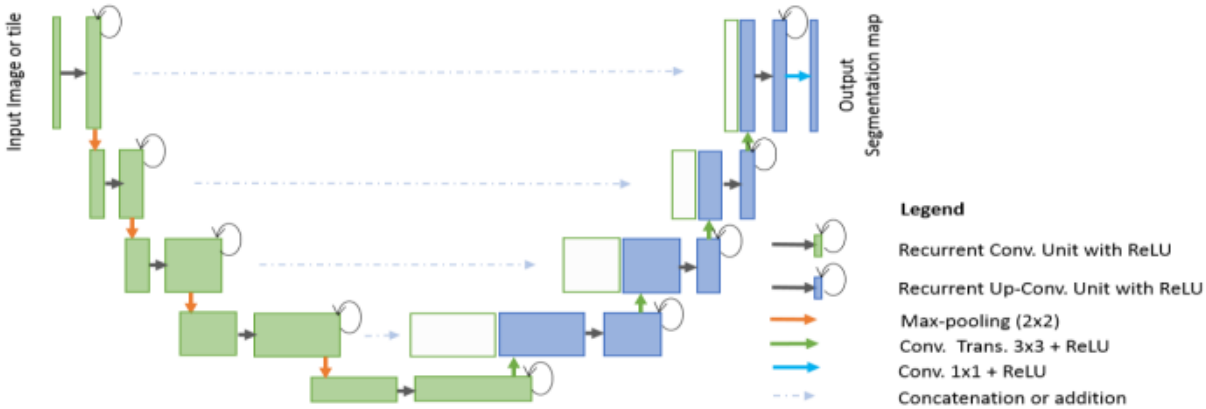


Figure 1.18: RU-Net architecture using recurrent convolutional layers (RCL).

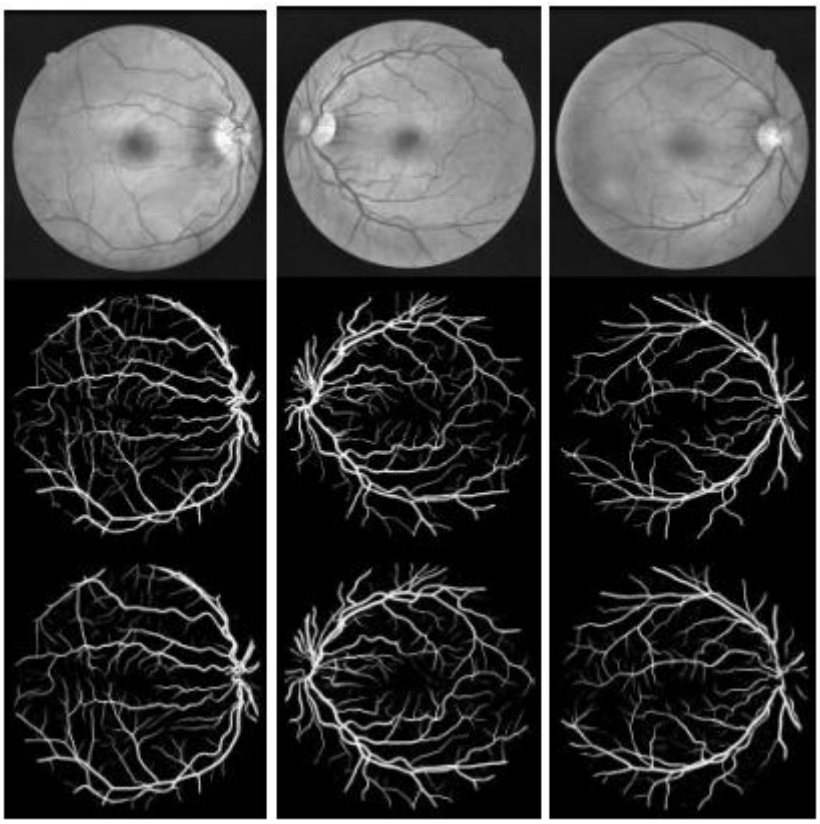


Figure 1.19: Segmentation results on DRIVE: first row represents grayscale images, second row represents ground truth, last row represents prediction results by R2UNet.

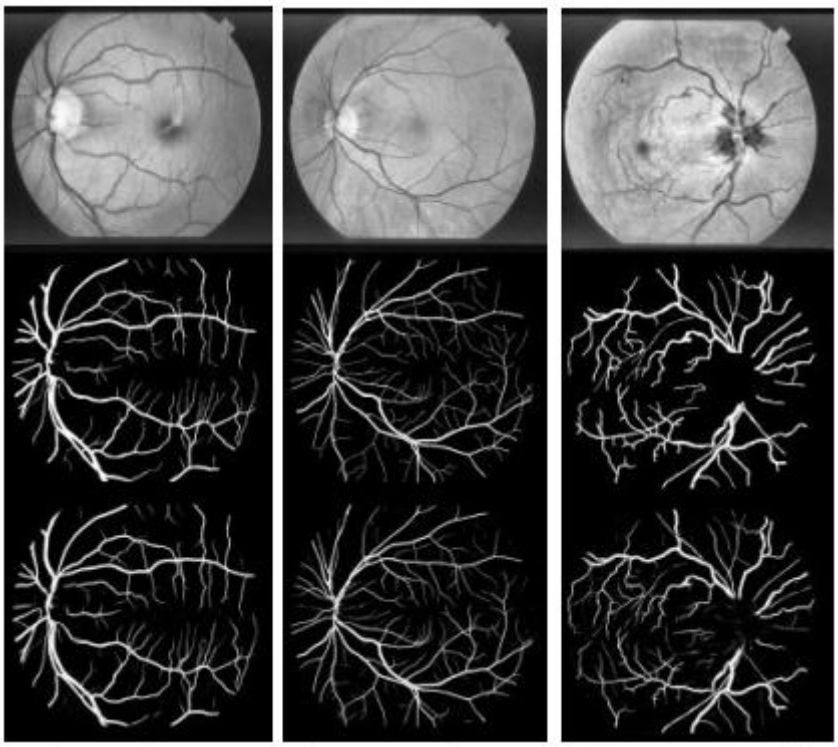


Figure 1.20: Segmentation results on STARE: first row represents grayscale images, second row represents ground truth, last row represents prediction results by R2UNet.

The Accuracy value is 0.9565 and 0.9706 for DRIVE and STARE respectively. Zengqiang Yan [87] suggested new segment-level loss Get used to measuring the loss value by segments, not just by pixels. The accuracy value was 0.9542 and 0.9612 for DRIVE and STARE, respectively. Figure 1.21 shows an overview of the method.

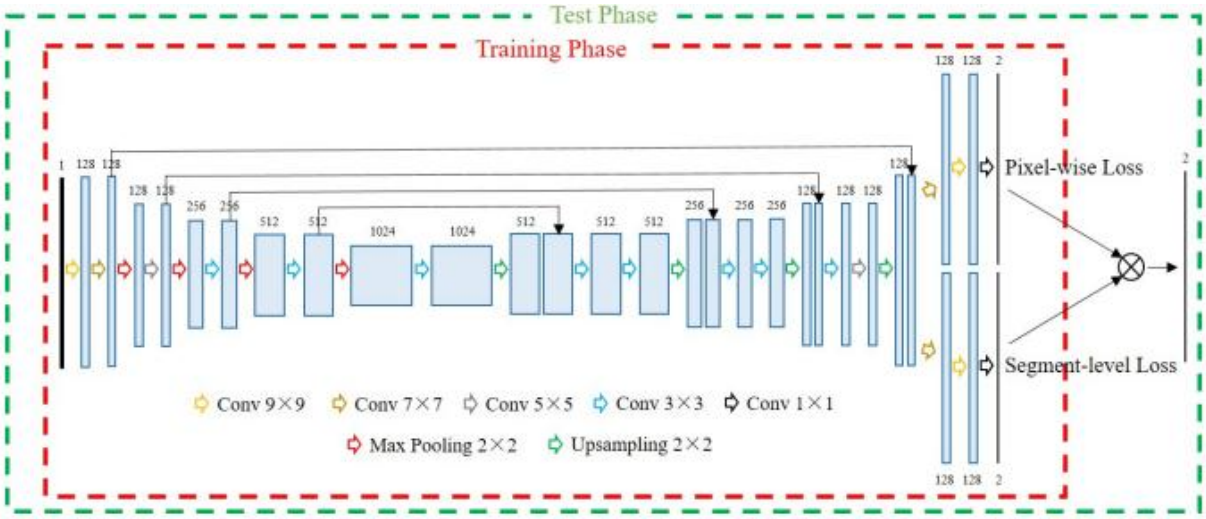


Figure 1.21: The overview of the proposed joint-loss deep learning framework.

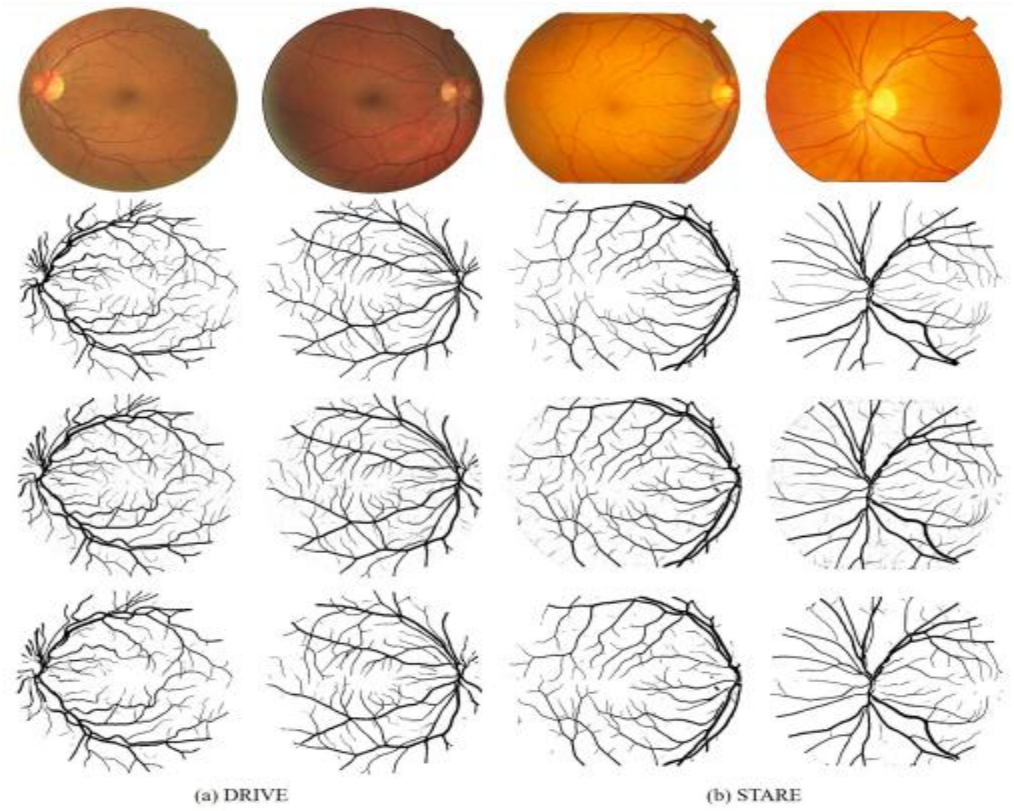


Figure 1.22: Segmentation results on the datasets DRIVE, STARE. First row: the original images, then Hand notes, the probability maps and the corresponding hard segmentation maps.

Ozan Oktay [88] presented a new attention gate (AG) procedure added to the U-net structure, this procedure focuses on certain regions in order to train them, enabling AGs to reduce computations. The following model presents the stages of adding an AG into the U-net structure during the segmentation process.

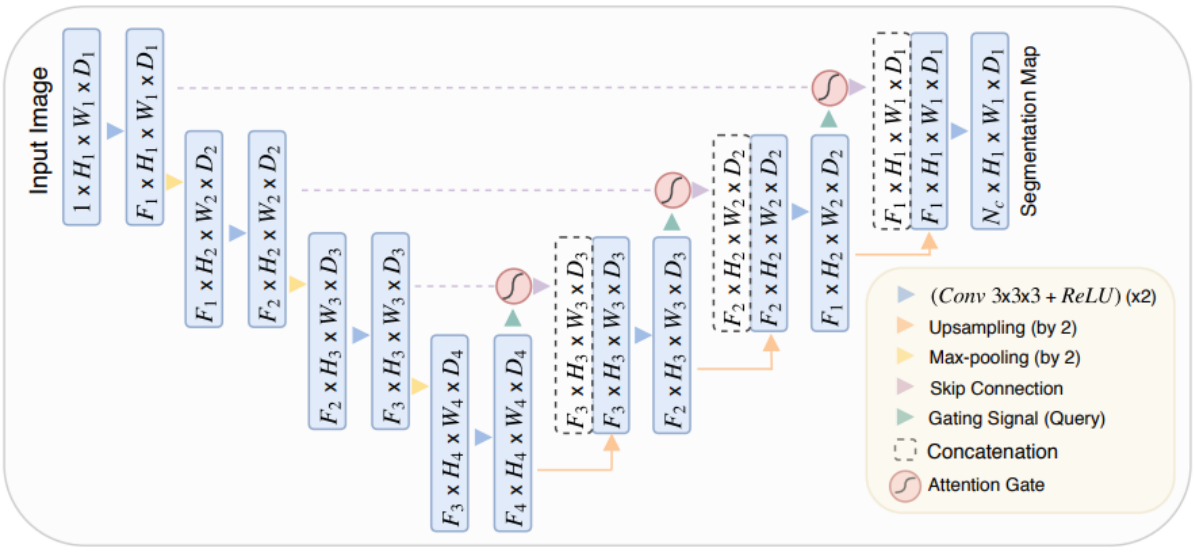


Figure 1.22: Attention U-Net Model Structure.

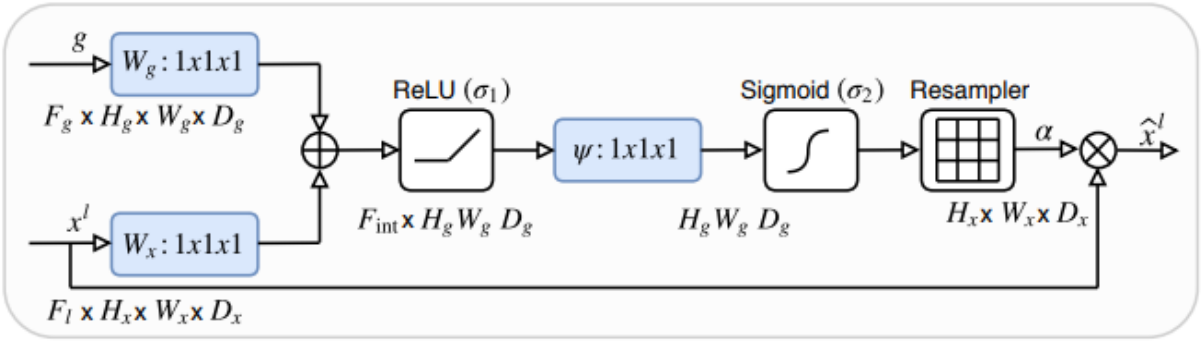


Figure 1.23: Attention gate (AG) outline.

Conclusion

In this chapter, we mentioned the importance of blood vessels in the retina and that it can be relied upon to diagnose diseases, as segmented images help us better to provide clinical information. Supervised methods and unsupervised methods, where we mentioned examples of each method and the results obtained. In the next chapter, we will present the details and methods of deep learning, where we will learn about the stages of development of deep learning, the most important advantages that it has, which prompted us to use it in our thesis.

Chapter 2: Deep Learning method

2. Deep Learning method

Yann LeCu defines deep learning as a learning data representation method with multiple levels of representation. These representation levels are obtained by composing simple but nonlinear levels that transform each level into a higher level of representation. Deep learning layers are not designed by humans, but are obtained through learning data and the general procedures associated with learning. [115]

Our model involves supervised methods based on deep learning and trains our model using convolutional networks. We will detail the steps in the following chapter.

2.1 Convolutional neural networks evolution

The targeted idea through the use of AI (artificial intelligence) is to create a set of programs and algorithms that enable us to find solutions to the problems that we face in life, but our perception of objects and their representation in our imaginations differs, of course, from the representations that the computer has, which we humans find difficult to present to the computer before.

Artificial neural networks Inspired by the formation of the brain and the principle of its work, this pattern allows one to perform a set of functions, the most important of which is pattern recognition. Deep learning we can say that it is derived from ANN algorithms

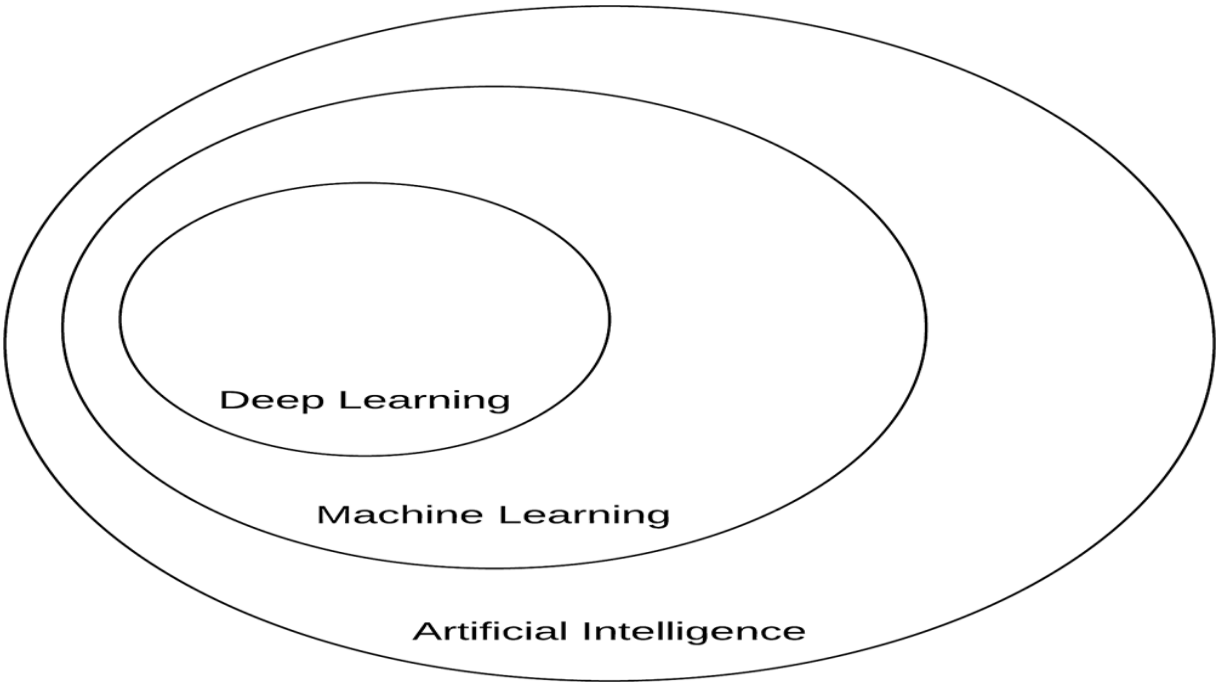


Figure 2.1: A diagram showing deep learning as a branch of machine learning, which in turn is part of artificial intelligence.

Deep learning was not new, it has existed since the 1940s, but under different names such as Artificial Neural Networks (ANNs), and Cybernetics.

McCulloch and Pitts in 1943 [89] presented the first network that can perform binary classification based on inputs that the user had to enter with the weights manually, which we can consider a defect in this network.

2.1.1 Perceptrons

In the fifties of the last century, the algorithm on which the perceptron depends was created by Rosenblatt [90-91]. One of its advantages is its ability to automatically learn weights to perform the classification process without human intervention. Figure 2.2 presents the Perceptron architecture. This procedure is considered the cornerstone of machine learning.

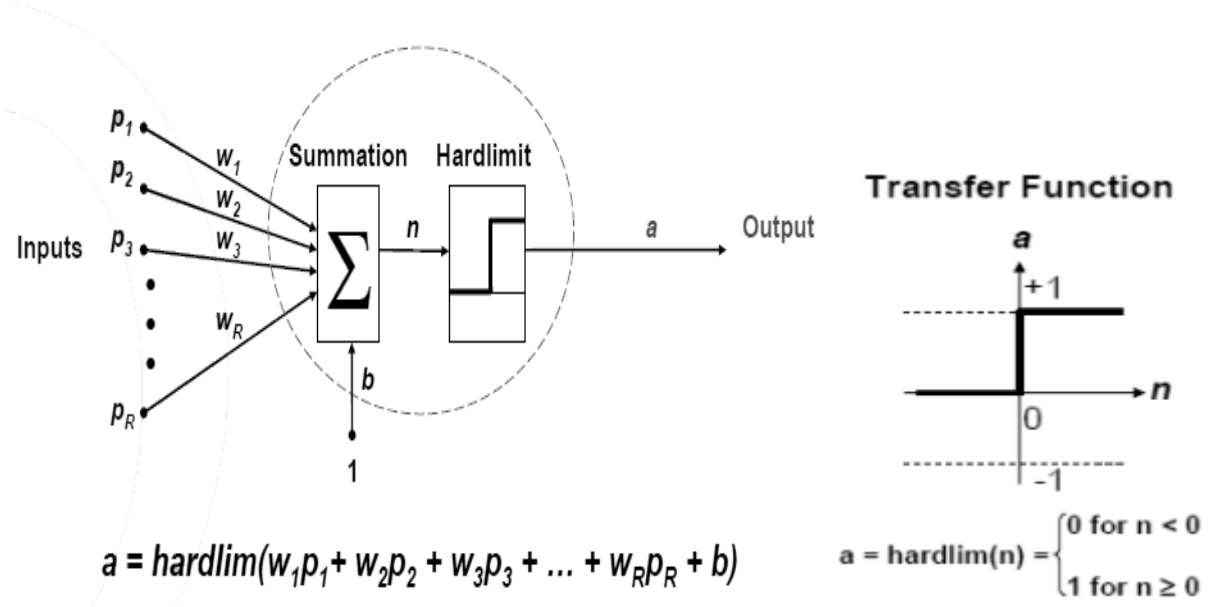


Figure 2.2: On the left: General architecture of the simple Perceptron network with number of inputs, computes a weighted sum, and applies a step function of hardlimit to obtain the final prediction. On the right: formula of hardlimit transfer function.

In MATLAB for example representation of the perceptron neuron is:

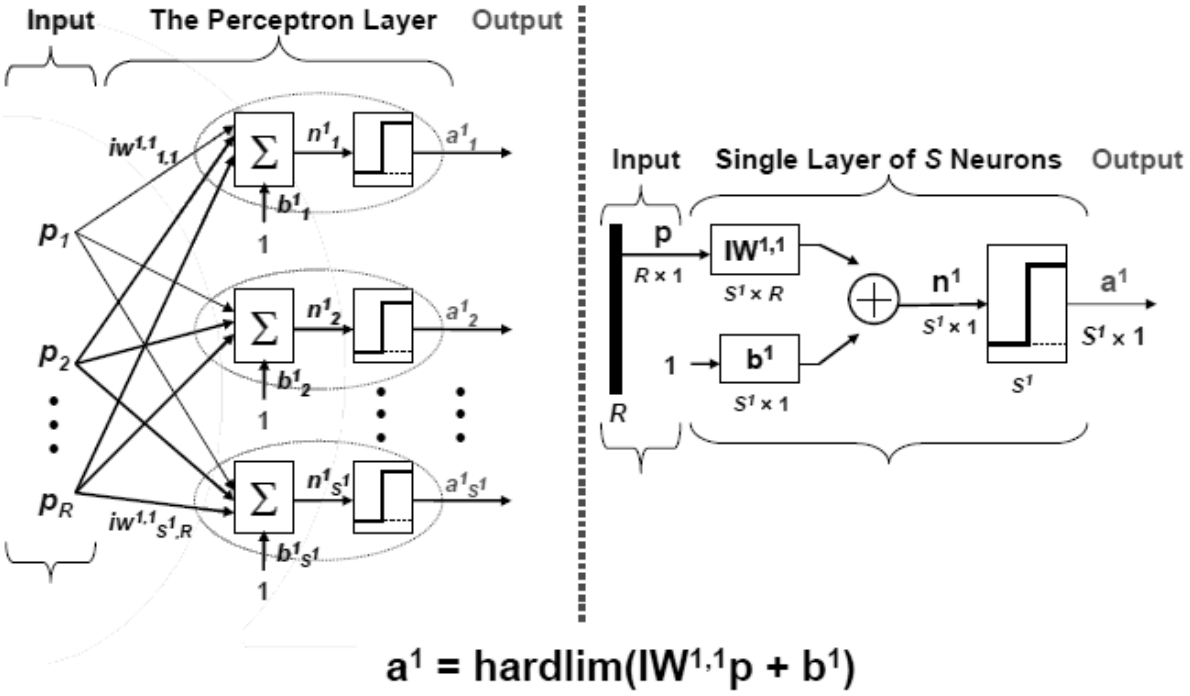


Figure2.3: Perceptron architecture en detail parameters

The Perceptron Learning

Perceptrons are trained by the following examples: $\{^{p}_{1}, ^{t}_{1}\}$, $\{^{p}_{2}, ^{t}_{2}\}$,…, $\{^{p}_{Q}, ^{t}_{Q}\}$.

