Image Classification of OCT Scans using Machine Learning

Tiago Oliveira Teixeira

Master Dissertation

FEUP Supervisor: Prof. Doctor Marco Paulo Lages Parente

FEUP Co-Supervisor: Guilherme Barbosa

FMUP Co-Supervisor: Prof. Doctor Manuel Falcão

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Master in Mechanical Engineering

July 2023

Resumo

De acordo com a Organização Mundial de Saúde (OMS), estima-se que, no mínimo, 2,2 mil milhões de pessoas em todo o mundo sofram de alguma forma de deficiência visual e que, pelo menos, mil milhões destes casos poderiam ter sido evitados. Existe uma necessidade crítica de desenvolver métodos eficazes para a deteção precoce e o diagnóstico exato de doenças relacionadas com a visão. Este trabalho centra-se na classificação de imagens de Tomografia de Coerência Óptica (OCT), uma técnica de imagiologia amplamente utilizada para captar doenças da retina.

O estudo proposto engloba uma análise abrangente de metodologias destinadas a automatizar a deteção de problemas da retina. O seu principal objetivo é avaliar e comparar a eficácia de métodos tradicionais e de redes neuronais profundas neste contexto. Para facilitar essa avaliação, é utilizado um conjunto de dados composto por 109.309 imagens de OCT da retina, abrangendo quatro condições médicas distintas: Neovascularização Coroidal (CNV), Edema Macular Diabético (DME), Drusen e retinas normais. Nos métodos tradicionais, o estudo emprega os métodos Histograma de Gradiente Orientado (HOG) e Padrão Binário Local (LBP). Em alternativa, a abordagem de aprendizagem profunda utiliza três modelos distintos de redes neurais convolucionais (CNN). São aplicadas técnicas de transferência de aprendizagem, utilizando modelos pré-treinados como o VGG16 e o ResNet50V2, ambos disponíveis através da interface Keras. Além disso, é introduzido um modelo proposto, caracterizado pelo seu número reduzido de parâmetros treináveis.

Para avaliar e comparar os resultados, são aplicados quatro métodos de pré-processamento diferentes às imagens originais e são implementadas técnicas de aumento de imagens nos modelos CNN. Os resultados experimentais demonstram que os métodos baseados em redes neuronais profundas superam as técnicas de extração de características tradicionais. O método HOG atinge uma precisão de teste de 73,20%, enquanto o método LBP atinge uma precisão de 54,60%. Em comparação, o ResNet50V2, o VGG16 e o modelo proposto atingem uma precisão de teste de 96,60%, 96,80% e 96,60%, respetivamente. A abordagem empregue neste estudo produziu consistentemente resultados comparáveis ou superiores a vários métodos existentes no estado da arte.

Os resultados experimentais sublinham a eficácia das arquiteturas CNN quando integradas na transferência de aprendizagem para a deteção automática de doenças em imagens da retina. Estas observações também servem para demonstrar a robustez dos modelos desenvolvidos, evitando a necessidade de treinar um modelo de raiz. Além disso, a inclusão de um filtro *Non-Local Means* no modelo proposto apresenta a oportunidade de obter resultados comparáveis e, simultaneamente, reduzir os custos computacionais. Como consequência, esta abordagem diminui efetivamente os tempos de treino resultando numa metodologia altamente eficiente.

Entre os modelos avaliados, o modelo VGG16 demonstrou o melhor desempenho quando combinado com o aumento de imagens e o pré-processamento 1. Atingiu uma exatidão de 96,80%, uma precisão de 96,88%, uma sensibilidade de 96,80% e uma pontuação Fbeta (beta=2) de 96,79%. Seguiu-se o modelo proposto com o pré-processamento 2, que alcançou uma exatidão de 96,60%, uma precisão de 96,79%, uma sensibilidade de 96,60% e uma pontuação Fbeta (beta=2) de 96,58%. Por último, o modelo ResNet que utiliza imagens originais da base de dados obteve uma exatidão de 96,60%, uma precisão de 96,77%, uma sensibilidade de 96,60% e uma pontuação Fbeta (beta=2) de 96,56%.

Image Classification of OCT Scans using Machine Learning

Abstract

According to the World Health Organization, it is estimated that a minimum of 2.2 billion individuals globally suffer from some form of vision impairment, and at least one billion of these cases could have been prevented or left untreated. There is a critical need to develop effective methods for early detection and accurate diagnosis of vision-related conditions. This work focuses on the classification of Optical Coherence Tomography (OCT) images, a widely used imaging technique for capturing retinal disorders.

The proposed study encompasses a comprehensive analysis of methodologies aimed at automating the detection of retinal problems. Its primary objective is to assess and compare the effectiveness of handcrafted and deep neural network methods in this context. To facilitate this evaluation, a dataset comprising 109,309 retina OCT images is utilized, encompassing four distinct medical conditions: Choroidal Neovascularization (CNV), Diabetic Macular Edema (DME), DRUSEN, and NORMAL. In handcrafted features, the study employs the Histogram of Oriented Gradient (HOG) and Local Binary Pattern (LBP) methods. Alternatively, the deep learning approach leverages three distinct convolutional neural network (CNN) models. Transfer learning techniques are applied, utilizing pre-trained models such as VGG16 and ResNet50V2, both available through the Keras framework. Additionally, a proposed model is introduced, characterized by its reduced number of trainable parameters.

To evaluate and compare the results, four different preprocessing methods are applied to the original images, and data augmentation techniques are implemented in the CNN models. The experimental results demonstrate that deep neural network-based methods outperform handcrafted feature extraction techniques. The HOG method achieves a test accuracy of 73.20%, while the LBP method achieves an accuracy of 54.60%. In comparison, ResNet50V2, VGG16, and the proposed model, achieve test accuracies of 96.60%, 96.80%, and 96.60% respectively. The approach employed in this study consistently yielded comparable or superior results when compared to several existing state-of-the-art methods.

The experimental findings underscore the efficacy of the proposed CNN architectures when integrated with transfer learning for the automated detection of diseases in retinal images. These experiments also serve to demonstrate the robustness of the developed models, avoiding the necessity of training a model from scratch. Moreover, the inclusion of a Non-Local Means filter in the proposed model presents the opportunity to achieve comparable results while concurrently reducing computational costs. As a consequence, this approach effectively decreases training times resulting in a highly efficient methodology.

Among the evaluated models, the VGG16 model demonstrated the highest performance when paired with data augmentation and preprocessing 1. It achieved an accuracy of 96.80%, precision of 96.88%, recall of 96.80%, and an Fbeta score (beta=2) of 96.79%. Following closely behind was the proposed model with preprocessing 2, which attained an accuracy of 96.60%, precision of 96.79%, recall of 96.60%, and an Fbeta score (beta=2) of 96.58%. Lastly, the ResNet model utilizing original images from the database achieved an accuracy of 96.60%, precision of 96.77%, recall of 96.60%, and an Fbeta score (beta=2) of 96.56%.

Keywords: Optical Coherence Tomography (OCT), Convolutional Neural Networks (CNN), Diabetic macular edema (DME), Choroidal Neovascularization (CNV), Drusen, HOG, LBP, Deep Learning, ResNet50V2, VGG16, Transfer Learning

Ackowledgements

Throughout the writing of this dissertation, I have received a lot of support and assistance, and I am grateful to all of those with whom I have had the pleasure to work during this project. This dissertation serves as the final step of my Master's degree at FEUP (Faculdade de Engenharia da Universidade do Porto).

I would first like to express my gratitude to my thesis supervisor, Prof. Doctor Marco Parente, for his availability, support, and the opportunity to work in the field of Machine Learning. It has been a challenging but rewarding experience to explore the realm of artificial intelligence in the field of medicine, particularly in ophthalmology.

I would like to extend my appreciation to Eng. Guilherme Barbosa for his invaluable help and guidance throughout the development of this topic. His assistance and solid support have been instrumental in my progress.

I am grateful to Prof. Doctor Manuel Falcão for his expertise in explaining how the retina works and providing invaluable insights into the daily life of ophthalmology in a hospital.

To my friends who have contributed during this journey, I am grateful for your daily thoughts and insights. Your intellect and humanitarianism have motivated me and made me a better person. I hope to continue growing with your friendship in the years to come.

Lastly, I would like to convey my deepest appreciation to my family, as they are the fundamental catalysts behind my achievements. I am profoundly grateful to them for embodying unwavering work ethics and providing constant encouragement throughout my journey. A mere expression of gratitude cannot suffice to acknowledge the profound impact they have had on my life. I am sincerely thankful to my family for affording me the opportunity to complete my studies in the city I have come to cherish, the "Antiga, Mui Nobre, Sempre Leal e Invicta Cidade do Porto." It is through their unwavering support that I have been able to pursue and achieve my academic goals.

Contents

Abbreviations

AI	Artificial Intelligence
AMD	Age-related macular degeneration
AUC	Area Under Curve
CNN	Convolutional Neural Network
CNV	Choroidal neovascularization
DL	Deep Learning
DME	Diabetic macular edema
DR	Diabetic retinopathy
FD-OCT	Fourier-domain OCT
GAN	Generative adversarial network
HOG	Histogram of Oriented Gradient
LBP	Local Binary Pattern
MH	Macular Hole
ML	Machine Learning
MRI	Magnetic resonance imaging
NLM	Non-Local Means
OCT	Optical Coherence Tomography
OCTA	Optical Coherence Tomography Angiography
OvO	One-vs-one
OvR	One-vs-rest
ReLU	Rectified Linear Units
ROC	Receiver Operating Characteristic
RBF	Radial Basis Function
RPE	Retinal Pigment Epithelium
SD-OCT	Spectral-Domain OCT
SIFT	Scale-invariant feature transform
SS-OCT	Swept-Source OCT
SVM	Support Vector Machine
TD-OCT	Time-Domain OCT
TNR	True negative rate
TPR	True positive rate

List of Figures

List of Tables

1 Introduction

The utilization of Artificial Intelligence (AI) in healthcare has been a longstanding aspiration dating back to ancient times, as evidenced by the desire to create intelligent machines [1]. Today, with advancements in technology, AI has emerged as a transformative force in healthcare, promising improved diagnostics, personalized treatments, and enhanced patient care.

1.1 Motivation

In the field of healthcare, the integration of AI has gained substantial prominence, leading to transformative advancements in various medical practices. Presently, Machine Learning (ML) algorithms have demonstrated exceptional precision, reducing the likelihood of errors and erroneous decision-making. However, achieving such performance levels was not an overnight occurrence; it has been a gradual evolutionary process.

This study aims to explore and compare the methodologies employed for automating the detection of OCT images, specifically focusing on traditional and deep learning approaches. While deep learning models currently dominate the landscape and constitute the primary focus of this research, investigating traditional methods provides valuable insights into the historical progression and techniques used prior to the emergence of deep learning. By comprehensively examining the evolution of these methods, this research aims to identify opportunities for enhancing disease detection, thereby potentially saving lives and significantly impacting healthcare outcomes.

This master's thesis was conducted as part of the fifth year of the Integrated Master's program in Mechanical Engineering, specializing in General Mechanics, at the Faculty of Engineering, University of Porto.

1.2 AI application in Medicine

The utilization of artificial intelligence in healthcare encompasses diverse domains and is prevalent across the field. Notably, medical imaging emerges as a prominent area where AI algorithms play a pivotal role in the interpretation of complex visual data, resulting in expedited and more precise diagnoses. Remarkably, these algorithms have demonstrated performance on par with or comparable to that of medical professionals [2].

AI plays a pivotal role in personalized medicine, leveraging the analysis of patient data to identify patterns and prognosticate treatment outcomes. It further enhances patient support through the utilization of virtual assistants, providing advanced assistance and guidance. Moreover, the implementation of AI-powered predictive analytics optimizes the efficiency and effectiveness of healthcare operations, contributing to remarkable advancements in healthcare enhancement. The diverse applications of AI in healthcare can be visually represented in Figure 1, showcasing the various areas where AI is employed.

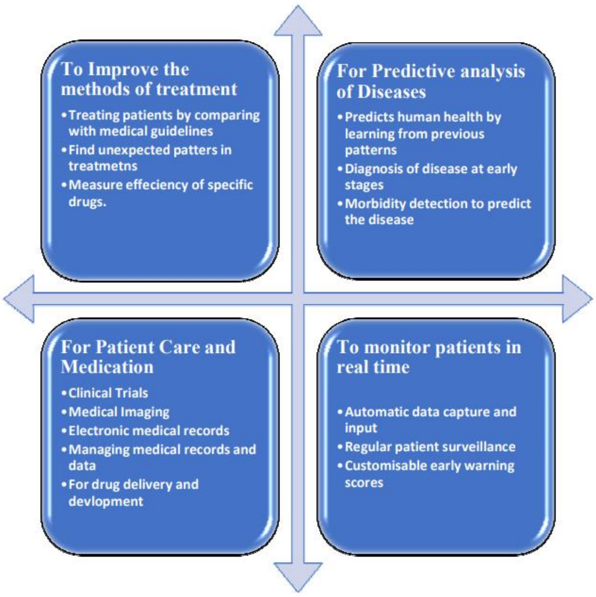

To Improve the methods of treatment
•Treating patients by comparing with medical guidelines
•Find unexpected patters in treatmetns
•Measure effeciency of specific drugs.
For Predictive analysis of Diseases
•Predicts human health by learning from previous patterns
•Diagnosis of disease at early stages
•Morbidity detection to predict the disease
For Patient Care and Medication
•Clinical Trials
•Medical Imaging
•Electronic medical records
•Managing medical records and data
•For drug delivery and devlopment
To monitor patients in real time
•Automatic data capture and input
•Regular patient surveillance
•Customisable early warning scores

Figure 1: Role of AI in Healthcare [3].

Advancements in machine learning algorithms have led to the replication of numerous medical tasks traditionally requiring human expertise. Deep learning applications, in particular, are increasingly trained using extensive annotated datasets, freeing medical specialists to focus on more productive tasks and projects. In fact, the potential of AI in medicine, including its applications in ophthalmology, is vast and holds tremendous promise for enhancing healthcare delivery in clinical practice [2].

The application of AI in ophthalmology is not a new phenomenon [4]. However, the importance of AI in this field has grown significantly, especially considering the escalating prevalence of vision impairments worldwide. It is estimated that a minimum of 2.2 billion individuals globally suffer from some form of vision impairment, and at least one billion of these cases could have been prevented or left untreated. OCT, a capable diagnostic technique, has emerged as a valuable tool for identifying various eye disorders, including leading causes of vision impairment such as age-related macular degeneration, glaucoma, and diabetic retinopathy [5].

1.3 OCT Exam in Retina Diseases

The breakthrough of OCT three decades ago revolutionized the field of ophthalmology, providing non-invasive imaging capabilities that have greatly aided in the treatment of eye diseases. OCT has become the standard imaging modality due to its high-resolution, cross-sectional imaging of the retina, retinal nerve fibre layer, and optic nerve head [6].

Over the years, OCT technology has undergone continuous development, evolving into a more powerful imaging tool. The inherent advantage with its non-invasive nature and fast acquisition time has led to the widespread installation of OCT in eye clinics worldwide, providing invaluable insights into the retinal architecture in various ocular diseases [7].

However, despite its significant contributions, the current ophthalmology department faces challenges in managing the overwhelming number of patients in need of eye care. To address the increasing patient load, an AI-powered workflow holds the potential to expedite patient care by assisting eye care professionals in making quicker and more informed decisions. By leveraging artificial intelligence, the time taken for patients to progress from initial eye scans to treatment can be significantly reduced, improving overall efficiency and patient outcomes.

1.4 Structure of this work

The present work is structured into eight distinct chapters, each serving a specific purpose in the overall research. The initial chapter focuses on the introduction, providing an in-depth contextualization and motivation for the study at hand. Chapters 2 and 3 constitute essential components of the research, as they are dedicated to the thorough exploration of fundamental knowledge concerning eye anatomy and OCT exams, respectively.

Chapter 4 of the thesis offers a comprehensive understanding of the fundamental principles underlying Machine Learning and Deep Learning. The chapter delves into the core concepts, methodologies, and techniques of the areas with direct application to OCT images enabling a comprehension of its underlying principles. The second part of the thesis transitions into the practical application of Machine Learning techniques in OCT image classification, employing both traditional and deep learning methods: chapter 5 serves as the starting point, presenting a thorough analysis of the state-of-the-art in the field. It encompasses an examination of previous works undertaken and highlights the utilization of publicly available databases.

Building upon this foundation, Chapter 6 provides a detailed account of the sequential steps and decision-making processes involved throughout the development procedure, culminating in the application of traditional methods and deep learning techniques. In penultimate chapter, the main results are presented along with detailed discussions that explore the complexities and implications of the applied methodologies. Finally, Chapter 8 comprises the study's conclusive elements, summarizing the findings, fulfilling the objectives outlined in the thesis, and suggesting future research directions.

The project aims to evaluate the potential of Machine Learning in ophthalmology, specifically OCT scans and enhance user trust in Artificial Intelligence models.

2 Eye Anatomy

The eye is a complex and fascinating organ that plays a crucial role in how humans perceive the outside world. This dissertation examines how OCT scans, a non-invasive imaging tool used for the diagnosis and monitoring of various eye disorders, may be classified using machine learning approaches.

Understanding its anatomy and function is essential to properly comprehend some of the pathologies that OCT scans can detect. Consequently, this chapter provides a comprehensive explanation of these concerns.

2.1 Anatomy and function of the Human Eye

It is estimated that the sense of sight accounts for 80% of all human perception [8]. As the principal organ for capturing, filtering, and transmitting light information to the brain for processing, the eyes play a crucial part in this procedure. These processes result in a visually perceived representation of our surrounding environment, demonstrating the crucial role that eyes plays in order to understand the world.

The human eye is a spherical structure that is located on the frontal surface of the skull. The dimensions of an adult eye are relatively constant, with a sagittal diameter of approximately 24 millimeters and a transverse diameter of 24.5 to 25 millimeters, weighing approximately 7.5 grams. The formation of the major eye structures take place during the fifth month of fetal development and by birth, the eyes are roughly two-thirds the size of an adult eye. The growth of the eye progressively slows from the second year until puberty [9].

The process of human visual recognition initiates as soon as light penetrates the pupil and is guided through the cornea and lens: the pupil regulates the amount of light entering the eye acting as an aperture that is regulated by the surrounding iris, cornea and lens are responsible for creating the optical image on the retina [10]. The retina then converts the image into electrical energy, which is then conveyed to the brain through intricate neural pathways, linking the eye to the visual cortex and other parts of the brain through the optic nerve. Figure 2 represents eye anatomy.

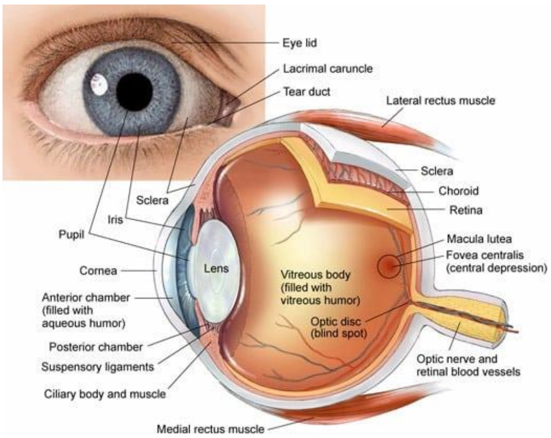


Figure 2: Sagittal and external human eye anatomy [11].

Layers of the eye

The human eyeball may be divided into three concentric tunics, as Figure 3 illustrates, each of which serves a different and distinctive function.

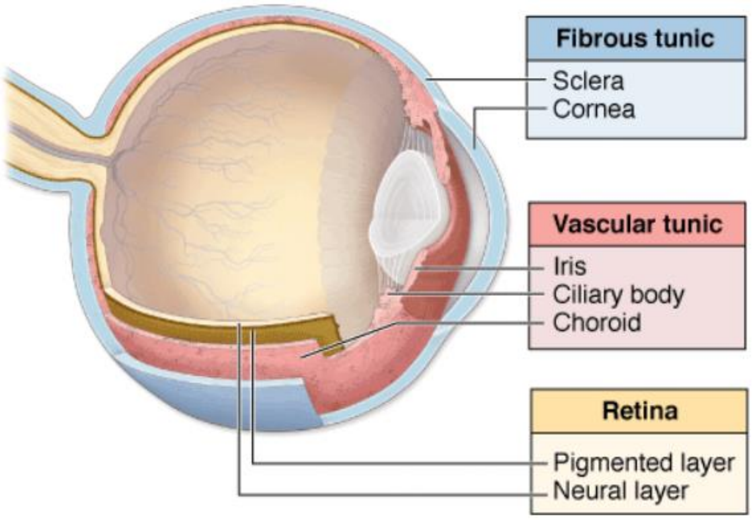


Figure 3: Layers of the eye [12].

Fibrous tunic: the outer layer of the eye is crucial for structural stability and keeping the shape of the eye. This layer is made up of the sclera (the white component of the eye) and the cornea (the layer at the front of the eye).
The anterior chamber, which is located in between cornea and the iris, is filled with a thin fluid known as aqueous humour. The constant creation of aqueous humour aids in the inflation of the eye globe, the regulation of intraocular pressure, and the provision of necessary nutrients to the avascular tissues within the eye, including the posterior cornea and lens [9].
Intraocular pressure, often known as the pressure within the eye, is an essential measure determined by the ease with which fluid drains from the eye.

Vascular tunic: the iris, choroid, and ciliary body are among the important tissues that make up the vascular tunic, or uvea, a vital part of the eye. This layer is distinguished by a thick concentration of blood vessels and pigment that help to hydrate the surrounding tissues and guarantee proper eye function [10].
The vitreous humour, a material that resembles gel and is present in the posterior chamber between the lens and the retina, is crucial to preserving the structural integrity of the eye. By applying pressure to the retina and choroid, the vitreous humour aids in maintaining the stability of these tissues and contributes to the overall shape of the eye [9].

Nervous tunic: it is made up of the retina, and its primary job is to receive light from an image and convert it into electrical impulses. There are an estimated 200 million photoreceptors in the retina, which include both rods and cones as well as a complex neuronal network that enables the processing and transfer of these electrical impulses from the optic nerve to the visual cortex in the brain for interpretation and perception [9].

Main functions of the elements of the eye

The subsequent section presents an overview of the primary functions of the principal elements of the eye. The retina of the eye undergoes a more in-depth analysis as it is the visible component on OCT scans [9].

Cornea: in addition to serving as a protective element for the eye, the cornea functions as a lens and serves as the primary optical structure responsible for retinal image formation. As the first refractive surface that light encounters, it plays a crucial role in the visual process.

Iris: coloured part of the eye that regulates the amount of light that enters it, adjusting the size of the pupil.

Sclera: is the outermost layer and gives structural stability and shape to the eye.

Pupil: a circular opening located at the centre of the iris, is capable of dilating. The variation in pupil size affects the quality of the images projected onto the retina, influencing the degree of diffraction and depth of focus, as well as the amount of light incident on the retina.

Lens: a transparent, elastic, but solid ellipsoid body that focuses the light on the retina, the third and innermost layer of tissue.

Optic Nerve: conveys to the brain all visual information.

Choroid: is a thin membrane that lies between retina and sclera, is irrigated by the blood vessels and it is maintained attached to the ciliary body. It is predominantly formed of a thick capillary plexus as well as small arteries and veins and feeds oxygen and nutrients to the majority of the back of the eye [10].

Retina: the retina is the fundamental sensory layer of the eye. The main function is to detect light and generate impulses that are transmitted to the brain via the optic nerve.

2.2 Retina Anatomy

Main layers

As mentioned before, the retina plays a critical role in vision due to its responsibility for translating light into a biochemical message which is translated into electrical impulses and transmitted to the brain.

The retina exhibits a laminar organization of ten main layers from outside (nearest the blood vessels enriched choroid) to inside (nearest the vitreous humour) and is approximately 0.5 mm thick. This segmentation is useful as it facilitates the identification of anomalous pathologies conditions that usually present a typical position within the retinal tissue [13].

All vertebrate retinas are formed of three layers of nerve cell bodies and two layers of synapses. The outer nuclear layer contains rods and cones; the inner nuclear layer contains bipolar, horizontal, and amacrine cells; and the ganglion cell layer comprises ganglion cells and displaced amacrine cells. Dividing these nerve cell layers are two neuropils (dense networks of interconnected neurons, axons and synapses where synaptic contacts occur) [13].

Figure 4 and Figure 5 shows cross-sectional retinal scans obtained using OCT scans with the identification of all the layers. The retinal layers from the vitreous to choroid are [14], [15]:

Internal limiting membrane: border dividing the vitreous body from the retina.

Nerve fibre layer: contains the axons of the ganglion cells (these nerve fibres are packed together and converge to the optic disc, where they leave the eye as the optic nerve).

Ganglion cell layer: cell bodies of the ganglion cells are situated here. Transmembrane receptors transform the chemical messages from bipolar cells and amacrine cells into the intracellular electrical [16].

Inner plexiform layer: consists of synaptic connections between the axons of bipolar cells and dendrites of ganglions cells.

Inner Nuclear Layer: bipolar nerve cells, the horizontal cells, and the amacrine cells are located here.

Outer Plexiform Layer: containing synaptic connections of photoreceptor cells (between the dendrites of the integration cells and the axons of the photoreceptor cells.)

Outer Nuclear Layer: where the cell bodies of the photoreceptors are located.

External Limiting Membrane: consists of densely packed connections between photoreceptors and supporting cells rather than a layer in the traditional sense.

Receptor layer: where photoreceptors (rods and cones) are located: There are 6.3-6.8 million cones and 110-125 million rods in each human [14]. The optic nerve conducts and further relays electrical impulses to the brain from light that has come into touch with the photoreceptors and, therefore, their light-sensitive photopigments.

Retinal Pigment Epithelium: separates the photoreceptor cells from the external retina of the choroid.

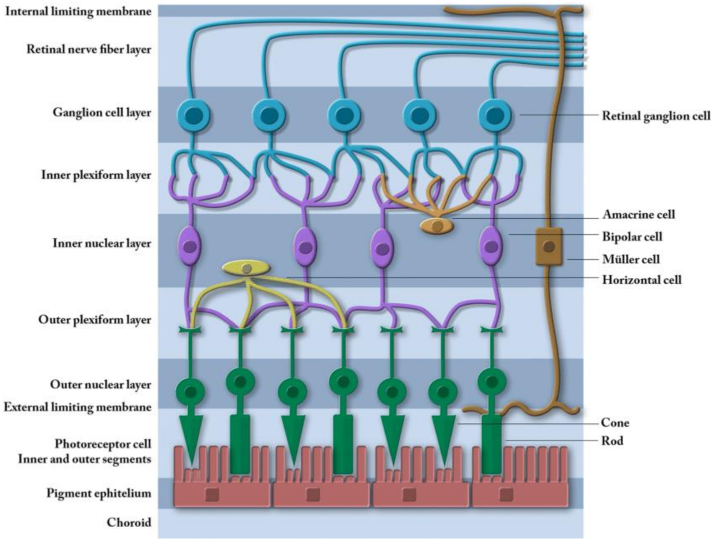


Figure 4: Retina layers [17].

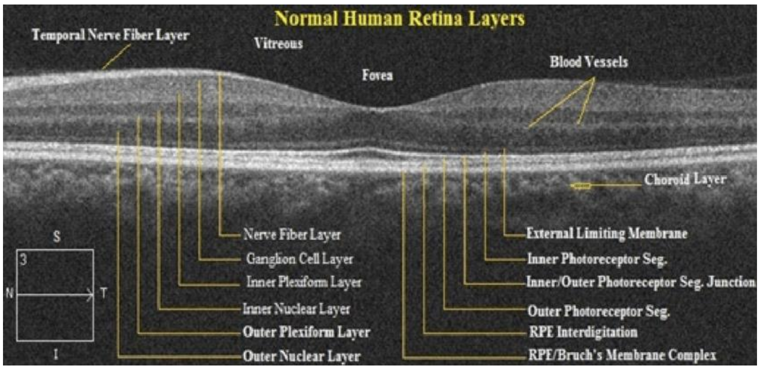


Figure 5: Labeled OCT Imaging [18].

Principal retinal components

As illustrated in Figure 6, the fovea and optic disc have substantial structural differences from the remainder of the human retina, which can be linked to their different functional and morphological properties.

.

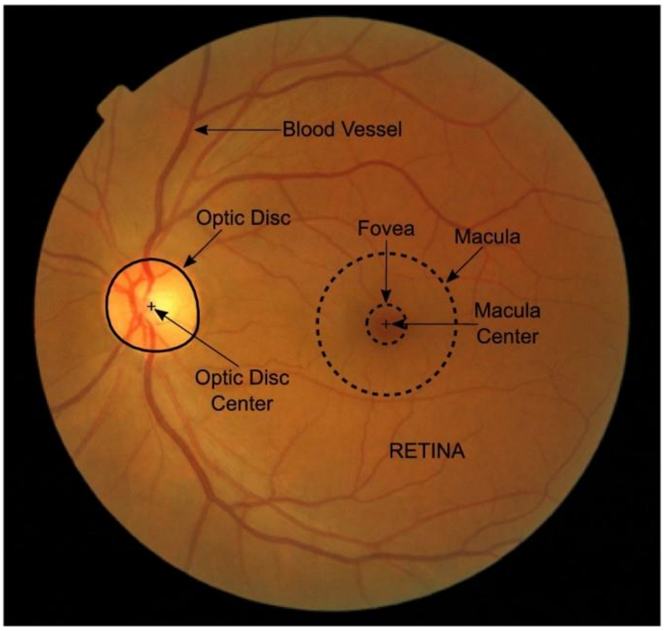


Figure 6: Main anatomical structures in a retinal image (left eye) [19].

Fovea: is the centre of the macula and the region of maximum visual acuity (ability to discern the smallest details of an object or letter at a specific distance). The fovea contains only cone photoreceptors and lacks rod photoreceptors. Moving away from the centre towards the edge of the retina, the density of cones gradually decreases. The inner layers are moved aside, enabling light to reach the photoreceptors unhindered. Additionally, there are more ganglion cells grouped in the foveal area than everywhere else [9], [10].

Optic disc: The optic disc is situated about 3 mm (15 degrees of visual angle) to the nasal side of the macula [20].
Since it lacks photoreceptors, it is responsible for the blind spot in the field of vision. The optic nerve comprises ganglion cell axons that go to the brain as well as incoming blood vessels that enter the retina to vascularize the retinal layers and neurons [13], [10].

Macula: is a specialised region for seeing fine detail and colour (possesses the largest amount of cone cells, which are what give humans their ability to see coloured.). In addition to the lens's function as a short wavelength filter, the macula also serves this purpose. Since the fovea is the most crucial component of the retina for human vision, protection against damage from bright light, especially UV radiation, is essential: once the sensitive cones in the fovea are damaged, visual system is lost [13].

Summary

This chapter provides a comprehensive analysis of the structure and function of the human eye, with a particular focus on the retina. An extensive knowledge of retinal anatomy is essential for the accurate interpretation of optical coherence tomography data. Through meticulous examination of the structural and functional properties of the retina, medical professionals are capable of identifying abnormalities that may be contributing to their patients' visual difficulties.

To summarize, the eye represents an exceptional organ that is vital for human vision. Its intricate structure and functions play an integral role in the interpretation of OCT scans and the diagnosis of ocular conditions. A complete understanding of the various components of the eye and their respective functions is indispensable in the promotion of eye health and the preservation of visual acuity throughout the lifespan.

3 OCT Exam Fundamentals

In ophthalmologic imaging, optical coherence tomography is a powerful and often-used technique that generates pictures of biological tissue in situ and in real time by detecting the light reflected from the structure being investigated [21].

OCT technology has become fundamental in ophthalmology by helping in early and differential diagnosis, decision, and therapeutic guidance. In fact, in the last years, OCT is one of the most frequently ordered diagnostic tests in ophthalmology - the rapid commercialization of equipment for producing authentic retinal tomograms in virtually real-time, without causing discomfort to the patient, and in a non-invasive way came soon after the practical demonstration of its feasibility (the first of its kind being introduced in 1996) [22]. Its basic operation is analogous to an optical ultrasound imaging, which creates high resolution cross-sectional images of the retina with a high spatial resolution [23], [24].

The examination of the components of the eyes using optical coherence tomography technology is highly advantageous, as these structures are predominantly or partially transparent, enabling the acquisition of reflected images via the passage of sufficient light – a fundamental condition for the successful use of OCT [21]. As illustrated in Figure 7, OCT bridges the gap between optical confocal microscopy and ultrasonic imaging. Not only have axial resolutions ranging from 1 to 15 µm, but also accomplish higher penetration depths of 1-3mm compared to confocal microscopy devices [25].

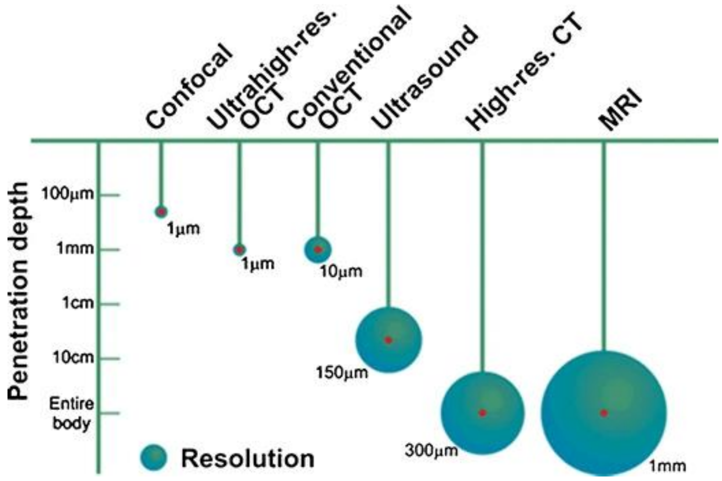


Figure 7: Resolution and penetration depth of some imaging methods in commercially available OCT [23].

Depending on the wavelength used, the resolution is in the range of 1 to 15 µm, which is at least twice as high as can be achieved with the best conventional methods such as magnetic resonance imaging (MRI) or high-resolution ultrasonography [21]. In Figure 8 is it possible to see different depth resolutions according to the light source wavelength range.

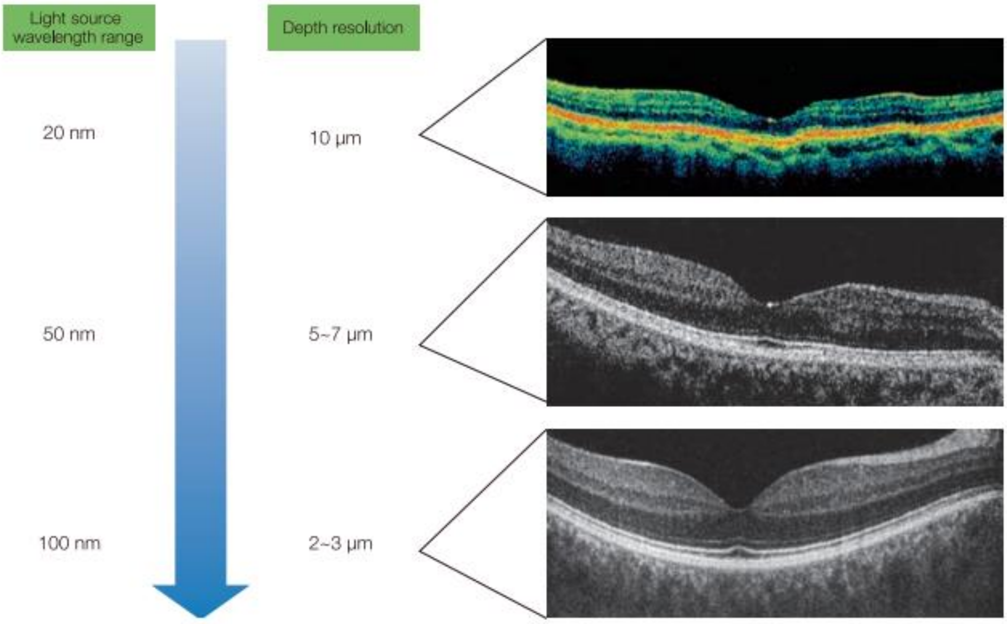


Figure 8: Depth resolution and light source wavelength range [26].

3.1 Physical Principles of Optical Coherence Tomography

Light interacts with a structure in three ways: transmission, absorption, and specular or diffuse reflection. When a structure is transparent, some of the light passes through unchanged, while some is absorbed and some is reflected in other directions: only a small portion of the light returns to the emission source, and the OCT examines this component. Due to its intrinsic properties, this reflected fraction only makes up a tiny amount of the incident light, ranging from one millionth to one billionth of the initial light intensity [21].

To perform an OCT, a laser-generated beam of light with an infrared SLD wavelength of around 840 nm must be transmitted into biological tissue, and then the reflected light must be analysed. As mentioned before, the physical principle of OCT is similar to ultrasound imaging, but instead of sound waves, it uses light. Ultrasound imaging takes into account the time that an emitted sound signal takes to echo back from the structure. As light travels faster than sound, the principle had to be adapted to measure the time for the incident light to be reflected at its source, which is approximately 30 femtoseconds (30 x 10^-15 sec) [27], [21].

Consequently, OCT uses low-coherence interferometry, a technology that uses the principle of interferometry to analyse this delay and is capable of deducing the thickness of eye components that the light has travelled through. This procedure, depicted in Figure 9 can be described as the following: light from a low-coherence source is directed into a Michelson interferometer, which is constructed using a 2x2 fibre-optic coupler and divides the incident optical power into reference and sample arms. Furthermore, a fibre-coupler divides the beam of light into two parts, one of which is projected onto a reference and the other onto the eye (sample).

The two waves resulting from this process are reflected, with the wave projected onto the reference returning as a single echo and the wave projected into the eye returning as numerous echoes depending on the structures it travelled through. The light is then recombined and directed forward into a detector. Moreover, an interferometer is used to compare these waves, measuring the coherence (the ability of light to interfere) between them. A full-depth reflectivity profile is produced for each sample point, creating an A-scan. When the focussed beam is moved directly over the sample, it produces a 2D cross-sectional scan known as a B-scan. A 3D OCT picture is created by adding the B-scan parts [22].

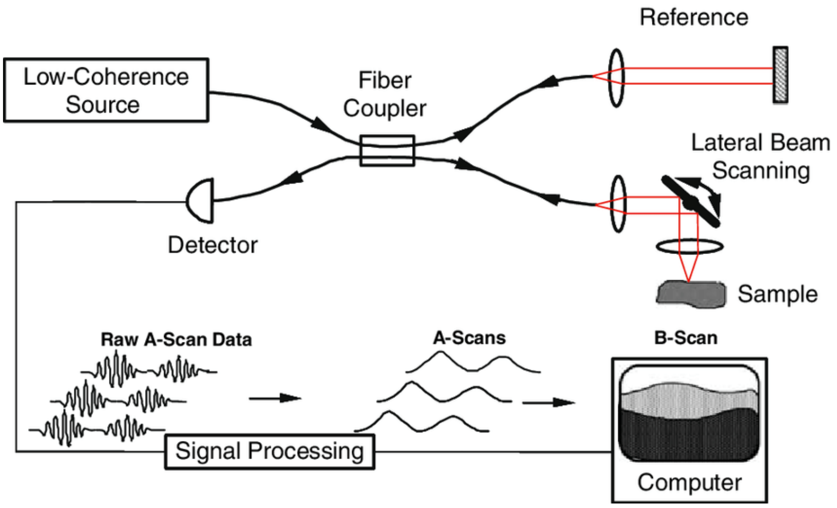


Figure 9: Schematic of a generic fibre-optic OCT system. Bold lines represent fibre optic paths, red lines represent free-space optical paths, and thin lines represent electronic signal paths [28].

OCT Techniques

OCT techniques can be classified into different types and generations. The first generation of OCT scans utilizes the time-domain technique (TD-OCT), whereas the second and third generations employ Fourier Domain techniques. Spectral Domain OCT (SD-OCT) is the second-generation technique, and Swept-Source OCT (SS-OCT) is the third-generation technique. The evolution of OCT technology represented in Figure 10 from the first generation to the third generation has brought remarkable improvements in terms of imaging speed, resolution, and depth [27].

