UNIVERSIDADE D
COIMBRA

Patrícia Nobre Silva

AUTOMATIC DETECTION OF CATARACT IN FUNDUS IMAGES

Dissertation in the context of the Integrated Master in Engineering Physics - Instrumentation branch, supervised by Professor Dr. Luís Alberto da Silva Cruz and Engineer Nuno Filipe Girão Almeida, presented to the Physics Department at University of Coimbra.

September 2019

UNIVERSIDADE D COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Patrícia Nobre Silva

AUTOMATIC DETECTION OF CATARACT IN FUNDUS IMAGES

Dissertation presented to the
Physics Department at University of Coimbra
to obtain the Master's degree in Engineering Physics

Supervisors:
Prof. Dr. Luís Alberto da Silva Cruz (University of Coimbra)
Eng. Nuno Filipe Girão Almeida (Retmarker S.A.)

Jury:
Prof. Dr. António Miguel Lino Santos Morgado (University of Coimbra)
Prof. Dr. João Pedro de Almeida Barreto (University of Coimbra)
Prof. Dr. Luís Alberto da Silva Cruz (University of Coimbra)

Coimbra, 2019

This work was developed in collaboration with:

Retmarker S.A.

Abstract

Opacities found in fundus images are an obstacle to the detection of other anomalies related to eye diseases, sometimes blocking diagnoses. They interfere with other eye lesions diagnosis software, degrading image quality and visibility.

An automatic approach to the detection and quantification of cataracts would be very helpful to the ophthalmological field to avoid the mentioned problem and to enable analysis of large amounts of images in an easier, faster and inexpensive way, without needing the expertise of trained specialists to do so.

This work presents two approaches to detect eye cataracts, one based on "general" Machine Learning and the other on Deep Learning.

To enhance visibility and contrast, two pre-processing methods of the retinal images are explored and compared. Three Machine Learning algorithms (Support Vector Machine, Decision Tree and Bagged Trees) that classify each image into one of two classes - cataract or no cataract - were used. Feature extraction is based on a Discrete Wavelet Transform, more specifically, the Haar Transform.

A method based on Convolutional Neural Networks is also introduced, which uses the same dataset to train and test a similarly aimed Deep Learning classifier.

Keywords: automatic approach, Bagged Trees, cataract, Convolutional Neural Network, Decision Tree, Deep Learning, fundus image, Machine Learning, opacity, Support Vector Machine, Discrete Wavelet Transform

Resumo

As opacidades encontradas em retinografias são um obstáculo à deteção de outras anomalias relacionadas com doenças visuais, impedindo por vezes diagnósticos. Estas interferem com outros softwares de diagnóstico de lesões do olho, degradando a qualidade e a visibilidade das imagens.

Uma abordagem automática à deteção e quantificação de cataratas, seria muito útil para o campo da oftalmologia para evitar o problema mencionado e para que fosse possível analisar uma maior quantidade de imagens de uma maneira mais fácil, rápida e barata, sem que fosse necessário o conhecimento de um especialista na área para o fazer.

Este trabalho apresenta duas abordagens para detetar cataratas no olho, uma baseada em Aprendizagem Máquina "geral" e outra em Aprendizagem Profunda.

Para aumentar a visibilidade e o contraste, dois métodos de pré-processamento das imagens da retina são explorados e comparados. Três algoritmos de Aprendizagem Máquina (Máquina de Vetores de Suporte, Árvore de Decisão e Árvores "Bagged") que classificam cada imagem numa de duas classes - catarata ou sem catarata - são usados. A extração de características é baseada na Transformada em Ondeletas Discreta, mais especificamente, na Transformada de Haar.

Um método baseado em Redes Neuronais Convolucionais é também apresentado, usando o mesmo conjunto de dados para treinar e testar um classificador em Aprendizagem Profunda para a mesma função.

Palavras-chave: abordagem automática, Árvore de Decisão, Aprendizagem Máquina, Aprendizagem Profunda, catarata, Árvores "Bagged", Máquina de Vetores de Suporte, opacidade, Rede Neuronal Convolucional, retinografia, Transformada em Ondeletas Discreta

Acknowledgments

This project represents the end of five very challenging years. During this time, the following people played important roles in my education and to them I am very thankful.

To my supervisor Professor Luís Cruz for the valuable guidance throughout this project and for being available and present, welcoming me into his laboratory in the Department of Electrical and Computer Engineering of the University of Coimbra.

To my other supervisor, Engineer Nuno Almeida, that was always there to help me and with who I learned so much during the last year.

To João Diogo Ramos and everyone else at Retmarker, for making me feel welcome and supporting my work.

A great thank you to my family, specially my parents and my brother, that made this journey easier. Without their support, the last five years would not have been possible.

To my boyfriend that has always been by my side through the good and the bad, believing in me even when I did not believe in myself.

Last but not least, to everyone else that was a part of my academic journey, thank you for all the teachings, guidance, companionship and support. I will cherish the memories of my student years forever.

Contents

List of Figures

List of Tables

List of Acronyms

AC Accuracy.
AI Artificial Intelligence.
AMD Age-Related Macular Degeneration.
ANN Artificial Neural Network.

BN Bayesian Network.
BT Bagged Trees.

CNN Convolutional Neural Network.

DL Deep Learning.
DR Diabetic Retinopathy.
DT Decision Tree.
DWT Discrete Wavelet Transform.

GC Green Channel.

LOCS Lens Opacities Classification System.

ML Machine Learning.

NN Neural Network.

OD Optic Disk.

PCA Principal Component Analysis.

RGB Red, Green and Blue.

SE Sensitivity.
SP Specificity.
SVM Support Vector Machine.

1

Introduction

Opacities found in fundus images are an obstacle to the detection of other anomalies related to eye diseases. As they degrade image quality and visibility, they interfere with software that detects eye lesions in this type of images, such as microaneurysms, sometimes blocking diagnoses. It is then important that an automatic approach is developed to detect these cataracts.

Modern medical science benefits greatly from the development of technology, specially in image analysis. Nowadays, computerized systems that integrate medical devices are improving health care quality and productivity [9]. Regarding automatic diagnosis, Machine Learning has been widely applied in different diseases such as glaucoma, breast cancer and diabetes [31]. From that we can imagine the potential of applying state-of-the-art techniques to cataract detection in retinal imaging.

Traditional cataract diagnosis present low efficiency with the increasing number of patients [6], making trained professionals scarce resources and invalidating the usual methods. Besides, clinical grading is quite subjective since it depends on the individual and its experience. Therefore, "reducing costs and simplifying the process for early cataract diagnosis is a crucial means of improving eye care" [38]. Consequently, it is consistent from the social and economic points of view to develop automatic cataract detection systems [40].

The main requirements of such a system are that it should be able to take retinographies as input and result in a binary classification – cataract or no cataract. It is important to be a rapid, repeatable and accurate analysis, so it can give trustworthy results in real time and detect change. In essence, such an automated approach would have a function to extract features and an image classification algorithm capable to analyze fundus images from different angles, that would run on a PC and be compatible with other similar diagnosis tools, especially the ones from Retmarker S.A..

This method would not require pupil dilation to obtain the retinal image or the expertise of trained eye professionals, presenting itself as an interesting solution especially in places where other exams are not possible due to the lack of experts or resources. Used as an early detection approach, it would be a powerful way to accurately and objectively diagnose thousands of patients, detecting the disease efficiently and preventing it from progressing and turning into blindness. It would be possible to guide risk patients for referral to further care.

A system like this would also be useful to detect images with opacities, cataractous or not, preventing different software of misdiagnosing other eye conditions because the image is not clear enough or good to be analyzed. The truth is, if a patient has a cataract and some other eye lesion (as long as it is not something urgent resulting from some accident of sorts), the cataract is the main problem. It needs to be treated, which, in this case, means surgically removed, as fast as possible. Only then other screening can be done to access the remaining problems. Therefore, it is easy to understand the importance and practical application of such approach.

Benefits from a method like this would be compatibility and integration with screening of different retinal diseases, as well as solving the interference caused by the unhealthy fundus images with other software, since the problematic ones would be filtered out.

Essentially, the motivation of this work is to develop an automatic cataract classification system for fundus images so that cataractous patients can receive preliminary diagnosis and get help timely, conveniently and even remotely. Meanwhile, hospitals can focus more on cataract treatment instead of early screening.

The question that remains is, is this really a reliable possibility? Retinal imaging has been widely used in clinical applications, but automatic detection of cataract based on it was only proposed in recent years.

Document Structure

This document encloses a total of 5 chapters organized as follows:

The present chapter is Chapter 1, which includes the Introduction to this work.

Chapter 2 contains details about the Background Information and State of the Art in the areas of knowledge involved into this work. It describes the human eye, vision impairment, fundus imaging, cataract, diagnosis and treatment of the disease. It also discusses Artificial Intelligence, Classical Machine Learning, Deep Learning and

their differences. Lastly, a compilation of research done in the field of automatic cataract classification is presented, explaining the work of the authors and results obtained, in chronological order.

In Chapter 3 one can find the methods used to achieve the proposed goal. Procedures like the pre-processing of the input images and the feature extraction will be thoroughly described, as well as the dataset, learning models and classifiers, transfer learning, performance parameters used and the coding structure.

Chapter 4 will address the results obtained, while Chapter 5 presents the conclusions of the work.

2

Background Information and State of the Art

2.1 The Eye

The human eye is the organ responsible for the sense of sight, allowing us to perceive the shapes, colors and dimensions of our surroundings by processing light. It "is mostly optically transparent, allowing a window into both the central nervous system along with the systemic vasculature" [14].

The eye is a complicated structure composed of interconnected subsystems [24], which can be seen in Figure 2.1. The front part, visible by the naked-eye, includes the colored part called **Iris**, the **Cornea**, a clear dome over the Iris, the black round opening in the Iris, which is the **Pupil**, the **Sclera** that is the white part, and, finally, the **Conjunctiva**, a thin layer of tissue that covers the entire front of the eye, except the Cornea [34].

Light reflects off objects and when these are in the field of vision, it enters the eye. First, the light is focused by the clear front surface, the **Cornea**. After, it passes the **Aqueous Humor**, a transparent watery fluid that circulates throughout the front part of the eye and keeps the pressure inside constant. Then, the **Iris** controls the amount of light that reaches the back of the eye by automatically adjusting the size of the **Pupil**. The **Crystalline Lens** further focuses the light, adjusting shape depending on whether the light reflects off something near or far. The light then pierces the center of the eye, through a clear gel known as **Vitreous Humor** [17]. The focused light beam reaches the **Retina**, the light-sensitive inner lining of the back of the eye, which converts optical images into electrical impulses. The **Optic Nerve** then transmits these signals to the visual cortex, which is the part of the brain that controls our sense of sight [29], allowing us to see.

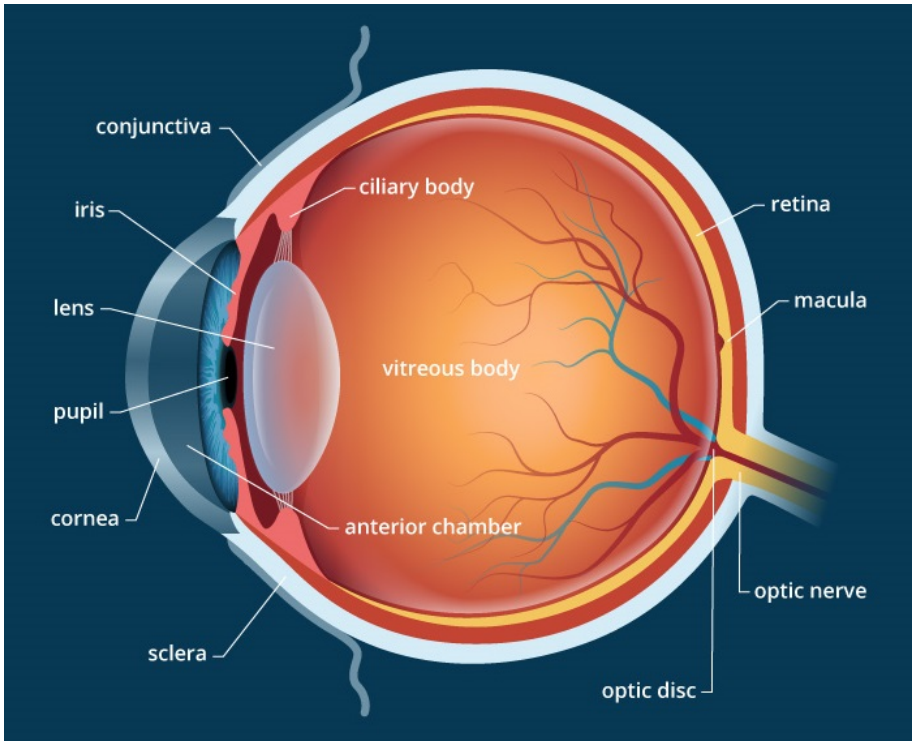

Figure 2.1: Schematic of the anatomy of the human eye, seen from above [29].

The Retina, which is the most important structure in this case, has different components. The **Optic Disk (OD)** is a round disc that connects with brain nerves [39]. The vast **Blood Vessels** bring nutrients to the nerve cells, they are divided into arteries, veins and capillaries and all converge into the OD. The **Macula** is a small yellowish extra-sensitive area at the center of the Retina, allowing central vision [34], whose darker center is called **Fovea**. As it is the focal point of the eye, it has special light-sensitive nerve endings called photoreceptors. These can be rods or cones and convert the light into electrochemical signals [17].

Vision Impairment

Fact sheets posted by the World Health Organization in 2018 state that approximately 1.3 billion people live with some form of vision impairment and around 80% of it is considered avoidable [36], this is why prevention is of such importance.

They also affirm that, currently, the most prevalent causes of vision impairment worldwide are Uncorrected Refractive Errors and Cataracts. Besides these, there are also diseases like Age-Related Macular Degeneration (AMD), Glaucoma, Diabetic Retinopathy (DR), Corneal Opacity and Trachoma, most of them manifest themselves in the retina [9]. Cataracts are more prevalent in low and middle-income

countries, whereas the others are the most common diseases in high-income societies [36]. This work will focus only on cataracts.

2.1.1 Fundus Imaging

The retina is visible with proper illumination due to the fact that the anterior portion (the front part) of an healthy eye is optically transparent. However, properties of some eye parts make direct inspection of the retina difficult. Therefore, fundus imaging is complicated because illumination and imaging beams cannot overlap, or reflections from the cornea and lens will worsen image quality and contrast. This type of imaging solves this using separate paths in the pupil plane, more specifically, an outer illumination beam and an inner imaging beam [14].

A fundus image, also called retinography or retinal image, consists of a photograph of the back of the eye, the posterior segment, taken by a fundus camera. It can be seen in Figure 2.2 a scheme showing the elements of the retina in a fundus image.

Fundus photography is a cost-effective and simple technique. It can be examined at another location or time by specialists and provides photo documentation for future reference [2].

"One fundus image contains plenty of information to reflect the health of the eyes and the body to some extent, playing an important role on the diagnosis and treatment

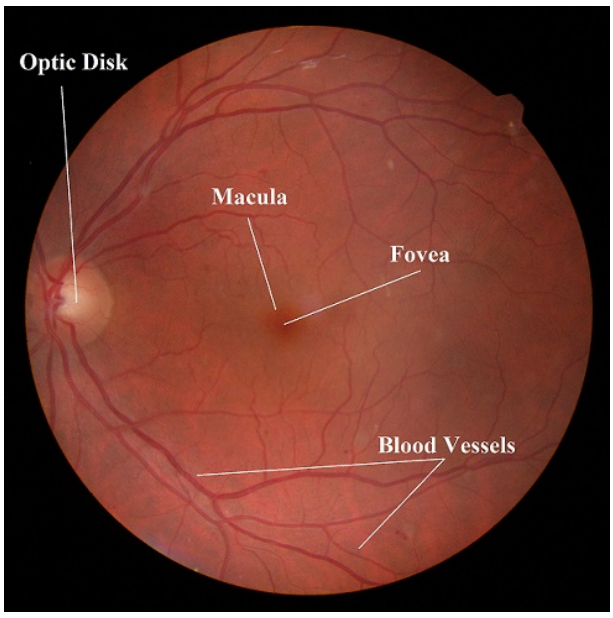

Figure 2.2: Components of a retinal image.

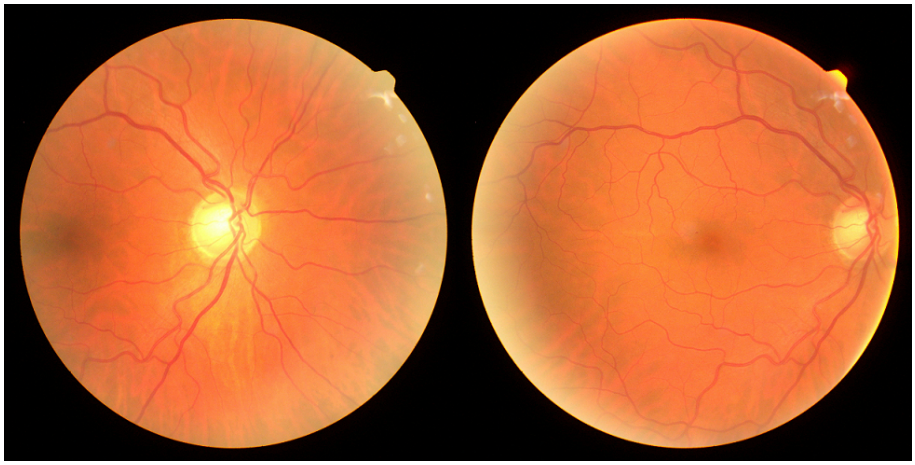

Figure 2.3: Fundus images with and without the OD centered.

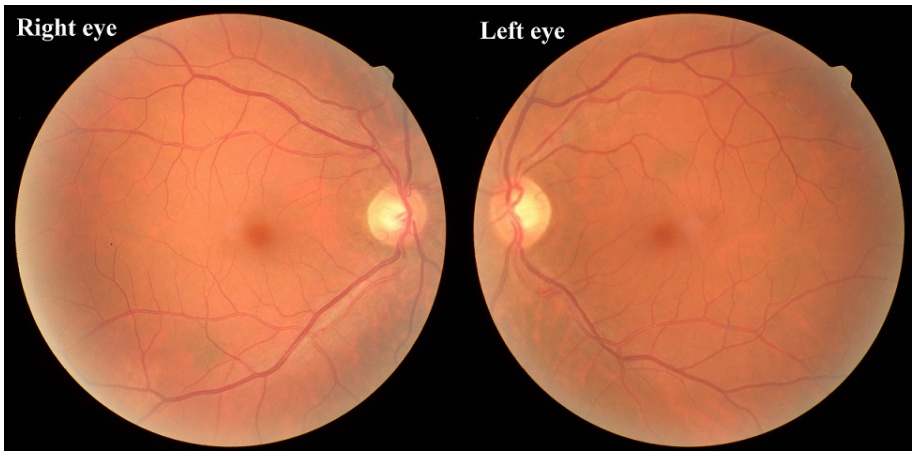

Figure 2.4: Fundus images of the right and left eyes, respectively.

of some diseases" [31].

The ophthalmological field had been seeking to examine the retina for a long time. Several scientists from many areas of study worked towards what we have today.

"From the initial photograph of the human retina in the $19th$ century until today, there have been huge advances in ophthalmic imaging" [14]. In 1910, Allvar Gullstrand, a swedish ophthalmologist and optician, developed the first fundus camera, which is widely used nowadays to accomplish retinal imaging, great for examinations and population screening programs [9]. This type of cameras can take fundus photographs of the eye in different angles, as shown in Figure 2.3, in the first image the OD is centered unlike the second one. In addition, it is also easy to distinguish between right and left eye images where the OD is not centered, noticing which way it is "turned", as can be seen in Figure 2.4.

Due to being safe, harmless and cost-effective, retinal images are used by ophthal-

mologists for screening certain eye diseases. From it, doctors can detect eye diseases such as cataract, glaucoma and DR, plus they can examine the current condition of the patient and predict the visual acuity based on the clearness degree of the image. Other changes are crucial for predicting hypertension and cardiovascular diseases, such as vessel width, tortuosity and branching angle [39], for example.

The research and study of the analysis of fundus image have been broad in the last two decades. There are reports of segmentation and localization of retinal structures, as well as retinal lesions and aneurysms [9]. Based on these techniques, researchers are developing diagnostic systems for retina diseases and lesions, including microaneurysms, DR, AMD, glaucoma and cardiovascular diseases [38].

A more widespread adoption of retinal imaging is expected in the clinical practice for early identification of several chronic diseases and long-term conditions, to aid medical experts, decreasing associated care costs and also facilitate the establishment of large-scale computer aided screening and prevention programs [2].