The error e, which is the difference between perceptron output a and the target vector t is minimized, e =t - a, the weights (W) and biases (b) are adjusted each time until we get the optimal values according to the following equations:

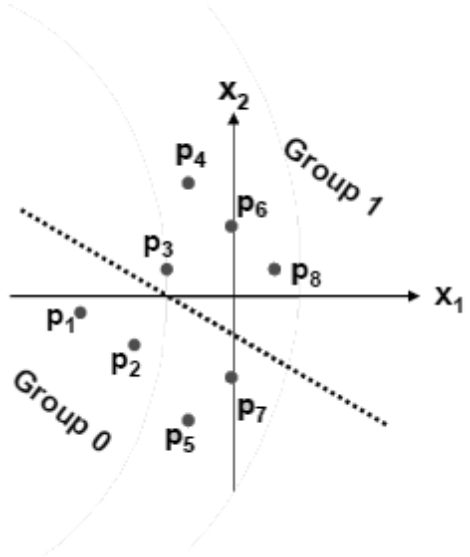
$$W_{new} = W_{old} + \Delta W = W_{old} + ep^{T} \qquad (2.1)$$

$$b_{new} = b_{old} + \Delta b = b_{old} + e \qquad (2.2)$$

Where $e = t - a$ (2.3)

Weights and biases are finally obtained that solve the problem according to the error, by adjusting and iterating the previous equations.

The following figure contains groups of data where we can create a Perceptron network and train it to perform the classification process:



Data	x_1	x_2	Group
p_1	-3	-0.5	0
p_2	-2	-1.2	0
p_3	-1.5	0.7	1
p_4	-1	3	1
p_5	-1	-3.5	0
p_6	0	2	1
p_7	0	-2.5	0
p_8	1	0.7	1

Figure 2.4: Example of input data for the Perceptron

% Let's upload the data

% Construct a perceptron

% Train the perceptron with (p) and (t)

% Simulate the perceptron with our data

a =

0 0 1 1 0 1 0 1 % Classified successfully

t =

0 0 1 1 0 1 0 1

% Let's give new points t1 and t2 to the perceptron network and note how to classify and verify it.

a_t1 = 0

a_t2 = 1

So the perceptron classifies t1 and t2 correctly.

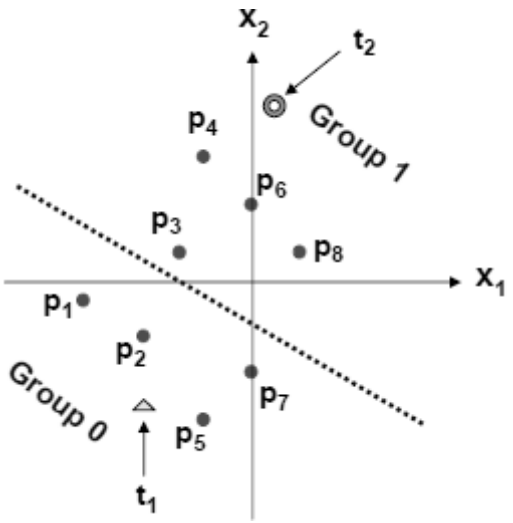


Figure 2.5: An example of a valid classification of points entered into a perceptron

The Perceptron can only do linear classification, as it cannot solve nonlinear problems.

Figure 2.6 presents the most popular nonlinear XOR dataset; Perceptrons cannot solve this problem or any linear network.

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

p_1	p_2	a
0	0	0
0	1	1
1	0	1
1	1	0

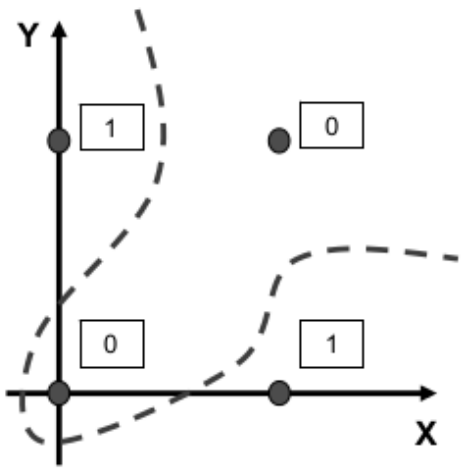
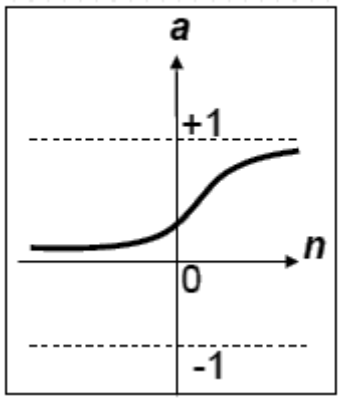


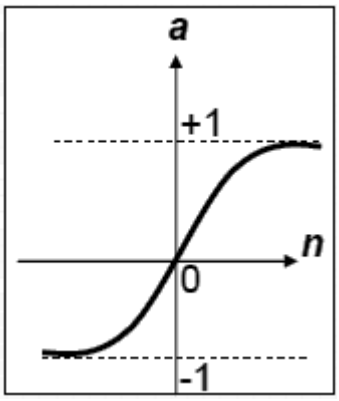
Figure 2.6: XOR is an example of a problem that the Perceptron cannot solve.

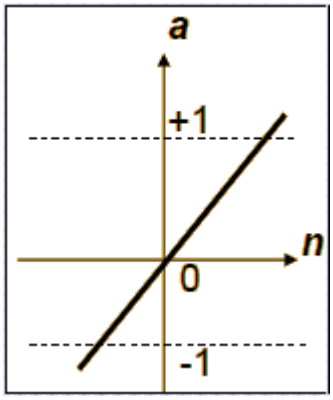
2.1.2 Multilayer Layer Perceptron (MLP)

The problem of nonlinear classification remained open until Werbos (1974) [92], Rumelhart (1986) [93] and LeCun (1998) [94] strengthened the neural networks, where they presented a training model of multi-layered feedback as shown in Figure 2.6, it can converge towards any function through a sigmoid TF layer, and a linear TF output layer.

Transfer Functions







Logsig (n) = $\frac{1}{1+e^{(-n)}}$ **tansig (n) =** $\frac{2}{1+e^{(-2n)}} - 1$ **purelin (n) =** n

Now, with multi-layered feedback, they can solve the XOR problem and various non-linear problems.

Neural networks can converge on a continuous function, but they must have the necessary parameters to do this.

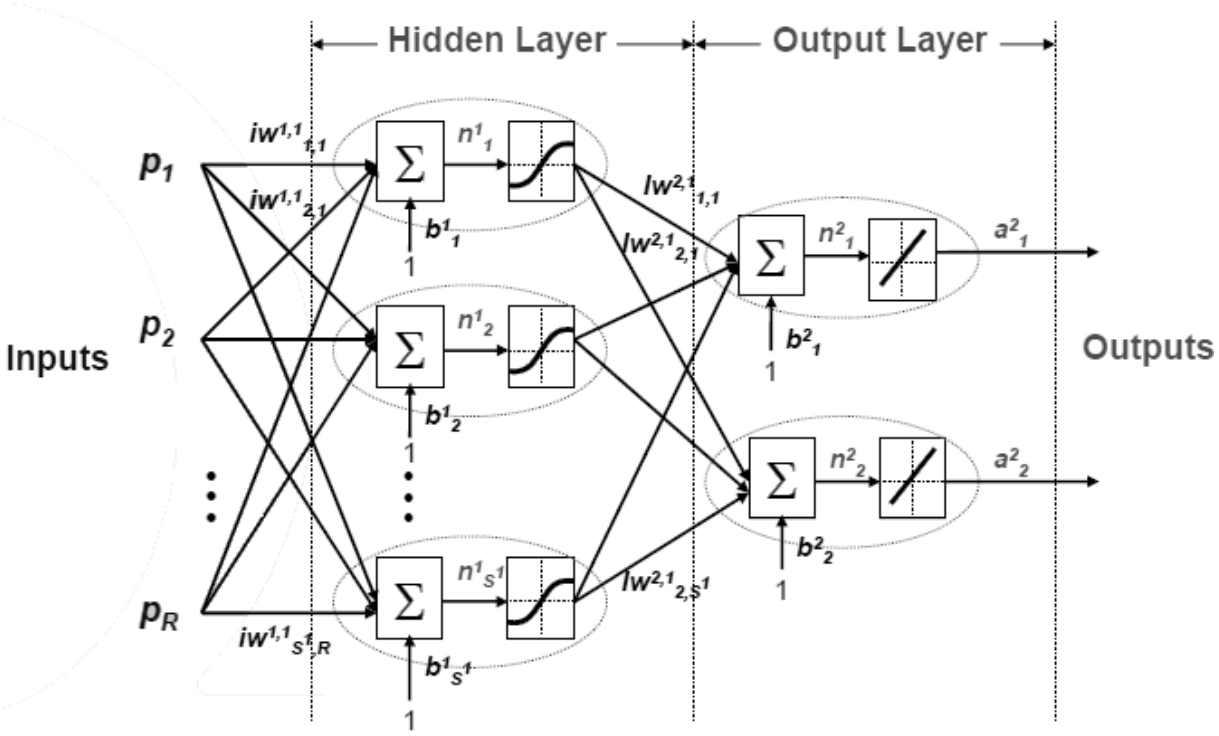


Figure 2.7: Multi-layer network architecture.

We can solve the XOR problem in the figure with the following example:

```
p = [0 0 1 1; 0 1 0 1];

t = [0 1 1 0];

net = newff(minmax(p), [4 1], {'tansig','purelin'},'traingd');

% The first argument defines the range of the input

% The second argument the structure of the network. There are two  layers.

% 4 is the number of the neurons in the first hidden layer,

% 1 is the number of neurons in the output layer,

% Next the activation functions in the layers are defined.

% In the first hidden layer there are 4 tansig functions.
```

```
% In the output layer there is 1 linear function.

% 'learngd' defines the basic learning scheme – gradient method

% We determine the number of iterations 500;

net.trainParam.show = 1;

% After applying the model, the classification process was correct
```

Backpropagation is the main element for building modern neural networks, as it can train and learn from mistakes, but due to the lack of necessary equipment that enables it to carry out large training operations, specialists have not been able to train these networks.

Deep learning is the embodiment of the concept of neural networks. What distinguishes deep learning is the availability of fast hardware and huge amounts of storage space, as well as training data, which enables us to train neural networks and increase the number of layers that can learn hierarchically from lower layers to higher layers of the network.

The work done by (LeCun 1988) [95] by applying deep learning by Convolutional Neural Network to do handwritten character recognition automatically is considered one of the most important examples that provided an illustrative idea to people about what can be done by deep learning, the model can learn patterns by filters that are Stacked within the layers of the model, the filters in the lower layers represent the edges and corners that are used in the upper layers to distinguish image categories.

CNN's have been used in almost all fields and are currently the most powerful image classifier that advances computer vision.

We can divide algorithms into three categories: supervised, unsupervised, and semi-supervised learning.

In supervised algorithms, we present the target patterns in the input and output data, and then the algorithm tries to learn between the input points to set the correct target output. Like the teacher who teaches the student to record, the correct answer, and if the student makes a mistake, the teacher re-guides you again to be better, and more able to estimate the solution.

In unsupervised learning, the algorithm automatically tries to do feature discovery without any signs, such as a student who has a set of questions and answers but does not have a teacher to guide him.

The use of pixels to describe images is more imperative in machine learning models than it was before.

In MLP or feedforward neural networks, each neuron of the input layer is connected to another neuron of the next layer, which is called fully connected (FC), but on the contrary, in CNN, we do not use the concept of (FC) until the very last layer.

We use a nonlinear activation function, such as ReLU, on all convolutional layers to avoid overfitting. Finally, when we reach the end of the network, we apply a layer or two of the FC to get the final classification.

In a CNN hundreds or thousands of filters are applied that allow us to extract the important features that feed an output into the next layer of the network. The CNN model can learn from the values of these filters by its own parameter.

In the field of image classification, the model can learn properties from each layer as follows:

- The first layer allows defining edges
- In the second layer can define shapes
- These shapes used to define features such as facial structures and various pieces, etc. from other upper layers.

The CNN model can make predictions from the last layer of the model. Convolutional neural networks also provide two basic services for us: local invariance and compositionality. Regarding local invariance, which tells us that the image contains an object located in a specific location in the image, pooling layers enables us to obtain local invariance.

The second service is compositionality. Each filter presents a specific swath of features. This allows composition to learn more features as we go deeper. As we touched on earlier, we can get edges from pixels, shapes from edges, and we get complex objects from shapes. Automatically during the training phase. Convolutional neural networks are powerful in computer vision because of their ability to build higher-level features from lower-level features. The figure 2.8 explains the hierarchical principle of building features.

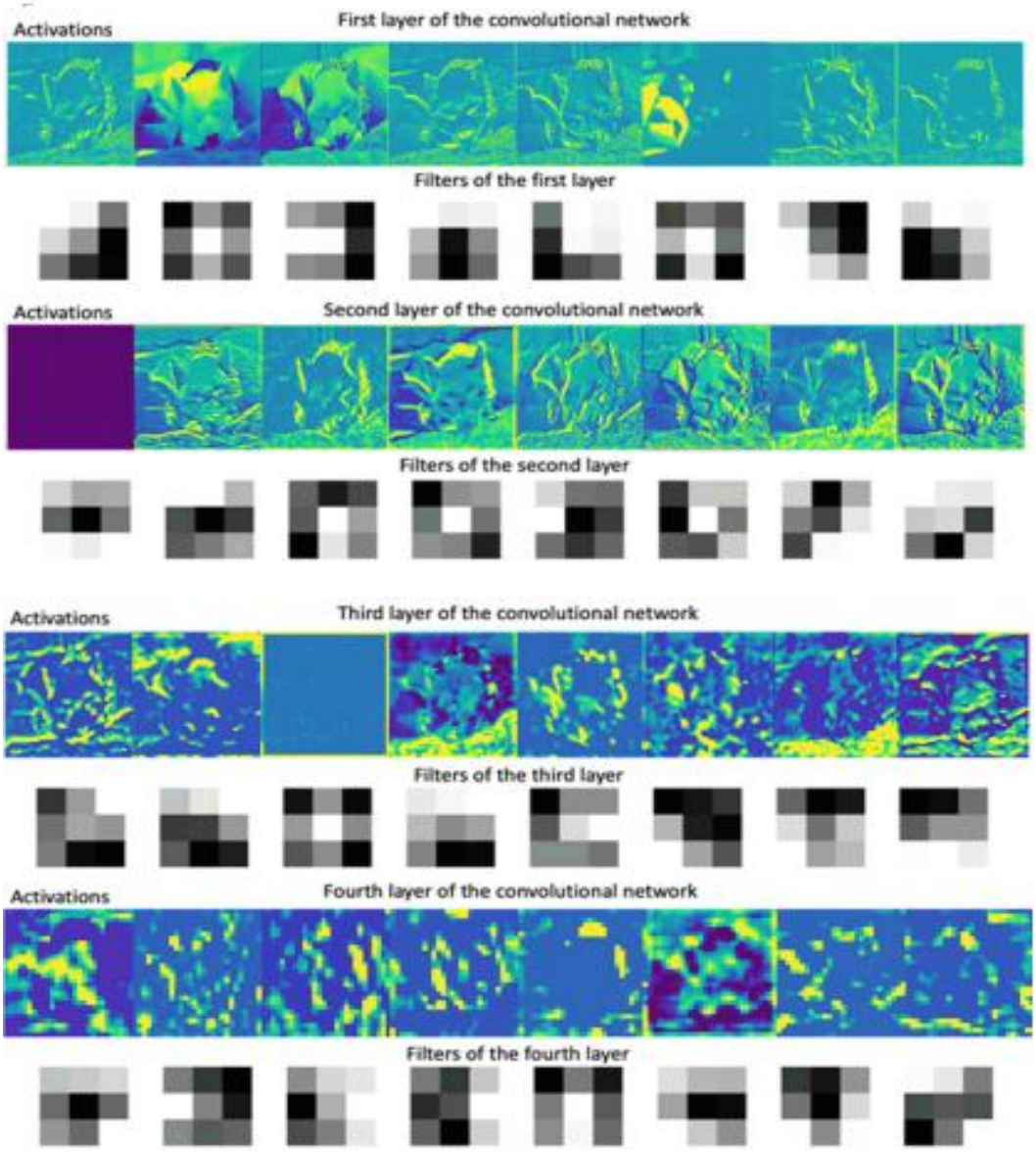


Figure 2.8: Example activations, filters for layers 1, 2, 3 and 4 in a network trained using ImageNet

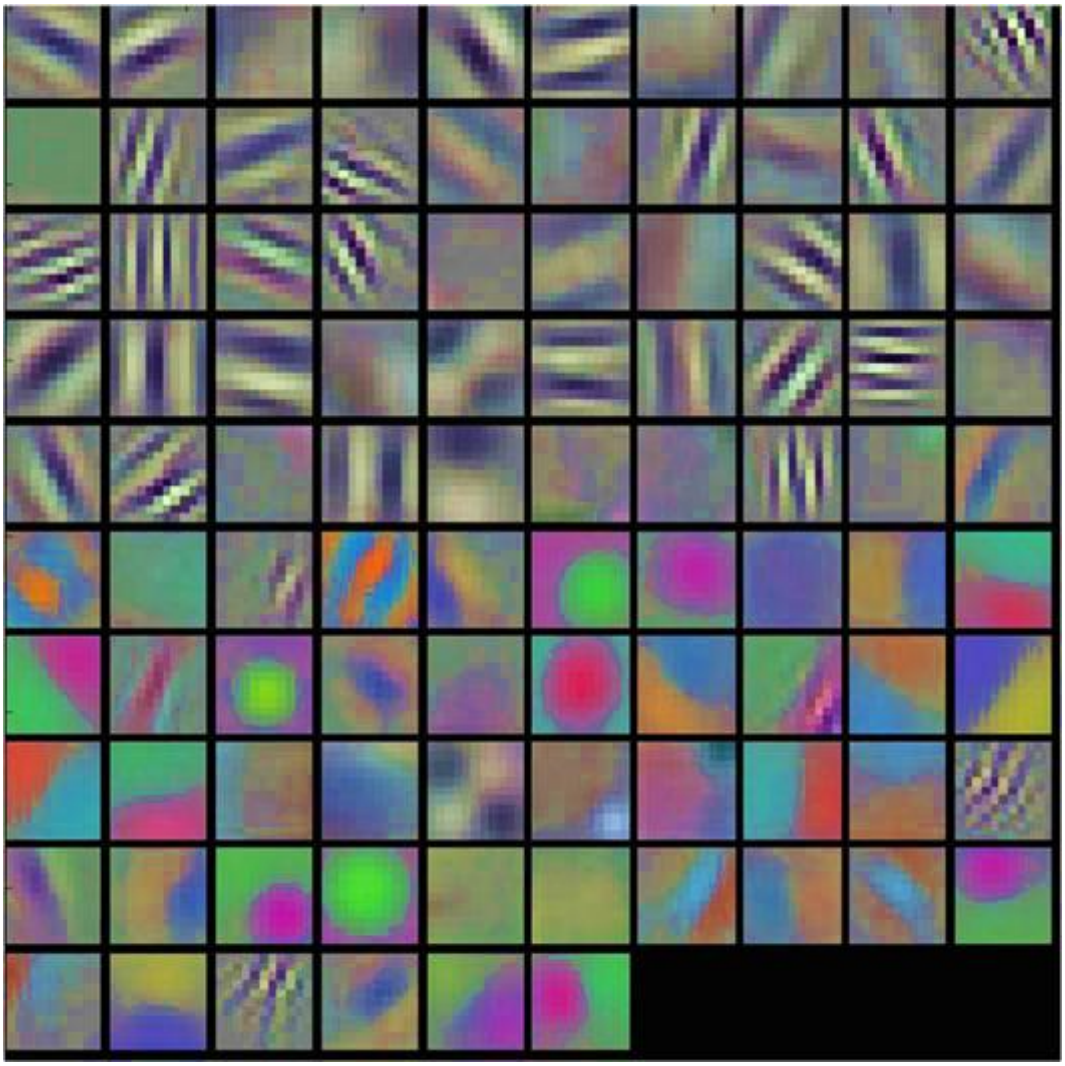


Figure 2.8 (continued)

Before we go into further details, we must explain the concept of image convolutions, and what is the role of convolutions in deep learning. If we want to explain the concept of an (image) convolution in the field that we specialize in, then it means a mathematical multiplication process that occurs between two matrices that have the same dimension and then follows the addition process, and this is the adequate definition of the process.

If we want to explain the concept of an (image) convolution in the field that we specialize in, then it means a mathematical multiplication process that occurs between two matrices that have the same dimension and then follows the addition process, and this is the adequate definition of the process. The following equation explains the convolution that occurs between image I and kernel K:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} K(i-m, j-n)I(m,n) \qquad (2.4)$$

The kernel slides over the entire image from left to right and from top to bottom, and a convolution process occurs.

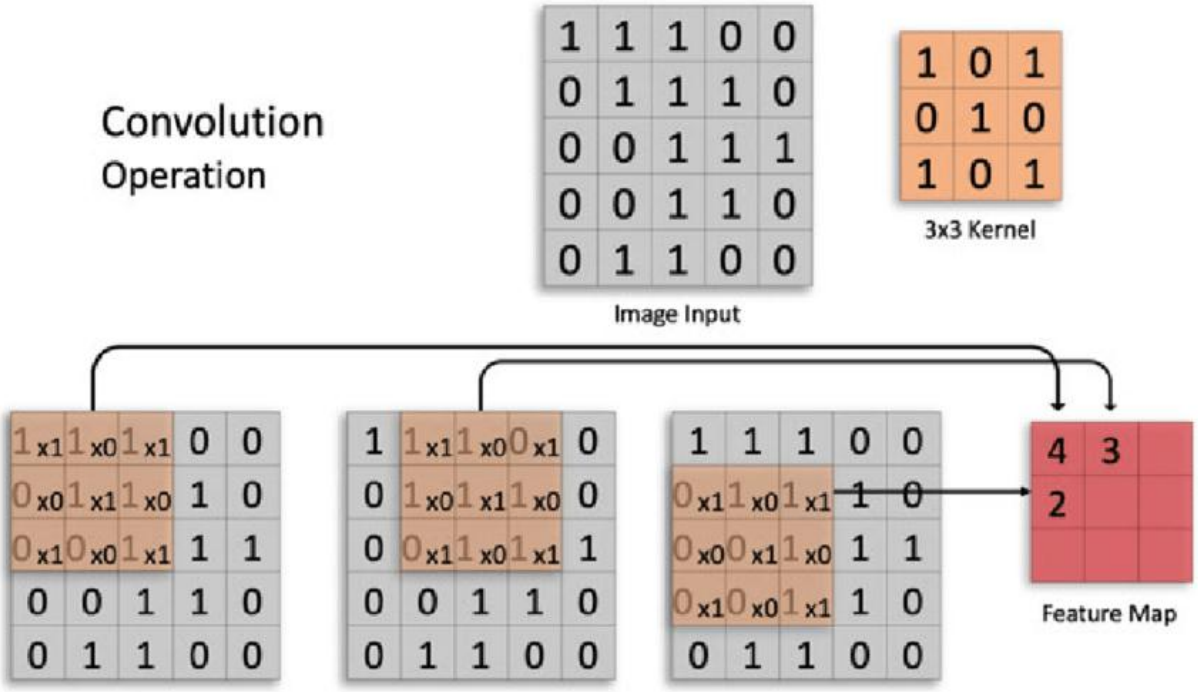


Figure 2.9: An example of 3 × 3 kernel

Convolution process is simple and easy, we have to do:

Determine the coordinates (x, y) from the original image, then locate the center of the kernel on the coordinates (x, y), then apply the last of the previous equation, which results in kernel output. If we take the kernel value for example:

$$K=\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad (2.5)$$

The stages of convolution are as follows:

$$O_{i,j}=\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} * \begin{bmatrix}85 & 70 & 100\\ 50 & 255 & 190\\ 140 & 240 & 25\end{bmatrix} = \begin{bmatrix}85\times 1/9 & 70\times 1/9 & 100\times 1/9\\ 50\times 1/9 & 255\times 1/9 & 190\times 1/9\\ 140\times 1/9 & 240\times 1/9 & 25\times 1/9\end{bmatrix} \qquad (2.6)$$

Therefore,

$$O_{i,j}=\sum\begin{bmatrix}9.4 & 7.7 & 11.1\\ 5.5 & 28.3 & 21.1\\ 15.5 & 26.6 & 2.7\end{bmatrix} = 128.33 \approx 128. \qquad (2.7)$$

2.2 Feature extraction

We worked with two deep networks, VGG 16 and Resnet 34 as downsampling in U-net. As we know, U-net is recommended by [97] for cell segmentation which has given excellent results and success. O. Ronneberger and his group decided to create this network because of the previous model that was done by Ciresan et al. [102] it allows us to predict the class of each pixel by training the network as it obtained excellent results in the EM segmentation challenge at ISBI 2012.