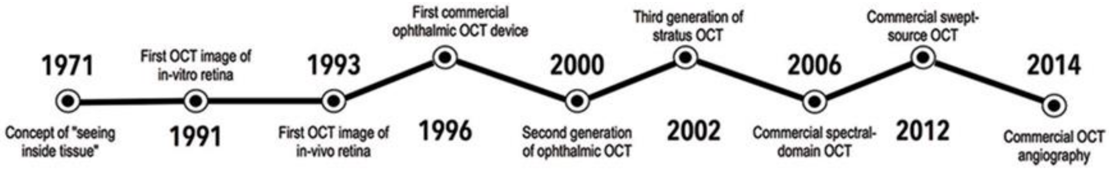

1971
Concept of "seeing inside tissue"
1991
First OCT image of in-vitro retina
1993
First OCT image of in-vivo retina
1996
First commercial ophthalmic OCT device
2000
Second generation of ophthalmic OCT
2002
Third generation of stratus OCT
2006
Commercial spectral-domain OCT
2012
Commercial swept-source OCT
2014
Commercial OCT angiography

Figure 10: Optical coherence tomography evolution [24].

TD- OCT, uses the physical movement of the mirror to scan various depths of layers of the retina. Figure 11 illustrates that, for each A-scan, the reference mirror is displaced between two endpoints that correspond to the extreme limits of the depth to be explored. The signal detected during this mirror displacement comprises a sequence of intensity variations, each corresponding to the reflection from a distinct anatomical structure. This method permits in-depth analysis of the reflected signal's intensity point-by-point and enables its representation using a greyscale. The signal coming from the detector is thus a time domain signal that is easily converted into the distance since the speed of movement of the mirror is known [22].

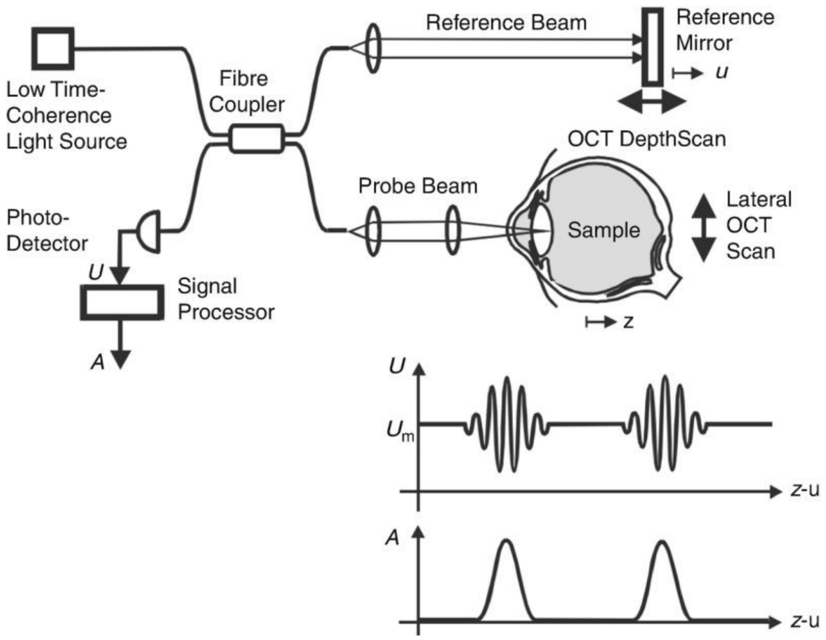


Figure 11: Working principle of TD-OCT: To record one depth profile of the sample (A-scan) the reference arm needs to be scanned and has to be repeated for each lateral scan position [29].

In FD-OCT, a spectrometer is employed to measure the interference spectrum produced by the reflected light, and the depth of the scatter is determined from the frequency or wavelength of the light. The reference mirror is stationary in FD-OCT, allowing for a more efficient measurement process [21].The use of a spectrometer in FD-OCT enables faster data acquisition, as all depths are measured simultaneously rather than sequentially. This results in improved image acquisition times and higher imaging speeds, with data acquisition speeds for typical Fourier-domain devices being approximately 45–100 times faster than those of TD-OCT. Furthermore, FD-OCT can capture 18,000–40,000 A-scans per second due to the simultaneous acquisition of spectral data [23].

Additionally, because FD-OCT provides better use of light, it can collect higher resolution scans with higher image quality and detect considerably weaker backscattered signals._The SD-OCT and swept-source OCT SS-OCT represent the second and third generation of OCT techniques, respectively, and both techniques use the Fourier transform for information extraction that allows the creation of A-scans [22].

In SD-OCT, low-coherence light source is used, emitting a continuous wave and a spectrometer to detect the interference of backscattered light from the eye. In contrast, SS-OCT uses a different technology to obtain the spectral information - a rapidly adjustable laser is used to acquire images even faster with higher sensitivity and deeper penetration. This light source emits light that rapidly changes frequency, sweeping through a range of wavelengths [22]. Comparing to SD-OCT, this technique results in a faster imaging speed and a deeper imaging depth [30]. Figure 12 shows SD-OCT and SS-OCT differences.

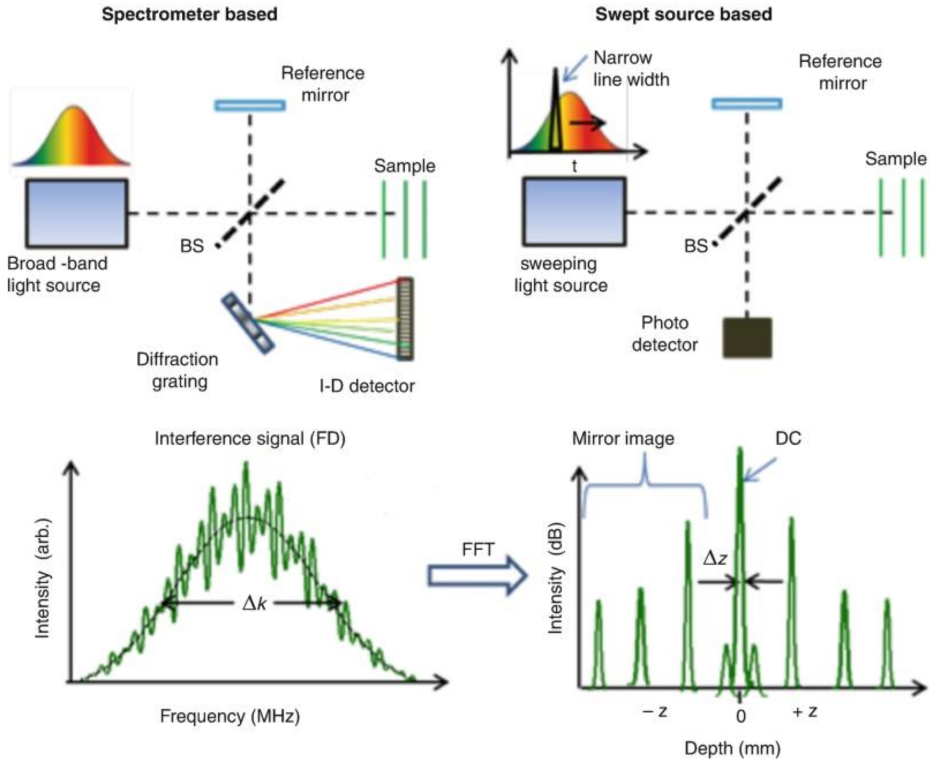


Figure 12: Optical setup of SD-OCT and SS-OCT [29].

3.2 OCT artifacts

OCT imaging presents challenges in both acquisition and interpretation. The images may contain artifacts, which are distortions that do not reflect the true structure being imaged as it is possible to see in figure below. Thus, caution is necessary when interpreting OCT data for clinical applications [31]. OCT artifacts can be categorized into three types: patient-related, operator-related, and software-related. While patient and operator-related artifacts can be controlled to some extent, software-related errors are inevitable and more common [24].

Patient-related artifacts arise from eye movements, which can be minimized by eye-tracking software and operator-related artifacts arise from decentered scans, out-of-registration images due to cuts, and degraded images due to poor focus. Software-related artifacts occur due to failed segmentation algorithms, leading to the misidentification of inner and outer retinal boundaries, and incomplete segmentation artifacts. Some artifacts are associated with specific diseases, such as segmentation failure, which commonly occurs in age-related macular degeneration [24].

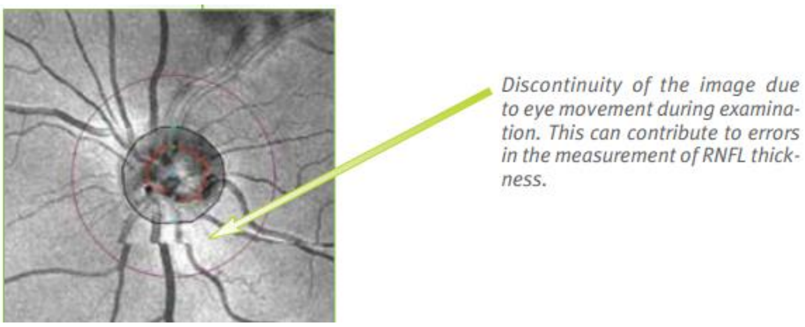

Discontinuity of the image due to eye movement during examination. This can contribute to errors in the measurement of RNFL thickness.

Figure 13: Sources of error with OCT [21].

The primary element affecting the quality of OCT pictures is speckle noise. OCT images have granular multiplicative speckle noise as a result of the coherent nature of the image capture technique, making it challenging to properly evaluate them [32]. Figure 14 shows an example.

There are several post-processing techniques that may be used to reduce OCT artifacts. Filtering, denoising, segmentation, and registration are a few examples of these techniques. While denoising techniques may be used to lessen speckle noise, a typical form of artefact in OCT imaging, filtering techniques can be used to minimize noise and improve visual contrast. For quantitative analysis, segmentation algorithms can be employed to locate and separate certain areas of interest, such retinal layers. Techniques for registration can be used to remove motion artefacts and enhance the alignment of photographs taken at various periods in time [33].

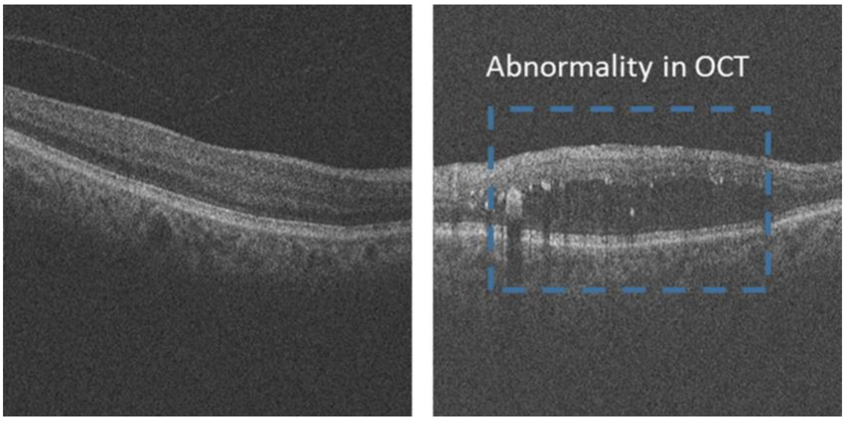


Figure 14: OCT image examples with speckle noise [33].

3.3 OCT visualization

With the advancement of software used in OCT, multiple techniques have emerged for obtaining images of the eye's anatomy. OCT cross-sectional and 3D rendering models provide valuable insights into the different anatomical features of the eye as it is possible to visualize in Figure 15. In current clinical practice, OCT is mainly used to observe structures in cross-section. The technological advancement of Fourier Domain OCT allows for rapid acquisition of volumetric data of ocular structures, which can be conveniently obtained in a clinical setting [22].

For patients with disorders that result in structural damage, including as glaucoma, macular holes, age-related macular degeneration, macular edema, and others, 3D OCT offers the potential to give measures that are more sensitive, specific and improve longitudinal follow-up [34]. Two common techniques of OCT are structural OCT and optical coherence tomography angiography (OCTA) – those techniques can be split based on the type of information they provide however for the purpose of this research, only OCT image analysis is analyzed.

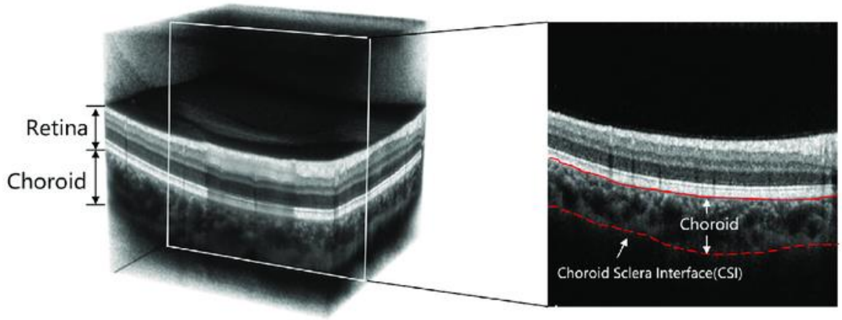


Figure 15: Examples of a 3D OCT volume and 2D OCT B-scan image [35].

To diagnose and detect distinct disorders, structural OCT collects data on the retina's cross-sectional structure, commonly referred to as B-Scans: The B-scans exhibit a granular speckle pattern, which is a characteristic of interferometric OCT measurement. The speckles are an inherent property of the imaging technique, and when two B-scans are acquired from the same retinal location, the speckle pattern in areas of static tissue remains relatively unchanged [36].

They can offer precise measurements of the volume and thickness of the retina and provide details about the severity and progression of the disease. Its high-resolution scans and measurements are helpful in the diagnosis and monitoring of a variety of eye disorders. For instance, structural OCT is used to quantify the retinal nerve fibre layer's thickness in glaucoma, which is a crucial sign of the disease's development.

3.4 Image analysis techniques

Due to the time complexity and subjective inaccuracies involved in evaluating ophthalmic images, there is significant interest in automating this process. OCT images can be subjected to a variety of automated analysis techniques, including segmentation of retinal layers, measurement of layer thickness and shape, noise reduction, curvature correction, and segmentation of blood vessels in 2D and 3D datasets. These automated processes may be divided into several OCT image analysis methodologies: feature segmentation, artifacts removal, image generation and classification diagnosis [37].

Among these methodologies, image segmentation is a very active area in the field of medical image analysis due to the difficulty in automatically localizing and extracting structures of interest. [38]. In addition, artifacts removal techniques on OCT images are very important as the images are contaminated with speckle noise and several works have been reported in literatures for OCT noise reduction [37], [38].

Another sort of application that could be employed in conjunction with OCT data is image generation. Synthetic image production has proven useful for machine learning model training and the development of large datasets for research purposes. In recent years, the generative adversarial network (GAN) has become the approach of choice for picture production in the medical imaging industry [39].

As the primary focus of this study is image classification analysis, a thorough analysis of the techniques employed in this field is presented, however, it is essential to keep in mind that the other three categories are as significant for OCT image interpretation. The category of OCT classification diagnosis can be categorized into two main groups: traditional methods and deep learning methods.

3.5 Advantages of OCT in Medical Diagnosis

When compared to a normal standard retinal scan, OCT provides higher-resolution images of the retina, which enables ophthalmologists to visualize the retinal layers with greater detail. OCT has developed into a popular diagnostic tool in the management of many retinal disorders due to its noninvasive nature, short response time, and reliability. This is true both for initial diagnosis and as a follow-up measure. Besides that, is also far more effective than conventional imaging techniques, which take longer and use more resources to produce and analyse results [24].

A minimum of 2.2 billion people worldwide suffer from some form of vision impairment and at least one billion of these instances may have been avoided or left untreated. OCT has become a capable diagnostic technique capable in identifying a wide range of eye disorders including several leading causes of vision impairment, such as age-related macular degeneration, glaucoma, and diabetic retinopathy [5]. The impact of vision impairment on the quality of life among adult populations is well documented: adults with vision impairment experience a range of challenges that affect their ability to participate in the workforce and maintain productivity, and have higher rates of depression and anxiety. Furthermore, these issues are particularly concerning among older adults, such as social isolation, difficulty walking, and a higher risk of falls and fractures [5].

In addition to its role in ophthalmic diagnosis, recent studies have unveiled the presence of biomarkers for various diseases within OCT images, including Parkinson's and Alzheimer's disease [40].

3.6 Retinal Diseases

By generating detailed cross-sectional images of the retina, OCT allows for the identification and characterization of different ocular conditions. Understanding the distinctive features and manifestations of these diseases in OCT scans is essential for accurate diagnosis, treatment planning, and monitoring of patients' ocular health. The visual representation of patients' vision with ocular diseases can be observed in Figure 16.

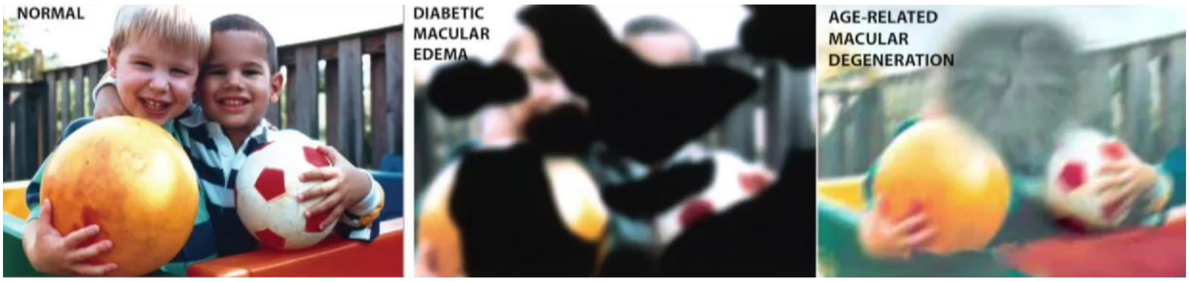


Figure 16:Comparison between a normal vision and disorders that cause visual loss [41].

In this dissertation, it is explored three common diseases in OCT images: drusen, choroidal neovascularization, which are both age-related macular degeneration diseases, diabetic macular edema as well as normal retina.

Age-related macular degeneration

Age-related macular degeneration (AMD) comprises approximately 8.7% of global blindness, rendering it a substantial contributor to visual impairment on a global scale. Particularly prevalent among individuals aged 60 years and above, AMD stands as the leading cause of blindness in developed nations. Furthermore, it was projected that the number of patients with the disease would reach approximately 196 million by the year 2020 with a subsequent rise to around 288 million by the year 2040 [42].

It is a degenerative ocular condition characterized by the thickening of the Retinal Pigment Epithelium (RPE) layer, resulting in the progressive impairment of central vision. This condition manifests in two primary forms: wet AMD, known as neovascular AMD or CNV, and dry AMD also referred as non-neovascular AMD or Drusen. The disease starts in the dry form and remains dry through the early and intermediate stages. Late stages can be either advanced dry or wet.

Drusen

Dry AMD is characterized by the presence of drusen, yellow deposits made up of lipids and proteins, can serve as an initial indicator of AMD. While a minimal amount of drusen does not typically result in vision loss, an increased accumulation of drusen poses a greater risk to the integrity of our sharp visual acuity.

Dry AMD represents the majority of diagnosed cases, accounting for approximately 80-90% of instances. Over time, the deposition of these drusen increases, leading to damage inflicted upon the RPE layer and subsequent loss of photoreceptor cells [43]. Figure 17 provides a visual representation of a drusen case as observed in OCT scan.

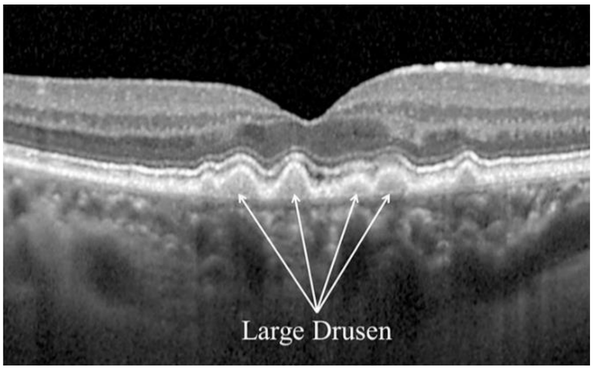


Figure 17: OCT Scan of Drusen: Visual Representation [44].

Choroidal Neovascularization (CNV)

Wet AMD, which accounts for 10-20% of cases, is characterized by the presence of abnormal blood vessel growth in the choroid layer of the eye. These new vessels have the tendency to leak fluid, resulting in a "wet" condition within the retina. Excessive fluid leakage can cause vision distortion or, in severe cases, complete vision loss. Immediate treatment is crucial to prevent further leakage and minimize the damaging impact on vision. Figure 18 provides a visual representation of a CNV case as observed in OCT scan.

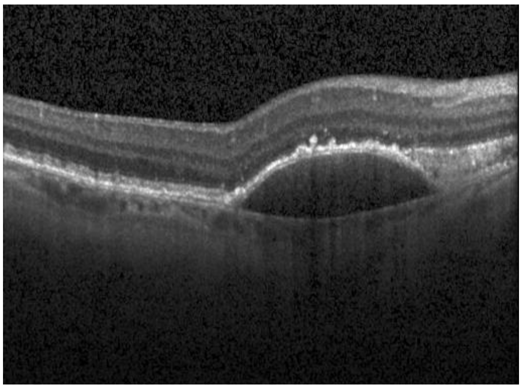


Figure 18: OCT Scan of CNV: Visual Representation [41].

Diabetic Macular Edema

Diabetic macular edema (DME) is a complication that arises from diabetic retinopathy, a condition affecting the blood vessels in the retina. Among individuals with diabetes, DME is the leading cause of vision loss [45]. Prolonged elevated blood sugar levels, can lead to damage in the small blood vessels throughout the body, including the eyes.

DME is a chronic condition characterized by the accumulation of fluid in the centre of the macula, a region of the retina responsible for clear and focused vision. This fluid buildup causes the macula to thicken, impairing the function of the cells responsible for sharp, straight-ahead vision needed for activities like reading and driving. As a consequence, blurred vision, which can be severe, is experienced [46]. Figure 19 provides a visual representation of a DME case as observed in OCT scan.

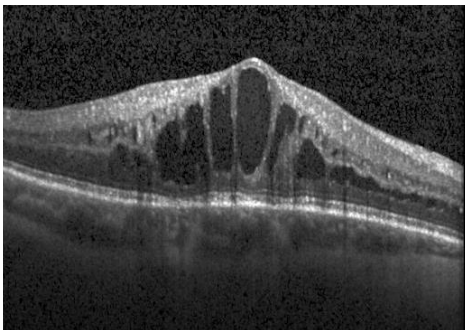


Figure 19: OCT Scan of DME: Visual Representation [41].

3.7 Importance of use ML in Detecting OCT Diseases

AI has the potential to revolutionize the field of medical imaging by using trained algorithms to analyze a large number of images and identify important structures for diagnosis. This can provide valuable support to medical staff and streamline the diagnostic process.

However, there are obstacles to overcome in applying machine learning to medical OCT images: one challenge is the limited availability of data for training the models -having a sufficient amount of labeled data is crucial for effective machine learning projects. Another obstacle is the lack of interpretability in automatic learning algorithms. Although these algorithms produce accurate results, the specific factors or features influencing their decisions are not explicitly explained. This can be problematic in medical settings where clinicians and patients may need to understand the reasoning behind a diagnosis or treatment recommendation. Interpretability is essential for building trust and confidence in machine learning predictions.

One common problem in most of hospitals is the time required to analyze a large number of daily scans. This can lead to significant delays in treatment, affecting all patients, including those with urgent needs. Another consequence is the heavy workload for medical professionals, resulting in a stressful environment and increased risk of misdiagnosis due to excessive workload.

Additionally, to this, machine learning algorithms can also aid in large-scale analysis of OCT data, contributing to research efforts and the discovery of new patterns, correlations, and insights. This advancement can enhance our understanding of various eye diseases, improve treatment strategies, and overall patient care – in fact, some studies have indicated that ML models demonstrate competitive performance and results comparable to those achieved by human experts possessing substantial clinical expertise in the interpretation of OCT images [41], [47].

3.8 Summary

Optically coherence tomography, often known as OCT, is a non-invasive imaging method that uses light to produce high-resolution images of the interior organs of the eye. To improve imaging speed, resolution, and sensitivity, several OCT methods have been developed, including time-domain, spectral-domain, and swept-source OCT.

OCT offers several benefits for diagnosing medical conditions, including its non-invasiveness, capacity to take extremely precise images of ocular tissue, and simplicity of usage. Furthermore, OCT scans can provide information about ocular pathologies in their early stages, enabling early intervention and treatment. OCT can produce extremely accurate scans of ocular tissue, but it is essential to be aware of any artefacts that could appear during image processing. Understanding these artefacts is essential for correctly interpreting OCT images. Medical professionals may diagnose patients correctly and successfully monitor their eye health if they have a firm understanding of the physical principles behind OCT, possible artefacts, and different visualization techniques.

Machine learning has the potential to revolutionize the analysis of OCT images. Trained ML algorithms can analyze a vast number of images and identify crucial structures for diagnosis, providing valuable support to medical professionals. However, there are challenges in applying ML to medical OCT images: limited availability of labeled data for training models is a significant obstacle, as is the lack of interpretability in automatic learning algorithms. Nonetheless, ML algorithms can also contribute to large-scale analysis of OCT data, facilitating research, and improving understanding, treatment strategies, and patient care in various eye diseases.

In conclusion, OCT is a useful technique for the detection and treatment of a variety of retinal pathologies. With the continual development of new methodologies and technologies, it is predicted that OCT use in clinical practice will increase over the next years [22]. Additionally, the integration of ML algorithms enhances OCT image analysis by efficiently processing images, extracting critical structures, and supporting medical professionals. Despite challenges, ML algorithms contribute to comprehensive OCT data analysis, advancing research, improving treatment strategies, and enhancing patient care in diverse eye diseases.

4 Machine Learning Fundamentals

This chapter provides a comprehensive review of the context and relevance of machine learning in image classification in ophthalmology, with a particular emphasis on Optical Coherence Tomography images. It explains both traditional methods and advanced deep learning algorithms used for diagnosing image classifications in OCT scans, based on previous studies. The mathematical principles behind these different models are also presented.

4.1 Machine learning: Concepts and Applications in Ophthalmology

The availability of large real-world datasets has proven critical in accelerating health data research, resulting in the creation of new discoveries and solutions. Many of these innovations use advanced statistical and computational methods, such as ML.

In fact, ophthalmology has discovered a wide range of uses for ML (automated diagnosis, disease prediction, prognostication, and image segmentation) and given its significant dependence on imaging is particularly suitable for ML because of the crucial role of imaging where OCT scans can be applied to identify diseases including glaucoma, age-related macular degeneration, and diabetic retinopathy [48].

Machine Learning is an Artificial Intelligence area which is characterized as a set of methods for automatically detecting patterns in data and then using the found patterns to predict future data or conduct various types of decision making under uncertainty [49]. Machine learning and artificial intelligence are terms that are frequently used interchangeably, yet AI need not just be dependent on learning-based methods: compared to ML, AI includes a significantly wider spectrum of computer science approaches that can consistently execute human cognitive functions [50], [51], [52]. This comparison is represented in Figure 20.

Many state-of-the-art algorithms in ophthalmology and OCT are based on deep learning. In fact, a wide range of methods, based on machine learning, and particularly based on deep learning, enable precise assessments of various eye diseases [50].

Deep Learning is an area of machine learning dealing with artificial neural networks, which is a class of algorithms inspired by the structure and function of the brain. Additionally, detailed information on this topic is provided [53].

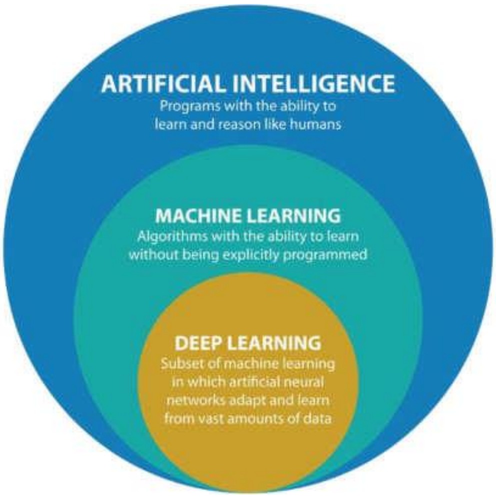

ARTIFICIAL INTELLIGENCE
Programs with the ability to learn and reason like humans
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Figure 20: Areas of study of Artificial Intelligence [54].

Learning models

As previously stated, a machine learning algorithm has the capability to learn from data. Tom M. Mitchell provided a technical definition that is frequently used for this discipline to explain if a computer program can be considered to learn: “A computer program is said to learn from experience E with respect to some class of task T and a performance measure P, if its performance at tasks in T, as measured by P, improves because of experience E” [55].

In other words, the task “T” can be seen as an activity that the computer program is designed to do; the performance “P”, is a way to quantify how well the computer program is doing on that task and experience “E” refers to the data or input that the program receives over time, which it can use to improve its performance on the task.

Machine Learning is usually divided into three main groups in relation to learning models:

Supervised Learning Algorithms: it is the most used ML form in practical applications [45]. This learning algorithm aims to identify a correlation between input and output variables using a training set comprising numerous examples of input x and output y pairs (a human intervention is frequently required since the output is frequently difficult to collect automatically). In order to provide precise predictions of outcomes for new data points based on the algorithm. It has to learn the key characteristics within each data point in the dataset to determine the answer and, as a result, when a new data point is introduced into it, the algorithm should be able to anticipate the outcome based on the relevant features collected from the dataset. Supervised learning can be classified into two categories: classification where the objective is classified something into a distinct set of classes or categories and regression that refers to the ability to predict values of a continuous variable [47], [52].

Unsupervised Learning Algorithms: the machine receives a collection of data without any anticipation of an output. The algorithm can identify patterns of similarity because of the enormous amount of data and uses techniques to group similar items [51].

Reinforcement Learning: algorithms for reinforcement learning interact with their environment, creating a feedback loop between the learning system and its experiences - External circumstances are always changing, and the machine's response must take these new circumstances into account [51], [1].

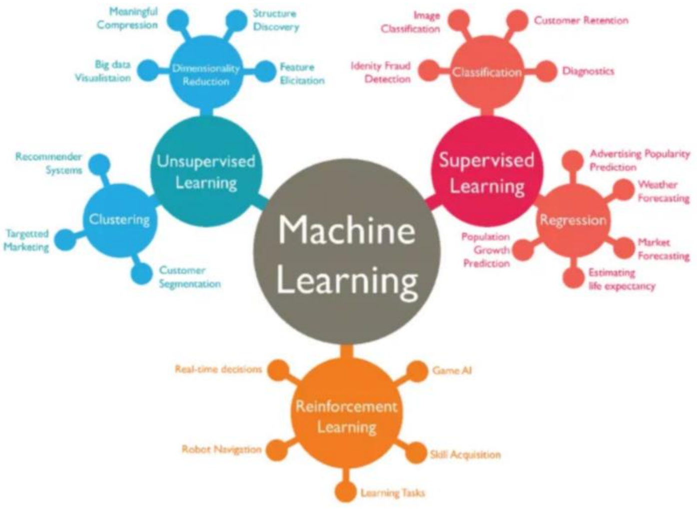


Figure 21: Three main categories of Machine Learning [56].

As it is possible to visualize in Figure 21, the Supervised Learning Classification approaches are more common in image classifications and for that reason are also the most common approach on OCT machine learning applications.

Machine Learning Algorithms for Classification

Classification is a supervised learning algorithm where a training set of correctly identified or labelled data is available. The model learned from training data to identify the category or class of the input feature is called classifier which can be a binary classifier or a multi-class classifier. In the literature, various machine learning algorithms, including Support Vector Machines, Random Forest, and Naïve Bayes, have been explored for OCT classification problems. However, this work exclusively focuses on the utilization of Support Vector Machines, thereby placing greater emphasis on this particular algorithm.

Support Vector Machine (SVM)

SVM is a powerful ML technique used for classification problems in high-dimensional feature spaces. The goal of SVM is to find a hyperplane, a separation boundary that ensures all samples are correctly classified on each side of it, that separates the data, with the maximum margin between the samples and the hyperplane. This hyperplane can be represented by a linear function, $w^T x + b$, where the SVM predicts that the positive class is present when $w^T x + b$ is positive. Likewise, it predicts that the negative class is present when $w^T x + b$ is negative [1].

A larger margin allows for greater flexibility in accommodating new data points and improves generalization. Support vectors, representative data points of each class, define the hyperplane's position and margin. Altering support vectors impacts the hyperplane, while other data points have no effect.

In practical scenarios, data may have noise or may not be linearly separable. To handle complex nonlinear boundaries, data points are mapped to a higher-dimensional space using a function $\phi(x)$. This mapping allows the data to become linearly separable. However, calculating the distance or similarity between each pair of data points in this augmented higher-dimensional space can be computationally demanding [51]. Figure 22 depict linear and nonlinear boundaries, respectively, while illustrating the representation of support vectors and margin.

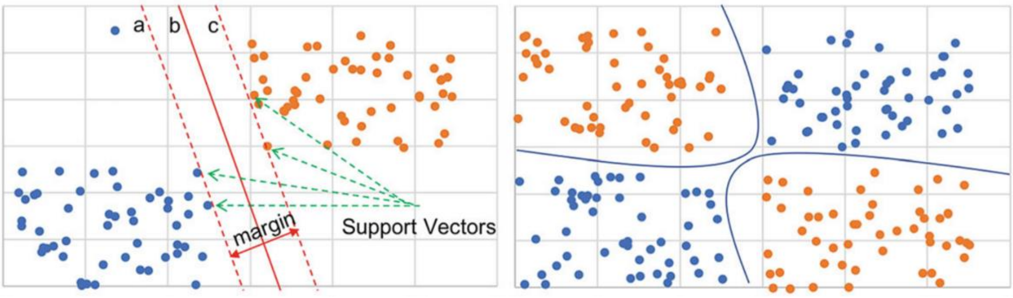


Figure 22: Linear and Non-linear Hyperplane [51].

To address this, the kernel trick is used. A kernel function, which represents the dot product of the mapping function, can be employed to implicitly compute the similarity or distance between data points in the high-dimensional feature space without explicitly mapping them. Kernels are similarity functions that possess dot product properties, allowing the substitution of a single function for a higher-dimensional feature vector. By choosing an appropriate kernel function, similarity computation becomes efficient.

Kernel function K can be represented by the equation below:

$$K(x, y) = \langle f(x), f(y) \rangle \tag{4.1}$$

Where, *x* and *y* are *n*-dimensional inputs; *f(x), f(y)* are functions mapping *n* dimension to *m* dimension space and $\langle a, b \rangle$ symbol represents a dot product of two vectors a and b [51]. Common choices for the kernel are linear kernel, polynomial kernel and RBF kernel.

Gamma, and C are another important hyperparameters in SVM that play a crucial role in the model's performance and behavior. The hyperparameter C determines the degree of emphasis placed on avoiding misclassification of training examples. A higher C value chooses a smaller margin hyperplane, resulting in a lower misclassification rate on the training data. Conversely, a lower C value allows for a larger margin, even if some training examples are misclassified. The gamma parameter controls the influence of a single training example: a higher gamma value focuses on points that are close to the hyperplane, leading to a more precise decision boundary. In contrast, a lower gamma value considers points that are farther away, resulting in a smoother decision boundary. Figure 23 and Figure 24 represents gamma and C hyperparameter respectively.

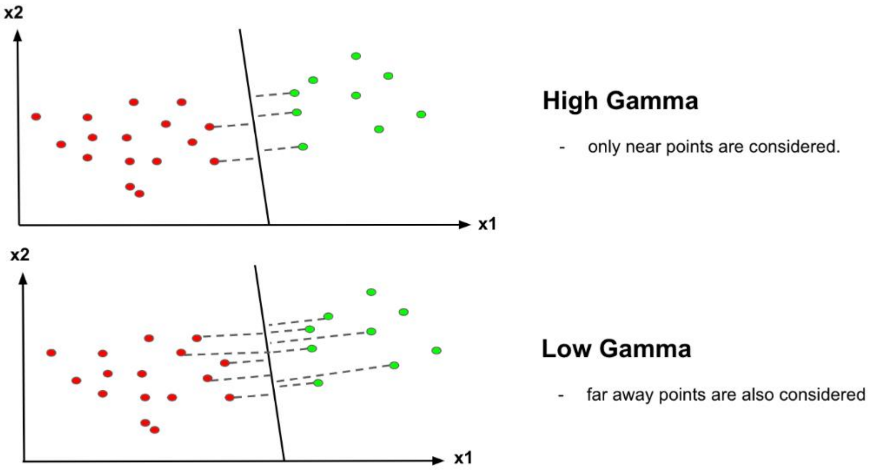

x2
x1
High Gamma
- only near points are considered.
x2
x1
Low Gamma
- far away points are also considered

Figure 23: Gamma Hyperparameter [57].

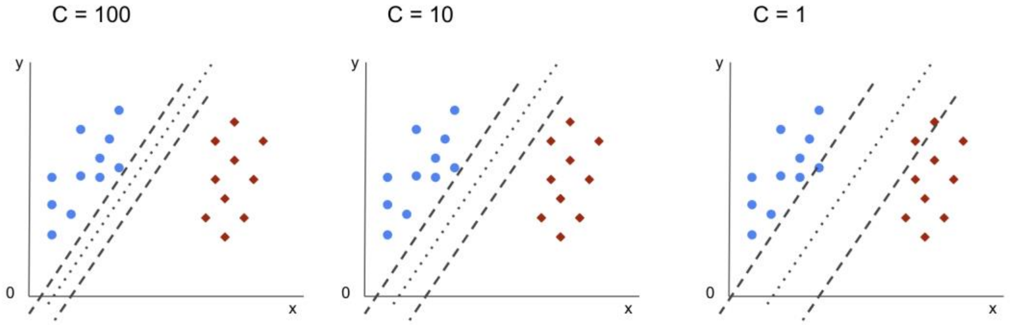


Figure 24: C hyperparameter [58].

Tuning these hyperparameters is crucial for achieving optimal SVM performance. Ultimately, the objective of SVM is to find the optimal separating hyperplane that maximizes the margin of the training data.

4.2 Handcrafted Algorithms for Feature Extraction

In ML, feature extraction is essential for distinguishing between images. Features are unique characteristics of an image that describe it and help differentiate it from others. In this work, it is used two handcrafted algorithms called Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) to extract features from images. These algorithms capture important information about the image's structure and texture.

Histogram of Oriented Gradient (HOG)

HOG is a feature descriptor that extracts useful information from an image by analyzing the changes in intensity or gradients across the image. It counts the occurrences of gradient orientation in localized regions of the image and provides the edge direction by extracting the gradient and orientation of the edges. The algorithm works by dividing the image into smaller regions, calculating gradients and orientations for each region, and creating a histogram for each region using the gradients and orientations of the pixel values. The gradient magnitude and orientation at each pixel provide information about the local image structure. The gradient orientations are then quantized into a fixed number of bins, typically ranging from 0 to 180 degrees.

This results in a simplified representation of the image that focuses on the shape or structure of an object, and provides an effective method for feature extraction [59].

The steps of feature extraction with HOG applied are:

Calculating Gradients: Figure 25 shows how horizontal and vertical gradients are calculated for each pixel where I (x,y) represents the intensity value in (x,y).

$$f_x(x,y) = I(x+1,y) - I(x-1,y) \tag{4.2}$$

$$f_y(x,y) = I(x,y+1) - I(x,y-1) \tag{4.3}$$

After getting gradient value, gradient orientation (θ) and magnitude (m) from equation 4.4 and 4.5 are calculated for every pixel of an image [59].

$$M(x,y) = \sqrt{f_x(x,y)^2 + \, f_y(x,y)^2} \tag{4.4}$$

$$\theta(x,y) = tan^{-1}\left(\frac{f_y(x,y)}{f_x(x,y)}\right) \tag{4.5}$$

Bin Orientation: The image is partitioned into regions of fixed size, called blocks, which are further divided into smaller regions called cells as represented in Figure 25. Each cell produces one histogram, and each block has a corresponding descriptor generated by combining the histograms from its constituent cells.
To ensure accuracy, all histograms have the same number of bins which represent the orientation of the gradient in degrees from 0 to 180.
The magnitude of the gradient vectors determines the contribution of each gradient to the histogram, which is divided between the two nearest bins. For example, if a gradient vector has an angle of 85 degrees, then 1/4th of its magnitude goes to the bin centered at 70 degrees, and 3/4ths of its magnitude to the bin centered at 90. The result is a compact and informative histogram representation of local image structure that is robust to variations in lighting and other image appearance changes [60].

