UNIVERSIDAD DE SEVILLA

DOCTORAL THESIS

A Contribution to Deep Learning based Medical Image Diagnosis Aids

Author:
D. Javier Civit Masot

Supervisors:
Dr. Saturnino Vicente Díaz
Dr. Manuel J. Domínguez Morales

A thesis submitted in fulfillment of the requirements for the degree of Doctor on Computer Sciences

in the

TEP-108 Research Lab: Robotics and Technology of Computers
Departamento de Arquitectura y Tecnología de Computadores

July 22, 2020

Declaration of Authorship

I, D. Javier Civit Masot, declare that this thesis titled, "A Contribution to Deep Learning based Medical Image Diagnosis Aids" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSIDAD DE SEVILLA

Abstract

Escuela Técnica Superior de Ingeniería Informática

Departamento de Arquitectura y Tecnología de Computadores

Doctor on Computer Sciences

A Contribution to Deep Learning based Medical Image Diagnosis Aids

by D. Javier Civit Masot

In this work, an in-depth study about the use of Deep Learning techniques to support healthcare professionals for the recognition of pathologies using medical images is carried out.

Most of the research presented in this work is focused on the detection of glaucoma using images of the eye fundus; however, in order to demonstrate the feasibility of the processing systems implemented in this work, other types of images are used (in this case, X-ray images) to detect another completely different pathology, such as the detection of patients with COVID-19.

Thus, in this work the classic detection techniques for these pathologies are studied, an in-depth study of the techniques based on Deep Learning is carried out, several treatment models are implemented with specific pre-processing stages adapted to the problem itself; and, finally, these systems are tested using large databases in order to demonstrate the feasibility of the those classification systems.

The results obtained demonstrate that Deep Learning techniques can be used as a diagnosis aid of those diseases that require medical images analysis. In this way, the human workload required for these tasks is greatly reduced.

Acknowledgements

A mi familia, que siempre están ahí en los momentos buenos y malos y sé que puedo contar con ellos en cualquier momento, en especial a mi madre, por su incondicional apoyo.

A mis directores de tesis, Manuel y Saturnino, por su apoyo, dedicación y guía.

A todos los miembros del departamento, fuente de conocimiento e inspiración.

En especial a aquellos que me brindaron su apoyo o consejo en algún momento durante el transcurso de este trabajo.

A todos aquellos que no han sido nombrados expresamente y que de forma directa o indirecta han influido en mis pasos y en cómo soy, gracias...

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
AUC	**A**reas **U**nder (the) **C**urve
BAL	**B**ronchoalveolar **L**avage
CDR	**C**up (to) **D**isc **R**atio
CNN	**C**onvolutional **N**eural **N**etwork
DL	**D**eep **L**earning
FDA	**F**ood (and) **D**rug **A**dministration
GPU	**G**raphic (and) **P**rocessing **U**nit
IR	**I**ncrement **R**atio
ISTN	**I**nferior **S**uperior **T**emporal **N**asal
OC	**O**ptic **C**up
OD	**O**ptic **D**isc
RT-PCR	**R**everse **T**ranscription **P**olymerase **C**hain **R**eaction
RRP	**R**adii **R**atio **P**arameter
ROC	**R**eceiver **O**perating **C**haracteristic
TPU	**T**ensor **P**rocessing **U**nit

Chapter 1

Introduction

1.1 Medical Imaging and Deep Learning

As far as 1951 we can find papers in MEDLINE with the term artificial intelligence (AI), when a tortoise neurological research robot was first described (Fletcher, 1951). AI has impacted daily life through applications such as image recognition, natural language translation, self-driving cars, etc (Krizhevsky, Sutskever, and Hinton, 2012; Collobert et al., 2011). Similar success in health diagnostics is expected and some researchers have even suggested that AI applications will partially replace some medical disciplines or create new roles for physicians (Coiera, 2018).

Medical imaging has been for many years one of the most valuable sources of diagnostic information, however it is very dependent on human expert interpretation. The need and availability of diagnostic images is rapidly exceeding the capacity of human specialists, particularly in low and middle-income countries (L. Zhang et al., 2018). Automated or assisted diagnosis from medical imaging through AI (mainly deep learning based), may help addressing this issue (King Jr, 2017). Articles using deep learning models that claim to exceed human diagnostic performance have led to considerable excitement. This data should however be taken with a "pinch of salt" as some studies are biased in favour of the new technology, it is not clear that the findings are generalisable and applicable to a real-world setting. Several AI algorithms have been approved by the Food and Drug Administration (FDA) of the United States (Topol, 2019). In general,the methodology and reporting of the studies evaluating deep learning models is very variable and international standards for protocols that recognise the challenges of deep learning are needed to ensure quality of future studies (X. Liu et al., 2019).

1.1.1 Cloud based Medical Image segmentation

Segmentation is the process of automatic or semi-automatic detection of limits within a 2D or 3D image. A well-known difficulty in the segmentation of medical images is the high variability in the data sources and capture technologies. First, anatomy shows very significant variations. In addition, many different image acquisition systems are used

(X-ray, CT, MRI, PET, SPECT, endoscopy, etc.) to create biomedical images. The segmentation result can also be used to obtain additional diagnostic information. Among the possible applications, we can find automatic measurement of organs, cell count or simulations based on the acquired information.

As already mentioned, the application of Deep Learning methods to medical image analysis has quickly grown in recent years (Litjens, Kooi, Bejnordi, Setio, Ciompi, Ghafoorian, Van Der Laak, Van Ginneken, and Sanchez, 2017) due to their success with different problems, including segmentation. The effectiveness of these systems improves with the number and variety of the training set images. This suggests the development of cloud-based services that can be trained with several dataset initially and retrained with new datasets samples periodically. Reducing training times is an important requirement in this scenario, and Google TPUs are currently one of the most powerful resources available to train and carry out predictions for cloud-based segmentation. Another important aspect is that, in a cloud-based service, images will come from very different sources and, thus the networks must be trained as independently as possible from the acquisition source. Several segmentation researchers (Sevastopolsky, 2017; Al-Bander, B. Williams, et al., 2018) have used several datasets. However, they always train and test with each of these datasets independently. This methodology is not suitable for our application scenario. In the work detailed in [Javier Civit-Masot, Luna-Perejon, et al., 2019] we use a new approach where we preprocess and mix the data from several datasets and use it to create independent datasets for training and validation. There are some works where multiple datasets are used simultaneously (e.g. [Choi et al., 2018]) but they are not related to image segmentation.

Two techniques used in this work are transfer learning where a pretrained network usually trained with data from imagenet (Krizhevsky, Sutskever, and Hinton, 2012) is retrained to solved the specific medical problem. In the work [J. Civit-Masot et al., 2020] this technique is used with a MobileNetV2 network to detect glaucoma in fundus images while in the work [Javier Civit-Masot, Luna-Perejón, et al., 2020] this is used to separate Covid-19, Pneumonia and Healthy subjects in X-Ray chess images. Incremental learning is a variation of this approach where a network previously trained with data from a dataset is retrained with data from a different dataset to improve the source independence of the system. This approach is used in the work [Civit-Masot et al., 2020] where a U-Net is initially trained with a fundus image dataset is later quickly retrained with another.

1.1.2 Convolutional Neural Networks

In this work we use two different families of convolutional neural networks (CNN). In the works [Javier Civit-Masot, Luna-Perejon, et al., 2019; Civit-Masot et al., 2020; J. Civit-Masot et al., 2020] we use a convolution deconvolution network to perform fundus image segmentation. In the work [Javier Civit-Masot, Luna-Perejón, et al., 2020] we use a VGG16 classification network (Simonyan and Zisserman, 2014) to classify X.Ray

images and in the work [J. Civit-Masot et al., 2020] we use an mobileNetV2 (Sandler et al., 2018) to classify fundus images.

U-Net is a fully convolutional deep learning network that has been shown to be effective in several medical segmentation problems. In the work [Javier Civit-Masot, Luna-Perejon, et al., 2019] we focused on the study of generalized U-Net architectures as a method to solve the image segmentation problem in the cloud. It was initially published in the international congress MICCAI 2015 (Medical Image Computing and Computer-Assisted Intervention) and the original paper currently has more than 3800 citations (Ronneberger, Fischer, and Brox, 2015).

The basic architecture of the network is shown in Figure 1.1. The network consists of descending layers formed by two convolution layers with RELU activation and dropout. The result of each layer is sub-sampled using a 2x2 max pool layer and used as input to the next layer. The 5th layer corresponds to the lowest level of the network and has a structure like the other descending layers. From this layer the data is oversampled (in the original version by transposed convolution), merged with the output data of the corresponding downwards layer and applied to a block similar to those used in the descending layers. The last layer of the network is a convolution layer with a width equal to the number of classes to be segmented.

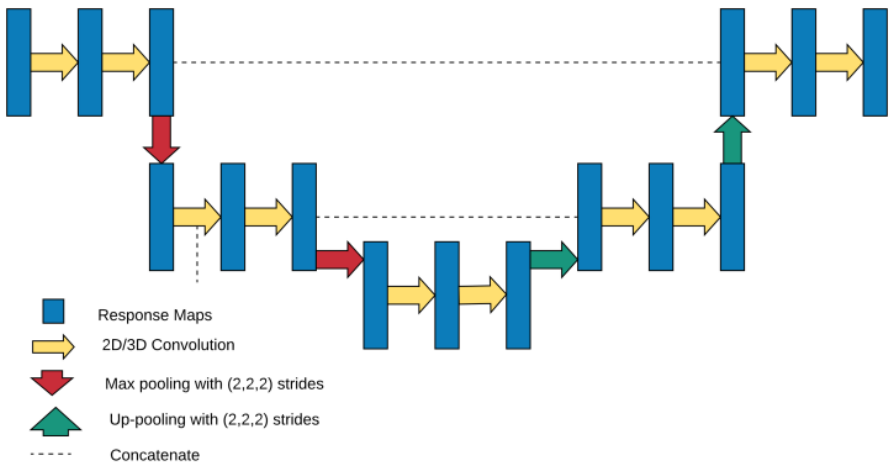


FIGURE 1.1: Basic U-Net Architecture modified to three stages

The details of the implementation are different on most U-Net based projects. They can vary among other things in the following characteristics:

- Layer width: Traditionally, when going down in the network the width of the layer is doubled, and when going up it is divided by 2. This, however, is not always the case. For example, in work [Sevastopolsky, 2017] the structure shown

on Figure 1.2 is used. When the relation between the width of a layer to that of the one above it in the network is constant, this parameter is called layer increment ratio (IR). In the original U-Net this increment ratio is 2, but implementations may use smaller values to maintain a reasonable number of trainable parameters in the network.

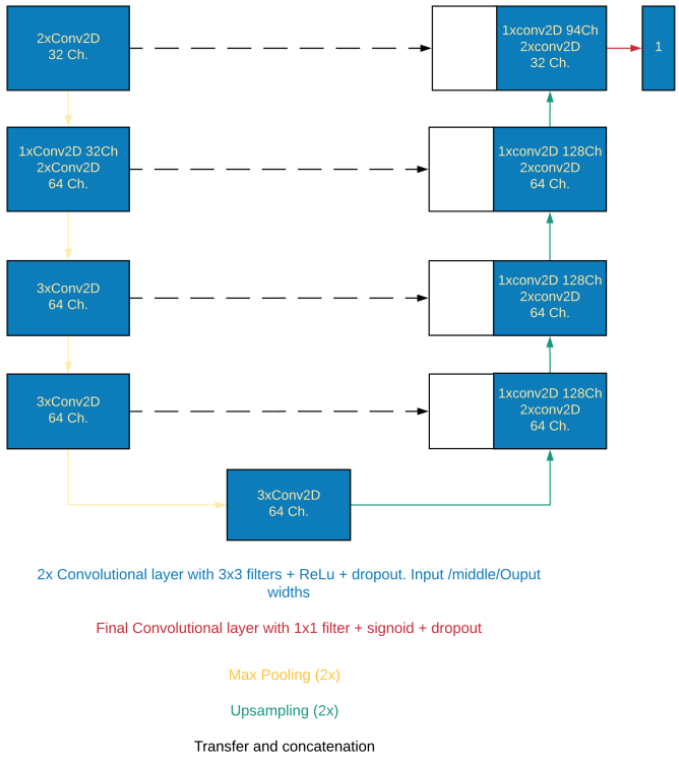


FIGURE 1.2: Sevastopolsky's U-Net, Glaucoma, Optic Disc and Cup

- Transposed convolution or upsampling: In the U-Net up stages we double the resolution of the image in every stage. There are two main approaches for this, either we directly replicate the data to create a higher resolution image, or we use transpose convolution, i.e. a trainable upsampling convolutional layer whose parameters will change during training. Many current U-Net implementations use direct oversampling instead of transposed convolution. An evaluation of this topic can be found in [6]. In our case, we must use transpose convolution for TPU implementations, as direct upsampling is not supported on this architecture.
- Drop-out and Normalization layers: These are used to avoid overfitting the data. This happens when the system learns all the details of the training dataset but can't generalize the prediction when validating with other datasets.

- Optimization algorithms: A decision that has great impact on learning process speed, as well as the accuracy of obtained predictions, is the choice of the optimization strategy (Ruder, 2016).

In our work, we will focus on the influence of the layer widths and the use of normalization and drop-out, as these are some of the aspects that vary widely between different implementations and affect both learning speed and prediction quality. We will make trials with U-Net implementations that are deeper than the standard 5-layer network and with different layer increment ratios.

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the work [Simonyan and Zisserman, 2014]. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet (Alom et al., 2018) by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was originally trained for weeks and was using NVIDIA Titan Black GPU's. The structure of VGG16 can be seen in Figure 1.3.

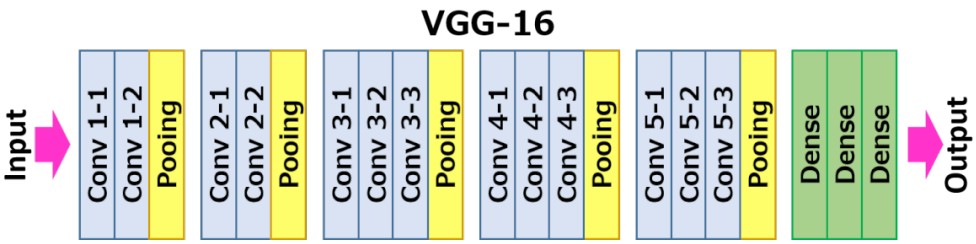


FIGURE 1.3: VGG16 architecture

The MobileNetV2 architecture was introduced in the work [Sandler et al., 2018] and is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations in the input. MobileNetV2 uses lightweight depthwise convolutions to filter features in the intermediate expansion layer. Additionally it removes non-linearities in the narrow layers in order to maintain representational power. The structure of MobilenetV2 can be seen in Figure 1.4.

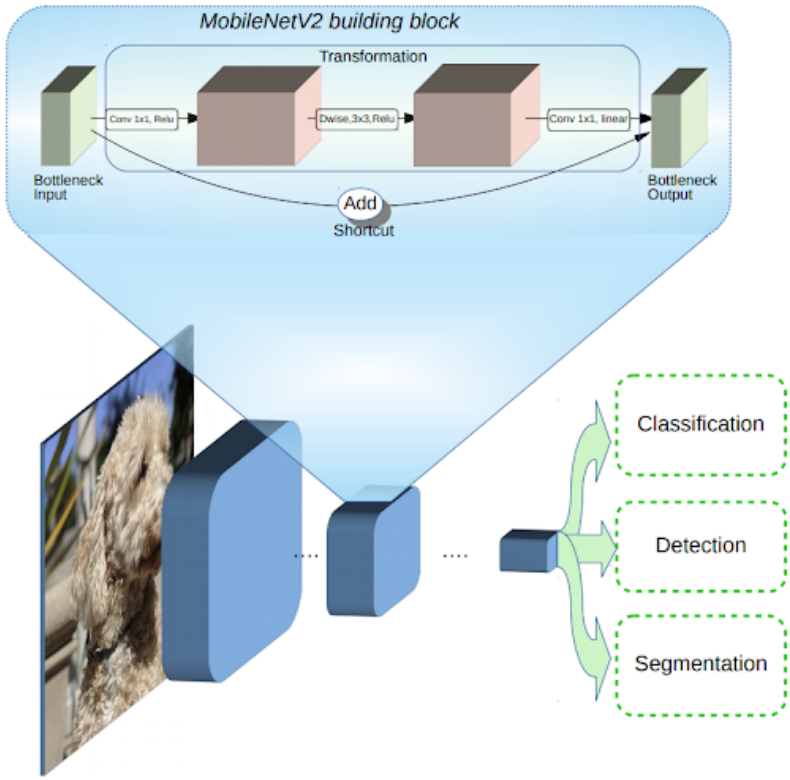


FIGURE 1.4: MobileNet V2 architecture architecture

1.2 Application cases

1.2.1 Glaucoma

Glaucoma is a disabling decease that can lead to blindness in about 2 to 5% of the cases and sight impairment in 10% of the cases (Quigley and Broman, 2006). Although Loss of vision can occur even with the best treatment, correct therapy and follow-up will stabilize the majority of patients with glaucoma.

The key to detection and management of glaucoma is understanding how to examine the optic disc (OD) (Bourne, 2006). The OD is an oval 'plughole' down which the retinal nerve fibres descend through a sheet known as the lamina cribrosa. The retinal nerve fibres are then bundled together to form the optic nerve. The optic cup (OC) is the white, cup-like area in the center of the optic disc. The tissue between the border of the cup and the disc is the neuroretinal rim. This tissue consists mainly of nerve fibers with some glial cells and is usually pink. Most normal discs are more vertically oval and their cup more horizontally oval. A typical retina fundus image is shown in Figure 1.5

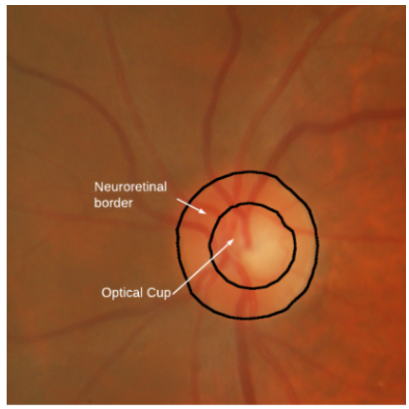


FIGURE 1.5: Optic Disc and Cup

Several indicators are used to aid the diagnosis of glaucoma from fundus eye images. The cup to disc ratio (CDR) (MacIver, MacDonald, and Prokopich, 2017) which is the rate between the diameters of the optic disc and cup is the most widely used. In the mean CDRs of the glaucoma and normal eyes were 0.65±0.13 and 0.39±0.15, respectively allowing CDR to be used as a diagnostic aid. Another diagnostic approach is based on the ISTN rule based on the shape of the neuroretinal RIM. According to this rule in normal eyes, the thickness of the neuroretinal rim along the cardinal meridians of the OD decreases in the order inferior (I)>superior (S) >nasal (N) >temporal (T) (Das, Nirmala, and Medhi, 2016). In any case accurate OC/OD segmentation is required to be able to apply these techniques. This segmentation is an error prone process even for expert ophthalmologists specially in typical work overloaded scenarios.