Ciresan's model had two major flaws. First, there are a lot of overlapping corrections and slow execution. Secondly, it requires more max-pooling layers that reduce the accuracy. U-net works with fewer images and gives us accurate segmentations. In Figure 1, we can see the visualization of U-Net architecture, the U-shape of architecture is the reason behind its name. The building blocks of the method and each operation, convolution, max pool, downsampling, upsampling, feature space, etc, are given below.

2.3 ARCHITECTURE

We built our model with a set of layers:

- Convolutional
- Activation Function
- Pooling or Maxpooling
- Batch normalization
- Dropout

2.3.1 Convolutional Layers

The conv layer is the basic component of the CNN. One of the most important components of the conv layer is the filters that are always square and have width and height. The number of filters applied to the input image represents the depth of the network.

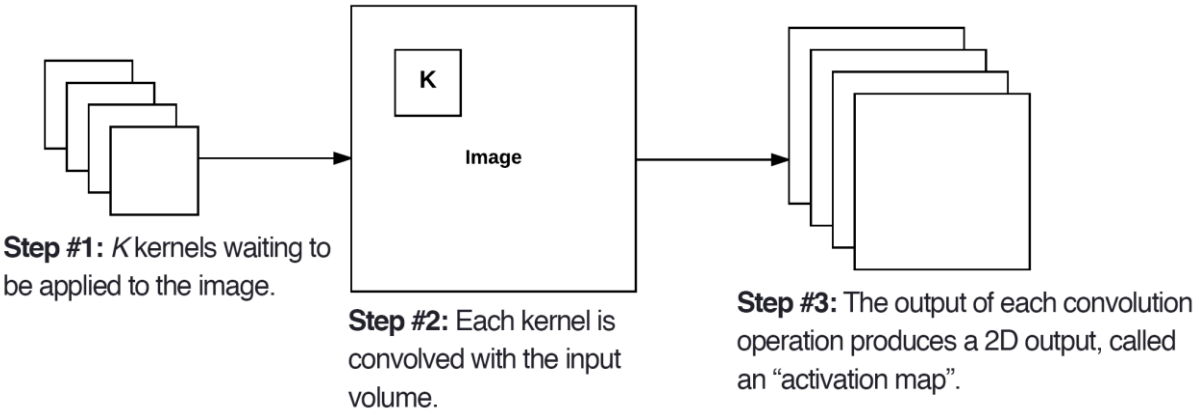


Figure 2.10: The stages of applying kernels to the input layers that produce the activation map.

2.3.2 Activation Layers

After each conv layer we apply activation function ReLU, there are other types of activation function such as ELU...

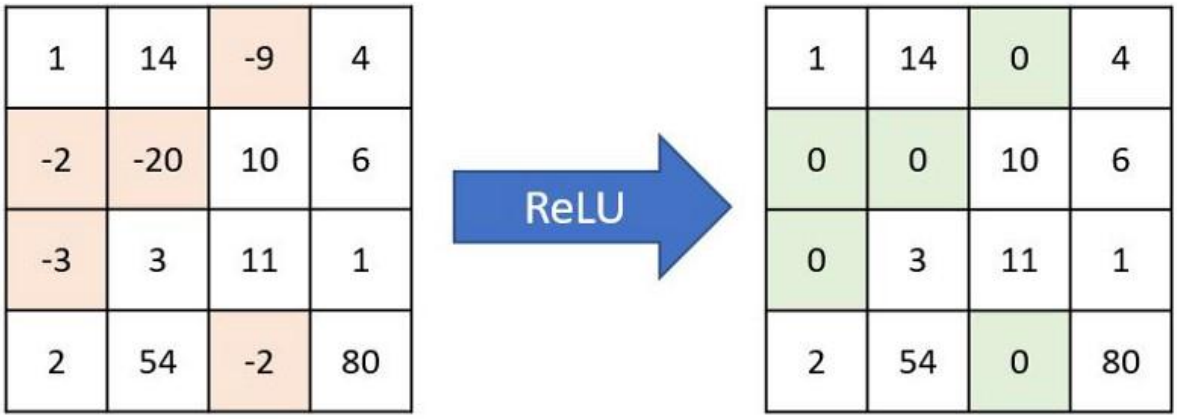


Figure 2.11: ReLU operation.

2.3.3 Pooling Layers

The pooling layer (also called downsampling) is dedicated to reducing the size of the spatial resolution of the map entity, which leads to a drastic decrease in the number of parameters (and computations) that ultimately reduce the risk of over-learning of the model. It reduces dimensionality while retaining the most important information. Another important benefit of the pooling step is that it makes the CNN invariant to minor transformations, distortions, and translations of the input volume. The most known types of pooling are max-pooling, average pooling, and even L2- pooling standards.

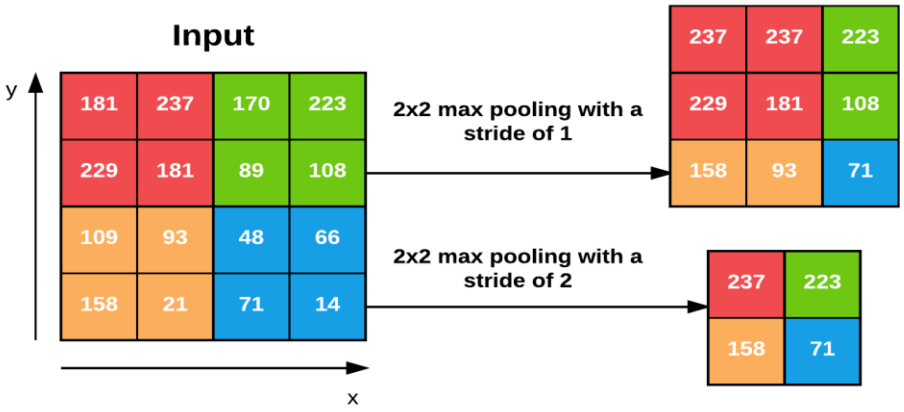


Figure 2.12: Apply max pooling to input 4×4 volume with different strides.

Some networks allow pixel classification such as LeNet [95] and AlexNet [104], Figure 4.4 contains a structure used for classification. feature maps are controlled by fully connected layers that consist of the elements we mentioned before, such as number of filters (kernels), types of Regularization (dropout and batch normalization)...

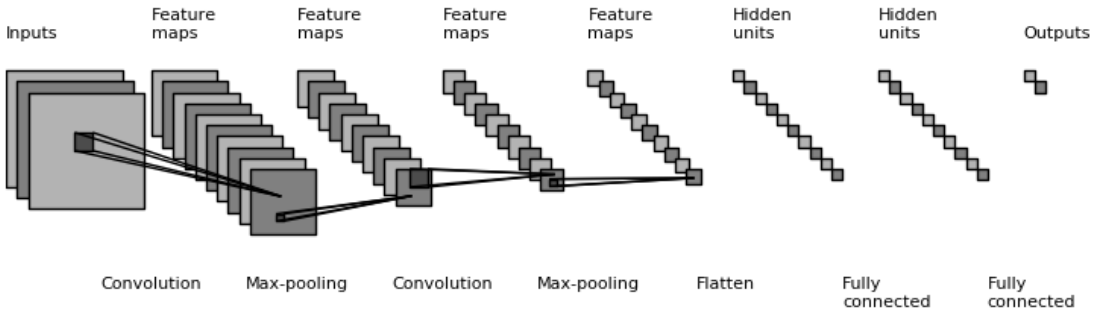


Figure 2.13: Architecture for classic Neural Network [95] .

When the neuron weights are stable, we can say that the neural network has learned successfully. The weights are adjusted to learn specific features of the training data.

When we hear the term deep learning, most of us imagine that we are talking about a deep neural network accumulated on top of each other, as well as a large number of layers that get deeper as we get deeper, but there is a somewhat vague idea represented in that how many layers do we need to be able to say that the neural network is deep?

For the answer to be clear, the experts have unanimously agreed that there is no specific depth. We can say that the network has become deep. However, I am convinced of what was presented in [116], where he presented proofs that we can consider the network deep if it contains more

than two layers. We can also say that the network is very deep if it contains more than 10 layers.

2.3.4 Sigmoid function

The sigmoid function allows the output to be predicted, by plotting the input to values between zero and one. It enables us to reach the importance of each feature by assigning values to the components. Where the value of zero deletes the feature directly, while the value of one keeps every information.

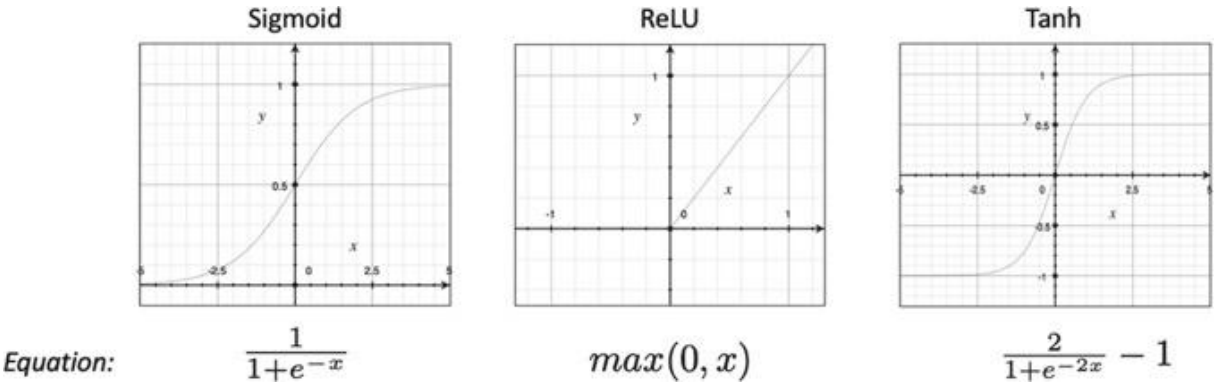


Figure 2.14: Sigmoid function with some of the most common functions

2.3.5 Softmax function

The softmax function converts vector values of K real values that are positive, negative or zero into values between zero and one, which enables us to interpret them as probabilities. If the entered value is very small or neglected, then the probability is small, on the contrary, if the input is large, it will be converted to a large probability, and the values will remain confined between zero and one. The softmax formula is as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (2.8)$$

$\vec{z}$: The input vector to the softmax function (z_0, ...z_K).

z_i: Elements of softmax function input, they can take any real value such as (-0.47, 6.17, 3.51)

e^{z_i}: The standard exponential function that includes all entered values, This gives a positive value above 0.

$\sum_{j=1}^{K} e^{z_j}$: This equation ensures that all values of the output will sum to 1 and are in the range (0,1).

K: It represents the number of categories to be classified for.

Suppose we have an array of three real values, we want to convert them into a probability

distribution.

$$\begin{bmatrix} 9 \\ 4 \\ 0 \end{bmatrix} \qquad (2.9)$$

We will compute the exponential of each element of the input array.

$e^{z_1}=e^{9}=8103$

$e^{z_2}=e^{4}=54.6$

$e^{z_3}=e^{0}=1$

Then we calculate the following sum:

$\sum_{j=1}^{K} e^{z_j}= e^{z_1} + e^{z_2} + e^{z_3} = 8103 + 54.6 + 1 = 8158.6$

Finally, we get the following probabilities:

$\sigma(\vec{z})_1=\frac{8103}{8158.6} = 0.9821.$

$\sigma(\vec{z})_2=\frac{54.6}{8158.6} =0.0067.$

$\sigma(\vec{z})_3=\frac{1}{8158.6} =0.00001.$

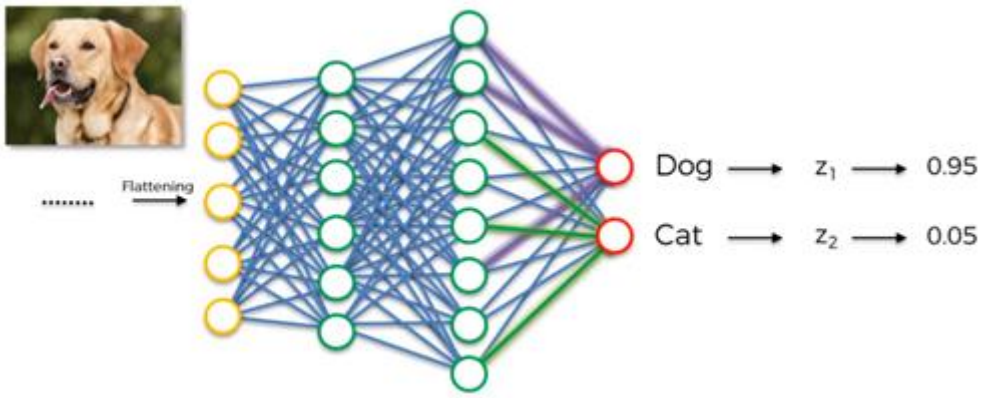


Figure 2.15: Example of implementing the Softmax function inside the Flattening process

2.3.6 Fully-connected Layers

FC layers are always placed at the end of the network to do the final classification. Some models use one or two layers of FC before applying the softmax classifier. The following figure shows the steps.

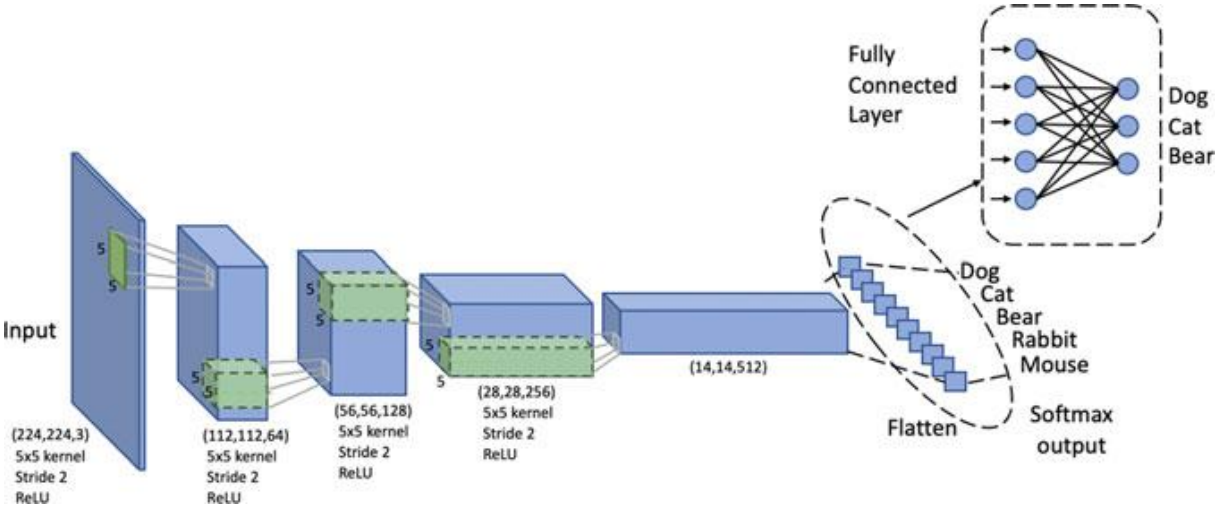


Figure 2.16: A simple model of a CNN that has a FC at the end of the structure

2.4 Optimization Methods

The CNN model based on the Stochastic Gradient Descent (SGD) algorithm to obtain the best convergence.

Loss functions we use to estimate the quality of the prediction (good or bad) when we classify the input data.

Convolutional networks rely on basic elements to learn patterns from data, the most important of which is optimization algorithms. We can say that we obtained a classifier with high accuracy because the classifier was able to obtain ideal values for weights W and b.

The question we can ask is how can we get the ideal weight matrix W and bias vector b values to increase the classification accuracy, and since we are in deep learning there are millions of parameters, which makes this process take longer if it is manual or relying on randomness. Therefore, an optimization algorithm had to be relied upon to optimize the W and b values for us.

One of the most popular algorithms used to train convolutional neural networks called- gradient descent. Its working principle is simple. You have to iteratively evaluate your parameters, then calculate the loss value, and then take a very small step towards the direction that minimizes the loss value. This is the general principle of the algorithm.

The Loss Landscape and Optimization Surface:

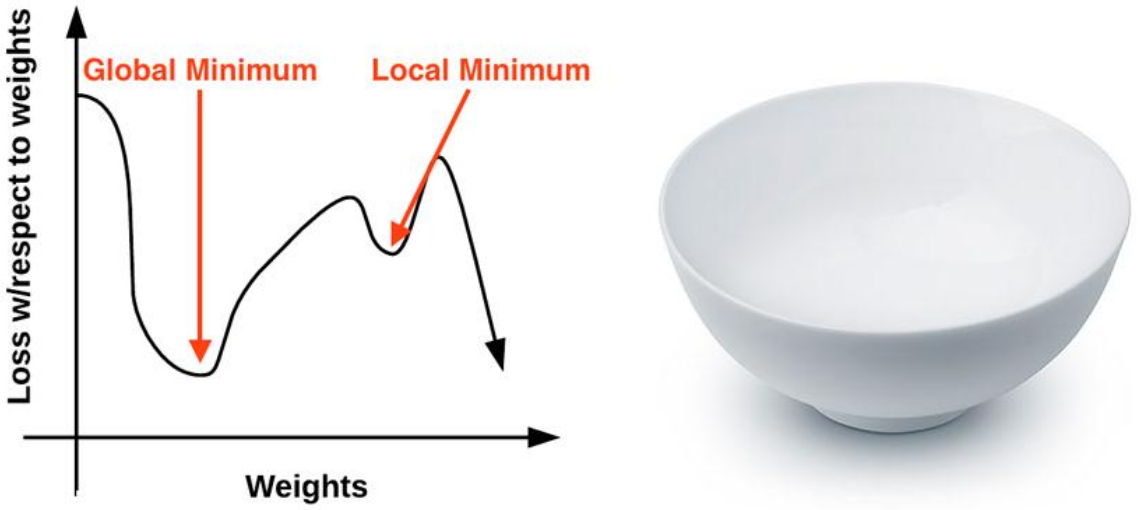


Figure 2.17: On the left is the loss value as a 2D plot, and on the right: Gradient stages to reach the lowest loss value located at the bottom of the container.

As shown on the left of the figure, there are many local minimum values, but we are looking for a global minimum. Why can't we directly reach the global minimum by eye or by hand? The reason is that it is not visible and we do not know its exact position (we are like blind people) and the optimization algorithm is the stick that leads us towards loss minimum and not falling into local maximum.

The gradient descent algorithm has one of its most important disadvantages, which is called **computational wasteful.**

The appropriate alternative to gradient descent is to use Stochastic Gradient Descent (SGD), which represents a simple modification of gradient descent by increasing the number of steps along the gradient, which speeds up the convergence process and does not harm the classification loss.

SGD has the advantage of being one of the most popular algorithms for training deep convolutional neural networks. Although it has been more than 60 years since it was first used. It is still the engine that allows us to train networks to learn different patterns.

Most specialists are still striving to provide some positive characteristics in optimization algorithms, including: reducing the time and at the same time obtaining reasonable accuracy.

2.4.1 Adaptive Learning Rate Methods

With the continued development of deep learning new techniques of optimization have emerged that seek to improve SGD through adaptive learning rates.

The following equation introduces how SGD works:

W += -lr * dW (2.10)

W: Weight matrix.

lr: The learning rate.

dW: The gradient of W.

We will mention the most important adaptive learning rate, which is common in deep learning:

2.4.1 Adagrad

Presented by Duchi et al [117] where it is based on larger updates on parameters.

2.4.2 Adadelta

Introduced by Zeiler in 2012 [118], it is considered as a complement to Adagrad as it seeks to reduce the learning rate. It is updated every time from squared gradients

2.4.3 RMSprop

Created independently of Adadelta, but the notable thing is that it has the same actual W-weight matrix update as the Adadelta update.

2.4.4 Nadam

It was proposed by Timothy Dozat at Ph.D. student at Stanford University [119] is of little use, but we must understand that there is a difference in this algorithm.

2.4.5 Adam

The Adam (Adaptive Moment Estimation) proposed in 2014 by Kingma and Ba is RMSprop but with additions to it.

We must understand an important idea in the field of previous optimization algorithms, which is that there is no definitive and clear evidence that determines the use of a specific algorithm. Schaul et al. in 2014 by conducting a test on Stochastic Optimization [120] and it was found that there is no specific winner among all these algorithms.

2.6 Regularization

It is one of the most important concepts that allows the model to generalize and fine-tune and reduces the effects of overfitting. It contains a group of types that help modeling, which we will mention in detail. Among the most important types of regularization are: dropout, data augmentation and early stopping

Regularization helps us generalize by ensuring better model performance and correct classifications. It further allows the ability to generalize new images to our model.

Conclusion

In this chapter, we introduced the concept of deep learning, where we discovered that it is not newly emerging, but that deep learning has existed since the forties of the last century, but in different manifestations based on different schools and intellectual researches. Deep learning is derived from artificial neural networks (ANNs), which represent a group of algorithms that it allows learning of different patterns inspired by the function and structure of brain neurons. Deep learning algorithms can learn in a hierarchical manner, as we mentioned in this chapter, where layers are stacked on top of each other, extracting features that enable us to learn the objects.

Chapter 3: Implementation Methodology

3. Implementation Methodology

Convolutional networks are one of the most common and popular parts of deep learning, as they resulted after improvements to artificial neural networks, consisting of layers that enable us to classify data features [96]. It has now become common to rely on convolutional neural networks for their strength and effectiveness in the field of image classification and segmentation. Figure 3.1 shows the general layout of the proposed vascular segmentation model.

U-net is recommended by [97] for medical image segmentation. The results show that the method is superior in segmentation. The U-Net architecture has been applied many times to segment the retinal vasculature [98; 99]. We propose a U-net network structure based on VGG 16 [100] the depth convolutional network and deep residual learning where we use the network Resnet 34 [101]. We increased the depth by pushing the deep to 16 weight layers.