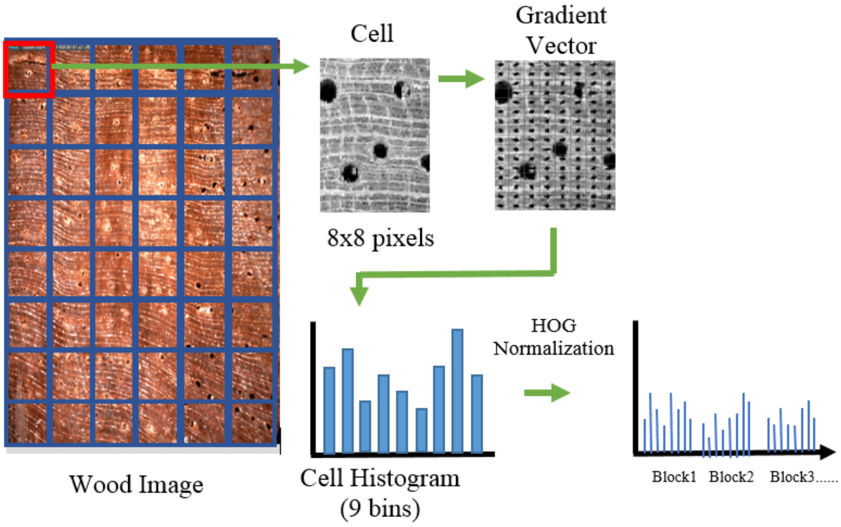


Figure 25: Example of HOG feature process [61].

Block Normalization: Normalization is an important step in HOG feature extraction, which ensures that the histogram values are invariant to global changes in illumination and contrast. The normalization is performed using L2 normalization represented by equation 4.6.

$$v_n = \frac{v_i}{\sqrt{||v^2||+\varepsilon^2}} \tag{4.6}$$

Local Binary Pattern (LBP)

LBP utilizes a binary code to represent local texture information in an image. The original LBP operator forms labels for each pixel by comparing its intensity value to that of its neighboring pixels using a 3x3 threshold. The resulting binary patterns, which can take 256 distinct values, can be used to form a histogram and represent the texture features of an image, as represented in Figure 26 [62].

The notation (P, R) indicates a neighborhood of P locations for sampling on a circle with a radius of R. These differences are recorded as binary patterns as follows:

$$LBP = \sum_{p=0}^{P-1} s\left(g_p - g_c\right)2^P, \qquad s(x) = \begin{cases} 1\, if\ x \geq 0 \\ 0\ otherwise \end{cases} \tag{4.7}$$

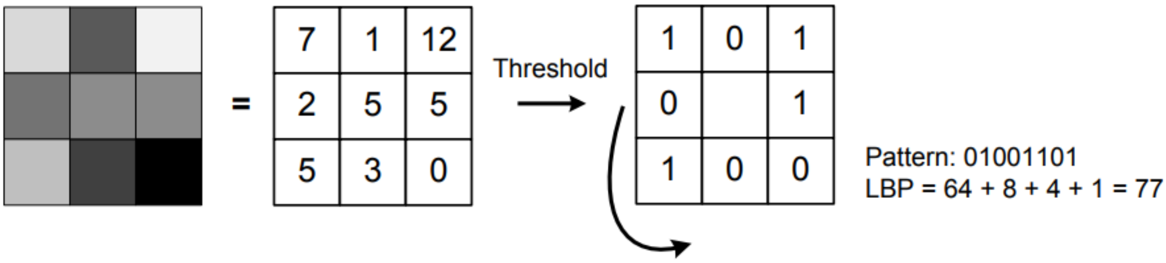


Figure 26: LBP feature [62].

The $LBP_{P,R}$ operator is capable of generating 2^P binary patterns from the P pixels in the neighbor set, resulting in 2^P output values. When the image is rotated, the gray values of the patterns move around a reference pixel. As a result, the $LBP_{P,R}$ value change for each rotation angle, except for patterns that contain only 0s or 1s, which remain constant [63].

In order to enhance the discriminative capacity of the LBP operator, researchers have developed extensions such as Uniform LBP [63]. A notable feature of Uniform LBP patterns is their ability to offer an additional degree of grayscale and rotation invariance. Uniform LBP patterns are categorized based on their uniformity: those with a maximum of two bitwise transitions from 0 to 1, or vice versa, are considered uniform, whereas those that exceed this threshold are classified as non-uniform. The results depicted in the figure below display how pixels with lower (or higher) intensity than the central pixel is represented in black (or white). Regions in the image where all the surrounding pixels are either black or white are characterized as flat, indicating an absence of any distinctive features. Conversely, clusters of continuous black or white pixels can be identified as "uniform" patterns, which can be interpreted as either corners or edges.

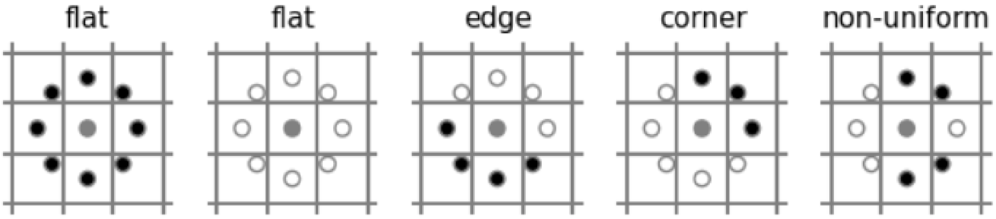


Figure 27: Different Pattern schemes [43].

The notion of "uniform" patterns is defined based on a measure of uniformity denoted by U(pattern), which quantifies the number of transitions between 0 and 1 present in the pattern. Patterns with a U value of 0 are composed entirely of either 0s or 1s, whereas those with a U value of 2 have precisely two transitions between 0 and 1. Patterns that possess a U value of at most 2 are identified as "uniform". This is represented in equation 4.8 by:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_0^{P-1} s\left(g_p - g_c\right) if\ U(LBP_{P,R}) \leq 2 \\ P + 1\ otherwise \end{cases} \tag{4.8}$$

The computation of a discrete occurrence histogram for "uniform" patterns has been demonstrated to be a highly effective texture feature for both an entire image and specific regions of an image. Through the computation of an occurrence histogram, a fusion of both structural and statistical approaches can be achieved: the LBP operator is capable of detecting microstructures such as edges, lines, spots, and flat areas, while the histogram serves to estimate the underlying distribution of these microstructures.

With an increase in the number of sampling points P, the resulting histogram dimensionality also increases accordingly. Based on this value, the number of patterns and uniform patterns can be derived, where P+1 uniform patterns exist for a given P. Consequently, the final dimensionality of the histogram becomes P+2, with an additional entry allocated for all non-uniform patterns. In relation to OCT images, Uniform LBP patterns and a generalized gray-scale and rotation invariant operator can aid in detecting microfeatures, which can improve the accuracy of layer segmentation and aid in the diagnosis of ocular diseases by recognizing these features irrespective of their orientation in the image.

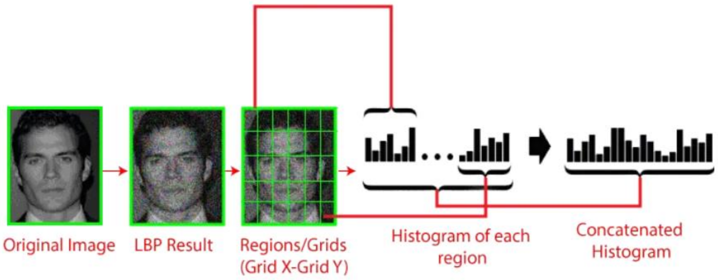


Figure 28: LBP steps [64].

The figure above illustrates the generation of an image histogram using local LBP, which extracts the texture of specific regions dividing the image in regions. Local LBP is chosen over global binary pattern in order to analyzing specific regions of interest within OCT images, such as identifying pathological changes in certain retinal layers. Lemaitre's work [65] showed that locally mapped features with an SVM classifier outperforms global mapped features.

4.3 Artificial Neural Networks (ANN)

Initially, neural networks were built with the goal of simulating the activities of the human brain, however the complex operations of the brain cannot be accurately replicated by the neural network, as is now generally recognized. At the moment, the most advanced computers still cannot match the human brain in terms of complexity in pattern recognition, however, over the years, there has been a great deal of interest in trying to understand all the internal mechanisms that the brain uses to perform functions such as pattern recognition [46], [66]. When the brain is examined, several levels of processing become visible and it is believed that each level accumulates characteristics or representations at more abstract levels. In the traditional concept of the visual cortex, for instance, the brain first detects edges, then segments, then surfaces, and ultimately objects [49].

In Figure 29 it is possible to visualize how brains process information: Each neuron consists of a cell body that contains a cell nucleus, dendrites and a single long fibre called the axon. At synapses, a neuron connects with 10 to 100000 additional neurons by an electrochemical reaction [52].

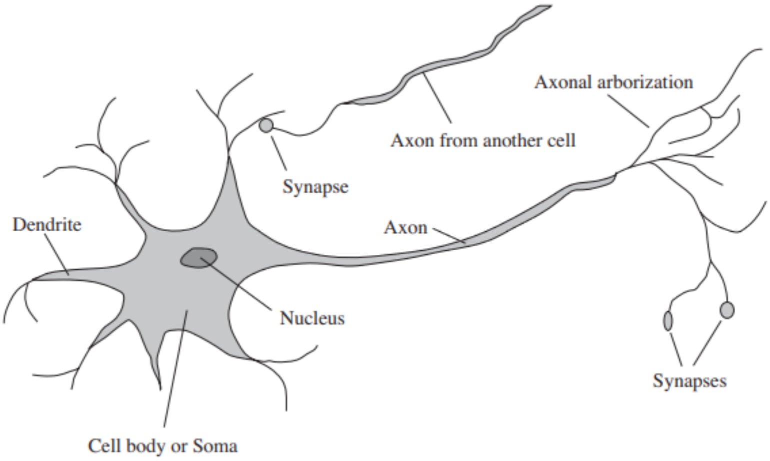


Figure 29: Brain Neurons Anatomy [52].

An ANN is a type of network where each neuron is directly connected to every other neuron in the adjacent layers. The ANN can be divided into three types of artificial neurons as represented in Figure 30: input layers is where initial data of the neural network is processed; hidden layers represent the intermediate layers where the information is processed and weights are applied to the inputs, directing them through an activation function as the output; output layer is the final layer in the neuronal network where the desired predictions are obtained.

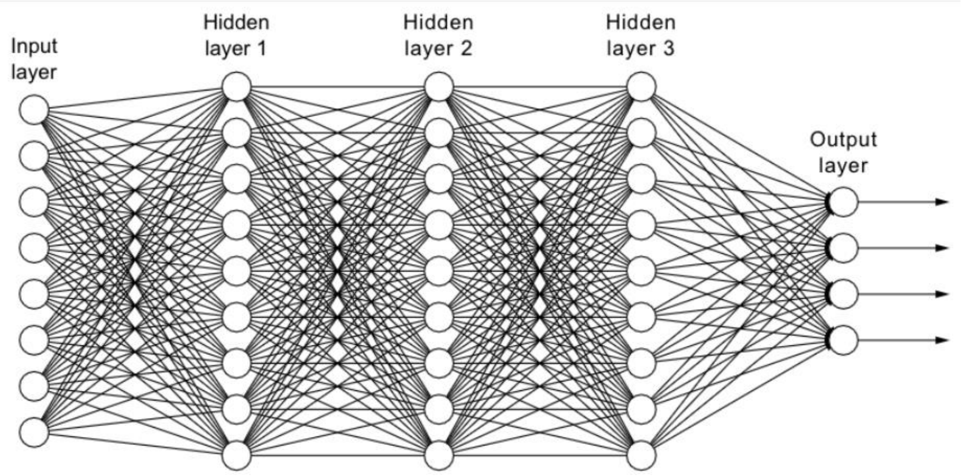


Figure 30: Fully connected artificial neural network with three hidden layers [67].

Artificial neurons are mathematical functions designed as models of organic neurons and are also the basic components of an ANN. Figure 31 shows a simplified model of an artificial neuron, with x_N standing for the input layers that provide values to the node's output side. The output of each neuron is given by $f\left(\sum_{i=1}^{N}(w_j \times x_N + b)\right)$ where f represents the activation function which receives both the weighted sum of inputs, $\sum_{i=1}^{N} w_j \times x_N$, and the bias, b.

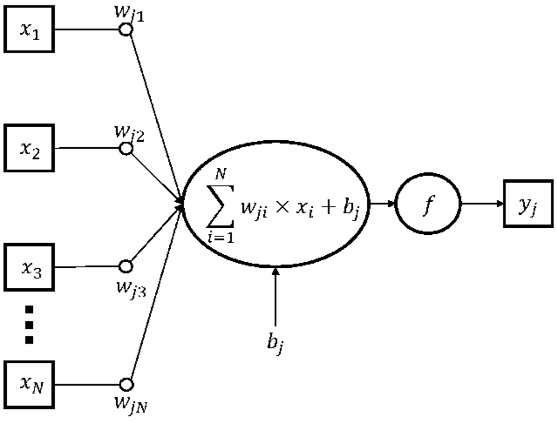


Figure 31: Simplified representation of an artificial neuron [68].

Several topologies for neural networks may be created depending on the number of layers, nodes in each layer, activation functions, and loss functions, among other factors. ANNs may be developed to approximate any function by changing the weights of neuron interconnections. The size, complexity, and design of the network are influenced by hyperparameters such as the number of layers and nodes in each layer. While designing a neural network, a variety of hyperparameter combinations are examined, and the most effective ones are chosen [51].

Activation Functions

Activation functions play a crucial role in neural networks as they introduce nonlinearity and enable the modeling of complex relationships between inputs and outputs. These functions are applied within each layer of the network and are responsible for transforming the data of the nodes before passing it to the next layer. By utilizing activation functions, neural networks are capable of learning intricate non-linear patterns in the data, thereby enhancing their ability to handle complex functions [51].

In the context of this thesis, two commonly used activation functions are highlighted:

Rectified Linear Units (ReLU): ReLU is a widely employed activation function that eliminates negative inputs. It returns a value of 0 for negative inputs and retains the positive input as it is. This activation function is extensively used in convolutional neural networks. By removing negative values, ReLU avoids saturation issues and effectively handles negative gradients when the threshold is set to zero [69]. Mathematically, ReLU can be expressed as:

$$\mathrm{f}(x) = max(0, x) \tag{4.9}$$

Softmax: Softmax is another frequently used activation function, particularly in scenarios where a neural network has multiple classes. It calculates the probability for each class relative to all the classes and is typically employed in the last layer of the network for making predictions. Softmax ensures that the predicted probabilities sum up to 1, enabling the network to generate class probabilities for multi-class classification tasks [70]. Mathematically, Softmax can be expressed as, where X is the input vector and w_i is the predicted probability of $y = j$:

$$P(y = j|X) = \frac{e^{X^T} w_j}{\sum_{k=1}^{K} e^{X^T} w_k} \tag{4.10}$$

Loss Functions

The loss function is commonly referred to as the objective function, error function, or cost function. It serves to quantify the discrepancy between the output generated by the algorithm and the desired target value. During the training process, minimizing the loss helps the model learn to predict the correct labels with greater confidence.

In the case of multiclass classification problems, the Cross-Entropy loss function is often used. In the last layer of the neural network, the softmax activation function is applied to convert data into probability values. These probabilities, derived from the softmax function, are then utilized by the loss function for evaluation. The purpose of this evaluation is to compare the network's predictions with the true labels or targets and calculate the error. Subsequently, the error is propagated back through the network using the backpropagation algorithm to update the model's parameters, leading to improved performance.

Mathematically, the Categorical Cross-Entropy loss function is defined as the negative sum of the element-wise multiplication of the true distribution t and the logarithm of the predicted distribution p associated with class i. The true distribution t is a one-hot vector, containing a 1 at the appropriate index and zeros elsewhere. This loss function is expressed as follows:

$$L_{CE}(t,p) = - \sum_{i=1}^{n} t_i \log(p_i), \text{for } n \text{ classes} \tag{4.11}$$

Learning

Training a neural network involves finding the appropriate weights for the network, which is accomplished through several steps. Initially, random weights are assigned to the network. The training algorithm then iterates through multiple cycles, known as epochs, until a stopping criterion is met. During each epoch, a forward pass is performed, where the weights and activation function of each neuron are applied as the input propagates through the network from the input layer to the output layer. This involves calculating the output of the network with the current weights and comparing it to the expected output, resulting in a loss value [51].

The choice of loss function may vary depending on the specific application. Different loss functions are used to quantify the dissimilarity between the predicted output and the expected output.

Following the forward pass, a backward phase is conducted. This phase aims to adjust the weights of the network to minimize the error. This is achieved through the process of gradient descent, which involves computing the partial derivatives of the overall loss or cost function with respect to each weight. The gradients indicate the direction and magnitude of adjustment required for each weight to reduce the error. Figure 32 represents a simplified model of the direction of gradient.

When employing the Adam optimizer to train the proposed model, the weights are updated using an adaptive learning rate. Unlike traditional approaches that rely on the current gradient, Adam considers the average of prior gradients, providing a more efficient optimizer that facilitates faster convergence for deep networks [51]. Adam is an efficient optimizer that achieves faster convergence for deep networks [71].

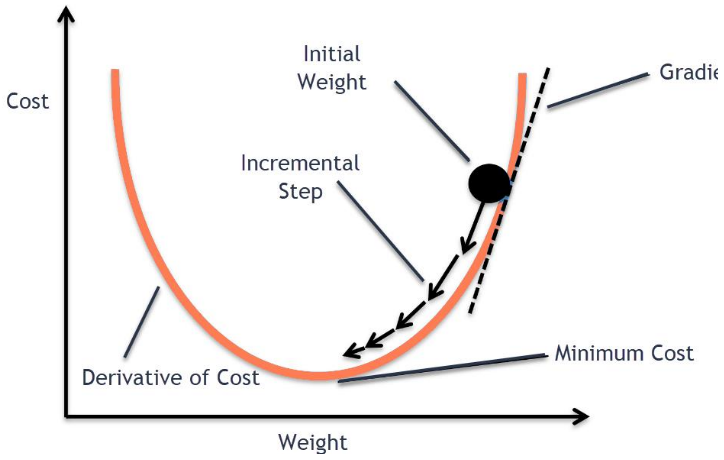


Figure 32: Gradient descent [72].

4.4 Deep learning training overview

Generally, a higher number of datasets used for training an algorithm tends to result in a better final model, however, it is important to note that without the appropriate set of features, an increased number of datasets may not yield any additional accuracy [51].

The primary challenge in ML is demonstrate proficiency on new and previously unseen inputs- this capability is known as generalization. Typically, a specific training set is used during the training of a machine learning model, which allows for the computation of the training error. In order to measure the generalization error of a model, its performance is often evaluated on a separate test set that was acquired independently from the training set.

The effectiveness of a ML algorithm is determined by two factors: first, the capacity to decrease training error, and second, the ability to minimize the disparity between training error and test error. Most machine learning algorithms contain various options that we may use to regulate the learning algorithm's performance; these settings are known as hyperparameters [1].

Usually, the majority of data is often allocated for training. It is common practice to divide a dataset into three distinct subsets for training and testing. Typically, around 80% of the data is allocated for training and 20% for testing. The training data is split into two subsets: one subset is used to learn the model parameters, while the validation set, is used at the end of a training epoch (a single pass through the training data set) to evaluate generalizability and update hyperparameters accordingly [1].

Finally, the test set is used to evaluate the model's performance on unseen data after the training is completed. This three-way split of the data helps to prevent overfitting and provides a more reliable estimate of the model's performance on new data. The data used in training and testing should be mutually exclusive in order for algorithm performance evaluation to be fair. For OCT, this means that test and training data should come from distinct patients [50]. Figure 33 represents the workflow of Machine Learning pipeline.

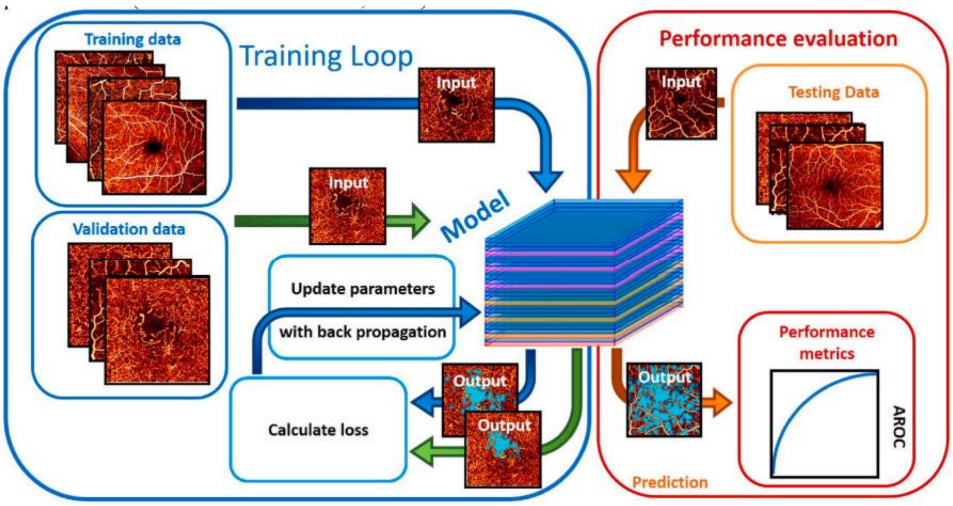


Figure 33: Training and evaluation in machine learning [50].

Underfitting and overfitting are two of the most difficult problems in machine learning. Overfitting is also known as the bias problem, and underfitting is known as the variance problem. When the model fails to attain a low error value on the training set, this is referred to as underfitting. Overfitting, on the other hand, happens when there is a large difference between the error values of the training and test sets [51], [1].

The performance of an estimator is influenced by a fundamental trade-off between bias and variance. When an estimator exhibits high bias and low variance, it is unable to adequately adapt to the data points in a given sample set, resulting in a considerable error. Conversely, an estimator with high variance and low bias tends to excessively adapt to all data points in a sample set, which may not accurately represent the entire dataset, leading to poor generalization and ultimately higher error. It is crucial to strike a balance between bias and variance to achieve optimal estimator performance, where the estimator appropriately adapts to the data while maintaining the ability to generalize to unseen samples from the true dataset [73].

In the context of learning, the objective is to minimize both the training error and the discrepancy between the training and test errors. This entails optimizing the balance between bias and variance, aiming to find the model complexity that achieves the optimal bias-variance trade-off. This trade-off can be visualized in Figure 34, where the optimal point is reached.

Suppose $f(x)$ as an approximation for the true function $Y = f(x) + \varepsilon$, with $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2$. The expected prediction error at a specific point x_0, also known as test or generalization error, can be mathematically expressed as [74]:

$$\begin{aligned} Err_x &= E\left(\left(Y - f(x)\right)^2 \middle| x = x_0\right) \\ &= Irreductible\ Noise + bias^2 + variance \\ &= \sigma^2 + [E((f(x)) - f^*(x_0)]^2 + E[(f(x_0)) - E(f^*(x_0))]^2 \end{aligned} \tag{4.12}$$

The bias of an estimator is defined as the expected difference between the estimates produced by the estimator and the true values present in the underlying data. It quantifies the systematic deviation of the estimator from the true values on average.

On the other hand, the variance of an estimator is the expected value of the squared difference between the estimate obtained from a specific model and the expected value of the estimate across all possible models in the estimator. It measures the variability or spread of the estimates generated by the estimator [73].

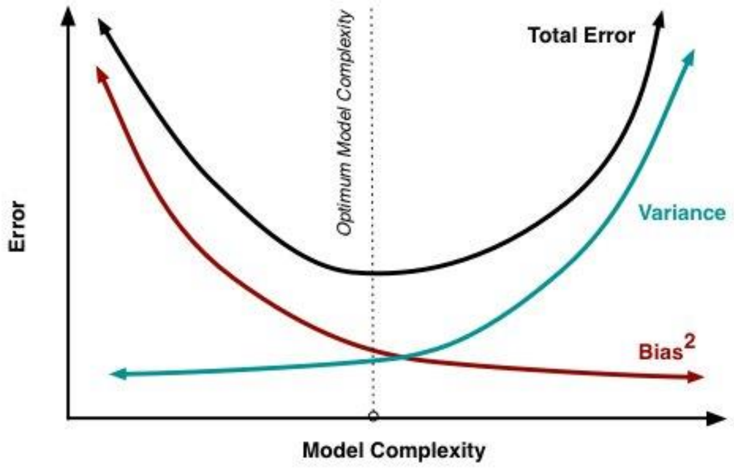


Figure 34: Bias and variance trade-off [75].

The limited size of OCT datasets, which may consist of only hundreds of cross-sectional images, can result in deterioration in the model's performance on the test data despite improving on the training data. Overfitting, which is more severe with smaller datasets, is a significant problem in machine learning.

Moreover, objective functions, also referred as loss functions, are design features that affect algorithm performance and evaluate the error caused by parameter values (the significance and implications of these functions, is examined in section 4.6) [50].

Another issue with employing supervised learning techniques to classify OCT images is that various examiners tend not to agree on the categorization of certain images. It is best to generate durable ground truth labels by incorporating review from multiple graders since supervised learning algorithms learn from these labels [50].

Convolutional Neural Networks (CNN)

Deep learning has seen significant success in recent time with applications like speech recognition, image processing, language translation, etc. This type of neural networks in general refer to neural networks with many layers and large number of neurons, often layered in a way that is generally not domain specific [51]. Contemporary state-of-the-art image analysis network are deep learning architectures called CNN [50].

Image classification based on CNN analyses the image as a matrix of pixel values and depending on the kind of layer does matrix operations with these values to obtain a vector containing a probability value corresponding to each of the classes. To the traditional neural network design (totally connected network), this DL architecture adds a number of layers that automatically extract relevant features from the input images. These features are dynamically chosen and altered during the learning process.

A generic CNN architecture can be visualized in Figure 35, which usually comprises of several convolutional layers, followed by fully connected layers, and an activation function.

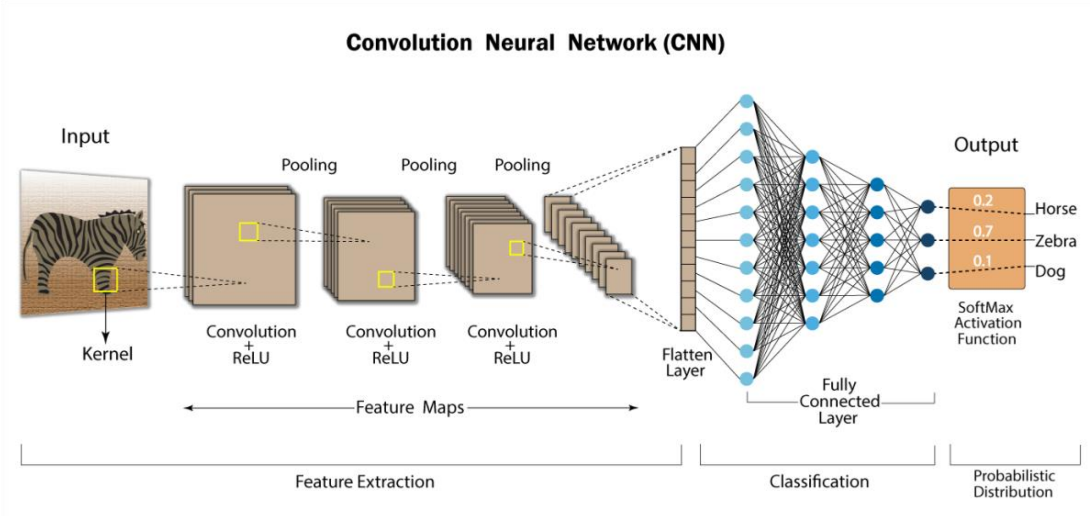
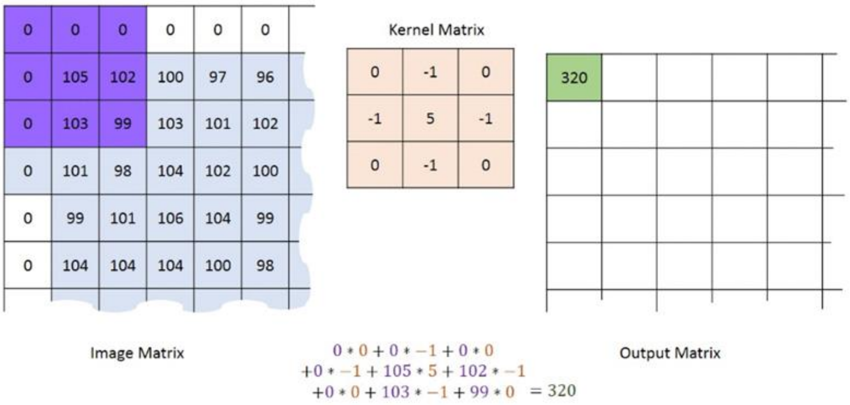


Figure 35: Schematic representation of a convolutional neural network [76].

Convolutional Neural Networks utilize mathematical operations known as convolutions for feature extraction. In this process, two functions are combined to create a new function as an output. CNNs use filters, commonly referred to as kernels, to convolve input pictures, producing a feature map as a result. The filters move along the input image, convolving it, and have a typical form of 3x3 or 5x5. The mathematics that underlies the convolution process is shown in Figure 36.

Each layer convolves its input with multiple filters learned by the network. Higher layers often learn increasingly higher-level characteristics, such as textures and complete forms, whereas lower layer filters typically learn lower-level elements, such as lines and corners [77].



0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

320				

Output Matrix

$$0*0+0*-1+0*0$$
$$+0*-1+105*5+102*-1$$
$$+0*0+103*-1+99*0 \quad =320$$

Figure 36: Arithmetic behind convolution [78].

The convolved features are controlled by three parameters:

- Depth: defines the number of filters to apply during the convolution;
- Stride: defines the number of “pixel’s jump” between two slices (if the stride is equal to 1, the filter will move with a pixel’s spread of one);
- Padding: is used to make dimension of output equal to input by adding zeros to the input frame of matrix. allowing more spaces for kernel to cover image and is accurate for analysis of images.

The choice of activation function for each convolutional section is a critical aspect to consider since it determines the behavior of the model. Typically, after convolution, the ReLU activation function is applied, which allows the neural network to capture non-linear relationships. ReLU works by setting all negative values to zero while keeping the current situation for all positive values reducing the vanishing gradient problem and accelerates model training. Another common activation function is the Sigmoid function that presents an output value between 0 and 1 [79].

After this, the input picture is then subjected to a pooling operation with the aim of reducing the input image's dimensionality and the operation's computational performance. There are several pooling methods but the commonly used are “Max Pooling” and “Average Pooling”. “Max Pooling”, as represented below, computes the maximum value present in the region covered by the kernel for each patch of the feature map, whereas “Average Pooling” calculates the average of all the values contained in the region covered by the kernel for each patch. Figure 37 represents a pooling operation example.

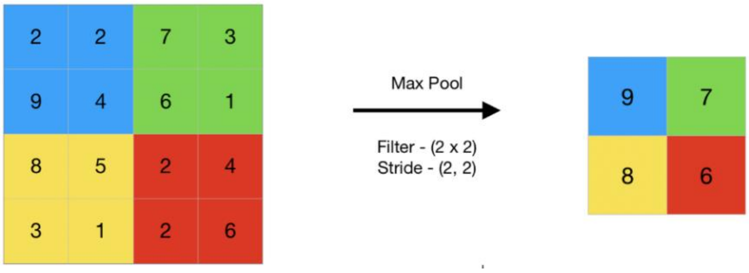


Figure 37: Example of a pooling operation [80].

Upon completion of the final Convolution Layer, ReLU activation, and Pooling Layer, the resulting output feature map (matrix) is transformed into a one-dimensional vector through a process known as flattening. The feature vector obtained from the flattening layer is then fed to a fully-connected layer that functions similar to a traditional neural network [76].

This fully-connected layer is utilized to classify images into distinct categories after the model has been trained. The Softmax activation layer is often applied to the final layer of the network to serve as a classifier. At this layer, the classification of the given input into distinct categories is performed. As last parameter, gradient descent and Adam are the most common optimization algorithm with the objective to modify the weights and biases of the network with the objective to minimize a certain cost function.

To summarize the entire process, the network is defined by the number of the filters, the stride lengths, the number (and sequence of) convolution pooling combinations, and the neural network [51].

Transfer Learning

Nevertheless, obtaining specific image datasets for various applications is often challenging, leading to limited data availability. Transfer Learning is a technique that enables a model to be trained and refined for a particular task and then applied to a closely related but different task [81].

Using this method, a previously trained model is applied to a new problem, enabling the training of deep neural networks with less data. This is very helpful since real-world issues sometimes lack the millions of labelled data points needed to train such sophisticated models. The basic idea is to apply the information a model has learned from a task that has numerous labelled training data to a new task with insufficient training data. Instead of initiating the learning process from scratch, the learned patterns from solving a related task are utilized. Figure 38 represents this approach.

Utilizing a portion of a trained model, by preserving the already converged weights, is referred to as freezing. In addition to the aforementioned benefits, this technique enables a reduction in training time and computational resources required for training these highly complex models [82].

In a typical transfer learning workflow, the process begins with the selection of a pre-trained base model that has been trained on a large dataset. The pre-trained weights of the base model are then loaded into the model and to ensure that the pre-trained weights are not modified during training, all the layers in the base model are frozen. Afterwards, a new model is constructed on top of the base model by incorporating additional layers. This new model is trained using the new dataset, with only the weights of the newly added layers being updated. This approach allows the model to learn task-specific features while leveraging the learned representations from the pre-trained base model.

Alternatively, fine-tuning can be employed, which involves unfreezing the entire model or a portion of it, and re-training it on the new data using a very low learning rate. This technique aims to adapt the pre-trained features to the new data, potentially leading to significant improvements in performance. Fine-tuning allows for incremental adjustments to the pretrained features based on the specific requirements of the new task [83].

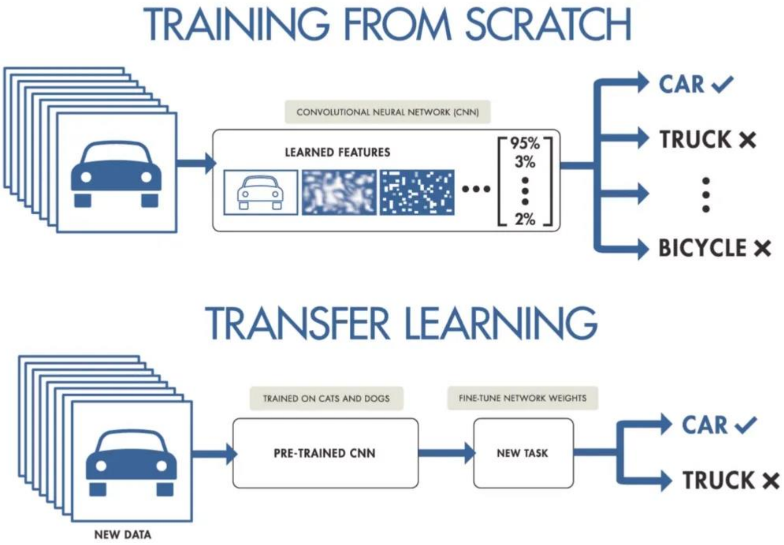


Figure 38: Transfer Learning diagram [84].

4.5 Regularization Methods

Regularization techniques are a collection of methods used to address the problem of overfitting in neural networks, enhancing the accuracy of Deep Learning models when confronted with new data from the problem domain. According to Goodfellow et al. [1], regularization involves modifying a learning algorithm to minimize generalization error while maintaining training error. As model complexity increases, the training error decreases initially, but the test error eventually starts to rise.

Consequently, to achieve low test error and strong generalization capabilities, it becomes essential to regulate the complexity of the neural network. Various regularization strategies exist to achieve this purpose.

Early Stopping

Early stopping is a regularization technique used to find the optimal point at which a model has learned enough to generalize well to unseen data without overfitting. As it is represented in Figure 39, early stopping involves monitoring the model's performance on a validation set during the training process and stopping the training when the performance starts to degrade. By doing so, early stopping prevents the model from memorizing the training data too closely and encourages it to learn more generalizable patterns.

The goal of early stopping is to strike a balance between bias and variance, as stopping too early may increase bias while stopping too late may increase variance [85]. The optimal model complexity is reached when the test error starts to increase again, indicating that further training would lead to overfitting. By stopping at this point, the model's performance on unseen data can be maximized.

Early stopping does not require introducing additional parameters or modifying the loss function. Instead, it relies on monitoring the validation set performance and storing the best model parameters during training. Once the performance no longer improves after a certain number of iterations, the training process is halted, and the last best parameters are used as the final model [86].

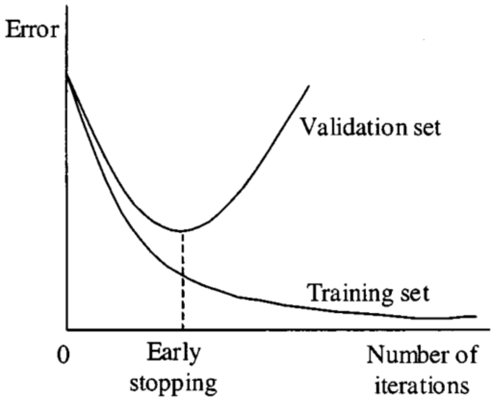


Figure 39: Early Stopping [86].

Data Augmentation

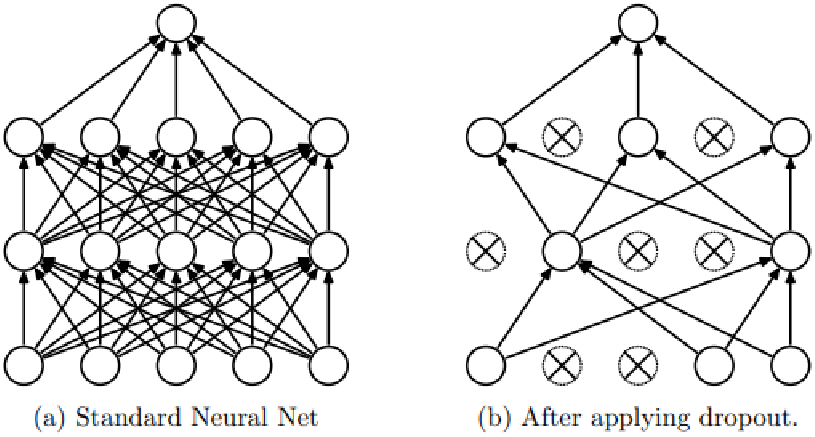
Data augmentation is a technique used in machine learning to overcome challenges caused by limited and expensive training data. It works by creating modified copies of existing data, which helps to make the training set larger and more diverse. This is done by making small changes to the data or generating new data points with different variations and perspectives. The main goal of data augmentation is to improve how well machine learning models perform and generalize. By expanding the dataset, data augmentation increases the diversity and variability of the available data, which can lead to better model performance and generalization.

In the specific context of OCT images, data augmentation proves particularly beneficial due to the inherent imperfections present in the acquired images. Through the generation of new and slightly different training examples, machine learning models can better learn the patterns of imperfect OCT images, ultimately resulting in higher accuracy and improved predictions in real-world scenarios.

Dropout

To mitigate the risk of overfitting in neural networks, reducing the number of trainable model parameters can be an effective approach. Overfitting often occurs when neural networks have larger sizes, with more layers and nodes per layer.

A regularization technique called dropout was introduced by Srivastava et al. [87] to address this issue. It involves randomly disabling neurons with a predefined dropout-rate probability P during network training. Each unit in the neural network has a chance of being "dropped out" in each training iteration. By doing so, the network is encouraged to learn more robust and generalizable representations, enhancing its ability to generalize and avoiding overfitting scenarios. Figure 40 and Figure 41 represents a neural network using dropout layers [69].


(a) Standard Neural Net

(b) After applying dropout.

Figure 40: Neural Network Dropout [87].

Additionally, training and evaluating multiple neural networks individually can be computationally expensive and memory-intensive. Dropout serves as a technique that approximates ensembling of exponentially many neural network architectures in an efficient and straightforward manner. It can be viewed as an ensemble technique, where multiple sub-networks are trained simultaneously by "dropping" certain connections between neurons.

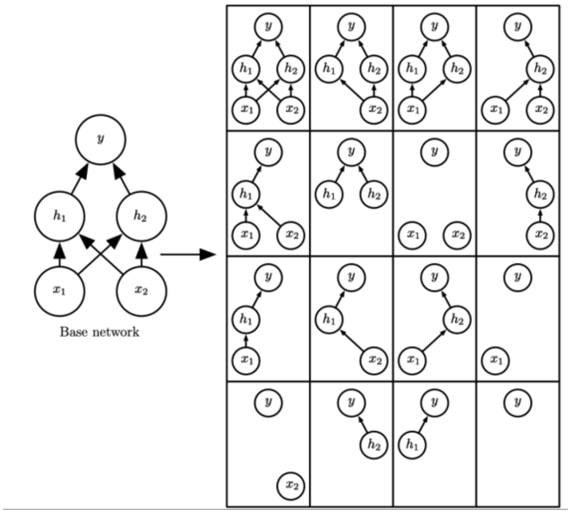


Figure 41: Possible networks constructed with dropout [1].