Some papers have used several deep learning networks in parallel to improve the results that would be obtained by using a single network implementation. As an example, the work [Krizhevsky, Sutskever, and Hinton, 2012] combines the results of five CNNs to improve the results on the LSVRC-2010 ImageNet training set. This technique has also been applied to glaucoma identification obtaining interesting results (Diaz-Pinto et al., 2019). However, all these approaches use a set of CNNs to obtain the same type of results (e.g. the patient has or does not have Glaucoma) and then obtain a combined result by some sort of final voting.

A completely different approach is based on the segmentation of the optic disc and cup. There are several methods that can help predict glaucoma from the segmented disc and cup data in fundus images. First, the ratio between the diameters of the optic disc and cup, known as cup to disc ratio (CDR), is a very useful predictor for Glaucoma. Additionally the order of the widths of the different borders (inferior, superior, temporal and nasal- ISTN) can be used too. Several works have implemented deep learning approaches to segment optic disc and cup in order to be able to estimate the CDR or use the ISTN approach. An important problem of these approaches is that, in a few cases, they produce segmentation results with shapes that are not compatible with the opthalmological knowledge that requires these shapes to be similar to ellipsoids.

In the work [J. Civit-Masot et al., 2020] we use an ensemble approach to glaucoma prediction but, instead of using several convolutional networks to directly predict glaucoma, we use the following approach:

- We segment cup and disc using a generalized U-Net. The process detailed in [Ronneberger, Fischer, and Brox, 2015] is used to calculate the CDR as a glaucoma predictor.
- We use RANSAC (Fischler and Bolles, 1981) to find out if the predicted shapes are similar enough to ellipses.
- We use transfer learning on a MobileNet V2, pretrained with weight from the imageNet 1K challenge to directly predict glaucoma.
- We combine all our result to provide the ophthalmologist with a Glaucoma like hood score.

So, based on these previous approaches and the works done by this research group, In the work [J. Civit-Masot et al., 2020] we combine a dual convolutional neural network (CNN) to classify discs and cups (see Figure 1.5) using data augmentation and feature extraction (extracting physical and positional features), with a classification system based on a pre-trained CNN with transfer learning techniques.

This feature extraction technique is combined with the eye fundus classification CNN for glaucoma detection in a novel work that obtains a diagnosis aid system with results better than previous works.

1.2.2 COVID-19

Coronaviruses are enveloped, unsegmented, and positive-sense single-stranded RNA viruses. Six species of coronavirus are known to cause disease in humans, most of them generally cause mild respiratory disease; however, fatal coronaviruses have periodically emerged in recent decades, such as the 2002 Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV) and the Middle East Respiratory Syndrome Coronavirus in 2012. In December 2019, the Office of the World Health Organization in China was informed of cases of pneumonia of unknown etiology detected in Wuhan, and a new coronavirus, called SARS-CoV-2, was extracted from samples of the lower respiratory tract of several patients (Repici et al., 2020).

Since then, until July 18st of 2020, more than 14.1 million cases have been confirmed worldwide, and the infection has spread to many countries around the world. USA has the highest rate in America with more than 3.68 million cases and more than 104.000 deaths. In Europe, Spain has one of the highest rates with more than 260.000 confirmed infections and more that 28.000 deaths (Dong, Du, and Gardner, 2020). On March 11th of 2020, the World Health Organization (WHO) declared the infection as a pandemic and, since then, several countries have applied restriction measures to their population in order to reduce the spread of the disease (Sohrabi et al., 2020).

The most common symptoms of the disease related to SARS-CoV-2, called Coronavirus Disease 2019 or COVID-19 by WHO on Feb 11th of 2020, are fever, weakness, cough, and diarrhea. More than half of patients report shortness of breath and few develop acute respiratory distress syndrome. After septic shock, refractory metabolic acidosis and coagulation dysfunction can lead to death, with a fatality rate about 6% worldwide (Rothan and Byrareddy, 2020). Some countries has a higher death rate, like United Kingdom and Italy with a value around 14%, and Spain with a 11.3%.

Person-to-person transmission occurs primarily through direct contact or air drops. The highest risk of transmission is within about 1 meter of the infected person; however, the maximum distance is still undetermined (Q. Li et al., 2020).

Most countries are using a huge amount of clinical and epidemiologic information to determine who should be tested. According to empirical studies like the one presented in [Lauer et al., 2020] or the one detailed in [Cascella et al., 2020], most patients with confirmed COVID-19 develop fever and/or symptoms of acute respiratory illness (like cough or difficulty breathing). If a person is under investigation, it is recommended that practitioners immediately put in place infection control and prevention measures.

The first recommendation is testing for all other sources of respiratory infection (to exclude COVID-19). Moreover, in order to assist in the decision making process and to determine who to test, some epidemiologic factors are recommended to be used. These factors include anyone who has had close contact with a patient with laboratory-confirmed COVID-19 within 14 days of symptom onset or a history of travel from affected geographic areas (Organization et al., 2020).

Once, these factors determine that testing should be done, the WHO recommends collecting specimens from both the upper respiratory tract (naso and oropharyngeal samples) and lower respiratory tract such as expectorated sputum, endotracheal aspirate, or Bronchoalveolar Lavage (BAL) (Z. Xu et al., 2020); but the collection of BAL samples should only be performed in mechanically ventilated patients as lower respiratory tract samples seem to remain positive for a more extended period. All the samples require storage at four degrees celsius.

In the laboratory, the amplification of the genetic material extracted from the saliva and/or mucus samples are carried out through a Reverse Transcription Polymerase Chain Reaction (RT-PCR), which involves the synthesis of a double-stranded DNA molecule from an RNA mold (Lan et al., 2020). Once the genetic material is enough, the search is done for those portions of the genetic code of the COVID-19 that are conserved. This comparison is performed using the initial gene sequence released by the Shanghai Public Health Clinical Center & School of Public Health (Fudan University, Shanghai, China), and subsequent confirmatory evaluation by additional labs. If the test result is positive, it is recommended that the test is repeated for verification. In patients with confirmed COVID-19 diagnosis, the laboratory evaluation should be repeated to evaluate for viral clearance prior to being released from observation.

As detailed above, actual procedures to diagnose COVID-19 patients require several hours to obtain a result. Moreover, those exams may be negative if the patient was infected recently.

In other viral diseases that affects breathing, such as influenza or SARS, the damage produced to the lungs can be observed using pulmonary X-ray images. The works presented in [Serebriakova et al., 2012] and [Z. Q. Lin et al., 2015] detailed these effects in influenza, while other works like [Tse et al., 2004] and [Xie et al., 2006] explain the effects in SARS patients.

So, it is logical to think that this relationship is maintained with COVID-19 patients, since this disease mainly attacks the lungs. However, classic pneumonia patients experience some symptoms similar to those COVID-19 patients in the early stages of the contagion, although with lower virulence. Even so, this fact must be taken into account to correctly diagnose this disease.

However, the study of medical images has experienced a great progress with the inclusion of Machine Learning systems capable of automatically extract the necessary characteristics to make a correct diagnosis (Ker et al., 2017).

Moreover, in the last years these technology has evolved to a concrete branch known as Deep learning. While in Machine Learning the user gives the system a huge amount of rules to solve the problem, in Deep Learning the user gives the system a network model and only a few instructions to modify the model when errors occur. So, using Deep Learning, it is easier and faster to train the classification system.

All of them require a dataset made up of several images corresponding to ill patients and healthy patients (all of them previously labeled by a professional). Using

this knowledge, neural network-based systems are able to automatically analyze those images and extract the characteristics necessary to diagnose the illness.

These systems require several steps like a pre-processing stage, the correct choice of the network architecture, a training stage (that sometimes requires supervision), among others. In Deep Learning systems, although the network model is already established, it is very common to use a pre-processing step to adapt the inputs to the ones needed by that model.

These techniques have been used in multiple industrial and medical systems, obtaining very good results (Luna-Perejon, Manuel Jesus Dominguez-Morales, and Civit-Balcells, 2019; Manuel J Dominguez-Morales et al., 2019). Regarding its application to the medical images analysis, there are several studies that demonstrate that the results obtained are better that the ones obtained by classical diagnostic systems (Litjens, Kooi, Bejnordi, Setio, Ciompi, Ghafoorian, Van Der Laak, Van Ginneken, and Sánchez, 2017). Furthermore, its application and effectiveness have been proven in other works (Asri et al., 2016; Jhuo et al., 2019; Javier Civit-Masot, Luna-Perejon, et al., 2019).

So, based on these premises, the work presented on [Javier Civit-Masot, Luna-Perejón, et al., 2020] consists of using Machine Learning techniques applied to medical X-ray images of the lung of the patients to obtain an aid system for COVID-19 diagnosis. It is important to emphasize that there are other imaging tools to detect COVID-19 like RM or CT; however, the objective of this work is not to obtain images from patients, but using an existing dataset that meets all the requirements.

And, to achieve this purpose, a public dataset that contains X-ray images about healthy, pneumonia and COVID-19 patients all over the world is used. This dataset has mainly X-Ray images and this is the justification of choosing X-Ray images.

With the information included in the dataset, a Deep Learning system is trained and the classification results are detailed in this work.

Chapter 2

Objectives, Materials and Methods

2.1 Overall Objectives

The idea behind this Thesis is to build the basis for future medical image based explainable diagnostic aid tools. To achieve this end the main objectives of this thesis are:

1. To study the feasibility of achieving medical image segmentation image segmentation as a web service and evaluate its performance. This requires:
   (a) To select a suitable medical image segmentation architecture. We will use optic disc and cup segmentation in eye fundus images as our specific application example.
   (b) To tune the parameters of the selected architecture and study its performance with different configurations. To this end an implementation and training environment that allows very high performance training is needed to be able to analyse a wide set of different configurations.
   (c) To prune the selected architecture to reduce its computation costs in the cloud and allow possible implementations in embedded devices.
   (d) To Study the performance of the selected architecture when trained with images from a specific dataset (acquired with a specific instrument) and used to make predictions on images from other dataset.
   (e) To study the feasibility of training with combined datasets and its impact on the system performance.
   (f) To study the possibility of incremental training, i.e., training initially with images from a dataset and performing quick retrains with more data as these become available.

   (g) To study and evaluate post-processing approaches that ensure that the produced result are acceptable to the human expert and provide a meas sure of the acceptability of the proposed result.

2. To study the feasibility of achieving medical image classification using light weight networks and evaluate its performance.This requires:
   (a) To select suitable medical image classification architectures. We will use two different application examples: Glaucoma detection from eye fundus images and covid-19 and pneumonia detection from chest X-ray images.
   (b) To tune the parameters of the selected architectures and study their performance. To this end an implementation and training environment that allows very high performance training is needed to be able to analyse a wide set of different configurations.
   (c) To make a decision on which of the selected architectures has better performance with lower computation load to reduce the operating costs in the cloud and allow possible implementations in embedded devices.
3. To study the feasibility of combining segmentation based and classification based results in a classification ensemble and evaluate its performance. This requires:
   (a) Studying the different approaches to glaucoma detection using optic cup and disc segmentation data.
   (b) Selecting a specific set of methodologies to implement glaucoma detection from the segmentation data.
   (c) Analyze the possible alternatives available for combining the results of the segmentation and classification networks in a diagnostic prediction.
   (d) Evaluate the performance of the proposed diagnostic aid ensemble.
4. To study the feasibility of creating reporting tools for the physician that provide insight into the basis that supports the proposed diagnostic. This requires:
   (a) Analyzing the intermediate output data provided by the diagnostic aid ensemble.
   (b) Establishing which elements from the available data may provide useful information to support the physician's final diagnostic.
   (c) Providing the data in a systematic report.

In the work [Javier Civit-Masot, Luna-Perejon, et al., 2019] we cover mainly the objectives 1a, 1b, 1c, 1d and 1e. In the work [Civit-Masot et al., 2020] we mainly cover 1e and 1f. In the work [Javier Civit-Masot, Luna-Perejón, et al., 2020] we cover objectives 2a, 2b and some aspects of 2c.

In the work [J. Civit-Masot et al., 2020] we cover almost all the objectives of this thesis although 1a, 1b, 1c, 1d, 1e and 1f are mainly referenced from the works [Javier

Civit-Masot, Luna-Perejon, et al., 2019] and [Civit-Masot et al., 2020]. Objectives 2a,2b and 2c are shared between works [Javier Civit-Masot, Luna-Perejón, et al., 2020] and [J. Civit-Masot et al., 2020] but applied to different diagnosis scenarios. Finally objectives 4a, 4b and 4c are only covered in the work [J. Civit-Masot et al., 2020].

In the remaining sections of this chapter we will study the required materials and methodologies necessary to achieve these goals while in the next chapter we will study the results obtained using these methodologies.

2.2 Initial Disc and Cup Segmentation Architectures.

For this work, a toolset of functions was developed to generalize U-Net models, allowing a quick and adequate implementation on cloud-based GPU and TPU architectures. We used the cooperative iPython notebook development environment Google Colaboratory (. The environment has very good support for Keras Francois Chollet, 2017, with the possibility of implementing and training networks based on GPUs and TPUs in Google Cloud.

TPUs are a new type of processors designed for deep learning network acceleration that use a systolic array for multiplication and can decrease the learning times for convolutional neural networks (CNNs) several times. Training on TPUs allows us to test wider and deeper architectures that will be out of the memory limits of many current single GPU systems. The higher training speeds also allows us to prune the network to make it lighter with small effects in prediction efficiency.

Although we will do most of our training and predictions directly on TPUs, we will perform a small set of trials on GPUs to verify this claim for U-Net based segmentation.

We initially based our first work on the notebooks by Sevastopolsky, 2017, we made many very significant modifications:

- We use a completely different dual image generator for both for training and testing. For TPU training, we need larger static datasets and thus we make use of static data augmentation including images with modified brightness and modified parameters for adaptive histogram equalization. This, together with the use of images from three different publicly available datasets for training and validation, improves the system robustness allowing the use of images acquired with different instruments. Aggressive data augmentation has been shown as a very effective approach to avoid overfitting in image segmentation Zoph et al., 2019.
- We use the version of Keras included in TensorFlow. This is necessary to be able to execute it on TPUs. To our best knowledge, this is the first time that generalized U-Nets have been implemented and trained on TPUs.
- We use a parameterizable recursive U-net model. This model allows us to easily change many parameters necessary to compare different implementations of U-Net. Specifically, we can change the network depth and width, the use of drop-out

and batch normalization, the use of upsampling (although this type of layer is not currently supported by Keras in TPUs) or transpose convolution and the width ratio between successive layers (IR). IR was originally introduced in Howard et al., 2017a and is widely used as an effective pruning method. In our work we will always choose the networks with smaller IR and, thus, smaller number of trainable parameters when two networks produce similar results. Even though we don't train or perform predictions in the user device, in which case pruning would be essential, when using cloud-based resources, pruning improves timing and reduces operational costs. Reducing the network initial width and its depth are alternative pruning methods that we also explore.

We use 120 image batches for both training and testing, and we train for 15 epochs using 150 training steps and 30 testing steps per epoch. We use an Adam optimizer algorithm in most cases with a .00075 learning rate, although in a few cases we have had to lower this value to ensure convergence. These values have proven suitable for TPU-based training in U-Net architectures and provide good results with training times below 30 minutes even for the most complex implementations. In this training times, we include the recompilation processes carried out by the TPU XLA just-in-time compiler.

2.2.1 Initial Datasets.

Regarding the datasets, we use publicly available RIM-ONE v3, DRISHTI and DRIONS datasets. The use of multiple datasets simultaneously, both for training and for validation, allows a greater independence from the capture devices. RIM ONE-v3 (Fumero et al., 2011), from the MIAG group at the University of La Laguna (Spain), consists of 159 fundus images which have been labeled by expert ophthalmologists for both disc and cup. DRISHTI-GS (Sivaswamy et al., 2014), from Aravind Eye Hospital, Madurai (India), consists of 101 fundus images also labeled for disc and cup. DRIONS-DB (Carmona et al., 2008) from Miguel Servet Hospital, Saragossa (Spain), consists of 110 images on which only the optic cup has been labelled.

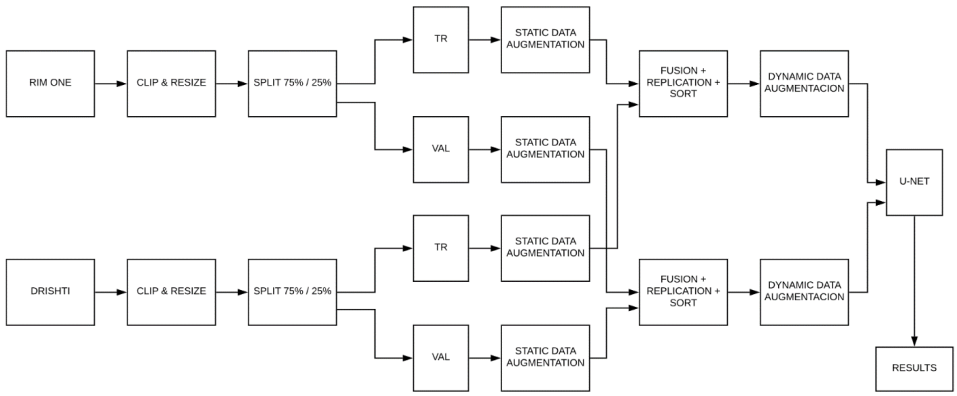
FIGURE 2.1: Images from different datasets

We provide a Google Colaboratory iPython notebook at GitHub [1] for disc and cup segmentation. The code for both cases is the same, and the only difference is the loading and pre-processing of images and masks.

As already mentioned, to perform disc and cup detection as a service in the cloud, it is necessary that we are independent, as much as possible, from the specific characteristics of the captured image. As an example, in Figure 2.1 we can see that images coming from the three different datasets have very different characteristics.

Our approaches for disc and cup segmentation are very similar. Figure 2.2 shows the methodology used for cup segmentation. In this case we use the only two datasets that include the required data (RIM-ONE and DRISHTI). Originally, we start by clipping and resizing the original images in the datasets. When we segment the disc, we remove a 10% border in all the edges of the image to reduce black borders in the images. When we segment the cup, we select the area that contains the disc plus an additional 10% from the original images. After clipping, we resize the images to 128x128 pixels and perform a clip limited contrast equalization.

FIGURE 2.2: Multi-dataset based training approach. The diagram shows only two datasets for simplicity.



After the equalization, we split the dataset. For each dataset, we use 75% of the images for training and 25% for validation. It is essential to split the datasets before performing any data augmentation, in order to ensure that the training and validation sets are completely independent from each other. After splitting we perform, for each dataset, static data augmentation by creating images with modified brightness and different adaptive contrast parameters.

After the static data augmentation, we merge the data from the different datasets. This process is done independently for the training and validation dataset. In the fusion process, we perform data replication and shuffling so that we provide longer vectors as

[1] https://github.com/javicivit/TPU-UNET

input for our dynamic image generators. The image generators do data augmentation by performing random rotations, shifting, zooming and flipping on the extended fused dataset images.