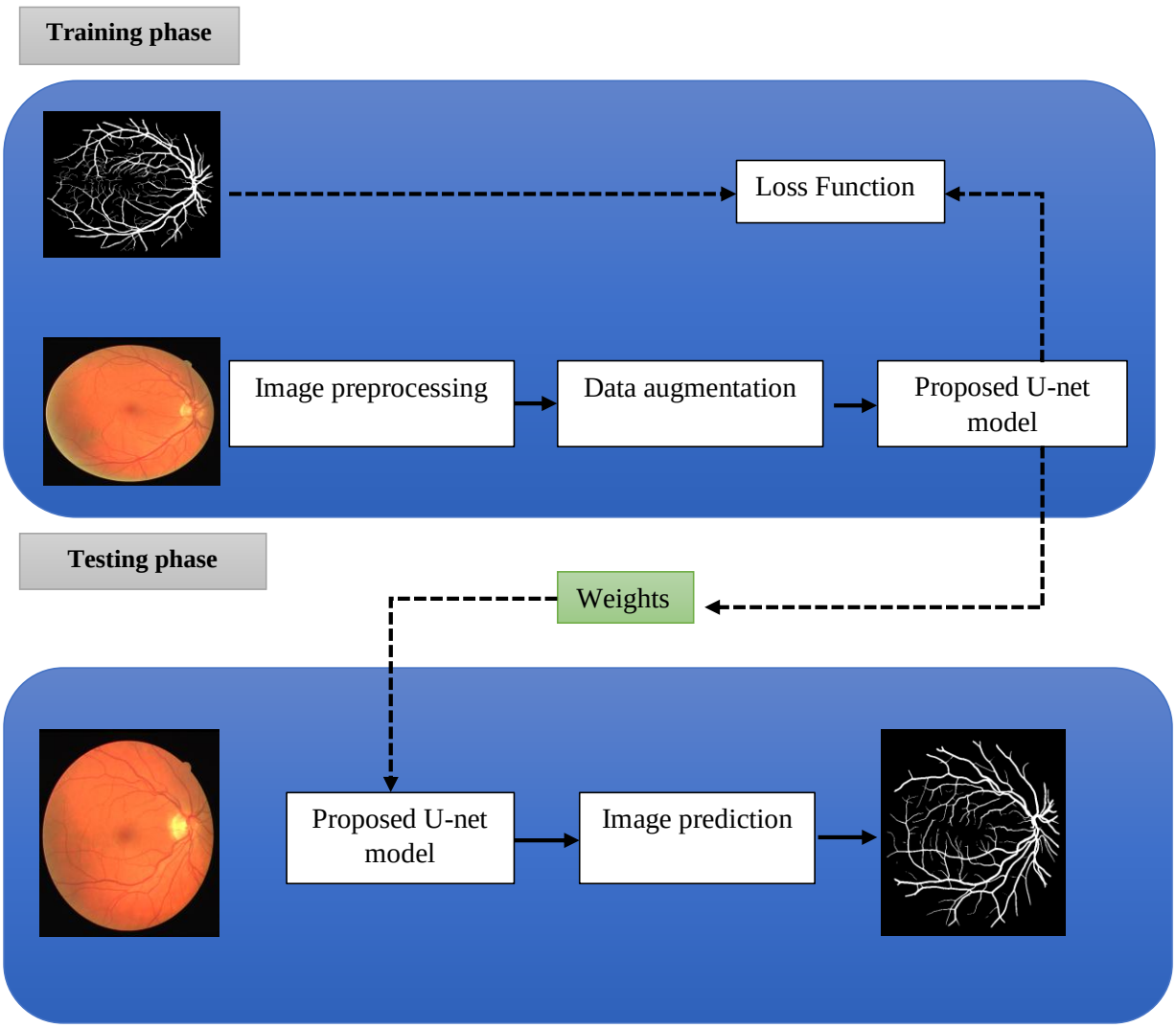


Figure 3.1: General outline of the proposed method

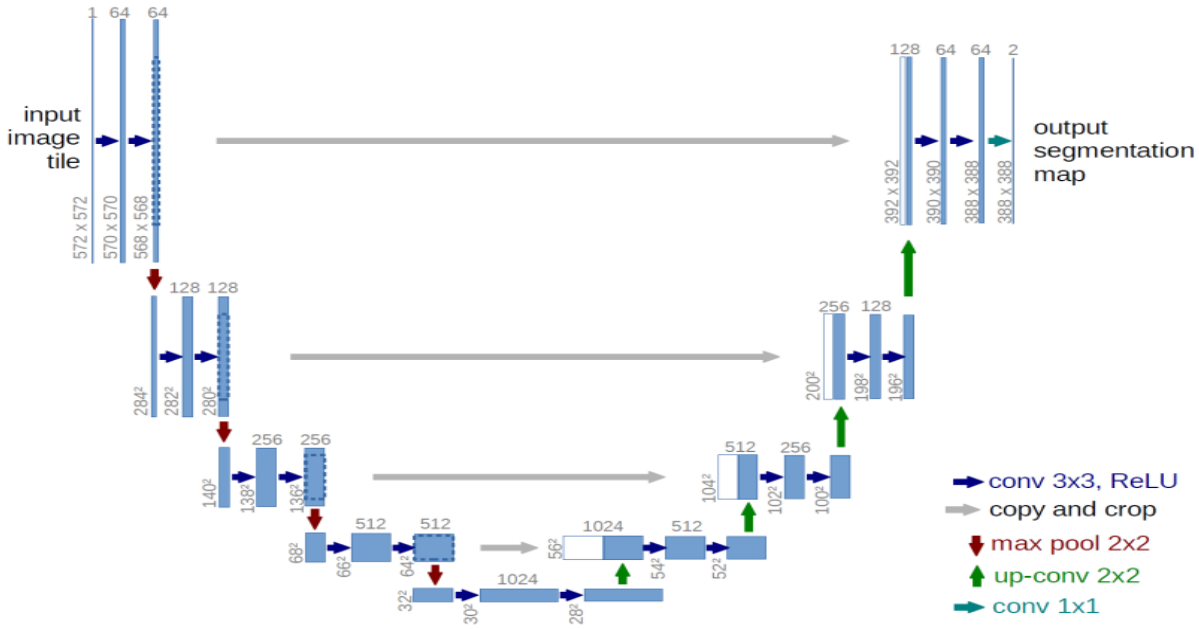


Figure 3.2: The U-Net architecture.

The architecture contains two paths: Encoder and Decoder. The path on the left is called the contraction path (encoder path) or downsampling, and the path on the right is called the decoder path or upsampling. Downsampling is the operation where the output represents information from the input for another step. Upsampling has the same structure as downsampling but in the opposite direction; It is the network that takes the output features from downsampling as input and provides the same match of the actual input. The operation of the encoder allows for reducing the size input matrix and increasing the feature maps, and on the contrary, the decoder path is returning the matrix to its original size by unpooling and reducing the feature maps. The advantage of U-net is that it increases image quality and resolution by replacing pooling with upsampling operators because we combined high-resolution features from the downsampling with upsampling output. The U-net allows the user to do the segmentation by an overlap-tile strategy (pixel induction by mirroring the input image). The network gives us good and important results whenever we have a large number of images, so we have augmented our data DRIVE and STARE database by application deformations such as rotated images, flipping, and adding noise with different.

The data entered into the model has a fundamental role in the extent of learning, as the model must be able to absorb large amounts of input so that it can also identify new models, so pre-processing of the entered data must be done for the model to benefit from it in training, in this section we will present the various procedures that We followed it to do pre-processing of the retinal images.

3.1 Preprocessing of images

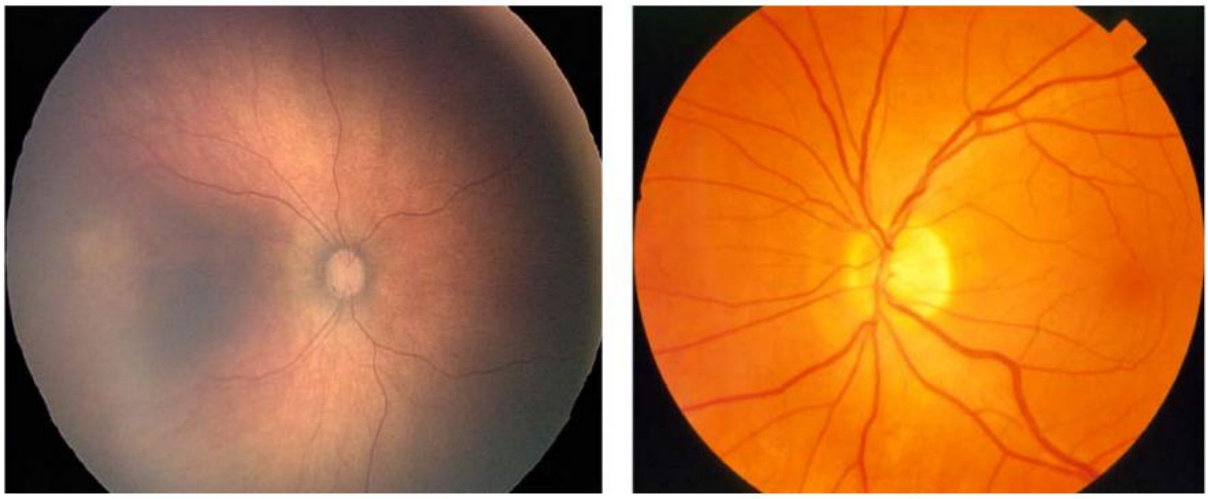
First, we pre-processed the images before submitting them to our model. This stage allows for improving the quality of the image which stimulates CNN to discover certain characteristics. U-net gives excellent results when having a large number of images. In [70] they used green channel intensity in their models, and others used the color image as it is in reality, as we do not prefer to use color images that do not show the contrast between the blood vessel and the background, unlike the gray image, in which the difference is clear, which helps us in providing useful information us in training. The steps of this stage are mentioned in detail as follows:

a) We converted the RGB image to gray where the contrast of the blood vessels is higher. The formula used was the following:

$$Image = R * 0.299 + G * 0.578 + B * 0.114 \qquad (3.1)$$

where R, G, and B represent channels of the input image. The largest percentage was taken from the green channel (G) because it contains the highest contrast [103].

b) Limited contrast histogram equalization (CLAHE) was used for enhancing the local contrast of the image. This step allowed us to increase the contrast between the background and the micro-vessels, especially, those that were difficult to segment later. CLAHE processing is preferred before any segmentation process if possible The following form shows the clear difference between the thickness of the blood vessels in the very thin child and the adult, so we have to improve contrast to enhance the blood vessels before retail.



a b

Figure 3.3: (a) Example image of an infant retina (b) Example image of an adult retina.

c) Local adaptive Gamma correction was used to correct the retinal image and standardize the intensity of light over the entire image as well as reduce the reflection of the centerline.

The formula is defined as follows :

$$O_T = I + R \times \sum_{i}^{n} w_i (D_b - D_d) \qquad (3.2)$$

where R is the retinal vascular image detail enhancement coefficient; I is the input image; O_T is the output image; D_b and D_d are the bright and dark details of retinal images respectively. Because most of the images of the retina are not illuminated over the entire image.

The pre-processing results are shown in Figure 3.4.

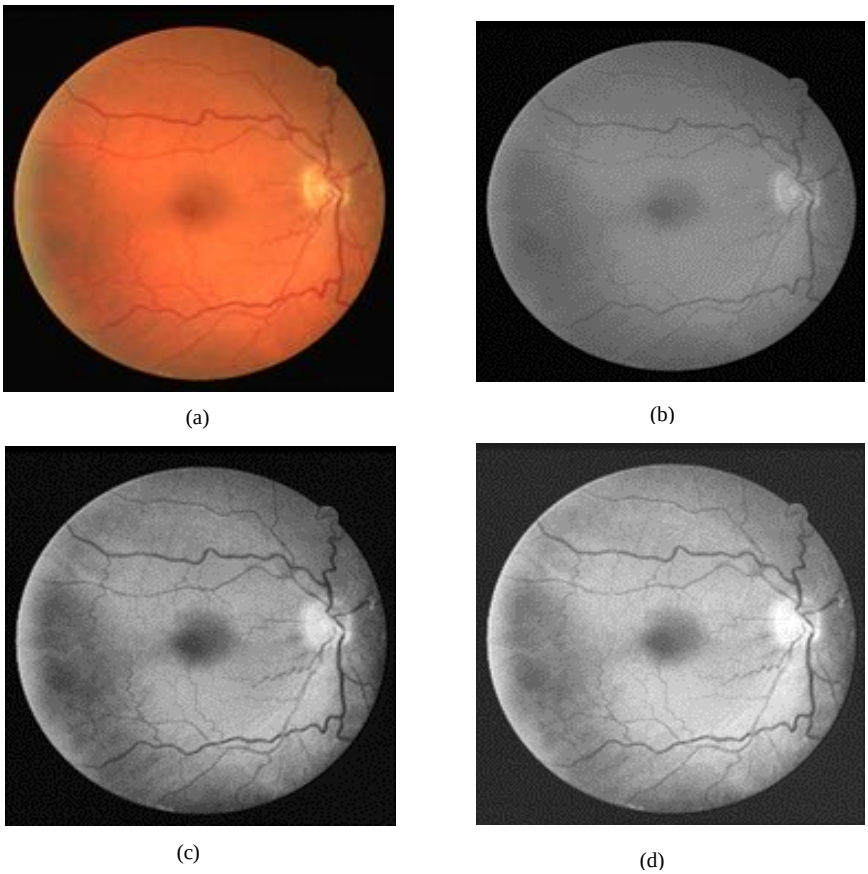


Figure 3.4: Pre-process stage. (a) Original image. (b) Gray image. (c) CLAHE image. (d) Gamma image

3.2 Image augmentation

After finishing the pre-processing phase of the image, the next step was image augmentation. Deep learning requires a large amount of data to reach high performance; convolutional networks have flexibility in front of various changes that occur to the image, such as geometric transformation and deformation, and illumination.

The rate of overfitting of the form increases if the input data is small. The idea of increasing data allows the creation of new training samples based on the original samples through the changes we mentioned before. We aim by increasing the data to increase the generalizability of our model and its ability to more accurately identify different vessels of different sizes which can sometimes reduce the value of accuracy.

We augmented the dataset through the following operations:

Rotating image by an angle of 30° to 330° with role 30, gave 12 images from only one image. We changed the color of the image by the following formula:

$$n = random\,(-20, 20) \tag{3.3}$$

$$I = image + n \tag{3.4}$$

where n corresponding integer number. I, represents the image that we obtained after adding color to the original image.

By flipping the image in different dimensions (-1, 0, 1), we applied gamma and also changed the color , and added noise to each image we flipped. The sum of the images we obtained after image pre-processing and data augmentation became 980 images from 20 original images. Data augmentation techniques its shown in Table 4.1.

Table 3.1 : Data augmentation techniques

Technique	Parameter
Rotate image	[0,360^{o},30^{o}]
Random change gamma	Range: [0,1]
Flip image	Range: [-1,1]
Color change	Range: [-20,20]

In the original image of Figure 3.5, there were incorrectly segmented vessels due to their small size. After we applied the process CLAHE, the image became more enhanced, and thus the small vessels became clearer and the segmentation efficiency was improved.

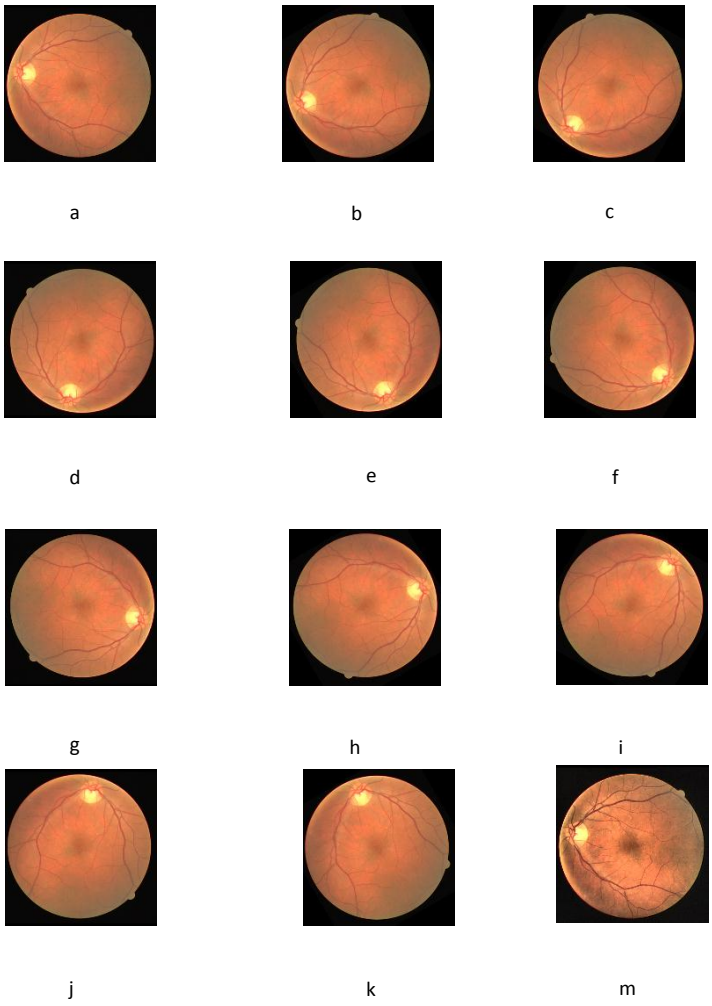


Figure 3.5: Data augmentation by rotating. (a) Original image. (b)

Rotating 30° image. (c) Rotating 60° image. (d) Rotating 90° image. (e) Rotating 120° image. (f) Rotating 150° image. (g) Rotating 180° image. (h) Rotating 210° image. (i) Rotating 240° image. (j) Rotating 270° image. (k) Rotating 300° image. (l) Rotating 330° image. (m) CLAHE image.

In the original image of Figure 3.4, many vessels are incorrectly segmented due to their small size. After we apply the process CLAHE, the image becomes more enhanced, and thus the small vessels become clearer, and segmentation efficiency is improved.

In the image database used, the same color exists between the blood vessels and the background of the image which caused a problem in the process of learning characteristics of the whole blood vessel, therefore, the process of enhancement of the original image enhanced the learning of blood vessels. The division of the blood vessels was incorrect in some images as the segmented images were full of noise due to texture existence, but after enhancement processing, the blood vessels became clearer and this led to supporting blood vessel segmentation. There were extremely dark original images where only thick blood vessels were visible, therefore, the segmentation of the thin blood vessels was difficult. After image optimization, the vessels became clearer and we could see the small details.

After the preprocessing and data augmentation, our flipping and images rotating did not change the shape and texture information of blood vessels which meant that the properties and parameters for the model of convolutional neural network used for training were enhanced. CLAHE image processing changed and improved the pixel value, moreover, it saved us the directions of the vessels and showed us more micro-vessels.

3.3 U-NET

To achieve the best result in the semantic segmentation process, many contributions retained some important details [97]. However, the training process, in terms of data quality, size, and format, was difficult but the ideal choice of the trained model test was fast and less memory-consuming than expected. Therefore, scholars had done training using a strategy that relied on the strong use of data augmentation [97]. Another way was through data enhancement and expansion according to this feature [103].

In addition to the suggestions mentioned above, we changed the structure of the network to solve various problems and we changed the network's main goal that was set for our benefit.

3.4 VGG-16

Krizhevsky et al. [105] provided a structure that aided in computer vision and data representation, but that was insufficient. The original structure had been optimized for better accuracy after that. By [106-107] smaller steps and windows were used for the first convolutional layer. VGG had an increased depth by adding more convolutional layers, starting from 11 layers to 19 layers, which used only small (3×3) convolution filters in layers.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3.6: ConvNet configurations [100].

As a result, the results obtained were very accurate in terms of classification, localization, and image recognition data sets. VGG can generalize to an untrained data set. **We chose to work with VGG exactly with 16 layers of coding because the data representation is high performance and the error was lower [100]. We used the Alexnet model with the U-net first, but our hardware could not handle the data, so we changed the model to the VGG-16 network, which gave us amazing results. We chose VGG with layer 16 because the error value is lower compared to the other VGG layers.**

The reason for doubling filters is to ensure fairness and no bias between layers, the first layers of the model contain fewer filters compared to the final layers, but they are characterized by their large spatial size, which indicates the presence of data that can be learned from.
After reducing the size by max-pooling, the number of filters is increased in the next layer to avoid our layers being unbalanced and potentially biased, which indicates that the first layers of the network have a high ability to filter and output values more than the deep layers, so it is

preferable after each size reduction process if volume size 50% or more that we increase the number of filters.

The following figure provides us with the details of the proposed network, consisting of two stages, the first is encoding, where the feature extraction process is carried out utilizing a regular set of convolutional layers, such as the one contained in the VGG-16 network, with the reduction of dimensions caused by the max-pooling operator. In the second stage, the process is Decoding, where the segmented image is finally obtained after merging the layers from both stages (Encoding- Decoding) and increasing the size by Up-sampling.

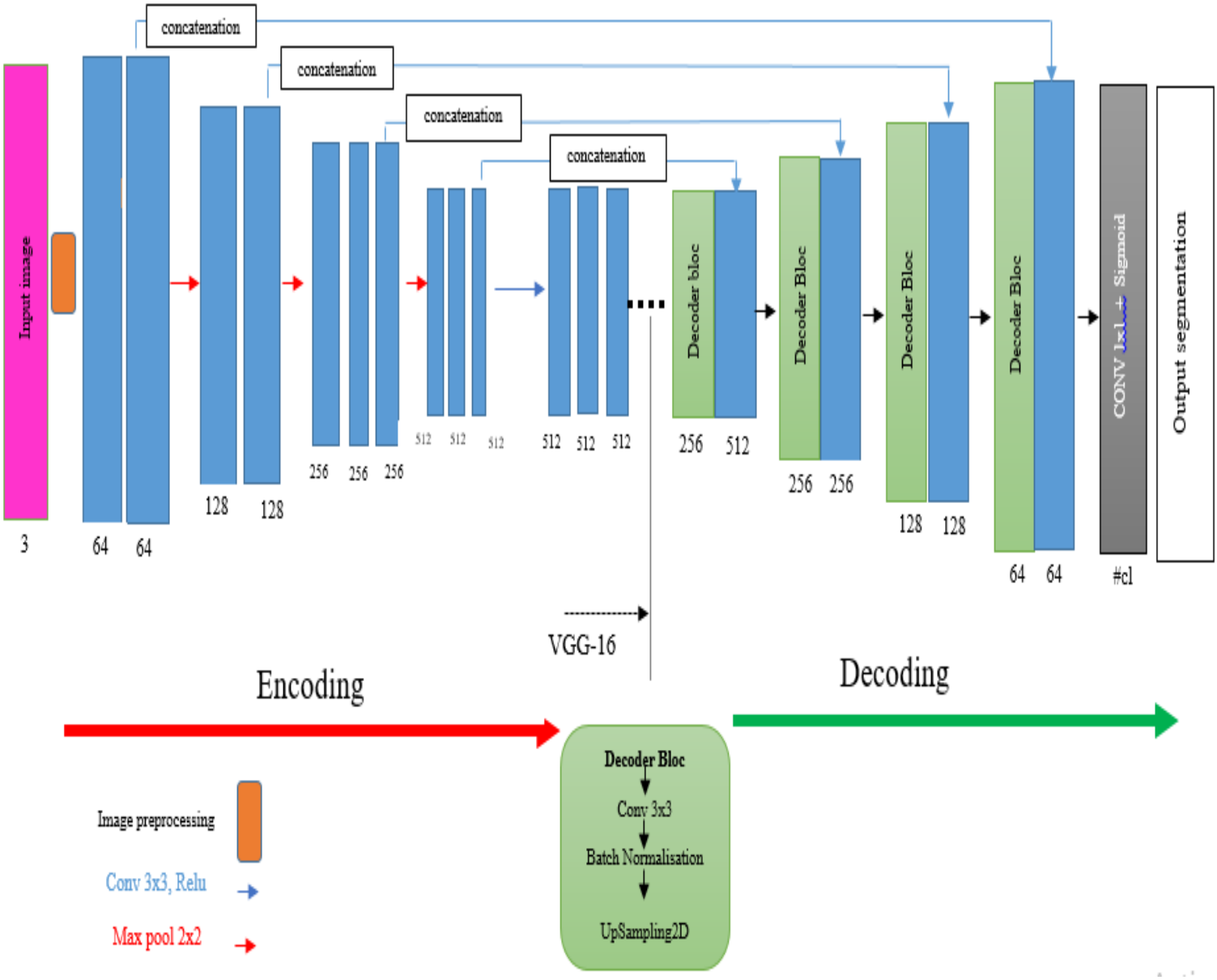


Figure 3.7: Proposed architecture consists of a U-net with VGG 16 as encoding to extract features.