Cataracts occur in the crystalline lens of the eye, not in the retina. However, the lens dulling reduces the light that focuses on the retina, degrading the quality of the fundus image [9], which makes this type of images suited for the task of cataract classification and grading.

2.1.2 Cataract

A cataract is a painless eye disease caused by protein denaturation that results in an opacification (dulling or clouding) of the lens inside the eye that develops gradually. This phenomenon blocks the light from passing clearly through the lens and leads to a decrease in vision, resulting in poor visual acuity and even blindness at later stages [6]. Nowadays, as said before, it is one of the most common causes of visual impairment globally and its incidence rate increases with age.

This disease interferes with the subject's daily routine, worsening his life quality. As the cataract progresses, it can change the eye's ability to focus [32].

As it is mentioned in Allen *et al.* 's research [1], if affects simple things we take for granted, like color awareness, changes in contrast, handling brightness, driving, reading and even recognizing faces, since it blurs the vision. For this reason, it is encouraged to be early detected and treated.

There are three main types of cataracts: **Nuclear Sclerotic**, **Cortical** and **Posterior Subcapsular**. During the aging process, people can develop either one of

these or a combination of them [32].

1. **Nuclear Sclerotic Cataract** is the most common type and is caused by the hardening and yellowing of the lens nucleus. It progresses slowly and may require many years of gradual development before it begins to affect vision [32].
2. **Cortical Cataract** refers to opacities in the peripheral edge of the lens. "Changes in the water content of the lens fibers create fissures that cause the light that enters the eye to scatter, creating problems with blurred vision, glare, contrast and depth perception" [32]. People with diabetes are at risk for developing this type of cataract.
3. **Posterior Subcapsular Cataract** begins as a small clouding on the back surface of the lens but develops fast, within months. It forms beneath the lens capsule, "a small membrane that encloses the lens and holds it in place" [32]. This cataract affects vision around lights and diabetics and people who have extreme myopia can develop it.

To detect cataract by inspection of a fundus image it is important to check for the clearness or clarity of the structures of the retina, specifically for the details, or lack of, due to blurriness. If the OD, the macula and all vessels are clear, the retinography is normal. If the capillary are not clear, it is defined as a mild cataract. If all the vessels are not clear, it is a medium cataract. Lastly, if all the structures are not visible or just barely visible, it consists of a severe cataract [39]. Basically, as the visibility decreases, the severity of cataract increases, it is based on this fact that automatic methods are explored.

2.1.2.1 Diagnosis

Clinical grading of cataract is performed by comparing the observed picture with a set of standard photos with different cataract severities.

The diagnosis is usually performed by doing a slit-lamp examination by experienced ophthalmologists and its diagnosis is based on the clearness degree of the retina, that can be subjective and prone to error. Then it is classified with the Lens Opacities Classification System (LOCS) [9], seen in Figure 2.5, the Wisconsin Cataract Grading System, the Wilmer Scale, the Oxford Clinical Cataract Classification or the American Cooperative Cataract Research Group [38], to name a few.

Besides slit-lamp, there are already other methods such as Light-Focus, Iris Image

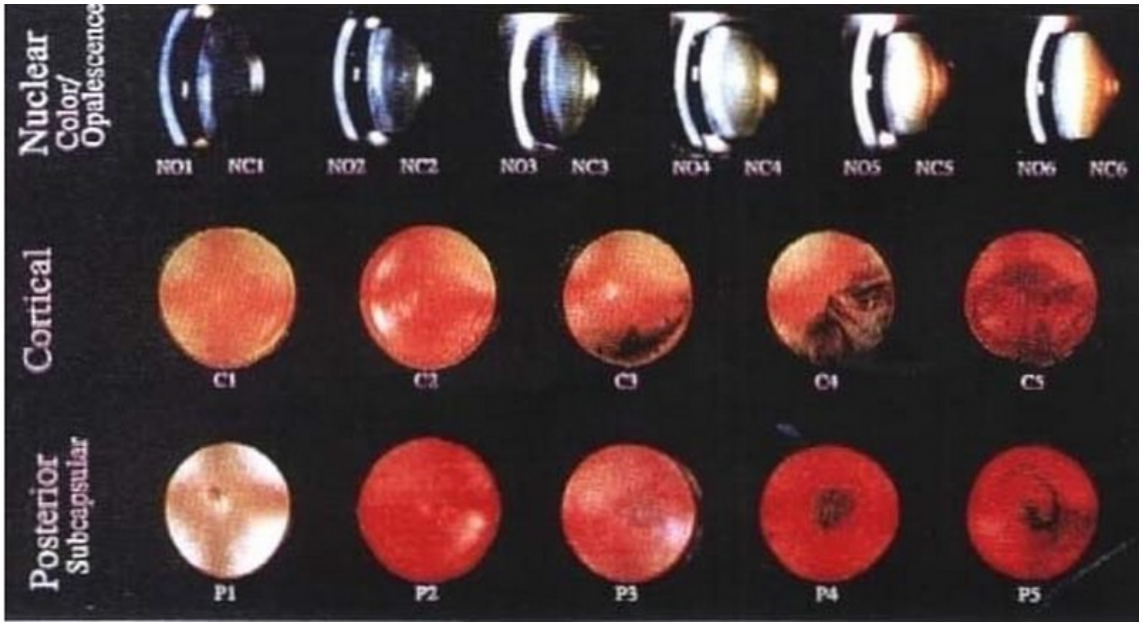

Figure 2.5: Lens Opacities Classification System [3].

Projection and Ophthalmoscopic Transillumination [40]. These methods are good but not efficient since they demand manual assessment, *i.e.*, a doctor that presently examines one patient at a time, which is time consuming and costly. Also, because it requires pupil dilation and a strong light directed to the eye, it can be hard, not suitable and even expensive for some. This makes ophthalmologists a scarce resource that causes large scale screening of cataracts in the early stage very hard [9]. Whereas fundus images could be more easily obtained only with the help of technicians, for example.

2.1.2.2 Treatment

Treatment for cataracts has become a concern due to the impairment the disease causes to the patient.

There is no effective treatment for this eye disease besides removal surgery, but early detection can prevent visual impairment from turning into blindness.

Surgery is conducted when both the individual and doctor agree that the cataract noticeably interferes with the daily life and work [9]. However, by avoiding prolonged exposure to the sun, its progression can be slowed down, if early diagnosed.

"An accurate diagnosis of the severity of the cataract is still needed prior to any surgical interventions to ensure both the safety of the patient and high-quality treatment" [37].

2.2 Artificial Intelligence

We live in an era of automation, where costumers demand speed and efficiency, and now more than ever they can actually get it.

Artificial Intelligence (AI) terms are widely spoken about nowadays, but with all the hype around it, it is easy to get lost, misunderstand and even misuse the concepts.

Understanding the latest advancements in AI can be well summed up in two terms: Machine Learning (ML) and Deep Learning (DL). Examples of it can be spotted everywhere nowadays, even though sometimes we cannot even tell; it is how Netflix knows which show we would want to watch next or how Gmail decides what is and is not spam.

While some applications of AI may still seem far from being possible, there are already a lot of technologies that use intelligent machines, as well as businesses and industries who rely on it, either to function or to improve parameters like speed, efficiency and reliability.

AI is the future, but also part of our everyday lives already. It refers to machines being able to demonstrate human intelligence and act according to it. This includes tasks such as problem solving, object recognition, planning, learning, understanding language and sounds [21].

Machine intelligence has been imagined by many for almost a century. But only in 1956 the term "Artificial Intelligence" was discussed and officialized as a field [5]. However, it was not until 2015 that AI really bloomed, unleashing applications millions of people use on a daily basis. This explosion was mostly enabled by GPUs' wide availability, making processing tasks faster, cheaper and more powerful, and also by storage availability and the Big Data Movement [5].

AI started in the fifties but it was not until the eighties that ML began to appear. DL is the latest concept, that exploded around 2010, driving big advancements in the field (Figure 2.6).

The goal of computer scientists was to build elaborated technology that exhibited characteristics of human intelligence, this being enabled by state-of-the-art computers. This falls in the category of General AI, which refers to the machines that possess every characteristic of human intelligence, and sometimes even more. We have seen machines like this in several science fiction movies, such as The Terminator of the self titled movie or C-3PO from Star Wars [5].

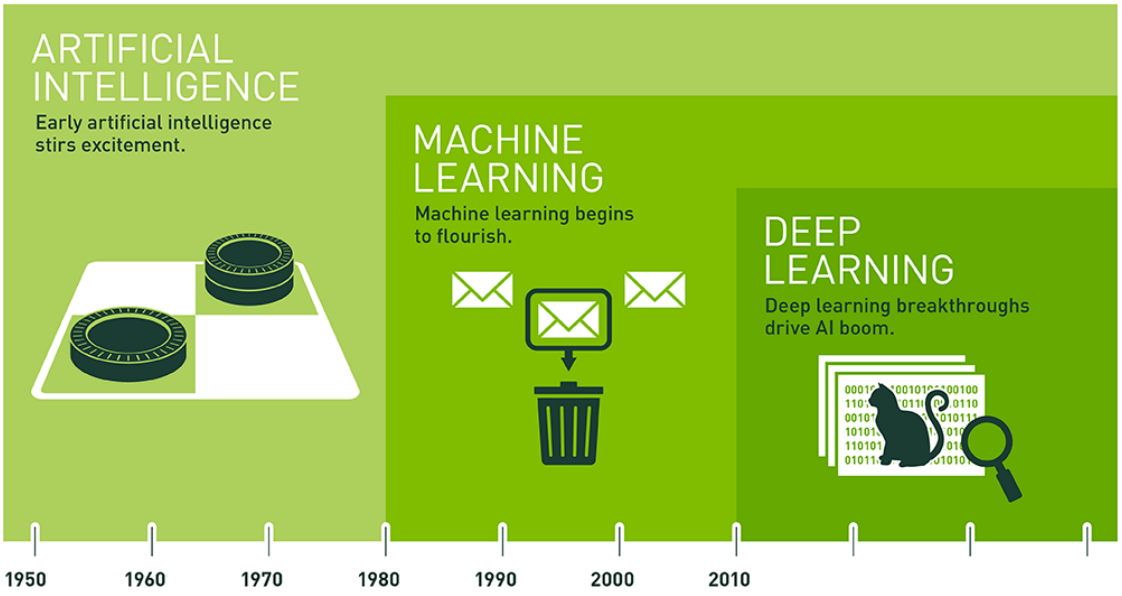

Figure 2.6: The evolution of AI [5].

What is already accomplished nowadays is Narrow AI, machines that execute a specific duty as well as a human and sometimes even better. Some examples are Facebook's face recognition system and Pinterest's image classification service [5]. Technologies like this demonstrate human intelligence traits and can work them extremely well, but are lacking in other areas [21]. How this was possible, leads us to ML.

2.2.1 Classical Machine Learning

Professor Arthur Samuel was an American pioneer of AI research [35] that, in 1959, coined the term "Machine Learning", defining it as "the ability to learn without being explicitly programmed" [21].

A basic definition of ML is "algorithms that parse data, learn from it and then apply what they have learned to make informed decisions" [8], making a determination or prediction about something in the world.

Algorithms are the core of ML. An algorithm is a set of rules to be followed when solving problems. They take in data and perform calculations, which can either be very simple or more complex, to find an answer. Algorithms should deliver the correct answer in the most efficient manner. They need to be trained to learn how to classify and process information. The efficiency and accuracy of the algorithm are dependent on how well it was trained [33].

ML is not just using an algorithm to analyze data and make a prediction, but to use the outcome to improve future predictions.

Several automated tasks across multiple industries and businesses are supported by ML. Many AI applications in customer service use ML algorithms, making workflows more reliable, increasing productivity and assisting self-service. A perpetual incoming of customer opinions and questions feeds these algorithms, leading to fast accurate predictions [8].

ML is simply a way of achieving AI. But it is possible to achieve AI without it. However, this would require hand-writing millions of lines of code with complex rules, which would be hard and take too long. Instead, ML is used as a way of training an algorithm, using large amounts of data and allowing it to adjust itself and improve [21], learning how to perform a specific task.

Algorithmic approaches for ML include inductive logic programming, Decision Tree (DT) learning, reinforcement learning, Bayesian Network (BN), clustering, among others [5].

One of the most relevant applications of ML over the years has been improving computer vision (the ability of a machine to recognize an object in an image or video), but it still required a lot of hand-written code. People wrote edge detection filters to identify an object's boundaries, shape detectors and even classifiers to recognize details. Although they were progressing, the method would fail if there was low visibility, for example. Computer vision was still too prone to error [5].

In some cases, thousands of pictures were gathered and people tagged them. Then, the algorithm would build a model that could accurately distinguish between the input groups, ideally as well as a human. The training and testing of an algorithm can be repeated several times, until the accuracy level is high enough, only then we can say that the machine successfully learned what we were trying to "teach". A machine being capable to learn means it executes a task with the data fed into it and is progressively getting better at it.

Learning Types

There are essentially four ML types [7]:

1. **Supervised Learning** uses labelled examples, so the output for the given input is known before itself. The machine must be able to assign the given input to the output.

2. **Unsupervised Learning** uses unlabelled inputs, therefore the output is unknown. The data is provided, the model finds insights about it and groups it together accordingly.

3. **Semi-supervised Learning** is an halfway between Supervised and Unsupervised Learning. The inputs are a combination of labelled and unlabeled data.

4. **Reinforced Learning** consists of exposing the machine to an environment where it gets trained by trial and error. "It learns from past experience and tries to capture the best possible knowledge to make accurate decisions based on the feedback received" [7].

It is also possible to combine multiple learning models to achieve better results. This is know as **Ensemble Learning**. This method has great potential to achieve higher accuracy and less error than the individual models. "If each base learning model is viewed as an expert, multiple experts may be better than any single one if their individual judgments are appropriately combined, *i.e.*, ensuring that individuals in a group make different errors in different instances" [38].

2.2.2 Deep Learning

DL goes yet another level deeper and can be considered a subfield of ML. Although it has different capabilities, technically it is ML and its behaviour is similar [8].

The data analysis in a DL model follows a logic structure similar to how a person reaches a conclusion. This makes for machine intelligence that is far more capable than that of standard ML models, not needing guidance or adjustments like the latter [8].

DL networks need large quantities of items in order to be trained, because the system learns from exposure to millions of data.

An Artificial Neural Network (ANN) is a DL model for implementing ML. A Neural Network (NN) is inspired by our understanding of the biology of the human brain, such as the neurons' connections. They include discrete layers, connections and directions of data propagation, different from biological brains in which a neuron can make connections with any other if they are close enough. The input passes through each layer of neurons until the final output is produced [5]. ANNs are then algorithms that mimic the structure and function of the brain, where the multiple layering achieves the desired depth for DL [21]. The layers can be seen as a nested

hierarchy of related concepts or DTs, where the answer to one question leads to a set of deeper related questions [33].

Just as the brain can recognize patterns and help us categorize and classify information, ANNs do the same for computers. The brain is constantly trying to make sense of the information it is processing, and to do this, it labels and assigns items to categories. When we encounter something new, we try to compare it to a known item to help us understand and make sense of it, just as ANNs [33]. They have a lot of advantages, such as extract meaning from complicated data, detect trends and identify patterns too complex for humans to notice, learn by example and, of course, speed improvement in comparison to human work.

Observing an image, distinctive features of it are examined by the neurons. The ANN presents a probability vector saying it might be "x" with 86% probability, "y" with 7% sure, and so on [5]. Feature diversity is the key to high quality classification results, so it is important to use independent feature sets to achieve it.

The simplest NN were computationally intensive, until recently. Only in the last years they were considered a practical approach through the use of GPUs [5].

As a network is getting tested, it is normal to come up with wrong answers because it needs to be trained, it needs to see a great number of images until the weightings of the neurons are so accurate that the software is correct most times, only then we can say the NN has taught itself what something looks like [5].

As said before, DL is becoming popular in part because of the Big Data Movement that is presenting lots of opportunities in the field of AI. In the next decade we will progress in ways we cannot imagine yet [8].

Whether using ML or DL, one thing is certain, if the data being used is flawed, then the insights and information extracted will be flawed as well. Data is at the heart of the matter, it is important to do data cleansing, which consists of detecting, correcting or even removing corrupt or inaccurate records from a record set, table or database and identifying incomplete, incorrect or irrelevant parts of the data [33]. In order to advance, the data driving the algorithms and decisions needs to be high-quality, otherwise the insights from the data cannot be trusted [5].

Today we already reached a point where some image recognition DL softwares are better than humans, especially in scenarios of detection of cancer indicators or tumors in scan images [5], for instance.

High quality preventive health care, fully automated houses and cars that drive

themselves are not just utopias anymore. As DL becomes more refined, enabling even more real-world applications of ML, what we imagine as science fiction today may be a reality soon.

Main Differences

AI is a broader concept than ML, addressing the use of computers to mimic the cognitive functions of humans. ML focuses on the ability of machines to receive a set of data and learn for themselves, changing algorithms as they learn more about the information they are processing [33].

In ML, the feature (key parameter the system uses to do the classification and produce an output) selection is done by the programmers, humans are the ones who study the problem and decide what features better describe what they want the computer to learn. In DL, the system itself chooses what it thinks are the best features from what they are trying to learn, doing that by being exposed to a huge amount of data, as mentioned before.

Understanding the difference between both terms is to realize DL is ML, but the next evolution of it, the way machines make exact decisions without someone to instruct them, powering the most human-like AI so far [8].

With the advancements in technology, specifically better computers, GPU processing speed allied to the Big Data Movement, ideas that we only dreamt to be possible are coming off the paper to real world applications. The advancements in AI are an exciting prospect for many businesses and industries, both nowadays and in the future.

2.3 Automatic Cataract Detection Systems

Yang *et al.* [39] propose the classification of retinal images by using a NN. The research included three main parts, pre-processing, feature extraction and classifier construction. An Improved Top-Bottom Hat Transformation was done in the first part to enhance the contrast between the background and the blood vessels as well as a Trilateral Filter to reduce the noise in the retinography. The transformation done is a mathematical morphological method that enhances the quality of the image. It is improved by doing an histogram equalization operation after, making the image clearer [39]. As classification features, they extracted luminance, which is the intensity of white pixels in the image, and texture, which represent the characteristics and appearance of each pixel. The classifier is constructed by a two-layer Back Propagation Neural Network, which then classifies the photos, based on their clearness degree, in a four-class grading scale (normal, mild, medium or severe). Back Propagation Neural Network is a widely used multi-layer hierarchical NN with upper neurons associated with lower neurons. It can accurately predict results with inputs that have never been seen by the NN [38]. The true positive rate was an average of 82.5%.

In the work of Zheng *et al.* [41], pre-processing included resize and extraction of the green channel from the 460 Red, Green and Blue (RGB) retinographies. After pre-processing, they calculate the two dimensional Discrete Fourier Transform of the fundus images and use the calculated spectrum as features, followed by Principal Component Analysis (PCA) to reduce dimensions. PCA is a common unsupervised learning method for dimension reduction by seeking a projection that best represents the data in a least squares sense [41], resulting in a reduction of computation cost. The authors considered opacifications in the eye lens as a low-pass filter, which absorbs and scatters the light, filtering the details of the fundus. They report that as the cataract severity progresses, the high frequency components are fewer and the low frequency ones increase. The classification uses the Linear Discriminant Analysis classifier promoted by the AdaBoost algorithm. Linear Discriminant Analysis is similar to PCA but supervised [41]. AdaBoost trains individual classifiers using a part of training samples by re-sampling and voting for the final decision; a weight is given to each training sample, which determines the probability of it being selected for a component classifier; when a training sample is misclassified, its weight will increase as well as the chance of being chosen for an individual classifier [41]. The classification accuracy was 95.22%.

To extract features suitable for the task, Guo *et al.* [9] explored both the Discrete Wavelet Transform (DWT) with Haar transform and the sketch-based with Discrete Cosine Transform. After this step, a Multi-class Fisher Discriminant Analysis algorithm was used for a binary classification (cataract, not cataract). "MDA finds a linear transformation that best discriminate among classes and the classification is performed in the transformed space based on some metric such as Euclidean distance" [16]. For the training and testing of the algorithm, they were able to use a real-world dataset with 445 fundus images. The correct classification rate for the wavelet transform was 90.9%. For the sketch-based method, the result obtained was a bit worse, 86.1%, but still good overall.