2.3 Architectures for Multidataset and Incremental Training

In this section We present results from two different U-Net variations. The architectures are a fairly standard 5 layer U-Net and a deeper 6 layer version. Our five layer architecture is much lighter than the original U-Net (Ronneberger, Fischer, and Brox, 2015). It has only 40 channels in the first layer instead of 64 and the layer increment ratio, i.e. the ratio of the number of channels in a layer to that of the next one (Howard et al., 2017b) is 1.2 instead of 2. If we add the fact that we initially resize our images to 128x128 this reduces the number of trainable parameters in our network to below 1 million. Our alternative architecture implementation uses six layers instead of the original 5 but reduces the layer increment ratio by 10% and, in this way, keeps the number of parameters very similar in both implementations.

Both networks can be trained on cloud TPUs (Jouppi et al., 2018) freely available on Google colaboratory. In our experience TPUs provide a training time speedup over GPUs above 3.0 for generalized U-Nets (Javier Civit-Masot, Luna-Perejon, et al., 2019). A more general study (Wei, Brooks, et al., 2019) reports speedups between 3 and 10. For both networks, when training with images from only one data set, results are similar to those presented in other publications.

We don't implement a web service for disk segmentation but test the possibility using the same web resources and networks that would be used to implement the service of training a network that can segment images from different data sources. This is not the case found in other papers (e.g. [Al-Bander, B. Williams, et al., 2018; Sevastopolsky, 2017; Shankaranarayana et al., 2017; Zilly, Buhmann, and Mahapatra, 2017]) where the data used for training and for making predictions come from the same source. In the service scenario we would have to be able to perform segmentation on data coming from many different clinics and, thus, from several different capture sources.

Our networks have been implemented using Google Collaboratory python notebook environment. This tool supports Keras (Francois Chollet, 2017) and allows training and testing networks based on GPUs and TPUs in Google cloud. We use a recursive flexible U-net model that allows easy modifications to the U-Net implementation. We also perform aggressive static and dynamic data augmentation using a variation of the approach proposed in the work [Zoph et al., 2019].

For training and testing we use very large 450 image batches as this improves the performance on the TPU implementation. For both networks the training we use is 25 epochs, 40 training steps and 6 validation steps for each epoch. We decided to use an

Adam optimizer with a 0.0007 learning rate. The values obtained have been proven adequate for the finality of training both U-Nets and give excellent results with reasonable times for train. We perform cross validation using repeated random sub-sampling. The loss function used is based on the Sørensen-Dice coefficient (Dice, 1945) which we will subsequently call Dice.

As our datasets we use RIM-ONE v3 and DRISHTI. The first of them (Fumero et al., 2011), from a department of the Spanish University, La Laguna, includes 159 images with tags from expert in the field of ophthalmology. The second one (Sivaswamy et al., 2014), created in the Hospital Aravind Eye, Madurai, in India includes the amount 101 images that also has been labeled by experts.

These datasets include accurately annotated disc segmentation by expert ophthalmologists. This type of precise segmentation requires a lot of work from the medical professional and the possibility of using bounded box or other less work demanding weak annotations (Fu et al., 2018) would make much easier to implement web based diagnostic aids in the future and should be further studied.

As mentioned, our aim is to study the possibility of implementing disc segmentation as a cloud service and this requires to be independent from the characteristics of the image acquisition devices. Figure 2.3 shows that images in each dataset, which were clearly captured with different instruments, have very different characteristics.

FIGURE 2.3: Images from RIM ONE and DRISHTI datasets

Figure 2.4 shows the methodology used for segmentation when using either a mixed dataset for training and validation or when training with a single dataset and validating with both datasets.

As a first step we clip and resize the original images in the datasets and reduce the black borders present in the images. After this step images are resized to 128x128 and perform a contrast limited adaptive histogram equalization (CLAHE) (Reza, 2004).

After this process we carry out data set splitting using 75% of the data for training and 25% for validation. We have to partition the datasets before any data augmentation process to guarantee that the sets used for training and validation are totally independent. We perform four trials using different random seeds to implement randomized sub sampling cross validation and use the mean results of these trials.

After the data set splitting step we implement static data augmentation by producing images with modified CLAHE parameters and brightness.

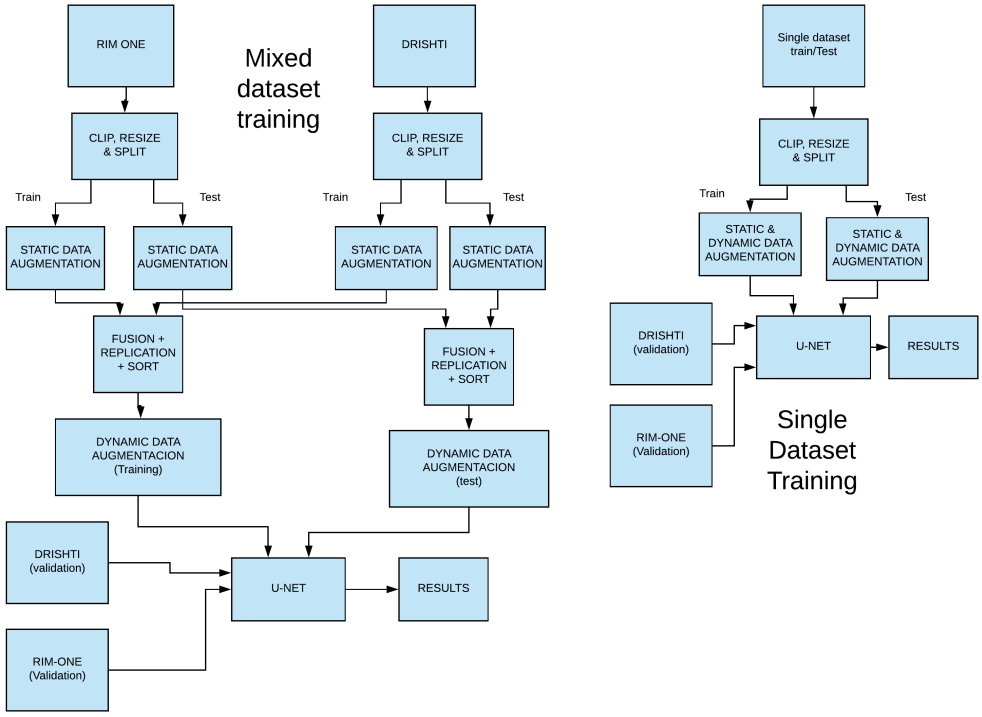


FIGURE 2.4: Segmentation Methodology for combined and single datasets

After the static data augmentation step, when training using a combined dataset, we fuse the data coming from the DRISHTI and RIM ONE datasets. This process has to be performed independently for the validation and training datasets. In this process we also implement data replication and shuffling to provide longer vectors to the dynamic augmentation image generators. This step is needed specially for TPU based training as the performance of these processors improves greatly for large image batches. We use image generators (Francois Chollet, 2016) that perform dynamic data augmentation by implementing random shifting, rotations, flipping and zooming on the already staticlly augmented dataset.

As the relation between the OD and the OC diameters (CDR), is one of the best established glaucoma indicators is , we use a new parameter called RRP -Radii Ratio parameter- based on the ratio between the radius of the predicted and that of the ground truth segmented discs. We use the disc area to estimate the radii of both discs.

The RPP provides an additional quality parameter which we define as the percentage of the test images where the estimated radius error is below a certain percentage. In our work we consider the RRP as the percentage of images with a radius error under 10%.

We will compare our network with the results other papers that perform Deep Learning based optic disc segmentation and use the RIM ONE or the DRISHTI datasets. Zilly et. al. (Zilly, Buhmann, and Mahapatra, 2017) use a three layer CNN with significant pre and post-processing and uses both the DHISHTI dataset. Sevastopolsky (Sevastopolsky, 2017) uses a simpler U-Net architecture than those analyzed in this paper and provides results for the RIM ONE data set. Al-Bander (Al-Bander, B. Williams, et al., 2018) uses a modified dense U-Net architecture and provides results for both datasets but training independently for each of them. Shankaranarayana (Shankaranarayana et al., 2017) uses a modified residual U-Net and provides results for the RIM ONE dataset.

2.4 Global System Approach

This section presents the dataset used for training the Machine-Learning system in this work, as well as the global architecture of the system implemented to diagnose glaucoma based on the properties of the disc and the cup.

2.4.1 Dataset

The database used in this work combines two publicly available datasets: RIM-One V3 and DRISHTI. This is the one used in a previous work (Javier Civit-Masot, Luna-Perejon, et al., 2019), and it is important to continue with this combination in order to compare the results obtained in this work with the ones obtained before.

Both datasets provide labels indicating if the images correspond to a patient with glaucoma or not. The labeling process includes the supervised evaluation of each of the dataset samples by a professional in the field. Thus, this professional certifies that each of the images from the datasets corresponds to a patient with glaucoma or a healthy patient. Works that perform cup and disc segmentation also need the ophthalmologists to manually perform this segmentation and, thus, provided also the labeled images indicating the ground truth for the disc and cup areas.

The DRIONS dataset used in previous studies is not useful in this case as it does not provide segmentation data for the cup which is essential in our case. That is why, in this work, it is not included.

DRISTI-GS dataset from Aravind Eye Hospital, Madurai (India), is made up of 101 color fundus images labeled for both disc and cup; and RIM-ONE dataset from the University of La Laguna is composed of 151 images also labeled for disc and cup.

Figure 2.5 shows an image from each dataset and makes clear that, even though both provide good quality data for segmentation the characteristics of images from both datasets are significantly different.

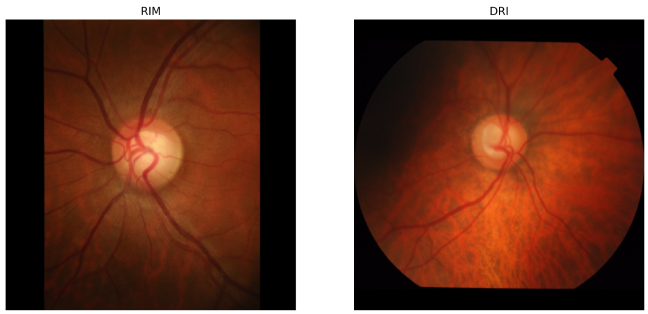


FIGURE 2.5: Images from RIM-ONE and DRISHTI datasets.

In our work we use 75% of the images from each dataset for training and the remaining 25% of the images for validating the results. However, static (offline) and dynamic (online) data augmentation stages are included in the system's architecture, so the total number of images used for training and testing is much higher than in the original datasets. This can be observed in Table 2.1.

TABLE 2.1: Dataset summary

Dataset	Images	Images after D.A.	Train (75%)	Test (25%)
RIM-ONE	149	6980	5235	1745
DRISHTI	101	2380	1785	595
TOTAL	250	9360	7020	2340

The first column shows the number of images that are provided in those public datasets, the second column indicates the final amount of images used after data augmentation processes and, finally, the other two columns present the number of images used for training and testing purposes, respectively.

2.4.2 System architecture

Once the problem we want to solve in this work and the datasets used to train the machine-learning system are detailed, it is very important to describe the full system

architecture used for training and classification.

Our approach is based on two subsystems whose results are finally combined to produce a diagnosis assistance report for the ophthalmologist. The first subsystem is a based on two generalized U-Net based stages to segment the disc and cup plus a feature extraction post-processing stage. The second subsystem is based on a MobileNet V2 (Sandler et al., 2018) network used for direct fundus image classification. There is also a final fusion stage to produce the blended results as a report to assist the ophthalmologist in her or his diagnosis process.

The full system implemented and trained in this work is presented as a graphical abstract in Figure 2.6 for the first subsystem, and in Figure 2.7 for the second subsystem. Both figures show all the steps implemented for training and testing each subsystem.

Several stages can be appreciated in those figures for both subsystems, from the pre-processing stages to the final evaluations. However, results obtained from both subsystems are finally combined in the diagnosis aid tool, and this can be observed in Figure 2.8.

Next, both subsystem are detailed step by step.

Segmentation Subsystem

The first subsystem has been named as "segmentation subsystem" as it uses the segmentation process to train two independent systems for disc and cup features' extraction. The different stages implemented for this subsystem are detailed below.

Pre-processing In order to be able to use the dataset images in the segmentation subsystem we need to:

- Perform image trimming to remove borders
- Resize images to the subsystem input size. We use 128×128 images for the segmentation subsystem. We use resampling using pixel area relation for image size reduction.
- Perform contrast limited adaptive histogram equalization.

Static (Offline) Data Augmentation When we are training the system we perform static data augmentation on the training fraction of the combined dataset. This process consists in producing images with modified brightness or contrast parameters. Our data augmentation approach is loosely based on the work [Zoph et al., 2019].

Dynamic (Online) Data Augmentation When training we use image generators to perform on the fly data augmentation. This process performs moderate zooming and rotations. It is important to understand that glaucoma diagnosis is related to the orientation of the segmented image and, thus rotations should be limited to small angle values (below 15 degrees).

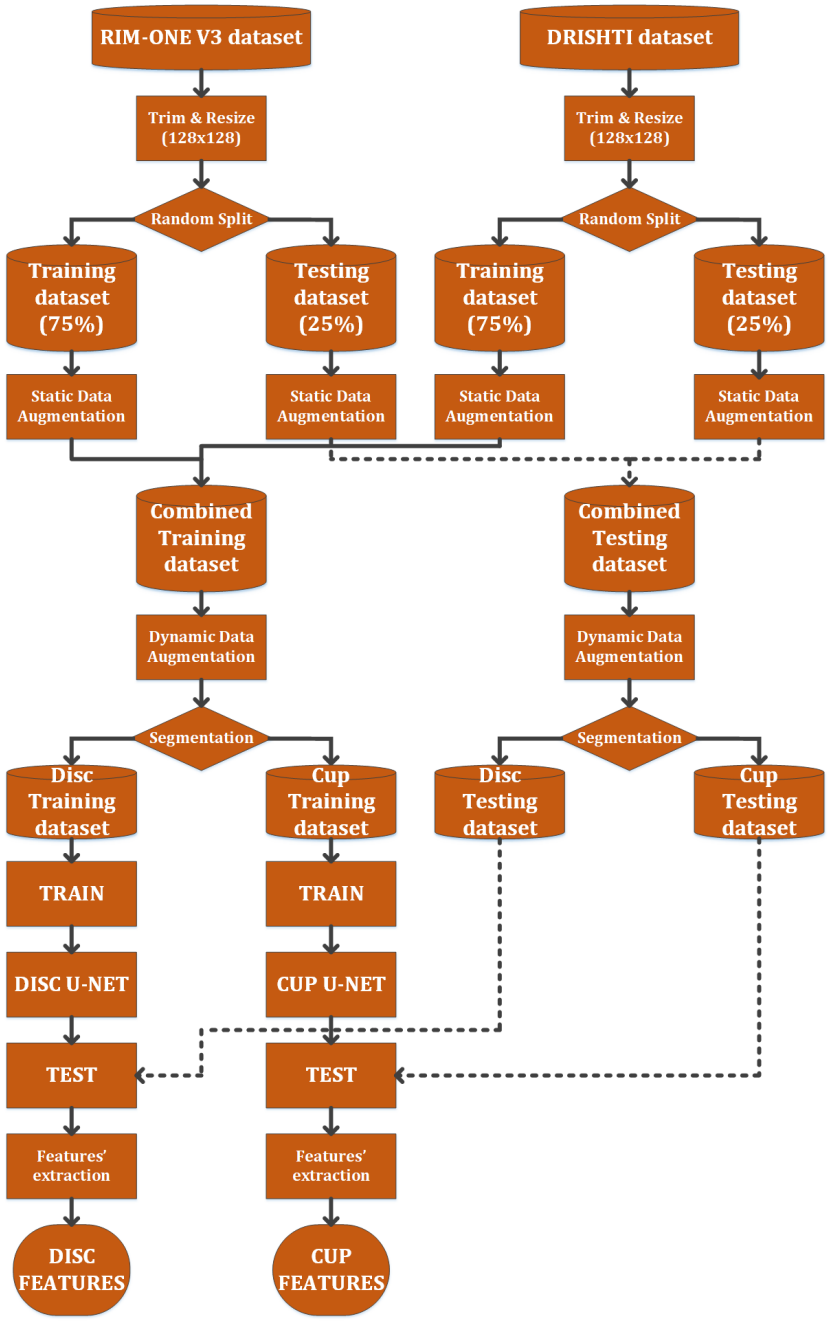


FIGURE 2.6: First subsystem. Disc and Cup segmentation subsystem.

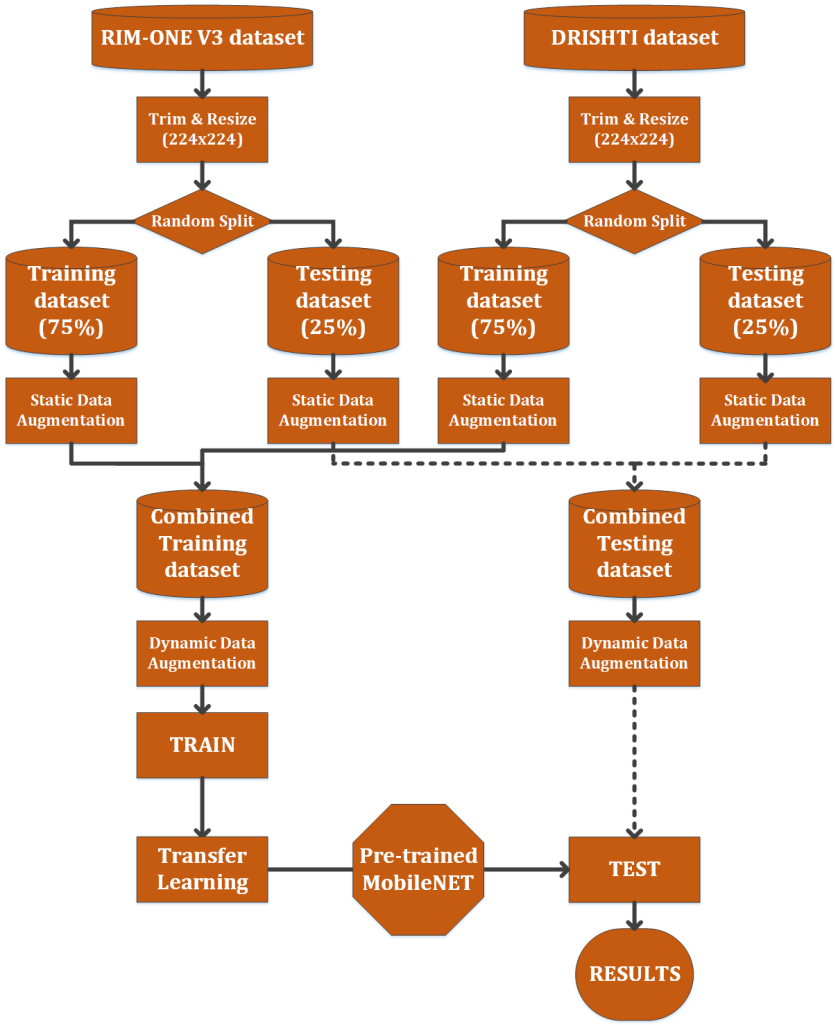


FIGURE 2.7: Second subsystem. Eye fundus image classification.