L1 and L2 Regularization

These methods introduce a penalty term into the loss function, augmenting it with additional terms to discourage the utilization of excessively large weights by the model - the fundamental principle is to identify and eliminate weights that contribute insignificantly to the model's accuracy [69].

L1 regularization enforces a penalty that is proportionate to the absolute values of the weights, while L2 regularization imposes a penalty proportional to the squared values of the weights. Equations 4.13 and 4.14 shows these two regularization methods.

L1 regularization, by augmenting the loss function with the sum of the absolute values of the weights, effectively constrains the weights towards zero. This constraint can even lead some weights to become exactly zero, thereby functioning as a feature extractor. In contrast, L2 regularization reduces the influence of large weights by appending the sum of the squared values of the weights to the loss function. The magnitude of the penalty exerted by regularization is determined by the parameter α, typically satisfying the range of $0 \leq \alpha \leq 1$ [69].

The selection between L1 and L2 regularization hinges upon the specific problem at hand and the desired characteristics of the model. L1 regularization tends to generate sparse models characterized by a subset of influential features, whereas L2 regularization leads to a more balanced distribution of weights across the model [88].

$$L1 = \lambda \sum_{i=1}^{N} |W_i| \tag{4.13}$$

$$L2 = \lambda \sum_{i=1}^{N} |W_i|^2 \tag{4.14}$$

Batch Normalization

Batch normalization is a technique used to normalize the activations of hidden layers during training. It serves multiple purposes, not only to enhance training speed and optimization but also to act as a regularization strategy.

One of the challenges in neural network training is the internal covariance shift. This occurs when the distribution of inputs to subsequent layers changes due to parameter updates during learning. This instability can hinder training convergence and impact the performance of later layers. Furthermore, batch normalization introduces a form of noise during training. Since each training sample can lead to different weight updates based on the current batch selection, batch normalization acts as a source of noise. Adding noise is a well-known technique to prevent overfitting, as it helps the model avoid fitting too much to the training data [69].

The Batch Normalization layer is a technique that operates in several steps to normalize the activation values within a batch of data. It starts by determine the mean μ and the variance σ^2 of the activation values across the batch, using equation 4.15 and 4.16. Then it normalizes the activation vector $\hat{x}_{norm}^{(i)}$ with equation 4.17 - that way, each neuron's output follows a standard normal distribution across the batch. Finally, the layer's output, $Z^{(i)}$, is obtained by applying a linear transformation to the normalized activations, using two learnable parameters: γ (scaling parameter) and β (shifting parameter): γ allows to adjust the standard deviation and β allows to adjust the bias, shifting the curve on the right or on the left side. The normalization procedure can be summarized in the following way with these 4 equations:

$$\mu = \frac{1}{n} \sum_i x^{(i)} \tag{4.15}$$

$$\sigma^2 = \frac{1}{n} \sum_i (x^{(i)} - \mu)^2 \tag{4.16}$$

$$\hat{x}_{norm}^{(i)} = \frac{x^{(i)}-\mu}{\sqrt{\sigma^2-\varepsilon}} \tag{4.17}$$

$$Z^{(i)} = \gamma * \hat{x}_{norm}^{(i)} + \beta \tag{4.18}$$

In the equations above, n represents the number of instances in a specific batch; $\hat{x}_{norm}^{(i)}$ is the zero-centered and normalized input for instance i; γ is the scaling parameter for the layer; β is the shifting parameter (offset) for the layer; ε is a constant used for numerical stability and to avoid division by zero and $Z^{(i)}$ is the output of the Batch Normalization operations. Thus, in total, four parameters must be learned for each batch-normalized layer: γ, β, μ and σ.

4.6 Performance Metrics

The dissertation employs various metrics to evaluate the performance of a classification model on automatic image classification using machine learning. The traditional accuracy metric may not be sufficient if the distribution of class labels is imbalanced. Precision-Recall metrics are especially useful when dealing with highly imbalanced classes, where precision measures the ability of the model to identify relevant data points while recall measures the ability of the model to find all relevant cases. Furthermore, the confusion matrix is a suitable method for summarizing the performance of a classification algorithm in the context of imbalanced classes [70].

The subsequent equations are commonly employed metrics in the field of machine learning. In these metrics, TP denotes true positives, FP represents false positives, FN describes false negatives, and TN denotes true negatives.

- **Accuracy**: this metric expresses the percentage of predictions that were made correctly.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4.19}$$

- **Recall** (or sensitivity): is also called the true positive rate (TPR), and measures the proportion of true positives among all actual positives in a classification problem. This metric provides an indication of model's reliability in labeling positives units in the dataset.

$$Recall = \frac{TP}{TP+FN} \tag{4.20}$$

- **Specificity** (or true negative rate, TNR): This metric measures the ratio of correctly identified negative cases to all actual negative cases.

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{4.21}$$

- **Precision**: measures the proportion of true positives among all positive predictions in a classification problem. This metric provides an indication of the model's reliability in labeling an individual as positive.

$$Precision = \frac{TP}{TP+FP} \tag{4.22}$$

- **F1-Score**: combines the precision and recall scores of a model. F1-score is especially useful when dealing with imbalanced datasets where one class has significantly more samples than the other [89].

$$\mathrm{F1} = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{4.23}$$

- **Fbeta-Score**: is an extension of the F1-score measure that incorporates an adjustable parameter known as beta as illustrated in equation 4.24. A smaller beta value, for instance 0.5, emphasizes precision to a greater extent. Conversely, a larger beta value bigger than 1, gives greater emphasis on recall. Specifically, the Fbeta score (with beta = 2) prioritizes the significance of recall over precision, indicating a focus on minimizing false negatives rather than false positives. When it comes to OCT images, this metric takes on added importance due to the nature of medical diagnoses - missing the detection of a critical finding can adversely impact patient outcomes. This prioritization is crucial for the early detection and prompt treatment of diseases, ultimately contributing to better patient care, improved treatment planning, and potentially saving lives.

$$\mathrm{F}_{\beta}\mathrm{Score} = (1 + \beta^2) \times \frac{Recall \times Precision}{(\beta^2 \times Precision) + Recall} \tag{4.24}$$

- **ROC Curve and AUC score:** The Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, represented in Figure 42, is a widely used metric. ROC curve is a probability curve that illustrates the model's performance at various classification thresholds by plotting the True Positive Rate (Recall/Sensitivity) against the False Positive Rate (1-Specificity). An ideal classifier would achieve a TPR of 100% with zero false positives, resulting in a ROC curve that reaches the upper left corner. The AUC, representing the area under the ROC curve, serves as a numerical measure of the model's predictive quality. A higher AUC indicates superior class prediction capabilities, with values closer to 1 implying stronger performance. The threshold refers as a critical parameter in the classification process, defining the probability point at which a sample is labeled positive or negative [90].
  In this work, a ROC curve and AUC score is implemented for each class using the one-vs-rest (OvR) approach. This approach involves constructing separate ROC curves for each class, where one specific class is designated as the positive class, while the remaining classes are considered as the negative classes. The utilization of the OvR approach facilitates the identification of classes that present greater difficulties in terms of evaluation.
  Additionally, both macro average and micro average ROC curves are plotted to further assess the performance of the classifier. The micro average approach involves the computation of a singular ROC curve by considering all predictions and true labels collectively, regardless of the specific class. On the other hand, the macro average ROC curve is generated by averaging the individual ROC curves obtained for each class [91]. Figure 42 represents an example of a ROC Curve and AUC score.

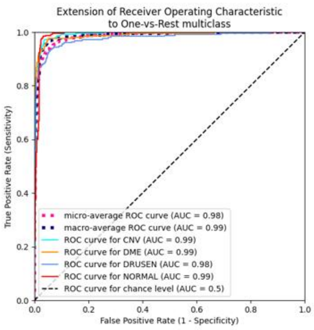
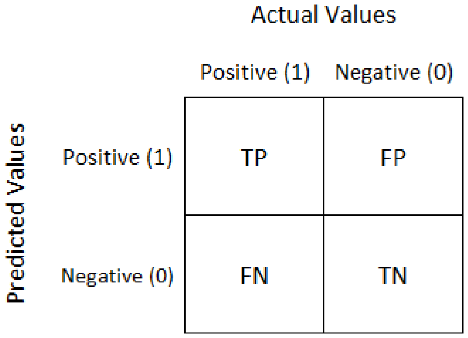


Figure 42: ROC Curve and AUC score example.

- **Confusion Matrix**: The most comprehensive performance metric for a classification model is the confusion matrix, represented in Figure 43 which provides an in-depth analysis of the model's behavior. In the case of a binary classifier, the confusion matrix displays a matrix that helps to evaluate the model's overall performance. The rows of the matrix correspond to the model's predictions, whereas the columns correspond to the true labels of the data samples. By analyzing the confusion matrix, we can gain insight into the model's strengths and weaknesses, which can guide us in refining further training to improve the model's performance.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 43: Confusion matrix [91].

4.7 Summary

This chapter provides a comprehensive overview of ML fundamentals and their applications. It begins by discussing the fundamental concepts of ML algorithms, with a specific focus on classification algorithms. It emphasizes the role of ML in ophthalmology and its potential for improving medical diagnosis and treatment.

The chapter also delves into the key ML models, namely Artificial Neural Networks and Convolutional Neural Networks. It explains their architectures, training processes, and evaluation metrics, shedding light on their capabilities in handling complex problems and analyzing data patterns. Additionally, the section introduces regularization methods, which aid in mitigating overfitting and enhancing the performance of ML models. Within the context of the dissertation, this section serves as a crucial reference for employing ML techniques. By incorporating ANN, CNN, and regularization, the research aims to achieve accurate and dependable results.

It also delves into the key ML models, namely Artificial Neural Networks and Convolutional Neural Networks. It explains their architectures, training processes, and evaluation metrics, shedding light on their capabilities in handling complex problems and analyzing data patterns. Additionally, the section introduces regularization methods, which aid in mitigating overfitting and enhancing the performance of ML models. Within the context of the dissertation, this section serves as a crucial reference for employing ML techniques. By incorporating ANN, CNN, and regularization, the research aims to achieve accurate and dependable results.

5 State of the Art

5.1 OCT Image Classification Databases

There are diverse publicly available OCT image classification datasets where machine learning techniques for autonomous diseases identification and diagnosis may be developed and tested. Some of these databases are dedicated to categorizing various types of disorders and contain diverse sample sets. The subsequent datasets are among the most popular for OCT image classification:

Srinivasan: has 3231 SD-OCT scans from 45 patients: 15 normal subjects, 15 patients with AMD and 15 patients with DME**.** All SD-OCT volumes were acquired in Institutional Review Board-approved protocols using Spectralis SD-OCT (Heidelberg Engineering Inc., Heidelberg, Germany) imaging at Duke University, Harvard University, and the University of Michigan. All the images are in TIFF file [48], [85]. Table 1 summarizes the specifications of Srinivasan database.

Table 1: Specifications for Srinivasan database [92].

Class	Number of Patients	OCT B-Scans
Normal	15	1407
AMD	15	723
DME	15	1101
Total	45	3231

Duke OCT: the Duke database contains 38400 SD-OCT B-Scans data of 384 patients, including 115 elderly subjects without AMD and 269 subjects with intermediate AMD, aged between 50 and 85 years. The imaging system used for data collection was the Bioptigen SD-OCT imaging system. Participants in the research required to be between the ages of 50 and 85 and have intermediate AMD with big drusen (>125 mm) in either both eyes or in one eligible eye and severe AMD in the other eye, with no history of vitreoretinal surgery or ophthalmologic disease that might affect acuity in either eye. The data is available in a mat file format [86].
Table 2 summarizes the specifications of Srinivasan database.

Table 2: Specifications for the Duke database [93].

Class	Number of Patients	OCT B-Scans
Normal	115	11500
AMD	269	26900

Kermany: 109312 OCT JPEG images were collected from adult patients in multiple institutions from 2013 to 2017. These images were obtained from 5761 patients and included 37206 images with CNV, 11349 with DME, 8617 with drusen, and 51140 normal images. The OCT scans were obtained using the Spectralis OCT system from Heidelberg Engineering. The patients were from the Shiley Eye Institute of the University of California San Diego, the California Retinal Research Foundation, Medical Center Ophthalmology Associates, the Shanghai First People’s Hospital, and

Beijing Tongren Eye Center. All OCT imaging was performed during patients' routine clinical care, and there were no exclusion criteria based on age, gender, or race [41]. Table 3 summarizes the specifications of Kermany database.

Table 3: Specifications for the Kermany database [41].

Class	Number of Patients	Mean Age (Years)	OCT B-Scans
CNV	791	83 (Range: 58-97)	37206
DME	709	57 (Range: 20-90)	11349
Drusen	713	82 (Range: 40-95)	8617
Normal	3548	60 (Range: 21-86)	51140

Noor Hospital: the Noor Eye Hospital in Tehran, Iran, used the Heidelberg SD-OCT imaging equipment to obtain the OCT scans. Patients were chosen for the dataset based on certain inclusion criteria, including age above 50, the absence of any other retinal disease in the patient's OCT B-scans, and acceptable image quality [43]. There are two options for processing a dataset of medical images in this dataset: the first option involves reading all images, which would result in a total of 16,822 images being used and second option involves keeping only the "worst-case condition" (if a patient was detected as a CNV case, only CNV-appearing B-scans were included for the training procedure) [87].
A retinal specialist labels each OCT B-scan and the dataset is shown in Table 4.

Table 4: Specifications for the Noor Eye Hospital database [43].

Class	Number of Patients	Eyes (Right eye, left eye)	OCT B-Scans
Normal	120	187 (95,92)	5667
Drusen	160	194 (112,82)	3742
CNV	161	173 (83,90)	3240

OCTID: the database consists of high-resolution spectral domain OCT volumetric scans that were obtained at the Sankara Nethralaya eye hospital in Chennai, India, using a Cirrus HD-OCT device (Carl Zeiss Meditec, Inc., Dublin, CA). The five categories present in the database are Normal (NO), Macular Hole (MH), Age-Related Macular Degeneration (AMD), Central Serous Retinopathy, and Diabetic Retinopathy (DR). Clinicians at SN hospital made diagnoses on the diseases, and image class labelling was done in accordance with their findings. In all, there are 102 MH, 55 AMD, 107 DR, 102 Central Serous Retinopathy and 206 NO retinal images [88].

The five databases are accessible to the general public and the information from the databases is summarized in Table 5.

.

Table 5: Database Characteristics

Database	N.Patients	N.Images	Eye diseases	File Format	Device (manufacturer)
Srinivasan	45	3231	AMD, Normal, DME	TIFF	Heidelberg SPECTRALIS SD-OCT
Duke	384	38400	AMD, Normal	MAT	SD-OCT imaging system (Bioptigen)
Kermany	5761	109312	CNV, Drusen, Normal, DME	JPEG	Heidelberg SPECTRALIS SD-OCT
Noor Hospital	441	16822	Normal, Drusen, CNV	TIFF	Heidelberg SPECTRALIS SD-OCT
OCTID	-	572	MH, AMD, DR, Normal, Central Serous Retinopathy	JPEG	Cirrus HD-OCT machine (Carl Zeiss Meditec)

5.2 Image Classification: Traditional Methods

Preprocessing, feature extraction, and classifier design are the three main components of traditional ML approaches. At the preprocessing step, methods such image denoising is used to remove unnecessary or redundant information from the input data, making it possible for important information to be extracted later. Following this, feature descriptors including scale-invariant feature transform (SIFT), LBP and HOG, are used to make the manual extraction of features easier. Lastly, a classifier such as random forest algorithm, a Bayesian classifier, or a support vector machine performs the classification process using the extracted features [43].

A summary of earlier works utilizing traditional methods is provided, followed by an exposition on the main classification techniques employed.

Srinivasan et al. [85] utilizes multiscale HOG descriptors as feature vectors of a support vector machine-based classifier to classify patients with Normal, AMD and DME diseases. The Duke OCT database was used as the dataset, consisting of cross-sectional volumetric scans acquired from 45 subjects, including 15 normal subjects, 15 patients with AMD, and 15 patients with DME. It was obtained an accuracy of 95.56% for patient-wise classification of normal, AMD, and DME cases.

Albarrak et al. [89] proposed an automated method for identifying AMD in 3D OCT images. Preprocessing of the retinal volumes is initially conducted to extract a Volume of Interest (VOI) that includes the retina, followed by flattening of the retina (warping) to improve image quality. The pre-processed volume is then subjected to a feature extraction approach to obtain a set of local histogram-based feature vectors. The combination of image decomposition and LBP histograms leads to a more accurate feature descriptor for classification purposes. The proposed method achieved an accuracy of 91.4%, sensitivity of 92.4%, and specificity of 90.5%. The database had 140 volumetric 3D images.

Lemaitre et al. [90] proposed a classification framework with five distinctive steps including preprocessing (non-local means, flattening, alignment), feature detection (LBP, LBP-TOP), mapping (global, local), feature representation (histogram bag of-words), and classification (random forest, k-NN, RBF-SVM, logistic regression, and gradient boosting). Their study shows that this method outperforms previous studies, achieving a sensitivity and specificity of 81.2% and 93.7%, respectively. According to the authors, the greatest results are obtained when utilizing 3D features and high-level representations of 2D characteristics using patches. Preprocessing had varying results depending on the classifier and feature sets, though. A private dataset of 32 OCT volumes was used for the studies, 16 of which contained instances of DME, while the other 16 volumes had cases of normal vision. A total of 128 B-scans with a resolution of 512 x 1024 pixels made up each volume.

Sun et al. [91] proposed a framework for automated detection of dry age-related AMD and DME from retina OCT images, based on sparse coding and dictionary learning. The proposed method utilizes two techniques for image classification: spatial pyramid with sparse coding as well as a multiclass linear SVM. This study employed 168 AMD, 297 DME, and 213 normal OCT B-scans of a retina from a private dataset in addition to the Duke OCT database. For the Duke dataset the proposed approach performed better, achieving a patient-wise accuracy of 97.78%. However, a single OCT scan's average preprocessing time was 9.2 seconds, which could prevent its usage in real-time scenarios.

Liu et al. [92] presents an effective data-driven approach for identifying normal macula and multiple macular pathologies, such as macular edema, macular hole, and age-related macular degeneration, in retinal OCT images. The authors use a machine learning approach based on global image descriptors formed from a multi-scale spatial pyramid and dimension-reduced Local Binary Pattern histograms to encode texture and shape information in retinal OCT images and their edge maps, respectively. The approach captures the geometry, texture, and shape of the retina in a principled way, and a binary non-linear SVM classifier is trained for each pathology to identify its presence. The proposed method was tested on a large dataset of 326 OCT scans from 136 subjects, and the results showed that the method is very effective, with all AUC values greater than 0.93.

Hasan et al. [63] utilized handcrafted feature extraction methods, including HOG, LBP, SIFT, and Speeded-Up Robust Features (SURF), to extract features from OCT images. The researchers used Kermany OCT image dataset comprising four types of retinal diseases (CNV, DME, Drusen and normal images). They trained the dataset in both imbalanced and balanced groups, where each balanced group consisted of around 2,000 images of each class. For classification, they employed the SVM classifier. The results indicated that the HOG feature achieved the best performance.

Table 6 summarizes the previous works mentioned.

Table 6:Selection of previous works that used feature-based techniques.

Authors	Database	Methods	Results
Srinivasan et al. [85]	Duke OCT	HOG as feature extractors and SVM as classifier	Accuracy of 95.56%
Albarrak et al. [89]	Private Database	Use Bayesian classifier on the feature vectors produced by combining the principles of volume decomposition and LBP.	Accuracy: 91.4%, Sensitivity: 92.4% Specificity: 90.5%.

Authors	Database	Methods	Results
Lemaitre et al. [90]	Private Database	A classification framework with five distinctive steps: preprocessing, feature detection, mapping, feature representation and classification	The best parameters obtained a sensitivity of 81.2% and a specificity of 93.7%.
Sun et al. [91]	Duke OCT and Private Database	A classification approach based on dictionary learning and sparse coding was developed.	Accuracy of 97.78% on the Duke database.
Liu et al. [92]	Private Database	A ML approach based on multi-scale spatial pyramid and LBP histograms was used to diagnose macular pathologies in OCT images.	The proposed approach achieved high effectiveness in identifying macular pathologies in OCT images, with AUC > 0.93.
Hasan et al. [63]	Kermany Database	Use four feature extractors (HOG, LBP, SIFT and SURF) and SVM as classifier.	The best parameters obtained a precision of 79% and recall of 79,4% using HOG as feature extraction method.

5.3 Image Classification: Deep Learning Methods

The difficulty of traditional algorithms to properly generalize on AI problems led to the development of deep learning: when working with high-dimensional data, the difficulty of generalizing new examples increases exponentially and typical machine learning generalization techniques are shown to be ineffective for learning complex operations in high-dimensional domains, which are usually linked with significant processing costs [1].

Deep learning (DL) algorithms provide the advantage of automatic feature extraction over other machine learning methods, which relieves the programmer's responsibility of explicitly choosing the necessary features. This benefit is further highlighted by the simultaneous application of all such steps within the model itself, as shown in Figure 44.

There has been a noticeable increase in interest recently in using DL methods in the areas of medicine and healthcare, with a focus on medical image analysis. In particular, the use of CNN architectures in the categorization of retinal diseases using OCT images has produced good outcomes [43].

In this section, it is explained in more detail how deep learning, artificial neural networks operate, followed by a summary of works utilizing deep learning methods. CNN models, which demonstrate higher performance in resolving such problems, must be adequately examined given the emphasis of this research on the image classification capabilities of DL models.

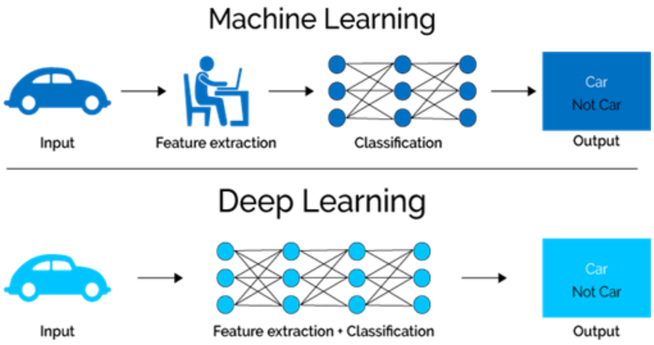


Figure 44:Machine Learning and Deep Learning difference [99].

Image Classification OCT Deep Learning Methods

Deep learning methods in OCT image classification have shown superior results than traditional methods. In this regard, several studies employing convolutional neural networks CNN for OCT image classification are described, with emphasis on those utilizing the same databases mentioned before.

Lee et al. [94] were the first to demonstrate the capability of DL models to distinguish AMD from normal OCT images. It used a modified version of the VGG16 convolutional neural network as the deep learning model for classification and the training process involved multiple iterations, each with a batch size of 100 images and a starting learning rate of 0.001 using stochastic gradient descent optimization. The model's loss was recorded after each iteration, and every 500 iterations, its performance was assessed using cross-validation with the validation set. Using a private dataset comprising 48312 normal and 52690 AMD macular OCT scans, the model achieved an accuracy of 87.63% at the OCT level, 88.98% at the volume level, and 93.45% at the patient level.

Kermany et al. [41] achieved an accuracy of 96.6% with a sensitivity of 97.8%, a specificity of 97.4%, and a weighted error of 6.6% in a multi-class comparison between choroidal neovascularization, diabetic macular edema, drusen, and normal, using Kermany database. In this study, a limited model was also trained to classify four categories using only 1000 randomly selected images from each class during training (it achieved an accuracy of 93.4%, a sensitivity of 96.6%, a specificity of 94.0%, and a weighted error of 12.7%.) to compare the transfer learning performance using limited data with the results obtained using a large dataset. The transfer learning methods were adapted from an Inception V3 architecture and training of layers was carried out using Adam Optimizer in batches of 1000 images per step, having a learning rate of 0.001. The training was conducted for all categories over 10,000 steps or 100 epochs, considering the convergence of the final layers for all classes would have occurred by then. This study revealed the competitive performance of the transfer learning approach, which does not require a dataset of millions of photos or a highly specialized deep learning model.

Li et al. [95] follow a similar methodology to that used by Kermany et al. [41], except for the use of the VGG16 network instead of InceptionV3 for transfer learning. Li et al. [95] achieved a prediction accuracy of 98.6%, with a sensitivity of 97.8% and a specificity of 99.4%, demonstrating transfer learning method based on the VGG-16 network shows significant effectiveness on classification of retinal OCT images. VGG-16 was composed of 13 convolution layers, five max-pooling layers, and three fully-connected layers.

Hwang et al. [47] conducted a study where they trained three different CNN models (VGG16, InceptionV3, and ResNet50) to classify OCT images into four categories: normal macula, dry AMD (drusen), inactive wet AMD, and active wet AMD. It used a private database and Kermany database. To evaluate the performance of the models, four reviewers were recruited to compare the AI models and clinical reviewers. The verification dataset consisted of 3,872 images divided into four categories and the clinical OCT images from Kermany database were divided into three categories. After training, the models showed high accuracy during verification, with reported accuracies of 91.20%, 96.93%, and 95.87% for VGG16, InceptionV3, and ResNet50 models, respectively, for the classification of normal, dry AMD, and wet AMD cases and have identical or even better accuracy than clinical reviewers. The models were trained using an Adam Optimizer in batches of 64 images per step, with a learning rate of 0.001. The training was run for 100 epochs.

Serener et al. [96] conducted a study to compare AlexNet and ResNet18 models for the classification of dry and wet AMD. The results of the study indicated that for the classification of both dry and wet AMD, the ResNet model yielded superior results when compared to the AlexNet model, with an accuracy of 99.5%, sensitivity of 98.0% and specificity of 100.0% for Dry AMD and an accuracy of 98.8%, sensitivity of 95.6% and specificity of 99.9% for Wet AMD. The proposed CNN network was trained on the Mendeley dataset and tested on four datasets, including Mendeley, OCTID, Duke, and SD-OCT Noor dataset, achieving high accuracy rates of 99.73%, 98.08%, 96.66%, and 97.95%, respectively. The authors compared the proposed method with alternative methods and showed that their algorithm is more efficient in detecting AMD. The sigmoid function is used as the classifier in this proposed network.

Kaymak et al. [97] presented a novel approach to transfer learning by utilizing the AlexNet model. This method was found to achieve superior performance compared to the approach employed by Kermany et al. [41]. During the training process, the network parameters were set to specific values, namely a learning rate of 0.01, momentum of 0.9, and weight decay of 0.0005. The model was generated by training for 800 epochs.

Shatil et al. [98] evaluated a CNN model and two transfer learning models (ResNet152V2 and DenseNet169) using the Kermany Database. The models were fine-tuned and compared for disease classification. Test accuracies were CNN: 98.34%, ResNet152V2: 99.17%, and DenseNet169: 99.38%. All models used categorical cross-entropy loss and Adam optimizer. During transfer learning, base models were frozen, trained with a learning rate of 0.001, and mini-batches of size 80. Later, base models were unfrozen, retrained with a learning rate of 1e-4. Early stopping with patience 8 saved the model with least validation loss. The CNN model had a learning rate of 0.0001, batch size 16, and early stopping after ten epochs without reduced validation loss. Reduce learning rate on plateau was used with patience 5 and minimum learning rate 1e-6.

In their study, Tasnim et al. [99] conducted a comparative analysis of four different models for the purpose of detecting retinal diseases. The evaluated models include a vanilla CNN model, Xception model, ResNet50 model, and MobileNetV2 model. The detection accuracies on the test set were found to be 98.00% for the vanilla CNN model, 99.07% for the Xception model, 97.00% for the ResNet50 model, and 99.17% for the MobileNetV2 model. MobileNetV2 model achieved the highest accuracy among the evaluated models, closely followed by the Xception model. These findings indicate the effectiveness of the proposed approach in automating the detection of retinal diseases. The Kermany database served as the dataset for this study.

Asif et al. [64] proposed a model based on the ResNet50 architecture with modifications: the fully connected layer of ResNet50 was replaced with a new fully connected layer. The proposed model was trained and evaluated on Kermany dataset. The proposed model achieved an improved overall classification accuracy of 99.48%, with only 5 misclassifications. It outperformed existing methods on the same dataset. The Adam optimizer was used, along with L2 regularization (0.001) in the dense layers to mitigate overfitting. The training process utilized an initial learning rate of 0.001, a batch size of 32, and 20 epochs.

ReduceLROnPlateau was employed to decrease the learning rate when improvement ceased (patience=5, min_lr=0.000001). The early stop method was also implemented to halt training when the model stopped improving and was trained using the cross-entropy loss function, with data shuffling enabled.

Shurrab et al. [100] conducted a study comparing two pretraining approaches, namely Self-Supervised Learning and Transfer Learning, for training the ResNet34 neural architecture. The research findings demonstrated that the SSL ResNet34 model achieved superior performance compared to the TL ResNet34 model. The SSL ResNet34 model exhibited an overall accuracy of 95.2%, sensitivity of 95.2%, and specificity of 98.4%. In contrast, the TL ResNet34 model achieved scores of 90.7% for overall accuracy, 90.7% for sensitivity, and 96.9% for specificity. During training, the cross-entropy loss function was employed, along with the Adam optimizer. The batch size was set to 16.

Arora et al. [101]utilized the Kermany database and focused on leveraging the VGG16 model architecture. Their model achieved an accuracy of 99% and precision of 98.8%, surpassing other state-of-the-art approaches examined in their study. To enhance the model's performance, the fully connected layers were carefully designed. Fine-tuning was performed by unfreezing all layers of the model and retraining it as a whole using the complete dataset, incorporating the early stop algorithm.

Finally, Nugroho [102] conducted a study comparing handcrafted feature extraction methods with deep neural network-based methods for classifying OCT images. The dataset used in the study was Kermany Database. The handcrafted feature extractors evaluated were HOG and LBP, while the deep neural network models employed were DenseNet-169 and ResNet50. The results demonstrated that the deep neural network-based methods outperformed the handcrafted feature extraction methods. The DenseNet-169 achieved an accuracy of 88%, and ResNet50 achieved an accuracy of 89%. In contrast, the handcrafted feature extraction methods achieved accuracies of 50% for HOG and 42% for LBP.

Numerous research papers have employed deep learning methods for various applications. In the present work, specific references have been chosen for comparison purposes since they utilize the same database as that of this dissertation. The decision was made to compare the outcomes of the present study with those of the selected references in the existing literature, as the use of a common database can help facilitate such comparisons. Table 7 summarizes the previous works mentioned.

Table 7: Selection of previous works that used deep learning-based methods.

Authors	Database	Methods	Results
Lee et al. [94]	Private Database (48312 normal and 52690 AMD macular OCT scans)	Modified version of the VGG16 CNN was utilised.	Accuracy of 87.63% at the OCT level, 88.98% at the volume level, and 93.45% at the patient level.
Kermany et al. [41]	Kermany Database	A method for transfer learning to categorise CNV, DME, drusen, and normal cases using the InceptionV3 architecture.	Accuracy: 96.6% Sensitivity: 97.8% Specificity: 97.4%

Authors	Database	Methods	Results
Li et al. [95]	Kermany Database	A method for optimizing the VGG16 network that was trained on the ImageNet database.	Accuracy: 98.6% Sensitivity: 97.8% Specificity: 99.4%
Hwang et al. [47]	Kermany Database + Private Database	A method for fine-tuning the VGG16, InceptionV3, and ResNet50 architectures.	Accuracy for: VGG16: 91.20%; InceptionV3: 96.93%; ResNet50: 95.87%;
Serener et al. [96]	Kermany Database	A comparison of the AlexNet and ResNet18 models for identifying dry and wet AMD.	ResNet18 - Dry AMD: Accuracy: 99.5% Sensitivity: 98.0% Specificity: 100.0% ResNet18 - Wet AMD: Accuracy: 98.8% Sensitivity: 95.6% Specificity: 99.9%
Kaymak et al. [97]	Kermany Database	It was used AlexNet as transfer learning method	Accuracy: 97.1% Sensitivity: 99.6% Specificity: 98.4%
Shatil et al. [98]	Kermany Database	Evaluated a CNN model and two transfer learning models (ResNet152V2 and DenseNet169).	Accuracies for: CNN: 98.34% ResNet152V2: 99.17% DenseNet169: 99.38%.
Tasnim et al. [99]	Kermany Database	Used 4 models: vanilla CNN model, Xception model, ResNet50 model, and MobileNetV2 model.	Accuracies for: Vanilla CNN model: 98.00%, 99.07% for the Xception model, 97.00% for the ResNet50 model, and 99.17% for the MobileNetV2 model.
Asif et al. [64]	Kermany Database	Proposed a model based on the ResNet50 architecture with modifications in the fully connected layers.	The proposed model achieved an accuracy of 99.48%.
Shurrab et al. [100]	Kermany Database	Compared Self-Supervised Learning and Transfer Learning, for training the ResNet34 neural architecture.	SSL ResNet34: accuracy of 95.2%, sensitivity of 95.2%, specificity of 98.4%. TL ResNet34: 90.7% for accuracy, 90.7% for sensitivity, and 96.9% for specificity.
Arora et al. [101]	Kermany Database	Detection Using Transfer Learning on VGG16.	Accuracy of 99% and precision of 98.8%

Authors	Database	Methods	Results
Nugroho [102]	Kermany Database	Handcrafted feature extraction methods with deep neural network-based methods (DenseNet-169 and ResNet50).	DenseNet-169 accuracy of 88%; ResNet50 accuracy of 89%. The handcrafted feature extraction methods achieved accuracies of 50% for HOG and 42% for LBP.

5.4 Summary

The chapter discusses the concepts and applications of ML in ophthalmology and is divided into two sections: image classification on OCT exams using traditional methods versus deep learning methods. The focus is on how these algorithms can be used to classify images in OCT images - a major focus was given to convolutional neural networks, the most used architecture in deep learning for image classification analysis.

Additionally, to this, the section discusses the advantages and limitations of each method and provides examples of studies that have used both approaches, showing the results of studies for both methods. The studies demonstrate that deep learning methods have better results than traditional methods in terms of accuracy, sensitivity, and specificity. Artificial intelligence has the potential to revolutionize healthcare by improving diagnostic accuracy, reducing workload, and enhancing patient outcomes. However, the lack of publicly available datasets with complete metadata poses a significant challenge to the development of AI-based medical applications. The limited access to diverse datasets may result in biased and inaccurate predictions, leading to underperformance or even failure when transferred to different settings and patient groups [48].

Despite the healthcare sector being among the initial industries to acknowledge the vast potential of computer vision and the CNNs, the impact of an incorrect referral decision on a patient's outcome is a matter of significant concern. A false-positive result may lead to undue distress, unnecessary investigations, and an additional burden on the healthcare system [41], [109].

In summary, AI has the potential to change the healthcare industry, but its implementation requires careful evaluation of the quantity and quality of datasets, the precision and dependability of algorithms, and ethical and legal issues. It seems conceivable that AI will become increasingly important in disease detection and treatment as technology develops and more data becomes accessible.

6 Methodology

6.1 Database Selection

The OCT Kermany database [41] represented below was selected as the unique database partly due to its extensive collection of images, surpassing other databases in this regard. Moreover, it offers a more thorough database, since it can be divided into four distinct classes of diseases. It is worth noting that this database is frequently employed in similar studies, which highlights its utility for comparing results.

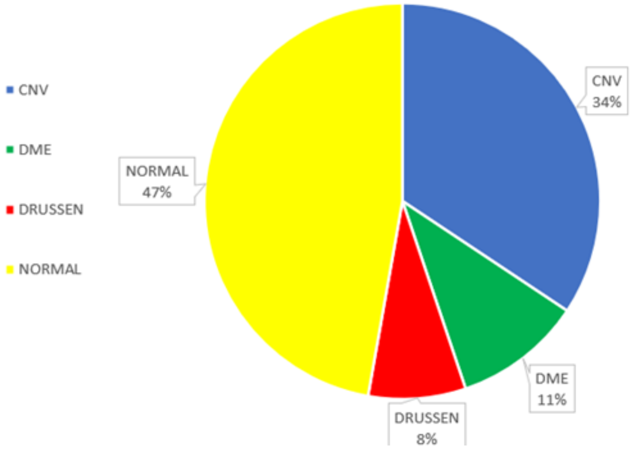


Figure 45: Distribution of Image Classes - Kermany Database.

In traditional methods is used a balanced dataset with less images. The primary objective of this methodology was to create a balanced dataset that would allow for a meaningful comparison of results with other state-of-the-art works. Table 8 shows the information about the two databases used in this work.

Table 8: Original and Balance database - number of images.

GROUP	CLASS				
	CNV	DME	DRUSEN	NORMAL	TOTAL
Original Database	37206	11349	8617	51140	108312
Balanced Database	1250	1250	1250	1250	5000

The latest version (version 3) of the dataset is employed for the analysis. It is worth noting that the majority of state-of-the-art studies have predominantly utilized version 2 of the database, which comprises a total of 84,568 images: the only difference is in normal images. Version 2 has 26315 and version 3 has 51140 images. Consequently, the dataset in version 2 is considered to be more balanced compared to version 3 due to the disparity in the number of normal images. To provide a visual representation of each disease examined in the study, Figure 46 showcases an image associated with each disease.

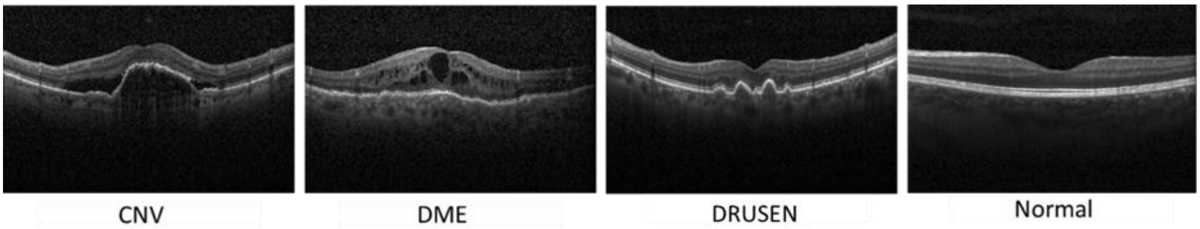


Figure 46: Comparison of OCT Scans for Different Diseases – Kermany Database [41].

6.2 Traditional Methods

All the previous works mentioned that employed traditional methods used small databases in comparison to the Kermany database. In order to ensure comparability with this study, a smaller balanced dataset that can be consulted at Table 8 is used. In traditional methods, the methodologies applied involved data preprocessing, feature extraction, and classifier training. Among the selected traditional approaches, only two or three diseases are compared in all works, except for Hassan study [70] that included the CNV disease in the comparison. Since CNV disease is known to be a challenging and important pathology, it is crucial to incorporate it in this analysis.

Preprocessing

Speckles and a large difference in the location of the retina between scans are two characteristics of OCT images that are frequently observed. These variances include changes in location, inclination angles, and the retina's inherent curvature. Therefore, regardless of the location in the retina, it is essential to account for these variances in OCT scans to guarantee consistent and accurate characterization of tissue disposition. This emphasizes the demand for methods that can eliminate these fluctuations, enabling a more dependable and precise examination of OCT images.

Regarding image preprocessing, several techniques are utilized. Firstly, image rescaling was performed to achieve a uniform size of 224x224. Additionally, white borders that appeared in the images were converted into black pixels as OCT images are mostly grayscale images, and these white regions can increase image denoising time. Non-Local Means (NLM) filtering was also employed, which preserves fine structures as well as flat zones by utilizing all possible self-predictions that the image can provide, rather than local or frequency filters such as Gaussian, anisotropic, or Wiener filters [90].

In contrast to local mean filters, which only compute the mean value of a small subset of pixels around the target pixel, non-local means filters compute the mean value of every pixel in an image. Based on how close each pixel is to the target pixel, the weight assigned to each pixel in the mean computation is determined, with pixels that are more similar receiving more weight. Non-Local means formula is defined in equation 6.1 as:

$$NL[v](i) = \sum_{j\epsilon I} w(i,j)v(j) \tag{6.1}$$

Considering a noisy image represented by a discrete set of pixels v = {v(i) | i ∈ I}, the estimated value NL[v](i), for a pixel i, is computed as a weighted average of all the pixels in the image. The weights w(i, j) are determined based on the similarity between the pixels i and j and satisfy the usual conditions $0 \leq w(i, j) \leq 1$ and $\sum_j w(i, j) = 1$ [110].

Moreover, an Otsu threshold and median filter are applied to remove noise to eliminate black dots inside the retina, a technique used in the works of Liu, Lemaitre, and Albarrack [98], [65], [96].