Fan *et al.* [6] used PCA to reduce the dimensionality of two sets of features (wavelet and sketch) extracted from fundus images. To build the classifier, they adopted four ML algorithms: Support Vector Machine (SVM), Gradient Boosting Decision Tree (GBDT), Bagging and Random Forest. SVM is a popular learning model used for classification and regression analysis, based on the structural risk minimization principle from statistical learning theory; it can perform linear and non-linear classification using different types of kernels [6]; it builds a hyperplane that separates the positive and negative examples while maximizing the smallest margin from two classes of data; SVMs are robust to overfitting and can scale up to high dimensionalities; besides, there is no need for parameter tuning since the derived ones theoretically provide the best results [38]. A DT is a supervised learning method that builds models in a tree structure; the bigger the tree, the more complex the decision rules and the fitter the model; it breaks down the dataset into smaller subsets, building nodes and branches; a DT classifier is similar to a flowchart diagram where the final nodes represent classification outputs and the first node the best predictor [30]. GBDT is an Ensemble Learning method, it optimizes the DTs by iteratively choosing a weak hypothesis that points in the negative gradient direction [6]. Bagging is an approach that generates multiple versions of a predictor and leverages them to get an aggregated predictor; the classification results are obtained in accordance with the majority voting of multiple base classifiers [6]. Random forest is also an Ensemble Learning method that uses bagging, it constructs several DTs in training and outputs the class that is the mode of the classes or the mean prediction of the individual trees [6]. The accuracy of each one of them is shown in Table 2.1. As we can confirm, the best result was an accuracy rate of 84.77%, obtained from the SVM algorithm with sketch features and before the PCA transformation. The authors concluded that the classification accuracy before and after the PCA transformation is nearly the same. However, they also report that the computation time

PCA	Features	SVM (%)	Bagging (%)	Random Forest (%)	GBDT (%)
Before	Wavelet	79.85	79.55	80.30	81.56
	Sketch	84.77	82.41	84.26	81.65
After	Wavelet	79.25	77.09	79.84	79.35
	Sketch	82.65	81.75	82.80	80.75

Table 2.1: Classification accuracy of the wavelet and sketch features, before and after PCA [6].

diminished from more than one second to less than one second by doing this type of analysis, which is a positive outcome since we ambition for faster approaches.

Previous studies use a single learning model for fundus image cataract classification and grading, Yang *et al.* [38] innovated presenting an ensemble learning based approach. From retinal images, they extracted wavelet, sketch and texture features, and for each feature set they built a SVM and a Back Propagation Neural Network. Then, the ensemble methods Majority Voting and Stacking combine the multiple base-learning models for the final classification. In Majority Voting, the class label of an unlabelled instance will be the one that obtains the highest number of votes, *i.e.*, the most frequent vote given by the multiple base classifiers; it does not require parameter tuning once the base classifiers have been constructed [38]. The Stacking approach employs a meta classifier to generate the final classification; it typically uses a two layer frame structure, where base classifiers are generated from the training dataset in the first layer and combined by the meta classifier in the second layer [38]. The accuracy rate of the ensemble classifier was 93.2%, outperforming the single learning models.

Building a good classifier generally requires a large amount of labelled examples, which can be expensive to obtain. Plus, a predefined set of image features may provide an incomplete, redundant or even noisy representation, according to Song *et al.* [31]. For that reason the authors [31] decided to use semi-supervised learning to build a classifier for automatic classification of cataracts. Their dataset included 476 labelled examples and 4902 unlabelled ones, in a total of 5378 fundus images. The features extracted from the retinographies were wavelet coefficients and texture. The algorithms used to build the classifier were a Bayesian Network (BN) and a DT, both supervised methods, but they used tri-training to learn a good hypothesis. A BN is a mathematical model based on Bayes formula for probabilistic relationships among sets of variables, capable of extracting additional information [31]. The tri-training algorithm generates three classifiers, using Ensemble Learning to improve the generalization and reduce error; an unlabelled example is labelled by one clas-

Features	BN (%)	DT (%)
Wavelet	88	86
Texture	69	70

Table 2.2: Classification accuracy of the wavelet and texture features, using BN and DT [31].

sifier if the other two are in accordance with the labelling under given conditions, and the final hypothesis is produced via Majority Voting [31]. The performances are shown in the Table 2.2. It is obvious that the wavelet feature outperforms the texture one, specially using a BN, with which the result was better. The authors also remark that the performance of the semi-supervised learning is not always better than supervised method.

Kolhe *et al.* [16] created a remote cataract detection system deployed on the cloud, so clients could access the system remotely. It used supervised and unsupervised learning algorithms to do the classification. The database contained 261 fundus images graded by ophthalmologists. The images were pre-processed by extracting the green channel and using Contrast-limited Adaptive Histogram Equalization to uniformly adjust the contrast. For feature extraction two methods were explored, DWT (Haar Transform) and Skeletonization with Discrete Cosine Transform. "Skeletonization is performed before the Discrete Cosine Transform to reduce foreground regions in a binary image to a skeletal remainder that largely preserves the extent and connectivity of the original region" [16]. The above methods output coefficients of DWT and Discrete Cosine Transform, whose high frequency components were considered as features, details hard to recognize in the space domain but easily discovered in frequency domain. After, PCA was applied to reduce dimensions and select appropriate features. A SVM was implemented to classify the images in two classes (non-cataract and cataract). Results showed a precision of 77.7%, a sensitivity of 93.0% and a specificity of 77.7%.

The research of Harini *et al.* [11] used a SVM to classify 60 images. For preprocessing a mean filter was used for smoothing the images and removing Gaussian noise, replacing each pixel with the average value of the intensities in the neighborhood. Feature extraction was done using the Haar Transform. The number of coefficients obtained through the used wavelet was compared with other wavelets such as Daubechies and Biorthogonal, and the Haar gave more high frequency information than the others. Canny edge detection was also done. The widths of the edges were dilated for clear visibility and then the edge components were counted.

Features	SVM (%)	Softmax (%)
Wavelet	86.00	81.91
DL	89.83	94.07

Table 2.3: Accuracy of the wavelet and DL features, using SVM and Softmax [27].

For a cataract, the number of components is low and for a normal eye, the edges are clear without discontinuity and hence the number of components is higher. The accuracy obtained was 91.11%, the sensitivity 90.00% and the specificity 93.33%.

Zhang's *et al.* method [40] uses a Convolutional Neural Network (CNN) for the task. CNN is another kind of ANN that has a high degree of invariance to translation, scaling, tilting and other forms of deformation [40]. They pre-processed the retinal images using a G-filter to reduce the interference of local uneven illumination and the eye's reflection. The accuracy rate obtained was 93.52% in cataract detection.

"Some might find it difficult to use manually select features and ML algorithm combinations to obtain better results. Due to its good automatic feature selection ability and classification accuracy, DL is becoming more effective in different fields" [27]. The research of Qiao *et al.* [27] focused on the comparison between the wavelet feature extraction and the DL aided one. They pre-processed the fundus images using the Maximum Entropy Method to calculate the optimal classification gray level threshold and then perform local gray level transformation. The SVM and Softmax algorithms execute the detection, resulting from the analysis of the features extracted by wavelet and by DL. Softmax is a DL function that outputs a vector that represents the probability distributions of a list of potential outcomes. The results obtained are discriminated in Table 2.3. Results showed that the higher accuracy (94.07%) was from the features extracted by DL and classified by Softmax, for about a 4% difference from the SVM classifier. Features extracted by the wavelet transformation lead to an average accuracy of 84%.

A DT was trained by Xiong *et al.* [37] to classify a total of 1355 fundus images. Pre-processing included image size normalization, green channel extraction and enhancement with a two dimensional Gaussian Matched Filter, in order to improve the visibility of retinal vessels and the OD boundary. Three types of features were extracted, namely the number of pixels of visible structures, the mean contrast between retinal structures and background and, lastly, the local standard deviation. The final accuracy of the two-class classification was 92.8%.

3

Methodology

After analyzing the research done in the field, one paper was chosen to guide this study. Starting this work in the fields of ML and DL, the work of Guo *et al.* [9] came across as a good way to kick-start this project. It was carefully studied and examined so it could be possible to improve and add to it, based on what was also learned from all the other papers.

The methodology followed is then similar to the one done by Guo *et al.* [9], but contains a lot of new implementations as well.

3.1 Dataset

Three different groups of datasets provided by Retmarker S.A. were used: **Public**, **Cataract** and **NonCataract**.

The first one contained 20 smaller public datasets, namely ARIA, CHASEDB1, Color Fundus Images of Healthy Persons & Patients with Diabetic Retinopathy (CFIHPPDR), Data 50 Healthy Persons (D50HP), Database for the purpose of Vessel-based Registration of Fundus Projection Images (DVRFPI), diaretdb0_v_1_1, diaretdb1_v_1_1, DMED, DRHAGIS, DRIMDB, DRIVE, e_optha_EX, e_optha_MA, FIRE, HRF, IDRID, M_ES.database, Messidor, Messidor-2 and ROC. The two other datasets (Cataract and NonCataract) were proprietary and contained, as the name reads, cataractous and not cataractous images, respectively. The properties of the datasets are compiled in Table 3.1, which identifies the dataset, its number of images, the resolution of these images and the presence of cataract as well as of other pathologies, identifying each one.

All the images were manually analyzed and graded one by one by the author. This was done in order to build an appropriate dataset to develop this work, that combined a wide and rich variety of fundus images, from different cameras and datasets.

Dataset	# Images	Resolution	Cataract	Other Pathologies
ARIA	212	768 × 576	False	AMD, DR
Cataract	11 761	2448 × 2448	True	-
CHASEDB1	28	999 × 960	False	DR
CFIHPPDR	60	720 × 576	False	DR
D50HP	99	1612 × 1536	False	-
DVRFPI	22	1200 × 1143	True	Several unnamed
diaretdb0_v_1_1	130	1500 × 1152	False	DR
diaretdb1_v_1_1	89	1500 × 1152	False	DR
DMED	169	2196 × 1958	False	-
DRHAGIS	40	4752 × 3168	True	-
DRIMDB	194	760 × 570	False	-
DRIVE	40	565 × 584	False	-
e_optha_EX	82	2544 × 1696	False	DR
e_optha_MA	381	2544 × 1696	False	DR
FIRE	268	2912 × 2912	True	DR, Myopia, Hyperopia
HRF	45	3504 × 2336	False	DR, Glaucoma
IDRID	597	4288 × 2848	True	DR
M_ES.database	35	720 × 576	False	DR
Messidor	1200	2304 × 1536	True	DR
Messidor-2	1756	2304 × 1536	True	-
NonCataract	11 997	2448 × 2448	True	-
ROC	100	1394 × 1392	False	DR
STARE	397	700 × 605	True	-

Table 3.1: Properties of the datasets used.

The images in the Cataract dataset were divided into 4 categories called “Errors”, “Not Cataract”, “No image” and “Cataract”. The first contained images with no quality, non-analyzable, a total of 937. Some examples are shown in Figures 3.1, 3.2 and 3.3. The second category had 1 694 images, the ones the author considered to demonstrate no presence of cataract, that were incorrectly labelled. The “No image” category is a compilation of 155 completely black images, an example can be seen in Figure 3.4. The remaining cataractous images were put into the “Cataract” category.

This assessment was done for every dataset except for the NonCataract, which was not completely examined because of lack of time and regarding the high amount of images. From the latter were only separated the 88 non-analyzable no quality images and the 70 completely black ones that provided zero information, as mentioned above for the Cataract dataset. The rest of the retinographies were put in the “Not_Cataract” category.

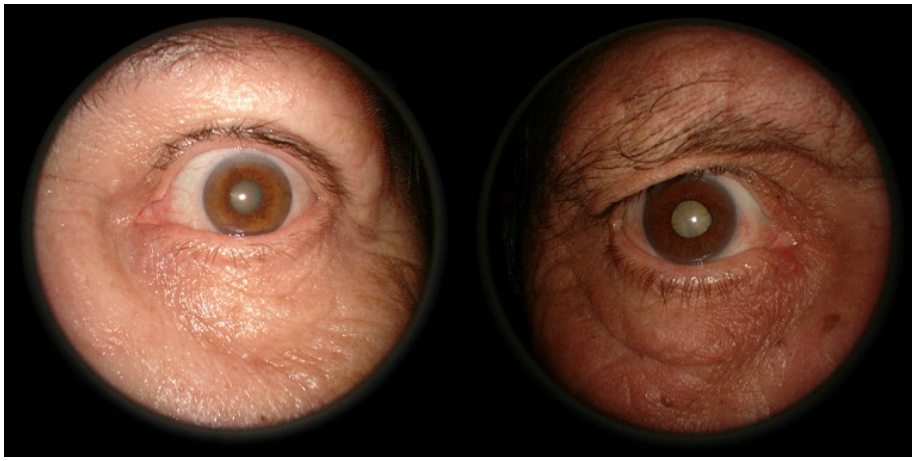

Figure 3.1: Two eye images that are not fundus photographs, from the Cataract dataset.

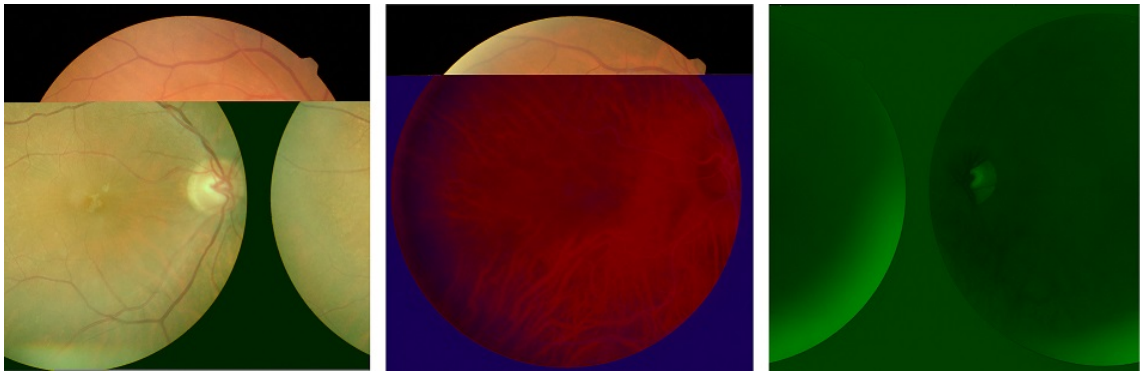

Figure 3.2: Three fundus images with no quality.

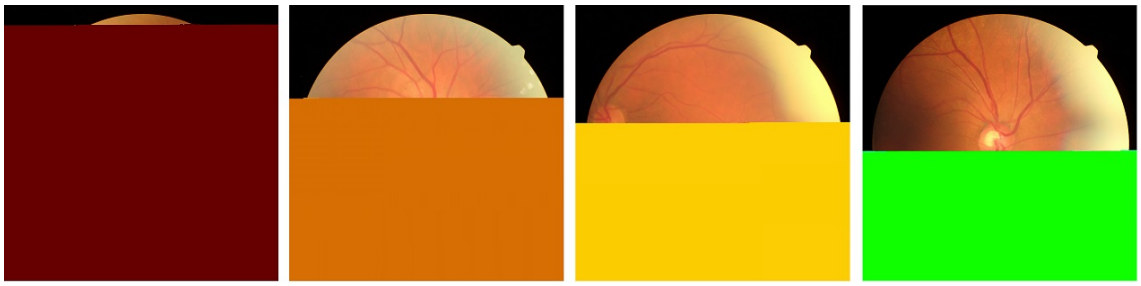

Figure 3.3: Incomplete fundus images that were considered to have no quality.

Figure 3.4: A black image present in some datasets, which in the upper left corner says "Sem imagem", meaning "No image".

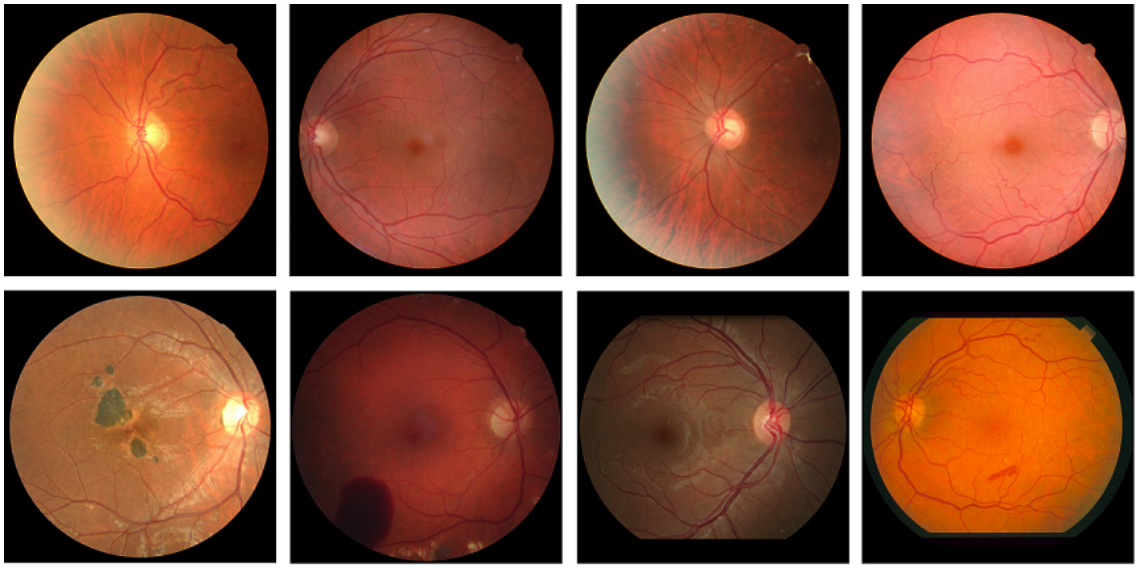

Figure 3.5: Examples of fundus images from the Not_Cataract dataset.

Examples of healthy and ill images can be seen in Figures 3.5 and 3.6, respectively.

Only few datasets presented relevant ground truth (healthy or cataractous), namely D50HP, CFIHPPDR, Cataract and NonCataract. Nevertheless, the last two were not totally reliable, for some images were mislabelled, but the majority was correctly identified.

The final dataset is then made up of two categories, containing a total of 25 217 fundus images with and without cataract. The Cataract and Not_Cataract categories have 9 089 and 16 128 photographs, respectively. Not all images from the initial datasets were used, since some were considered too small (less than 800×800 pixels) or not useful, such as being completely black, for example.

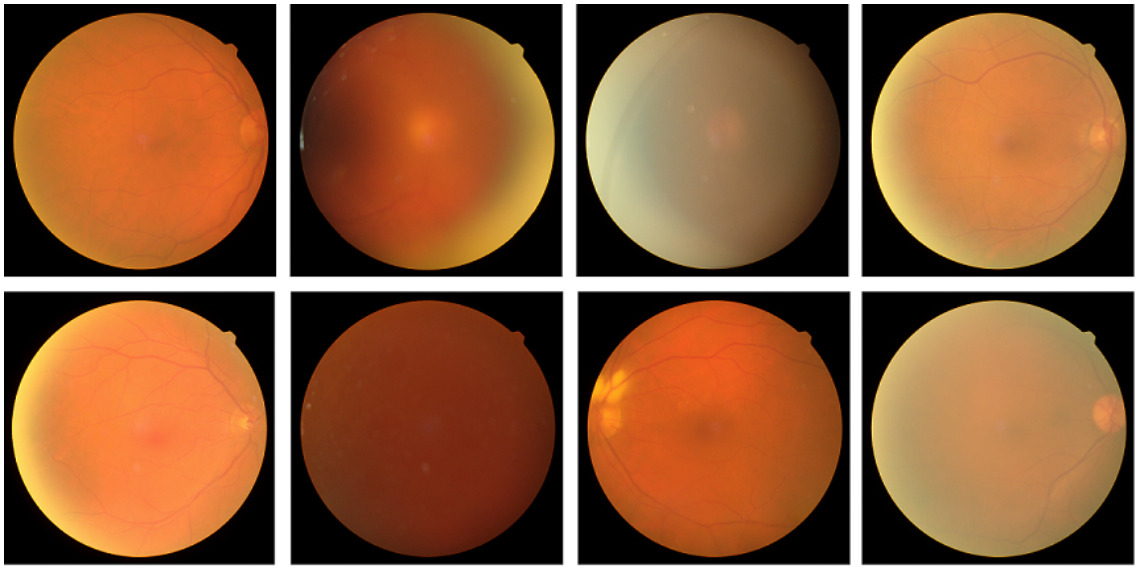

Figure 3.6: Examples of fundus images from the Cataract dataset.

3.2 Pre-Processing

The pre-processing step makes the images suitable for further steps, improving their visual appearance and making the manipulation of datasets easier.

Sometimes a few approaches are required to enhance the image's condition, such as its improvement and noise removal.

The pre-processing steps can be seen in Figure 3.7 and included padding of the images to make sure they were all square. Then cropping was done to minimize the black mask that has no useful information. This transformations were finalized with resizing the images to the most common resolution of the dataset, which was 2448×2448 pixels. The mentioned changes were done to improve dataset homogeneity.

Another important step present in all the mentioned research is the extraction of the Green Channel (GC) from the RGB image. This is done because, according to previous studies, the green component shows the most details of the original color fundus image.