Segmentation Network To segment the disc and the cup from fundus images we use a generalized U-net architecture and train it using Google cloud TPUs. U-net is widely used fully convolutional network that has been widely used for medical image segmentation. This part of the architecture is fully described in the work [Civit-Masot et al., 2020]. In our case we are using a 6 level network with 64 channels in the first descending stage and a layer channel increment ratio (IR) of 1.1. This model has less than 2.5M trainable parameters and produces good results for both segmentation cases. Although the model has one more stage than the original U-net and the same number of channels in the first layer the reduction of the IR from 2 to 1.1 has decreased the number

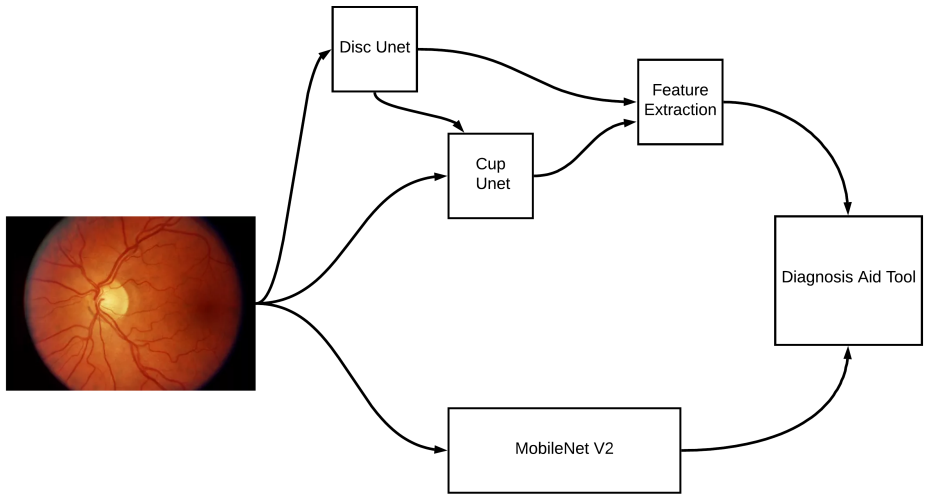


FIGURE 2.8: Diagnosis Tool Architecture.

of parameters from 138M to less than 2.5M. The proposed U-Net implementation block diagram is shown in Figure 2.9.

Training Our network is implemented as a recursive function in Keras 2-3-0-tf under Tensorflow 2.2.0. We use 120 image samples as this size is suitable for training using TPUs, GPUs or even CPUs. We use Adam optimizer with dynamically variable learning rates (between 1e-3 and 2e-4) and perform the training process during 100 epochs.

Post-processing It is quite common that some segmentation results are not acceptable to ophthalmologists the main reasons for this are the following:

- The cup and the disc should always be always a single connected region.
- The shape of both regions should be approximately elliptical.
- The size of the optical disc is similar in images captured with the same instrument.

To solve the first problem, in the few cases where segmentation produces multiple regions we select only the one with the largest area. In these cases we decrease the certainty score for the ophthalmologist.

Next we have to establish the similarity between the segmented area and an ellipse. Initially we tried approaches based on the ellipse Hough transform (Guil and Zapata, 1997) obtaining poor results for our scenario. An approach fitting an ellipse model using Random sample consensus (RANSAC) produces much better results an facilitates the calculation of an ellipse similarity score. When this score is bellow a certain threshold we also decrease the certainty score.

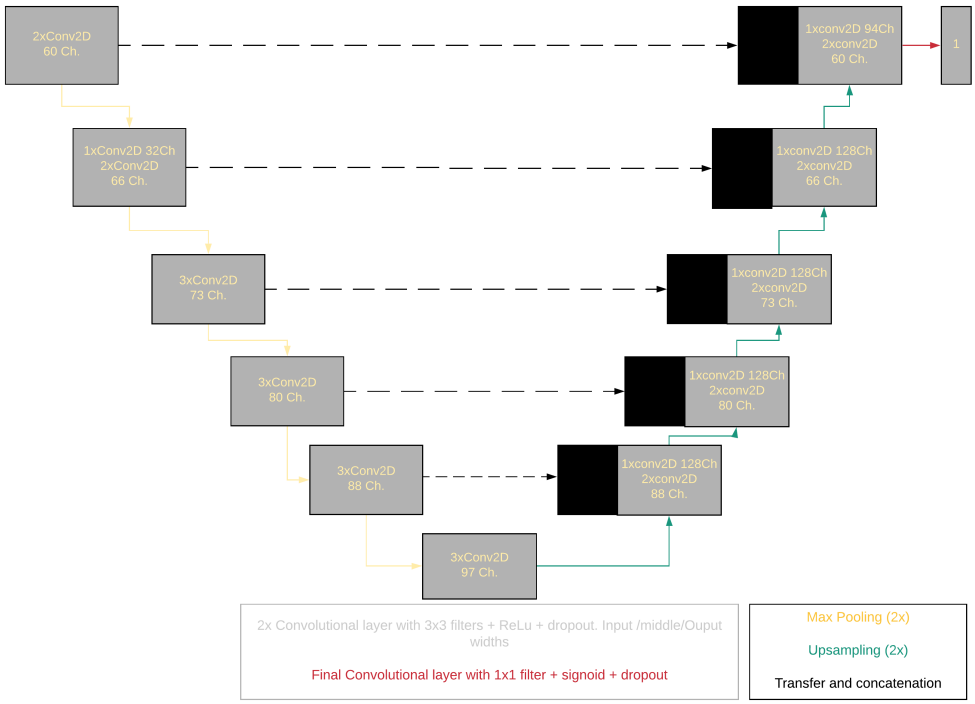


FIGURE 2.9: Generalized U-Net architecture.

As a last post filtering stage we penalize those cases were the size of the optic disc is outside a 4 standard deviation interval centered on the disc size mean. This interval is specific for each acquisition instrument. In our case all the image in each dataset have been captured with the same instrument.

Direct Classification Subsystem

The other subsystem implemented in this work has been named "direct classification subsystem" as it trains a classical CNN without any segmentation process, so the full images are used to train the system by "brute force". The different stages implemented for this subsystem are detailed in order below.

Pre-processing In order to be able to use the dataset images in the classification subsystem we apply process the images in the same way as for the segmentation subsystem but resize images to the 224×224 images for the classification subsystem.

Static Data Augmentation When we are training the system we perform static data augmentation on the training fraction of the combined dataset. This process consists in

producing images with modified brightness or contrast parameters and is very similar to the approach used for the segmentation subsystem.

Dynamic Data Augmentation Static augmentation has proven sufficient in this case and no further improvement was obtained when enabling the dynamic augmentation component.

Classification Network Initially we implemented the classification network using a VGG16 (Simonyan and Zisserman, 2014) pretrained with the ImageNet 1K challenge (Russakovsky et al., 2015) weights. This network has been successfully used by other researchers (Diaz-Pinto et al., 2019) for fundus image classification. This network is relatively large (about 15M parameters) and, thus would make future embeded implementations of our proposed system very difficult. There are, however, newer more efficient alternatives that can lead to similar performance figures. In our case we decided to base our implementation in MobileNet v2. This network is much lighter (less than 2.5M parameters) thus making the embedded implementation of our system feasible. The accuracy of this network on the ImageNet challenge is very similar to that of VGG16, however, its accuracy density, i.e. the accuracy divided the number of parameters is an order of magnitude higher (Bianco et al., 2018).

For our system we remove the top layers of the original MobileNet V2 and add a final classifier based on an average polling layer whose output is flattened and fetched to an 80 node dense layer, a dropout stage and a final 2 node layer to distinguish between the two required classes.

At the top of the model we include an average polling layer, a dense layer with 64 nodes and dropout and a final dense layer with 2 nodes to classify our two classes. This can be seen in Figure 2.10.

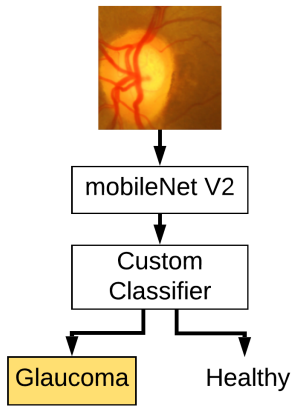


FIGURE 2.10: Classification subsystem

Training Our network is implemented as a recursive function in Keras 2-3-0-tf under Tensorflow 2.2.0. We use 64 image batches as this size is suitable for training using TPUs, GPUs or even CPUs. We use a RMSprop optimizer with initial 1e-3 learning rate with decay and perform training for 50 epochs. This has proven suitable as we are just training the last stages of the Mobilenet V2 network pretrained with ImageNet plus the additional classifier network.

Once both systems obtains information independently, these outputs may be fused in order to obtain the final output of the diagnosis aid tool (as shown in Figure 2.8). This fusion is detailed in the next subsection.

2.4.3 Data Fusion and Report Generation

The final objective of our system is to help the ophthalmologist in his or her diagnosis. Most Machine learning assistance tool are "oracle based" in the sense that they provide a diagnosis with, in the best case a probability estimation on the reliability of the result.

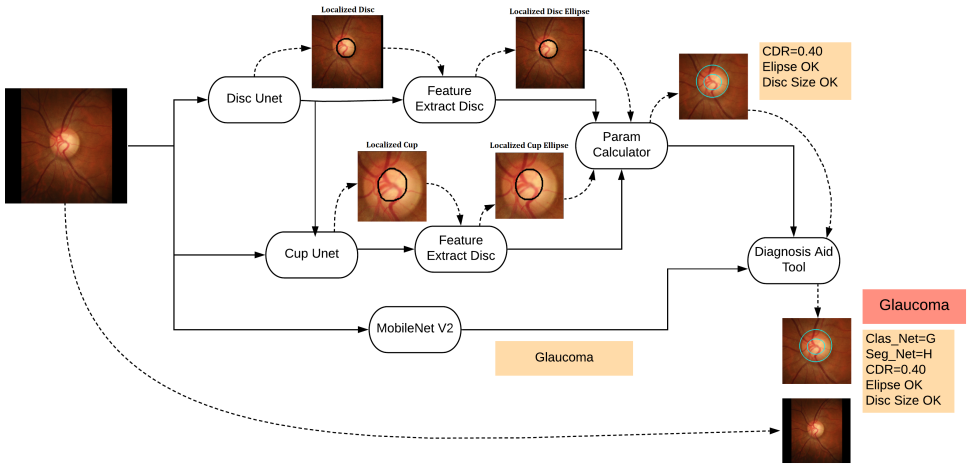


FIGURE 2.11: System diagram with intermediate data and reports.

To be widely accepted by the medical community it is necessary to provide some explanation on the basis on which the result is obtained (Adadi and Berrada, 2018). Our system does not pretend to be a full-flagged glaucoma diagnosis assistance tool but it provides the physician with:

- The result of the classification subsystem with the assigned probability.
- The result of the segmentation subsystem with the associated calculated CDR.

- The accuracy of the ellipse form matching post-processing stage to let the physician know if the forms of the obtained disc and cup are similar to what should be expected.
- The likeness that the size of the disc is correct.

All these aspects are shown in simplified form in Figure 2.11.

2.5 Covid-19 classification Architecture.

2.5.1 Dataset

In this section, the dataset used for this work and the system's architecture are detailed. First, the dataset is presented.

We are using a publicly available dataset with X-ray images from healthy, pneumonia and covid-19 patients publicly available at [2]. The dataset was split for training and assessment using the Hold-out technique, consisting of randomly selecting a sample subset for the training of the models, and using the remaining subset to assess the model performance. A subset with the 80% of dataset samples was used for training, while the remaining 20% subset was used for evaluation. Table 2.2 shows the distribution.

TABLE 2.2: Dataset distribution for each subset.

Subset	COVID-19	Healthy	Pneumonia	Total
Total	132	132	132	396
Training	105	105	106	316
Test	27	27	26	80

Preliminary results using the established data set provided some outliers, with confidence values far removed from the rest of true positives. An analysis of these specific cases established that they were particular X-ray images showing the patient's torso from a lateral perspective. Additionally, the dataset also included few Magnetic Resonance images. Due to the small number of images in the dataset of these two types, and the fact that they are all of the COVID-19 class, a model with so many parameters cannot assimilate and generalize the characteristics necessary to classify them correctly. Considering that the X-rays taken from the front are the most common and that their performance in medical centers do not imply any type of difficulty in relation to other anatomical planes, restricting the use of the model to classify frontal images is not a relevant limitation.

[2] https://public.roboflow.ai/classification/covid-19-and-pneumonia-scans

So, finally those images were not taken into account for training nor for testing, including a pre-processing stage to eliminate them before starting with the training process.

2.5.2 Processing architecture

The architecture used for this work is based on a VGG-16 model trained using TensorFlow with Keras, a pre-processing stage and a final classification using the confidence parameter obtained after the training. This architecture can be observed in Figure 2.12. These stages are detailed below:

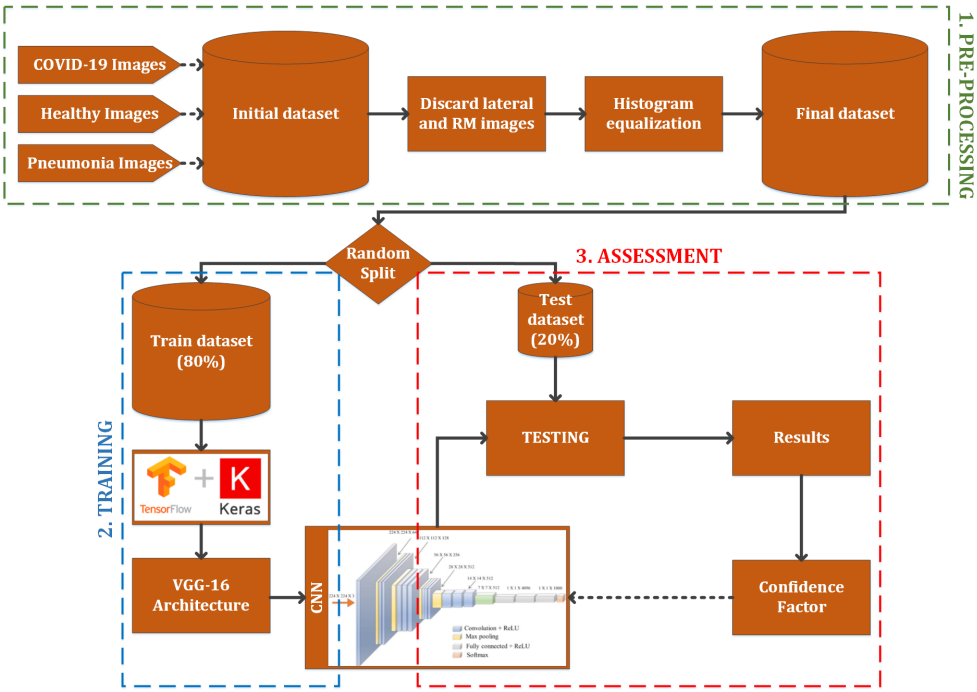


FIGURE 2.12: Processing architecture used in this work.

- Pre-processing: the images stored in the original dataset contains lung X-ray images of healthy patients, patients with pneumonia and COVID-19 positives. However, some images of the COVID-19 positive cases were not obtained with the same parameters as detailed above, so these images must not be taken into account. Moreover, in order to work with images of the same characteristics, an histogram equalization is applied. These two treatments compose the pre-processing stage. The results of the pre-processing step can be observed in Figure 2.13.

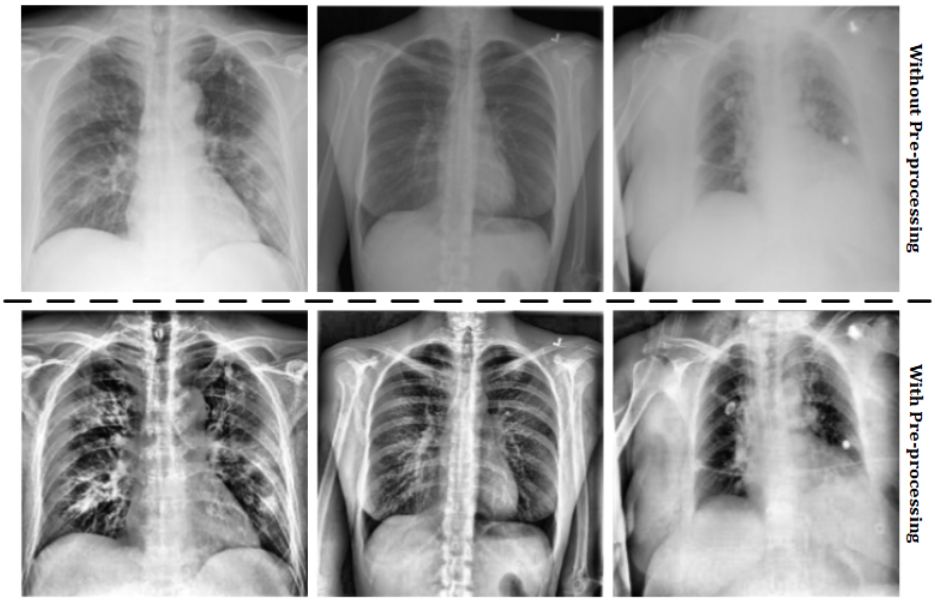


FIGURE 2.13: Pre-processing results.

- Training: using TensorFlow framework with Keras, a VGG-16 architecture (Simonyan and Zisserman, 2014) is implemented and combined with a final inference layer to train a classification system with three classes (healthy, pneumonia and COVID-19). The output of this stage is the convolutional neural network model.
- Assessment: after the model is obtained, the testing dataset is used to evaluate the classification effectiveness, obtaining a confidence factor. This one is used to analyze the CNN performance in order to evaluate the usefulness as a diagnostic tool.

Once the system architecture and the dataset used to obtain the classification mechanism have been specified, the results obtained will be presented in the next section.In this section, the dataset used for this work and the system's architecture are detailed. First, the dataset is presented.

Chapter 3

Results

3.1 Segmentation Architecture Selection

Regarding the Disc segmentation (Table 3.1), for our experiments we initially use a network that is very similar to the original U-Net: 5-stage, no batch normalization and default dropout rates (0.3). We always use transpose convolution, as in the original implementation as direct upsampling is not currently supported on TPUs.

Table 3.1 shows the Dice coefficients for the learning and for the test sets for several evaluated network alternatives. The first row in the table defines the main U-Net architecture parameters, i.e. the network depth (D), the number of filters in the first layer (W), the use of batch normalization and the increment ratio (IR). As an example, 6/40/Y/1.1 means that we use a 6 layer generalized U-Net with 40 channels in the first layer, batch normalization and a 1.1 layer to layer channel increment ratio (IR).

Apart from the base case and its modification including batch normalization, we provide data from pruned networks where we try to obtain the same or greater performance with a smaller number of trainable parameters. To achieve this goal, we decrease the increment ratio while increasing the number of filters in the first layer, the depth of the network or both. The column MTP in Tables 3.1 and 3.2 shows the millions of trainable parameters in the network.

TABLE 3.1: Disc Segmentation results

D/W/BN/IR	Train/Test	Best/Worst/Std.	RRP	MTP
5/32/N/1.5G	84/70	97/55/10	75	3.5
5/32/Y/1.5G	94/91	99/69/7	95	3.5
5/40/Y/1.2G	90/79	98/64/9	95	1.1
6/40/Y/1.3G	95/91	98/64/9	96	3.3
6/40/Y/1.1G	**95/91**	**97/59/9**	**95**	**.9**
7/40/Y/1.2G	95/92	98/61/11	97	2.6
7/64/Y/1.3T	**96/94**	**99/62/8**	**97**	**14**

The Dice coefficient is defined, as usual, as twice the number of active pixels in the intersection of the true and the predicted masks divided by the sum of the active pixels in both masks. In the tables, our Dice coefficient are shown as percentages. For each proposed network architecture, we provide the mean Dice coefficient for the training and testing sets, the Dice coefficient for the best and worst predicted images in the testing set, and the standard deviation for the Dice coefficient over the testing set.