Specifically, each B-scan was denoised using NLM, then thresholded using Otsu's method to convert images to a binary format and detect different retina layers, followed by a median filter to smooth the image and reduce the effect of small variations as shown in Figure 47. The evaluation of each feature extraction and classifier model was performed several times to assess the impact of increasing data preprocessing complexity on the results.

Alternative techniques used in other works, such as cropping, could be used for preprocessing B-scan images. However, given the curved structure of the retina, indiscriminate cropping may not be a suitable approach, as it risks removing significant and informative data, and may lead to a suboptimal outcome. Furthermore, given that the studies may not involve the same diseases, adopting the same preprocessing procedure across studies may impact the accuracy and comparability of the results.

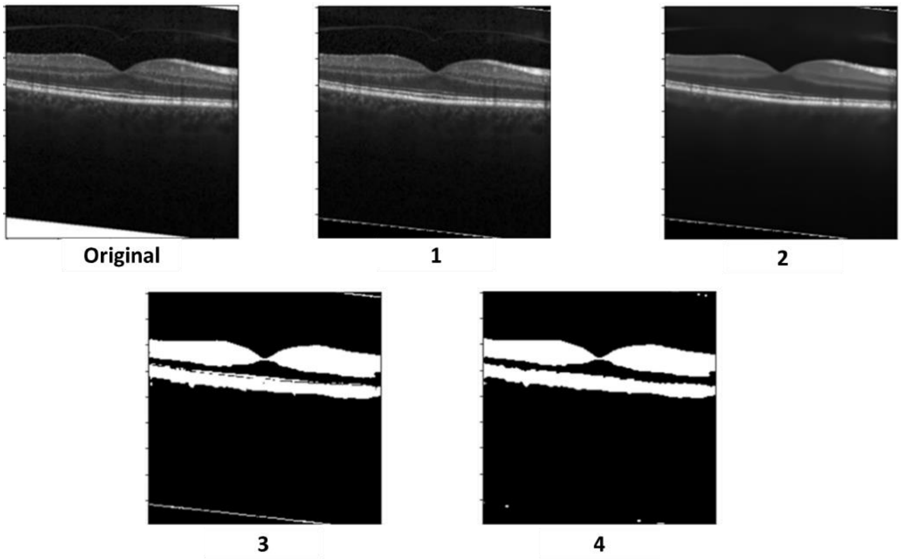


Figure 47: Preprocessing steps: 1- White boarder filled with black pixel; 2-Non-Local Means Filter; 3- Otsu Threshold; 4-Median Filter.

Feature Extraction

The process of extracting these features is depicted in Figure 48, which illustrates the steps involved in the feature extraction pipeline. During the feature extraction process, different parameters are experimented to obtain a diverse range of feature representations. These variations in parameters allows to capture different aspects and characteristics of the input images, enhancing the discriminative power of the extracted features.

Once the features are extracted, they are fed into a SVM classification model. The SVM model analyzes the extracted features and calculates the probability of an input image belonging to a specific class.

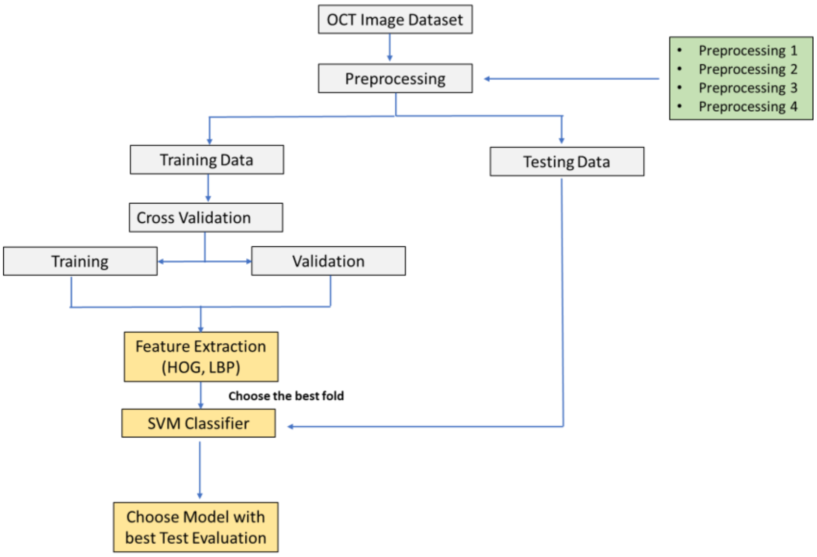


Figure 48 Working procedure with handcrafted feature extraction.

Histogram of Oriented Gradient (HOG)

In this study, HOG features were extracted from images with different cell sizes ([8 × 8], [16 × 16], and [32 × 32]), different cells per block ([2 × 2], [4 × 4]), and 9 orientation histogram bins. The block normalization method used was L2. Table 9: Hyperparameters used - HOG. summarizes the hyperparameters used.

Table 9: Hyperparameters used - HOG.

Method	Hyperparameter	Value
HOG	Number of orientations	9
	Pixels per cell	[16 × 16] and [32 × 32]
	Cells per block	[2 × 2], [4 × 4]

Local Binary Pattern (LBP)

In this study, LBP features were extracted from images using different sampling points (8, 16, and 24) and radius values (1, 2, and 3), respectively, and the image was segmented into varying numbers of regions to achieve optimal results. Table 10 summarizes the hyperparameters used.

Table 10: Hyperparameters used - LBP.

Method	Hyperparameter	Value
LBP	Number of Points	8,16,24
	Radius	1,2,3
	Method	Uniform

Classifier

To classify extracted features, a Support Vector Machines is used. The SVM has been recognized as a powerful tool for the classification of complex datasets with small to medium size [111].

A one-versus-one approach, with Radial Basis Function (RBF) kernel, was implemented to classify the images. The one-vs-one (ovo) strategy for multi-class classification involves training a binary classifier for each pair of classes. This is different from the one-vs-all (ova) method, which trains binary classifiers to distinguish one class from all other classes and requires fewer classifiers compared to one-vs-one approach. Additionally, comparisons have demonstrated that the ovo technique is superior for training SVMs [112].

SVMs with RBF kernel have two important hyperparameters, C and gamma. Careful selection of C and gamma is important for optimal SVM performance, as they can have a significant impact on the performance and require hyperparameter tuning.

Cross-Validation

To improve the SVM model's generalization and avoid bias and overfitting, a 5-GroupFold cross-validation technique was used, with each group representing a patient ID. The training data was divided into 5 parts, and cross-validation was performed 5 times, with each group used once as the validation set and the remaining 4 as the training set. The model's robustness and reliability were evaluated by reporting train and validation scores obtained from best model in cross-validation.

Choosing the best model from cross-validation is preferred because it represents the highest-performing option, ensuring better accuracy and reliability. It accounts for variations in the data and helps identify the most suitable settings, resulting in improved performance. Additionally, a grid search algorithm was used to explore the hyperparameters of the SVM, with a parameter grid defined using logarithmic ranges for the regularization parameter C and gamma.

Evaluation metrics

All the models were evaluated based on several metrics, including precision, recall, F-beta score (with beta=2.0), and accuracy. The goal was to select the model that performed the best according to these metrics on the test dataset. To ensure reliable results, each model went through a 5-fold cross-validation process. After the cross-validation, the model that demonstrated the highest performance on the validation metrics was chosen for further assessment on the independent test dataset.

6.3 Deep Learning methods

In deep learning approaches, this study aimed to compare the performance of transfer learning architectures with a novel main model. Specifically, three CNN architectures were selected for evaluation in the context of this problem: VGG 16, ResNet50V2 - these architectures were chosen based on their demonstrated effectiveness in prior research studies.

Transfer learning offers a notable advantage by facilitating the training of a CNN architecture using pre-trained weights rather than initializing them from scratch. In this study, the selected models had been previously trained on the ImageNet dataset, a large dataset with millions of images of different categories that is widely used in field of ML. The objective of employing transfer learning in this study is to adapt the original transfer learning methods in order to be implemented in OCT image classification tasks. There are two main approaches to apply transfer learning [71]:

> **Feature Extraction**: pre-trained models that have been trained on a large dataset are directly used for the classification process by removing the classification layer. The classification layer is removed from the pre-trained model, and the remaining network is treated as a feature extractor. Additional classifiers can then be added to this feature extractor model to perform classification for the new task.
>
> **Fine-Tuning**: the pre-trained model is further customized by retraining specific layers. This can involve unfreezing certain layers or adjusting the weights of the entire network. Fine-tuning allows for the adaptation of the pre-trained model to better suit the characteristics of the new task or dataset.

Once an appropriate transfer learning method is selected, the initial step is to identify a suitable performance metric that effectively evaluates the model's performance for the given task. This metric allows for an estimation of the model's capabilities. The validation set plays a crucial role in assessing the generalization abilities of the transfer learning model after fine-tuning. By utilizing the chosen performance measure, it becomes possible to compare and evaluate different transfer learning architectures and parameter settings. The objective is to identify the configuration that exhibits the best performance on the validation data, indicating its potential to deliver good results on an independent test set. By carefully selecting and optimizing hyperparameters, such as learning rate, regularization strength, and network architecture, it is possible to improve the model's ability to generalize, reduce overfitting, and achieve better results on unseen data.

The methodology employed in this approach for transfer learning comprises two main steps. In the first step, an iterative process is undertaken to select the optimal approach. Various methods, inspired by prior research, are introduced when the performance of the previous model on the original dataset is considered unsatisfactory. The objective is to compare and train different ML models to determine the most suitable one for the specific task.

The second step involves utilizing the best model obtained from the previous step. This model is subjected to testing in multiple configurations, where data augmentation techniques are applied, along with different preprocessing methods used in traditional approaches. Each model is subjected to six training configurations, enabling a comparison between the outcomes achieved with and without data augmentation and with three distinct preprocessing methodologies: no data preprocessing, as well as two preprocessing approaches that demonstrated superior performance in traditional methods. Each pixel is divided by 255 resulting in values that range from 0 to 1. This normalization technique helps the model to converge faster during training and makes it less sensitive to differences in pixel intensity across images. Through this procedure, the potential of different data preparation techniques is explored, aiming to improve the training process and achieve optimal model performance.

An additional objective was to develop a custom model and evaluate its performance relative to the transfer learning models. Figure 49 depicts the flowchart illustrating the process for each transfer learning model. Furthermore, Figure 50 portrays the comprehensive flowchart encompassing all deep learning methods, which incorporates the proposed custom model as well.

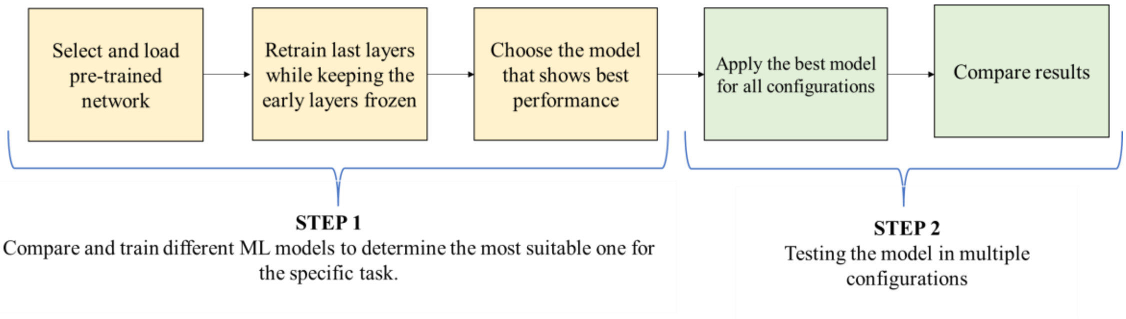

Select and load pre-trained network
Retrain last layers while keeping the early layers frozen
Choose the model that shows best performance
Apply the best model for all configurations
Compare results
STEP 1
Compare and train different ML models to determine the most suitable one for the specific task.
STEP 2
Testing the model in multiple configurations

Figure 49 : Process Diagram for each transfer learning model.

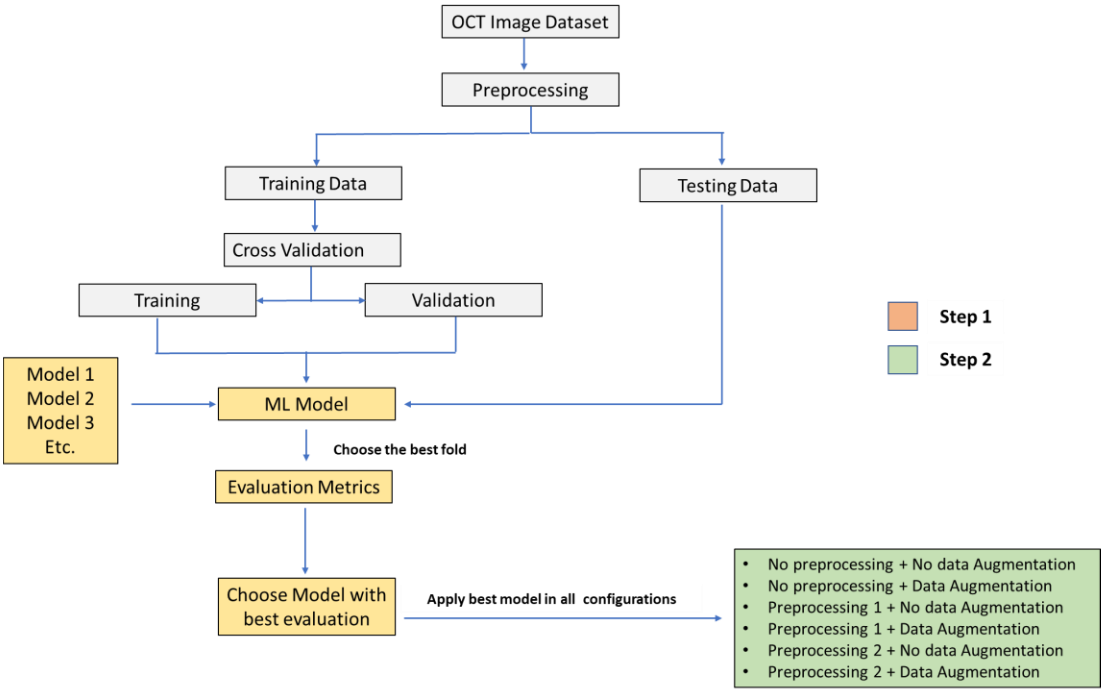

OCT Image Dataset
Preprocessing
Training Data
Testing Data
Cross Validation
Training
Validation
Model 1
Model 2
Model 3
Etc.
ML Model
Choose the best fold
Evaluation Metrics
Choose Model with best evaluation
Apply best model in all configurations
No preprocessing + No data Augmentation
No preprocessing + Data Augmentation
Preprocessing 1 + No data Augmentation
Preprocessing 1 + Data Augmentation
Preprocessing 2 + No data Augmentation
Preprocessing 2 + Data Augmentation
Step 1
Step 2

Figure 50: Working procedure of Deep Learning methods.

Models

VGG16

The VGG16 model introduced in the research paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition" by K. Simonyan and A. Zisserman from the University of Oxford, is a convolutional neural network architecture. It has garnered notable recognition for its impressive performance on the ImageNet dataset, which comprises millions of images distributed across 1000 distinct classes. Specifically, the VGG16 model achieves a remarkable accuracy of 92.7% on the ImageNet dataset, highlighting its efficacy in the task of large-scale image recognition [113].

The VGG16 architecture encompasses 138,355,752 parameters, featuring five convolutional blocks and three dense layers. The first two blocks consist of a pair of convolutional layers, while the remaining three blocks contain three convolutional layers each. A kernel size of 3x3 and a padding size of 1 is employed in all convolution layers to maintain the output size after each convolutional layer. After convolutional layers, each block has a max pooling layer to reduce the output size and a max pooling operation with a size of 2x2 and stride of 2 is applied.

Finally, after the convolutional blocks, the VGG16 architecture has three fully connected layers: the first two fully connected layers consist of 4096 neurons each, while the final fully connected layer comprises 1000 neurons. Subsequently, a softmax layer is incorporated.

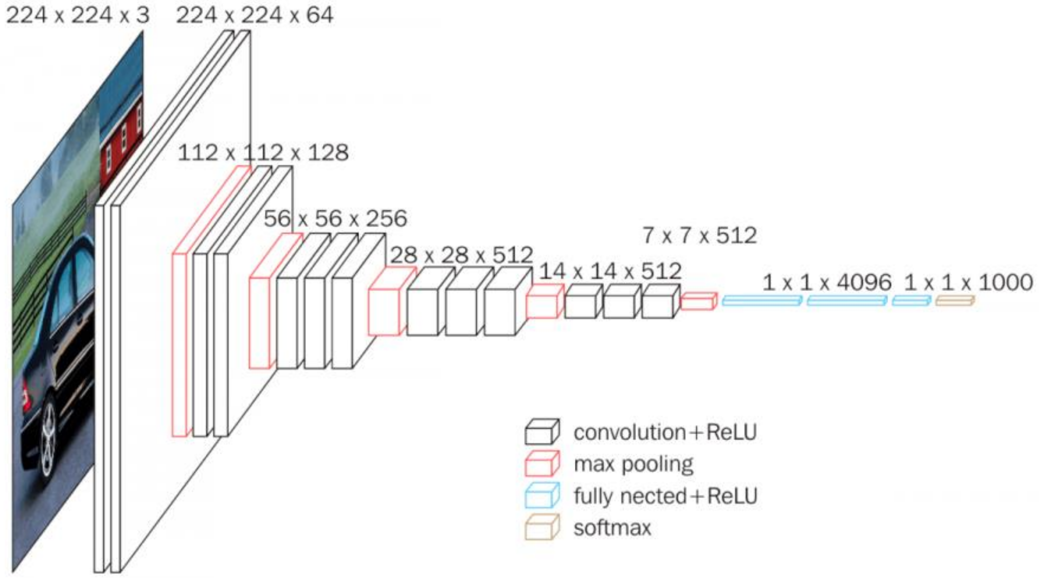


Figure 51: Vgg16 Architecture [114].

ResNet50V2

The ResNet (Residual Neural Network) architecture, introduced by He et al. [115] in 2015, encompasses multiple versions, including ResNet50, ResNet101, and ResNet152. This architecture has widely used for OCT image classification.

ResNet50V2, a variant of ResNet50 [116], has shown superior performance compared to ResNet50 and ResNet101 on the ImageNet dataset. With a total of 25 613 800 parameters, ResNet50V2 consists of 50 layers, organized into several residual blocks as shown in Figure 52.

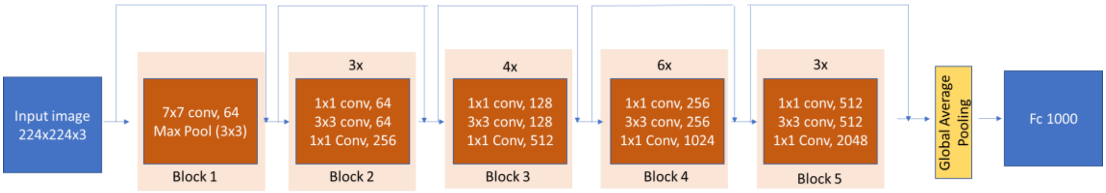


Figure 52: ResNet50V2 architecture diagram [117].

During the training of deeper networks, adding more layers to the network causes the accuracy to saturate or even decline abruptly. This degradation in accuracy is primarily caused by the vanishing gradient effect, which becomes more prominent in deeper networks.

The main objective of the ResNet architectures is to overcome the vanishing problem by allowing the network to effectively learn residual mappings. This is achieved by incorporating shortcut connections that enable the direct passage of input from one layer to deeper layers. These connections enhance the flow of information and mitigate the challenges associated with training very deep networks.

Each residual block comprises multiple convolutional layers and incorporates shortcut connections that allow the direct flow of information from earlier layers to subsequent layers. In Block diagram of ResNet50V2 architecture, the notation k × k, n in the convolutional layer block represents a filter of size k and n channels. The number on the bottom of the convolutional layer block represents the repetition of each unit. Furthermore, ResNet50V2 employs different types of shortcut connections compared to ResNet50 such as Batch Normalization and ReLU activation to the input before performing the convolution operation with the weight matrix. Figure 53 represents the difference between residual blocks at both models.

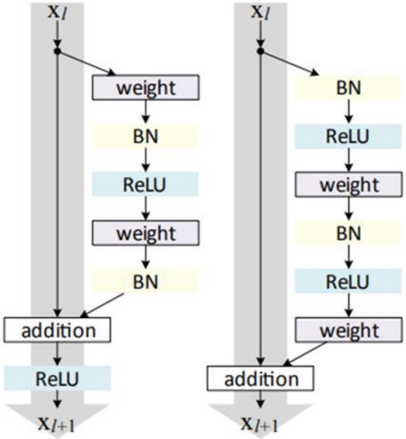


Figure 53: Residual Blocks at ResNet50 (left) and ResNet50V2 (right) [116].

Proposed Model

In the proposed model, there is a difference in image sizes compared to the transfer learning models. While the transfer learning models use images of size 224x224x3, this proposed model employs a smaller image size of 128x128x3. This discrepancy introduces a potential difference in performance and the level of detail captured by the models.

The computational requirements of the model can be significantly reduced using a smaller image size. Additionally, a smaller image size leads to a decrease in model complexity, as it reduces the number of parameters and overall complexity of the model. Despite the smaller image size, the proposed model aims to achieve comparable performance to the transfer learning methods. The decision to use a smaller image size involves a trade-off between capturing intricate details and achieving better computational efficiency or model simplicity. It is crucial to carefully evaluate this trade-off based on the specific requirements and constraints of your project.

In the proposed model, the feature extraction phase consists of four layers. The initial two layers employ a 3x3 filter with 32 feature maps and a max-pooling filter of size 2x2. A dropout layer with a rate of 0.2 follows these two layers. The second feature extraction layer involves a convolutional layer with 64 feature maps and a max-pooling filter of size 2x2. Again, a dropout layer with a rate of 0.2 is utilized. The final feature extraction layer consists of a convolutional layer with 128 feature maps and a max-pooling filter of size 2x2. Another dropout layer with a rate of 0.2 follows this layer. Subsequently, fully connected layers with 100 and 50 neurons (employing a ReLU activation function), along with a dropout layer of 0.2, are incorporated before the output layer. The output layer consists of four nodes.

Figure 54 represents the architecture of the model.

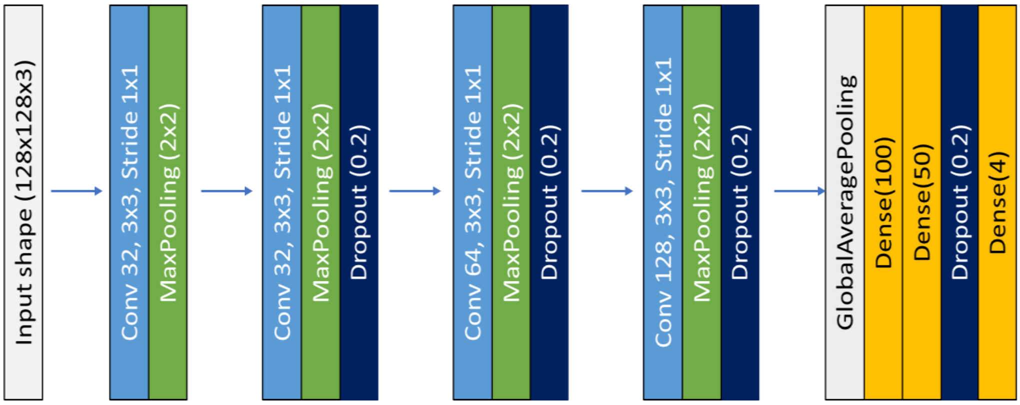


Figure 54: Proposed model architecture.

Transfer Learning Approach

Based on the dataset characteristics, the selection of a transfer learning strategy is influenced by factors such as dataset size and similarity to the source data. There are four distinct options available [118]:

1. In the case of a small dataset that closely resembles the source data, it is recommended to retrain only the output layer.
2. When dealing with a large dataset that exhibits high similarity to the source data, it is advisable to retrain all layers using the initial weights from the pretrained model.
3. If the dataset is small and less similar to the source data, it is preferable to retrain the last layers while keeping the early layers frozen.
4. In the scenario where the dataset is large and significantly differs from the source data, the model should be trained from scratch.

This scenario involves the utilization of fine-tuning techniques to refine the model. However, it is worth noting that in certain cases, feature extraction approaches inspired by relevant studies were employed instead.

Cross-Validation

In the context of model evaluation, a stratified group 5-Fold is utilized as an alternative approach to traditional cross-validation methods in order to be effective in imbalance dataset. This technique involves dividing the dataset into five distinct folds, while ensuring that the relative representation of different classes and groups remains proportionate across each fold. By incorporating stratification, the distribution of classes or groups from the original dataset is maintained consistently throughout the folds.

It is important to note that, in this scenario, the grouping is based on individual patient IDs. By treating patient IDs as distinct groups, the stratified group 5-Fold technique guarantees that images from same patient are not split into the training and validation for each fold. This approach is implemented to address the potential bias that may arise when patient-related data is distributed unevenly during model training and validation. Consequently, each fold encompasses a diverse selection of patients for training, validation, and testing images, thereby minimizing any potential bias and enhancing the fairness of the evaluation process.

Data Augmentation

To evaluate the impact of Data Augmentation on model's performance, a consistent set of augmentation techniques are applied during the training process. The transformations performed on the training images were randomized, introducing variability to the augmented dataset. Six distinct augmentation strategies are employed, including rotation, zoom range, shifting, and brightness adjustments. Table 11, shows the techniques used to train the proposed model. The chosen techniques were carefully evaluated to ensure their ability to create augmented images that maintain the characteristics and qualities of the original images, thereby enhancing the applicability and realism of the augmented dataset.

By employing Data Augmentation, all images in the training set experienced transformation during each epoch. Consequently, the model was exposed to diverse variations of the images in every epoch. Assuming *N* epochs were conducted, the total number of training image variations would be calculated as the product of the number of epochs and the initial number of training images [119].

Figure 55 shows an example of data augmentation applied to one image.

Table 11: Augmentation techniques and values.

Augmentation Techniques	Value
Zoom range	Between 90% and 110% of the original size
Rotation range	Between -10º and 10º
Width shift range	Between -5% and 5% of total width
Height shift range	Between -5% and 5% of total height
Fill mode	Nearest
Brightness range	1.4-2

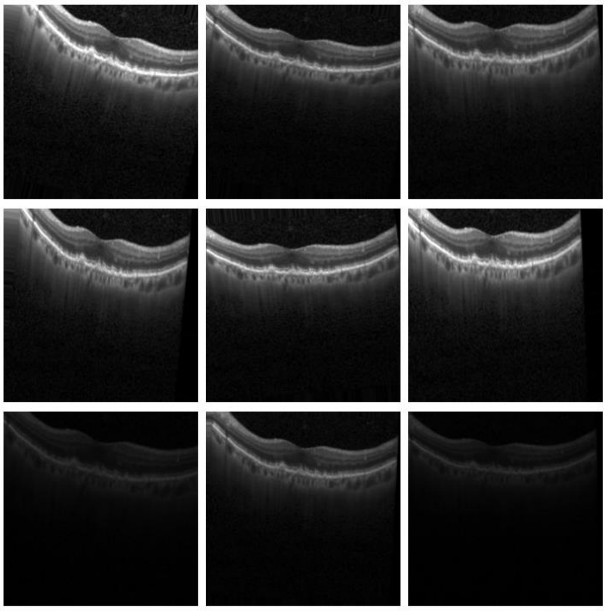


Figure 55: Example of Data Augmentation applied to an image from the Kermany Database [41].

Hyperparameters

All three models were compiled using categorical cross-entropy as the loss function and Adam as the optimizer. The models were trained using three different batch sizes: 16, 32, and 64. However, only the model with the best validation f-beta score (beta=2.0) was selected for further evaluation.

Typically, models have been compiled utilizing a standardized initial learning rate of 1e-4. Nonetheless, certain models have been subjected to training employing varying learning rate. To prevent overfitting, two callbacks were implemented. The first callback, *ReduceLROnPlateau*, was used to dynamically reduce the learning rate when the model's performance plateaued. This callback had a patience value of 3. During training, if the validation loss stops improving for a specified number of epochs, the learning rate is adjusted to prevent the model from getting fixed in a suboptimal solution. The learning rate factor determines the extent of the reduction applied to the learning rate. By reducing the learning rate, the model takes smaller steps during optimization, allowing it to potentially find a better, more optimal solution.

The minimum learning rate was set to 1e-6. The second callback employed was the early stop method: this callback was designed to halt the training process if the model's performance did not improve for a specified number of epochs. In this case was used an early stopping of 7 epochs.

Additionally, some models in the process utilized a weight decay of 0.001 as a regularization technique to counter overfitting. Table 12 summarizes the hyperparameters used.

Table 12: Hyperparameters used.

Hyperparameters	Value
Loss function	Categorical Cross-Entropy
Optimizer	Adam
Max epochs	30
Early Stopping	7
L2 regularization	0.001
Initial Learning rate	0.0001
Minimum Learning Rate	0.000001
LR factor	0.2
Batch size used	16,32,64

Weighted loss function

Due to the imbalanced nature of the dataset, algorithms often exhibit bias towards the majority values, resulting in poor performance when it comes to predicting the minority values. This disparity in class frequencies significantly impacts the overall predictability of the model.

To address this issue, a modification can be made to the training algorithm by considering the distribution of the classes. This is achieved by assigning different weights to the majority and minority classes, influencing their classification during the training phase.

The objective is to penalize misclassifications made by the minority class by assigning a higher weight to it, while simultaneously reducing the weight for the majority class.

The calculation of these weights is based on the inverse proportion of the class frequencies – this is represented in equation 6.2. By assigning appropriate class weights, the model can assign greater importance to the minority class during training. This serves to mitigate bias and enables the model to better handle the minority classes, ultimately leading to more balanced and accurate predictions. The weighted loss function places greater emphasis on minimizing misclassifications in the minority class compared to the majority class. This adjustment guides the model to prioritize minimizing errors on the minority class, contributing to improved overall performance [120].

$$w_j = \frac{n_{images}}{n_{classes} * n_{images_j}} \tag{6.2}$$

Where, J represents the class; wj represents weight of each class, n_{images} is the total number of images in the dataset, $n_{classes}$ is the total number of unique classes in the target and n_{images_j} is the total number of images of the respective class.

By employing these class weights, equation 6.3 denotes a weighted loss function that incorporates cross entropy loss to address multiclass problems. In this context, t_i symbolizes the true label pertaining to class i, whereas p_i denotes the associated probability assigned to class i.

$$L_{CE}(t,p) = -\sum_{i=1}^{n} w_i * t_i \log(p_i)\,, \text{for } n \text{ classes} \tag{6.3}$$

Evaluation metrics

In all utilized models, the evaluation results include macro precision, macro recall, loss, macro F-beta score (beta=2.0), macro accuracy and weighted accuracy (which is equivalent to accuracy in test set because of balanced dataset). The models using the same pre-trained architecture are compared, and the model that achieved the best results in the test metrics is chosen. Each model undergoes a 5-fold cross-validation process, and the model exhibiting the highest performance based on the validation metrics is chosen for further assessment on the test dataset.

Weighted accuracy considers class frequencies and utilizes weights based on the inverse proportion of the class frequencies. In the context of imbalanced classification problems, both macro average and weighted average are employed as evaluation metrics, providing different viewpoints on model performance. When the macro average and weighted average exhibit similar values, it implies consistent performance across all classes, regardless of imbalances. A notable discrepancy with the macro average being significantly lower than the weighted average suggests potential challenges for the model in effectively predicting minority classes, while performing relatively better on majority classes. Conversely, a lower weighted average compared to the macro average indicates relatively poorer performance on minority classes when compared to majority classes.

For each model, a graph is generated to visualize the loss and F-beta score curves: the F-beta score was considered a crucial metric in this work and it was used instead of accuracy not only due to the dataset's imbalance. Furthermore, it also possesses the unique property of balancing precision and recall, with a particular emphasis on false negatives. This balance is essential in medical image analysis, as both false positives and false negatives can have significant implications in individuals' health. This metric aligns with the real-world impact of correctly identifying positive cases in OCT image classification. Detecting diseases or abnormalities accurately is of utmost importance, as it influences subsequent medical decisions and treatments.

6.4 Framework

The algorithm development in this work requires a computational tool to construct the necessary deep learning models and traditional methods. Python provides a wide range of libraries that are useful for this purpose and for that reason this programming language is used. For traditional feature extraction methodologies, the sklearn library is used.

In deep learning methods the Keras API is employed for transfer learning, facilitating the loading and preparation of pre-trained models. To optimize training time for the models, Tensorflow's GPU version is utilized, which effectively utilizes Nvidia's CUDA and cuDNN libraries for GPU acceleration.

Additionally, essential libraries such as Numpy and Matplotlib are implemented to perform crucial tasks, such as numerical computations and data visualization.

All the code for implementing Deep CNN and Handcrafted for feature extraction is available in this Github repository: https://github.com/tolitei/OCT-Image-Classification/tree/main.

7 Results and Discussion

7.1 Traditional Methods

Various preprocessing techniques were employed to determine the optimal approach, in conjunction with two distinct feature extraction methodologies. This study encompasses a wide range of feature representations and preprocessing procedures, aimed at evaluating the effects of different preprocessing techniques and two feature descriptors on the experimental outcomes. Each preprocessing technique involved an additional step: Preprocessing 1 only involved converting white border pixels to black; preprocessing 2 employed NLM, preprocessing 3 used Otsu thresholding, and preprocessing 4 included a median filter.

Histogram of Oriented Gradient (HOG)

After testing several configurations, the best results using HOG as a feature descriptor were achieved by setting the number of pixels per cell to 32, using 9 bin orientations and a 2x2 cell per block. In addition, preprocessing method number 1 provided the best results. Although smaller pixel per cell values is generally preferred for capturing more local information, in the case of this OCT dataset, a larger pixel per cell value of 32 provided better results. This may be due to the fact that using a larger pixel per cell can capture more global information about the image, which could be more effective in detecting relevant features and patterns in the images.

Overall, the differences in performance between the training, validation, and testing sets could be due to overfitting, which occurs when the model learns the training data too well and is not able to generalize to unseen data. In this case, the model may be memorizing the features of the training data but is not able to generalize to the validation and testing sets. Table 13 summarizes the results obtained and Table 14 demonstrate test results obtained for each class. Additionally, Figure 56 shows the confusion matrix obtained in the test results - the global accuracy obtained was 73.20%. Appendix A summarized the values obtained for each preprocessing using the same hyperparameters.

Table 13: Best results obtained with the best hyperparameters and preprocessing 1 – HOG feature.

Metrics	Train	Validation	Test
Precision	95.92%	77.28%	74.84%
Recall	95.91%	77.26%	73.20%
Fbeta-Score	95.91%	77.26%	72.74%
Accuracy	95.91%	77.26%	73.20%

Table 14: Test Results obtained with the best hyperparameters and preprocessing 1 – HOG feature.

Feature Extraction	**Metrics**	**Disease**				
		CNV	DME	DRUSEN	NORMAL	Average
HOG	Precision	74.02%	63.63%	81.64%	80.08%	74.84%
	Recall	83.20%	84.00%	51.60%	74.00%	73.20%
	Fbeta-Score	81.18%	78.94%	55.69%	75.17%	72.74%
	Accuracy	83.20%	84.00%	51.60%	74.00%	73.20%

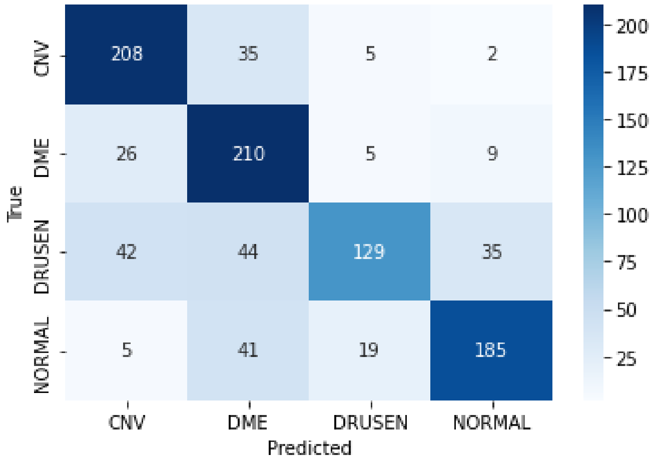


Figure 56: Test Results using HOG - Confusion Matrix.

Local Binary Pattern (LBP)

Based on the results of the experiment, it was found that using LBP as a feature descriptor resulted in inferior results compared to HOG. The best performing configuration of LBP used preprocessing number 2, a radius of 2, 16 points. Using radius of 2, the LBP operator considers a wider neighborhood around the central pixel. This inclusion of a larger spatial context allows for the capture of more global or larger-scale patterns in the image nonetheless it can result in a reduced sensitivity to local details.

The values of the training set were higher than both the validation and test sets. The results are summarized in Table 15, and show that the validation set achieved better results than the test set, which could indicate overfitting to the training set and Figure 57 shows the confusion matrix obtained in the test results - the global accuracy obtained was 54.6%. Additionally, Table 16 demonstrates test results obtained for each. In Appendix B the values obtained for each preprocessing using the same hyperparameters are summarized.

Table 15: Best results obtained with the best hyperparameters and preprocessing 2- LBP

Metrics	**Train**	**Validation**	**Test**
Precision	82.03%	60.47%	57.67%
Recall	82.00%	60.34%	54.60%
Fbeta-Score	81.99%	60.28%	52.64%
Accuracy	82.00%	60.34%	54.60%

Table 16: Test Results obtained with the best hyperparameters and preprocessing 2 – LBP feature.

Feature Extraction	Metrics	Disease				
		CNV	DME	DRUSEN	NORMAL	Average
LBP	Precision	64.93%	49.85%	63.04%	52.85%	57.67%
	Recall	40.00%	70.00%	23.20%	85.20%	54.60%
	Fbeta-Score	43.32%	64.76%	26.55%	75.90%	52.64%
	Accuracy	40.00%	70.00%	23.20%	85.20%	54.60%

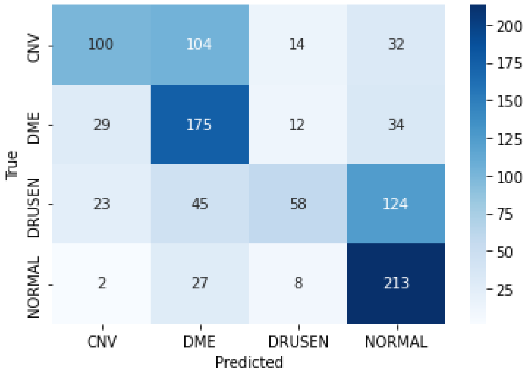


Figure 57: Test Results using LBP - Confusion Matrix.

Result Analysis

Overall, the experiment highlights the importance of selecting appropriate feature descriptors and hyperparameters for the specific task at hand. While Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) have proven effective in other types of image analysis, they did not yield satisfactory performance in this particular case. Table 17 presents a concise summary of the best results obtained for image classification using each feature extractor in this study.

A comparative analysis of the best results obtained from the two feature extraction methods clearly demonstrates that HOG outperforms LBP in terms of performance. By employing HOG feature extraction and examining the corresponding confusion matrix in Figure 56 and Table 14, it becomes evident that only the classes CNV and DME exhibited reasonable results, with accuracies of 83.20% and 84%, respectively. However, these accuracies remain relatively low. On the other hand, the classes Normal and Drusen performed poorly, with Drusen accounting for 121 misclassifications out of 268. Examining the confusion matrix in Figure 57 and Table 16, it becomes evident that LBP is not a reliable feature extractor, as it only generated correct predictions in 546 out of 1000 cases. Only Normal and DME image class has an accuracy bigger than 50%.

Table 17: Best test results of each feature descriptor.

Feature Extraction	Best Preprocessing	Metrics	Disease				
			CNV	DME	DRUSEN	NORMAL	Average
HOG	1	Precision	74.02%	63.63%	81.64%	80.08%	74.84%
		Recall	83.20%	84.00%	51.60%	74.00%	73.20%
		Fbeta-Score	81.18%	78.94%	55.69%	75.17%	72.74%
		Accuracy	83.20%	84.00%	51.60%	74.00%	73.20%
LBP	2	Precision	64.93%	49.85%	63.04%	52.85%	57.67%
		Recall	40.00%	70.00%	23.20%	85.20%	54.60%
		Fbeta-Score	43.32%	64.76%	26.55%	75.90%	52.64%
		Accuracy	40.00%	70.00%	23.20%	85.20%	54.60%

Both approaches demonstrate their impracticality for real-world application in the medical domain due to their tendency for generating a substantial number of misclassifications.