Table 3.2: Construction of VGG 16 + U-NET [121]

	#	*Input image*			*Output*			*layer*	*stride*	*kernel*	*in*	*out*	*Param*
Encoding	1	576	560	3	576	560	64	Conv3-64	1	3x3	3	64	1792
	2	576	560	64	576	560	64	Conv3-64	1	3x3	64	64	36928
		576	560	64	288	280	64	maxpool	2	2x2	64	64	0
	3	288	280	64	288	280	128	Conv3-128	1	3x3	64	128	73856
	4	288	280	128	288	280	128	Conv3-128	1	3x3	128	128	147584
		288	280	128	144	140	128	maxpool	2	2x2	128	128	0
	5	144	140	128	144	140	256	Conv3-256	1	3x3	128	256	295168
	6	144	140	256	144	140	256	Conv3-256	1	3x3	256	256	590080
	7	144	140	256	144	140	256	Conv3-256	1	3x3	256	256	590080
		144	140	256	72	70	256	maxpool	2	2x2	256	256	0
	8	72	70	256	72	70	512	Conv3-512	1	3x3	256	512	1180160
	9	72	70	512	72	70	512	Conv3-512	1	3x3	512	512	2359808
	10	72	70	512	72	70	512	Conv3-512	1	3x3	512	512	2359808
		72	70	512	36	35	512	maxpool	2	2x2	512	512	0
	11	36	35	512	36	35	512	Conv3-512	1	3x3	512	512	2359808
	12	36	35	512	36	35	512	Conv3-512	1	3x3	512	512	2359808
	13	36	35	512	36	35	512	Conv3-512	1	3x3	512	512	2359808
Decoding		36	35	512	72	70	512	Up-sampling	2	2x2	512	512	0
	14	72	70	512	72	70	512	Conv3-512	1	3x3	512	512	4719104
	15	72	70	512	72	70	256	Conv3-256	1	3x3	512	256	1179904
		72	70	256	144	140	256	Up-sampling	2	2x2	256	256	0
	16	144	140	256	144	140	256	Conv3-256	1	3x3	256	256	1179904
	17	144	140	256	144	140	128	Conv3-128	1	3x3	256	128	295040
		144	140	256	288	280	128	Up-sampling	2	2x2	128	128	0
	18	288	280	128	288	280	128	Conv3-128	1	3x3	128	128	295040
	19	288	280	128	288	280	64	Conv3-64	1	3x3	128	64	73792
		288	280	64	576	560	64	Up-sampling	2	2x2	64	64	0
	20	576	560	64	576	560	64	Conv3-64	1	3x3	64	64	73792
		576	560	64	576	560	64	Conv3-64	1	3x3	64	64	36928
	#	576	560	64	576	560	1	-	-	-	64	1	65
Total params													22,568,257

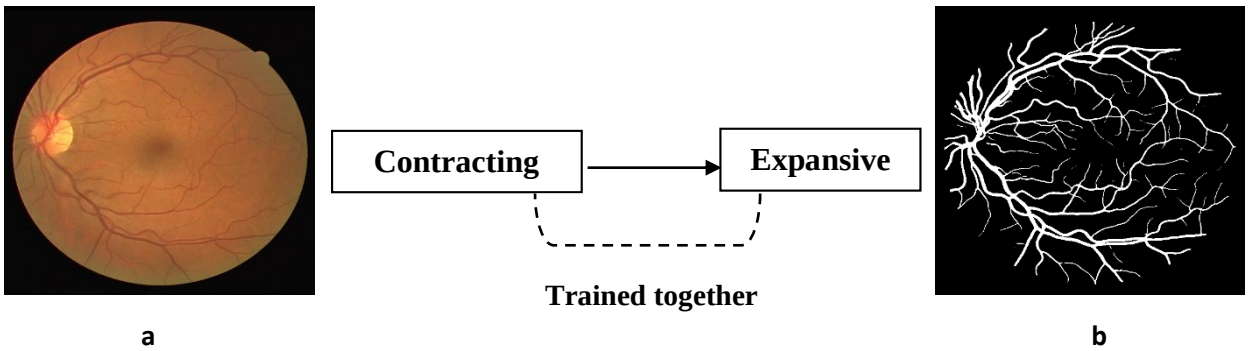


Figure 3.8: segmentation process under U-Net+VGG16. a: input image. b: segmentation result.

As shown in table 3.2 and figure 3.7, we used the encoding structure in the contracting path by using the structure VGG 16 with its layers to get the extraction features, then we used the decoding structure to represent the expansive path.

Moreover, in detail, we used five levels. In the first level, where the image size for DRIVE was 576x 560, we passed the image through a set of convolutional layers where we used filters with small receptive 3 x 3 in double 64 channels followed by max-pooling; the output image size became 288x 280. In the second level, we applied double convolution with a 3×3 filter in 128 channels and max-pooling; the output image size became 144x140. In the third level, we applied double convolution with a 3×3 filter in 256 channels followed by max-pooling; the output image size became 72x70. In the fourth level, we applied convolution with a 3×3 filter in three 512 channels followed by max-pooling; the output image size became 36x35. In the last level, we applied three 512 convolutions with a 3×3 filter as shown in table 1 for the encoding stage.

All layers were used within the rectification (ReLU (Krizhevsky et al [105])) non-linearity. In the decoding stage, we followed the previous stages of encoding, but in the opposite direction, where we applied the operator Up-sampling with stride equal to 2, to restore the original dimensions of the image.

In some high-performance networks, they used large receptive fields in the first conv layers (e.g. 11×11 with stride 4 in (Krizhevsky et al [105]), or 7×7 with stride 2 in (Zeiler & Fergus [106]; Sermanet et al [107])), our network configurations are characterized by the fact that we used small 3 × 3 receptive fields in whole net, which helped us to integrate three non-linear rectification layers and makes the decision more effective. In addition, some networks used a lower size of convolution Ciresan et al. [108] but their network depth was less compared to ours.

The hybrid U-net consisting of the network VGG 16 gave an excellent representation of the data compared to different networks [100] with a U-net network to give us the maximum positives.

Our network structure is shown in Figure 3.7. The main advantages of the network are the perfect representation of data.

Deep learning specialists disagree about how many layers the network needs to be deep. Through [109] the depth of the network has not been confirmed, which enables us to consider it deep.

3.5 RESIDUAL UNIT

Research has confirmed that increasing the depth of the network improves its performance, but not all systems are easy to optimize. However, increasing the depth creates a degradation problem, which negatively affects network performance.

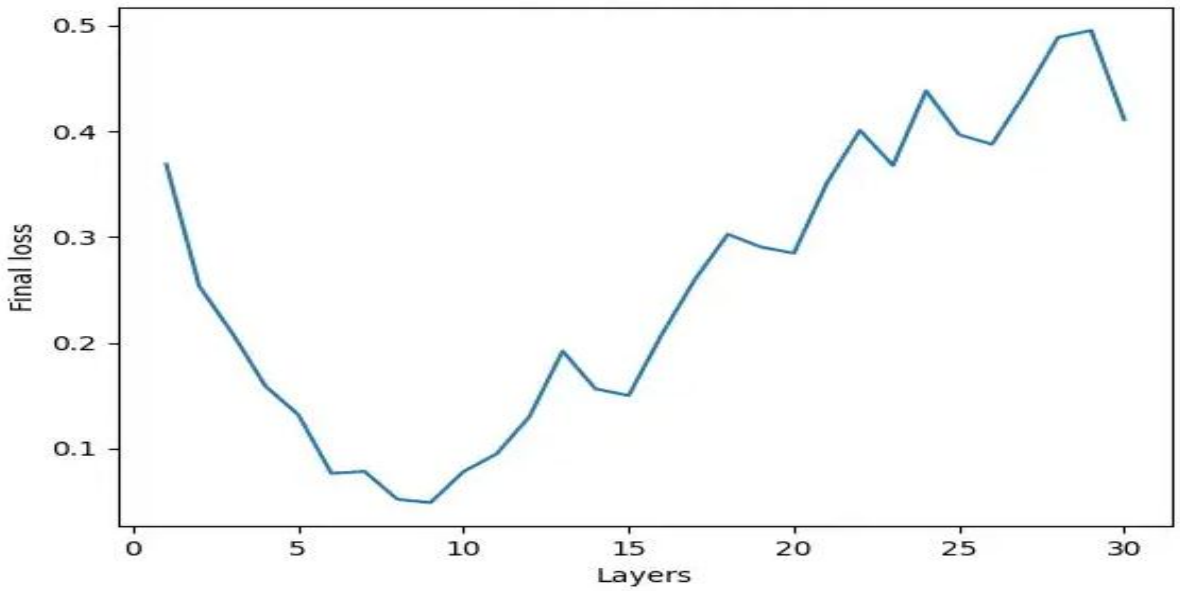


Figure 3.9: Neural network loss increases with number of layers

Before we continue, we should mention two basic concepts in deep learning: underfitting and overfitting. We note that underfitting occurs when the model cannot obtain a sufficiently low loss score on the training data, causing the model to be unable to learn the basic parameters in the training set.

We say we have overfitting when our model can train well.

For our model to be ideal, it must have two basic points:

- Increasing or decreasing convolutional layers allows the model to be controlled in terms of underfit or overfit.
- The value of the training loss should be the lowest, and the error gap between training and testing should be small and reasonable.

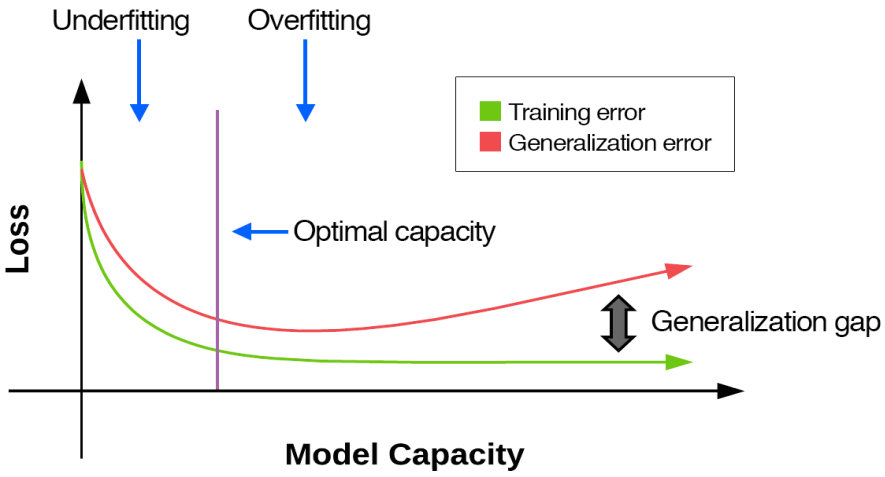


Figure 3.10: The relationship between underfitting (left) and overfitting (right). The generalization gap stabilizes in the underfitting condition. An increase in the generalization loss means that overfitting occurs.

He et al. [101] gave us a model with no higher training error and solved the degradation problem. This technique allows the layer to fit with the residual mapping layer.
It became possible to train very deep networks that have a depth of 50-100 or more. The residual module is the main component on which ResNet is based. Pooling layers are used sparingly in Resnet.

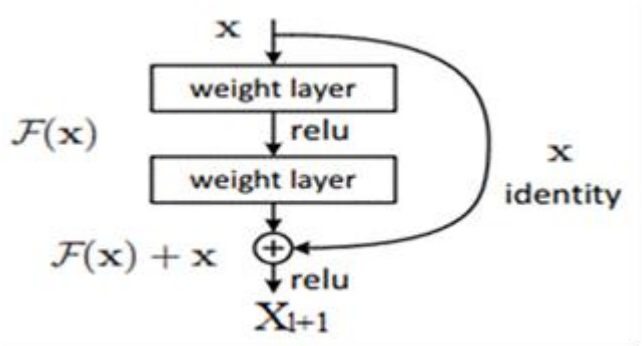


Figure 3.11: Residual learning blocks.

The residual network is composed from series of blocks, as it generally appears in the following equations:

$$P_l = t(x_l) + F(x_l + w_l) \qquad (3.5)$$

$$x_{l+1} = f(P_l)$$

Where $F(\cdot)$ is the residual function, $f(P_l)$ is function activation. x_l and X_{l+1} the input and output of the l-th residual unit, $t(x_l)$ is the identity mapping function $t(x_l) = x_l$.

Residual learning blocks are based on identity mappings input X is added to the output $F(x)$ of a series of operations, therefore, it is called 'residual'.

When we made these additions, it proved that the model becomes able to learn more and faster. In most networks, if the learning rate is high, it is difficult to converge the network such as the VGG net or Alex net, unlike the Residual network which depends on residual modules has the ability to allow this because it depends on the residual modules.

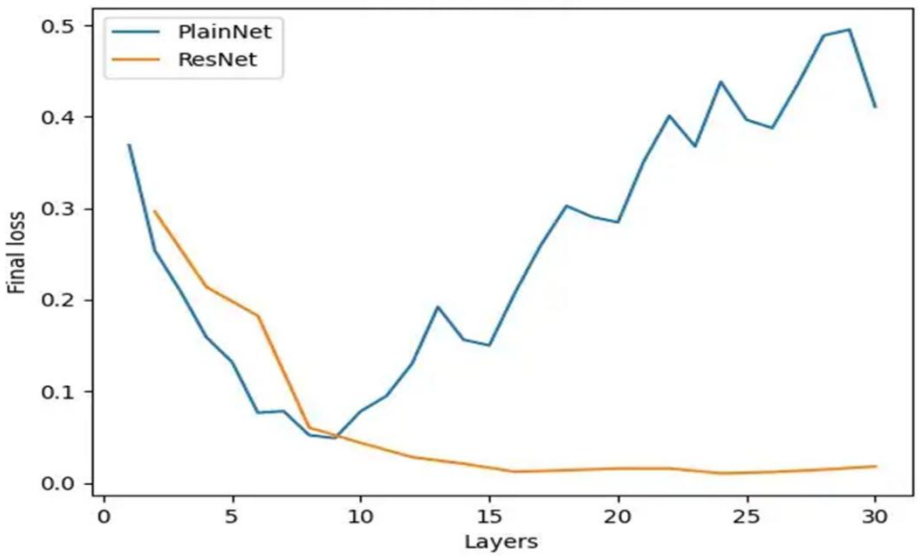


Figure 3.12: ResNets does not have degradation problem

The block residual learning allowed to increase in the training speed of the model and improve its effectiveness. The processing mentioned in Figure 3.11 could solve the degradation problem perfectly.

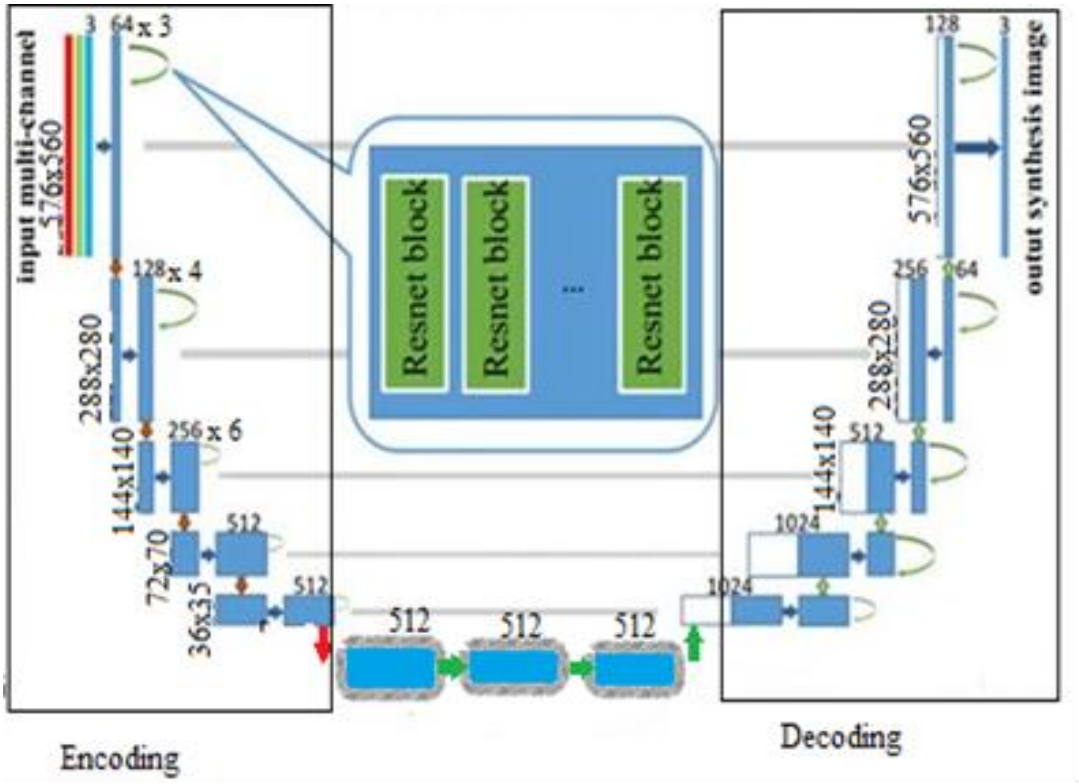


Figure 3.13: The architecture of Resnet 34 +U-net. [122]

We have proposed this model, Resnet 34 + U-net, for semantic segmentation which consists of strengths of both residual neural network and U-net. The main feature of this combination is that it facilitates the work of the network by the residual unit; residual connections with low and high levels advance the diffusion of information without degradation giving us minimal parameters and better performance.

As shown in Figure 3.13, The network has two stages: Encoding and decoding. In the first stage, we encoded our image input into a consolidated representation. In the second stage, we retrieved the representations to classify pixels, in other words, semantic segmentation. Encoding and decoding were built with residual units and identity mapping plus convolution blocks which included batch normalization layer and ReLU activation.

Table 3.3: Construction of Resnet 34+U-net [121]

	Repetition	*#*	*Input image*			*Output*			*layer*	*stride*	*kernel*	*in*	*out*	*Add*	*Param*
Encoding		1	576	560	3	576	560	64	Conv3-64	2	7x7	3	64	-	9472
			576	560	64	288	280	64	maxpool	2	2x2	64	64	-	0
	X 2	2	288	280	64	288	280	64	Conv3-64	1	3x3	64	64	False	36928
		3	288	280	64	288	280	64	Conv3-64	1	3x3	64	64	True	36928
			288	280	64	144	140	64	maxpool	2	2x2	64	64	-	0
	X 3	4	144	140	64	144	140	128	Conv3-128	1	3x3	64	128	False	73856
		5	144	140	128	144	140	128	Conv3-128	1	3x3	128	128	False	147584
		6	144	140	128	144	140	128	Conv3-128	1	3x3	128	128	True	73856
			144	140	128	72	70	128	maxpool	2	2x2	128	128	-	0
	X 5	6	72	70	128	72	70	256	Conv3-256	1	3x3	128	256	False	295168
		7	72	70	256	72	70	256	Conv3-256	1	3x3	256	256	False	590080
		8	72	70	256	72	70	256	Conv3-256	1	3x3	256	256	True	590080
			72	70	256	36	35	256	maxpool	2	2x2	256	256	-	0
	X 2	9	36	35	256	36	35	512	Conv3-512	1	3x3	512	512	False	1180160
		10	36	35	512	36	35	512	Conv3-512	1	3x3	512	512	False	2359808
Decoding			36	35	512	72	70	512	Up-sampling	2	2x2	512	512	-	0
		11	72	70	512	72	70	256	Conv3-256	1	3x3	512	256	False	1769728
		12	72	70	256	72	70	256	Conv3-256	1	3x3	256	256	True	590080
			72	70	256	144	140	256	Up-sampling	2	2x2	256	256	-	0
		13	144	140	256	144	140	128	Conv3-128	1	3x3	256	128	False	589952
		14	144	140	128	144	140	128	Conv3-128	1	3x3	128	128	True	147584
			144	140	128	288	280	128	Up-sampling	2	2x2	128	128	-	0
		15	288	280	128	288	280	64	Conv3-64	1	3x3	128	64	False	147520
		16	288	280	64	288	280	64	Conv3-64	1	3x3	64	64	True	36928
			288	280	64	576	560	64	Up-sampling	2	2x2	64	64	-	0
		17	576	560	64	576	560	64	Conv3-64	1	3x3	64	64	False	73792
		18	576	560	64	576	560	64	Conv3-64	1	3x3	64	64	True	36928
	#		576	560	64	576	560	1	-	-	-	64	1	-	65
Total params															24,899,393

First, in the encoding path, we used the maxpool operator with a stride of two, to decrease the feature map by half. Moreover, for decoding, we applied Up-sampling on the feature and concatenation with the feature from the encoding path. Finally, we used the convolution layer and Sigmod activation to get the required segmentation. The details are shown in the previous table 3.3.

We have used 32 convolutional layers in the encoding path, as the Resnet 34 contains, without 2 convolutional layers (fully connected and a 1 × 1 convolution and a Sigmod activation layer) that was used during the classification by Resnet 34.

Moreover, table 3.4 contains Hyperparameters for training our models.

Table 3.4: Hyperparameters used to train the U-net model

Parameter	value
Optimizer	Adam
Test split	0.1
Learning rate	10-4
Regularization (Dropout)	0.3
Epochs	30
Batch Size	4
Batch normalisation	True

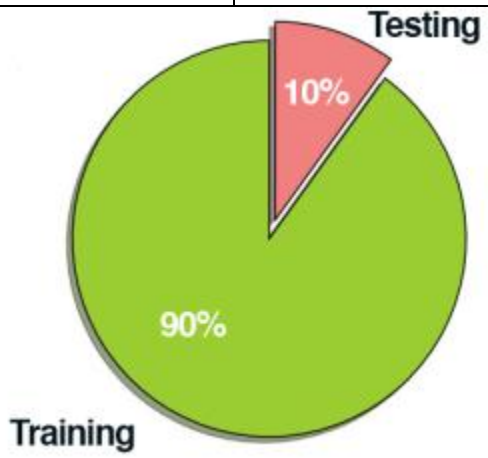


Figure 3.14: Training and testing data splits

We split data using the instructions provided in the scikit-earn library.

Dropout enables us to avoid overfitting by increasing testing accuracy, which sometimes causes a decrease in accuracy.

Where the connection is randomly disconnected between two layers according to the value that we specify, in our case we used 0.5. This step allows us to help the model generalize as it ensures that there is no single node quality.

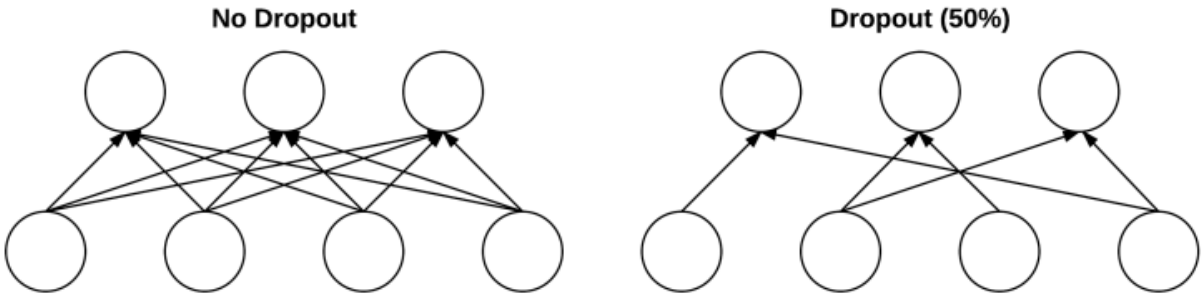


Figure 3.15: The dropout effect appears on the networking and communication layers in between.

Batch Normalization is used to speed up the training of a Deep Network by normalizing the activation process before moving on to the next layer in the network.

It has been shown that Batch Normalization reduces training times for a neural network, which can help stabilize the network. When using it we can finally notice the stability of the curves and the value of the loss more.

If we have x to be our mini-batch of activation, then we can calculate the normalized $\hat{x}$ through the following equation:

$$\widehat{x_i} = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}} \qquad (3.6)$$

While training, we compute the μ_β and σ_β over each mini-batch β, where:

$$\mu_\beta = \frac{1}{M}\sum_{i=1}^{m} x_i \qquad \sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2 \qquad (3.7)$$

We set ε equal to a small positive value such as 1e-7 to avoid taking the square root of zero.

Applying this equation implies that the activations leaving a batch normalization layer will have approximately zero mean and unit variance.

We used hyperparameters on these which gave us what works the best. We also used a third class called the validation set that we extracted from our training data to help us define and fine-tune our hyperparameters. Then we moved to the stage of collecting accurate results from the test data.

In the DRIVE database, we have the training dataset as well as the test set, independent of each other, we have two models of the test set, and each set contains manually segmented images of the vessels.

One of the most important points that the segmentation model must have is the total separation between the training and the test dataset so that the model becomes fair and has not actually been seen or learned from the test examples, as we use the test set only to evaluate the model.

3.6 IMPLEMENTATION DETAILS

We simulated our experiments with Python 3.6. We also used environments Tensorflow 2.0.0, Keras 2.2.4, and NumPy libraries to operate the network. Computer configurations were NVIDIA GeForce GTX 1080 Ti GPU with 11 GB VRam, CPU i7-7700K @ 4.20GHz, and 16 GB RAM, using Win10/64 bit. All the images were taken as 576x560 size in the DRIVE database, and we took 688x592 with STARE without the operation resizing to not lose important details in the images.

Conclusion

In this chapter we applied the two networks VGG-16 and Resnet-34 and hybridized each network with U-net to segment the retinal blood vessels, we mentioned the reason why we chose to use these two networks to do the segmentation. The proposed two networks allowed making predictions for retinal images, where the VGG-16 through the encoding stage contributed to making a distinguished representation of blood vessels through their special structure and by their depth and the number of filters that were used to extract the important features for segmentation. Also Resnet-34, which enabled us to extract blood vessels, avoiding the occurrence of overfitting and with high accuracy, as we will mention and evaluate the results we obtained and compare them with the different contributions in this field in the next chapter.

Chapter 4: Results and discussion

4. Results and discussion

Our models were applied to DRIVE and STARE retinal image bases, results were compared with available ground truth to calculate the models' performance metrics. We relied in our comparisons on the contributions associated with the U-net that segmented the blood vessels of the retina to distinguish more of the work that we have done and what distinguishes the results that we obtained. Our models have been rated by Accuracy (Acc), Specificity (Spec), Sensitivity (Sens), Recall, and F1-Score.