We define a new additional parameter (Radii Ratio parameter- RRP) that is very useful to estimate the accuracy of the CDR. This parameter is defined as the percentage of test images for which the radius of the predicted disc has less than a 10% error when compared with the ground truth mean radius. As an example, when using the deepest network included in the table, for 97% of the images our estimation of the disc radius has an error smaller than 10% (RRP=97).

The base case is the only architecture in the table where we don't use batch normalization. When training on a single dataset, batch normalization has a moderate effect on the network performance. However, when training using multiple datasets, we see a clear overfitting effect when we don't use batch normalization. This can be clearly seen in Figure 3.1, where the learning curve on the left side, which corresponds to a network without batch normalization, clearly overfits the data, while the curve on the right side, which corresponds to the same system with batch normalization, shows much better results for both training and testing sets.

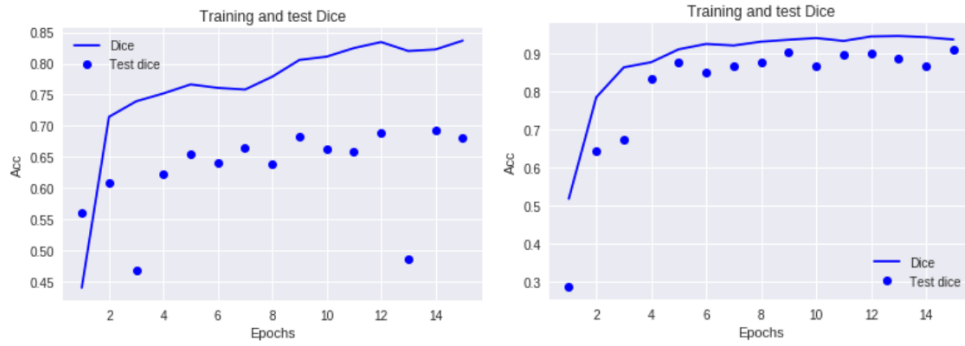


FIGURE 3.1: Batch Normalization effect. The left side learning curve corresponds to a network without batch normalization. The right side one corresponds to the equivalent network with batch normalization.

We can see that deep networks with few parameters like the 6/40/Y/1.1, which has only 917492 trainable parameters, achieve good results for disc segmentation. In this specific case, the CNN achieves a RRP of 95. The best and the worst segmentations for this network are shown in Figure 3.2. This network is highlighted in the table.

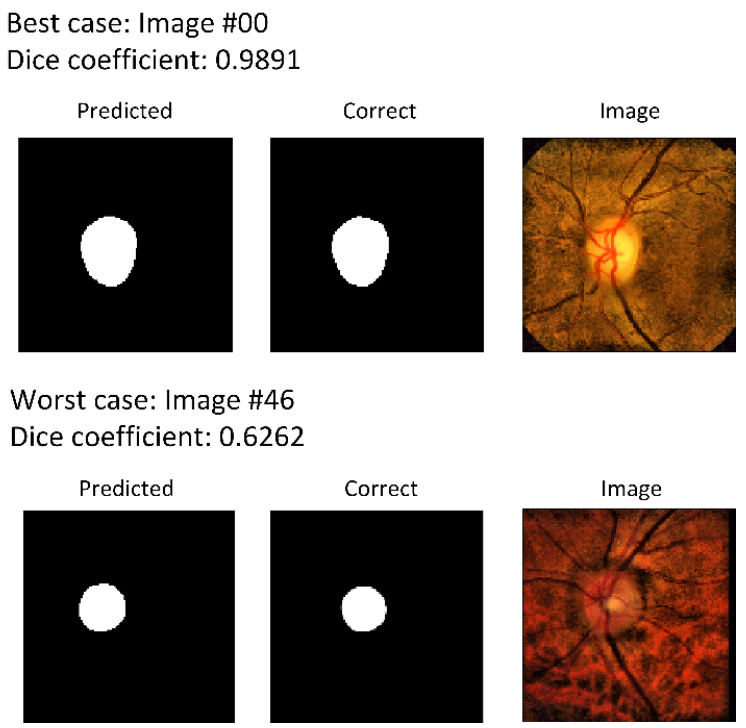


FIGURE 3.2: Best and worst disc segmentation with 6/40/Y/1.1 net

As a reference, we include in Table 3.1 a very wide and deep network (7/64/Y/1.3) which has over 14 million trainable parameters. Although the performance of this network is better than in any other case, the small improvement does not justify the additional complexity of the network. We also highlight this case in the table.

We can consider the effects of dynamic data augmentation by training the system without using the dual image generator. If we only use the images with contrast and brightness modifications, the prediction results are significantly worse. As an example, if we use this approach for our base case (with batch normalization), the worst-case Dice is 38% and the RRP falls to 86% .

We have not included the training time in the table as, in our case, this is almost independent of the network complexity. In all our experiments the training time was between 25 and 28 minutes. This seems to be caused by the dynamic data augmentation implemented in the dual image generator. As TensorFlow TPU support is currently not well documented, we initially considered the possibility that the generator might be running on CPU. The generator must produce the image batches that are used for training and testing.

Currently colab notebooks run on an Intel(R) Xeon(R) CPU @ 2.30 GHz using a single core with two threads [26]. In our training experiments we train for 15 epochs with 150 train and 30 test batches per epoch. As we use 120 image batches, we have to generate 270000 training images and 54000 testing images. This represents less than 5ms per generated image. Changing our code so that we don't use dynamic data augmentation produces only slightly better training times for a similar number of training and testing images. Thus, it seems that the training time is not dominated by this factor as we originally supposed.

We provide a GPU version of our notebook in GitHub to allow the calculation of the TPU/GPU speedup. This speedup has some dependency on the network characteristics. For example, with a 6/40/Y/1.1 network we get a 2.5 training speed improvement, while for a 5/32/Y/1.5 we get a 2.2 speedup. As Keras support for TPUs is in early beta stage, performance comparisons will surely change in the future. Many architectures can't be trained with our default batch size on the Tesla T4 GPUs in Google colab due to memory limitations. In these cases, the speedup training on TPUs can be above 3.0. We finish the architecture name in Tables 3.1 and 3.2 with a T if the architecture must be trained on TPUs to keep the 120 image batches, and with a G otherwise.

As already mentioned in the cup case, we start by selecting the disc area. After this, the segmentation process is identical to the one used for disc segmentation. In Table 3.2 we show the mean Dice coefficient for the training and testing sets, the Dice coefficient for the best and worst image in the testing set, the standard deviation of the Dice on the testing set, and the Radii ratio parameter. The number of trainable parameters in the network is shown as a reference, although this value is clearly the same as for the Table 3.1 for the same network architecture.

TABLE 3.2: Cup Segmentation results

D/W/BN/IR	Train/Test	Best/Worst /Std.	RRP	MTP
4/72/Y/2.0T	98/94	99/60/9	74	44
4/72/Y/1.2T	97/93	99/55/12	74	4.7
4/96/Y/2.0T	98/94	99/58/10	80	78
5/32/N/1.5G	91/85	97/61/8	61	3.5
5/32/Y/1.5G	96/93	99/45/11	72	3.5
5/64/Y/1.3T	***96/90***	***99/53/11***	***77***	***4.9***
6/64/Y/1.1T	96/92	99/56/11	72	2.4
6/64/Y/1.3T	97/93	99/68/9	77	8.5
6/72/Y/1.2T	**97/94**	**99/62/10**	**77**	**5.6**
6/96/Y/1.1T	**97/94**	**99/56/11**	**77**	**5.3**
6/96/Y/1.2T	97/94	99/51/11	78	10.2

We can clearly appreciate the importance of RRP for CDR prediction, as alternatives like the base case don't look too bad from other perspectives but they are not able to predict the radius and, thus, the CDR correctly in a significant portion of the cases.

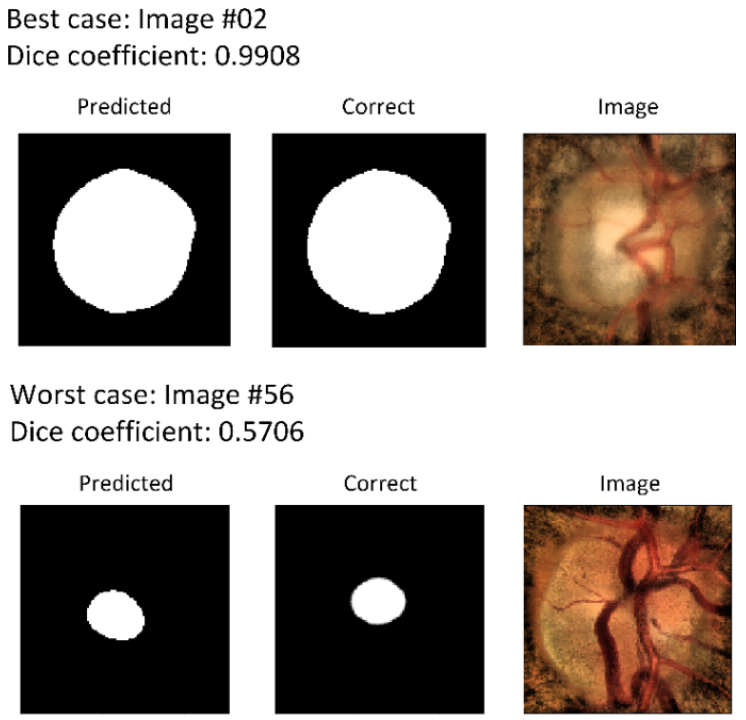


FIGURE 3.3: Best and worst cup segmentation with 4/72/Y/2.0 net

In Figure 3.3 we can see the best and worst prediction using the 4/72/Y/2.0. After consulting with several ophthalmologists, we believe that, in many cases, discrepancies produced by the larger networks correspond to very difficult cases and they are very similar to the discrepancies found when the same images are analyzed by human experts. In Figure 3.4 we can see that even the networks like 4/72/Y/2.0 with over 44 million parameters do not significantly overfit the data.

Some of the most sophisticated models presented in Table 3.1 fail to get good RRPs for the cup case. As an example, the 7/64/Y/1.3 architecture, with over 14 million parameters, only obtains a RRP of 73. Thus, we introduce in Table 3.2 new architectures for cup detection, but try to keep the number of parameters to a reasonable level. Although we have made trials with very different architectures, in general we got interesting results both from wide architectures with high increment ratios and few layers, and from wide and deep architectures with low increment ratios.

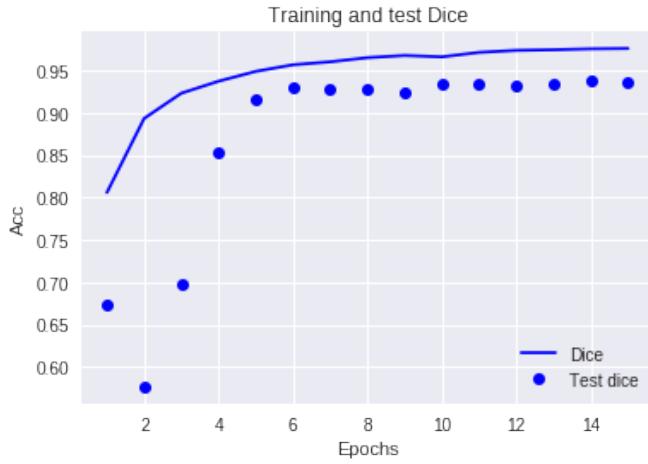


FIGURE 3.4: Learning curve for 44M parameter 4/72/Y/2.0

An important aspect is illustrated in Figure 3.5 Sometimes the networks predict images that, although they are not too bad when measured using Dice or even RRP, are considered as very bad predictions by ophthalmologists.

Our objective is to find an architecture where:

- The worst image Dice is above 55, and the test image subjective quality is acceptable for an expert ophthalmologist.
- At least 75% of the images predict the cup ratio with an error smaller than 10% , i.e. RRP>75.
- The number of parameters is under 6M.

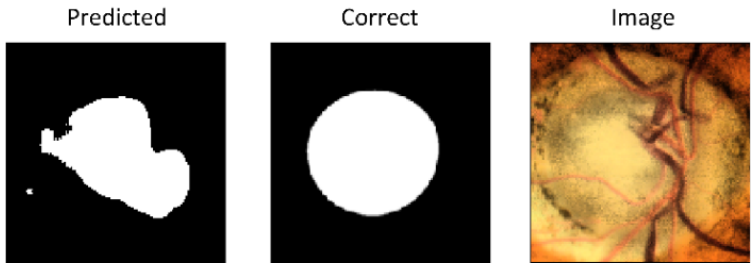


FIGURE 3.5: Bad prediction from 5/64/Y/1.3

These criteria are met by the 6/72/Y/1.2 and 6/96/Y/1.1 networks. The 5/64/Y/1.3 is quite near the required criteria but, as already shown in Figure 10, it does not satisfy

the subjective quality requirement. Both 6/72/Y/1.2 and 6/96/Y/1.1 produce reasonably good results, but the subjective quality of the first alternative is slightly better and, thus, we would recommend this choice.

In Table 3.3, we compare our results with those obtained by other researchers. We only include results that are directly related to our proposed solutions, i.e., those works that perform cup and disc segmentation using a deep learning approach. As in the other tables, the results are presented as percentage Dice coefficients. It is important to note that all the other authors train and test with each specific dataset independently. This approach is not suitable for our objective, i.e. providing segmentation as a cloud-based service. In the table, we show that with our training methodology we can get world-class results over the whole group of datasets.

TABLE 3.3: Comparison with existing methods in the literature. Our work using a combined dataset obtains a dice value of 0.94 for OD and OC segmentation

Author	Method	Cup Drishti	Disc Drishti	Cup RIM-ONE	Disc RIM-ONE	Disc DRIONS
[1]	Ensemble learning CNN (DL)	0.87	0.97	-	-	-
[2]	Fully Conv. DenseNet	0.83	0.95	0.69	0.90	0.94
[3]	Modified U-Net CNN	-	-	0.82	0.94	0.94
[4]	Fully Conv. and adversarial net.	-	-	0.94	0.98	-

[1] (Zilly, Buhmann, and Mahapatra, 2017)
[2] (Al-Bander, B. Williams, et al., 2018)
[3] (Sevastopolsky, 2017)
[4] (Shankaranarayana et al., 2017)

In the case of disc segmentation, we obtain a Dice coefficient of 94%, which is the same as that obtained by [Sevastopolsky, 2017] for RIM-ONE and DRIONS, and by [Al-Bander, B. M. Williams, et al., 2018] for DRIONS, but better than the result of [Al-Bander, B. Williams, et al., 2018] for RIM-ONE (90%), and not as good as the results of [Zilly, Buhmann, and Mahapatra, 2017] and [Al-Bander, B. Williams, et al., 2018] for DRISHTI (97% and 95%) and [28] for RIM-ONE (98%). In the case of cup segmentation, our result (94%) is equal to that obtained by [28] for RIM-ONE, and better than the other results from [Sevastopolsky, 2017], [Al-Bander, B. Williams, et al., 2018] and [Zilly, Buhmann, and Mahapatra, 2017]. This is only a first step, and we should retrain our system when we have further data available from more sources. The high-speed possible with TPU-based training makes this concept feasible in practice.

3.2 Combined Dataset Results with selected Architectures

We want to find out how our system behaves when it is trained with the combined datset and compare these results with those obtained when only a single dataset (i.e. either RIM ONE or DRISHTI) is used to train the system. We will also compare the results to those obtained by other researchers who use a single dataset for both training and validation.

As the measure of the similarity between the correct and predicted disc forms the already mentioned Dice coefficient, also known as F1 score, is used. This figure of merit is widely used and allows us to compare our results with those from other researchers. The Dice coefficient is defined as:

$$DC = \frac{2TP}{2TP + FP + FN} \tag{3.1}$$

In this equation TP indicates true positives, FP false positives, and FN false negatives. The Jaccard index, which is also very widely used in image segmentation can be directly calculated from the DC and thus we don't include JI in our result tables.

In Table 3.4 Disc segmentation for our three different study cases are shown. In the first two we train using just a single data set and validate using the part of that dataset not used for training and the other dataset, while in the last scenario we train and validate with a mixed data set. Our three study scenarios are the following:

- 75% of the DRISHTI dataset is used for training and after validation is carried out first with the rest of DRISHTI data set and then with the full RIM ONE data set.
- 75% of the RIM ONE dataset is used for training and validation is carried out first with the rest of RIM ONE data set and then with the full DRISHTI data set.
- 75% of a mixed data set is used to train the networks and then we validate with the rest of the mixed data set.

We can see in Table 3.4 that, with the generalized 6-layer net in the scenarios where we train with a single dataset, either DRISHTI or RIM ONE, results when testing with images from the same dataset used for training are good with Dice coefficients above 0.98 (DRISHTI) and 0.96 (RIM1) for OD segmentation. However, when we validate these networks with the other data set results are below 0.66 or even below 0.50 in some cases.

In the third scenario where we train with a mixed data set, we get results that are more similar when testing with images coming both datasets. In this case we get a 0.96 Dice coefficient for the DRISHTI test subset and a 0.87 for the RIM ONE subset.

We can see that for the 5-layer network with larger layer increment ratio and similar number of trainable parameters, results are in general very similar.

TABLE 3.4: OD segmentation Dice coefficient

Author	DRI	RIM ONE
(Zilly, Buhmann, and Mahapatra, 2017)	0.97	-
(Al-Bander, B. Williams, et al., 2018)	0.95	0.90
(Sevastopolsky, 2017)	-	0.94
(Shankaranarayana et al., 2017)	-	0.98
Drishti Trained(6L)	0.98	0.50
RIM Trained(6L)	0.66	0.97
Multi-dataset(6L)	0.96	0.87
Drishti Trained(5L)	0.99	0.65
RIM Trained(5L)	0.69	0.98
Multi-dataset(5L)	0.94	0.87

In Table 3.4 we include results from other papers that have performed OD segmentation using Deep Learning methods and have trained with one of the datasets used in our study. All these researcher papers have trained and tested with each of independently. Thus they are related to our first two scenarios but they never test a network trained using images from a data set with images from a different one.

Although we use networks with a small number of trainable parameters, when training with a single dataset we get results that are similar to those obtained by other research papers. When training with the DRISHTI dataset we obtained a Dice value of 0.98 for OD segmentation. This value is slightly above 0.97 (Zilly, Buhmann, and Mahapatra, 2017) . In the RIM ONE trained case we obtain a Dice value of 0.97. This also compares well with 0.98 (Al-Bander, B. Williams, et al., 2018).

The most significant results in table I come from the data that can not be obtained in the other studies. The results obtained when we train with a dataset and predict using data captured with another source show that, in this case, we always get poor prediction results. This demonstrates that it will not be feasible to create a service using training data captured with a single acquisition device.

We also see in Table 3.4 that when training with a combined dataset the network produces good results for both datasets although not as well as when the training and prediction sets are parts of the same global dataset.