For a matter of comparison, it was decided to also use luminance images to see which ones would perform better. Luminance (L) can be calculated through the following expression [39], which combines the three different color channels of an RGB image.

$$L = 0.3 \times R + 0.59 \times G + 0.11 \times B$$

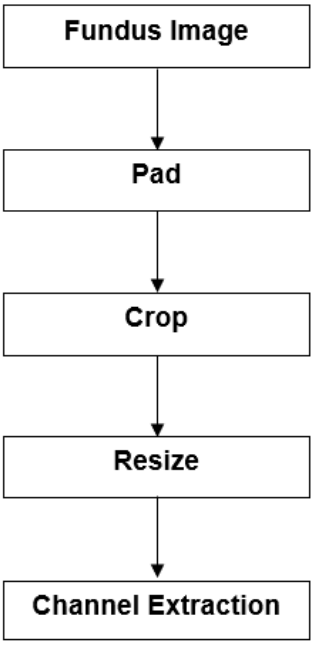

Figure 3.7: Overview of the pre-processing procedure.

With R being the red channel, G the green and B the blue one.

Therefore, after the mentioned pre-processing steps, the dataset was doubled. One category had the dataset with all images in the GC and another had the dataset with all luminance images. This datasets were then ready to go through feature extraction.

3.3 Feature Extraction

As Guo *et al.* [9] and several others ([6], [11], [16], [27], [31] and [38]) mention the use of the DWT, more specifically, the Haar transform, this method was considered suitable for the task proposed and was adopted to perform the feature extraction. The steps included in this process can be seen in Figure 3.8.

Non cataract fundus images show clear optic structure details, contrasting with cataractous images where details are less visible. An intuitive approach to select features is then to use the localized features related to the high frequency components. These details are usually hard to be recognized and quantitatively assessed in space domain but obvious in frequency domain [9].

DWT allows both time and frequency analysis, showing good contrast between blood vessels or edges (high frequency components) and background (low frequency compo-

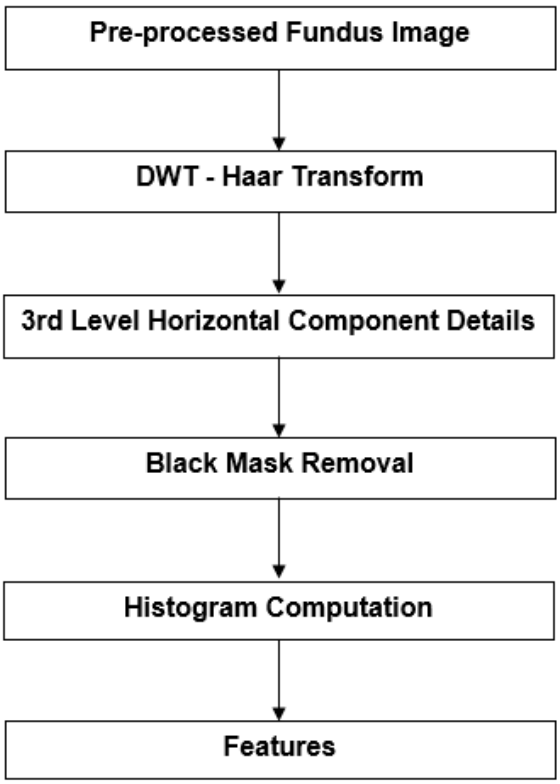

Figure 3.8: Overview of the feature extraction process.

nent) in fundus images [11]. It allows the decomposition of the image into a wanted number of levels to distinguish the high from the low frequency components. Its coefficients quantify the horizontal, vertical and diagonally oriented details at each level of the Haar wavelet transform. These coefficients are then counted according to their amplitude and it is noticeable that severe cataracts have low values of coefficients [11].

"Many functions can be used as the mother functions for wavelet transform. Since the calculation with Haar transform can be easily carried out with only additions so that no multiplications are needed because most elements of the Haar wavelet transform matrix are zero, it can achieve high computing efficiency" [9].

Therefore, the Haar decomposition in 3 levels was computed by using the "haart2()" Matlab function that performs the two dimension Haar transform. According to the paper of Guo *et al.* [9], we are only interested in the third level horizontal details of the wavelet transform.

Since only the round center of the fundus photograph is supposed to be considered, a function to remove the border (*i.e.*, the black mask) of the image was written. It takes as input the fundus image and the level of details wanted, and the output is

the image without the mask, an uneven-sided matrix. What the function does is finding center of the image, the medium point of the image's height and, from there, the point where the black mask stops. With it, it calculates the radius of the circle, that is our region of interest, and creates a binary mask that obeys the equation of the circle, enabling the elimination of the part that has no useful information. This method is very simple, but it is sufficient for the required purpose.

Having the wanted region of interest, the coefficients are read and plotted into a graph that shows the intensity of the coefficients in each pixel of the image. This plot is very useful to visually notice the difference between an healthy and a cataractous image. In Figure 3.9 we can see two examples of these graphs, the first for a cataractous image and the second for an healthy one, from Guo *et al.* 's [9] paper.

It is clear from Figure 3.9 that distribution of the third level horizontal coefficients of the Haar decomposition of a healthy image is more irregular and erratic, comparing to the distribution of a cataractous fundus images that is even and more stable. This shows the practicality of the approach, since the difference between sick and healthy images is prominent.

An histogram of the coefficients is computed with ten pre-defined edges, or regions, from Guo *et al.* 's [9] paper. It is the frequency count of each bin of the histogram that corresponds to each one of the ten features.

Guo *et al.* 's [9] work implements a four-class classification, rather than just a two-class distinction. However, a four-class classification was not done in this work, because the data was not separated into the four classes needed, corresponding to four stages of severity of the disease. Without a solid ground truth it would be

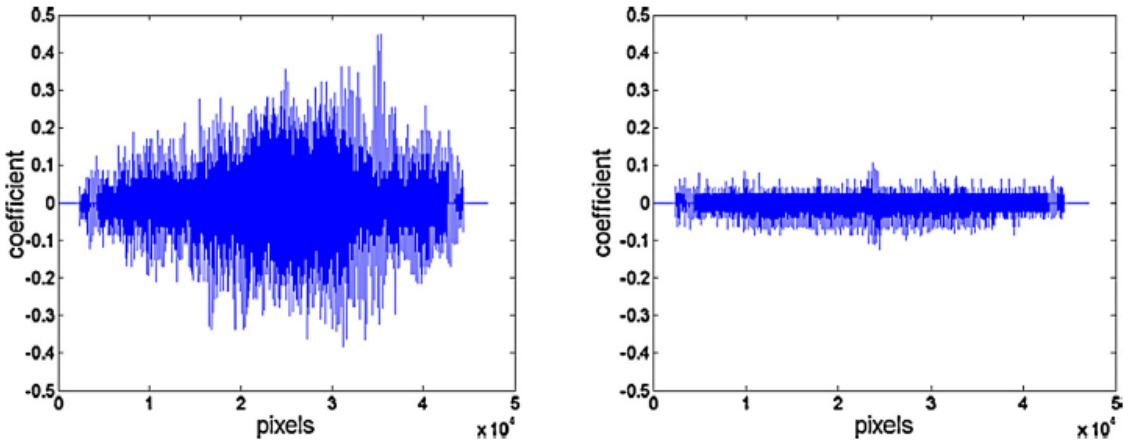

Figure 3.9: Third level horizontal coefficient amplitudes for each pixel for an healthy image and for a cataractous one, respectively [9].

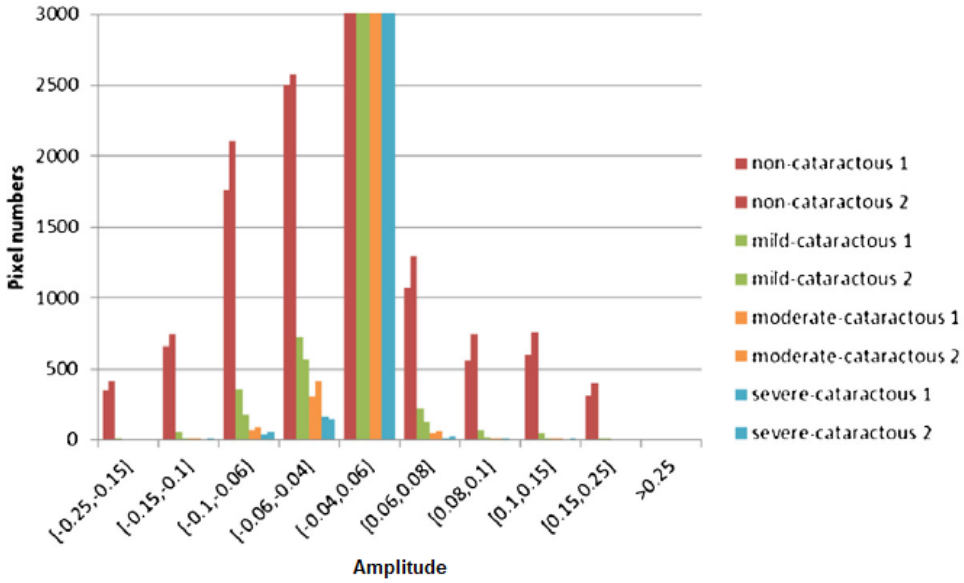

Figure 3.10: Histogram of the wavelet third level horizontal coefficients [9].

impracticable to carry out an accurate classification and grading of cataract.

The histogram plot presented in Guo *et al.* 's [9] paper is then for a four-class classification, it can be seen in Figure 3.10. It shows the frequency count of the coefficients according to their amplitude in each one of the ten intended regions.

Figure 3.10 shows a much wider and taller distribution of the histogram for non cataractous images, contrary to sick images that have less to none coefficients in the outer regions and a small amount in the center regions, excluding the center region in which both cases present the highest amount of coefficients and are hard to compare.

Essentially, the number of coefficients in different amplitude regions has a significant difference, which implies that the number of coefficients in those regions can be used as features for cataract classification and even grading [9].

In summary, each fundus image is converted into a set of features that is later classified by an ML algorithm.

3.4 Learning Models and Classifiers

3.4.1 Support Vector Machines

SVM is a popular classification and regression method. Using it, it is possible to train a method on a representative set of input–output pairs and obtain good results without having to train on all possible input–output pairs.

Supposing we are given a training dataset $D = \{f(x_i, y_i) \mid i = 1, 2, ..., n\}$ of input vectors x_i and associated targets y_i, the goal of regression is to fit a function $f(x)$ which approximates the relation inherited between the dataset points so that it can be used to predict new cases. SVM regression can be described as the following optimization problem [18]

$$min \quad \frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i + C\sum_{i=1}^{N}\xi_i^*$$

subject to [18]

$$y_i - < w,x_i > - b \leq \varepsilon + \xi_i$$

$$< w,x_i > + b - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

where <> denotes the dot product, w the vector of coefficients, $C > 0$ a regularization constant, b an offset value and ξ_i, ξ_i^* the slack-variables for pattern x_i. The prediction of a new input vector can be obtained by [18]

$$f(x) = < w,x > + b$$

A SVM with a Gaussian kernel has two-layers. The first is a set of template matchers that measure the similarity of the input pattern to each of the training samples. "The second layer computes the discriminant function as a linear combination of the similarity scores with learned weights, where the kernel function measures the similarity between the input pattern and the training sample" [15]. The samples for which the corresponding weights are non-zero are the support vectors.

3.4.2 Decision Trees

DT analysis is a non-parametric supervised predictive modelling tool, widely used for classification and regression tasks. They are constructed via an algorithmic approach that identifies ways to split a dataset based on different conditions. "The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features" [10]. The decision rules are generally in the form of "if-then-else" statements, and the deeper the tree, the more complex the rules and the fitter the model [10].

A DT is a tree shaped graph, showed in Figure 3.11. A node is where an attribute is picked and a question is asked; branches represent decisions, the answers to the question; and the leaves or leaf nodes represent the outcome/output or class labels [10] [22].

DTs classify the examples by sorting them down the tree from the root to a leaf node, providing the classification to the example. Each node in the tree acts as a test case for some attribute and each edge descending from that node corresponds to one of the possible answers to the test case. This process is recursive, being repeated for every subtree rooted at the new nodes [10].

To generate a DT from data, the algorithm used was Matlab's Classification and Regression Tree. It uses Gini index as cost function to evaluate the split in feature selection in the case of the classification tree, and least square as the metric in the case of the regression tree [22].

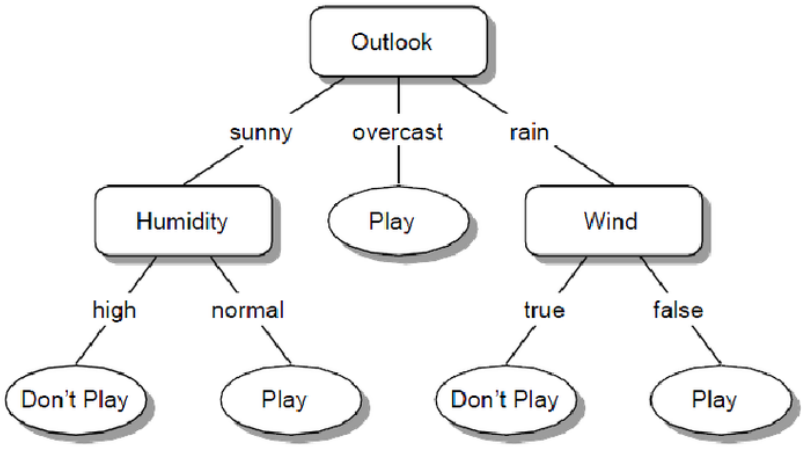

Figure 3.11: A Decision Tree concerning the decision to play golf depending on the weather [12].

Gini index is a measure of inequality or impurity in the sample, ranging from 0 to 1. A value of 0 means the sample is perfectly homogeneous and all the elements are similar, whereas a value of 1 means maximum inequality among elements. It can be calculated as [22]

$$Gini\ index = 1 - \sum_{i=1}^{n} p_i^2$$

with p_i being the probability of each class and i the number of classes [22].

3.4.3 Bagged Trees

BT is an ensemble method that combines several DTs to produce better predictive performance. Since individual DT tend to overfit, BT combine the results of many DT, reducing the effects of overfitting and improving generalization. They also select a random subset of predictors to use at each decision split as in the Random Forest algorithm [20].

Bagging reduces the variance of a DT. It creates several subsets of data from training samples chosen randomly, which are used to train the DTs. As it uses an ensemble of different models, the average of the predictions from all the different trees are used, which is more robust than a single DT [23].

3.4.4 Convolutional Neural Networks

CNNs consist of multi-layer architectures where the successive layers are designed to learn progressively higher-level features, until the last layer which produces the label. This process can be seen in Figure 3.12 which shows an example of a CNN that distinguishes types of vehicles.

Feature extraction is an integral part of the classification system, rather than a separate procedure. After training, the last layer can be seen as a linear classifier operating on optimized features extracted by the previous layers. Feature extraction contains a stack of convolution (C) and subsampling (S) layers. The C-layers compute convolutions over the previous layers x_{in} with some small trainable convolution kernels k [15]

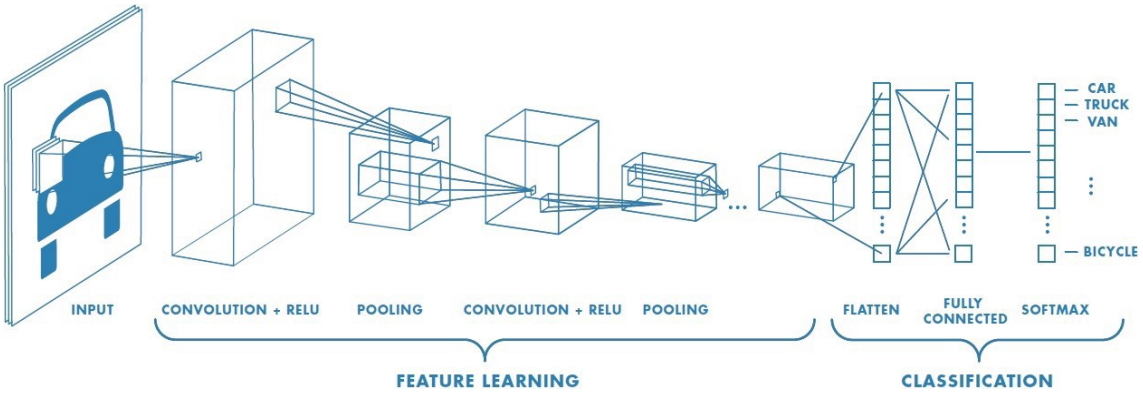

Figure 3.12: A Convolutional Neural Network to distinguish types of vehicles [28].

$$x_{out} = S(\sum_i x_{in} \otimes k_i + b)$$

where S is a non-linear function (hyperbolic tangent sigmoid) and b is a scalar bias. Depending on the values of the kernel coefficients, the convolution operation can implement a local edge detector, a low-pass filter or others. On each C-layer, multiple convolution kernels can be used, creating several different feature maps.

The spatial S-layers take the average of a $n \times n$ pixel block, multiply it by a trainable scalar β, add a bias and pass the result through a sigmoid [15]

$$x_{out} = S(\beta \sum_i x_{in}^{n \times n} + b)$$

The result is a feature map of lower resolution where some position information about features has been eliminated, creating some level of distortion invariance in the representation.

Alternated layers of convolution and subsampling can extract features from increasingly large receptive fields, with increasing robustness to irrelevant variabilities of the inputs. The overall effect of these layers is to extract a feature vector v from the input x, written as $v = c(x)$ [15].

The last layer of a CNN computes the product of the feature vector v with a weight matrix W, adds a bias vector and passes the result through sigmoid functions. For training, the Euclidean distance between the output vector and a target output vector T^i is used as the loss function to be minimized [15]

$$L = \parallel S(W.s + b) - T^i \parallel^2$$

where W is a trainable weight matrix of the last layer and i is the class label of the input x. The network is trained by minimizing L.

Transfer Learning

Transfer learning is a ML method where a model developed for a task is reused as the starting point for another task, using knowledge transfer from trained methods. What has been learned in one setting is exploited to improve generalization in another setting, providing an optimization that allows rapid progress and good performance. It is a popular approach in DL where pre-trained models are used as the starting point on computer vision tasks given the vast compute and time resources required to develop NN models from scratch [4]. An example of the steps necessary to implement transfer learning can be seen in Figure 3.13.

This method was also helpful for the task of cataract classification. Two pre-trained Convolution Neural Networks, namely AlexNet and GoogleNet, were used. The last three layers of each net were removed and modified to suit the problem at hand, such as two classes instead of the many more they were fit to distinguish.

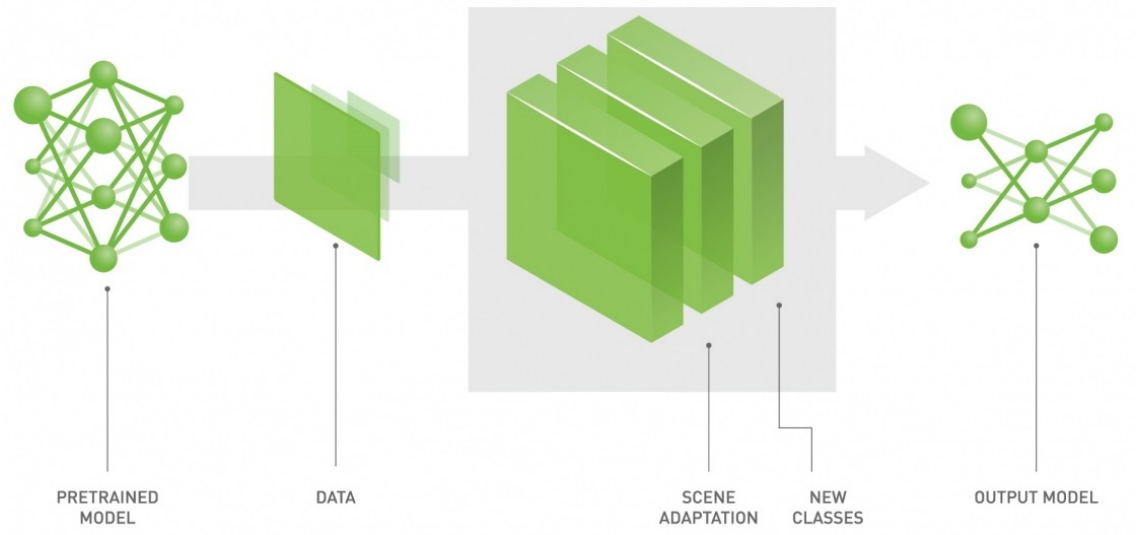

Figure 3.13: Transfer learning steps [25].