Tables 4.2 and 4.3 provide performance measures. In addition, Tables 4.5 and 4.6 provide a clear comparison of our results with the results obtained in different contributions.

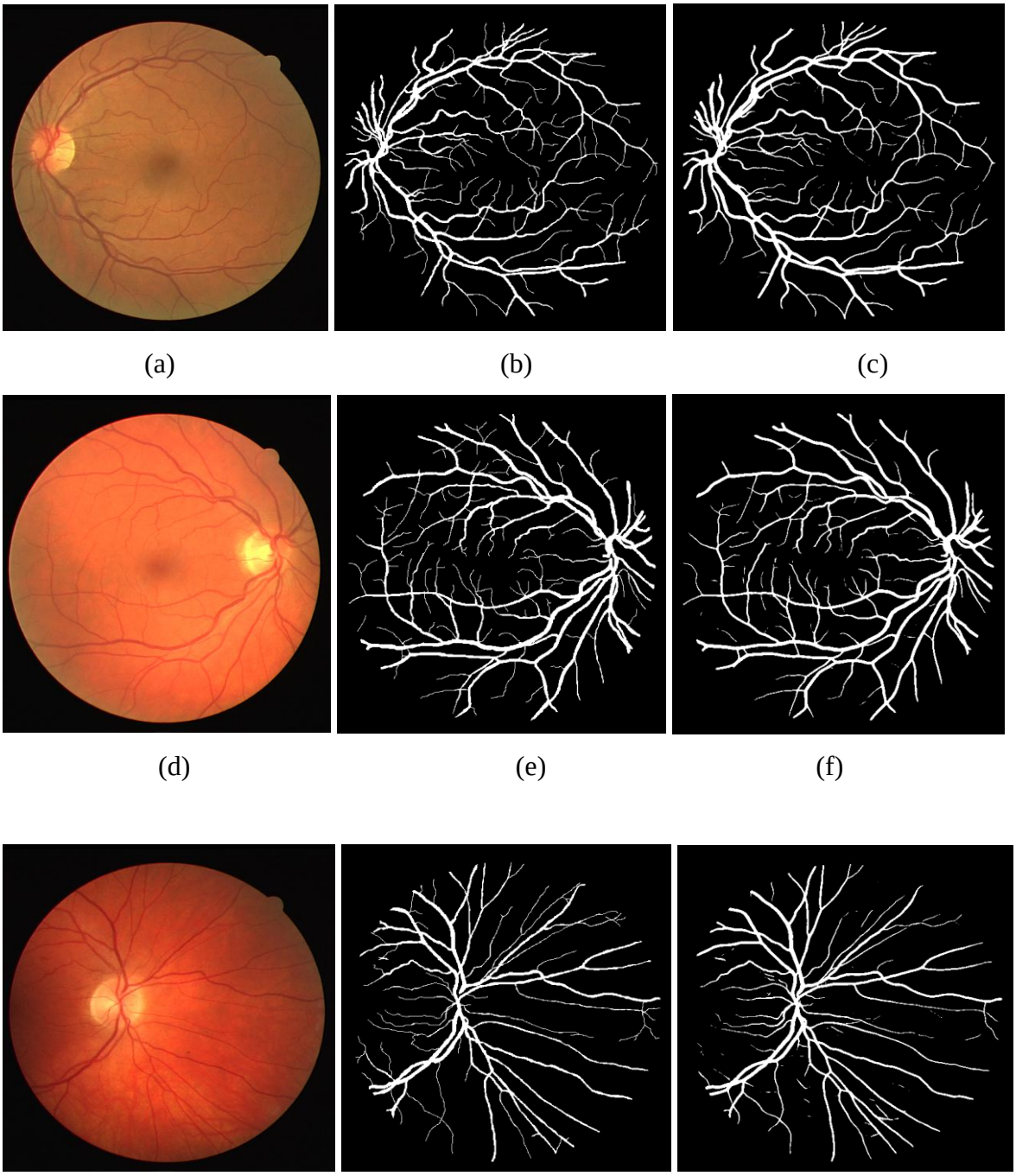
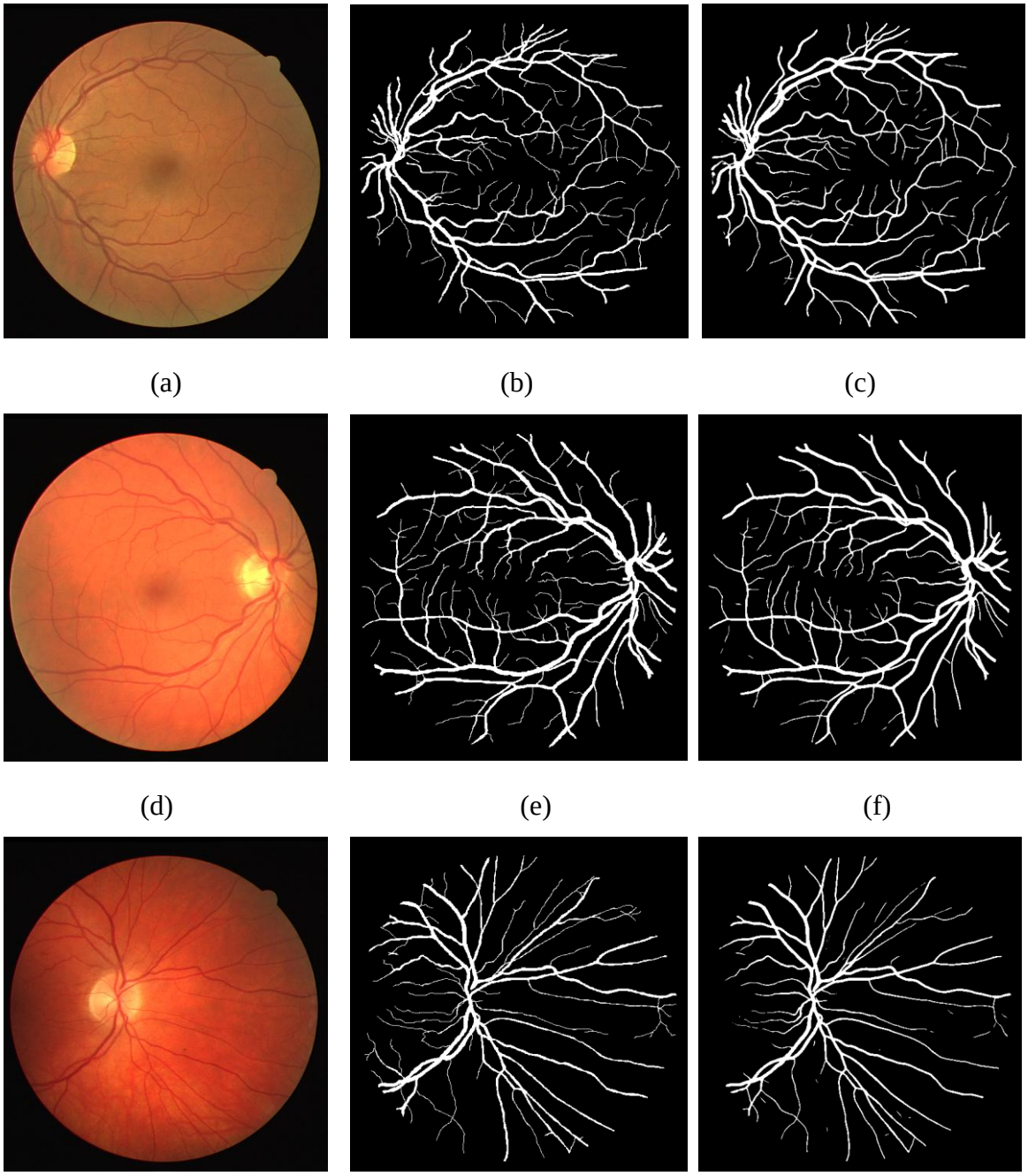


Figure 4.1: Test results on DRIVE VGG 16 +U-net. a, d, g: original images in DRIVE. b, e, h: Ground truths. C, f, i: segmented output for them. [121]



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.2: Test results on DRIVE Resnet 34 +U-net. a, d, g: original images in DRIVE. b, e, h: Ground truths. C, f, i: segmented output for them. [121-122]

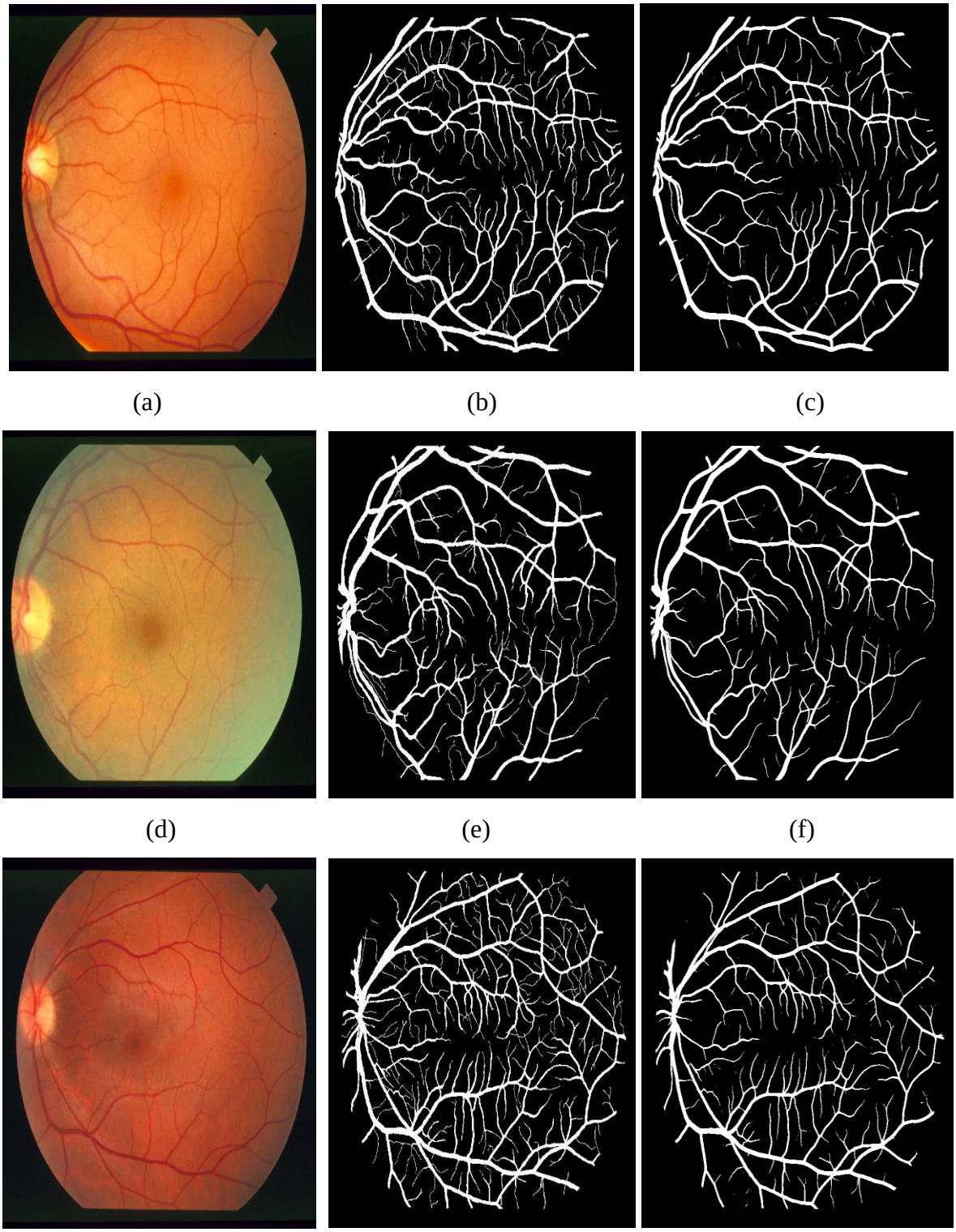
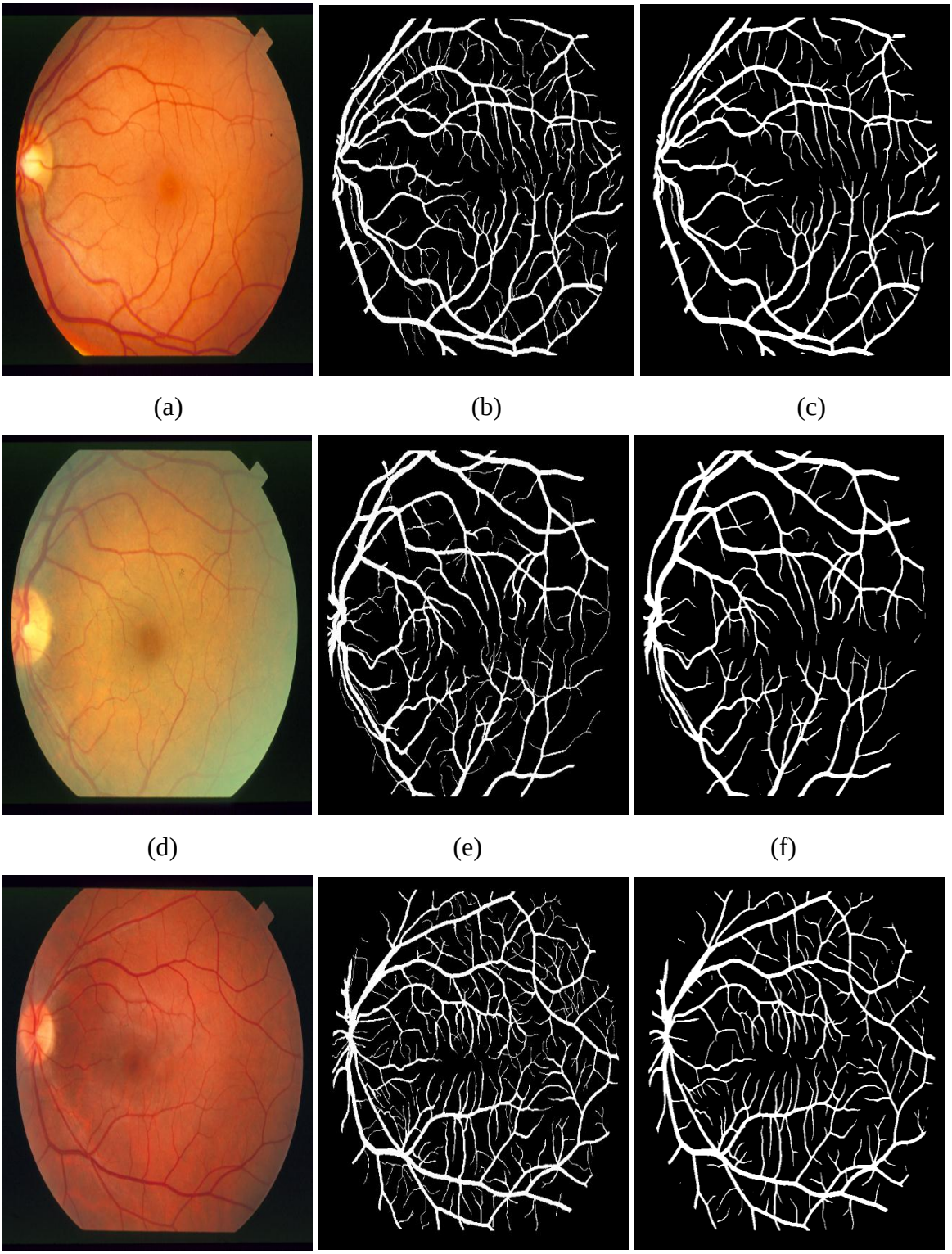


Figure 4.3: Test results on STARE VGG 16 +U-net. a, d, g: original images in STARE. b, e, h: Ground truths. C, f, i: segmented output for them. [121]



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.4: Test results on STARE Resnet 34 +U-net. a, d, g: original images in STARE. b, e, h: Ground truths. C, f, i: segmented output for them. [121-122]

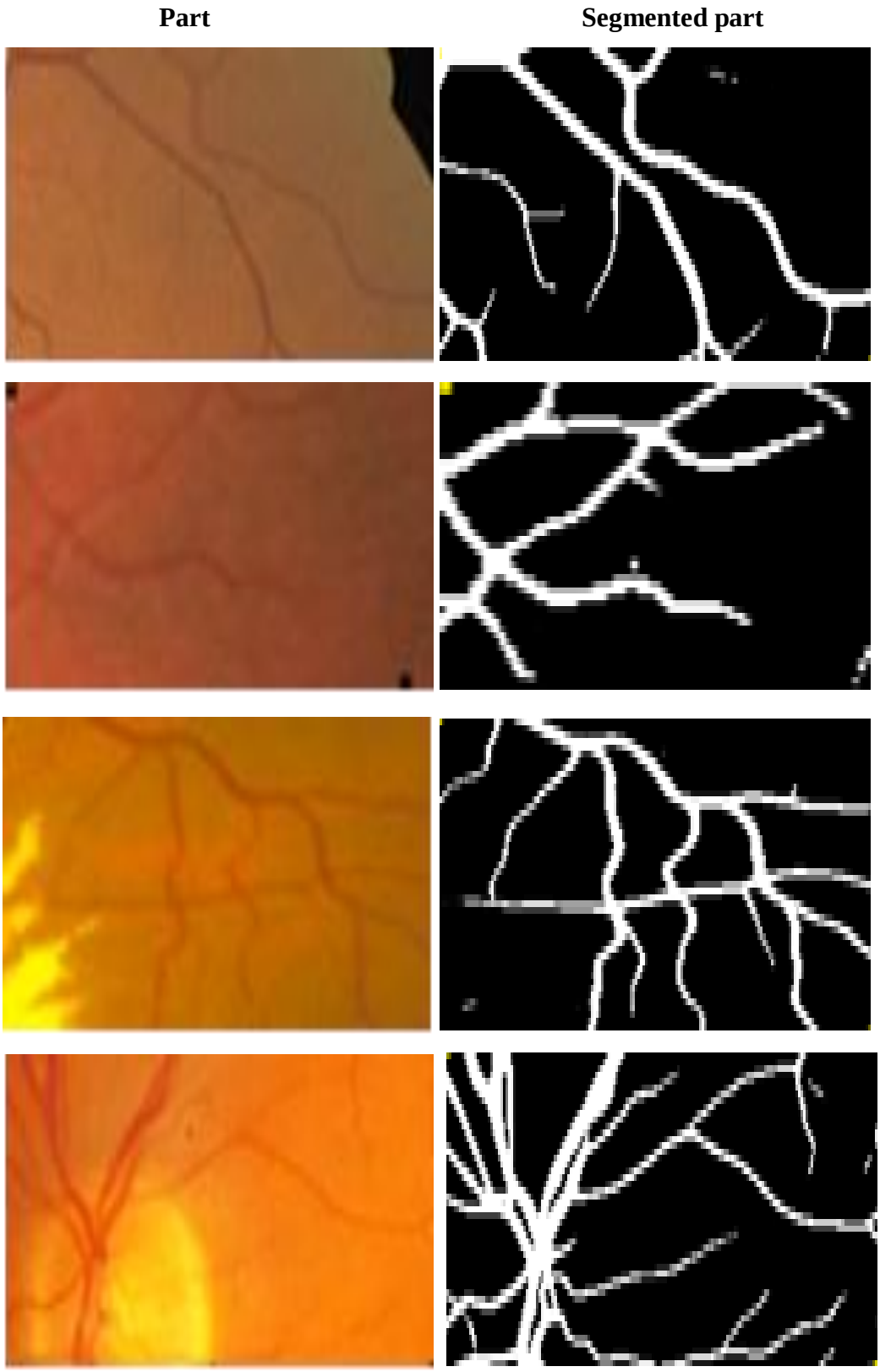


Figure 4.5: Parts of segmentation by VGG 16+U-net where we can see vessels with different sizes [121]

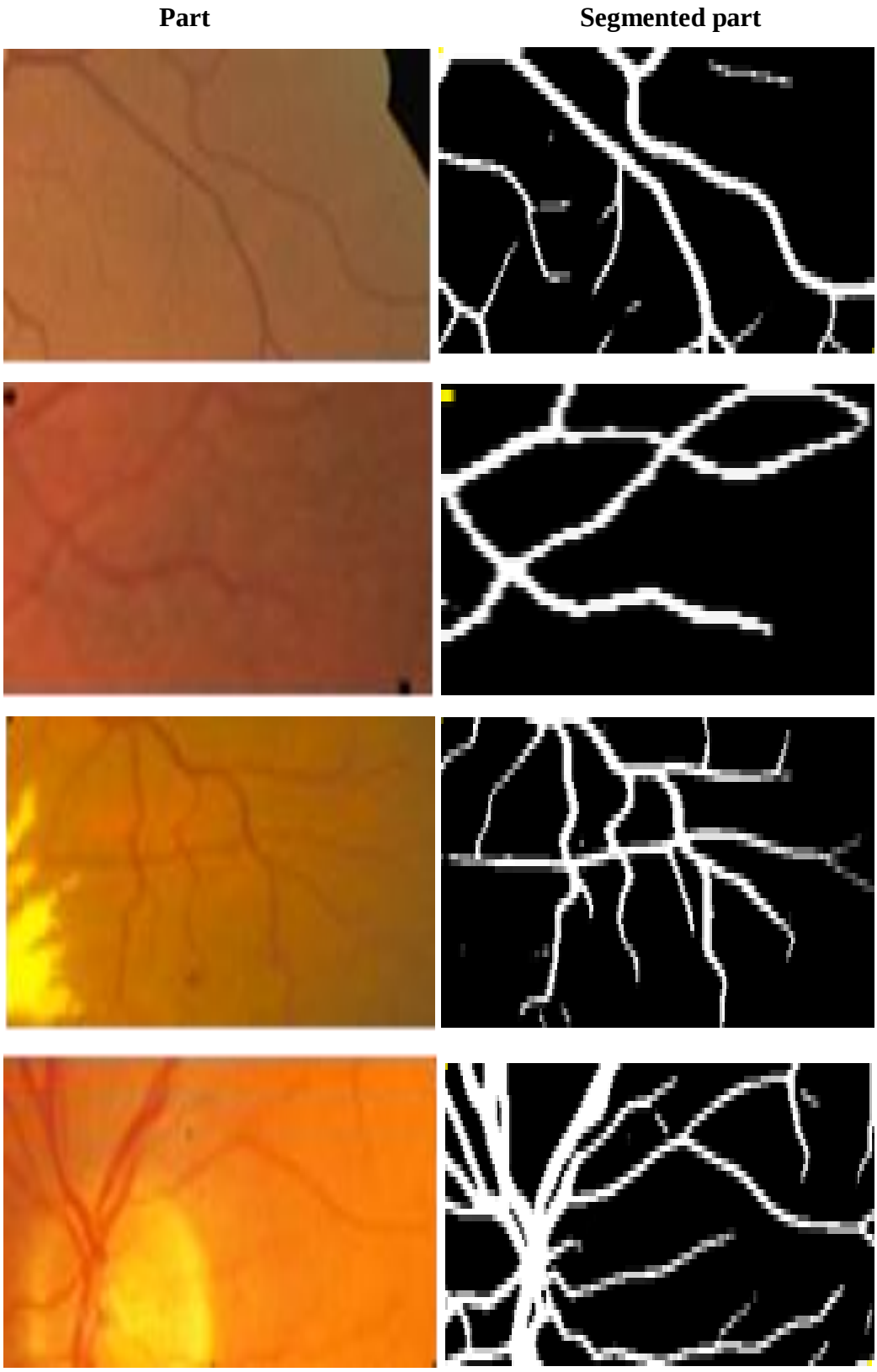


Figure 4.6: Parts of segmentation by Resnet 34+U-net where we can see vessels with different sizes. [121-122]

In this research paper, we focus on loss, accuracy, sensitivity, specificity, recall, and F1-Score to measure model performance. The loss function quantifies the error between the output of the algorithm and the given target value. Our goal is to make this value as minimal as possible. We used Loss Dice as the loss function:

$$\text{Loss Dice } (y,\acute{y}) = \frac{\mathbf{2}|\boldsymbol{y}\cap\boldsymbol{y'}|}{|\boldsymbol{y}|+|\boldsymbol{y'}|} \qquad (4.1)$$

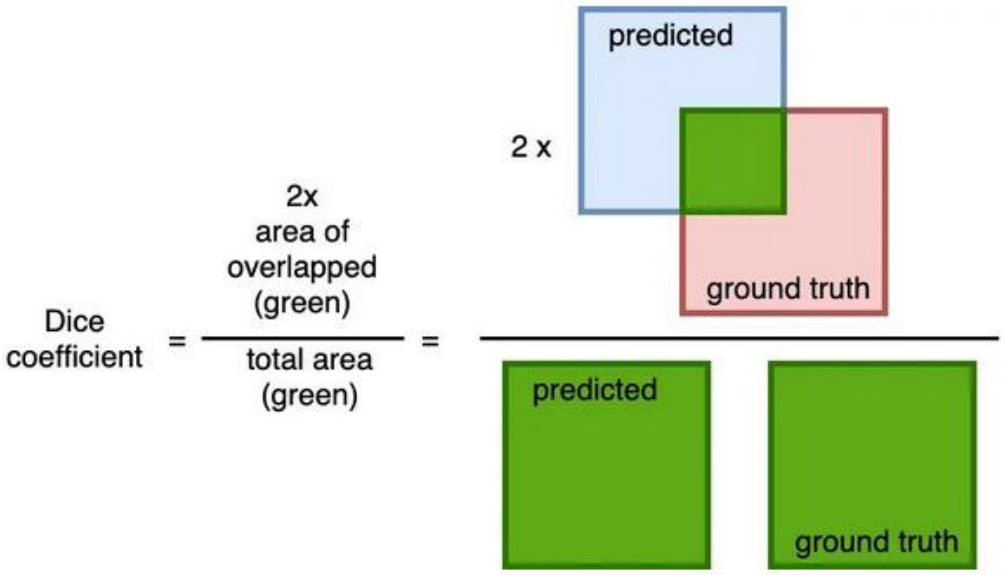


Figure 4.7: Diagram showing the Loss Dice method

We used the Adam (Adaptive Moment Estimation) optimization algorithm suggested by Kingma and Ba in 2014, which reduces the value of the error function to get accuracy, which is the value of convergence between predicted and actual value.