The real and the predicted disc shapes are usually not circular but usually approximately elliptical, Usually the ratio of the horizontal cup and disc diameters is somewhat larger than that of the diameters in the horizontal direction (Lingam et al., 2017); however, in most works on the subject (including all those referenced in Table 3.4 the cup to disk ratio is calculated using the mean diameters of the optical cup and the optical disc. Although there are several possible interpretations of the mean diameter (or radius) they have very small differences with real eye fundus data. In this paper we consider

that the mean radius of the optical disc or cup is the square root of its area divided by π.

In Table 3.5 we show the the predictions that estimate the OD radius with an error smaller than 10% as a percentage. This data is relevant fro a clinical point of view as the CDR, the ratio between the cup and disc radii, is a well established glaucoma indicator. The fact that the 5 layer network performs better, when trained with the combined dataset when using this clinically significant parameter would make us choose this network for our web based service.

In the first two scenarios, when we train with a specific dataset, almost all the radii for the testing data from the same dataset are predicted with less than 10% error. However, the radii prediction for the other dataset is much worse and, in some case, we never get errors below 10%. As we can see in Table 3.5 this situation improves very significantly when we train with a mixed dataset.

TABLE 3.5: Radio Ratio Parameter.

	DRI	**RIM ONE**
Drishti Trained(6L)	100	38
RIM ONE Trained(6L)	62	100
Multi-dataset(6L)	100	82
Drishti Trained(5L)	100	25
RIM ONE Trained(5L)	62	100
Multi-dataset(5L)	100	97

3.3 Incremental Training Results with the selected architectures

In this section We want to find out how our system behaves, when training with one set and then retraining lightly with some data from the other, and see if the results similar to those obtained when a single set of data is used (i.e. RIM ONE or DRISHTI) to train the system. Table 3.6 shows the results of disk segmentation for our two cases. On the first train, we use only DRISHTI data and validate using remaining of that data set and RIM ONE. In the second scenario, we make a brief retrain (3 epochs) using RIM ONE and the data set. Our scenarios are defined as follows:

- 75% of DRISHTI is used for training and validation is carried out first with the rest DRISHTI and then with the complete RIM ONE.
- 75% of RIM ONE is used to retrain the network and then we validate with the test part of both sets.

We can see in Table 3.6 that when we train with DRISHTI the tests with images from that same data set obtain very good Dice values. Specifically, we obtain an average Dice

of 0.98 (DRISHTI) but only 0.64 (RIM1). The situation is worse than it seems as in the worst case for some RIM images the segmentation does not produce any pixels.

TABLE 3.6: Segmentation Dice with retraining

Author	DRI	RIM
(Zilly, Buhmann, and Mahapatra, 2017)	0.97	-
(Al-Bander, B. Williams, et al., 2018)	0.95	0.90
(Sevastopolsky, 2017)	-	0.94
(Shankaranarayana et al., 2017)	-	0.98
Drishti Trained	0.98	0.64
RIM Retrained	0.89	0.80

When we retrain the network with the other data set, the Dice values are 0.89 (DRISHTI) and 0.80 (RIM). For the worst case, we get a Dice of 0.69. Therefore, we can see that with a light retraining, the network can quickly learn the specific characteristics of the second data set. In Table 3.6 we include results of the other papers analyzed before.

When we train with a single set of data, we obtain results for that set that are similar to those obtained by other papers. When training with the DRISHTI data set, we obtained a dice value of 0.98 for OD segmentation. This value is slightly above 0.97 Zilly, Buhmann, and Mahapatra, 2017.

As in the previous section the most significant results in Table 3.6 come from what is not available from other studies. The results obtained when we do a quick retraining show that, in this case, we get good prediction results for all test images.

3.4 Ensemble training

In this section we will use Figure 2.11 that shows all the intermediate images and data produced by our system to explain the obtained results and compare them with those from other sources.

3.4.1 Segmentation Subsystem

We compare our Disc and Cup segmentation results with other works that use Deep learning based segmentation and use the same fundus image data sets. We have to take into account two important distinguishing features of our work:

- We want to be independent from the specific characteristics of the capture device and, thus, we train with a combined dataset while the compared works train and test independently with each specific dataset.
- We want our system to be very lightweight to be able to implement it in an embedded system in the future.

In Table 3.7 we show the Dice coefficient scores for disc and cup segmentation from Sevastopolsky (Sevastopolsky, 2017) who uses a very light U-Net and provides results for RIM ONE. We also include results from [Zilly, Buhmann, and Mahapatra, 2017] who use a three-layer CNN including sophisticated pre and postprocessing and apply it independently to both data sets. Al-Bander (Al-Bander, B. Williams, et al., 2018) uses a heavily modified, dense U-Net and provides results for both data sets. Shankaranarayana (Shankaranarayana et al., 2017) uses a residual U-Net and provides results for RIM ONE.

TABLE 3.7: Disc and Cup Dice Coefficients

Author	Method	Cup RIM-ONE	Disc RIM-ONE	Cup Drishti	Disc Drishti
[1]	DenseNet	0.69	0.90	0.83	0.95
[2]	Modified U-Net	0.82	0.94	-	-
[3]	Ensemble learning CNN	-	-	0.87	0.97
[4]	Fully Conv. adversarial net	0.94	0.98	-	-
Our work	Generalized U-Net	0.84	0.92	0.89	0.93

[1] (Al-Bander, B. Williams, et al., 2018)
[2] (Sevastopolsky, 2017)
[3] (Zilly, Buhmann, and Mahapatra, 2017)
[4] (Shankaranarayana et al., 2017)

We can see, that even though we train with a mixed dataset and use very light segmentation networks our results are fully in line with those obtained by other researchers with heavier networks who train and test specifically with each dataset.

After post-processing by preforming the ellipse conversions we see that the mean values are practically identical which shows that ellipse based approximation is a very good option to codify the disc and cup shapes. Only in very few cases the ellipse extracted from the segmented cup or disc differs significantly from the segmentation provided by the U-net. The RANSAC fitter gives us enough information to signal this cases very easily. When this happens we include this information in the final report for the physician so that he or she knows that the segmentation result has less confidence in this case. An example where the extracted ellipse does not fit well enough with the segmentation data is shown in Figure 3.6. This specific case (image G8 from the RIM ONE Dataset) is correctly identified as a glaucoma subject both by segmentation and direct classification. It is clear that, although the predicted ellipse is not as large as the

correct result the calculated CDR (0.6) is enough to classify the image as coming from a Glaucoma patient.

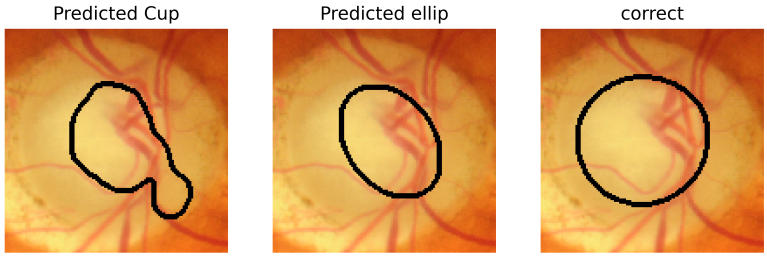


FIGURE 3.6: Case where ellipse feature extraction has low confidence.

Regarding the parameter calculation block we estimate the CDR by the relation between the height of the cup bounding box to the height if the cup image. As we cut the cup image to the Disc bounding box, plus a 10% margin on each border which we take into account, this corresponds to the vertical CDR which is the most widely use CDR version. The typical values of this parameters are 0.65±0.13 for glaucoma patients and 0.39±0.15 for healthy individuals (MacIver, MacDonald, and Prokopich, 2017). Thus a value between 0.52 and 0.54 seems the most adequate for discriminating both cases. Experimentally we find that the value 0.52 produces the best results with our datasets.

Once we have calibrated our CDR based classifier we can analyze the specificity (recall) and sensitivity of our approach. There are many works that segment the optic disc and cup using many different technologies, however, only a few try to use the segmentation data to do real glaucoma predictions. Work [Nayak et al., 2009] was one of the firsts to compute the sensitivity and specificity of their glaucoma predictions using and approach that mixed a morphology based CDR calculation with vessel segmentation in different regions.

TABLE 3.8: CDR based methods sensitivity and specificity.

	Sp	Se
CDR+vessel (Nayak et al., 2009)	0.80	1.00
Watershed (Pinto, 2019)	0.73	0.60
Generalized U-net	0.93	0.76

They reported very good result but based their work in only 15 test cases. On the other hand, the work [Pinto, 2019] used a Stochastic Watershed transformation approach to segmentation with a much larger dataset and obtains a specificity value above 70% with a sensitivity over 60% . When we consider only our segmentation subsystem we get a specificity over 90% with a sensitivity over 75%. It should be clarified that, in

a diagnostic assistance tool, which tries to help in a diagnostic but, not to completely carry out automatic diagnosis the main problem are type II error (i.e. false negatives) where a patient with glaucoma is identified as healthy. Sensitivity, the probability that a person with glaucoma is detected as such, is more important than specificity which is the probability that a healthy patient is detected as such.

Table 3.8 condenses the sensitivity and specificity data for CDR based diagnosis tools. In the table Se stands for sensitivity and Sp for specificity.

In Figure 3.7 we can see the normalized confusion matrix for the U-Net based classifier. We can see that approximately a quarter of the glaucoma cases are classified as healthy using this approach.

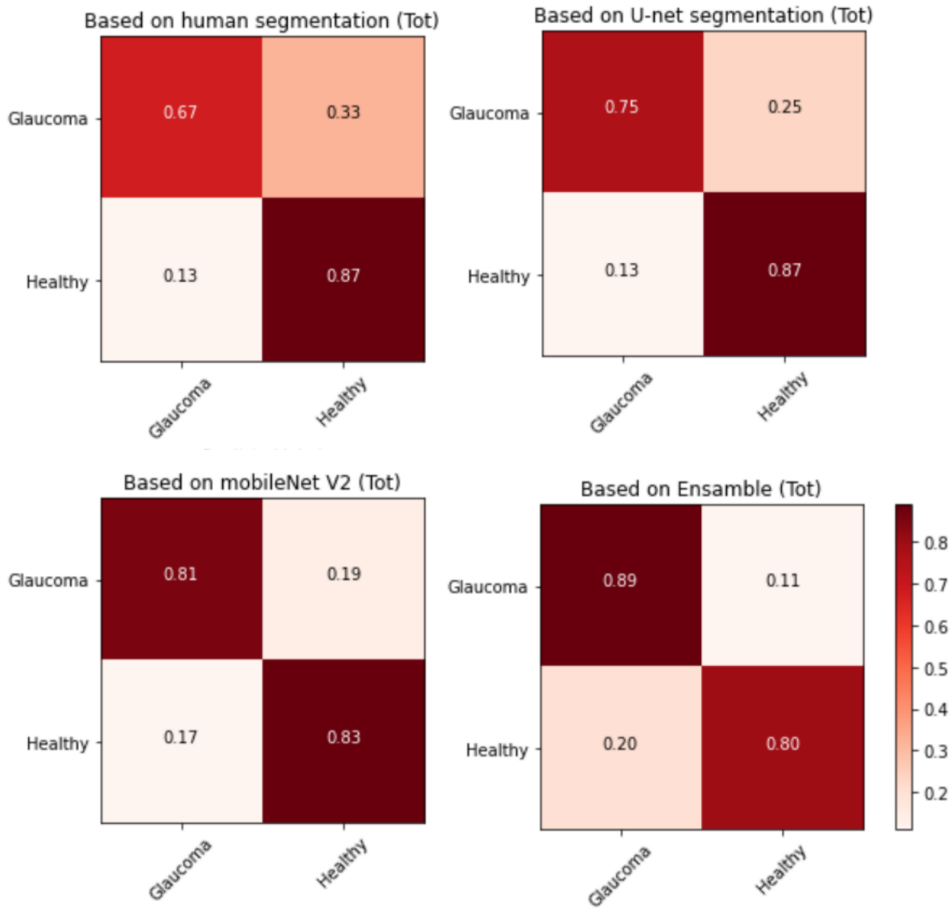


FIGURE 3.7: RIM-one confusion matrices.

The ROC curve for the U-Net classifier is shown in Figure 3.8. The area under the curve is 0.91 which is better than the results obtained in the work [Pinto, 2019] for the CDR based classifier.

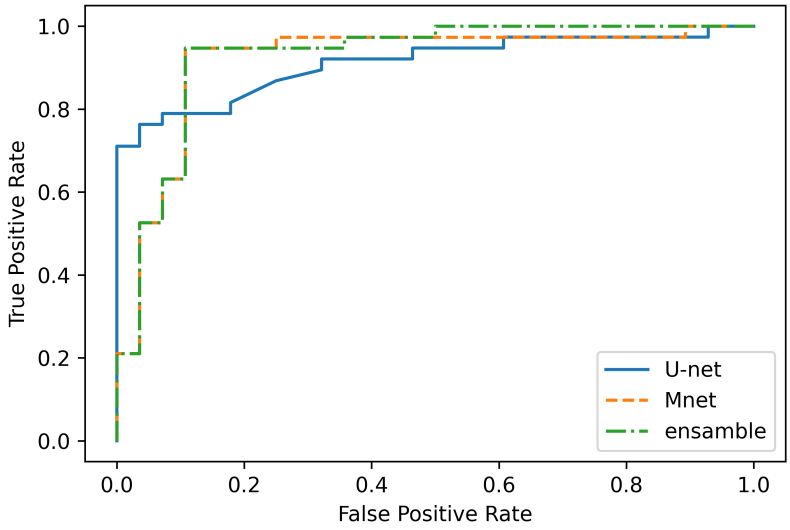


FIGURE 3.8: ROC for Glaucoma Class.

3.4.2 Classification Subsystem.

Our classification subsystem is based on the very lightweight MobileNet V2. In Table 3.9 we compare our system with several classifiers implemented using different networks in work [Pinto, 2019]. The compared networks were VGG16 (Simonyan and Zisserman, 2014), ResNet50 (He et al., 2016) and Xception (François Chollet, 2017).

We can see that our results are comparable, specially regarding sensitivity, with those obtained by implementations that require at least 20 times more computing performance (Bianco et al., 2018).

In Figure 3.7 we can see the normalized confusion matrix for the MobileNet V2 based classifier. We can see that under 20% of the glaucoma cases are classified as healthy using this approach.

The ROC curve for the MobileNet classifier is shown in Figure 3.8. The area under the curve is 0.93 which is somewhat inferior to other possible alternatives. We have to consider, however, that the implementation is much lighter and that it is designed to be part of an ensemble that is planned to deliver good results as a combined network.

3.4.3 Ensemble Network

In Table 3.9 we have included also the specificity and sensitivity of the network combining both the U-Net CDR based pattern extraction classifier and the MobileNet direct classifier. It should be clarified that, as our aim is to build a diagnostic assistance tool and, thus we want to avoid false negatives as much as possible our voting scheme decides that a patient is a glaucoma candidate whenever any of the two networks indicates so. This option improves mainly the network sensitivity. In our case we get a sensitivity of 0.91 which is fully in line with the best available alternatives implemented with a much higher computational cost.

TABLE 3.9: CNN based classifiers Specificity and sensitivity

Network	AUC	Acc	Sp	Se	GF
VGG16	0.96	0.89	0.88	0.90	15
ResNet50	0.96	0.90	0.89	0.91	15
Xception	0.96	0.89	0.85	0.93	10
MobileNetV2	0.93	0.86	0.82	0.89	0.5
Ensamble	0.96	0.88	0.86	0.91	1.5

In Figure 3.7 we could see the normalized confusion matrix for the Ensemble based classifier. We can see that only about 10% of the glaucoma cases are classified as healthy using this approach. This has improved the false negative rate very significantly in comparison with the individual networks that compose the ensemble.

The ROC curve for the Ensemble classifier is shown in Figure 3.8. In a typical ROC curve construction we modify the threshold on the probability of the result belonging to the analyzed class. In our type of ensemble we have two thresholds that can be chosen independently. Thus, to construct the curve we can chose a strictly increasing function that establishes the relation between the classifiers thresholds. In Figure 3.8 we see and example where we use a linear relation to tie both thresholds. Changing this function to a non-linear relation we can obtain almost any curve that is under the union of the curves for both classifiers. The AUC value provided in Table 3.9 is an upper limit on the possible values of AUCs for ROC curves that we could construct for the ens amble.

3.4.4 Reporting Tool

Medical image processing will experiment a breakthrough when ML based diagnostic assistance tools became widely available and accepted in medical daily practice. A problem regarding the adoption of systems is their lack of understandability for the medical professional. This fact has been highlighted by several recent articles, (e.g. [Knight, 2017] and [Michael et al., 2018]) which emphasize the importance of visible (as opposed to black-box) approaches to machine learning based diagnostic assistance.

We do not claim that our tool is a full flagged explainable glaucoma diagnosis aid prototype. However we have done an important effort to provide the ophthalmologist with additional data to be able to judge the validity of the proposed diagnosis. This data (see Figure 2.11) includes information on the adequacy of the size of the segmented disc, the adequacy of the shapes of the disc and the cup, the calculated CDR and the probability of the decision for the direct classification subsystem. We also always provide the initial and the segmented fundus images.

It is clear that understanding our report requires more training than understanding an 'Oracle based' glaucoma or healthy diagnosis but it also gives the physician, who is responsible for the diagnostic decision, much more information on which to base his or her decision.

3.5 Covid-19 Classification Results

For the model proposed in subsection 2.5.2 training was performed with an initial learning rate of 0.001, a batch size of 32 images and 40 epochs. The used optimizer was a Adam with a learning rate decay equal to the initial learning rate divided by the number of training epochs.

3.5.1 Effectiveness Results

We compared the effectiveness using different metrics, distinguishing between micro and macro metrics.

Macro metrics averages the unweighted mean per label. They consists of accuracy, sensitivity (also named macro recall), specificity, macro precision and macro F1-score.

$$Specificity = \sum_{c} \frac{TN_c}{TN_c + FP_c}, c \in classes \tag{3.2}$$

$$Precision_m = \sum_{c} \frac{TP_c}{TP_c + FP_c}, c \in classes \tag{3.3}$$

$$Recall_m(sensitivity) = \sum_{c} \frac{TP_c}{TP_c + FN_c}, c \in classes \tag{3.4}$$

$$F1 - score_m = 2 * \frac{precision_m * recall_m}{precision_m + recall_m}, \tag{3.5}$$

where m index refers to macro metric and $classes = \{COVID-19, healthy, pneumonia\}$. The term TP_c refers to the number of samples with class c that were classified correctly as c. The term FP_c means the set of samples with different class of c that were classified as c by the model. FN_c refers to the set of samples with class c that were classified as

other different class. TN_c indicates the number of samples with a class other than c that were not classified as c.

The results obtained for macro metrics are shown in Table 3.10. Both models presents effectiveness values over 0.85 for each metric. The higher results are the specificity values, which implies that in general the model has a low probability of generating false positives. The rest of the metrics, however, also present high values, denoting a good performance of the models identifying both true positives and negatives.

TABLE 3.10: Results for Macro average metrics.