3.5 Performance Parameters

The performance of classifiers is usually measured using the following parameters:

- Sensitivity (SE), or true positive rate, measures the proportion of positive images that are correctly labeled [11].
- Specificity (SP), or true negative rate, is the same but for negatives, *i.e.*, the proportion of negative images that are correctly labeled [11].
- Accuracy (AC) is the fraction of correct predictions, *i.e.*, the proportion of data designated with the correct label [11].

The above values can be calculated through the following expressions [11].

$$SE = \frac{TP}{TP + FN} \times 100\%$$

$$SP = \frac{TN}{TN + FP} \times 100\%$$

$$AC = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%$$

where [11]

- True Positive (TP) = Correctly identified, *i.e.*, predicted "cataract" and have cataract.
- False Positive (FP) = Incorrectly identified, *i.e.*, predicted "cataract" and do not have cataract.
- True Negative (TN) = Correctly rejected, *i.e.*, predicted "not cataract" and do not have cataract.
- False Negative (FN) = Incorrectly rejected, *i.e.*, predicted "not cataract" and have cataract.

In this project, a "positive" is an image with signs of cataract and a "negative" is an image without those signs.

Figure 3.14 helps to visualize the four mentioned concepts, which together make up a Confusion Matrix. This type of matrix is a table whose x-axis contains the prediction outcome and the y-axis contains the actual class [19]. Since we only have

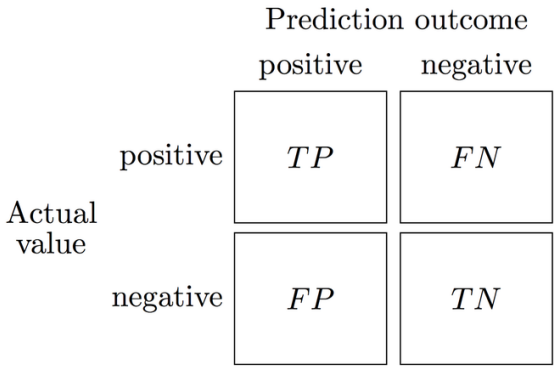

Figure 3.14: Confusion Matrix for a two-class problem [19].

two classes (cataract and not cataract), the Confusion Matrix in this work would only have four cells, filled with the values of TP, FN, FP and TN, as 3.14 shows.

However, a Confusion Matrix can be much bigger if the problem to solve contains multiple classes. In this cases, the diagonal of the table is where the classifier got the correct result, which can be in numerical form (number of correct guesses) or in percentage.

As in most automatic diagnosis system, sensibility is more important than specificity. Although we should always aim for the best results of both parameters, it is crucial to correctly identify the sick individuals as such, rather than miss the disease. Telling someone who does not have cataract that they do will result in a medical assessment to prove the assumption was false, which is not as serious as telling a cataractous patient that he does not have cataract, resulting in no medical evaluation leading to the progression and proper lack of treatment of the disease. The FN, the ones that have cataract but were classified as healthy, are the true concern here, as comparing to the FP, healthy individuals that were misclassified as cataractous. Ideally, all the false results should be minimized, specially the FN, and sensibility should be maximized.

3.6 Code Structure

The core part of the ML code is structured in four functions. The overview of the process steps can be seen in Figure 3.15.

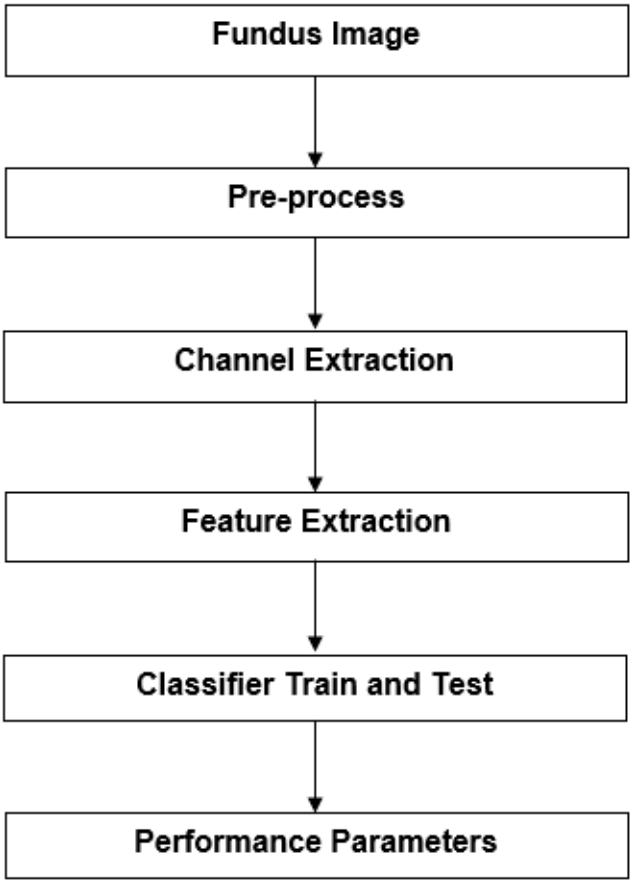

Figure 3.15: Overview of the ML process.

The first is called "preprocess" and pre-processes the RGB fundus images. It padds them so they all become square, crops the maximum possible of the black mask and then resizes the images to 2448×2448 pixels, saving the output in a category that has the same name as the dataset category but adds the suffix "_cropped", keeping separate categories to cataractous and not cataractous images.

After comes the "channel_extraction" function that extracts both the GC or the luminance image based on what the user asks for, from the "_cropped" category created before. It saves the output images in a new directory called the name of the dataset but adding the suffix "_preprocessed". Besides also keeping the healthy and sick images separated, it creates individual categories for each image, with the original name of the image.

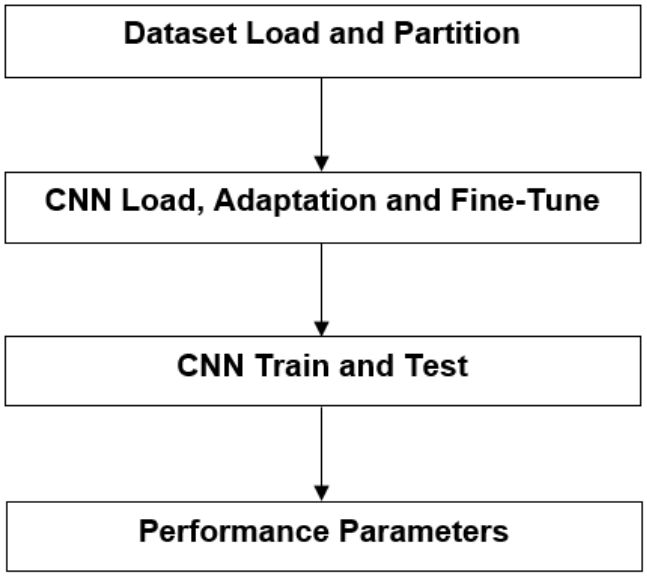

Figure 3.16: Overview of the DL process.

Then there is a function that performs the feature extraction and it is called "cataract-FeatureExtraction". It does the DWT, more specifically, the Haar transform of each image, reads the coefficients from the region of interest, *i.e.*, ignoring the black mask, and plots them into an histogram with ten pre-defined edges. The frequency count of each bin of the histogram corresponds to each one of the ten features. This function saves a text file with the features for each image, plus its histogram and details plot in ".png" files. Lastly, it creates a ".csv" file which contains all the features from all the images in the dataset. It is this file that is later used to train and test the classifiers.

The classifiers are trained in Matlab's Classification Learner app and then tested in the last function that also provides the performance parameters after each test.

The DL part, whose steps can be seen in Figure 3.16, loads the dataset images, dividing them for training and testing. Then the pre-trained network is loaded and the last three layers are adapted to be suited to the problem we are trying to solve. Some training parameters are specified, as well as fine-tune learning rates, and the training begins. After comes the testing that results in some useful performance parameters such as accuracy, sensibility and specificity.

4

Results

This chapter contains the results of the present work, displaying the performance of the ML and DL methods used.

4.1 Classical Machine Learning

4.1.1 Pre-Process and Feature Extraction

Figure 4.1 depicts three fundus images, the first belongs to an healthy individual, the second to a cataractous patient, and the last presents a severe cataract, all from the Not_Cataract and Cataract datasets.

In Figures 4.2, 4.3 and 4.4 we can see the same three fundus images. These RGB images were divided into their channels (Red, Green and Blue, respectively) and calculated into a specific combination of the channels resulting in a Luminance image.

As it can be seen, the channel that presents better contrast and more details is

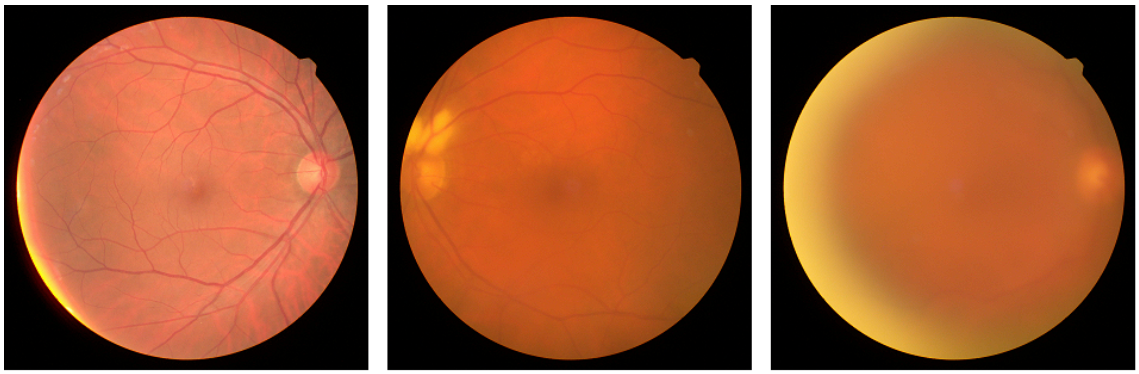

Figure 4.1: Three fundus images, the first without cataract, the second with cataract and the last one with severe cataract.

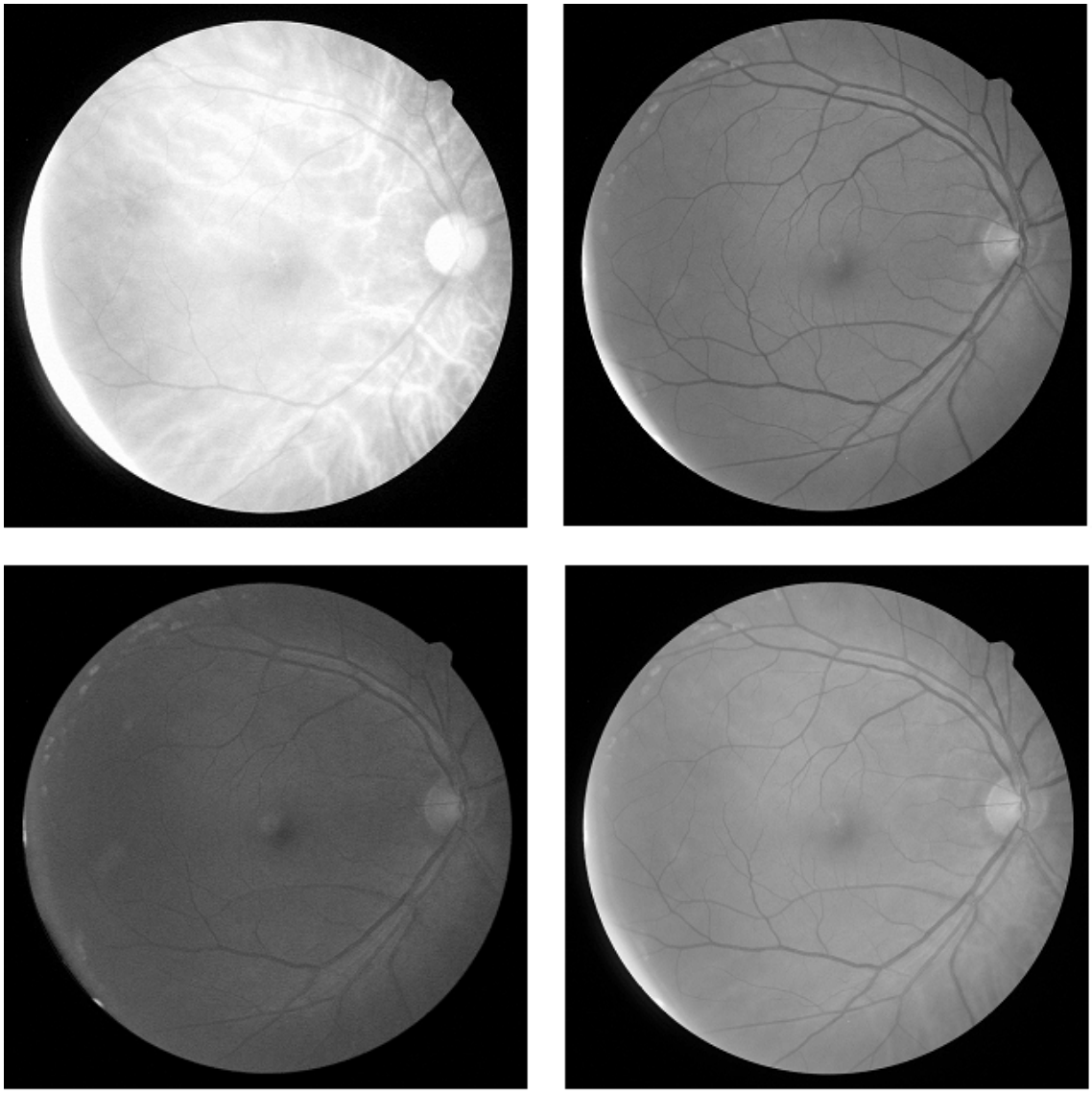

Figure 4.2: Not cataractous fundus image with only the Red, Green and Blue channels individually, plus the Luminance image that combines all of them.

the green one, as previously assumed. Visually we can also inspect that the next best one would be luminance and that the remaining ones are useless in the case of cataract images, specially the blue channel.

Therefore, the two best types of images (GC and luminance) were adopted to solve the automatic classification of cataract problem in this project, as mentioned in the previous section.

The pre-processing included padding, cropping and resizing of the images to the desired size, which in this case was 2448×2448 pixels, since it was the size of more than 80% of the dataset.

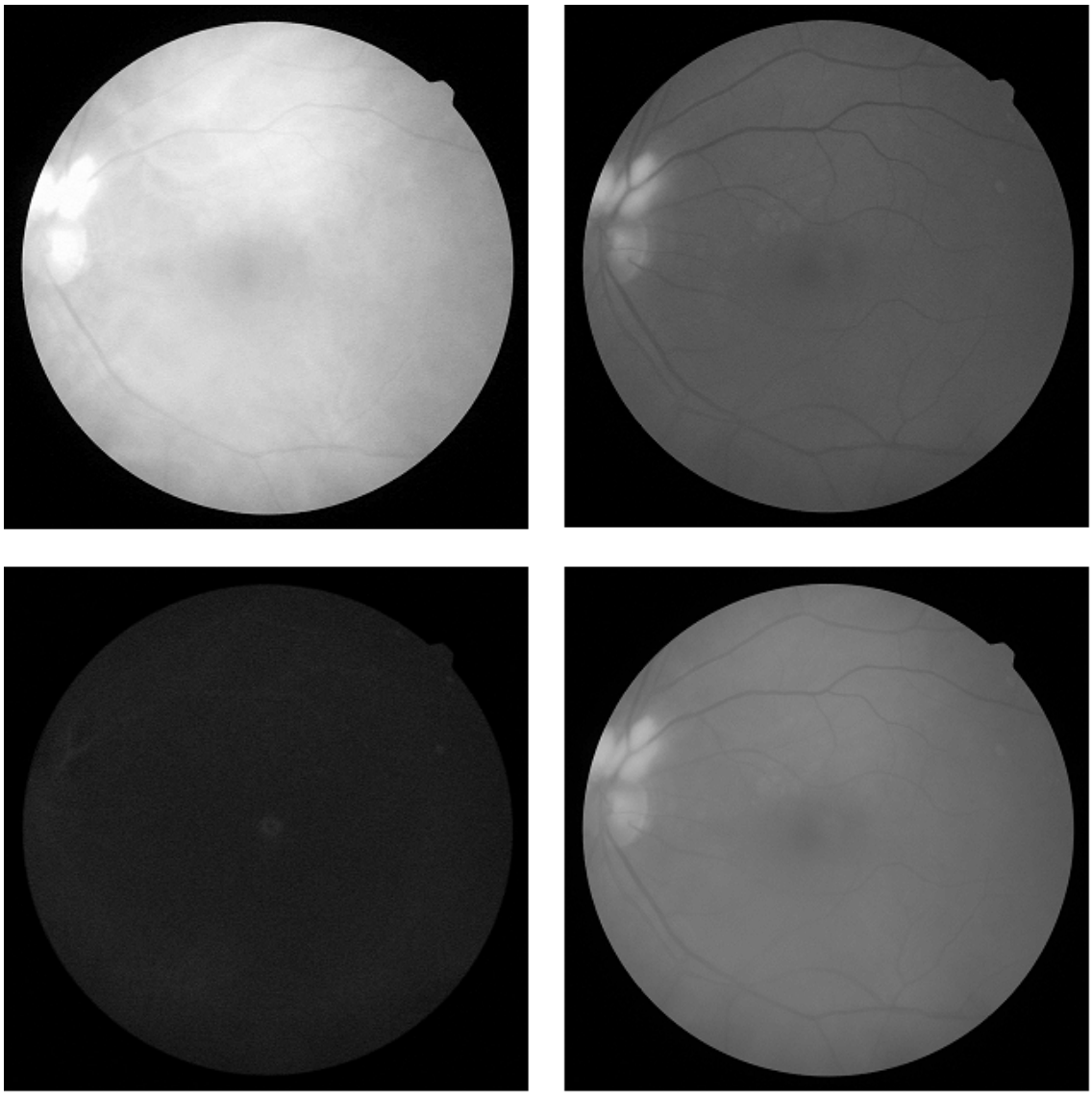

Figure 4.3: Cataract fundus image with only the Red, Green and Blue channels individually, plus the Luminance image that combines all of them.

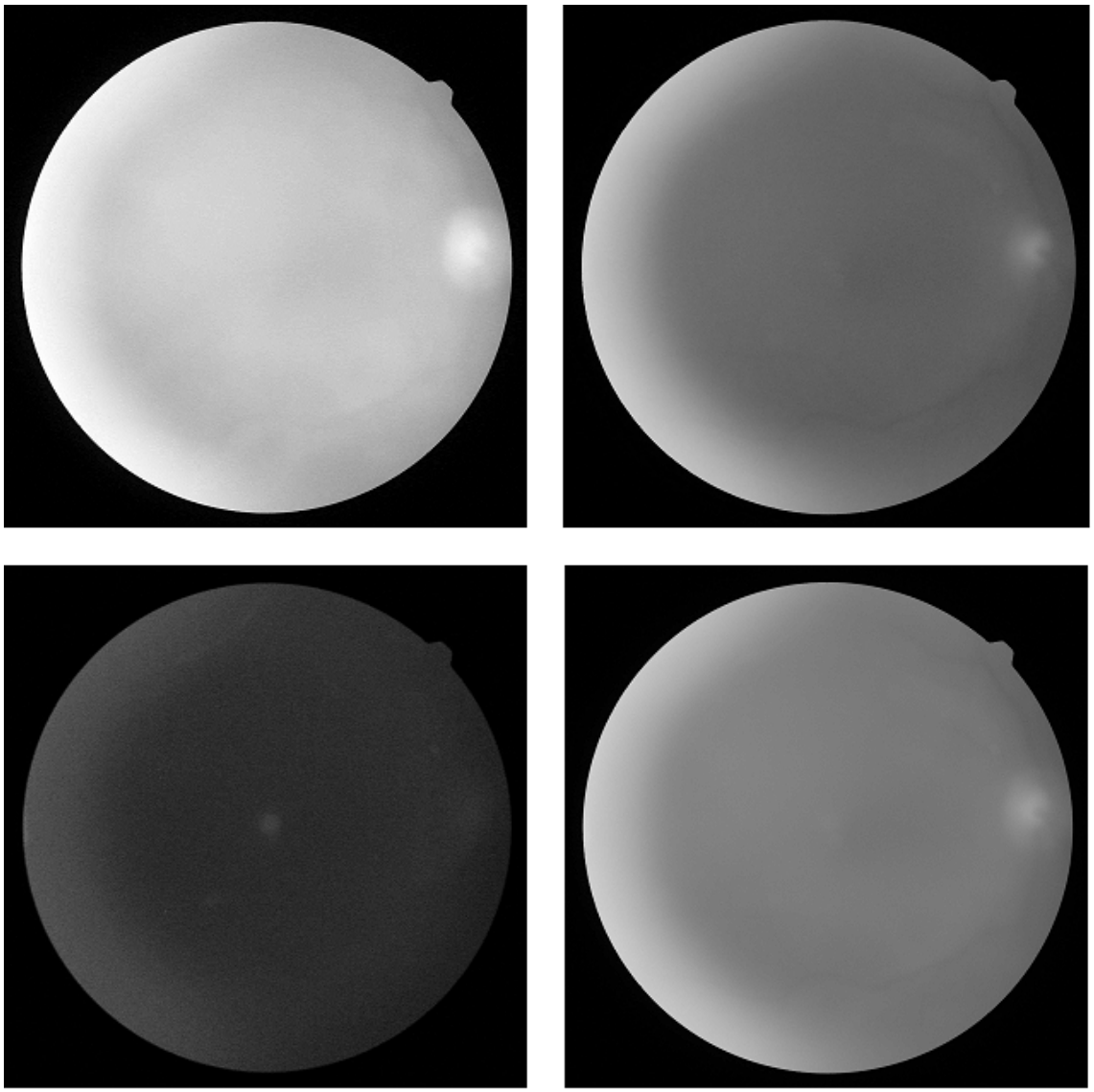

Figure 4.4: Severe cataract fundus image with only the Red, Green and Blue channels individually, plus the Luminance image that combines all of them.