We used the confusion matrix, which is shown in Table 4.1.

Table 4.1: Confusion matrix

Actual Value	***Predicted Value***	
	True	False
True	(True Positive)	FN (False Negative)
False	FP (False Positive)	TN (True Negative)

We used also the next indicators as the metrics:

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (4.2)$$

$$Specificity = \frac{TN}{TN+FP} \quad (4.3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4.4)$$

$$F1-Score = \frac{2*TP}{2*TP+FP+FN} \quad (4.5)$$

True Positive (TP): the sample is positive and classified as well.

True negative (TN): the sample is negative and classified as well.

False positive (FP): the sample is actually negative, but it is classified as positive which is the opposite.

False negative (FN): the sample is actually positive but it was classified as negative which is the opposite.

Accuracy: gives us how many times it was positive, negative, or correct.

Recall: give us how many samples are chosen.

F1-Score: gives us the average symmetrical between recall and Specificity.

4.1 Subjective evaluation

Figures 4.1-5.4 show different predictions of the algorithms VGG 16 +U-net and Resnet 34+U-net for DRIVE and STARE datasets, respectively. In Figures 4.1-5.2 parts (a, d, g), the segmentation was very accurate, and a large similarity between the original image and segmented output was observed. The segmentation of Figure 4.3-5.4 parts (a, d, g) was relatively accurate and the boundaries of the vessels were clear. In Figure 4.1 part i, we noticed that the large vessels were clear but the small ones were not segmented enough to be more clear. We can improve these images to be more visible by processing them, but it is better if our model does this process automatically. In Figure 4.2 part f and Figure 4.3 part (c, i), the prediction result was accurate as we can see the high match between the original image and the segmented

output. We can see the great similarity between the parts in figure 4.5 and figure 4.6, where the vessels are clearly visible and close. The vessel network appearance was more noticeable in Resnet 34 + U-net compared to VGG16 + U-net, but the accuracy value in the VGG 16+ U-net model was higher because it classified the pixel more accurately distinguishing it from the background with less error. As for Resnet 34+U-net, some errors have occurred, where we noticed that the background pixel has been classified as a vessel pixel, which justifies the frequent appearance of vessels in this model especially the image in the third line, in Figure 4.6. Moreover, most of the parts in the fundus retina were segmented with high accuracy, and some small blood vessels were accurately segmented as well, and this indicates the reliability of our methods for the segmentation of blood vessels in the retina.

Table 4.2: Performance of the VGG 16+U-net method on the DRIVE and STARE databases [121]

Data	***DRIVE***					***STARE***				
Image	Spec	Sens	Acc	Recall	F1-Score	Spec	Sens	Acc	Recall	F1-Score
1	0.978	0.874	0.968	0.987	0.982	0.9685	0.8646	0.9621	0.990	0.979
2	0.985	0.843	0.970	0.982	0.983	0.9843	0.6787	0.9689	0.983	0.983
3	0.991	0.669	0.959	0.964	0.977	0.9815	0.7779	0.9725	0.989	0.985
4	0.987	0.805	0.970	0.980	0.983	0.9724	0.6948	0.9564	0.981	0.976
5	0.990	0.745	0.967	0.974	0.982	0.9609	0.8222	0.9513	0.986	0.973
6	0.991	0.717	0.964	0.970	0.980	0.9538	0.9526	0.9538	0.997	0.975
7	0.993	0.725	0.968	0.972	0.982	0.9384	0.9729	0.9405	0.998	0.967
8	0.990	0.724	0.967	0.974	0.982	0.9369	0.9819	0.9395	0.999	0.966
9	0.990	0.752	0.970	0.978	0.984	0.9478	0.9714	0.9492	0.998	0.972
10	0.985	0.795	0.969	0.981	0.983	0.9257	0.9815	0.9290	0.999	0.960
11	0.985	0.793	0.968	0.979	0.982	0.9493	0.9599	0.9499	0.997	0.972
12	0.986	0.820	0.972	0.983	0.984	0.9493	0.9701	0.9505	0.998	0.973
13	0.987	0.785	0.967	0.976	0.982	0.9395	0.9710	0.9417	0.997	0.967
14	0.983	0.851	0.973	0.986	0.985	0.9438	0.9627	0.9451	0.997	0.969
15	0.986	0.826	0.974	0.986	0.986	0.9507	0.9435	0.9503	0.995	0.972
16	0.985	0.825	0.970	0.982	0.984	0.9396	0.9438	0.9399	0.994	0.966
17	0.986	0.783	0.969	0.980	0.983	0.9491	0.9341	0.9480	0.994	0.971
18	0.978	0.862	0.969	0.988	0.983	0.9798	0.8572	0.9753	0.979	0.987
19	0.982	0.912	0.977	0.991	0.987	0.9758	0.8765	0.9727	0.996	0.985
20	0.977	0.869	0.969	0.989	0.983	0.9580	0.8811	0.9541	0.993	0.975
Mean	**0.9862**	**0.7992**	**0.9695**	**0.980**	**0.983**	**0.9553**	**0.8999**	**0.9525**	**0.993**	**0.974**

The previous table 4 shows the performance of the various measures of the proposed vessel segmentation on STARE and DRIVE databases by the hybridization of VGG16 with U-net. The latter, table 4, gives us the results of Resnet 34+Unet performance.

Table 4.3: Performance of Resnet 34+U-net method on the DRIVE and STARE [121-122]

Data	***DRIVE***					***STARE***				
Image	Spec	Sens	Acc	Recall	F1-Score	Spec	Sens	Acc	Recall	F1-Score
1	0.977	0.876	0.968	0.987	0.982	0.951	0.918	0.949	0.994	0.972
2	0.986	0.831	0.970	0.980	0.983	0.975	0.735	0.963	0.985	0.980
3	0.986	0.749	0.962	0.972	0.979	0.976	0.796	0.968	0.990	0.983
4	0.989	0.761	0.968	0.976	0.982	0.736	0.736	0.952	0.983	0.974
5	0.990	0.726	0.966	0.972	0.981	0.950	0.869	0.944	0.989	0.969
6	0.990	0.715	0.963	0.969	0.979	0.933	0.985	0.936	0.999	0.965
7	0.994	0.691	0.966	0.969	0.981	0.918	0.987	0.922	0.9991	0.957
8	0.992	0.676	0.965	0.970	0.981	0.907	0.994	0.912	0.9996	0.951
9	0.990	0.740	0.970	0.977	0.984	0.923	0.992	0.927	0.9995	0.959
10	0.985	0.788	0.969	0.981	0.983	0.890	0.996	0.897	0.9997	0.942
11	0.984	0.786	0.967	0.979	0.982	0.923	0.988	0.927	0.9993	0.959
12	0.987	0.805	0.971	0.981	0.984	0.926	0.992	0.929	0.9995	0.961
13	0.987	0.771	0.965	0.975	0.981	0.915	0.989	0.920	0.9991	0.955
14	0.986	0.822	0.973	0.984	0.985	0.922	0.985	0.926	0.998	0.959
15	0.989	0.779	0.974	0.983	0.986	0.927	0.968	0.930	0.997	0.961
16	0.988	0.804	0.972	0.980	0.984	0.910	0.976	0.915	0.997	0.952
17	0.988	0.771	0.970	0.979	0.983	0.925	0.974	0.929	0.9979	0.960
18	0.982	0.854	0.972	0.987	0.984	0.970	0.919	0.968	0.9968	0.983
19	0.981	0.907	0.975	0.991	0.986	0.964	0.930	0.963	0.997	0.980
20	0.980	0.858	0.971	0.988	0.984	0.942	0.932	0.941	0.996	0.968
Mean	**0.987**	**0.785**	**0.969**	**0.979**	**0.983**	**0.924**	**0.933**	**0.936**	**0.996**	**0.964**

In addition to the previous steps, we have used different Loss Function with standard U-net as shown in the table 6.

Table 4.4: Performance with different loss function for DRIVE database

Loss function	***Specificity***	***Sensitivity***	***Accuracy***
Mean squared error	0.9909693	0.5516768	0.93760662
Loss Jaccard	0.9608514	0.79376398	0.9445924
Binary cross entropy	0.99046633	0.582055165	0.9538190879

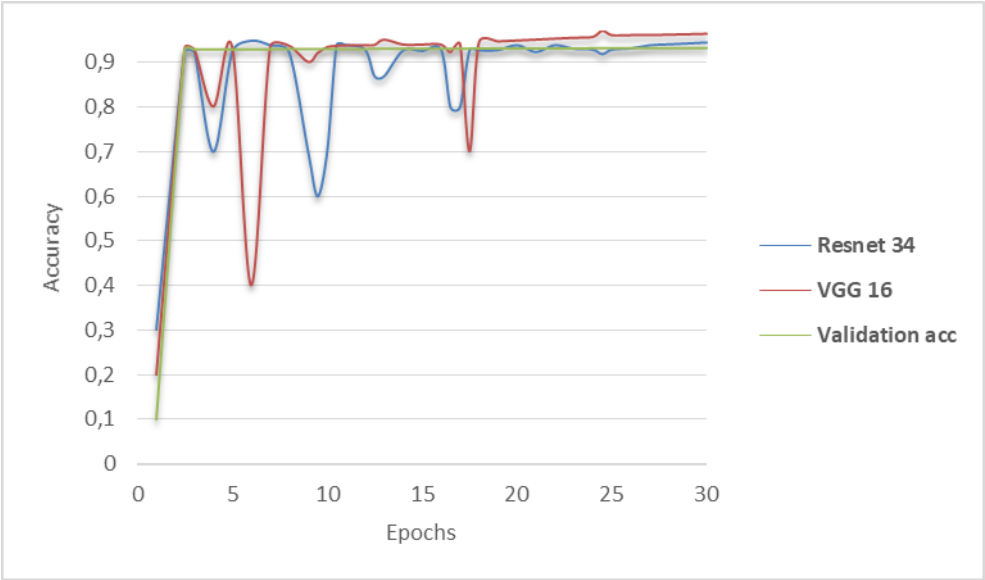


Figure 4.8: Curve of accuracy from VGG16+U-net and Resnet 34+U-net Applied in DRIVE database [121]

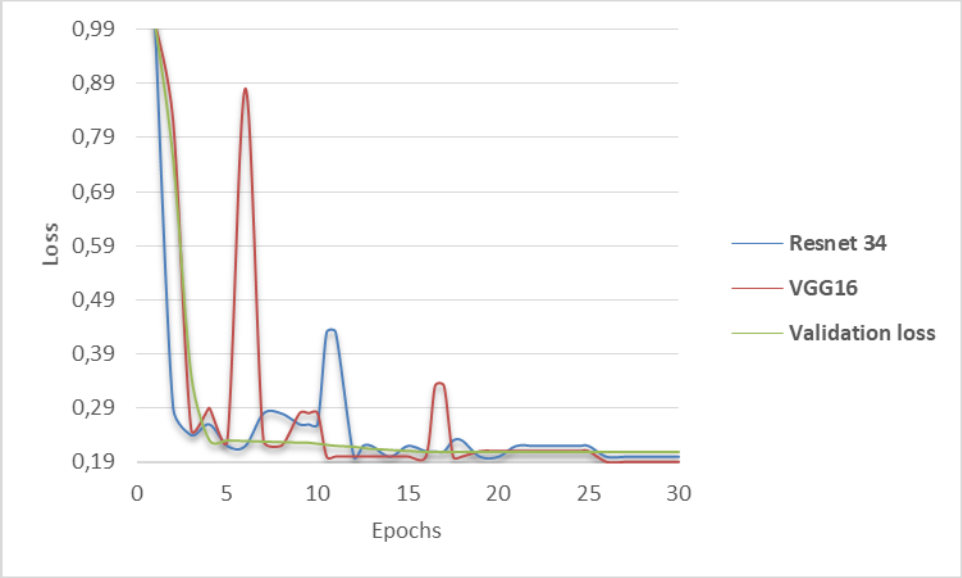


Figure 4.9: Curve of loss from VGG16+U-net and Resnet 34+U-net Applied in DRIVE database [121-122]

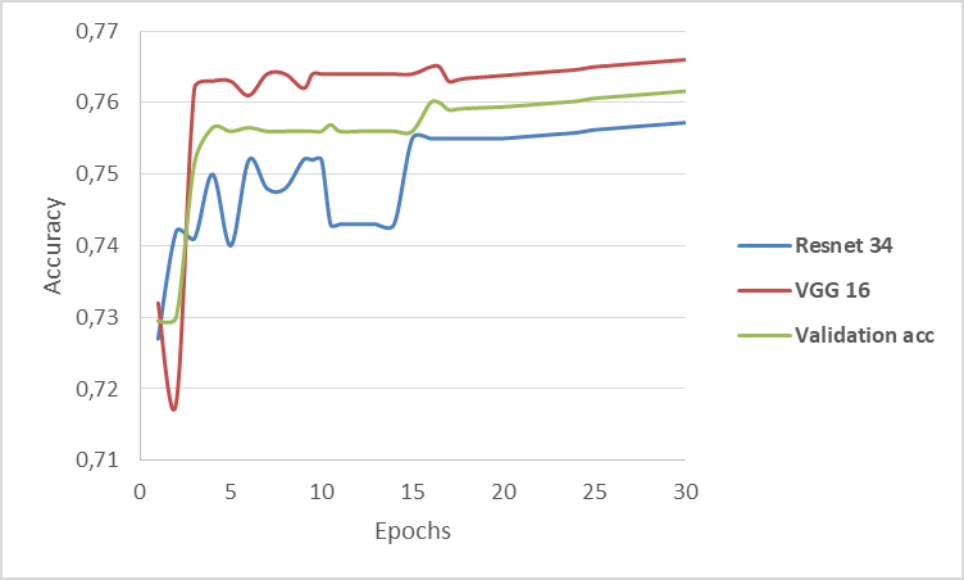


Figure 4.10: Curve of accuracy from VGG16+U-net and Resnet 34+U-net Applied in STARE database [121]

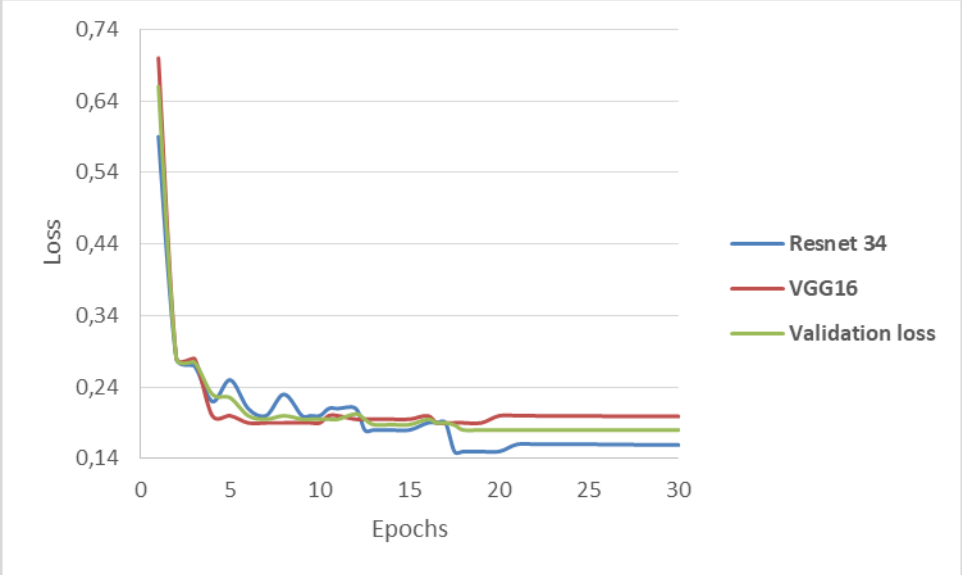


Figure 4.11: Curve of loss from VGG16+U-net and Resnet 34+U-net Applied in STARE database [121-122]

Figures 4.8 and 4.10 demonstrate the accuracy for DRIVE and STARE by VGG 16 + U-net and Resnet 34+U-net. Figures 4.9 and 4.11 show the loss for DRIVE and STARE by VGG 16 + U-net and Resnet 34. We can see the training process with 30 epochs as it gives us the model with a minimum loss value in Resnet 34+ U-net for the STARE dataset and maximum accuracy

with VGG 16+ U-net. The loss and accuracy for both models are faster convergent and stable with smoother movements especially model Resnet 34 which treats the degradation, vanishing, and exploding problems.

4.2 Comparisons

Table 4.5: Comparison of performance with methods on the DRIVE database [121]

Method	Acc	Spec	Rec	F1-S
U-net [86]	0.9531	0.9820	0.7537	0.8142
Residual U-net [86]	0.9531	0.8852	0.7726	0.8149
JL-U-net [110]	0.9542	0.9818	-	0.8102
Recurrent U-net [86]	0.9556	0.9780	0.7751	0.8155
R2 U-net [86]	0.9556	0.8589	0.7792	0.8171
LadderNet [85]	0.9561	0.8593	0.7856	0.8208
Att Unet [111]	0.9564	0.9789	-	0.8232
DUNet [80]	**0.9697**	0.8529	0.7963	0.8237
Iternet [112]	0.9575	0.9817	-	0.8218
Dynamic Unet [113]	0.9693	0.8284	0.8235	0.8259
VGG16+U-net	**0.9695**	**0.9862**	**0.9805**	**0.9833**
Resnet34+U-net	**0.9692**	**0.9870**	**0.9794**	**0.9832**

Table 4.6: Comparison of performance with methods on the STARE database [121]

Method	Acc	Spec	Rec	F1-S
U-net [86]	0.9690	0.8475	0.8270	0.8373
Residual U-net [86]	0.9700	0.8581	0.8203	0.8388
Recurrent U-net [86]	0.9706	0.8705	0.8108	0.8396
R2 U-net [86]	**0.9712**	0.8659	0.8298	0.8475
JL-Unet [110]	0.9612	0.9846	-	-
LadderNet [85]	0.9613	0.9804	-	0.7994
Att Unet [111]	0.9632	0.9816		0.8136
DUNet [80]	0.9729	0.8856	0.7428	0.8079
Iternet [112]	0.9644	0.9849	-	0.8157
SAUnet [114]	0.9521	**0.9930**	-	0.7736
VGG16+U-net	0.9525	0.9553	**0.9933**	**0.9742**
Resnet34+U-net	0.9363	0.9246	**0.9961**	**0.9694**

In Tables 4.5 and 4.6, we can see recent contributions on DRIVE and STARE datasets based on previous metrics. We conclude from the high-obtained accuracy, specificity, recall, and F1 score values that our models succeeded in classifying vessels from the background of the retina. Moreover, the F1-Score value had the highest margin in the two tables, and the accuracy in

Table 4.5 was the highest among all contributions after DUNet [80]. These results leave no doubt that our segmentation is ideal in most cases. Nevertheless, the implementation time was shorter compared to [113] where the execution time was reduced by 51 minutes on DRIVE images and 44 minutes on STARE, the device they used had 128 GB of RAM compared to the one we use of 16 GB of RAM. They also used two GeForce GTX 1080 Ti graphics cards, while we used only one 1080 RTX. However, our learning time was much less than the learning time of their model. as shown in Table 4.7.

Table 4.7: Comparison in training time

Method	***DRIVE***	***STARE***
A. Khanal and R. Estrada [113]	1:34 hours	-
proposed	00:43 hours	00:44

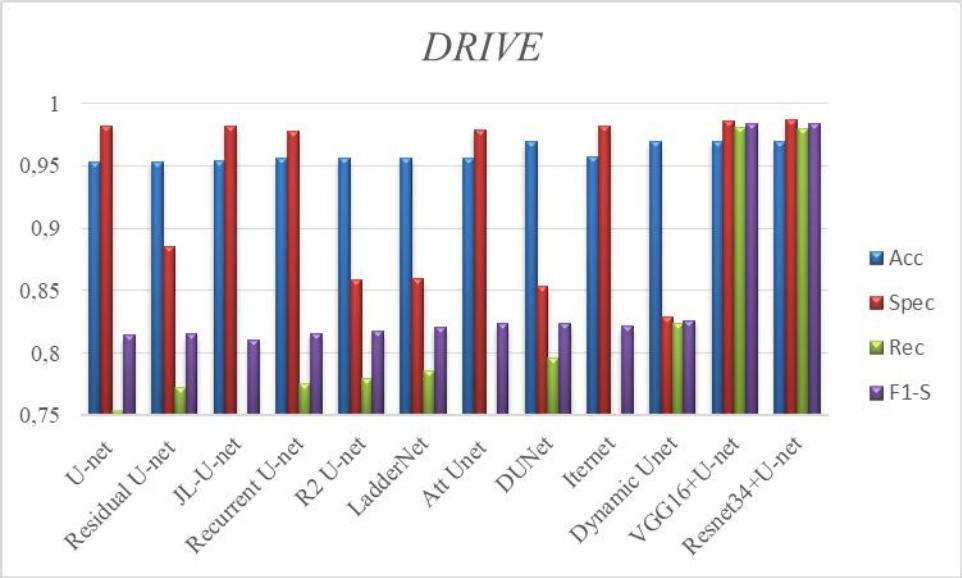


Figure 4.12: Performance in the metrics for DRIVE Database. [121]

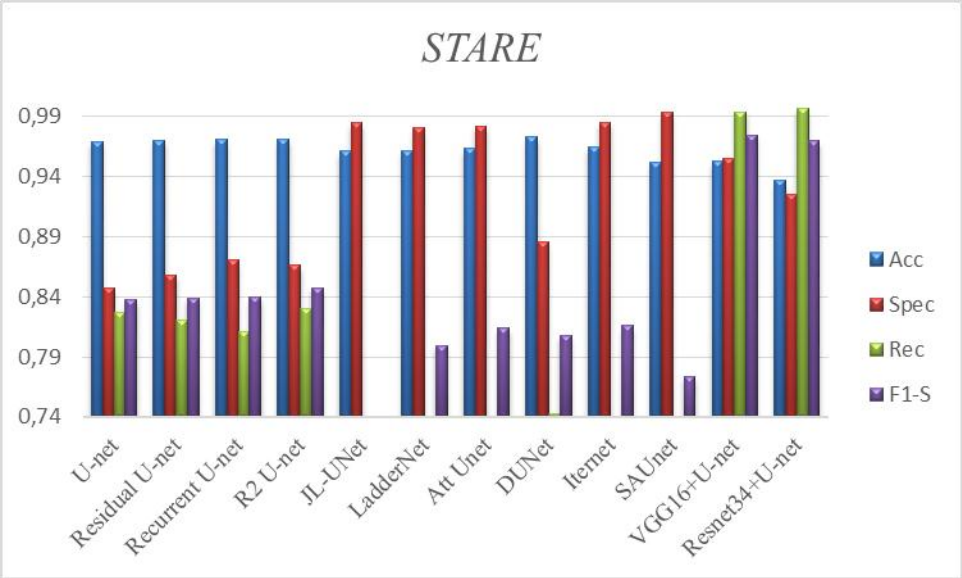


Figure 4.13: Performance in the metrics for STARE Database. [121]

In Figures 4.12-4.13, accuracy (ACC), specificity (Spec), Recall (Rec), and F1-score, it appears that the accuracy of the model Resnet 34 with U-net was a bit low for the STARE database but the value of the F1-score value was ideal, but for the DRIVE database, all measures ranked first compared to all the methods suggested with our models. Therefore, through all the results we obtained, our models have the potential to help doctors to segment retinal vessels and diagnose related diseases.