Model	**Accuracy**	**Precision**	**F1-Score**	**Specificity**	**Sensitivity**
Original	0.86	0.86	0.86	0.93	0.86
Equalized	0.85	0.85	0.85	0.92	0.85

As opposed to macro metrics, micro metrics shows the results averaging the total true positives, false negatives and false positives. The results for each class and model are shown in Table 3.11 and Table 3.12. Overall, both models obtain a high effectiveness in relation to COVID-19. The metrics reveal that the model is quite sensitive to the identification of this disease with this type of images, with a low rate of false negatives. In contrast, the model is less sensitive identifying cases of pneumonia.

TABLE 3.11: Results for micro average metrics for each class (model with original images).

Class	**Precision**	**Recall**	**F1-Score**
COVID-19	0.87	0.96	0.91
Healthy	0.83	0.93	0.88
Pneumonia	0.90	0.69	0.78

TABLE 3.12: Results for micro average metrics for each class (model with equalization).

Class	**Precision**	**Recall**	**F1-Score**
COVID-19	0.84	1.00	0.92
Healthy	0.81	0.81	0.81
Pneumonia	0.90	0.73	0.81

Macro metric results indicate that on average the first model has a slightly better performance. However, the micro average metrics values and the confusion matrices (Figure 3.9) reveal that the model using images without preprocessing achieves a higher

hit rate in healthy individuals. On the other hand, the model that uses equalized images has a higher hit rate in the other two classes. The first model distinguishes better between patients with or without one of the pathologies considered, but it is not as effective as the second model distinguishing between pneumonia and COVID-19 disease.

This fact could, in principle be expected as contrast enhancement increases details in the X-ray image and, in this way, increases the deferential characteristics between different deceases but may create some unexpected details in healthy images that may led to their classification as pathological. Several authors have use adaptive equalization on chest X-ray images (e.g. [Jaeger, Antani, and Thoma, 2011] but they use it a preprocessing step before segmentation and do not try to directly classify the enhanced images.

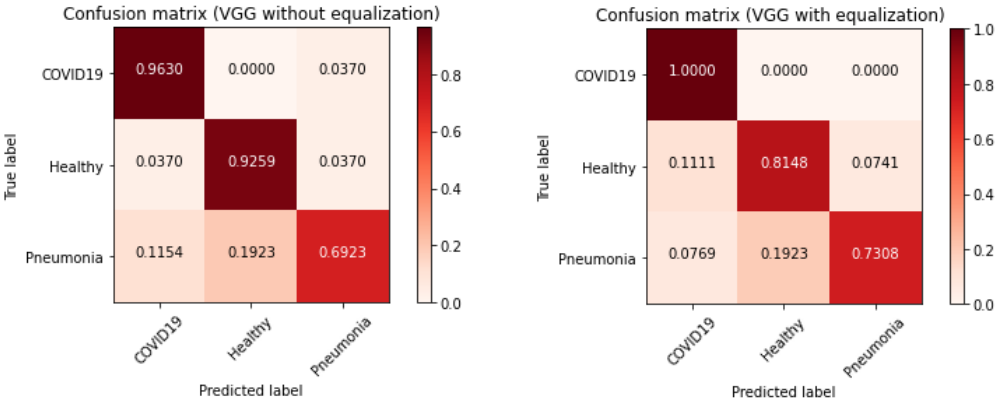


FIGURE 3.9: Confusion matrix of each model.

Given the purpose of the detection system, as a diagnostic support tool, the second model can be considered more suitable, since it is more sensitive to disease identification, with few false negatives for these two classes. False positives could be discarded by a specialist doctor.

Although the numerical results obtained reflect the goodness of the implemented system, it is very interesting to observe the X-ray images that have been used on it to appreciate the similarities and differences between patients with COVID-19, patients with pneumonia and healthy patients. Some of the images used in this work and the classification results of the system can be seen in Figure 3.10.

In Figure 3.10, the first row shows five COVID-19 pulmonary X-Ray images; the second row shows five healthy pulmonary X-Ray images; and the third row shows five pneumonia pulmonary X-Ray images. As can be observed, the first row only contains images that have been classified correctly (remember that COVID-19 class has a 100% success in the classification results, so no mistake has been done in this class). Healthy

and Pneumonia classes do not have a 100% success rate, that is why this figure includes some images that have been wrongly predicted. So, Figure 3.10, includes a subset of the dataset used for this work with all the different cases (positive and negative) in order to show the positive and negative aspects of this classification mechanism.

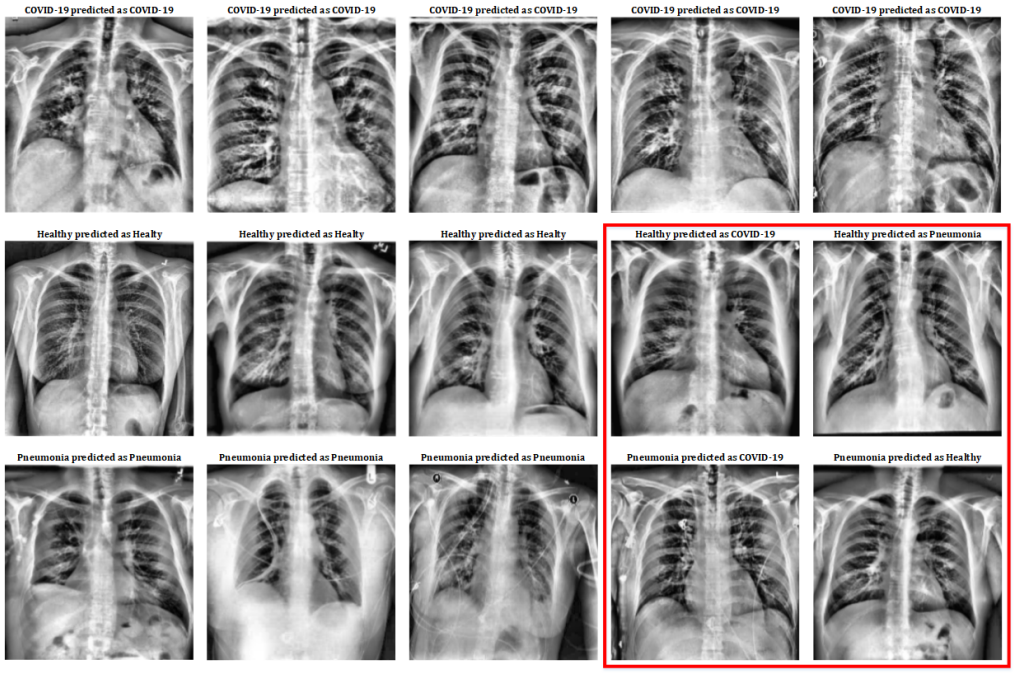


FIGURE 3.10: Classification results on X-ray images.

If we observe deeply Figure 3.10 in order to extract the medical details that cause these classes distinctions, a severe inflammation in the alveoli and bronchioles can be distinguished in images of COVID-19 patients; this is related to the damage that these patients suffer in their lungs. As for healthy patients, both the alveoli and bronchioles are less inflamed. Finally, those patients with pneumonia show appreciable inflammation too, but not as marked as in patients with COVID-19.

Even so, in the red box some erroneously classified cases can be seen. Among these cases, images of healthy patients are shown who, due to inflammation in the lungs (without becoming serious) are erroneously classified as patients with pneumonia or with COVID-19. On the other hand, some of the images of patients with pneumonia have also been misclassified: in some cases, they are mild pneumonia that is classified as healthy; and, in other cases, they have a more severe pneumonia that is erroneously classified as COVID- 19.

However, two important aspects should be highlighted in these results: on the one hand, the images of patients with COVID-19 are correctly classified at 100%; and, on the other hand, the images used of patients with pneumonia come from a previous study (older database) and, therefore, were not taken with current instruments (and, in some cases, with a different zoom). This last aspect may be the trigger for why the pneumonia class has been the worst performer.

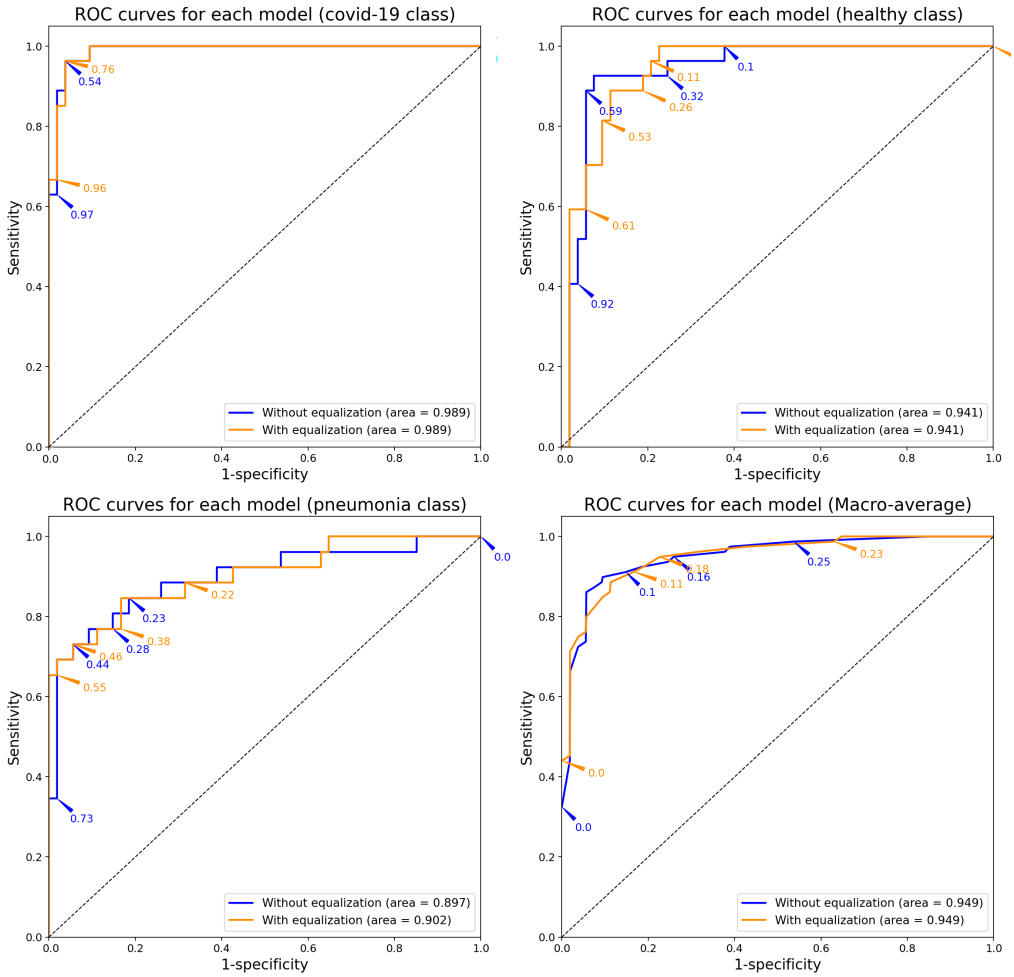


FIGURE 3.11: ROC curves of each model.

The Receiver Operating Characteristic (ROC) curves (see Figure 3.11) per each model and class reveal a good reliability in the classification. These curves were obtained from the results for each node of the output layer by changing the confident threshold. The

Areas Under the Curve (AUC) are higher than 90%. The pneumonia class is the one with the lowest confidence index. However, the trained model with previous treatment of the images shows a higher confidence index for the identification of COVID-19. The ROC curve together with the rest of the results previously shown reveal that the model has great sensitivity regarding the classification of this disease.

Chapter 4

Discussion

In this chapter we will analyze to what extend the result obtained in Chapter 3, that have fulfilled the Thesis objectives established in Section 2.1. In the next sections we will cover the different objectives and analyze in which section of Chapter 3 they have been covered and to what extend have their objectives been reached.

4.1 Feasibility of Segmentation as a Service

This objective has been covered mainly in the works [Javier Civit-Masot, Luna-Perejon, et al., 2019], [Civit-Masot et al., 2020] and [J. Civit-Masot et al., 2020]. We will discuss the specific objectives in the following subsections.

4.1.1 Segmentation Architecture Selection

This was one of the main objectives of the work [Javier Civit-Masot, Luna-Perejon, et al., 2019] and our later work is based on this publication. In this, and all our later works, we use generalized U-Net architectures for fundus image segmentation. In all cases we use both on line and off line data augmentation.

The obtained results are mainly covered in Section 3.1. The results related to the different U-net architectures used for Disc Segmentation can be found in Table 3.1. In this case we can see that many possible configurations provide good results for disc segmentation.

The results related to cup segmentation can be found in Table 3.2. In this case we can see that cup segmentation is more difficult than disc segmentation and requires heavier architectures to obtain good results.

4.1.2 Tuning & Pruning

In Section 3.1, which is based on the work [Javier Civit-Masot, Luna-Perejon, et al., 2019], we see that for disc segmentation we can get good results with a 6/40/Y/1.1

U-net that has under 1M trainable parameters. As a reference, the original U-Net (Ronneberger, Fischer, and Brox, 2015) has over 100M trainable parameters.

Regarding Cup segmentation in the work [Javier Civit-Masot, Luna-Perejon, et al., 2019] we settled for 6 layer networks with more filters in the first stage and around 5M trainable parameters as some of the simpler networks with good quantitative result produced segmented images which were not acceptable to the ophthalmologists as can be seen in Figure 3.5.

In Section 3.2 which is based on the work [Civit-Masot et al., 2020] we settle for the already mentioned 6/40/Y/1.1 U-net network and use it only for disk segmentation.

In Section 3.4.1 we specifically wanted to use the same network for cup and disk segmentation. In this case we used a 6/64/Y/1.1 with around 2.4M trainable parameters. Although this network was originally discarded in Section 3.1, the post-processing approach used in the work [J. Civit-Masot et al., 2020], where segmented disc and cup shapes are always elliptical, solved the qualitative problems that we originally had with this network.

4.1.3 Single Dataset Performance

This aspect has been analyzed in the work [Civit-Masot et al., 2020]. In Table 3.4 we can see that in the case of disc segmentation if we train and test with images from the same dataset we can get very good Dice coefficient values. In some cases this values can be above 0.99, however if we use this networks make predictions with images captured with different instruments the Dice values will be under 0.70 in most cases. Thus, it is very clear, that we must train with, at least, some images captured with the same instrument that we will later use if we want to have reasonable segmentation results.

4.1.4 Combined Dataset Performance

This aspect has been studied in the works [Javier Civit-Masot, Luna-Perejon, et al., 2019; Civit-Masot et al., 2020; J. Civit-Masot et al., 2020]. We will discuss only the results presented in Section 3.4.1 which correspond to the work [J. Civit-Masot et al., 2020]. We can see in Table 3.7 that our results, both for cup and disc segmentation are fully in line with those obtained the works [Al-Bander, B. M. Williams, et al., 2018; Sevastopolsky, 2017; Zilly, Buhmann, and Mahapatra, 2017; Shankaranarayana et al., 2017] which, in all cases, train and test with the same dataset. Thus we can see that if we train with a combined dataset we can get good prediction results with images captured with the different instruments included in the training set.

4.1.5 Incremental Training performance

This aspect has been studied in the work [Civit-Masot et al., 2020] and is covered in Section 3.3. In Table 3.6 we can see that, in the disc segmentation case, if we train with the DRISHTI dataset we get a Dice coefficient of 0.98 for images from that dataset while

this value falls below 0.65 when we try to predict using images from another dataset. If we perform a very quick 3 epoch retrain the Dice value for DRISTI lowers to around 0.90 but the Dice for the RIM ONE datset improves to 0.80.

Incremental training, which is just a variation of transfer learning, is essential if we want to implement segmentation as a service. Thus, the results of Section 3.3 are very important to our work as they show, at least in a preliminary fashion, that we can train our system with the initially available data and do quick retrains when data from new instruments becomes available.

4.2 Lightweight Image Classification

Both in the works [Javier Civit-Masot, Luna-Perejón, et al., 2020] and [J. Civit-Masot et al., 2020] we implement medical image classification using transfer learning techniques. The first work is associated to a glaucoma detection application scenario while the second is associated to a COVID-19 and pneumonia classification scenario.

4.2.1 Classification Architectures

The COVID-19 classification network result are presented in Section 3.5 and its architecture is presented in Section 2.5.2. In this case we use a classical VGG-16 network initially trained with Imagenet data. The confussion matrices for this network are shown in Figure 3.9 where we can see that no covid-19 patient is classified as healthy and only a very small percentage are classified as regular pneumonia.

The classification architecture in Subsection 3.4.2 uses a much newer network, mobileNetV2 also originally trained with Imagenet data. Its confussion matrices are shown in Figure 3.7.

4.2.2 Tuning and Selection

Although the networks in Sections 2.5.2 and 3.4.2 have been both used for medical image classification they are related to very different classification scenarios. The newer mobileNetV2 has around 2.5M parameters while the older vgg has above 15M parameters. In unpublished test using the covid-19 scenario we have been able to show that the performance of both networks is very similar and, thus, mobileNetV2 is, in general a much better architectural choice.

4.3 Segmentation and Classification Ensemble

This section is based on the work [J. Civit-Masot et al., 2020] where the classification system discussed in Section 3.4.2 and the segmentation system discussed in Section 3.4.1 are integrated to improve the diagnostic aid tool performance.

4.3.1 Approaches to Glaucoma detection from segmented fundus images

In Subsection 1.2.1 we discuss several approaches for Glaucoma detection using segmented fudus images. Both ISTN rule and a Cup to Disc ratio based approach are implemented in our current (June 2020) version of the diagnostic aid.

4.3.2 Methodology Selection

In the work [J. Civit-Masot et al., 2020] we implement a Glaucoma classification approach based on CDR. The results of this subsystem are described in Subsection 3.4.1.

The CDR approach present the advantage of its numerical nature as larger CDR values are associated with larger glaucoma probabilities. Thus we can obtain the ROC curve for this type of classificator as can be seen in Figure 3.8. The area under the curve for this classifier is better than that the CDR classifier in the work [Pinto, 2019] which is not based in deep learning.

4.3.3 Ensemble Fusion

The final ensamble network in our diagnostic aid is described in Subsection 3.4.3.

There are several possibilities when combining the results of an ensemble. In medical diagnostic aids it is important to reduce the number of false negatives thus, in our case, we choose an approach where the ensemble suggests a glaucoma diagnosis when either the segmentation based classifier or the direct classifier indicate a glaucoma diagnostic.

4.3.4 Ensemble Performance

In Figure 3.7 we provide the confusion matrices for a system based on the CDR values derived from human expert segmentation, the CDR values from the segmentation subsystem described in Subsection 3.4.1, the classification subsystem is described in Subsection3.4.2 and the ensamble constructed with both subsystems. We can clearly see that the ensemble approach clearly improves the performance of the diagnostic aid.

In Table 3.9 we present the area under the curve, sensitivity, specificity and required number of GigaFlops (GF) for different classifier alternatives. We can clearly see that the ensemble classifier clearly outperforms alternatives that require 10 times more processing power.

4.4 Reporting Tool Feasibility

The reporting tool is an essential part of any realistic diagnosis aids. Most current research works provide just an oracle based diagnostic suggestion where they indicate the

phisician what is the suggested diagnosis giving, at most a probability score associated with the suggestion.