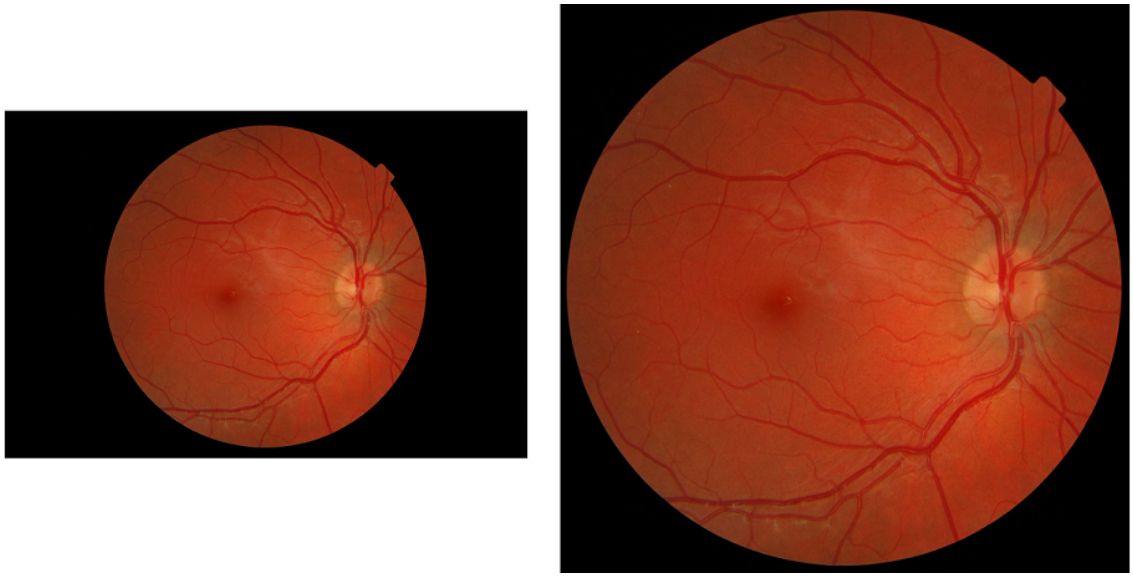

Figure 4.5: Difference between a smaller raw fundus image and the pre-processed one, respectively.

Figures 4.5 and 4.6 show a raw fundus image and the same image after pre-processing. In 4.5, the difference is noticeable in size (hence the images' size mismatch) and cropping, the black mask is much bigger before comparing to the output of the pre-processing step. The padding is not evident, since it is done before the cropping to ensure square images. In Figure 4.6, the differences are evident also in size and cropping, but in the padding too, in the top and bottom of the retinography.

After the image pre-process exemplified in Figures 4.5 and 4.6, the dataset was ready for the feature extraction step.

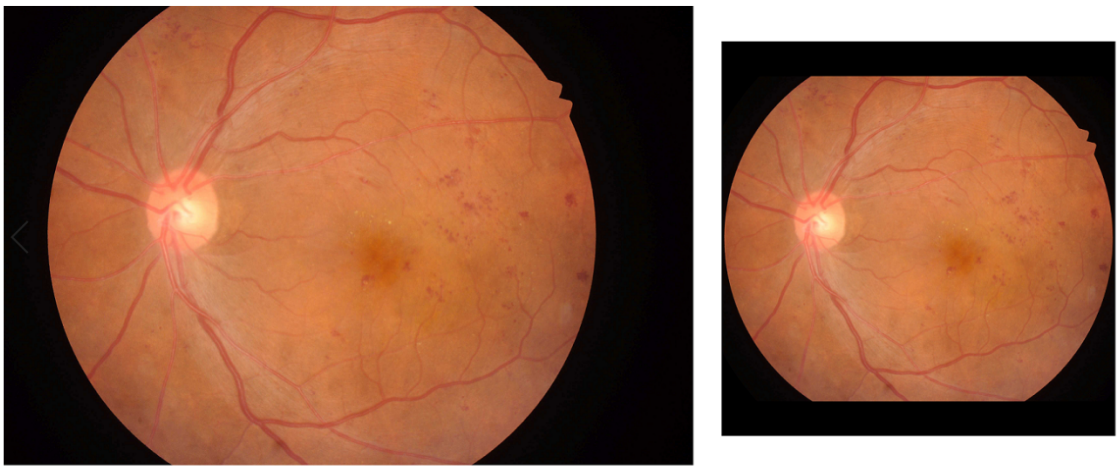

Figure 4.6: Difference between a bigger raw fundus image and the pre-processed one, respectively.

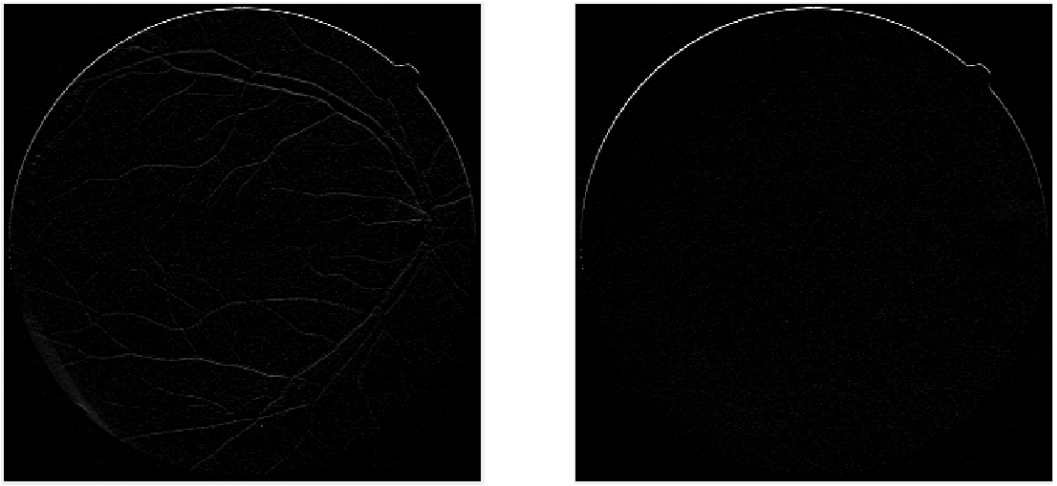

Figure 4.7: Third level horizontal details from the DWT of the model images.

First the Haar transform is applied for each image, from where the coefficients of the DWT are collected and plotted into the details graph, and later, the histogram.

Since the first and last images shown in Figure 4.1 represent well the classes we are trying to distinguish (not cataract and cataract, respectively), they were used to exemplify what was done to the whole dataset and will be called "model images" from now on.

The Haar transform of the GC model images presents the three levels of decomposition and orientation of details of each one of them, obtained through Matlab's app Wavelet Analyzer. From Figure 4.7 we can see the third level horizontal images we are interested in, calculated through Matlab's *haart2()* function. It is from these images that the coefficients of the DWT are retrieved. It is then possible to obtain the images' coefficient plots, seen in Figure 4.8. From them, we build the histograms, which can be seen in Figure 4.9.

It is then from the frequency count of the histograms from Figure 4.9 that the set of features for each image arrives. In this case, the features were the following:

$$features_{not\_cataract} = [399, 833, 2321, 3858, 57483, 1609, 812, 894, 476, 53]$$

$$features_{cataract} = [0, 2, 314, 2008, 66186, 249, 18, 0, 0, 0]$$

The healthy image presents higher values of features than the cataractous one, except in the fifth feature, that corresponds to the center of the histogram, in which both images present similar high values. This difference in the features is then a way of distinguishing the classes.

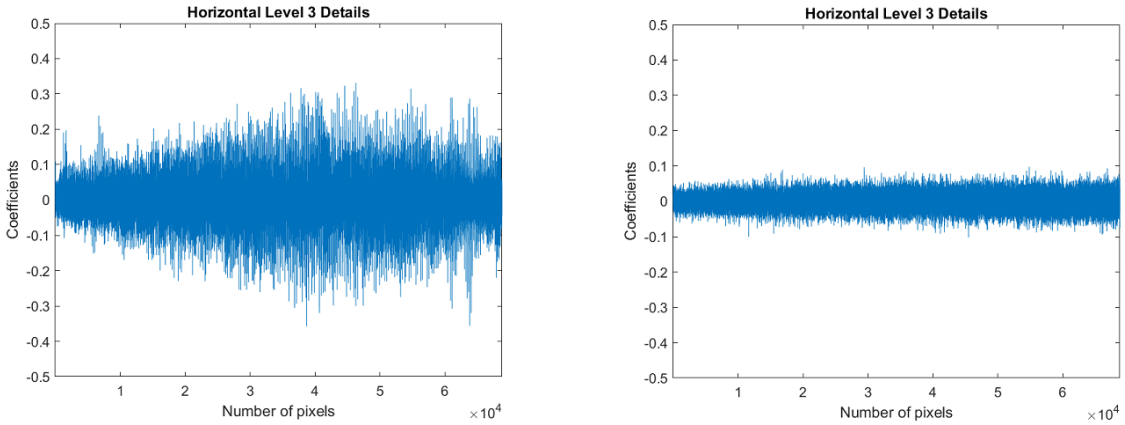

Figure 4.8: Third level horizontal DWT coefficient plots of the model images.

With each image being already pre-processed and the feature extraction done, the data was ready to be analyzed.

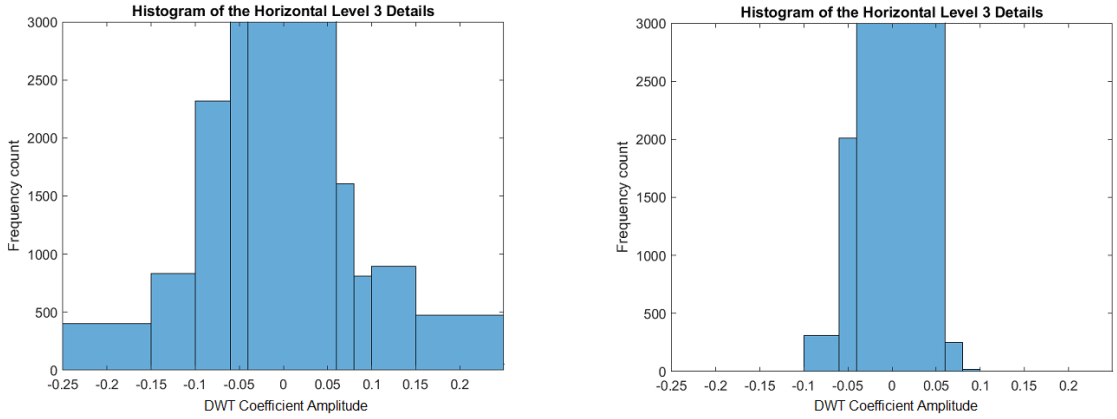

Figure 4.9: Third level horizontal DWT coefficient histograms of the model images.

4.1.2 Classifiers

Prior to choosing any specific classifiers, a preliminary analysis of several ML algorithms was done using Matlab's Classification Learner application. This app provides an intuitive user interface that allows the training and testing of many different ML classifiers based on the data the user inputs.

Feeding the Classification Learner the features extracted by the DWT of both the GC and the luminance images separately, it is possible to do a n-fold cross-validation just by choosing the value of n and the classifier(s) wanted.

Classifier		Accuracy (%)		Training Time (s)	
		5-fold	10-fold	5-fold	10-fold
DT	**Fine Tree**	**81.3**	**81.1**	10.81	**3.40**
	Medium Tree	80.8	80.8	9.90	3.60
	Coarse Tree	80.6	80.5	**9.55**	4.49
SVM	Linear	80.8	80.7	475.11	1257.50
	Quadratic	82.8	82.8	2785.70	5362.50
	Cubic	47.9	44.0	1992.00	3387.00
	Fine Gaussian	**83.8**	**83.9**	**291.36**	**372.09**
	Medium Gaussian	82.2	82.2	362.99	516.87
	Coarse Gaussian	81.0	81.0	435.43	649.71
Ensemble	Boosted Trees	81.6	81.6	**422.63**	**694.44**
	Bagged Trees	**82.9**	**83.0**	470.19	780.40
	Subspace Discriminant	80.8	80.8	423.62	793.07
	Subspace KNN	81.3	81.5	442.33	823.85
	RUSBoosted Trees	80.8	80.6	469.13	879.52

Table 4.1: 5- and 10-fold classification accuracies for the green channel extracted features and training times for each classifier.

It was found appropriate to use 5- and 10-fold validations, taking in consideration the size of the dataset.

Table 4.1 compiles the information gained from this analysis for the green channel extracted features. It shows the classifiers used by type, their accuracy and training time. The values and names in bold represent the best scores. The names of the classifiers are the same Matlab uses, deriving from the kernels used in the case of the SVMs.

Table 4.2 provides the same information as Table 4.1, but for the luminance extracted features.

Some classifier parameters that were default options in the Classification Learner app can be seen in Tables 4.3, 4.4 and 4.5, for both the features extracted from the GC and luminance images. In the case of the SVMs, the Box Constraint Level was 1, the Multiclass Method was One-vs-One and the data was standardized. All Ensemble classifiers had 30 learners, the DT methods had a Maximum Number of Splits of 20 and a Learning Rate of 0.1, whereas the Subspace methods presented the value 5 as the Subspace dimension.

Analyzing the results obtained, the classifiers chosen to use in this project were the most accurate ones, allied to being time efficient and recommended by both supervisors, namely the Fine DT, the Fine Gaussian SVM and the Bagged Trees

Classifier		Accuracy (%)		Training Time (s)	
		5-fold	10-fold	5-fold	10-fold
DT	**Fine Tree**	**84.7**	**84.6**	9.25	3.83
	Medium Tree	83.3	83.1	8.39	2.71
	Coarse Tree	81.0	81.0	**8.06**	**2.48**
SVM	Linear	81.6	81.6	399.00	803.92
	Quadratic	79.2	84.0	2620.00	5778.20
	Cubic	42.6	55.5	1587.40	3114.00
	Fine Gaussian	**86.0**	**86.0**	**191.03**	**363.80**
	Medium Gaussian	83.8	84.0	256.73	488.30
	Coarse Gaussian	81.8	81.8	322.17	614.59
Ensemble	Boosted Trees	84.7	84.5	**328.50**	**649.11**
	Bagged Trees	**85.0**	**85.0**	362.24	719.52
	Subspace Discriminant	81.2	81.2	371.65	732.14
	Subspace KNN	83.5	83.3	407.27	763.39
	RUSBoosted Trees	82.9	82.8	394.92	806.92

Table 4.2: 5- and 10-fold classification accuracies for the luminance extracted features and training times for each classifier.

DT	Max # Splits	Split Criterion	Surrogate Decision Splits
Fine Tree	100	Gini's diversity index	Off
Medium Tree	20		
Coarse Tree	4		

Table 4.3: Parameters of the DT classifiers used in the preliminary analysis.

SVM	Kernel Function	Kernel Scale
Linear	Linear	Automatic
Quadratic	Quadratic	Automatic
Cubic	Cubic	Automatic
Fine Gaussian	Gaussian	0.79
Medium Gaussian	Gaussian	3.20
Coarse Gaussian	Gaussian	13.00

Table 4.4: Parameters of the SVM classifiers used in the preliminary analysis.

Ensemble Classifier	Method	Learner Type
Boosted Trees	AdaBoost	DT
Bagged Trees	Bag	DT
Subspace Discriminant	Subspace	Discriminant
Subspace KNN	Subspace	Nearest Neighbors
RUSBoosted Trees	RUSBoost	DT

Table 4.5: Parameters of the Ensemble classifiers used in the preliminary analysis.

(BT). An SVM classifier was also used in the work of Guo *et al.* [9], which was also a choice factor.

4.1.3 Training and Testing

Randomly half of the dataset was used for training and the other half for testing.

The training of the classifiers was also done in the Classification Learner app. The trained model was then exported to the Matlab workspace from where is it possible to access, save and use it to classify new data.

Table 4.6 shows the performance measures of the classifiers after testing, for all images used, namely GC and luminance (L). The results obtained were good but not great. However, it is already possible to recognize the potential of such approach.

The goal would be to reach for an accuracy (AC) higher than, at least, 90% and improve SE. The SP values are good, but SE is more important in this case.

According to expectation, the GC images performed better than the luminance ones. This could be predicted since, by the naked eye, the first images presented more contrast and kept more details comparing to the latter.

It is also possible to observe that the SVM performed better than the other two classifiers in terms of accuracy and SP, but not in the SE parameter, which is our focus.

In order to keep improving the results and knowing some retinographies could be mislabelled, it was decided to remove the images that were misclassified ($FP+FN$) by the most accurate classifier (the SVM) from the dataset to be analyzed one by one and labelled accordingly. This was a total of 4094 retinographies.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	84.44	72.11	91.19
	SVM	**85.85**	70.61	**94.19**
	BT	84.80	**73.26**	91.11
L	DT	81.86	69.14	89.07
	SVM	**83.42**	72.22	**89.78**
	BT	82.79	**73.64**	87.97

Table 4.6: Performance measures of the classifiers after testing.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	96.17	94.58	96.91
	SVM	**99.31**	**98.48**	**99.71**
	BT	98.39	97.59	98.77
L	DT	95.28	91.08	97.21
	SVM	**96.69**	91.53	**99.06**
	BT	96.09	**92.39**	97.79

Table 4.7: Performance measures of the classifiers after testing, without misclassified images.

While the dataset was free of the misclassified images, it was decided to train and test the three classifiers on it, to see the difference in the results. Values obtained can be seen in Table 4.7.

It is possible to see that without the previously misclassified images, the results improved a lot. Accuracy increased around 15%, SE improved about 25% and SP around 5%, with all these values being close to 100%. The GC images kept being the better classified ones.

However, it is not a good practice to simply eliminate all the images the classifier failed its classification and achieve better results without them. After all, those results would be erroneous, misleading the public to believe the results of the classifier were that great while manipulating the input data to specifically achieve those results.

It was expected the results would improve, but the point is not to eliminate the "problematic" images, but to figure out a way to improve the classifier in order to obtain better results in those specific images, as well as to keep properly guessing the images that were already correctly classified.

This train and test was just an intermediate step to check how much the results would improve, and they improved a lot.

Therefore, after the misclassified images were rightfully labelled, they were put back into the dataset, in their own right classes. Now the dataset has 7 965 cataractous images and 17 252 healthy images, summing up to the previous size of the dataset, 25 217 photographs. From now on, this is how the dataset is used.

Once more, the three classifiers were trained and tested. The parameters obtained, along with the performance measures are shown in Table 4.8.

The results in Table 4.8 can be compared with those of Table 4.6, it is possible

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	91.73	85.89	94.29
	SVM	**93.94**	**91.55**	**94.67**
	BT	93.01	90.22	94.22
L	DT	91.33	82.85	95.46
	SVM	92.16	84.46	**95.91**
	BT	**92.41**	**86.33**	94.32

Table 4.8: Performance measures of the classifiers after testing, after the problematic images re-label.

Quality Dimensions	Possible Values
Color	{0,3}
Focus	{0,2,3}
Contrast	{0,3}
Illumination	{0,3}
Quality	{0,1}

Table 4.9: Variation of the parameters of the pre-existent fundus image quality classification algorithm [26].

to observe a big improvement. Accuracy improved an average of 7.86% for GC images and 9.28% for luminance images, considering all classifiers. Accordingly, SE improved an average of 17.23% and 12.88%, and finally SP improved an average of 2.23% and 6.12%, both respectively for GC and luminance images.

SVM continues to be the best classifier, now in regards to AC, SE and SP in the GC images, but in the luminance images, SVM has the second best value of SE.