Comparing these traditional methods to other studies proves challenging due to the prevalent use of private databases and the reliance on relatively small image sets, often 100 times smaller than those utilized in this study. Such discrepancies in dataset sizes can make fair comparisons difficult. Hence, this study compares its results with two similar approaches: Hasan et al. [46] that employed the same database and Nugroho [108] that utilized a larger set of images however it used images from the same four eye retina diseases.

Table 18 provides a comprehensive summary of the test results obtained in this study, allowing for a meaningful comparison with the results reported in the relevant literature.

Table 18: Comparison of the proposed model with other state-of-the-art methods.

Author	Feature Descriptor	Metrics			
		Precison	**Recall**	**F1-Score**	**Accuracy**
Proposed Study	HOG	**74.84%**	**73.20%**	**72.73%**	**73.20%**
	LBP	**57.67%**	**54.60%**	**51.72%**	**54.60%**
Nugroho [108]	HOG	56.00%	50.00%	37.00%	50.10%
	LBP	23.00%	42.00%	29.00%	42.35%
Hasan et al. [46]	HOG	79.66%	79.69%	79.64%	79.69%
	LBP	58.17%	57.60%	57.65%	57.60%

Consistent with previous research, the HOG feature extraction method outperformed the LBP feature extraction. Furthermore, similar outcomes were observed for both LBP and HOG features when compared to Hasan et al. [46] study that employed the same four disease classes.

Although the effects of preprocessing varied across the experiments, the best test results were consistently achieved using Preprocessing 1 and 2 for each feature. Preprocessing 3 and 4 were less effective in producing satisfactory outcomes. Specifically, utilizing the Otsu Thresholding and Median Filter methods did not yield superior results.

It is worth noting that these results are within the expected range and fall short of the performance achieved by deep learning methods, where the feature extraction is learned by the algorithm itself. Processing methods 1 and 2 yield superior outcomes and preprocessing 3 and 4 removes details from the image that are important for its classification. This lack of information is fatal for the correct classification of the models in question.

Consequently, when employing more sophisticated deep learning models, a substantial quantity of data and information becomes imperative. This type of algorithm benefits a lot from the use of detailed images, so preprocessing 1 and 2 are undoubtedly the most suitable choices for the subsequent phases of this study since the others remove a lot of important information and details from the image.

7.2 Deep Learning methods

Appendix C, D and E includes all the graphs and results obtained for the three architectures implemented. The most optimal model selected in test metrics in each model is deployed in six distinct configurations. The outcomes are compared across all deep learning methods. The two preprocessing methods, denoted as number 1 and number 2, were chosen since they performed better in traditional methods.

VGG16

In this pre-trained convolutional neural network, seven different ways are used to evaluate the original image database. In Table 19 it is possible to see results of all models used.

Although the employed methodology primarily focuses on fine-tuning the latter layers while keeping the initial layers unchanged to enable the model to effectively learn and adapt to the variations in feature space between ImageNet images and retinal OCT images, the initial models implemented closely resemble the approach adopted by Kermany et al. [41] and Hwang et al. [47]. In their works, solely the weights of the fc layers were randomly initialized, while the convolutional layers were frozen and employed as fixed feature extractors.

The first three models in this study followed this approach, with the only variations being the number of fc layers and the implementation of regularization techniques. However, what remained consistent among these models was the application of a learning rate of 0.001 instead of the initial learning rate specified in Table 12.

In the first approach, the last fc layers utilized the same number of neurons as the original fc layers (4096), while only the last fully connected layer was modified to accommodate the four output classes of interest. Figure 58 shows the architecture implemented.

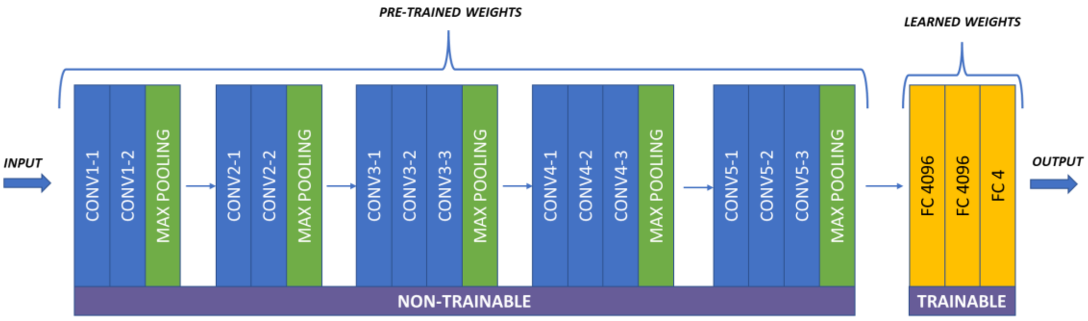


Figure 58: Transfer Learning with VGG16: Architecture of Model 1.

The results obtained in model 1 revealed a significant disparity between the validation and train values. All evaluated metrics exhibited a minimum difference of 12.9%, and the validation loss function displayed an increasing trend in the final epochs, as demonstrated in Appendix C. This difference between the validation and training metrics indicated the presence of overfitting. Consequently, it became apparent that further refinements were necessary to address this issue and optimize the model's performance. One approach to achieve this is to employ regularization techniques to mitigate overfitting.

In the subsequent approaches, it is explored the inclusion of dropout layers in the model. Figure 59 shows the architecture implemented. This choice is motivated by the work of Li [101], which incorporated dropout layers between fc layers. Model 2 introduces an additional dropout layer between the first and second fc layers. Several dropout rates were experimented with, and the most favourable outcomes were observed when the dropout layers had a dropout rate of 0.15.

Model 2 demonstrates comparable performance in terms of the f-beta score in both training and validation curves. The consistent pattern observed in the training and validation f-beta curves indicates an effective learning and adaptation improvement despite the gap between them. However, it is important to note that the training dataset itself lacked sufficient information for the model to effectively learn the problem. This limitation is evident from the elevated values observed in both the training and validation loss curves. These high loss values indicate that the model struggled to accurately fit the given data, implying a lack of proper generalization.

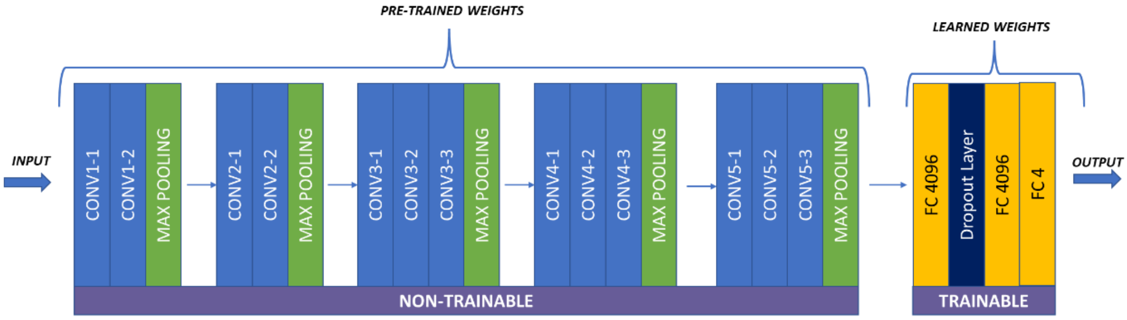


Figure 59: Transfer Learning with VGG16: Architecture of Model 2.

One significant drawback of Model 1 and 2 is its requirement to train a large number of parameters, approximately 120 million. To address this issue, alternative approaches were explored to achieve comparable performance while reducing the number of parameters. Model 3 consists of one dense layer with 512 neurons, with a dropout rate of 0.15 applied between them, followed by a dense layer with 100 neurons, and a final layer with 4 neurons. Figure 60 shows the architecture implemented. The total number of trainable parameters in Model 3 was significantly reduced to 12,897,272 parameters and exhibits similar trends in the training and validation curves.

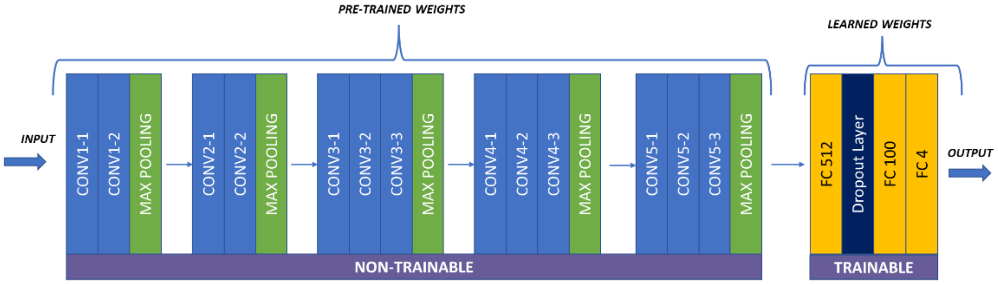


Figure 60: Transfer Learning with VGG16: Architecture of Model 3.

The evaluation of the initial three models reveals that, despite demonstrating good performance on the test set, their outcomes cannot be deemed satisfactory. There was a progressive improvement from model 1 to 3, so the changes applied improved the results. As for the test consistently having better results than the validation in these first 3 models, a possible reason is the fact of the original split: while for each cross validation there are about 80% of the 108 309 images are used for training and 20% for validation while the training images are only 1000 images.

A closer examination of the validation values suggests the potential for improved generalization. Notably, the presence of overfitting is evident from the loss and F-beta validation curves. This indicates that the models encountered challenges in effectively generalizing to new, unseen data.

To address this limitation, subsequent models were developed, incorporating adjustments to the architecture. Specifically, the last convolutional layers were unfrozen, allowing for increased flexibility in learning task-specific features directly from the input data. By unfreezing more layers, these models aimed to enhance the network's capacity to capture intricate patterns and improve its ability to generalize effectively.

In Model 4, a modification was made by unfreezing the last convolutional layer, introducing a global average pooling and keep the fc layers the same as in Model 3. This adjustment, represented in Figure 61, aligned better with the established methodology. The purpose of global average pooling is to convert multi-dimensional feature maps into 1D-feature maps by averaging the values across spatial dimensions. This technique helps in reducing the number of parameters and mitigating overfitting risks.

Adding a global average pooling layer after max pooling provides several benefits. Max-pooling enhances translation invariance by allowing the model to recognize features regardless of their exact location. The global average pooling layer acts as a form of regularization by emphasizing discriminative features across the entire feature map, rather than relying solely on specific spatial locations. This approach reduces dimensionality, improving computational efficiency and making it more adaptable to variations in input data and enhancing generalization capabilities.

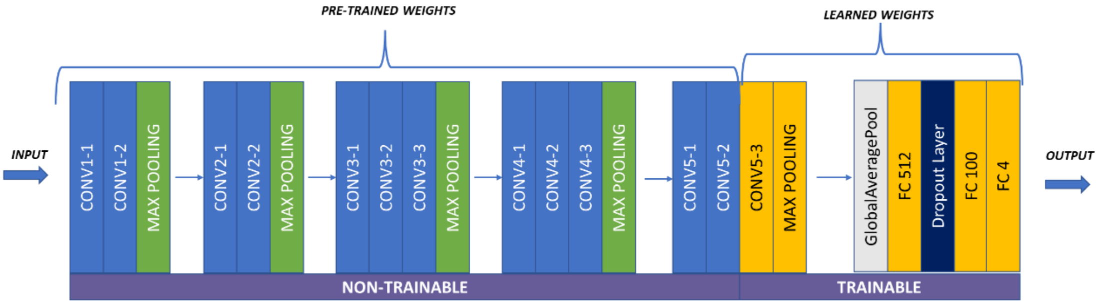


Figure 61: Transfer Learning with VGG16: Architecture of Model 4 and 5.

The performance improvement was evident through the similarity observed between the training and validation curves, as well as the attainment of test metrics surpassing 91%. Specifically, the training and validation metrics ranged between 97% and 98%, while the test metrics ranged between 91% and 94%.

Model 5, Model 6, and Model 7 are built upon the same architecture as Model 4, with each model incorporating specific adjustments. In Model 5, the learning rate is modified to 0.0001. When unfreezing more layers, it is crucial to adjust the learning rate accordingly. Reducing the learning rate allows for a more cautious and controlled update of the weights in the newly unfrozen layers. This slower update helps the model to adapt to the new task while preserving the previously learned knowledge. it enables a more stable optimization process by preventing drastic changes to the model's weights.

In Model 6, the last two convolutional layers are frozen, while Model 7 goes a step further by freezing the last three convolutional layers - this approach leads to a larger number of trainable parameters compared to Model 5, as fewer layers are eligible for weight updates.

Figure 62 shows the architecture implemented in those 2 models.

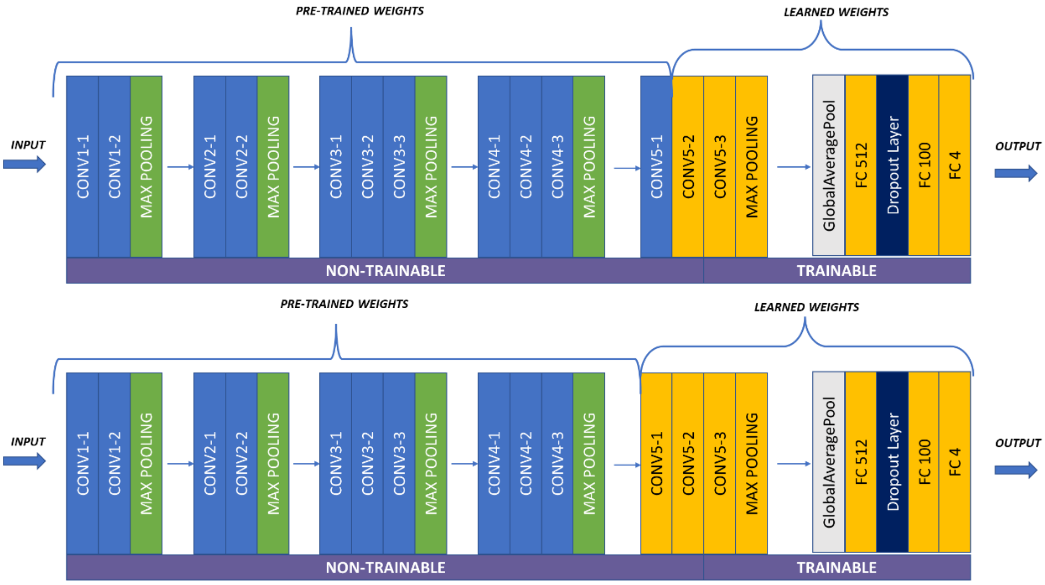


Figure 62: Transfer Learning with VGG16: Architecture of Model 6 and 7.

In general, the test metrics across Model 4, Model 5, Model 6, and Model 7 exhibited remarkable similarity, with Model 4 demonstrating a slight improvement. All models displayed metrics ranging from 93% to 94%. It is noteworthy that Model 4, despite having a lower number of trainable parameters compared to Model 6 and Model 7, along with a higher learning rate, achieved the best overall results. In the initial epochs, the training performance quickly becomes high. This suggests that even with a limited number of epochs or without extensive training, unlocking the convolutional part of the model enables a much better fit to the given data. Specifically, Model 4 yielded a precision of 93.92%, an f-beta score of 93.38%, and a recall and accuracy of 93.50%.

Although the metrics achieved by Model 4 align closely with those of the other models, the advantage of having fewer trainable parameters contributes to enhanced computational efficiency and resource allocation. Therefore, Model 4 emerges as the preferred choice due to its favorable performance metrics and optimized parameter utilization, despite being approximate to the other models in terms of overall results.

Table 19: Results obtained from all models.

MODEL	TRAINABLE PARAMETERS	METRICS	TRAIN	VALIDATION	TEST
1 Weights of the last three fully connected layers were randomly initialized. Initial lr= 0.001.	119,562,244	Precision	95.38%	82.48%	89.68%
		Recall	98.77%	85.42%	88.49%
		F-Beta score	98.01%	84.73%	88.19%
		Accuracy	98.13%	84.32%	88.49%
		Weight Accuracy	99.39%	80.17%	88.49%
		Loss	0.0567	0.5294	0.8041
2 Random weights on all fc layers and 1 dropout of 15% between first and second fc layer. Initial lr= 0.001.	119,562,244	Precision	91.13%	82.26%	91.65%
		Recall	93.97%	86.11%	90.70%
		F-Beta score	93.24%	85.00%	90.52%
		Accuracy	94.25%	86.99%	90.70%
		Weight Accuracy	93.08%	82.65%	90.70%
		Loss	0.2171	0.3148	0.2837
3 Modified fc layers architecture: layer of 512, dropout, layer of 100, layer of 4. Initial lr= 0.001.	12,897,272	Precision	89.11%	81.65%	92.29%
		Recall	94.02%	86.22%	91.80%
		F-Beta score	92.52%	84.56%	91.71%
		Accuracy	94.21%	88.08%	91.80%
		Weight Accuracy	94.10%	83.87%	91.80%
		Loss	0.2293	0.3200	0.2585
4 Unfroze the last convolutional layer while maintaining the same architecture as in model 3.	2,674,168	Precision	**96.15%**	**95.57%**	**93.92%**
		Recall	**98.46%**	**97.37%**	**93.50%**
		F-Beta score	**97.96%**	**96.98%**	**93.38%**
		Accuracy	**98.38%**	**97.91%**	**93.50%**
		Weight Accuracy	**98.55%**	**96.82%**	**93.50%**
		Loss	**0.0443**	**0.0640**	**0.3893**

MODEL	TRAINABLE PARAMETERS	METRICS	TRAIN	VALIDATION	TEST
5 Same procedure as in model 4 with initial lr= 0.0001.	2,674,168	Precision	96.40%	90.73%	92.50%
		Recall	98.81%	91.44%	90.70%
		F-Beta score	98.28%	91.28%	90.42%
		Accuracy	98.82%	91.12%	90.70%
		Weight Accuracy	99.08%	89.86%	90.70%
		Loss	0.0473	0.1774	0.3430
6 Unfroze the last two convolutional layers while maintaining the same architecture as in model 3.	5,033,976	Precision	98.72%	94.73%	94.26%
		Recall	99.69%	93.40%	93.40%
		F-Beta score	99.49%	93.66%	93.23%
		Accuracy	99.54%	96.64%	93.40%
		Weight Accuracy	99.86%	93.87%	93.40%
		Loss	0.0123	0.2043	0.5871
7 Unfroze the last three convolutional layers while maintaining the same architecture as in model 3.	7,393,784	Precision	98.56%	95.93%	94.15%
		Recall	99.58%	95.36%	93.29%
		F-Beta score	99.36%	95.47%	93.10%
		Accuracy	99.46%	96.50%	93.29%
		Weight Accuracy	99.72%	95.08%	93.29%
		Loss	0.0153	0.1377	0.6058

Upon examining the six configurations, it is possible to see that using data augmentation consistently leads to better training results. The best outcome is achieved in the iteration where is used preprocessing 1 and is used data augmentation. It this iteration it is possible to obtain a f-beta (beta=2.0) score of 96.79%, precision of 96.88%, recall and accuracy of 96.80%.

All iterations perform well in three out of the four diseases, except for Drusen, where a significant number of images are incorrectly predicted as CNV. This could be due to the similarities between the structures in the OCT images of both conditions. Preprocessing method 1 proves effective in producing better results compared to using only normal images.

Table 20 summarizes the metric results for the six configurations and Figure 63 shows the f-beta curve and confusion matrix from the testing dataset comparing the true labels and predicted labels.

With preprocessing 1, the original images are only complete the white part of the image with black and makes the background of the image uniform. However, it does not remove information and the original image still has the same detail. The additional preprocessing only removes quality from the image and ends up having worse results.

Furthermore, in this model data augmentation leads to improved results by increasing the number of training images. With a larger and more diverse dataset, the model gains exposure to various variations, patterns, and representations presents in the original images.

This process compels the model to acquire more generalized and robust understandings of the data, thus mitigating overfitting.

Table 20: Test results of the 6 configurations applied to Model 4.

CONFIGURATION	METRICS	CNV	DME	DRUSEN	NORMAL	Average
1 No preprocessing	Precision	85.76%	96.82%	98.03%	95.05%	93.92%
	Recall	96.40%	97.60%	80.00%	100%	93.50%
	F-Beta score	94.06%	97.44%	83.05%	98.97%	93.38%
	Accuracy	96.40%	97.60%	80.00%	100%	93.50%
2 Preprocessing 1	Precision	75.46%	97.52%	98.84%	91.89%	91.89%
	Recall	98.40%	94.40%	68.40%	90.10%	90.10%
	F-Beta score	92.76%	95.00%	72.89%	89.78%	89.78%
	Accuracy	98.40%	94.40%	68.40%	90.10%	90.10%
3 Preprocessing 2	Precision	81.78%	98.35%	97.87%	92.50%	92.63%
	Recall	98.80%	95.60%	73.60%	98.80%	91.70%
	F-Beta score	94.85%	96.13%	77.44%	97.47%	91.47%
	Accuracy	98.80%	95.60%	73.60%	98.80%	91.70%
4 No preprocessing and data augmentation	Precision	95.21%	95.40%	95.38%	95.73%	95.73%
	Recall	95.60%	99.60%	88.40%	95.70%	95.70%
	F-Beta score	95.52%	98.73%	89.98%	95.66%	95.66%
	Accuracy	95.60%	99.60%	95.38%	95.70%	95.7%
5 Preprocessing 1 and data augmentation	Precision	**92.42%**	**97.61%**	**98.29%**	**99.20%**	**96.88%**
	Recall	**97.21%**	**98.40%**	**92.40%**	**99.20%**	**96.80%**
	F-Beta score	**96.21%**	**98.24%**	**93.52%**	**99.20%**	**96.79%**
	Accuracy	**97.21%**	**97.61%**	**98.29%**	**99.20%**	**96.80%**

CONFIGURATION	METRICS	CNV	DME	DRUSEN	NORMAL	Average
6 Preprocessing 2 and data augmentation	Precision	83.84%	97.24%	97.98%	96.48%	93.89%
	Recall	97.60%	98.80%	78.00%	98.80%	93.30%
	F-Beta score	94.50%	98.48%	81.31%	98.32%	93.15%
	Accuracy	97.60%	98.80%	78.00%	98.80%	93.30%

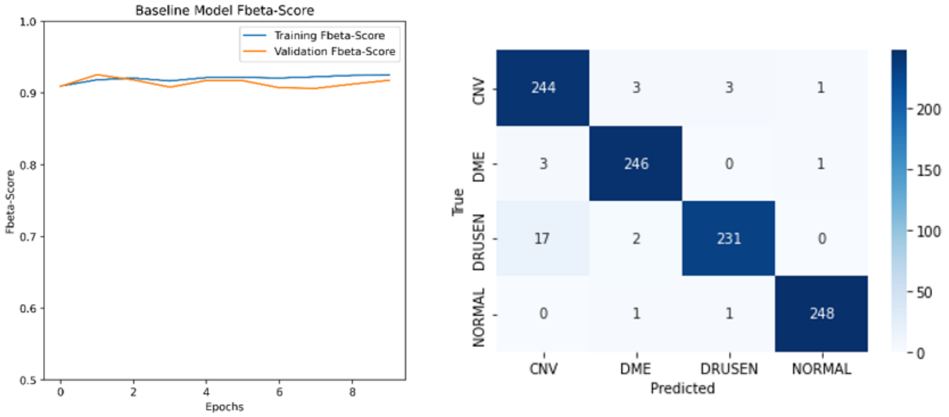


Figure 63: F-beta learning curve and Confusion matrix of best configuration using VGG16 (Data Augmentation and Preprocessing 2).

ResNet50V2

Using ResNet50V2 pre-trained CNN, four different ways are used to evaluate the original image database. In Table 21 it is possible to see results of all models used. In each approach, the parameters of the first four blocks were frozen, while the remaining layers were left unfrozen. For the VGG16 model, the initial attempts involved unfreezing only the fc layers. However, since there was no evidence of such an approach in the literature on ResNet architectures, this methodology was not implemented. Instead, in all models, the last block of ResNet50V2 was unfrozen, and modifications were made to the fc layers. Figure 64 shows the architecture implemented in all models in ResNet50V2.

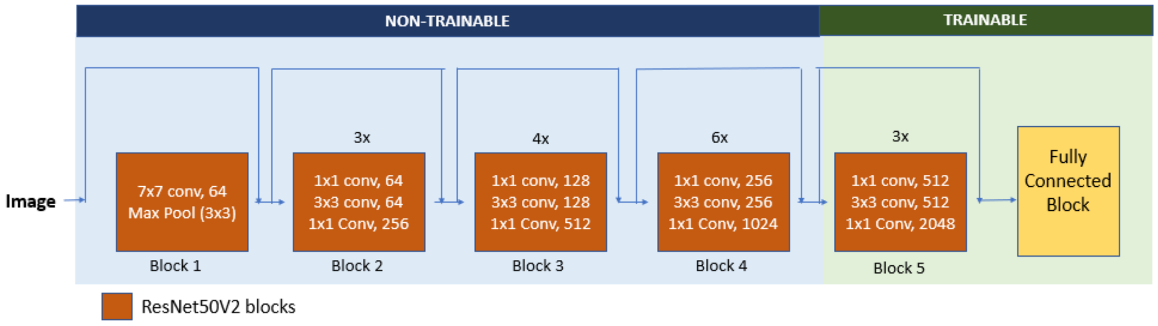


Figure 64: Fine-tuning methodology of all models in ResNet50V2 architecture.

The results from all models demonstrate consistently high metric scores, with training, validation, and testing all achieving scores greater than 96% in the best model determined through the implementation of a stratified 5-fold cross-validation approach. The observed high metric scores across all models provide evidence that the chosen approach of fine-tuning the last layers while keeping the early layers frozen is effective for this dataset. This approach proves to be preferable as it yields very good results across all models.

The difference in total training parameters among four models is not substantial different compared to VGG16 models. This is because in all models, only the fc layers were changed, while block 5 was unfrozen and used for fine-tuning the weights. Only, in model 4, the initial learning rate was adjusted to 5e-6, and learning rate factor was set to 0.5.

The learning curves of loss and f-beta scores, as depicted in Appendix D, exhibit a similar pattern across the first two models. The models stabilize quickly, with the learning curve of the best model in cross-validation already showing high values in the first epoch. In all models the training loss converge to values near zero and training f-beta score increase during training. Furthermore, the validation loss is consistently low, although slightly higher than the training loss. This trend is mirrored in the f-beta scores, with a similar pattern observed between the training and validation scores in the last epochs.

In Model 1, the architecture used was the same as the best model employed in VGG16, consisting of a layer with 512 neurons, a dropout layer, a layer with 100 neurons, and a final layer with 4 neurons. Figure 65 shows the architecture implemented.

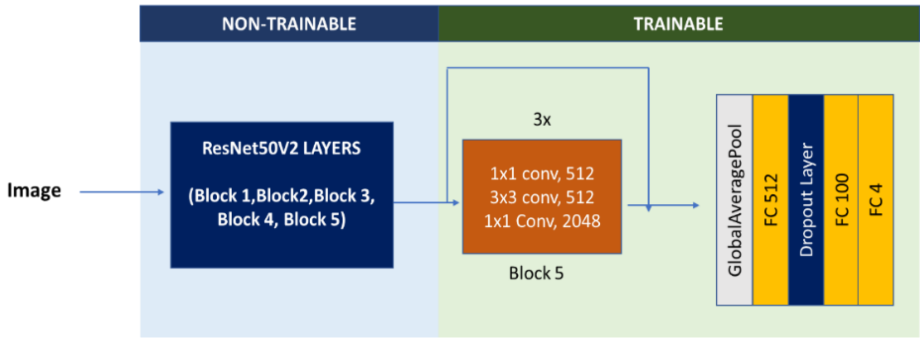


Figure 65: Transfer Learning with ResNet50V2: Architecture of Model 1.

Model 2, on the other hand, made a modification to the original ResNet50V2 model by replacing the fully connected layer with only 4 neurons. This simplified architecture aimed to reduce the complexity of the fully connected block in the model and it is an architecture inspired in [104]. Figure 66 shows the architecture implemented.

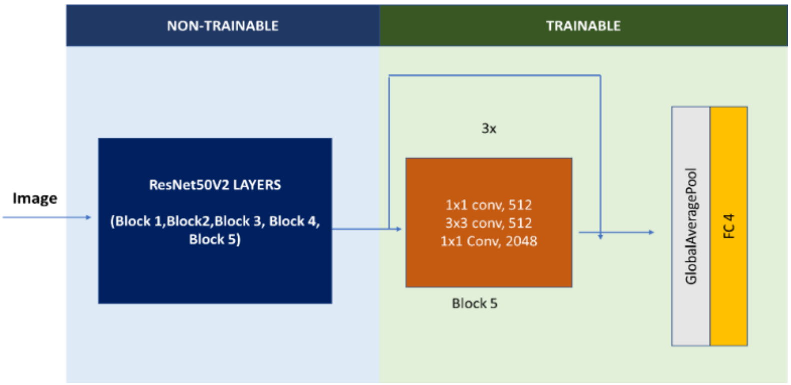


Figure 66: Transfer Learning with ResNet50V2: Architecture of Model 2.

Moving to Model 3, an introduction to regularization techniques was incorporated to improve the model's generalization capability and address overfitting concerns. This model included a dense layer with 512 neurons, a dropout rate of 0.2, and subsequent dense layers with 256 and 100 neurons. Each dense layer was accompanied by L2 regularization of 0.001 and a batch normalization layer. The design of Model 3 was inspired by the work conducted by Asif et al. [71].

In these three models, Model 3 reveals best test results despite being very similar. For that reason, in Model 4 the same model is used but alternating the initial learning rate was it was mentioned before to align with the work of Shurrab et al. [106] and explore the potential impact of varying learning rates on the results. Figure 67 shows the architecture implemented in Model 3 and 4.

The performance of all the ResNet50v2 models in the tests is consistently high and shows minimal variation. This can be attributed to the unique design of ResNet50v2, which incorporates residual blocks. These blocks establish direct connections between earlier layers and those closer to the output, preserving important characteristics throughout the network. This architectural feature effectively combats overfitting by allowing the model to retain and propagate valuable information from earlier layers to the later stages of the network.

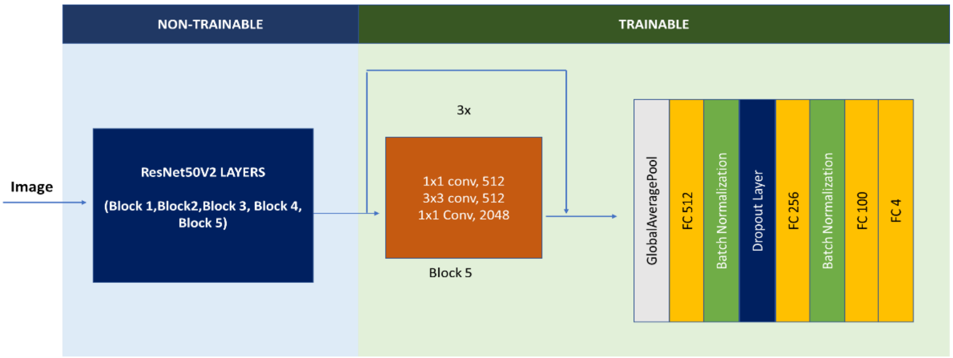


Figure 67: Transfer Learning with ResNet50V2: Architecture of Model 3 and 4.

Among the various models under consideration, Model 3 exhibited the most favorable outcomes and was thus selected for implementation in six distinct configurations. Evaluating Model 3's performance using the original database, it yielded precision, recall, accuracy, and F-beta score of 96.77%, 96.60%, 96.60%, and 96.56%, respectively.

The selection of Model 3 for the implementation of six different configurations was motivated by its superior performance. These configurations are outlined in detail in Table 22 where the model with the highest score is denoted in bold within the table.

Remarkably, the inclusion of data augmentation techniques did not yield substantial improvements in the performance metrics. When applying preprocessing methods, the integration of data augmentation demonstrated only a marginal increase in the test metrics, with a 1%-2% improvement over the results obtained using the original dataset. However, the comparison between the performance of the original images with data augmentation (configuration 4) and the original database without data augmentation (configuration 1) reveals similar results, with only a slight difference.

The additional step of data augmentation does not provide a significant advantage in terms of enhancing the test results despite shows more similar results between train and validation fbeta score and loss values during training process. These findings deviate from the behavior exhibited by VGG 16, where data augmentation notably enhanced the results. In the current context, misclassifications persisted predominantly in the case of Drusen, specifically when compared to CNV. Similar findings were also reported by Shurrab et al. [106].

Additionally, Figure 68 shows the f-beta curve and confusion matrix from the testing dataset comparing the true labels and predicted labels.

Table 21: Results obtained from all models.

MODEL	TRAINABLE PARAMETERS	METRICS	TRAIN	VALIDATION	TEST
1 Modified fc layers architecture: layer of 512 neurons, dropout, layer of 100 neurons, layer of 4 neurons.	16,071,672	Precision	99.17%	96.43%	96.54%
		Recall	99.81%	97.44%	96.30%
		F-Beta score	99.68%	97.22%	96.24%
		Accuracy	99.71%	98.12%	96.30%
		Weight Accuracy	99.91%	96.76%	96.30%
		Loss	0.0066	0.0693	0.2648
2 Modified fc layers architecture: layer of 4 neurons.	14,979,076	Precision	99.22%	97.22%	96.61%
		Recall	99.82%	98.10%	96.40%
		F-Beta score	99.70%	97.92%	96.36%
		Accuracy	99.73%	98.62%	96.40%
		Weight Accuracy	99.93%	97.57%	96.40%
		Loss	0.0059	0.0432	0.2289
3 Modified fc layers architecture: Layer with 512 neurons, a dropout rate of 0.2, and subsequent dense layers with 256 and 100 neurons. Each dense layer was accompanied by L2 regularization of 0.001 and a batch normalization layer.	16,178,936	Precision	**99.15%**	**96.54%**	**96.77%**
		Recall	**99.81%**	**97.62%**	**96.60%**
		F-Beta score	**99.67%**	**97.39%**	**96.56%**
		Accuracy	**99.70%**	**98.25%**	**96.60%**
		Weight Accuracy	**99.92%**	**97.03%**	**96.60%**
		Loss	**0.2521**	**0.0845**	**0.2196**
4 Same architecture as Model 3 with initial learning rate of 5e-6 and learning rate factor of 0.5.	16,178,936	Precision	99.24%	96.59%	94.98%
		Recall	99.83%	97.99%	94.59%
		F-Beta score	99.71%	97.69%	94.48%
		Accuracy	99.73%	98.33%	94.59%
		Weight Accuracy	99.92%	97.80%	94.59%
		Loss	0.8226	0.8612	1.029

Table 22: Test results of the 6 configurations applied to Model 3.

CONFIGURATION	METRICS	CNV	DME	DRUSEN	NORMAL	Average
1 No preprocessing	Precision	**91.14%**	**99.20%**	**99.10%**	**97.64%**	**96.77%**
	Recall	**98.80%**	**99.60%**	**88.40%**	**96.60%**	**96.60%**
	F-Beta score	**97.16%**	**99.52%**	**90.35%**	**96.56%**	**96.56%**
	Accuracy	**98.80%**	**99.60%**	**88.40%**	**96.60%**	**96.60%**
2 Preprocessing 1	Precision	85.86%	98.80%	99.50%	95.76%	94.98%
	Recall	99.60%	98.80%	79.76%	99.60%	94.44%
	F-Beta score	96.51%	98.80%	83.05%	98.80%	94.29%
	Accuracy	99.60%	98.80%	79.76%	99.60%	94.44%
3 Preprocessing 2	Precision	83.44%	98.42%	97.48%	97.60%	94.24%
	Recall	98.80%	100%	77.60%	98.00%	93.60%
	F-Beta score	95.29%	99.68%	80.90%	97.92%	93.44%
	Accuracy	98.80%	100%	77.60%	98.00%	93.60%
4 No preprocessing and data augmentation	Precision	91.20%	99.20%	99.54%	96.87%	96.70%
	Recall	99.60%	99.60%	87.60%	99.20%	96.50%
	F-Beta score	97.80%	99.52%	89.75%	98.72%	96.45%
	Accuracy	99.60%	99.60%	87.60%	99.20%	96.50%
5 Preprocessing 1 and data augmentation	Precision	92.19%	98.80%	98.65%	96.48%	96.53%
	Recall	99.20%	99.60%	88.00%	98.80%	96.40%
	F-Beta score	97.71%	99.44%	89.94%	98.32%	96.35%
	Accuracy	99.20%	99.60%	88.00%	98.80%	96.40%

CONFIGURATION	METRICS	CNV	DME	DRUSEN	NORMAL	Average
6 Preprocessing 2 and data augmentation	Precision	88.53%	97.64%	94.29%	99.15%	94.90%
	Recall	98.80%	99.60%	86.00%	94.40%	94.70%
	F-Beta score	96.55%	99.20%	87.54%	95.31%	94.65%
	Accuracy	98.80%	99.60%	86.00%	94.40%	94.70%

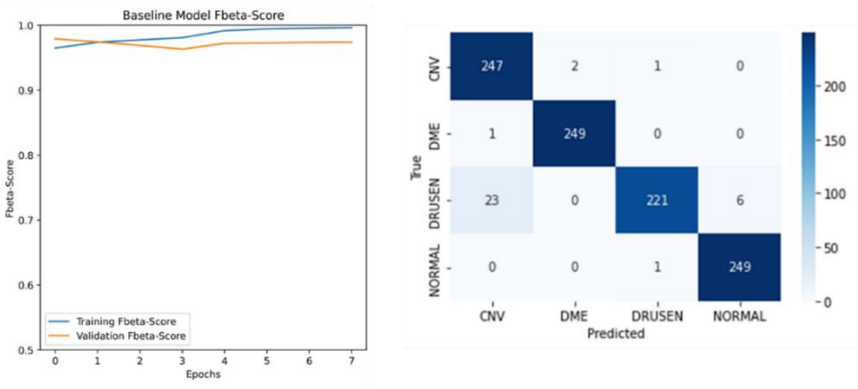


Figure 68: F-beta learning curve and Confusion matrix of best configuration using ResNet50V2 (Original Data).

Proposed Model

In this convolutional neural network model, four distinct methodologies are employed to assess the original image database. It is noteworthy that the input image size for this particular approach is relatively smaller, measuring 128x128x3, in contrast to the larger sizes used in the other two approaches. In Table 23 it is possible to see results of all models used.

In the proposed model it is just used initial the architecture presented in Figure 54: Proposed model architecture.. Model 1 starts with the same learning rate as another deep learning methods. However, as is possible to see in Appendix E, loss curve shows that learning curve used represented a low learning curve – Figure 69 shows types of learning curves when the hyperparameter learning rate is changed. Working with a low learning rate takes a long time to find the best solution and for that reason, in Model 2 it is used the same model however with the learning rate started with 0.001 instead of 0.0001.

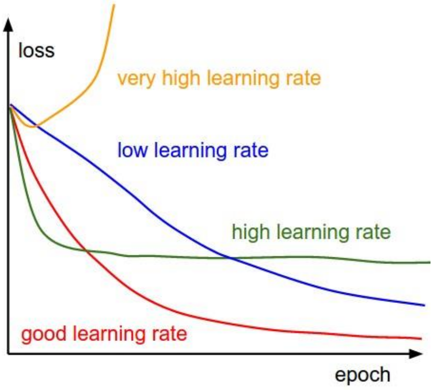


Figure 69: Impact of Learning rate on Neural Network Performance [121].

In Model 2 the results are better in train, validation and test and loss validation and f beta score validation follow the pattern of train curve. Optimal selection of the learning rate plays a crucial role in achieving the lowest possible training loss. Evaluating Model 2 performance using the original database, it yielded 92.37% of precision, 91.10% of recall and accuracy and 90.95% of F-beta score.

To improve the results, two additional models, namely Model 3 and Model 4, were introduced. In Model 3, represented in Figure 70, the number of dropout layers was reduced. This approach gives better results in training and it is performing better however is not capable of generalizing well and had bad metrics than in Model 2. Remove dropout layers did not improve the model generability. As anticipated, the modifications yielded a marginal improvement in the training outcomes. However, the performance on the test set exhibited a decline, indicating a failure of the model to generalize effectively. Rather than enhancing the model's generalization ability, the modifications had an adverse impact, exacerbating the issue.