As discussed in the work [Montes-Sanchez et al., 2020] most physicians consider that this approach is not adequate in real clinical practice and consider that the information that supports the decision should be provided by the tool.

4.4.1 Diagnostic Information Selection

In Figure 2.11 we provide an example of the information that our diagnostic aid provides to the physician. This includes:

- The final diagnostic suggestion.
- The raw fundus images.
- The diagnostic suggestions from the segmentation and the classification subsystems with their associated probabilities.
- The initial acceptability of the segmentation disc and cup shapes.
- The acceptability of the segmented disc size.

All this information is presented in an easy to read format and is very easy to understand by a human expert after minimal training. Thus our system can not be considered a full flagged explainable tool but clearly represents a step in this much required direction.

Chapter 5

Conclusions and Future work

5.1 Main Section 1

Deep learning based diagnosis aid tools are going to be part of the physician daily life in the near future. In this thesis we demonstrate that a lightweight tool which can be implemented in an embedded system can provide results at the same level of other tools that require much higher computing performance.

We have been able to show that by using data from different datasets, doing adequate image pre-processing and performing very significant data augmentation (both off line and on line), we have been able to perform cup and disc segmentation getting results with a similar performance to that obtained by other authors using a single dataset for evaluation and testing. This is, at least, a first approximation to the possibility of running this type of segmentation as a service on the cloud.

The use of a generalized parameterizable recursive U-net model allows to easily train and test any U-Net configuration. This allows a much greater flexibility for testing different architectures. Training on Google Cloud TPUs has allowed to test many different configurations of these networks, training them in a time almost independent of the network architecture. To our best knowledge, this is the first time that a U-Net architecture has been tested on TPUs. The speedup obtained with TPUs makes this implementation very attractive for systems like those we propose, where periodic retraining is required.

Many U-Net architectures have been proven adequate for optic disc segmentation. As an example, we have shown that both a trimmed standard U-Net and a deeper lightweight derivative can perform as well as other heavier alternatives for OD segmentation. However, only a small number of alternatives have provided good quality cup segmentation while keeping the number of network parameters at reasonable levels.

We have defined a new clinically significant parameter (Radii Ratio parameter- RRP) that can be useful to estimate the quality of the CDR estimations and thus, to give some confidence on the quality of the system for glaucoma prediction.

This work has also shown the importance of data augmentation as seen on the work [Zoph et al., 2019] and the significance of using a dataset that combines data from different sources. In a real life web service scenario, we would have to start training with the initially available data and retrain our system when more image data from different hospitals becomes available. It is necessary to study the behavior of this type of retrained network with previously trained data and new additional datasets. The possibility of improving the network architecture by the inclusion of residual blocks (Xiuqin et al., 2019) or the combination of a these blocks and a conventional U-Net (Kim et al., 2018) has been shown effective in several medical segmentation applications. The robustness of these networks when analyzing images from different instruments is an open issue for the future.

Our finally implemented tool (in the work [J. Civit-Masot et al., 2020]) is based on an ensemble which includes two subsystems using completely different technologies. The first of the subsystems is a segmentation based network based on the work [Javier Civit-Masot, Luna-Perejon, et al., 2019] plus a feature extraction post processing stage. The second subsystem is based on a very lightweight last generation classification network similar to that used in the work [Javier Civit-Masot, Luna-Perejón, et al., 2020] which is able to provide the same level of performance as other more traditional heavier networks.

A very important part of our system is the reporting tool which combines the output of both networks and provides the physician with enough data to understand the system's diagnosis proposal and, thus, be able to use it adequately in his or her own final decision. The importance of these type of explainable tools is described in the work [Montes-Sanchez et al., 2020] and will surely become widely used in the near future.

There are plenty of possibilities for expanding this work and using it to build a useful medically acceptable Glaucoma diagnostic assistance tool. First we would need to train the ensembles with more data coming from public and private datasets. Including a second lightweight classification subsystem (possibly based on EfficientNet (Tan and Le, 2019). This would improve the reliability and sensitivity of the results even further.

For our alternative covid-19 diagnosis aid application scenario (detailed in the work [Javier Civit-Masot, Luna-Perejón, et al., 2020]) a Deep-Learning classification system based on a particular convolutional neural network model (VGG16) has been trained and assessed to identify symptoms of pneumonia and COVID-19 patients. This network is heavier than the classification network used in the work [J. Civit-Masot et al., 2020] and provides similar performance. The covid-19 scenario was implemented before the Glaucoma classification and that is the reason for the improved implementation in this last scenario.

The database used in the covid-19 scenario is a combination of Healthy, Pneumonia and COVID-19 X-ray images from patients around the globe of both genders with different ages, and it is growing up day by day. The inputs used to train and test the system are those lung radiographs and the outputs is a classification between Pneumonia, COVID19 or Healthy, as well as a confidence value.

A pre-processing stage was preformed to all the X-ray images as they had been obtained from different machines with different calibrations, which caused a significant variation in the histogram of the images. Later a complete study of the system performance was carried out. The results indicate that the proposed model behaves well discriminating healthy cases when a contrast enhancement technique is applied prior to training. In fact, 100% of the COVID-19 cases were successfully classified, while the other two classes obtained very satisfactory good results.

So, the model has a high sensitivity regarding the identification of COVID-19 and a remarkable specificity with respect to the three classes. This turns the model into a well-behaved tool to screen cases and support diagnosis.

Chapter 6

Bibliography

Bibliography

Adadi, Amina and Mohammed Berrada (2018). "Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160.

Alom, Md Zahangir et al. (2018). "The history began from alexnet: A comprehensive survey on deep learning approaches". In: *arXiv preprint arXiv:1803.01164*.

Asri, Hiba et al. (2016). "Using machine learning algorithms for breast cancer risk prediction and diagnosis". In: *Procedia Computer Science* 83, pp. 1064–1069.

Al-Bander, Baidaa, Bryan M Williams, et al. (2018). "Dense fully convolutional segmentation of the optic disc and cup in colour fundus for glaucoma diagnosis". In: *Symmetry* 10.4, p. 87.

Al-Bander, Baidaa, Bryan Williams, et al. (2018). "Dense fully convolutional segmentation of the optic disc and cup in colour fundus for glaucoma diagnosis". In: *Symmetry* 10.4, p. 87.

Bianco, Simone et al. (2018). "Benchmark analysis of representative deep neural network architectures". In: *IEEE Access* 6, pp. 64270–64277.

Bourne, Rupert RA (2006). "The optic nerve head in glaucoma". In: *Community Eye Health* 19.59, p. 44.

Carmona, Enrique J et al. (2008). "Identification of the optic nerve head with genetic algorithms". In: *Artificial Intelligence in Medicine* 43.3, pp. 243–259. ISSN: 0933-3657.

Cascella, Marco et al. (2020). "Features, evaluation and treatment coronavirus (COVID-19)". In: *Statpearls [internet]*. StatPearls Publishing.

Choi, Yunjey et al. (2018). "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797.

Chollet, Francois (2016). "Building powerful image classification models using very little data". In: *Keras Blog*.

— (2017). *Deep Learning with Python*. 1st. Greenwich, CT, USA: Manning Publications Co. ISBN: 1617294438, 9781617294433.

Chollet, François (2017). "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258.

Civit-Masot, J. et al. (2020). "Dual Machine-Learning system to aid Glaucoma Diagnosis using disc and cup feature extraction." In: *IEEE Access* 8, pp. 127519–127529.

Civit-Masot, Javier, Francisco Luna-Perejon, et al. (2019). "TPU Cloud-Based Generalized U-Net for Eye Fundus Image Segmentation". In: *IEEE Access* 7, pp. 142379–142387.

Civit-Masot, Javier, Francisco Luna-Perejón, et al. (2020). "Deep Learning System for COVID-19 Diagnosis Aid Using X-ray Pulmonary Images". In: *Applied Sciences* 10.13, p. 4640.

Civit-Masot, J et al. (2020). "Multidataset Incremental Training for Optic Disc Segmentation". In: *Proceedings of the 21st EANN (Engineering Applications of Neural Networks)*. Cham: Springer-Nature.

Coiera, Enrico (2018). "The fate of medicine in the time of AI." In: *Lancet (London, England)* 392.10162, pp. 2331–2332.

Collobert, Ronan et al. (2011). "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.ARTICLE, pp. 2493–2537.

Das, Pranjal, SR Nirmala, and Jyoti Prakash Medhi (2016). "Diagnosis of glaucoma using CDR and NRR area in retina images". In: *Network Modeling Analysis in Health Informatics and Bioinformatics* 5.1, p. 3.

Diaz-Pinto, Andres et al. (2019). "CNNs for automatic glaucoma assessment using fundus images: an extensive validation". In: *Biomedical engineering online* 18.1, p. 29.

Dice, Lee R (1945). "Measures of the amount of ecologic association between species". In: *Ecology* 26.3, pp. 297–302.

Dominguez-Morales, Manuel J et al. (2019). "Smart Footwear Insole for Recognition of Foot Pronation and Supination Using Neural Networks". In: *Applied Sciences* 9.19, p. 3970.

Dong, Ensheng, Hongru Du, and Lauren Gardner (2020). "An interactive web-based dashboard to track COVID-19 in real time". In: *The Lancet infectious diseases*.

Fischler, Martin A and Robert C Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6, pp. 381–395.

Fletcher, KH (1951). "Matter with a mind; a neurological research robot." In: *Research; a journal of science and its applications* 4.7, p. 305.

Fu, Kun et al. (2018). "WSF-NET: Weakly supervised feature-fusion network for binary segmentation in remote sensing image". In: *Remote Sensing* 10.12, p. 1970.

Fumero, Francisco et al. (2011). "RIM-ONE: An open retinal image database for optic nerve evaluation". In: *2011 24th international symposium on computer-based medical systems (CBMS)*. IEEE, pp. 1–6. ISBN: 1457711907.

Guil, Nicolas and Emilio L Zapata (1997). "Lower order circle and ellipse Hough transform". In: *Pattern Recognition* 30.10, pp. 1729–1744.

He, Kaiming et al. (2016). "Identity mappings in deep residual networks". In: *European conference on computer vision*. Springer. Cham, pp. 630–645.

Howard, Andrew G et al. (2017a). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

— (2017b). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

Jaeger, Stefan, Sameer Antani, and George Thoma (2011). "Tuberculosis screening of chest radiographs". In: *SPIE Newsroom*.

Jhuo, Sing-Ling et al. (2019). "Trend prediction of influenza and the associated pneumonia in taiwan using machine learning". In: *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, pp. 1–2.

Jouppi, Norman et al. (2018). "Motivation for and evaluation of the first tensor processing unit". In: *IEEE Micro* 38.3, pp. 10–19.

Ker, Justin et al. (2017). "Deep learning applications in medical image analysis". In: *Ieee Access* 6, pp. 9375–9389.

Kim, Sewon et al. (2018). "Fine-grain segmentation of the intervertebral discs from MR spine images using deep convolutional neural networks: BSU-Net". In: *Applied Sciences* 8.9, p. 1656.

King Jr, Bernard F (2017). *Guest editorial: discovery and artificial intelligence*.

Knight, Will (2017). "The dark secret at the heart of al". In: *Technology Review* 120.3, pp. 54–61.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

Lan, Lan et al. (2020). "Positive RT-PCR test results in patients recovered from COVID-19". In: *Jama* 323.15, pp. 1502–1503.

Lauer, Stephen A et al. (2020). "The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application". In: *Annals of internal medicine*.

Li, Qun et al. (2020). "Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia". In: *New England Journal of Medicine*.

Lin, Zhi Qian et al. (2015). "Chest X-ray and CT findings of early H7N9 avian influenza cases". In: *Acta Radiologica* 56.5, pp. 552–556.

Lingam, Christopher Leung et al. (2017). "4. RISK FACTORS (OCULAR)". In: *Diagnosis of Primary Open Angle Glaucoma: WGA consensus series-10* 10, p. 127.

Litjens, Geert, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sanchez (2017). "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42, pp. 60–88. ISSN: 1361-8415.

Litjens, Geert, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez (2017). "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42, pp. 60–88.

Liu, Xiaoxuan et al. (2019). "A comparison of deep learning performance against healthcare professionals in detecting diseases from medical imaging: a systematic review and meta-analysis". In: *The Lancet Digital Health* 1.6, e271–e297. ISSN: 2589-7500. DOI: https://doi.org/10.1016/S2589-7500(19)30123-2. URL: http://www.sciencedirect.com/science/article/pii/S2589750019301232.

Luna-Perejon, Francisco, Manuel Jesus Dominguez-Morales, and Anton Civit-Balcells (2019). "Wearable fall detector using recurrent neural networks". In: *Sensors* 19.22, p. 4885.

MacIver, S, D MacDonald, and C Lisa Prokopich (2017). "Screening, Diagnosis, and Management of Open Angle Glaucoma". In: *Canadian Journal of Optometry* 79.1, pp. 5–71.

Michael, K Yu et al. (2018). "Visible machine learning for biomedicine". In: *Cell* 173.7, pp. 1562–1565.

Montes-Sanchez, Juan Manuel et al. (2020). "An Approach to Explainable AI for Digital Pathology". In: *12th International Conference on eHealth Telemedicine, and Social Medicine (eTELEMED 2020)*. Ed. by J Civit-Masot S Sendra Y Murata and A. Rajh. Valencia, Spain: International Academy, Research, and Industry Association, pp. 110–115.

Nayak, Jagadish et al. (2009). "Automated diagnosis of glaucoma using digital fundus images". In: *Journal of medical systems* 33.5, p. 337.

Organization, World Health et al. (2020). *Modes of transmission of virus causing COVID-19: implications for IPC precaution recommendations: scientific brief, 27 March 2020*. Tech. rep. World Health Organization.

Pinto, Andrés Yesid Dıaz (2019). "Machine learning for glaucoma assessment using fundus images". PhD thesis. Universitat Politècnica de València, Departamento de Ingeniería Electrónica.

Quigley, Harry A and Aimee T Broman (2006). "The number of people with glaucoma worldwide in 2010 and 2020". In: *British journal of ophthalmology* 90.3, pp. 262–267.

Repici, Alessandro et al. (2020). "Coronavirus (COVID-19) outbreak: what the department of endoscopy should know". In: *Gastrointestinal endoscopy*.

Reza, Ali M (2004). "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement". In: *Journal of VLSI signal processing systems for signal, image and video technology* 38.1, pp. 35–44.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. Cham, pp. 234–241.

Rothan, Hussin A and Siddappa N Byrareddy (2020). "The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak". In: *Journal of autoimmunity*, p. 102433.

Ruder, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.

Russakovsky, Olga et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3, pp. 211–252.

Sandler, Mark et al. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

Serebriakova, OM et al. (2012). "Peculiarities of clinical and X-ray picture of pneumonia in patients with influenza A (H1N1)". In: *Klinicheskaia meditsina* 90.6, pp. 70–72.

Sevastopolsky, Artem (2017). "Optic disc and cup segmentation methods for glaucoma detection with modification of U-Net convolutional neural network". In: *Pattern Recognition and Image Analysis* 27.3, pp. 618–624. ISSN: 1054-6618.

Shankaranarayana, Sharath M. et al. (2017). "Joint Optic Disc and Cup Segmentation Using Fully Convolutional and Adversarial Networks". In: *OMIA 2017*. Fetal, Infant and Ophthalmic Medical Image Analysis. Cham: Springer International Publishing, pp. 168–176. ISBN: 978-3-319-67561-9.

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

Sivaswamy, Jayanthi et al. (2014). "Drishti-gs: Retinal image dataset for optic nerve head (onh) segmentation". In: *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*. IEEE, pp. 53–56. ISBN: 1467319619.

Sohrabi, Catrin et al. (2020). "World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19)". In: *International Journal of Surgery*.

Tan, Mingxing and Quoc V Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *arXiv preprint arXiv:1905.11946*.

Topol, Eric J (2019). "High-performance medicine: the convergence of human and artificial intelligence". In: *Nature medicine* 25.1, pp. 44–56.

Tse, G MK et al. (2004). "Pulmonary pathological features in coronavirus associated severe acute respiratory syndrome (SARS)". In: *Journal of clinical pathology* 57.3, pp. 260–265.

Wei, Gu-Yeon, David Brooks, et al. (2019). "Benchmarking tpu, gpu, and cpu platforms for deep learning". In: *arXiv preprint arXiv:1907.10701*.

Xie, Xuanyang et al. (2006). "Mining x-ray images of SARS patients". In: *Data Mining*. Springer, pp. 282–294.

Xiuqin, P. et al. (2019). "A Fundus Retinal Vessels Segmentation Scheme Based on the Improved Deep Learning U-Net Model". In: *IEEE Access* 7, pp. 122634–122643. DOI: 10.1109/ACCESS.2019.2935138.

Xu, Zhe et al. (2020). "Pathological findings of COVID-19 associated with acute respiratory distress syndrome". In: *The Lancet respiratory medicine* 8.4, pp. 420–422.

Zhang, Luxia et al. (2018). "Big data and medical research in China". In: *bmj* 360.

Zilly, Julian, Joachim M Buhmann, and Dwarikanath Mahapatra (2017). "Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation". In: *Computerized Medical Imaging and Graphics* 55, pp. 28–41.

Zoph, Barret et al. (2019). "Learning Data Augmentation Strategies for Object Detection". In: *arXiv preprint arXiv:1906.11172*.

Appendix A

TPU Cloud-Based Generalized U-Net for Eye Fundus Image Segmentation

Information

- **Authors:** Javier Civit-Masot, Luna-Perejon, Vicente-Diaz, Corral, and Civit
- **Details:**
  - **Title:** IEEE Access
  - **Type:** Journal
  - **Editorial:** IEEE
  - **Date:** September 2019
  - **Volume:** 7
  - **Pages:** 142379 - 142387
  - **DOI:** https://doi.org/10.1109/ACCESS.2019.2944692

Appendix B

Multidataset Incremental Training for Optic Disc Segmentation

Information

- **Authors:** Civit-Masot, Billis, Dominguez-Morales, Vicente-Diaz, and Civit
- **Details:**
  - **Title:** 21th Engineering Applications of Neural Networks (EANN)
  - **Type:** Congress Proceedings
  - **Editorial:** IEEE
  - **Date:** June 2020
  - **Volume:** 2
  - **Pages:** 365 - 376
  - **DOI:** https://doi.org/10.1007/978-3-030-48791-1_28

Appendix C

Dual Machine-Learning System to Aid Glaucoma Diagnosis Using Disc and Cup Feature Extraction

Information

- **Authors:** J. Civit-Masot, Dominguez-Morales, Vicente, and Civit
- **Details:**
  - **Title:** IEEE Access
  - **Type:** Journal
  - **Editorial:** IEEE
  - **Date:** July 2020
  - **Volume:** 8
  - **Pages:** 127519 - 127529
  - **DOI:** https://doi.org/10.1109/ACCESS.2020.3008539

Appendix D

Deep Learning System for COVID-19 Diagnosis Aid Using X-ray Pulmonary Images

Information

- **Authors:** Javier Civit-Masot, Luna-Perejón, Domınguez Morales, and Civit
- **Details:**
  - **Title:** Applied Sciences
  - **Type:** Journal
  - **Editorial:** MDPI
  - **Date:** July 2020
  - **Volume:** 10
  - **Pages:** 4640
  - **DOI:** https://doi.org/10.3390/app10134640