Looking forward to keep improving the results, a pre-existent fundus image quality classification algorithm [26] available at Retmarker S.A. was found appropriate to use in the dataset to inspect some key parameters of the images, such as color, focus, contrast, illumination and quality.

This algorithm identifies images suited for manual grading, spotting lack of contrast, uniformity and focus, for example. This would allow us to identify images that would not be appropriate for this type of grading and remove them from the dataset.

The five quality dimensions estimated by the algorithm vary differently within the possible values showed in Table 4.9.

The whole raw dataset was run through the quality algorithm and their classification in the five mentioned parameters was the output.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	92.18	86.00	94.95
	SVM	**94.03**	**91.60**	**95.11**
	BT	93.10	90.94	94.31
L	DT	91.52	82.39	95.57
	SVM	**92.67**	85.10	**96.03**
	BT	92.47	**86.49**	95.13

Table 4.10: Performance measures of the classifiers after testing the dataset without the images excluded by the Quality Algorithm.

Since a great amount of the fundus images performed poorly in this quality algorithm [26], it was not viable to eliminate from the dataset all the images that had the lowest score in the quality dimension. So a stricter criterion was arranged to exclude "bad" images, *i.e.*, only the images that had the worse classification in all the five criteria would be eliminated. This analysis identified $217 + 80$ images from the Cataract and Not_Cataract folders, respectively. In essence, 297 images from the dataset were removed and visually inspected. They were indeed bad quality images, with extremely bad illumination.

The classifiers were then trained and tested again, now without these "bad" images. The results obtained from it can be seen in Table 4.10. In general, the results improved slightly in all the performance parameters evaluated.

However, the Cataract dataset (7686 images) is about half the size of the Not_Cataract (17234 images), thus why SP has been higher than SE. Since the in-training classifier sees more than double the images without cataract than with cataract, it is not a surprise that it distinguishes healthy images better than cataractous ones, due to class imbalance.

Another training and test was done where the classes were randomly balanced. All the cataract images were accounted for but only some healthy ones were used, the same number of cataract ones. Also, from now on, only the GC images were used since from the beginning they were the ones that performed better. The results can be seen in Table 4.11.

At last the SE increased, being higher than SP, which decreased a bit. Overall the results show a good compromise between these two parameters, since the focus was to get a higher SE. The accuracy is better, which was also a goal. These were the best results so far, but some other strategies were used to see if it was possible to increase them even further.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	92.36	94.76	89.96
	SVM	**94.29**	**96.71**	**91.86**
	BT	93.56	96.09	91.03

Table 4.11: Performance measures of the classifiers after testing the dataset without the images excluded by the Quality Algorithm and with balanced classes.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	94.47	96.96	91.83
	SVM	**96.20**	**98.69**	**93.57**
	BT	95.62	97.68	93.44

Table 4.12: Performance measures of the classifiers after testing the dataset without the images excluded by the Quality Algorithm and with balanced classes but with 70% of training images and 30% for testing.

Since the results from Guo *et al.* [9] were obtained using 70% of the dataset to train and the remaining 30% to test, for the sake of comparison, the results from Table 4.11 were calculated again for that percentage ratio, rather than the one we have been using so far, which is 50%/50%. The new parameters can be found in Table 4.12. The results are better, as expected, exceeding the average accuracy of 90.9% obtained in the work of Guo *et al.* [9].

4.1.3.1 Data Augmentation

As a way of enlarging the dataset, improving variety and generalization, it was decided to do Data Augmentation.

"Data Augmentation is a strategy that enables users to significantly increase the diversity of data available for training models, without actually collecting new data" [13]. Commonly used techniques are cropping, padding, rotating and flipping, to name a few [13].

The Data Augmentation strategy chosen was to apply a 45° rotation to every image. Doubling the fundus images on which the classifier is trained may provide a better SE value and hopefully improve the overall accuracy.

After the augmentation, the classifiers were again trained and tested and the results obtained can be seen in Table 4.13, for the Data Augmentation only on the training images. Furthermore, Table 4.14 shows the performance measures of the classifiers after doing Data Augmentation on all images (train and test). Knowing balanced

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	90.02	81.30	93.91
	SVM	**92.71**	**88.79**	**94.46**
	BT	92.49	88.60	94.23

Table 4.13: Performance measures of the classifiers after Data Augmentation on the training images.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	89.90	87.20	91.59
	SVM	**91.95**	**90.77**	**92.69**
	BT	91.74	90.49	92.52

Table 4.14: Performance measures of the classifiers after Data Augmentation on all images.

classes give better results, Table 4.15 presents the same process as in 4.14, but with balanced classes.

One can conclude that the best accuracies are mostly the ones from Table 4.13, with the SVM consistently being the best classifier. Regarding *se*, the results from Table 4.15 are the best, reinforcing the fact that balancing the classes, the *se* is better, even if the accuracy is not, although in previous sections both of the performance parameters improved when the classes were balanced.

However, all Data Augmentation results are worse than the ones obtained in Table 4.11. Since the dataset is bigger and a bit different from the one used before, the lower results may imply a better generalization to a real-life dataset, so reduced performance parameters might not be a setback.

Instead of analyzing the regular images and the rotated images and then classifying each one of them separately, it was thought to include in the feature extraction method the rotation of the image being analyzed and add its features to the ones from the regular image. Therefore, each image would result in a vector of 20 features, instead of the previous 10 features.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	DT	90.03	92.12	88.12
	SVM	**91.94**	**93.61**	**90.41**
	BT	91.62	93.34	90.05

Table 4.15: Performance measures of the classifiers after Data Augmentation on all images with balanced classes.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	SVM	92.86	95.56	91.65

Table 4.16: Performance measures of the SVM with 20 features from the regular and the rotated images, with balanced classes.

Another training and testing were done and the results can be seen in Table 4.16, this time only for the best classifier so far, the SVM, plus with balanced classes.

The results were better than the other Data Augmentation ones, both in accuracy and *se*, but still lower than the ones from Table 4.11.

4.1.3.2 Feature Selection

A feature selection was done to the results from Table 4.10. For a matter of ease and quickness of the procedure, the feature selection process was done using the software Weka, since Matlab does not use the exact same well-known methods as Weka.

Three different feature selection methods were used. First the Wrapper was applied to the features, then the method Information Gain and finally the Gain Ratio. All of these were used alone and then combined.

The Wrapper is based on greedy search algorithms, it evaluates all possible combinations of the features and selects the one that produces the best result. The features this method selected can be seen in Table 4.17.

Information Gain selects the attributes with higher mutual information. Table 4.18 shows the performance obtained regarding this method.

Gain Ratio measures the proportion of Information Gain to the intrinsic information of each feature, choosing the best ones. The results using this method can be seen in Table 4.19.

Tables 4.20 and 4.21 show the results obtained using the methods Information Gain + Wrapper and Gain Ratio + Wrapper, respectively. These were used in this specific order, *i.e.*, first Information Gain was applied and then Wrapper was applied to the resulting selected features by the Information Gain.

None of the feature selection methods were beneficial for the SVM in any parameter. For the DT, using the Wrapper or the combination Gain Ratio + Wrapper is useful in terms of SE, but roughly the same in accuracy. Just using Wrapper is favorable

Images	Classifier	Selected Features	AC (%)	SE (%)	SP (%)
GC	DT	2,3,4,5,6,10	92.18	88.50	93.83
	SVM	1,2,3,4,6,7,8,9,10	89.68	86.05	91.30
	BT	2,3,4,5,6,7,10	**93.42**	**90.25**	**94.84**

Table 4.17: Wrapper feature selection and performance parameters.

Images	Classifier	Selected Features	AC (%)	SE (%)	SP (%)
GC	DT	1,2,3,5,6,7,8,9	91.53	85.74	94.12
	SVM		83.22	90.51	**94.44**
	BT		**92.57**	**89.28**	94.04

Table 4.18: Information Gain feature selection and performance parameters.

Images	Classifier	Selected Features	AC (%)	SE (%)	SP (%)
GC	DT	1,2,3,5,6,7,8,9,10	91.51	84.69	94.56
	SVM		**93.42**	**90.80**	**94.59**
	BT		92.83	89.96	94.11

Table 4.19: Gain Ratio feature selection and performance parameters.

Images	Classifier	Selected Features	AC (%)	SE (%)	SP (%)
GC	DT	2,3,5	91.78	88.63	93.19
	SVM	2,7	89.05	85.68	90.55
	BT	2,3,5,6,9	**92.92**	**89.31**	**94.53**

Table 4.20: Information Gain and Wrapper feature selection and performance parameters.

Images	Classifier	Selected Features	AC (%)	SE (%)	SP (%)
GC	DT	2,3,5,10	92.11	88.79	93.60
	SVM	2,5,8,10	89.53	84.80	91.65
	BT	2,3,5,6,10	**93.14**	**90.07**	**94.52**

Table 4.21: Gain Ratio and Wrapper feature selection and performance parameters.

for the BT, since it improves all the performance parameters. Still, the results from Table 4.11 continue to be the best ones yet.

4.1.3.3 Horizontal, Vertical and Diagonal Features

As earlier mentioned, the features used to represent each image come from the third-level horizontal coefficients of the Haar transform. However, this DWT gives two more "types" of coefficients, the vertical and diagonal ones. As these were not used by recommendation of Guo *et al.* 's [9] work, it was decided to incorporate them into the method to further investigate their usefulness. Plus, they were considered with balanced classes, since it has been beneficial.

First, only the vertical coefficients were considered, since they simulate a 90° rotation of the input images. And then all of the coefficients were considered, making each image into 30 features (10 for each orientation). The performance measures can be seen in Tables 4.22 and 4.23, respectively.

In terms of accuracy, it is better to use all the features (horizontal, vertical and diagonal), rather than just the vertical ones. Nevertheless, the latter is beneficial for the SE. Both these results (Tables 4.22 and 4.23) are still not as good as the ones from Table 4.11 though.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	SVM	92.62	94.99	90.23

Table 4.22: Performance measures of the SVM with 10 vertical features.

Images	Classifier	AC (%)	SE (%)	SP (%)
GC	SVM	93.53	94.68	92.38

Table 4.23: Performance measures of the SVM with 30 features (horizontal, vertical and diagonal).

4.2 Deep Learning

4.2.1 Pre-trained Convolutional Neural Networks

AlexNet was the first pre-trained CNN used here to solve the problem of cataract classification. The last three layers of this 25-layer net were re-purposed to label two classes instead of the default, which were up to a thousand classes. The layers of the network can be seen in Figure 4.10, while the structure of it is shown in Figure 4.11 (split in half to fit the page).

```
 1   'data'    Image Input
 2   'conv1'   Convolution
 3   'relu1'   ReLU
 4   'norm1'   Cross Channel Normalization
 5   'pool1'   Max Pooling
 6   'conv2'   Grouped Convolution
 7   'relu2'   ReLU
 8   'norm2'   Cross Channel Normalization
 9   'pool2'   Max Pooling
10   'conv3'   Convolution
11   'relu3'   ReLU
12   'conv4'   Grouped Convolution
13   'relu4'   ReLU
14   'conv5'   Grouped Convolution
15   'relu5'   ReLU
16   'pool5'   Max Pooling
17   'fc6'     Fully Connected
18   'relu6'   ReLU
19   'drop6'   Dropout
20   'fc7'     Fully Connected
21   'relu7'   ReLU
22   'drop7'   Dropout
23   'fc8_2'   Fully Connected
24   ''        Softmax
25   ''        Classification Output
```

Figure 4.10: Layers of AlexNet.

The input size the net requires is 227×227 pixels, so the dataset is resized before entering the first layer.

The training options used included an initial learn rate of 0.0001, 20 as the maximum number of epochs and a mini batch size of 128, which all proved to be beneficial.

The CNN was then trained with the dataset prepared in this work, based on the layers and training options. After came the testing, from which resulted the values of Table 4.24.

The results are good for the first attempt, better than the first ones achieved with ML. However, they are not better than those from Table 4.11. Maybe AlexNet is

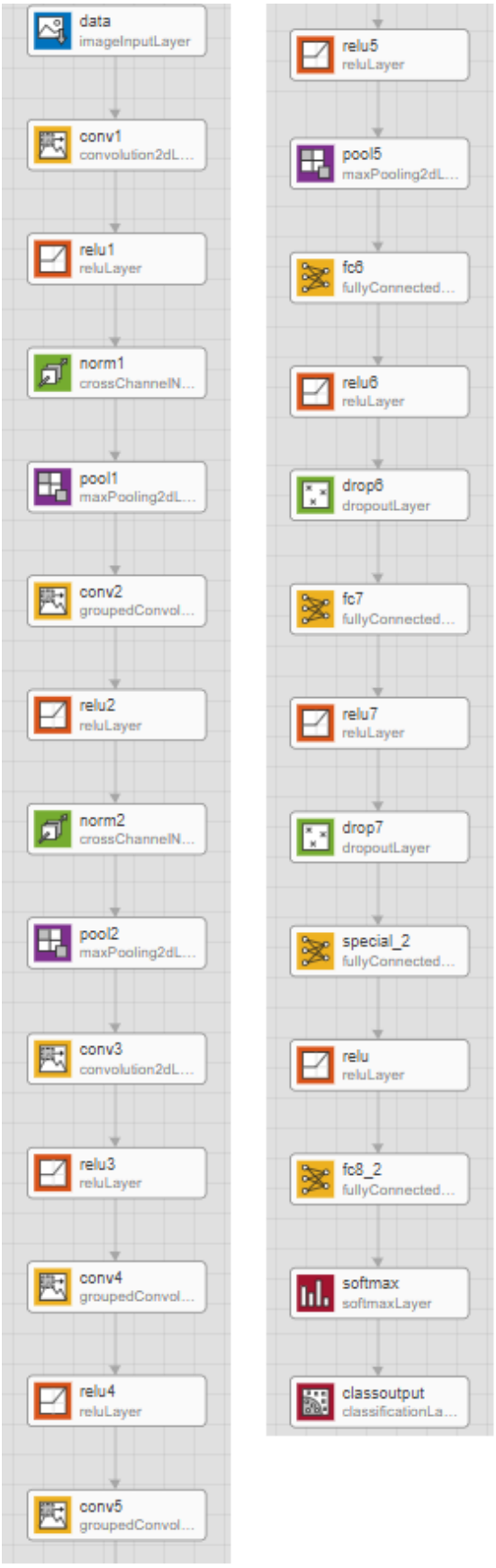

Figure 4.11: Structure of AlexNet.

Classifier	AC (%)	SE (%)	SP (%)
AlexNet	92.45	92.33	92.56

Table 4.24: Performance parameters of AlexNet.

Classifier	AC (%)	SE (%)	SP (%)
GoogleNet	91.41	92.02	91.14

Table 4.25: Performance parameters of GoogleNet.

too simple, perhaps a deeper net would accomplish higher results. Accordingly, a more complex CNN was used to try to achieve better accuracy and SE. GoogleNet was then loaded and modified to suit our purpose, the same way AlexNet did, by adjusting its final layers. The layering is showed in Figure 4.12 (split in half to fit the page).

The network is considerably bigger than the previous one used, having 144 layers, as seen in Figure 4.12, and only accepts input images of size 224×224 pixels. Using the same training options, the results obtained are presented in Table 4.25. It shows worse performance, comparing to Table 4.24, which made us go back to using AlexNet.

AlexNet was then used again, but this time training it with an SVM to do the classification. Since the SVM was the best classifier in the classical ML section, maybe it can aid here, providing a more effective classification. Basically, the features are extracted by the NN, but a SVM does the final classification. What was done was an experimentation to see in which fully connected (FC) layer the SVM could be put to maximize the final accuracy of the classifier. Remembering the layers from AlexNet, presented in Figure 4.10, there are three FC layers: $fc6$, $fc7$ and $fc8$, the SVM was introduced in each one of them separately and the performance was observed and written in Table 4.26. Plus a study of the performance of each available SVM kernel was also done and can be found in Table 4.26 as well.

The best kernel is the Gaussian, as in the classical ML section, but the linear SVM kernel comes in a close second. Regarding the FC layers, the sooner the SVM classification is done, the better, since the best accuracy is obtained when the classification is done were the $fc6$ would be, with the input from the previous layer that would feed the $fc6$, rather than in the place of $fc7$ or $fc8$. This result is a bit counterintuitive, in the sense that we would expect the best result to come out deeper in the network and not as soon as possible. However, that was what happened in this case, implying that the more transformations are done in each deeper layer, the worse the

1	'data'	Image Input
2	'conv1-7x7_s2'	Convolution
3	'conv1-relu_7x7'	ReLU
4	'pool1-3x3_s2'	Max Pooling
5	'pool1-norm1'	Cross Channel Normalization
6	'conv2-3x3_reduce'	Convolution
7	'conv2-relu_3x3_reduce'	ReLU
8	'conv2-3x3'	Convolution
9	'conv2-relu_3x3'	ReLU
10	'conv2-norm2'	Cross Channel Normalization
11	'pool2-3x3_s2'	Max Pooling
12	'inception_3a-1x1'	Convolution
13	'inception_3a-relu_1x1'	ReLU
14	'inception_3a-3x3_reduce'	Convolution
15	'inception_3a-relu_3x3_reduce'	ReLU
16	'inception_3a-3x3'	Convolution
17	'inception_3a-relu_3x3'	ReLU
18	'inception_3a-5x5_reduce'	Convolution
19	'inception_3a-relu_5x5_reduce'	ReLU
20	'inception_3a-5x5'	Convolution
21	'inception_3a-relu_5x5'	ReLU
22	'inception_3a-pool'	Max Pooling
23	'inception_3a-pool_proj'	Convolution
24	'inception_3a-relu_pool_proj'	ReLU
25	'inception_3a-output'	Depth concatenation
26	'inception_3b-1x1'	Convolution
27	'inception_3b-relu_1x1'	ReLU
28	'inception_3b-3x3_reduce'	Convolution
29	'inception_3b-relu_3x3_reduce'	ReLU
30	'inception_3b-3x3'	Convolution
31	'inception_3b-relu_3x3'	ReLU
32	'inception_3b-5x5_reduce'	Convolution
33	'inception_3b-relu_5x5_reduce'	ReLU
34	'inception_3b-5x5'	Convolution
35	'inception_3b-relu_5x5'	ReLU
36	'inception_3b-pool'	Max Pooling
37	'inception_3b-pool_proj'	Convolution
38	'inception_3b-relu_pool_proj'	ReLU
39	'inception_3b-output'	Depth concatenation
40	'pool3-3x3_s2'	Max Pooling
41	'inception_4a-1x1'	Convolution
42	'inception_4a-relu_1x1'	ReLU
43	'inception_4a-3x3_reduce'	Convolution
44	'inception_4a-relu_3x3_reduce'	ReLU
45	'inception_4a-3x3'	Convolution
46	'inception_4a-relu_3x3'	ReLU
47	'inception_4a-5x5_reduce'	Convolution
48	'inception_4a-relu_5x5_reduce'	ReLU
49	'inception_4a-5x5'	Convolution
50	'inception_4a-relu_5x5'	ReLU
51	'inception_4a-pool'	Max Pooling
52	'inception_4a-pool_proj'	Convolution
53	'inception_4a-relu_pool_proj'	ReLU
54	'inception_4a-output'	Depth concatenation
55	'inception_4b-1x1'	Convolution
56	'inception_4b-relu_1x1'	ReLU
57	'inception_4b-3x3_reduce'	Convolution
58	'inception_4b-relu_3x3_reduce'	ReLU
59	'inception_4b-3x3'	Convolution
60	'inception_4b-relu_3x3'	ReLU
61	'inception_4b-5x5_reduce'	Convolution
62	'inception_4b-relu_5x5_reduce'	ReLU
63	'inception_4b-5x5'	Convolution
64	'inception_4b-relu_5x5'	ReLU
65	'inception_4b-pool'	Max Pooling
66	'inception_4b-pool_proj'	Convolution
67	'inception_4b-relu_pool_proj'	ReLU
68	'inception_4b-output'	Depth concatenation
69	'inception_4c-1x1'	Convolution
70	'inception_4c-relu_1x1'	ReLU
71	'inception_4c-3x3_reduce'	Convolution
72	'inception_4c-relu_3x3_reduce'	ReLU
73	'inception_4c-3x3'	Convolution
74	'inception_4c-relu_3x3'	ReLU
75	'inception_4c-5x5_reduce'	Convolution
76	'inception_4c-relu_5x5_reduce'	ReLU
77	'inception_4c-5x5'	Convolution
78	'inception_4c-relu_5x5'	ReLU
79	'inception_4c-pool'	Max Pooling
80	'inception_4c-pool_proj'	Convolution
81	'inception_4c-relu_pool_proj'	ReLU
82	'inception_4c-output'	Depth concatenation
83	'inception_4d-1x1'	Convolution
84	'inception_4d-relu_1x1'	ReLU
85	'inception_4d-3x3_reduce'	Convolution
86	'inception_4d-relu_3x3_reduce'	ReLU
87	'inception_4d-3x3'	Convolution
88	'inception_4d-relu_3x3'	ReLU
89	'inception_4d-5x5_reduce'	Convolution
90	'inception_4d-relu_5x5_reduce'	ReLU
91	'inception_4d-5x5'	Convolution
92	'inception_4d-relu_5x5'	ReLU
93	'inception_4d-pool'	Max Pooling
94	'inception_4d-pool_proj'	Convolution
95	'inception_4d-relu_pool_proj'	ReLU
96	'inception 4d-output'	Depth concatenation
97	'inception_4e-1x1'	Convolution
98	'inception_4e-relu_1x1'	ReLU
99	'inception_4e-3x3_reduce'	Convolution
100	'inception_4e-relu_3x3_reduce'	ReLU
101	'inception_4e-3x3'	Convolution
102	'inception_4e-relu_3x3'	ReLU
103	'inception_4e-5x5_reduce'	Convolution
104	'inception_4e-relu_5x5_reduce'	ReLU
105	'inception_4e-5x5'	Convolution
106	'inception_4e-relu_5x5'	ReLU
107	'inception_4e-pool'	Max Pooling
108	'inception_4e-pool_proj'	Convolution
109	'inception_4e-relu_pool_proj'	ReLU
110	'inception_4e-output'	Depth concatenation
111	'pool4-3x3_s2'	Max Pooling
112	'inception_5a-1x1'	Convolution
113	'inception_5a-relu_1x1'	ReLU
114	'inception_5a-3x3_reduce'	Convolution
115	'inception_5a-relu_3x3_reduce'	ReLU
116	'inception_5a-3x3'	Convolution
117	'inception_5a-relu_3x3'	ReLU
118	'inception_5a-5x5_reduce'	Convolution
119	'inception_5a-relu_5x5_reduce'	ReLU
120	'inception_5a-5x5'	Convolution
121	'inception_5a-relu_5x5'	ReLU
122	'inception_5a-pool'	Max Pooling
123	'inception_5a-pool_proj'	Convolution
124	'inception_5a-relu_pool_proj'	ReLU
125	'inception_5a-output'	Depth concatenation
126	'inception_5b-1x1'	Convolution
127	'inception_5b-relu_1x1'	ReLU
128	'inception_5b-3x3_reduce'	Convolution
129	'inception_5b-relu_3x3_reduce'	ReLU
130	'inception_5b-3x3'	Convolution
131	'inception_5b-relu_3x3'	ReLU
132	'inception_5b-5x5_reduce'	Convolution
133	'inception_5b-relu_5x5_reduce'	ReLU
134	'inception_5b-5x5'	Convolution
135	'inception_5b-relu_5x5'	ReLU
136	'inception_5b-pool'	Max Pooling
137	'inception_5b-pool_proj'	Convolution
138	'inception_5b-relu_pool_proj'	ReLU
139	'inception_5b-output'	Depth concatenation
140	'pool5-7x7_s1'	Average Pooling
141	'pool5-drop_7x7_s1'	Dropout
142	'new_fc'	Fully Connected
143	'prob'	Softmax
144	'new_classoutput'	Classification Output