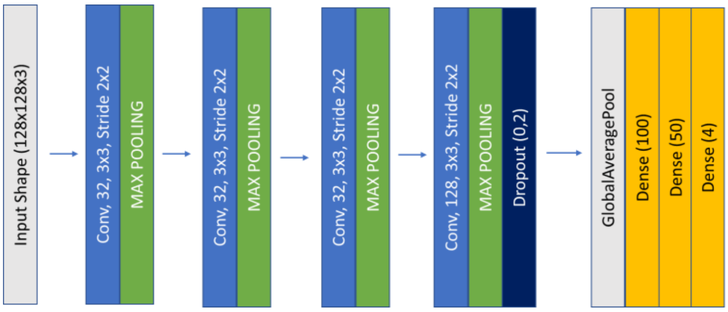


Figure 70: Proposed Model: Architecture of Model 3.

In Model 4, a dense layer consisting of 128 layers was added and dropout layers were placed again. Figure 71 visually represents these modifications in the models. These strategies were implemented with the objective of enhancing the training and validation scores, aiming to augment the model's capacity to effectively learn significant patterns from the training images.

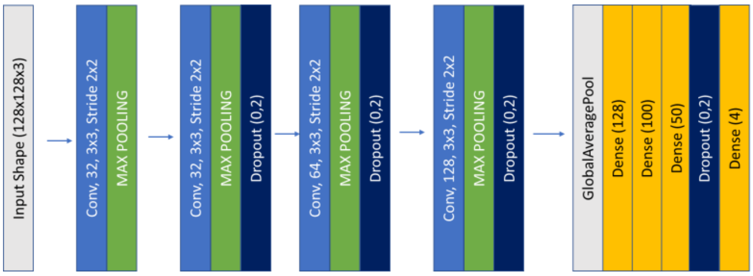


Figure 71: Proposed Model: Architecture of Model 4.

The performance and level of detail captured by the model can be influenced by the discrepancy in image size compared to other methods. This discrepancy may introduce variations in performance. In fact, the approaches employed in models 3 and 4 to enhance performance metrics did not yield better results. In Model 3, the removal of dropout layers, which were crucial for preventing overfitting, resulted in a decline in the model's ability to generalize effectively and in Model 4 the addition of an additional fully connected layer had a negative impact on the model's generalization ability.

In contrast to transfer learning models, all the models in this case demonstrated a regression in distinguishing individuals with diseases from normal retina layers. This discrepancy in results can be attributed to the smaller image size, which hampers the efficiency of capturing fine details. Consequently, these models are susceptible to this limitation.

The selection of Model 2 for the implementation of six different configurations was motivated by its superior performance. These configurations are outlined in detail in Table 23 where the model with the highest score is denoted in bold within the table. The results obtained from different preprocessing approaches, particularly with the inclusion of data augmentation, display notable variations. Comparing the methods utilizing original images and preprocessing 1 with data augmentation to those without data augmentation, a decrease in test performance is observed. Notably, the trend observed in the first four models indicates an increased classification of images as normal.

In future evaluations, it is advisable to consider increasing the image size, as the current trade-off between reducing model complexity and decreasing computational costs while maintaining a reliable model was not achieved. Additionally, the effectiveness of data augmentation techniques was not significant.

However, when preprocessing 2, which incorporates Non-Local Means (NLM), is applied, the test results demonstrate promising outcomes, achieving noteworthy performance metrics. Unlike other preprocessing methods, the utilization of NLM in the images proves to be a valuable approach for improving results despite the small image size.

In fact, this approach yields a faster model with fewer trainable parameters, yet it delivers comparable results when compared to transfer learning models. Table 24 summarizes the metric results for the six configurations and Figure 72 shows the f-beta curve and confusion matrix from the testing dataset comparing the true labels and predicted labels.

Table 23: Proposed Model- Results obtained from all models.

MODEL	TRAINABLE PARAMETERS	METRICS	TRAIN	VALIDATION	TEST
1 Proposed Model	120,650	Precision	90.17%	88.57%	91.23%
		Recall	93.99%	91.08%	90.00%
		F-Beta score	93.11%	90.49%	89.87%
		Accuracy	94.10%	92.51%	90.00%
		Weight Accuracy	92.82%	89.53%	90.00%
		Loss	0.1501	0.1974	0.3485%
2 Model 1 with an initial lr of 0.001	120,650	Precision	**92.64%**	**88.60%**	**92.37%**
		Recall	**96.83%**	**91.51%**	**91.10%**
		F-Beta score	**95.83%**	**90.77%**	**90.95%**
		Accuracy	**96.71%**	**91.62%**	**91.10%**
		Weight Accuracy	**97.02%**	**89.49%**	**91.10%**
		Loss	**0.094**	**0.1909**	**0.4741**
3 Model 2 with less 3 dropout layers	120,650	Precision	96.50%	90.24%	90.67%
		Recall	98.66%	90.60%	88.69%
		F-Beta score	98.18%	90.51%	88.26%
		Accuracy	98.55%	92.19%	88.70%
		Weight Accuracy	98.81%	86.80%	88.70%
		Loss	0.0442	0.2001	0.7116
4 Model 2 with one more fc layer of 128 neurons	137,162	Precision	92.39%	89.44%	84.74%
		Recall	97.15%	93.27%	81.00%
		F-Beta score	95.97%	92.31%	79.65%
		Accuracy	96.69%	94.09%	81.00%
		Weight Accuracy	97.72%	92.44%	81.00%
		Loss	0.0932	0.1829	1.0671

Table 24: Proposed Model: Test results of the 6 configurations applied.

CONFIGURATION	METRICS	CNV	DME	DRUSEN	NORMAL	Average
1 No preprocessing	Precision	89.84%	99.56%	99.49%	80.58%	92.37%
	Recall	95.60%	90.80%	78.40%	99.60%	91.10%
	F-Beta score	94.39%	92.42%	81.87%	95.11%	90.95%
	Accuracy	95.60%	90.80%	78.40%	99.60%	91.10%
2 Preprocessing 1	Precision	88.32%	97.55%	100%	71.34%	89.30%
	Recall	96.80%	95.60%	52.80%	99.60%	86.20%
	F-Beta score	94.97%	95.98%	58.30%	92.29%	85.38%
	Accuracy	96.80%	95.60%	52.80%	99.60%	86.20%
3 Preprocessing 2	Precision	**90.87%**	**98.01%**	**98.69%**	**99.59%**	**96.79%**
	Recall	**99.60%**	**98.80%**	**90.80%**	**97.20%**	**96.60%**
	F-Beta score	**97.72%**	**98.64%**	**92.27%**	**97.66%**	**96.57%**
	Accuracy	**99.60%**	**98.80%**	**90.80%**	**97.20%**	**96.60%**
4 No preprocessing and data augmentation	Precision	95.91%	95.28%	98.47%	51.86%	85.38%
	Recall	75.20%	72.80%	51.60%	100%	74.90%
	F-Beta score	78.59%	76.40%	57.02%	84.34%	74.09%
	Accuracy	75.20%	72.80%	51.60%	100%	74.90%
5 Preprocessing 1 and data augmentation	Precision	95.40%	97.67%	99.03%	47.34%	84.86%
	Recall	74.80%	67.20%	41.20%	100%	70.80%
	F-Beta score	78.17%	71.67%	46.64%	81.80%	69.57%
	Accuracy	74.80%	67.20%	41.20%	100%	70.80%

CONFIGURATION	METRICS	CNV	DME	DRUSEN	NORMAL	Average
6 Preprocessing 2 and data augmentation	Precision	89.81%	96.88%	100%	95.38%	95.52%
	Recall	98.80%	99.60%	83.20%	99.20%	95.20%
	F-Beta score	96.86%	99.04%	86.09%	98.41%	95.10%
	Accuracy	98.80%	99.60%	83.20%	99.20%	95.20%

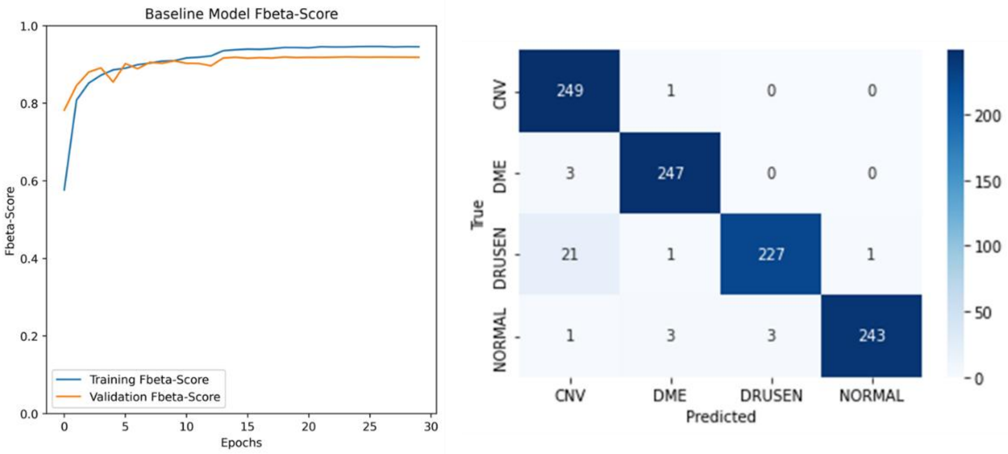


Figure 72: F-beta learning curve and Confusion matrix of best configuration using Proposed Model (Images with preprocessing 2).

Result Analysis

In this analysis, it is evident that deep learning methods have outperformed traditional methods. The performance metrics of these models, however, indicate that not all models achieved the best results in the same manner. By employing VGG16, it becomes apparent that applying data augmentation techniques and altering the original images can lead to superior results, sometimes surpassing the results obtained with original images by a margin of 4%. Notably, the best results obtained with VGG16 were achieved when utilizing data augmentation and preprocessing method 1. The model was trained using 2,674,168 parameters and achieves 96.88% precision, 96.80% recall and accuracy and 96.79% fbeta score.

The model ResNet50V2 exhibited similar metrics. It consisted of 16,178,936 trainable parameters. In this case, the application of preprocessing methods did not enhance the results significantly, however, when combined with data augmentation, there was a modest improvement of 1%-2% across most metrics. The best results were obtained when using only the original images. This finding suggests that data augmentation techniques were not as effective as with VGG 16 method since the overfitting results, using the original database, were lower using the original database. In fact, ResNet50V2 outperformed VGG16 in predicting results using the original database. ResNet50V2 was trained with 16,178,936 parameters and achieves 96.77% of precision, 96.60% recall and accuracy and 96.56% fbeta score.

Despite achieving similar results, ResNet50V2 requires an additional 13,504,768 parameters compared to VGG16, which incurs higher computational costs. This metric holds significance and should be taken into consideration. The chosen model for VGG16 performed better with a learning rate of 0.001 and by unfreezing only one convolutional layer. On the other hand, ResNet50V2 required unfreezing more layers to achieve similar results.

The proposed model, employing an image size of 128x128, achieves a precision of 96.79%, recall of 96.60% and accuracy and a fbeta score of 96.58 %. The reduction of the image size, aimed at simplifying and expediting the model, did not yield satisfactory results with the original images and preprocessing 1, even when data augmentation was applied. However, when employing a NLM filter, the obtained values closely resemble those achieved through transfer learning models. This suggests that this filter is effectively applied to small images, which can have significant implications in attaining superior results while utilizing less complex and faster models with reduced computational costs. Further investigations encompassing varying image sizes are necessary to ascertain the NLM filter's potential in achieving better outcomes with limited data sizes.

Non-Local Means (NLM) emerges as a promising filter, particularly when applied to images with fewer distinctive features. Its efficacy makes it a viable approach for future studies, facilitating enhanced performance and providing a fertile ground for further exploration.

To effectively evaluate the usefulness of data augmentation techniques and preprocessing methodologies, it is advisable to utilize additional transfer learning models. Regardless of the model employed, methods yielded inferior results when applied to Drusen images. Despite the implementation of data augmentation techniques and preprocessing, this pattern remains evident. Hence, the introduction of alternative transfer learning models becomes crucial in assessing the performance and attainment of satisfactory outcomes.

Table 25 presents the best results attained by each implemented model in conjunction with state-of-the-art results. While numerous studies have employed similar methodologies, Table 25 exclusively showcases studies that align with the version employed in this work, namely version 3, to yield a more optimal solution. Evidently, this research has yielded superior results compared to certain studies within the field. Moreover, upon observing the table, it becomes evident that employing the fbeta score with a beta value of 2.0 does not significantly differ from the f1-score.

Table 25 : Comparison of the model results with other state-of-the-art models.

Author	**Method**	**Metrics**				
		Precision	**Recall**	**F1-Score**	**Accuracy**	**Fbeta-score**
Li et al. [101]	VGG-16	98.61%	98.6%	98.59%	98.60%	98.59%
Proposed Study	**VGG16**	**96.88%**	**96.80%**	**96.80%**	**96.80%**	**96.79%**
Proposed Study	**Proposed Method**	**96.79%**	**96.60%**	**96.60%**	**96.60%**	**96.58%**
Proposed Study	**ResNet50V2**	**96.77%**	**96.60%**	**96.56%**	**96.60%**	**96.56%**
Kermany et al. [41]	Inception V3 architecture	96.60%	97.80%	-	-	-
Shurrab et al. [106]	ResNet34	-	90.7%	-	90.7%	-

8 Conclusions and Future Work

In this study, the performance of two handcrafted feature extraction methods, namely HOG and LBP, trained using a Support Vector Machine classifier, was measured. Additionally, the effectiveness of three CNN models was evaluated: two pre-trained deep learning models utilizing VGG16 and ResNet50V2 architectures through transfer learning, and one proposed model designed for feature extraction from OCT images.

The obtained results highlight the superiority of deep neural network methods over HOG and LBP due to their automatic and effective feature learning and selection capabilities. The utilization of transfer learning and fine-tuning significantly improved the effectiveness of these pretrained models. Among the models evaluated, the VGG16 model exhibited the best performance, closely followed by ResNet50V2, despite having fewer trainable parameters. Notably, the proposed model achieved comparable results by incorporating the Non-Local Means (NLM) filter and employing a small image size, thereby reinforcing the potential for reduced computational costs.

When utilizing OCT equipment, it is crucial to consider the trade-off between response time and accuracy achieved, and hence, despite yielding lower results compared to VGG16, due to its lower number of parameters the proposed model presents a hypothesis worth considering.

However, it is worth noting that all models demonstrated a tendency to exhibit lower capability in detecting Drusen disease. Furthermore, the results indicated that the effectiveness of data augmentation techniques was not consistent across all models. While data augmentation is generally considered beneficial for improving model performance, in this particular study, it did not consistently yield better results.

The study was conducted by comparing the proposed models with state-of-the-art models in the field of retinal disease detection. The results of this comparison were positive, indicating that the proposed models outperformed or achieved comparable performance to the existing models.

Throughout the process, four preprocessing techniques were employed on the original OCT dataset images to mitigate speckle noise. Additionally, the f beta score (beta=2.0) was used, prioritizing recall over precision. This measure holds particular importance in the context of medicine, as it emphasizes the significance of minimizing false negatives rather than false positives. Given the nature of medical diagnoses, the detection of critical findings holds significant implications for patient outcomes. This prioritization plays a vital role in early detection, prompt treatment, and improved patient care, ultimately contributing to better treatment planning and potentially saving lives.

In future studies, it is recommended to explore additional preprocessing methods for image denoising to assess their potential in improving results for detecting Drusen disease. Furthermore, incorporating different transfer learning methods could provide valuable insights into the effectiveness of data augmentation techniques. In addition, augmenting the test set is recommended to enhance the generalizability of the model. Expanding the dataset by including a larger number of images from diverse patients or incorporating synthetic images generated using GANs can serve as valuable resources for evaluating the model's performance. Increasing the diversity and quantity of data in the test set can contribute to improving the overall efficacy and robustness of the model. Finally, conducting a comparison study with specialists can offer valuable insights and validation of the model's performance. In future research, it would be beneficial to evaluate the model's generalization capabilities across multiple databases, encompassing various equipment and hospitals, to assess its performance in diverse real-world contexts.

In the current scenario, in some countries, AI algorithms are already being employed to monitor the day-to-day progression of fluid in patients with CNV at home. By utilizing a user-friendly device, patients can conveniently perform eye scans and an AI algorithm undertakes tasks such as identifying, localizing, quantifying, and mapping the fluid within the eye. This integrated approach aims to reduce the time between fluid detection and subsequent treatment, potentially improving the efficiency of patient care. As OCT machines are exclusively available in clinical settings, patients are required to physically visit these facilities for imaging, which plays a crucial role in determining disease progression and treatment intervals. However, frequent clinic visits can impose burdens on elderly patients and clinics, resulting in increased costs.

With the implementation of Home OCT, patients with stable conditions would only need to visit when fluid reaccumulates, while those requiring frequent injections would be informed about the necessity of in-clinic injections. However, ensuring optimal functionality of a home OCT system relies on meeting four key requirements: patients' ability to perform self-imaging using an OCT device, automated and accurate analysis of daily images, an affordable device capable of delivering high-quality images, and remote monitoring by clinics to ensure patient adherence [123].

By meeting these requirements, AI has the potential to bring about a significant transformation in the treatment paradigm, allowing for more personalized treatment approaches, reduced resource utilization, and ideally, lowered healthcare costs.

9 References

[1] I. Goodfellow, Y. Bengio e A. Courville, Deep Learning, MIT Press, 2016.

[2] Z. Ahmad, S. Rahim, M. Zubair and J. Abdul-Ghafar, “Artificial intelligence (AI) in medicine, current applications and future role with special emphasis on its potential and promise in pathology: present and future impact, obstacles including costs and acceptance among pathologists, practical and philosoph,” *Diagn Pathol,* vol. 16, no. 1, p. 24, March 2021.

[3] R. Manne e S. C. Kantheti, “Application of Artificial Intelligence in Healthcare: Chances and Challenges,” *Current Journal of Applied Science and Technology,* vol. 40, nº 6, 2021.

[4] S. G. Honavar, “Artificial intelligence in ophthalmology - Machines think!,” *Indian Journal of Ophthalmology,* vol. 70, nº 4, pp. 1075-1079, 2022.

[5] “Vision impairment and blindness,” [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment. [Accessed March 2023].

[6] E. A. Swanson and J. G. Fujimoto, “The ecosystem that powered the translation of OCT from fundamental research to clinical and commercial impact,” *Biomedical Optics Express,* vol. 8, no. 3, pp. 1638-1664, 2017.

[7] S. R. Singh e J. Chhablani, “Optical Coherence Tomography Imaging: Advances in Ophthalmology,” *Journal of Clinical Medicine,* vol. 11, nº 10, p. 2858, 2022.

[8] A. S. S. R. Miguel, Manual de higiene e segurança do trabalho, 13 ed., Porto: Porto Editora, 2014.

[9] C. Garhart e V. Lakshminarayanan, “Anatomy of the Eye,” em *Handbook of Visual Display Technology,* Springer, 2016.

[10] k. Irsch e D. Guyton, “Anatomy of Eyes,” em *Encyclopedia of Biometrics,* Springer, 2019.

[11] “Anatomy of the Eye,” [Online]. Available: https://columbiaeyeclinic.com/anatomy-eye/.

[12] A. Mescher, Junqueira's Basic Histology: Text and Atlas, McGraw-Hill, 2013.

[13] H. Kolb, “Simple Anatomy of the Retina,” 1995. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK11533/.

[14] H. Davson, The Eye, 3 ed., vol. 1, Academic Press, 1984.

[15] A. Born, R. Tripathi e B. Tripathi, Wolff's Anatomy of the Eye and Orbit, 8 ed., London: Chapman & Hall Medical, 1997, pp. 211-232; 308-334; 454-596.

[16] C. Contet, S. Goulding, D. Kuljis e A. Barth., “BK Channels in the Central Nervous System,” em *International Review of Neurobiology,* vol. 128, Elsevier, 2016.

[17] A. C. A. Cerveró and J. Riancho., “Retinal changes in amyotrophic lateral sclerosis: looking at the disease through a new window,” *Journal of Neurology,* vol. 268, pp. 2083-2089, 2021.

[18] G. Mohandass, R. A. Natarajan e S. Sendilvelan, “Retinal Layer Segmentation in Pathological SD-OCT Images Using Boisterous Obscure Ratio Approach and its Limitation,” *Biomedical and Pharmacology Journal,* vol. 10, 2017.

[19] E. J. Carmona and J. M. Molina-Casado, “Neural Computing and Applications,” *Simultaneous segmentation of the optic disc and fovea in retinal images using evolutionary algorithms.,* vol. 33, pp. 1903-1921, 2021.

[20] R. S. Snell and M. A. Lemp, Clinical Anatomy of the Eye, 2nd ed., Wiley- Blackwell, 2013.

[21] J. P. Nordmann, OCT & Optic Nerve, Paris: Laboratoire Théa, 2014.

[22] A. R. Figueiredo, A. Martins, A. C. Almeida and e. al, OCT, 1 st ed., Lisboa, Lisboa: Sociedade Portuguesa de Oftalmologia, 2016.

[23] D. P. Popescu, L.-P. Choo-Smith, C. Flueraru, Y. Mao, S. Chang, J. Disano, S. Sherif and M. G. Sowa, “Optical coherence tomography: fundamental principles, instrumental designs and biomedical applications,” *Biophysical Reviews,* vol. 3, pp. 155-169, 2011.

[24] M. Bhende, S. Shetty, M. K. Parthasarathy and R. S., “Optical coherence tomography: A guide to interpretation of common macular diseases,” *Indian Journal of Ophthalmology,* vol. 66, no. 1, pp. 20-35, January 2018.

[25] K. R. . Baran, G. J. Laurent, P. Rougeot, N. Andreff e B. Tamadazte, “reliminary results on OCT-based position control of a concentric tube robot,” em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Vancouver, Canada, 2017.

[26] N. Yoshimura e M. Hangai, OCT Atlas, Berlin: Springer, 2016.

[27] P. Serranho, A. Morgado e a. R. Bernardes, “Optical Coherence Tomography: A Concept Review,” em *Optical Coherence Tomography,* R. Bernardes e J. Cunha-Vaz, Edits., Springer, 2012, p. 139–156.

[28] J. A. Izatt and M. A. Choma, “Theory of Optical Coherence Tomography,” in *Optical Coherence Tomography,* W. Drexler and J. G. Fujimoto, Eds., Springer, 2008.

[29] S. Aumann, S. Donner, J. Fischer e F. Müller, “Optical Coherence Tomography (OCT): Principle and Technical Realization,” em *High Resolution Imaging in Microscopy and Ophthalmology,* J. Bille, Ed., Springer, 2019.

[30] M. Adhi, J. J. Liu, A. H. Qavi, et e all, “Choroidal Analysis in Healthy Eyes using Swept-Source Optical Coherence Tomography Compared to Spectral Domain Optical Coherence Tomography,” *American Journal of Ophthalmology,* vol. 157, nº 6, pp. 1272-1281, 2014.

[31] J. Chhablani, T. Krishnan, V. Sethi e I. Kozak, “Artifacts in optical coherence tomography,” *Saudi Journal of Ophthalmology: Official Journal of the Saudi Ophthalmological Society,* vol. 28, nº 2, pp. 81-87, 2014.

[32] Y. Ma, X. Chen, W. Zhu, X. Cheng, D. Xiang and F. Shi, “Speckle noise reduction in optical coherence tomography images based on edge-sensitive cGAN,” *Biomedical Optics Express,* vol. 9, no. 11, pp. 5129-5146, November 2018.

[33] P. K. Neha Gour, “Speckle denoising in optical coherence tomography images using residual deep convolutional neural network,” *Multimedia Tools and Applications,* vol. 79, 2020.

[34] M. L. Gabriele, G. Wollstein, H. Ishikawa, J. K. J. Xu, L. S. F. L. Kagemann e J. S. Schuman, “Three dimensional optical coherence tomography imaging: advantages and advances,” *Prog Retin Eye Res,* vol. 29, pp. 556-579, November 2010.

[35] W. Wenjun, Y. Gong, H. Huaying, J. Zhang, P. Su, Q. Yan, Y. Ma and Y. Zhao, “Choroidal layer segmentation in OCT images by a boundary enhancement network,” *Frontiers in Cell and Developmental Biology,* vol. 10, 2022.

[36] R. Rocholz, F. Corvi, J. Weichsel, al. and et, “OCT Angiography (OCTA) in Retinal Diagnostics,” in *High Resolution Imaging in Microscopy and Ophthalmology: New Frontiers in Biomedical Optics,* J. F. Bille, Ed., Cham: Springer, 2019.

[37] H. Rabbani, R. Kafieh e Z. Amini, “Optical Coherence Tomography Image Analysis,” 2016.

[38] A. Baghaie, Z. Yu and R. M. D'Souza, “State-of-the-art in retinal optical coherence tomography image analysis,” *Quantitative Imaging in Medicine and Surgery,* vol. 5, pp. 603-617, 2015.

[39] A. You, J. Kim, I. Ryu, et e al., “Application of generative adversarial networks (GAN) for ophthalmology image domains: a survey,” *Eye and Vision,* vol. 9, nº 6, pp. 1-11, 2022.

[40] A. Nunes, G. Silva, C. Duque, C. Januário, I. Santana, A. F. Ambrósio, M. Castelo-Branco e R. Bernardes, “Retinal texture biomarkers may help to discriminate between Alzheimer's, Parkinson's, and healthy controls,” *PLoS One,* vol. 14, nº 6, p. e0218826, 2019.

[41] D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. L. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi and et, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell,* vol. 172, no. 5, pp. 1122-1131, February 2018.

[42] T. D. L. Keenan, C. A. Cukras e E. Y. Chew, "Age-Related Macular Degeneration: Epidemiology and Clinical Aspects," *Advances in Experimental Medicine and Biology,* vol. 1256, pp. 1-31, 2021.

[43] S. Sotoudeh-Paima, A. Jodeiri, F. Hajizadeh e H. Soltanian-Zadeh, "Multi-scale convolutional neural network for automated AMD classification using retinal OCT images," *Computers in Biology and Medicine,* vol. 144, 2022.

[44] Istanbul Retina Institute, "OCT CLUB," 1 March 2021. [Online]. Available: https://en.octclub.org/yasa-bagli-makula-dejenerasyonu/. [Accessed May 8 2023].

[45] O. Musat, C. Cernat, M. Labib, A. Gheorghe, O. Toma, M. Zamfi e A. M. B. , "DIABETIC MACULAR EDEMA," *Romanian Journal of Ophthalmology,* vol. 59, nº 3, pp. 133-136, 2015.

[46] M. J. Hasan, M. S. Alom e U. Fatema, *Classification Performance Analysis of Retinal OCT Image using Handcrafted and Deep Learning Feature with Support Vector Machine,* 2021.

[47] D. Hwang, C. Hsu, K. Chang, D. Chao, C. Sun, Y. Jheng, A. Yarmishyn, J. Wu, C. Tsai, M. Wang, C. Peng, K. Chien, C. Kao, T. Lin, L. Woung, S. Chen e S. Chiou, "Artificial intelligence-based decision-making for age-related macular degeneration," *Theranostics,* vol. 9, nº 1, pp. 232-245, 2019.

[48] S. Khan, X. Liu, S. Nath, E. Korot, L. Faes, S. Wagner, P. Keane, N. Sebire, M. Burton e A. Denniston, "A global review of publicly available datasets for ophthalmological imaging: barriers to access, usability, and generalisability," *Lancet Digit Health,* vol. 3, nº 1, pp. e51-e66, 2019.

[49] K. P. Murphy, Machine learning: a Probabilistic Perspective, Cambridge, MA: MIT Press, 2012.

[50] T. T. Hormel, T. S. Hwang, S. T. Bailey, D. J. Wilson, D. Huang e Y. Jia, "Artificial intelligence in OCT angiography," *Progress in Retinal and Eye Research,* vol. 85, p. 100965, 2021.

[51] G. Rebala, A. Ravi e S. Churiwala, An Introduction to Machine Learning, Springer, 2019.

[52] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach, Third ed., Prentice Hall, 2010.

[53] A. Trask, Grokking Machine Learning, Manning, 2019.

[54] H. Demir e F. Sarı, "The Effect of Artificial Intelligence and Industry 4.0 on Robotic Systems," em *Advances in Robotics, Artificial Intelligence and Industry 4.0,* Istanbul, Istanbul University Press, 2020, pp. 51-72.

[55] T. M. Mitchell and M. Tom, Machine Learning, New York: McGraw-Hill, 1997.

[56] R. Lao, "A Beginner's Guide to Machine Learning," 22 January 2018. [Online]. Available: https://medium.com/@randylaosat/a-beginners-guide-to-machine-learning-dfadc19f6caf. [Accessed 19 March 2023].

[57] NCS, "SVM Hyperparameter Tuning using GridSearchCV," 10 March 2020. [Online]. Available: https://www.vebuso.com/2020/03/svm-hyperparameter-tuning-using-gridsearchcv/. [Accessed 8 May 2023].

[58] C. Sampaio, "Understanding SVM Hyperparameters," 26 April 2023. [Online]. Available: https://stackabuse.com/understanding-svm-hyperparameters/. [Accessed 8 May 2023].

[59] D. Utomo, T. Ummah, D. Sulistyaningrum, B. Setiyono e Soetrisno, "Vehicle detection using histogram of oriented gradients and real adaboost," *Journal of Physics: Conference Series,* vol. 1490, p. 012001, 2020.

[60] C. McCormick, “HOG Person Detector Tutorial,” 2013. [Online]. Available: https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/. [Accessed 17 April 2023].

[61] “Wood identification based on histogram of oriented gradient (HOG) feature and support vector machine (SVM) classifier,” em *International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2017.

[62] J. Ylioinas, A. Hadid e M. Pietikäinen, “Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns (CAIP),” em *Combining Contrast Information and Local Binary Patterns for Gender Classification*, Seville, 2011.

[63] T. Ojala, M. Pietikainen e T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, nº 7, pp. 971-987, 2002.

[64] S. Jain, “Local Binary Pattern Features for Texture Classification,” [Online]. Available: https://becominghuman.ai/local-binary-pattern-features-for-texture-classification-d0dfd86ebf29. [Accessed 18 April 2023].

[65] G. Lemaître, T. Leng, M. Rastgoo, J. Massich, C. Y. Cheung, T. Y. Wong, E. Lamoureux, D. Milea, F. Mériaudeau and D. Sidibé, “Classification of SD-OCT Volumes Using Local Binary Patterns: Experimental Validation for DME Detection,” vol. 2016, p. 3298606, 2016.

[66] A. Pregowska e M. Osial, “What Is An Artificial Neural Network And Why Do We Need It?",” *Frontiers for Young Minds,* vol. 9, p. 560631, 2021.

[67] “Neural Network Architectures,” 27 July 2019. [Online]. Available: https://freecontent.manning.com/neural-network-architectures/. [Accessed 19 March 2023].

[68] C. Åleskog, H. Grahn e A. Borg, “Recent Developments in Low-Power AI Accelerators: A Survey,” *Algorithms,* vol. 15, nº 11, p. 419, November 2022.

[69] I. Kandel e M. Castelli, “Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review,” *Applied Sciences,* vol. 10, nº 6, 2021.

[70] M. J. Hasan, M. S. Alom e U. Fatema, “Classification Performance Analysis of Retinal OCT Image using Handcrafted and Deep Learning Feature with Support Vector Machine,” May 2021.

[71] S. Asif, K. Amjad e Q. U. Ain, “Deep Residual Network for Diagnosis of Retinal Diseases Using Optical Coherence Tomography Images,” *Interdisciplinary Sciences,* vol. 14, nº 4, pp. 906-916, December 2022.

[72] A. Kumar, “Stochastic Gradient Descent Python Example,” 20 April 2022. [Online]. Available: https://vitalflux.com/stochastic-gradient-descent-python-example/. [Accessed 19 March 2023].

[73] A. Prasad, “The Bias-Variance trade-off : Explanation and Demo,” Towards Data Science, 6 April 2019. [Online]. Available: https://towardsdatascience.com/the-bias-variance-trade-off-explanation-and-demo-8f462f8d6326. [Accessed 8 May 2023].

[74] T. Hastie, R. Tibshirani e J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2016.

[75] H. F. Aarabi, “Towards global tempo estimation and rhythm-oriented genre classification based on harmonic characteristics of rhythm,” 2021.

[76] “Convolutional Neural Network: Deep learning,” 25 January 2022. [Online]. Available: https://developersbreach.com/convolution-neural-network-deep-learning/. [Accessed 19 March 2023].

[77] Z. Qin, F. Yu, C. Liu e X. Chen, “How convolutional neural networks see the world - A survey of convolutional neural network visualization methods,” vol. 1, nº 2, pp. 149-180, 2018.

[78] D. Johnson, “CNN Image Classification in TensorFlow with Steps & Examples,” 21 January 2023. [Online]. Available: https://www.guru99.com/convnet-tensorflow-image-classification.html. [Accessed 19 March 2023].

[79] A. Nguyen, K. Pham, D. Ngo, T. Ngo e L. Pham, “An Analysis of State-of-the-art Activation Functions For Supervised Deep Neural Network,” em *nternational Conference on System Science and Engineering (ICSSE),* Ho Chi Minh City, Vietnam, 2021.

[80] “CNN: Introduction to pooling layer,” 11 January 2023. [Online]. Available: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/. [Accessed 19 March 2023].

[81] N. Cui, “Applying Gradient Descent in Convolutional Neural Networks,” *Journal of Physics: Conference Series,* vol. 1004, p. 012027, 2018.

[82] N. Donges, “What is transfer learning? exploring the popular deep learning approach.,” Built In, [Online]. Available: https://builtin.com/data-science/transfer-learning. [Accessed 19 March 2023].

[83] F. Chollet, “Transfer learning & fine-tuning,” Keras, 2015 May 2020. [Online]. Available: https://keras.io/guides/transfer_learning/#setup. [Accessed 2023 May 2023].

[84] H. D. Regua, “Introducing Transfer Learning as Your Next Engine to Drive Future Innovations,” [Online]. Available: https://medium.datadriveninvestor.com/introducing-transfer-learning-as-your-next-engine-to-drive-future-innovations-5e81a15bb567. [Accessed 19 March 2023].

[85] Y. Yao, L. Rosasco e A. Caponnetto, “On Early Stopping in Gradient Descent Learning,” *Constructive Approximation,* vol. 26, pp. 289-315, 2007.

[86] R. Gencay e M. Qi, “Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging,” *IEEE Transactions on Neural Networks,* vol. 12, nº 4, pp. 726-734, July 2001.

[87] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever e R. Salakhutdinov., “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research,* vol. 15, pp. 1929 - 1958, June 2014.

[88] R. Moore e J. DeNero, “L1 and L2 regularization for multiclass hinge loss models,” *Symposium on Machine Learning in Speech and Language Processing,* 2011.

[89] J. Korstanje, “The F1 score,” 31 August 2021. [Online]. Available: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6. [Accessed 2023 March 20].

[90] “Sklearn.metrics.roc_auc_score,” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html. [Accessed 8 May 2023].

[91] S. Narkhede, “Understanding Confusion Matrix,” 9 May 2018. [Online]. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62. [Accessed 20 March 2023].

[92] P. Srinivasan, L. Kim, P. Mettu, S. Cousins, G. Comer, J. Izatt e S. Farsiu, “Fully automated detection of diabetic macular edema and dry age-related macular degeneration from optical coherence tomography images,” *Biomedical Optics Express,* vol. 5, nº 10, pp. 3568-3577, 2014.

[93] S. Farsiu, S. Chiu, R. O'Connell, F. Folgar, E. Yuan, J. Izatt e C. Toth, “Quantitative classification of eyes with and without intermediate age-related macular degeneration using optical coherence tomography,” *Ophthalmology,* vol. 121, nº 1, pp. 162-172, 2014.

[94] S. Sotoudeh-Paima, F. Hajizadeh and H. Soltanian-Zadeh, *Labeled Retinal Optical Coherence Tomography Dataset for Classification of Normal, Drusen, and CNV Cases,* vol. 1, 2021.

[95] P. Gholami, M. K. Parthasarathy, P. Roy e V. Lakshminarayanan, “OCTID: Optical Coherence Tomography Image Database,” *Borealis,* vol. 1, 2018.

[96] A. Albarrak, F. Coenen e Y. Zheng, “Age-related macular degeneration identification in volumetric optical coherence tomography using decomposition and local feature extraction,” em *International Conference on Medical Image Understanding and Analysis.*, 2013.

[97] Y. Sun, S. Li e Z. Sun, “Fully automated macular pathology detection in retina optical coherence tomography images using sparse coding and dictionary learning,” *Journal of Biomedical Optics,* vol. 22, p. 016012, 2017.

[98] Y.-Y. Liu, M. Chen, H. Ishikawa, G. Wollstein, J. S. Schuman e J. M. Rehg., “Automated macular pathology diagnosis in retinal OCT images using multi-scale spatial pyramid and local binary patterns in texture and shape encoding,” *Medical Image Analysis,* vol. 15, nº 5, pp. 748-759, 2011.

[99] J. Kaur, “Automatic log analysis using Deep Learning and Ai,” Xenonstack Inc, 8 March 2023. [Online]. Available: https://www.xenonstack.com/blog/log-analytics-deep-machine-learning. [Accessed 19 March 2023].

[100] C. S. Lee, D. M. Baughman e A. Y. Lee, “Deep Learning Is Effective for Classifying Normal versus Age-Related Macular Degeneration OCT Images,” *Kidney International Reports,* vol. 2, nº 4, pp. 322-327, 2017.

[101] F. Li and H. Chen, “Fully automated detection of retinal disorders by image-based deep learning,” *Graefes Arch Clin Exp Ophthalmol,* vol. 257, no. 3, pp. 495-505, 2019.

[102] A. Serener e S. Serte, “Dry and Wet Age-Related Macular Degeneration Classification Using OCT Images and Deep Learning,” em *Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT),* Istanbul, Turkey, 2019.

[103] S. Kaymak e A. Serener, “Automated Age-Related Macular Degeneration and Diabetic Macular Edema Detection on OCT Images using Deep Learning,” em *14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 2018.

[104] S. R. Shatil e M. M. J. Kabir, “Retinal OCT Image Classification Based on CNN and Transfer Learning,” em *Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition*, Springer Nature Switzerland, 2022, pp. 433-444.

[105] N. Tasnim, M. Hasan e I. Islam, “Comparisonal study of Deep Learning approaches on Retinal OCT Image,” em *International Conference on Innovation in Engineering and Technology (ICIET) 2*, Bangladesh, 2019.

[106] S. Shurrab, Y. Shannak e R. Duwairi, “Retina Disorders Classification via OCT Scan: A Comparative Study between Self-Supervised Learning and Transfer Learning,” *he International Arab Journal of Information Technology,* p. 20, 1 January 2023.

[107] A. Arora, S. Gupta, S. Singh e J. Dubey, “Eye Disease Detection Using Transfer Learning on VGG16,” em *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security*, 2022.

[108] K. A. Nugroho, “A Comparison of Handcrafted and Deep Neural Network Feature Extraction for Classifying Optical Coherence Tomography (OCT) Images,” em *2nd International Conference on Informatics and Computational Sciences (ICICoS)*, Indonesia, 2018.

[109] Editorial Team, “Computer Vision in Healthcare: Secret Guide for Winners,” [Online]. Available: https://anywhere.epam.com/business/computer-vision-in-the-medical-field. [Accessed 19 March 2023].

[110] A. Buades, B. Coll e J.-M. Morel, “A non-local algorithm for image denoising,” em *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* San Diego, CA, 2005.

[111] A. Géron, Hands-On Machine Learning with scikit-learn & TensorFlow, O'Reilly, 2019.

[112] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks,* vol. 13, no. 2, pp. 415-425, 2002.

[113]K. Simonyan e A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv 1409.1556,* 4 September 2014.

[114]M. u. Hassan, “VGG16 – Convolutional Network for Classification and Detection,” Neurohive, 20 November 2018. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/. [Accessed 8 May 2023].

[115]K. He, X. Zhang, S. Ren e J. Sun, “Deep Residual Learning for Image Recognition,” em *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015.

[116]K. He, X. Zhang, S. Ren e J.Sun, “Identity Mappings in Deep Residual Networks,” em *Computer Vision – ECCV 2016*, vol. 9908, Springer, 2016.

[117]A. Manzoor, W. Ahmad, M. Ehatisham-ul-Haq, A. Hannan, M. A. Khan, M. U. Ashraf, A. Alghamdi e A. Alfakeeh, “Inferring Emotion Tags from Object Images Using Convolutional Neural Network,” *Applied Sciences,* vol. 10, p. 5333, 2020.

[118]A. Karpathy, “Cs231n: Convolutional Neural Networks for Visual Recognition,” [Online]. Available: http://cs231n.stanford.edu/2016/. [Accessed 10 June 2023].

[119]W.-M. Lee, “Image Data Augmentation for Deep Learning,” Towards Data Science, 26 October 2022. [Online]. Available: https://towardsdatascience.com/image-data-augmentation-for-deep-learning-77a87fabd2bf. [Accessed 10 June 2023].