Conclusion

In this chapter, after applying both VGG-16 and Resnet-34 methods in the U-net model to segment retinal vessels from DRIVE and STARE databases, the segmentation results were excellent, as the blood vessels appeared clearly and without abnormalities in both methods. We used approved measures to measure the quality of the segmentation method, which proved the effectiveness of the two models in the segmentation process. Both models outperformed, in most measures, the methods used before for vessel segmentation, especially in the field of deep learning and the use of convolutional neural networks, as shown in the previous comparison tables.

General Conclusion

This thesis focused on the study of deep learning and its use for the segmentation of retinal blood vessels. The basic idea is to add two new U-Net networks depending on the original U-net by using VGG 16 and Resnet 34. We transformed the basic principle of image classification of these two deep convolutional networks into their segmentation and applied it to the vessels of the retina. The result of the first hybridization, VGG 16+U-net, was vivid evidence of the correctness of the method compared to the rest. This is the first contribution to vessel segmentation of the retina. In case there was a problem of degradation, we have shown that we can overcome it by applying the second hybridization between Resnet with 34 layers and U-net. The training period was the shortest with the highest performance compared to any similar study. Training efficiency can be improved in terms of time and accuracy by using dropout between convolutional layers and by normalizing the input data.

The use of pre-processing of images leads to an improvement in the results, because the images contain some of the microvessels and the details of the vascular tree, as they become clearer and enhanced, allowing us to extract more features that raise the value of segmentation accuracy.

This process gave the most competitive and accurate results in the field of blood vessel segmentation. For future research, we can use more architectures such as AlexNet and LeNet as encoders – decoders to get more alternative models. We will work to enhance and develop these algorithms to treat all cases and increase their accuracy and application in the medical field in the future.

Reference

[1] M. B. Wankhade. (2016) Analysis of disease using retinal blood vessels detection. Int. J. Eng. Comput. Sci., vol. 5, no. 12, pp. 19644–19647.

[2] T. Laibacher, T. Weyde, and S. Jalali. (2018) M2U-Net: Effective and efficient retinal vessel segmentation for resource cons-trained environments. arXiv:1811.07738. [Online]. Available: https://arxiv.org/abs/1811. 07738

[3] T. Y. Wong and P. Mitchell. (2007) The eye in hypertension. Lancet, vol. 369,no. 9559, pp. 425–435.

[4] https://www.researchgate.net/figure/Measures-of-retinal-vessel-tortuosity-branching-angle-and-optimality-deviation-A_fig1_42345064

[5] M. D. Abramoff, M. K. Garvin, and M. Sonka. Retinal imaging and image analysis. IEEE transactions on medical imaging, 3:169–208, 2010.

[6] Blausen.com staff. Medical gallery of blausen medical 2014. WikiJournal of Medicine 1 (2), DOI: 10.15347/wjm/2014.010, ISSN: 2002-4436.

[7] https://healthjade.com/human-eye/

[8] https://www.iowaretina.com/having-retina-surgery/

[9] Y. A. Tolias and S. M. Panas. (1998) "A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering," IEEE TMI, vol. 17, no. 2, pp. 263–273

[10] T. Walter and J. Klein. (2001) "Segmentation of color fundus images of the human retina: Detection of the optic disc and the vascular tree using morphological techniques," Paper Presented at the International Symposium on Medical Data Analysis. Springer, 2000 pp. 282–287.

[11] J. V. Soares, J. J. Leandro, R. M. Cesar Jr, H. F. Jelinek, M. J. Cree, "Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification," Medical Imaging, IEEE Transactions on 25 (9) (2006) 1214–1222.

[12] Ana Maria Mendonca and Aurélio Campilho. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. IEEE Transactions on Medical Imaging, 25(9):1200{1213, 2006.

[13] Michal Sofka and Charles V. Stewart. Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures. IEEE Transactions on Medical Imaging, 25(12):1531{1546, 2006.

[14] M. M. Fraz, S. A. Barman, P. Remagnino, A. Hoppe, A. Basit, B. Uyyanonvara, A. R. Rudnicka, and C. G. Owen. An approach to localize the retinal blood vessels using bit planes and centerline detection. Computer Methods and Programs in Biomedicine, 108(2):600{616, 2012.

[15] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, M.Goldbaum, Detection of blood vessels in retinal images using two-dimensional matched filters, IEEE Transactions on Medical Imaging 8 (1989) 263–269.

[16] A.D. Hoover, V. Kouznetsova, M. Goldbaum, Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response, IEEE Transactions on Medical Imaging 19 (2000) 203–210.

[17] L. Gang, O. Chutatape, S.M. Krishnan, Detection and measurement of retinal vessels in fundus images using amplitude modified second-order Gaussian filter, IEEE

[18] L. Sukkaew, B. Uyyanonvara, S.A. Barman, A. Fielder, K.Cocker, Automatic extraction of the structure of the retinal blood vessel network of premature infants, Journal of the Medical Association of Thailand 90 (2007) 1780–1792.

[19] C. Yao, H.-j. Chen, Automated retinal blood vessels segmentation based on simplified PCNN and fast 2D-Otsu algorithm, Journal of Central South University of Technology 16 (2009) 640–646.

[20] M. Amin, H. Yan, High speed detection of retinal blood vessels in fundus image using phase congruency, Soft Computing – A Fusion of Foundations, Methodologies and Applications (2010) 1–14.

[21] S. Chaudhuri, S. Chatterjee, and N. Katz. (1989) Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Trans. Med.Imag., vol. 8, no. 3, pp. 263–269.

[22] X. Y. Jiang and D. Mojon. (2003) Adaptive local thresholding by verification based multithreshold probing with application to vessel detection in retinal images. IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 1,pp. 131–137.

[23] L. Gang, O. Chutatape, and S. M. Krishnan. (2002) Detection and measurement of retinal vessels in fundus images using amplitude modified second-order Gaussian filter. IEEE Trans. Biomed. Eng., vol. 49, no. 2, pp. 168–172.

[24] B. Zhang, L. Zhang, and L. Zhang. (2010) Retinal vessel extraction by matched filter with first-order derivative of Gaussian. Comput. Biol. Med., vol. 40, no. 4, pp. 438–445.

[25] Lam B.S., Gao Y., Liew A.W. (2010): General retinal vessel segmentation using regularization based multiconcavity modeling. IEEE Transactions of Medical Imaging, vol.29, pp.1369-1381.

[26] Al-Diri B., Hunter A., Steel D. (2009): An active contour model for segmenting and measuring retinal vessels. IEEE Transactions of Medical Imaging, vol.38(9), pp.1488-1497.

[27] F. Zana and J. C. Klein. (2001) Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. IEEE Trans. Image Process, vol. 10, no. 7, pp. 1010–1019.

[28] L. Espona, M. J. Carreira, and M. G. Penedo. (2008) Retinal vessel tree segmentation using a deformable contour model. Proc. 19th Int. Conf. Pattern Recognit., Aug. 2008, pp. 1–4.

[29] O. Chutatape, L. Zheng, S. Krishnan. (1998) Retinal blood vessel detection and tracking by matched Gaussian and kalman filters. in: Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual Int Conf of the IEEE, Vol.6, IEEE, 1998, pp. 3144–3149.

[30] U. T. Nguyen, A. Bhuiyan, L. A. Park, K. Ramamohanarao. (2013) An effective retinal blood vessel segmentation method using multi-scale line detection. Pattern recognition 46 (3) 703–715.

[31] Fraz M.M., Remagnino P., Hoppe A., Uyyanonvara B., Rudnicka A., Owen C.,Barman S. (2012): Blood vessel segmentation methodologies in retinal images –A survey. Computer Methods and Programs in Biomedicine, vol.108, pp. 407- 433.

[32] Mookiah M.R., Acharya U.R., Chua C.K., Lim C.M., Laude A. (2013): Computer-aided diagnosis of diabetic retinopathy: A review. Computers in Biology and Medicine, vol.43, pp.2136- 2155.

[33] F. Zana, J.C. Klein, A multimodal registration algorithm of eye fundus images using vessels detection and Hough transform, IEEE Transactions on Medical Imaging 18 (1999) 419–428.

[34] Y. Yang, S. Huang, N. Rao, An automatic hybrid method for retinal blood vessel extraction, International Journal of Applied Mathematics and Computer Science 18 (2008) 399–407

[35] K. Sun, Z. Chen, S. Jiang, Y. Wang, Morphological multiscale enhancement, fuzzy filter and watershed for vascular tree extraction in angiogram, Journal of Medical Systems (2010).

[36] M.S. Miri, A. Mahloojifar, Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction, IEEE Transactions on Biomedical Engineering 58 (2011) 1183–1192.

[37] M. Vlachos and E. Dermatas. (2010) Multi-scale retinal vessel segmentation using line tracking. Computerized Med. Imag. Graph., vol. 34, no. 3, pp. 213–227.

[38] I. Liu, Y. Sun, Recursive tracking of vascular networks in angiograms based on the detection-deletion scheme, IEEE Transactions on Medical Imaging 12 (1993) 334–341.

[39] Z. Liang, M.S. Rzeszotarski, L.J. Singerman, J.M. Chokreff, The detection and quantification of retinopathy using digital angiograms, IEEE Transactions on Medical Imaging 13 (1994) 619–626.

[40] C. Ali, S. Hong, J.N. Turner, H.L. Tanenbaum, B. Roysam, Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms, IEEE Transactions on Information Technology in Biomedicine 3 (1999) 125–138.

[41] P. Kelvin, H. Ghassan, A. Rafeef, Live-vessel: extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images, in: Proceedings of the 10th International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer-Verlag, Brisbane, Australia, 2007.

[42] A.F. Frangi, W.J. Niessen, K.L. Vincken, M.A. Viergever, W.William, C. Alan, D. Scott, Multiscale vessel enhancement filtering, in: Medical Image Computing and Computer-Assisted Interventation MICCAITM98, Springer, Berlin/Heidelberg, 1998, p. 130.

[43] W.A. Barrett, E.N. Mortensen, Interactive live-wire boundary extraction, Medical Image Analysis 1 (1997) 331–341.

[44] K.A. Vermeer, F.M. Vos, H.G. Lemij, A.M. Vossepoel, A model based method for retinal blood vessel detection, Computers in Biology and Medicine 34 (2004) 209–219.

[45] V. Mahadevan, H. Narasimha-Iyer, B. Roysam, H.L. Tanenbaum, Robust model-based vasculature detection in noisy biomedical images, IEEE Transactions on Information Technology in Biomedicine 8 (2004) 360–376.

[46] P.J. Huber, A robust version of the probability ratio test,Annals of Mathematical Statistics 36 (1965) 1753–1758.

[47] C. Field, B. Smith, Robust estimation – a weighted maximum-likelihood approach, International Statistical Review 62 (1994) 405–424.

[48] E. Ronchetti, Robust model selection in regression, Statistics & Probability Letters 3 (1985) 21–23.

[49] B.S.Y. Lam, Y. Hong, A novel vessel segmentation algorithm for pathological retina images based on the divergence of vector fields, IEEE Transactions on Medical Imaging 27 (2008) 237–246.

[50] J. I. Orlando, E. Prokofyeva, and M. B. Blaschko. (2017) A discriminatively trained fully connected conditional random field model for blood vessel segmentation in fundus images. IEEE Trans. Biomed. Eng., vol. 64, no. 1,pp. 16–27.

[51] Q. Li, B. Feng, L. Xie, P. Liang, H. Zhang, and T. Wang. (2016) A cross-modality learning approach for vessel segmentation in retinal images. IEEE Trans.Med. Imag., vol. 35, no. 1, pp. 109–118.

[52] M. Al-Rawi, H. Karajeh. (2007) Genetic algorithm matched filter optimization for automated detection of blood vessels from digital retinal images. Computer methods and programs in biomedicine 87 (3) 248-253.

[53] Kundu, A and Chatterjee. (2012) Retinal vessel segmentation usin-g Morphological Angular ScaleSpace. In Proceedings of the 2012 Third Int Conf on Emerging Applications of Information Technology, Kolkata, India, 30 November–1 December 2012; pp. 316–319.

[54] Jiang, Z.; Yepez, J.; AN, S.; KO, S. (2017) Fast, accurate and robust retinal vessel segmentation system. Biocybern. Biomed. Eng. 37, pp. 412–421.

[55] Kavya k, Dechamma m.g, Santhosh kumar b.j. (2016) Extraction of retinal blood vessel using artificial bee-colony optimization. Journal of theoretical and applied information technology,88 (3).

[56] A. Asad, A. T. Azar, N. El-Bendary, A. E. Hassaanien, et al., (2014) Ant colony based feature selection heuristics for retinal vessel segmentation. arXiv preprint arXiv:1403.1735.

[57] E. Emary, H. M. Zawbaa, A. E. Hassanien, B. Parv. (2016) Multi objective retinal vessel localization using flower pollination search algorithm with pattern search. Advances in data analysis and classification 1–17.

[58] Joes Staal, Michael D. Abramoff, Meindert Niemeijer, Max A. Viergever, and Bram van Ginneken. Ridge based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging, 23(4):501{509, 2005.

[59] Joao V.B. Soares, Jorge J.G. Leandro, Roberto M. Cesar Jr., Herbert F. Jelinek, and Michael J. Cree. Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification. IEEE Transactions on Medical Imaging, 25(9):1214{1222, sep 2006.

[60] Diego Marın, Arturo Aquino, Manuel Emilio Gegundez-Arias, and Jose Manuel Bravo. A new supervised method for blood vessel segmentation in retinal images by using graylevel and moment invariants-based features. IEEE Transactions on Medical Imaging, 30(1):146{158, 2011.

[61] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M.D. Abramoff. Comparative study of retinal vessel segmentation methods on a new publicly available database. In Medical Imaging 2004: Image Processing, volume 5370, pages 648–656. SPIE, 2004.

[62] J. Staal, M.D. Abràmoff, and M. Niemeijer. Ridge-based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging, 23(4):501–509, 2004.

[63] J.V. Soares, J.J. Leandro, R.M. Cesar, H.F. Jelinek, and M.J. Cree. Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification. IEEE Transactions on Medical Imaging, 25(9):1214–1222, 2006.

[64] S. Roychowdhury, D.D. Koozekanani, and K.K. Parhi. Blood vessel segmentation of fundus images by major vessel extraction and sub-image classification. IEEE Journal of Biomedical and Health Informatics, 19(3):1118–1128, 2015.

[65] D. Marín, A. Aquino, M.E. Gegúndez-Arias, and J.M. Bravo. A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariantsbased features. IEEE Transactions on Medical Imaging, 30(1):146–158, 2011.

[66] N. Strisciuglio, G. Azzopardi, M. Vento, and N. Petkov. Supervised vessel delineation in retinal fundus images with the automatic selection of b-cosfire filters. Machine Vision and Applications, 27(8):1137–1149, 2016.

[67] Elisa Ricci and Renzo Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. IEEE Transactions on Medical Imaging, 26(10):1357–1365, 2007. doi:10.1109/tmi.2007.898551.

[68] D Marín, A Aquino, M E Gegundez-Arias, and J M Bravo. A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. IEEE Transactions on Medical Imaging, 30(1):146–158, 2011. doi:10.1109/ tmi.2010.2064333.

[69] M. Melinščak, P. Prentašic, and S. Lon caric. Retinal vessel segmentation using deep neural ´ networks. In 10th International Conference on Computer Vision Theory and Applications (VISAPP), 2015.

[70] M. Melinscak, P. Prentasic, and S. Loncaric. Retinal vessel segmentation using deep neural networks. 2015.

[71] H. Fu, Y. Xu, D. W. K. Wong, and J. Liu. Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), pages 698–701, 2016.

[72] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. International Journal of Computer Vision, 125(1-3):3–18, 2017. doi:10.1007/s11263-017-1004-z.

[73] Shuangling Wang, Yilong Yin, Guibao Cao, Benzheng Wei, Yuanjie Zheng, and Gongping Yang. Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. Neurocomputing, 149:708–717, 2015. doi:10.1016/j.neucom.2014.07.059.

[74] Pawel Liskowski and Krzysztof Krawiec. Segmenting retinal blood vessels with newline deep neural networks. IEEE Transactions on Medical Imaging, 35(11):2369–2380, 2016.

[75] Q. Li, B. Feng, L. Xie, P. Liang, H. Zhang, and T. Wang. A cross-modality learning approach for vessel segmentation in retinal images. IEEE Transactions on Medical Imaging, 35(1):109–118, 2016.

[76] H. Z. Fu, Y. W. Xu, and D. W. K. Wong. (2016) Retinal vessel segmentation via deep learning network and fully-connected conditional random fields,'' in Proc. IEEE Int. Symp. Biomed. Imag., pp. 698–701.

[77] L. L. Ming and X. S. Qi. (2019) Improved U-Net fundus retinal vessels segmentation. Appl. Res. Comput., vol. 24, pp. 1–6.

[78] X. R. Gao, Y. H. Cai, and C. Y. Qiu. (2017) Retinal blood vessel segmentation based on the Gaussian matched filter and U-net. in Proc. 10th Int. Congr. Image Signal Process., Biomed. Eng. Inform. (CISP-BMEI). Aug. 2017, pp. 1–5.

[79] L. M. Liang, X. Q. Sheng, and K. Guo. (2019) Improved U-Net retinal vessels segmentation. Appl. Res. Comput., vol. 6, pp. 1–6.

[80] Q. Jin, Z. Meng, T.D. Pham, Q. Chen, L. Wei, and R. Su. Dunet: a deformable network for retinal vessel segmentation. Knowledge-based systems, 178:149–162, 2019.

[81] Mahendra Kumar,A. Trio-Method for Retinal Vessel Segmentation using Image Processing.

[82] Daniela Herrera, Impact of loss function in Deep Learning methods for accurate retinal vessel segmentation.

[83] Changlu Guo, Dense Residual Network for Retinal Vessel Segmentation

[84] Changlu Guo, SA-UNet: Spatial Attention U-Net for Retinal Vessel Segmentation

[85] Juntang Zhuang, LadderNet: Multi-path networks based on U-Net for medical image segmentation

[86] Md Zahangir Alom, Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation

[87] O. Brinchmann-Hansen. The light reflex on retinal arteries and veins. a theoretical study and a new technique for measuring width and intensity profiles across retinal vessels. Acta Ophthalmologica. Supplement, 179:1–53, 1985.

[88] Cong Sun, Jie Jin Wang, David A Mackey, and Tien Y Wong. Retinal vascular caliber: systemic, environmental, and genetic associations. Survey of ophthalmology, 54(1):74–95, 2009.

[89] Warren S. McCulloch and Walter Pitts. “Neurocomputing: Foundations of Research”. In: edited by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988. Chapter A Logical Calculus of the Ideas Immanent in Nervous Activity, pages 15–27. ISBN: 0-262-01097-6. URL: http://dl.acm.org/citation.cfm?id=65669.104377 (cited on page 22).

[90] F. Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain”. In: Psychological Review (1958), pages 65–386 (cited on pages 22, 129, 130).

[91] F. Rosenblatt. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan, 1962 (cited on page 22).

[92] P. J. Werbos. “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences”. PhD thesis. Harvard University, 1974 (cited on pages 23, 129).

[93] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Neurocomputing: Foundations of Research”. In: edited by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988. Chapter Learning Representations by Back-propagating Errors, pages 696–699. ISBN: 0-262-01097-6. URL: http://dl.acm.org/citation.cfm?id=65669.104451 (cited on pages 23, 129, 137).

[94] Yann LeCun et al. “Efficient BackProp”. In: Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop. London, UK, UK: Springer-Verlag, 1998, pages 9–50. ISBN: 3-540-65311-2. URL: http://dl.acm.org/citation.cfm?id=645754.668382 (cited on pages 23, 166).

[95] Yann Lecun et al. “Gradient-based learning applied to document recognition”. In: Proceedings of the IEEE. 1998, pages 2278–2324 (cited on pages 24, 195, 219, 227).

[96] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436{444, may 2015.

[97] O. Ronneberger, P. Fischer, and T. Brox. (2015) U-Net: Convolutional networks for biomedical image segmentation. in P-roc. Int. Conf. Med. Image Comput. Comput.-Assist. Inter-vent, 2015, pp. 234–241.

[98] Y. Zhang and A.C.S. Chung. Deep supervision with additional labels for retinal vessel segmentation task. In A. Frangi, J. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, editors, Medical Image Computing and Computer Assisted Intervention - MICCAI, volume 11071. Springer, Cham, 2018.

[99] Q. Jin, Z. Meng, T.D. Pham, Q. Chen, L. Wei, and R. Su. Dunet: a deformable network for retinal vessel segmentation. Knowledge-based systems, 178:149–162, 2019.

[100] Karen Simonyan and Andrew Zisserman. (2015) Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556v6. [Online]. Available: https://arxiv.org/ pdf/1409.1556.pdf

[101] K. He, X. Zhang, and S. Ren. (2015) Deep residual learning for image recognition. in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–6.

[102] Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber,J., "Deep neural networks segment neuronal membranes in electron microscopy images," In: NIPS. pp. 2852{2860 (2012).

[103] J. Long, E. Shelhamer, and T. Darrell. (2016) Fully convolutional networks for semantic segmentation. arXiv:1605.06211v1. [Online]. Available: https://arxiv.org/pdf/1605.06211.pdf

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc. URL: http://dl.acm.org/citation.cfm?id=2999134. 2999257.

[105] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012) ImageNet classification with deep convolutional neural networks." In NIPS, pp. 1106–1114.

[106] Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. CoRR, abs/1311.2901, 2013. Published in Proc. ECCV, 2014.

[107] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In Proc. ICLR, 2014.

[108] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In IJCAI, pp. 1237–1242, 2011.

[109] https://towardsdatascience.com/deep-study-of-a-not-very-deep-neural-network-part-2-activation-functions-fd9bd8d406fc

[110] Z.Yan, Xin Yang, and Kwang-Ting Cheng. "Joint segment level and pixel-wise losses for deep learning based retinal vessel segmentation," IEEE Transactions on Biomedical Engineering, 65(9):1912{1923, 2018.

[111] O. Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. "Attention U-net: Learning where to look for the pancreas,". arXiv preprint arXiv:1804.03999, 2018.

[112] A. Kumar, R. K. Agrawal, Leve Joseph, "IterMiUnet: A lightweight architecture for automatic blood vessel segmentation," 2022, arXiv:2208.01485. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2208/2208.01485. pdf

[113] A. Khanal and R. Estrada, ''Dynamic deep networks for retinal vessel segmentation,'' 2019, arXiv: 1903.07803. [Online]. Available: https://arxiv. org/abs/1903.07803.

[114] Guo C , Szemenyei M , Yi Y , "SA-UNet: Spatial Attention U-Net for Retinal Vessel Segmentation, " April. 2020. arXiv:2004.03696. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/

2004/2004.03696. pdf

[115] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: Nature 521.7553 (2015), pages 436–444 (cited on pages 21, 126, 128).

[116] Adrian.R." Deep leaning for computer vision with python, 1st Edition" (September 2017)

[117] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: J. Mach. Learn. Res. 12 (July 2011), pages 2121–2159. ISSN: 1532-4435. URL: http : / / dl . acm . org / citation . cfm ? id = 1953048 . 2021068 (cited on page 84).

[118] Matthew D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: CoRR abs/1212.5701 (2012). URL: http://arxiv.org/abs/1212.5701 (cited on page 84).

[119] Timothy Dozat. Incorporating Nesterov Momentum into Adam. http://cs229.stanford. edu/proj2015/054_report.pdf (cited on page 86).

[120] Tom Schaul, Ioannis Antonoglou, and David Silver. "Unit Tests for Stochastic Optimization". In: CoRR abs/1312.6055 (2013). URL: http://arxiv.org/abs/1312.6055 (cited on page 86).

[121] Bachiri, M.E. ,Rahmoune, A. and Rahmoune, F. (2023) 'Retina blood vessels segmentation by combining deep learning networks', Int. J. Biomedical Engineering and Technology, Vol. 43, No. 1, pp.38–59. DOI: 10.1504/IJBET.2023.133720.

[122] Mohamed Elssaleh Bachiri, Adel Rahmoune, Faycal Rahmoune, "Combining Resnet 34 with U-net for the Segmentation of retinal blood vessels". in the first International Conference on Cyber Security, Artificial Intelligence and Theoretical Computer Science (ICCSAITCS '2022), December 19 – 20, 2022.

[123] https://my.clevelandclinic.org/health/body/21823-eyes.