Figure 4.12: Layers of GoogleNet.

Fully Connected Layer	SVM Kernel	AC (%)
$fc6$	Gaussian	**92.82**
	Linear	92.54
	Polynomial	90.74
$fc7$	Gaussian	**92.68**
	Linear	92.35
	Polynomial	88.84
$fc8$	Gaussian	**92.28**
	Linear	91.70
	Polynomial	83.03

Table 4.26: Accuracy of AlexNet with different kernels of a SVM classifier.

SVM will perform.

Because we have still not reached with DL the best results achieved with classical ML, one more attempt was done to try and get better performances.

4.2.2 Fully trained Convolutional Neural Network

Another experiment was the creation of a CNN from scratch, completely training it. Nevertheless, the layering was inspired by the one from AlexNet.

The layers of the network can be seen in Figure 4.13, two more layers (one FC and one RELU) were added comparing to AlexNet because this configuration showed better results.

The training options were the same as AlexNet, except the learn rate schedule used was "piecewise", instead of "none", since it proved more effective. The CNN results are shown in Table 4.27.

The results are above 90%, which is very good for a CNN trained in just a couple thousands of images, but are inferior than the ones previously obtained, as could be expected.

Classifier	AC (%)	SE (%)	SP (%)
CNN	90.71	90.79	90.21

Table 4.27: Performance parameters of the fully trained Convolution Neural Network.

```
 1   'data'          Image Input
 2   'conv1'         Convolution
 3   'relu1'         ReLU
 4   'norm1'         Cross Channel Normalization
 5   'pool1'         Max Pooling
 6   'conv2'         Grouped Convolution
 7   'relu2'         ReLU
 8   'norm2'         Cross Channel Normalization
 9   'pool2'         Max Pooling
10   'conv3'         Convolution
11   'relu3'         ReLU
12   'conv4'         Grouped Convolution
13   'relu4'         ReLU
14   'conv5'         Grouped Convolution
15   'relu5'         ReLU
16   'pool5'         Max Pooling
17   'fc6'           Fully Connected
18   'relu6'         ReLU
19   'drop6'         Dropout
20   'fc7'           Fully Connected
21   'relu7'         ReLU
22   'drop7'         Dropout
23   'special_2'     Fully Connected
24   'relu'          ReLU
25   'fc8_2'         Fully Connected
26   'softmax'       Softmax
27   'classoutput'   Classification Output
```

Figure 4.13: Layers of the fully trained Convolutional Neural Network.

5

Conclusions

This work was focused on developing approaches to achieve automatic cataract classification in fundus images.

Besides the methods described in Chapter 2, there are not many automatic published methods for cataract classification using fundus images, as earlier work focuses on other type of images and different imaging methods that are not useful in this case. As specified in Chapter 3, all the best methods and processes investigated in Chapter 2 were used, but much more was added.

The dataset prepared in this work is much bigger than the ones used in previous reported researches. It is also much more diverse and inclusive since it contains fundus images from multiple sources that also include some with different types of lesions and pathologies.

Following the outcomes presented in Chapter 4, this study suggests reliable yet cost effective and simple approaches to the problem described. The features used were appropriate, giving the classifiers a good way of distinguishing each class.

One can conclude that the best Machine Learning classifier was the Support Vector Machine, but with Bagged Trees coming in a close second. The Decision Tree was the worse of the three, giving good results (higher than 90%) but not as good as the other classifiers. The best performance achieved with 50% of training data and the other 50% for testing with the SVM was an accuracy of 94.29%, a sensitivity of 96.71% and a specificity of 91.86%, having balanced classes, *i.e.*, the same number of cataracts as of not cataracts for training.

The Data Augmentation strategy was better when it was done only to the training images and not the whole dataset, also giving considerable results, but still lower than using the "regular" dataset.

Regarding the Feature Selection methods, just the Wrapper or its combination with

the Gain Ratio were favorable for the Decision Tree, just the Wrapper was beneficial for the Bagged Trees, but none of them were worthy in the case of the SVM.

As Guo *et al.* [9] mentioned, the horizontal features from the Haar DWT really do offer the best results, since the just the vertical or the combination of the three coefficient directions (vertical, horizontal and diagonal) do not measure up.

Concerning Deep Learning, the results were inferior. Transfer Learning, specially with AlexNet, is more effective than the Convolutional Neural Network built from scratch, possibly for the lack of proper parameter tuning to adapt to the problem at hand or because the size of the training dataset might be insufficient for such approach.

Nevertheless, the experiment results illustrate the value and effectiveness of the approach, showing that the method is useful in a practical application of cataract detection. It has the potential to reduce the burden of experienced ophthalmologists as well as the diagnosing underlying costs, helping cataract patients to get an early intervention, improving health care quality.

It was not possible to obtain neither the computer code nor the dataset used in similar works, but it was still possible to compare results, even though we have to keep in mind that they were not obtained in the same conditions. However, the results achieved remain valid and relevant in their own context. The closest one in terms of performance is perhaps the method of Guo *et al.* [9], since the first part of the present work is very similar to it. Guo *et al.* achieved an average accuracy of 90.9% using 70% of the dataset to train and the remaining 30% to test. For comparison, the dataset used was also split in the same way and the results were an accuracy of 96.20%, which is an improvement of more than 5%, a great progress in these high accuracies.

Comparing to all the methods reviewed in Chapter 2, the results obtained were better than all of them, except one. The only exception was the work of Zheng *et al.* [41] that managed to get an accuracy of 95.22%. However, the dataset is about 54 times smaller and the authors used different methods, as other researchers did, so it is not easy to compare them. Also, this analysis is assuming that their dataset was divided equally for training and testing, which may not be true since they do not mention it. However, if the train and test ratio was 70%/30%, the results in this work are better than theirs, but this analysis is very superficial.

Despite the encouraging performance achieved by the present system, some issues remain. A limiting factor could be the lack of certainty of the ground truth of the

fundus images used. In other words, if the images are labelled incorrectly, they are put in the wrong class, affecting the overall performance of the method due to faulty training. Improving the quality of the dataset would most likely boost the accuracy.

Reinforcing the last point, another issue was the fact that a few dataset images that did not have a clear class were graded by a non-expert, which may be a source of error. Since the reference standard of those images was not clinically confirmed, some cataracts may have been missed and even if the method would find them, it was considered wrong, since the ground truth was incorrect. As said before, the quality of the Artificial Intelligence approach reflects the quality of the dataset, so a great dataset with a firm ground truth is what we should look for.

Another factor is the huge amount of fundus images with bad illumination. The acquisition of these images should be done more carefully.

Although the dataset is quite big compared to other reported researches in the field, it is not be big enough to properly train a robust Convolution Neural Network. So a bigger amount of images would probably result in a better Deep Learning method.

Future Work

Since the size of the images used in the Machine Learning part reflected the size of the majority of the dataset, a study could be done to research the optimal image size for both Machine Learning and Deep Learning algorithms. It would also be interesting to study and compare fundus images acquired by each specific camera, pointing out the best ones for the job, for example.

Several more tests could be done to assure the robustness of the methods, such as defocusing the previously rightly labeled images and running them by the classifier to check if the performance is maintained, for example.

Other feature selection methods could be tested, as well as other Machine Learning classifiers and Deep Learning structures, it is a matter of trial and error to see what works best for this specific problem.

A larger dataset should be used to build a stronger Deep Learning method. Using different types of Data Augmentation strategies could be helpful to enlarge the dataset, if there are no other fundus images available to train and test the algorithms.

Future work should focus on the improvement of the methods described. Hopefully other eye diseases detection software can also benefit from this work, furthermore improving heath care quality.

References

[1] David Allen and Abhay Vasavada. "Cataract and surgery for cataract." In: *BMJ (Clinical research ed.)* 333.7559 (July 2006), pp. 128–32. ISSN: 1756-1833. DOI: 10.1136/bmj.333.7559.128. URL: http://www.ncbi.nlm.nih.gov/pubmed/16840470%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1502210.

[2] Renátó Besenczi, János Tóth, and András Hajdu. "A review on automatic analysis techniques for color fundus photographs". In: *Computational and Structural Biotechnology Journal* 14.2015 (2016), pp. 371–384. ISSN: 20010370. DOI: 10.1016/j.csbj.2016.10.001. URL: http://dx.doi.org/10.1016/j.csbj.2016.10.001.

[3] Akshay Jawaharlal Bhandari. *LOCS III Classification*. URL: https://www.researchgate.net/figure/LOCS-III-Classification%7B%5C_%7Dfig1%7B%5C_%7D276136427 (visited on 04/26/2019).

[4] Jason Brownlee. *A Gentle Introduction to Transfer Learning for Deep Learning*. 2017. URL: https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (visited on 08/09/2019).

[5] Michael Copeland. *The Difference Between AI, Machine Learning, and Deep Learning? — NVIDIA Blog*. 2016. URL: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/ (visited on 11/10/2018).

[6] Weiming Fan et al. "Principal component analysis based cataract grading and classification". In: *2015 17th International Conference on E-Health Networking, Application and Services, HealthCom 2015* (2015), pp. 459–462. DOI: 10.1109/HealthCom.2015.7454545.

[7] Chethan Kumar GN. *Machine Learning Types and Algorithms*. 2018. URL: https://towardsdatascience.com/machine-learning-types-and-algorithms-d8b79545a6ec (visited on 04/20/2019).

[8] Brett Grossfeld. *A simple way to understand machine learning vs deep learning — Zendesk Blog*. 2017. URL: https://www.zendesk.com/blog/machine-learning-and-deep-learning/ (visited on 11/12/2018).

[9] Liye Guo et al. "A computer-aided healthcare system for cataract classification and grading based on fundus image analysis". In: *Computers in Industry* 69 (2015), pp. 72–80. ISSN: 01663615. DOI: 10.1016/j.compind.2014.09.005. URL: http://dx.doi.org/10.1016/j.compind.2014.09.005.

[10] Shubham Gupta. *Decision Tree Tutorials & Notes — Machine Learning*. URL: https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/ (visited on 07/03/2019).

[11] V. Harini and V. Bhanumathi. "Automatic cataract classification system". In: *International Conference on Communication and Signal Processing, ICCSP 2016* (2016), pp. 815–819. DOI: 10.1109/ICCSP.2016.7754258.

[12] Mariam Hassib. "Mental task classification using single-electrode brain computer interfaces". In: January (2012). URL: http://elib.uni-stuttgart.de/opus/volltexte/2012/7982/%7B%5C%7D5Cnhttp://elib.uni-stuttgart.de/opus/volltexte/2012/7982/pdf/MSTR%7B%5C_%7D3393.pdf.

[13] Daniel Ho, Eric Liang, and Richard Liaw. *1000x Faster Data Augmentation*. 2019. URL: https://bair.berkeley.edu/blog/2019/06/07/data%7B%5C_%7Daug/ (visited on 06/11/2019).

[14] Zizhong Hu, Qinghuai Liu, and Yannis M. Paulus. "New Frontiers in Retinal Imaging". In: *International Journal of Ophthalmic Research* 2.3 (2016), pp. 148–158. ISSN: 2409-5680. DOI: 10.17554/j.issn.2409-5680.2016.02.48.

[15] FJ Huang and Y LeCun. "Large-scale learning with svm and convolutional netw for generic object recognition". In: *2006 IEEE Computer Society Conference on Computer ...* (2006). ISSN: 10636919. DOI: 10.1109/CVPR.2006.164. URL: https://scholar.google.fr/scholar?hl=en%7B%5C&%7Dq=Large-scale+Learning+with+SVM+and+Convolutional+Nets+for+Generic+Object+Categorization%7B%5C&%7DbtnG=%7B%5C&%7Das%7B%5C_%7Dsdt=1%7B%5C%%7D2C5%7B%5C&%7Das%7B%5C_%7Dsdtp=%7B%5C#%7D0.

[16] Sucheta Kolhe and Shanthi K Guru. "Remote Automated Cataract Detection System Based on Fundus Images". In: (2016), pp. 10334–10341. DOI: 10.15680/IJIRSET.2015.0506152.

[17] Alan Kozarsky MD. *How the Human Eye Sees*. 2017. URL: https://www.webmd.com/eye-health/amazing-human-eye (visited on 04/25/2019).

[18] Huiqi Li et al. "Image based grading of nuclear cataract by SVM regression". In: *Medical Imaging 2008: Computer-Aided Diagnosis* 6915 (2008), p. 691536. DOI: 10.1117/12.769975.

[19] Victor Hugo Masias et al. *Confusion matrix for a two–class problem*. 2016. URL: https://www.researchgate.net/figure/Confusion-matrix-for-a-two-class-problem-TP-is-the-number-of-correct-predictions-that-an%7B%5C_%7Dfig13%7B%5C_%7D292304919 (visited on 05/28/2019).

[20] MathWorks. *Create bag of decision trees*. 2019. URL: https://www.mathworks.com/help/stats/treebagger.html (visited on 06/18/2019).

[21] Calum McClelland. *The Difference Between Artificial Intelligence, Machine Learning, and Deep Learning*. 2017. URL: https://medium.com/iotforall/the-difference-between-artificial-intelligence-machine-learning-and-deep-learning-3aa67bff5991 (visited on 11/06/2018).

[22] MLMath.io. *Math behind Decision Tree Algorithm*. 2019. URL: https://medium.com/@ankitnitjsr13/math-behind-decision-tree-algorithm-2aa398561d6d (visited on 07/03/2019).

[23] Anuja Nagpal. *Decision Tree Ensembles - Bagging and Boosting*. 2017. URL: https://towardsdatascience.com/decision-tree-ensembles-bagging-and-boosting-266a8ba60fd9 (visited on 07/16/2019).

[24] C. P. Niya and T. V. Jayakumar. "Analysis of different automatic cataract detection and classification methods". In: *Souvenir of the 2015 IEEE International Advance Computing Conference, IACC 2015* (2015), pp. 696–700. DOI: 10.1109/IADCC.2015.7154796.

[25] NVIDIA. *NVIDIA Transfer Learning Toolkit*. URL: https://developer.nvidia.com/transfer-learning-toolkit.

[26] João Miguel Pires Dias, Carlos Manta Oliveira, and Luís A. Da Silva Cruz. "Retinal image quality assessment using generic image quality indicators". In: *Information Fusion* 19.1 (2014), pp. 73–90. ISSN: 15662535. DOI: 10.1016/j.inffus.2012.08.001.

[27] Zhiqiang Qiao et al. "Classification of Cataract Fundus Images". In: c (2017), pp. 2–6.

[28] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. 2018. URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[29] Liz Segre. *Eye anatomy: A closer look at the parts of the eye*. 2019. URL: https://www.allaboutvision.com/resources/anatomy.htm (visited on 04/25/2019).

[30] Chirag Sehra. *Decision Trees Explained Easily*. 2018. URL: https://medium.com/@chiragsehra42/decision-trees-explained-easily-28f23241248 (visited on 04/20/2019).

[31] Wenai Song et al. "Semi-Supervised Learning Based on Cataract Classification and Grading". In: *Proceedings - International Computer Software and Applications Conference* 2 (2016), pp. 641–646. ISSN: 07303157. DOI: 10.1109/COMPSAC.2016.227.

[32] Tina D. Turner M.D. *Are There Different Types of Cataracts? - VisionAware*. 2019. URL: https://www.visionaware.org/info/your-eye-condition/cataracts/different-types-of-cataracts/125 (visited on 04/16/2019).

[33] Venkatesan. *Artificial Intelligence vs. Machine Learning vs. Deep Learning - Data Science Central*. 2018. URL: https://www.datasciencecentral.com/profiles/blogs/artificial-intelligence-vs-machine-learning-vs-deep-learning (visited on 11/08/2018).

[34] Brian S. Boxer Wachler MD. *The Eyes (Human Anatomy)*. 2018. URL: https://www.webmd.com/eye-health/picture-of-the-eyes%7B%5C#%7D1 (visited on 04/25/2019).

[35] Gio Wiederhold, John McCarthy, and Ed Feigenbaum. *Professor Arthur Samuel — Stanford Computer Science*. URL: https://cs.stanford.edu/memoriam/professor-arthur-samuel (visited on 11/08/2018).

[36] World Health Organization. *Blindness and vision impairment*. Oct. 2018. URL: https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment (visited on 02/02/2019).

[37] Li Xiong, Huiqi Li, and Liang Xu. "An Approach to Evaluate Blurriness in Retinal Images with Vitreous Opacity for Cataract Diagnosis". In: *Journal of Healthcare Engineering* 2017 (2017), pp. 1–16. ISSN: 2040-2295. DOI: 10.1155/2017/5645498.

[38] Ji Jiang Yang et al. "Exploiting ensemble learning for automatic cataract detection and grading". In: *Computer Methods and Programs in Biomedicine* 124 (2016), pp. 45–57. ISSN: 18727565. DOI: 10.1016/j.cmpb.2015.10.007. URL: http://dx.doi.org/10.1016/j.cmpb.2015.10.007.

[39] Meimei Yang et al. "Classification of retinal image for automatic cataract detection". In: *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services, Healthcom 2013* Healthcom (2013), pp. 674–679. DOI: 10.1109/HealthCom.2013.6720761.

[40] Linglin Zhang et al. "Automatic cataract detection and grading using Deep Convolutional Neural Network". In: *Proceedings of the 2017 IEEE 14th Inter-*

national Conference on Networking, Sensing and Control, ICNSC 2017 (2017), pp. 60–65. DOI: 10.1109/ICNSC.2017.8000068.

[41] Jin Zheng et al. "Fundus image based cataract classification". In: *IST 2014 - 2014 IEEE International Conference on Imaging Systems and Techniques, Proceedings* (2014), pp. 90–94. DOI: 10.1109/IST.2014.6958452.