[120]TensorFlow, “Classification on imbalanced data,” TensorFlow, [Online]. Available: https://www.tensorflow.org/tutorials/structured_data/imbalanced_data#confirm_that_the_bias_fix_helps. [Accessed 10 June 2023].

[121]M. Torres-Velázquez, W.-J. Chen, X. Li e A. McMillan, “Application and Construction of Deep Learning Networks in Medical Imaging,” *IEEE Transactions on Radiation and Plasma Medical Sciences,* p. 1, 2020.

[122]F. Li, H. Chen, Z. Liu, X.-d. Zhang, M.-s. Jiang, Z.-z. Wu e K.-q. Zhou, “Deep learning-based automated detection of retinal diseases using optical coherence tomography images,” *Biomedical Optics Express,* vol. 10, p. 6204, December 2019.

[123] J. Kim, O. Tomkins-Netzer, M. Elman, et e al., “Evaluation of a self-imaging SD-OCT system designed for remote home monitoring,” *BMC Ophthalmol,* vol. 22, p. 261.

[124] R. F. Spaide, J. G. Fujimoto e N. K. Waheed, “Image artifacts in optical coherence tomography angiography,” vol. 35, pp. 2163-2180, November 2015.

[125] S. J. Jeon, H.-Y. L. Park e C. K. Park., “Effect of Macular Vascular Density on Central Visual Function and Macular Structure in Glaucoma Patients,” *Scientific Reports,* vol. 8, nº 1, p. 16009, 2018.

[126] “What Is Macular Degeneration?,” February 2022. [Online]. Available: https://www.aao.org/eye-health/diseases/amd-macular-degeneration.

[127] J. S. Schuman, C. A. Puliafito, J. G. Fujimoto and J. S. Duker, Optical Coherence Tomography of Ocular Diseases, Third ed., SLACK Incorporated, 2012.

[128] P. Gholami, P. Roy, M. K. Parthasarathy e V. Lakshminarayanan, “OCTID: Optical coherence tomography image database,” *Computers & Electrical Engineering,* vol. 81, 2020.

[129] K. Halasz, S. Kelly, M. Iqbal, Y. Pathak e V. Sutariya, “Micro/Nanoparticle Delivery Systems for Ocular Diseases,” *ASSAY and Drug Development Technologies,* vol. 17, May 2019.

[130] J. Ruby, X. Li, T. Binford, Y. Yuan, W. Hu, Y. Yung e M. Pan, “Intelligent Detection of Glaucoma Using Ballistic Optical Imaging,” *Advanced Engineering Informatics,* vol. 40, pp. 107-127, 2019.

[131] S. E. R., “Random Forest Algorithms - Comprehensive Guide With Examples,” 15 February 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/. [Accessed 19 March 2023].

[132] S. Sotoudeh-Paima, A. Jodeiri, F. Hajizadeh e H. Soltanian-Zadeh, “Multi-scale convolutional neural network for automated AMD classification using retinal OCT images,” *Computational Biology and Medicine,* vol. 144, pp. 1-9, 2022.

[133] A. Thomas, P. M. Harikrishnan, A. Krishna, P. Ponnusamy e V. Gopi, “A novel multiscale convolutional neural network based age-related macular degeneration detection using OCT images,” *Biomedical Signal Processing and Control,* vol. 67, pp. 1-8, 2021.

[134] T. Ahonen, J. Matas, C. He e M. Pietikäinen, “Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features,” *Image Analysis,* pp. 61-70, 2009.

[135] R. Gandhi, “Naive Bayes Classifier,” Towards Data Science, 17 May 2018. [Online]. Available: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c. [Accessed 19 March 2023].

[136] M. u. Hassan, “ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks,” Neurohive, 23 January 2019. [Online]. Available: https://neurohive.io/en/popular-networks/resnet/. [Accessed 8 May 2023].

[137] R. Holbrook and A. Cook, “Overfitting and Underfitting,” [Online]. Available: https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting. [Accessed 19 March 2023].

[138] “scikit-image - Examples: Local Binary Pattern,” [Online]. Available: https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_local_binary_pattern.html. [Accessed 18 April 2023].

[139] R. Gandhi, “Support Vector Machine — Introduction to Machine Learning Algorithms,” 7 July 2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Accessed 19 March 2023].

Appendix A: HOG Feature Extraction Results

A.1 Results obtained with the best hyperparameters for each preprocessing

Preprocessing		Train	Validation	Test
1	Precision	**95.92%**	**77.28%**	**74.84%**
	Recall	**95.91%**	**77.26%**	**73.20%**
	Fbeta-Score	**95.91%**	**77.26%**	**72.74%**
	Accuracy	**95.91%**	**77.26%**	**73.20%**
2	Precision	94.92%	68.40%	71.55%
	Recall	94.89%	68.17%	69.40%
	Fbeta-Score	94.89%	68.10.%	68.48%
	Accuracy	94.89%	68.17%	69.40%
3	Precision	87.19%	62.87%	65.17%
	Recall	86.79%	62.19%	65.30%
	Fbeta-Score	86.90%	62.24%	64.82%
	Accuracy	86.79%	62.19%	65.30%
4	Precision	86.86%	61.98%	64.94%
	Recall	86.36%	61.01%	65.40%
	Fbeta-Score	86.44%	61.12%	65.04%
	Accuracy	86.36%	61.01%	65.40%

A.2 Test Results obtained with the best hyperparameters for each preprocessing

Preprocessing	Metrics	Disease				
		CNV	DME	DRUSEN	NORMAL	Average
1	Precision	**74.02%**	**63.63%**	**81.64%**	**80.08%**	**74.84%**
	Recall	**83.20%**	**84.00%**	**51.60%**	**74.00%**	**73.20%**
	Fbeta-Score	**81.18%**	**78.94%**	**55.69%**	**75.17%**	**72.74%**
	Accuracy	**83.20%**	**84.00%**	**51.60%**	**74.00%**	**73.20%**
2	Precision	65.77%	62.37%	79.03%	79.03%	71.55%
	Recall	88.40%	77.60%	39.20%	72.40%	69.40%
	Fbeta-Score	82.70%	73.98%	43.59%	73.63%	68.48%
	Accuracy	88.40%	77.60%	39.20%	72.40%	69.40%

Preprocessing	**Metrics**	**Disease**				
		CNV	DME	DRUSEN	NORMAL	Average
3	Precision	74.71%	62.83%	57.33%	65.78%	65.17%
	Recall	78.00%	83.20%	50.00%	50.00%	65.30%
	Fbeta-Score	77.31	78.13%	51.31%	52.52%	64.82%
	Accuracy	78.00%	83.20%	50.00%	50.00%	65.30%
4	Precision	75.87%	66.45%	54.93%	62.50%	64.94%
	Recall	78.00%	82.40%	51.20%	50.00%	65.40%
	Fbeta-Score	77.56%	78.62%	51.90%	52.08%	65.04%
	Accuracy	78.00%	82.40%	51.20%	50.00%	65.40%

Appendix B: LBP Feature Extraction Results

B.1 Results obtained with the best hyperparameters for each preprocessing

Preprocessing		Train	Validation	Test
1	Precision	68.49%	53.77%	53.06%
	Recall	68.46%	53.44%	49.81%
	Fbeta-Score	68.34%	53.38%	47.16%
	Accuracy	68.39%	53.24%	49.81%
2	Precision	**82.03%**	**60.47%**	**57.67%**
	Recall	**82.00%**	**60.34%**	**54.60%**
	Fbeta-Score	**81.99%**	**60.28%**	**52.64%**
	Accuracy	**82.00%**	**60.34%**	**54.60%**
3	Precision	73.97%	48.22%	48.51%
	Recall	73.09%	47.08%	47.75%
	Fbeta-Score	72.98%	46.92%	46.93%
	Accuracy	73.09%	47.08%	47.75%
4	Precision	52.24%	52.24%	48.15%
	Recall	51.18%	46.29%	48.40%
	Fbeta-Score	50.94%	46.11%	47.97%
	Accuracy	51.18%	46.29%	48.40%

B.2 Test Results obtained with the best hyperparameters for each preprocessing

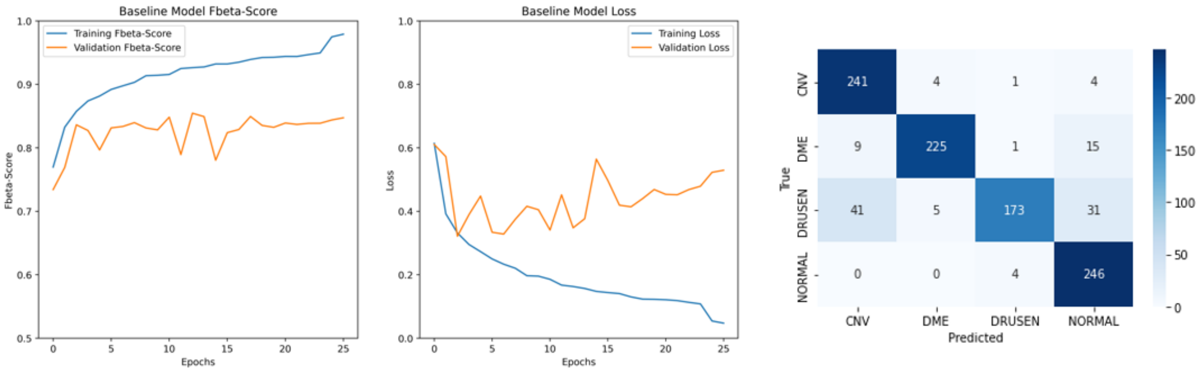
Preprocessing	Metrics	Disease				
		CNV	DME	DRUSEN	NORMAL	Average
1	Precision	62.26%	48.16%	54.38%	47.46%	53.06%
	Recall	26.40%	52.40%	24.80%	49.81%	49.81%
	Fbeta-Score	29.83%	51.49%	27.82%	47.16%	47.16%
	Accuracy	46.00%	67.60%	24.80%	95.66%	49.81%
2	Precision	**64.93%**	**49.85%**	**63.04%**	**52.85%**	**57.67%**
	Recall	**40.00%**	**70.00%**	**23.20%**	**85.20%**	**54.60%**
	Fbeta-Score	**43.32%**	**64.76%**	**26.55%**	**75.90%**	**52.64%**
	Accuracy	**40.00%**	**70.00%**	**23.20%**	**85.20%**	**54.60%**

Preprocessing	**Metrics**	**Disease**				
		CNV	DME	DRUSEN	NORMAL	Average
3	Precision	57.14%	46.03%	43.47%	47.42%	48.51%
	Recall	35.20%	69.60%	32.00%	54.20%	47.75%
	Fbeta-Score	38.12%	63.13%	33.78%	52.69%	46.93%
	Accuracy	35.20%	69.60%	32.00%	54.20%	47.75%
4	Precision	50.81%	50.91%	42.00%	48.89%	48.15%
	Recall	37.20%	66.80%	36.80%	52.80%	48.40%
	Fbeta-Score	39.30%	62.87%	37.73%	51.96%	47.97%
	Accuracy	37.20%	66.80%	36.80%	52.80%	48.40%

Appendix C: VGG16: Learning curves and Test Confusion Matrix

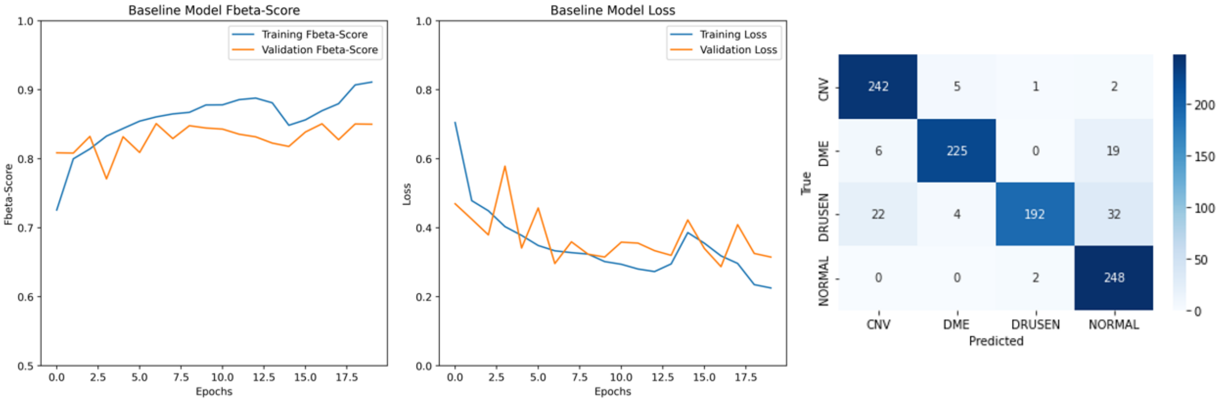
C.1 Models

Model 1



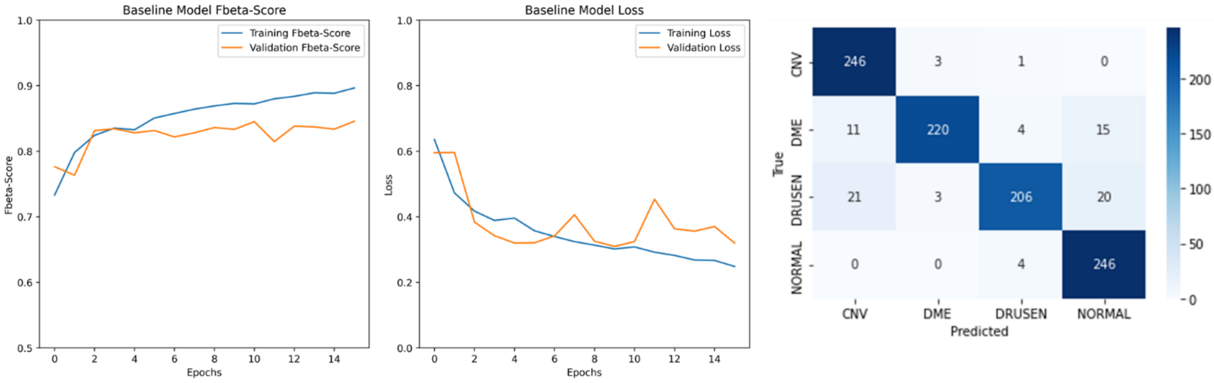
True \ Predicted	CNV	DME	DRUSEN	NORMAL
CNV	241	4	1	4
DME	9	225	1	15
DRUSEN	41	5	173	31
NORMAL	0	0	4	246

Model 2



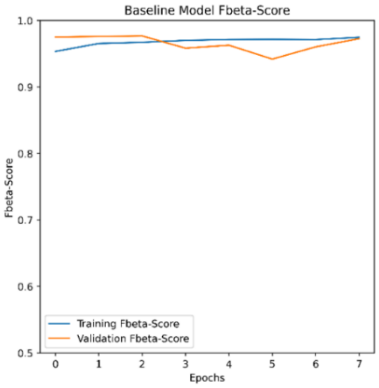
True \ Predicted	CNV	DME	DRUSEN	NORMAL
CNV	242	5	1	2
DME	6	225	0	19
DRUSEN	22	4	192	32
NORMAL	0	0	2	248

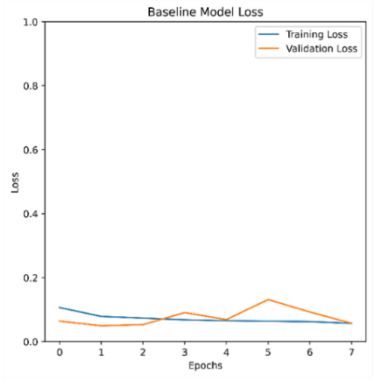
Model 3

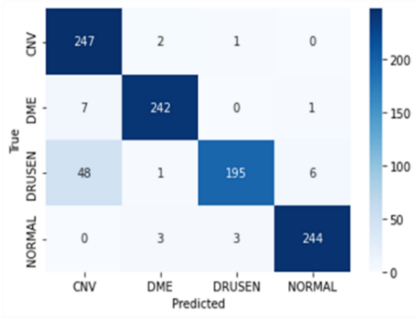


True \ Predicted	CNV	DME	DRUSEN	NORMAL
CNV	246	3	1	0
DME	11	220	4	15
DRUSEN	21	3	206	20
NORMAL	0	0	4	246

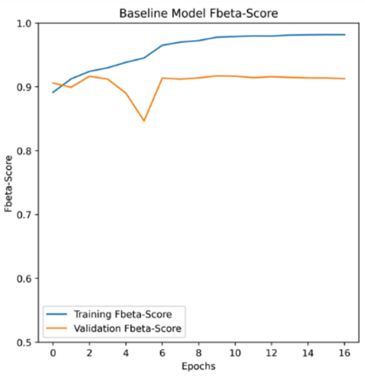
Model 4

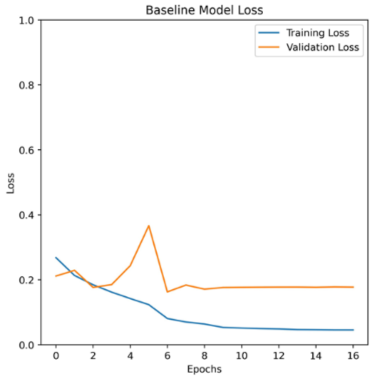


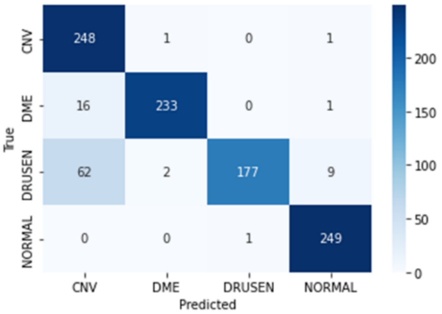




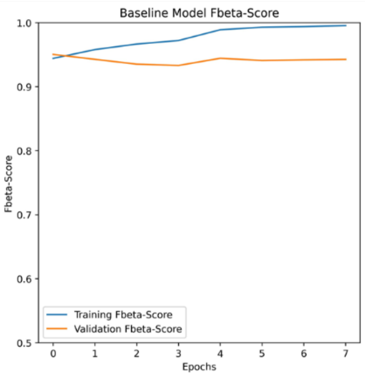
Model 5

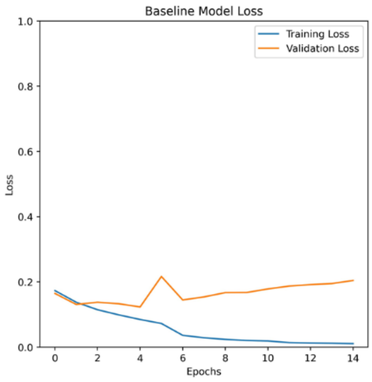


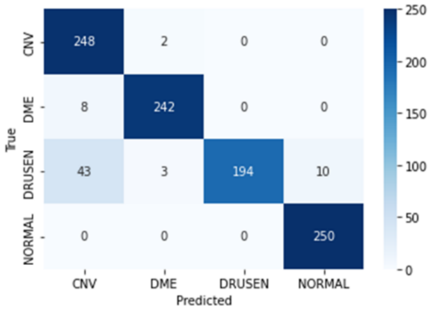




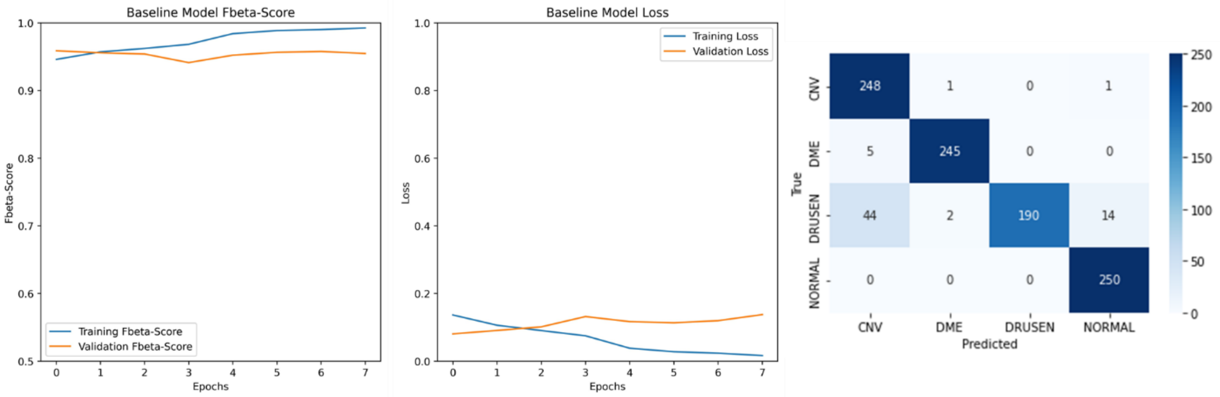
Model 6





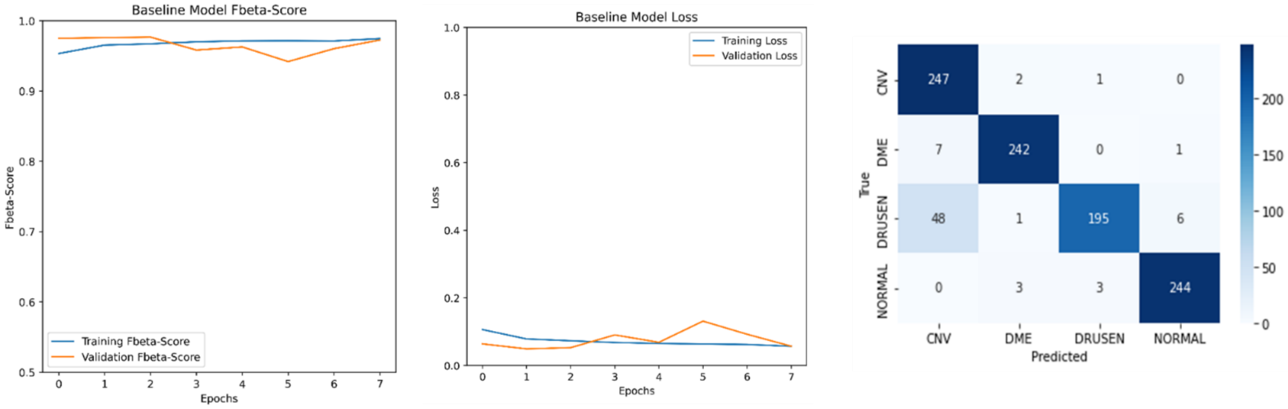


Model 7

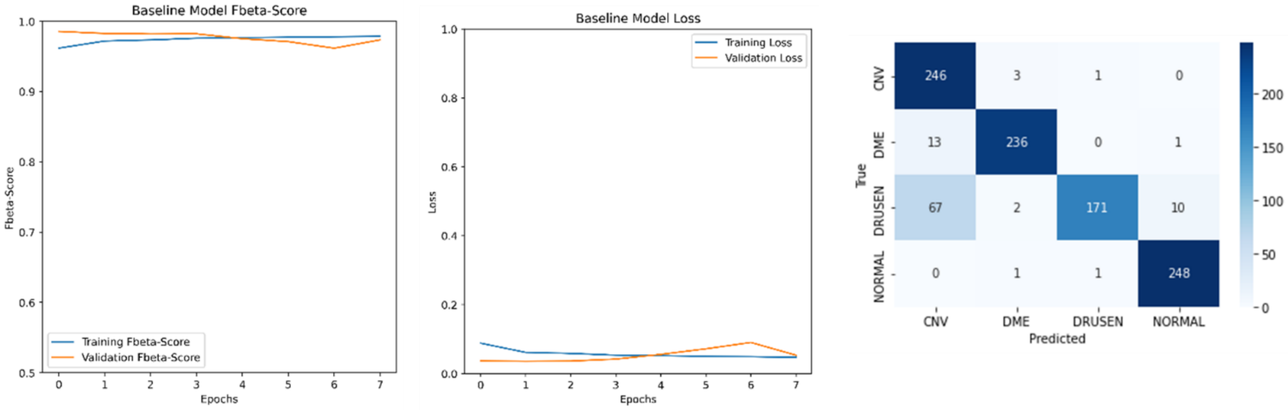


C.2 Configurations

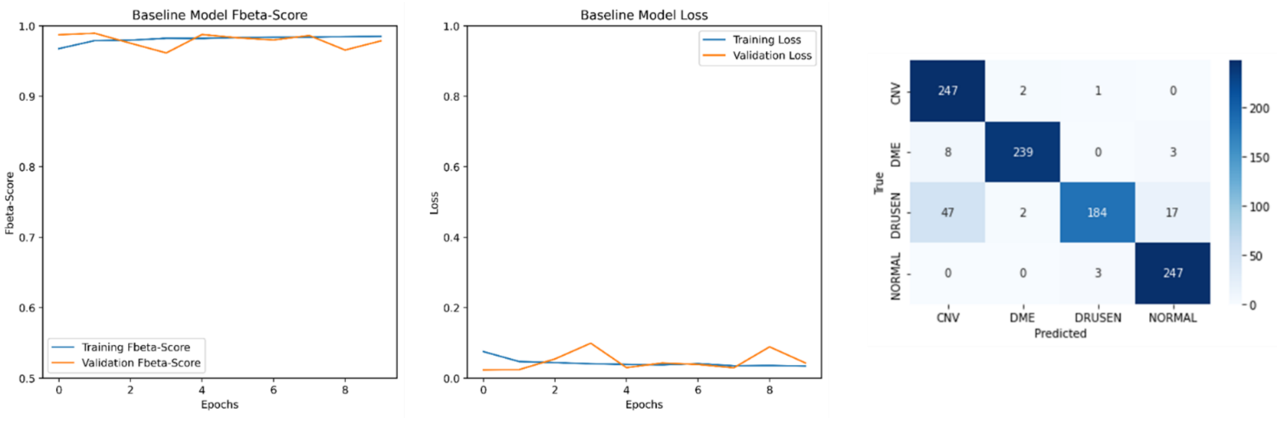
Configuration 1 (Original images)



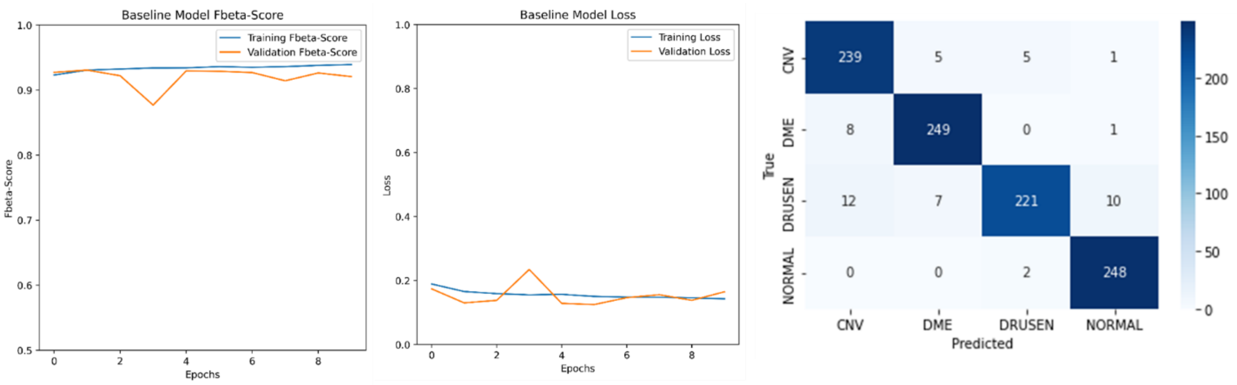
Configuration 2 (No data augmentation + Preprocessing 1)



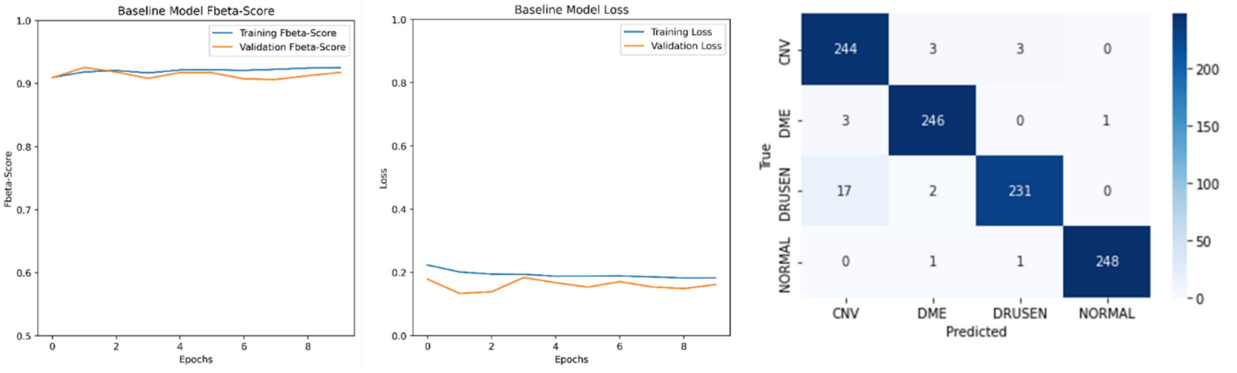
Configuration 3 (No data augmentation + Preprocessing 2)



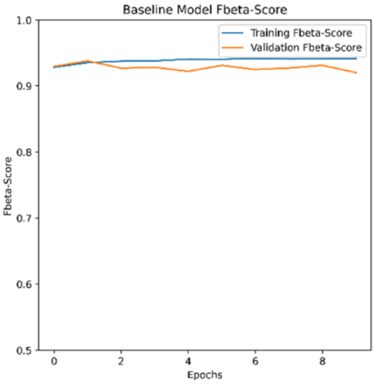
Configuration 4 (Data augmentation + Original Images)

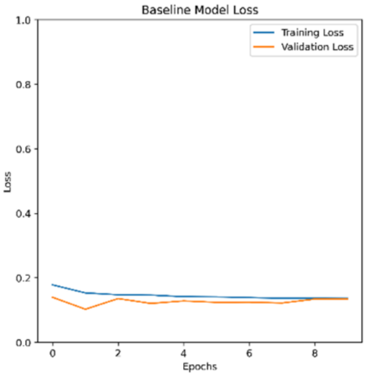


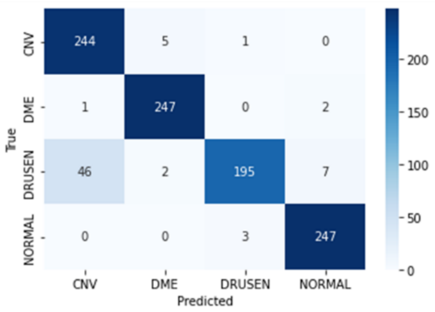
Configuration 5 (Data augmentation + Preprocessing 1)



Configuration 6 (Data augmentation + Preprocessing 2)



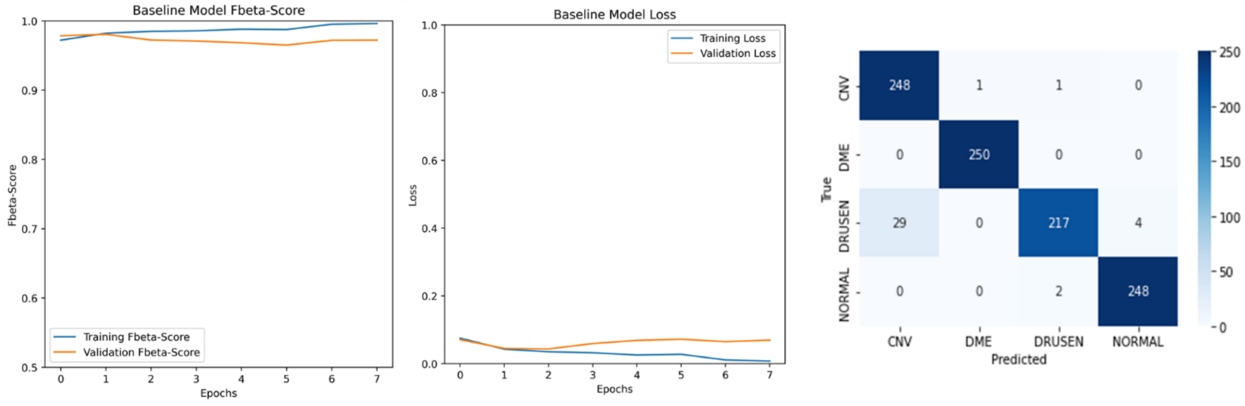




Appendix D: ResNet50V2: Learning curves and Test Confusion Matrix

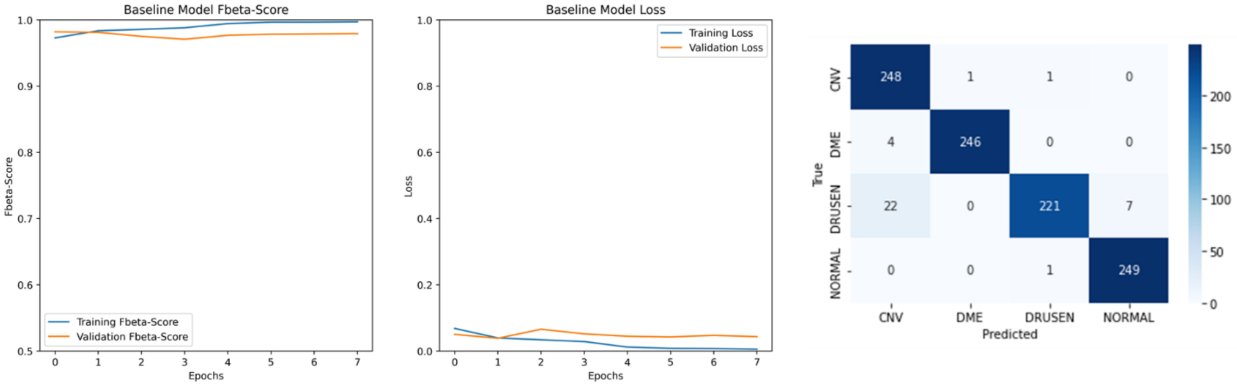
D.1 Models

Model 1



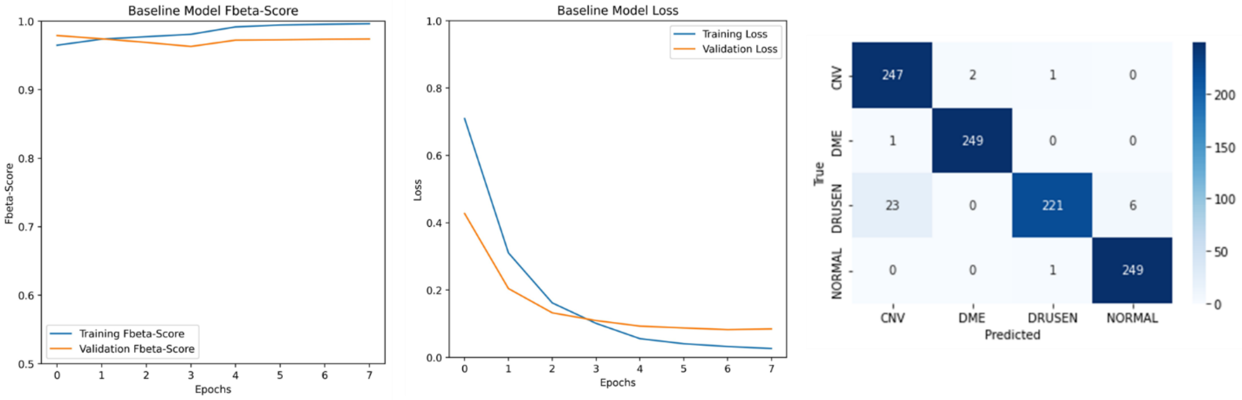
	CNV	DME	DRUSEN	NORMAL
CNV	248	1	1	0
DME	0	250	0	0
DRUSEN	29	0	217	4
NORMAL	0	0	2	248

Model 2



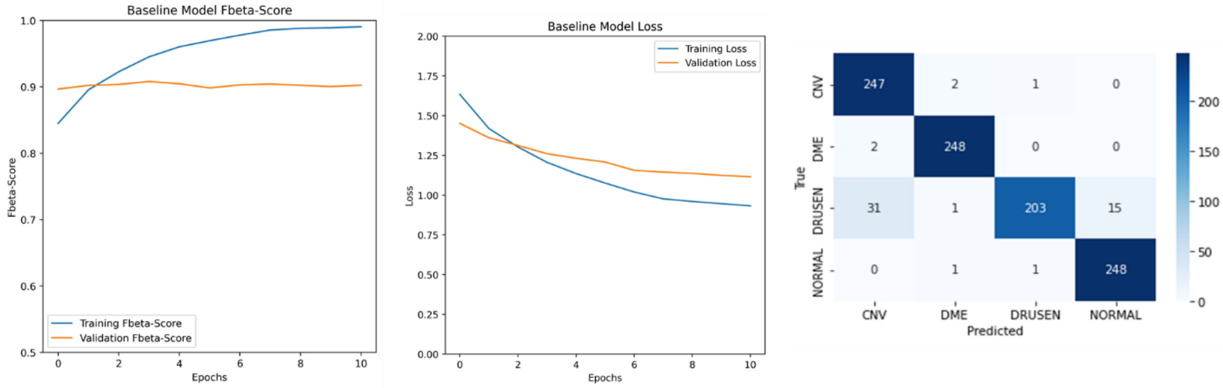
	CNV	DME	DRUSEN	NORMAL
CNV	248	1	1	0
DME	4	246	0	0
DRUSEN	22	0	221	7
NORMAL	0	0	1	249

Model 3



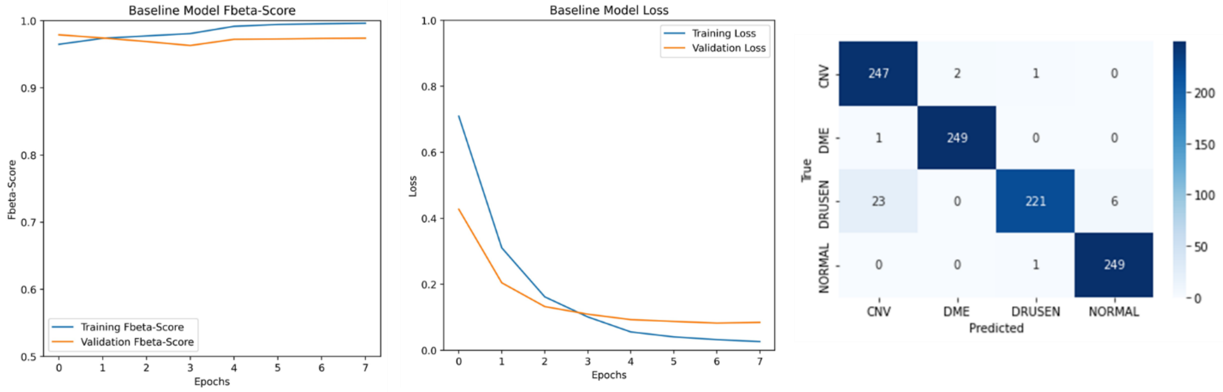
	CNV	DME	DRUSEN	NORMAL
CNV	247	2	1	0
DME	1	249	0	0
DRUSEN	23	0	221	6
NORMAL	0	0	1	249

Model 4

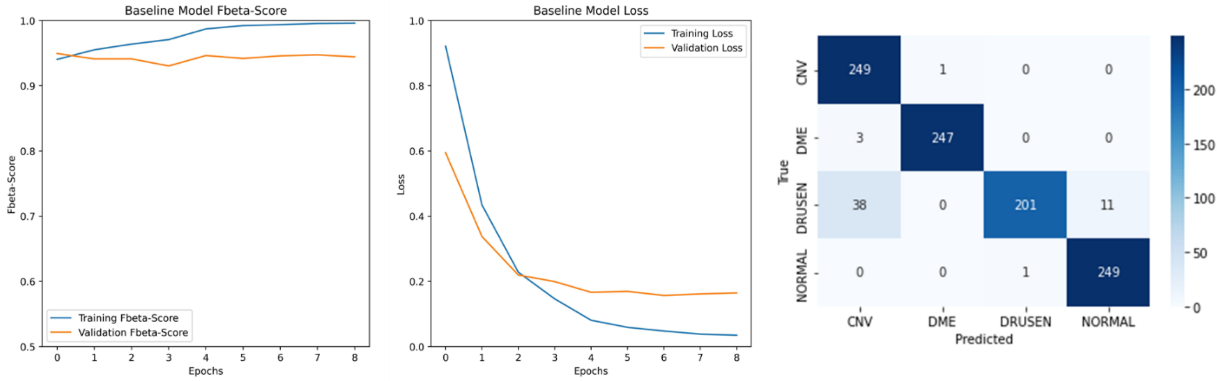


D.2 Configurations

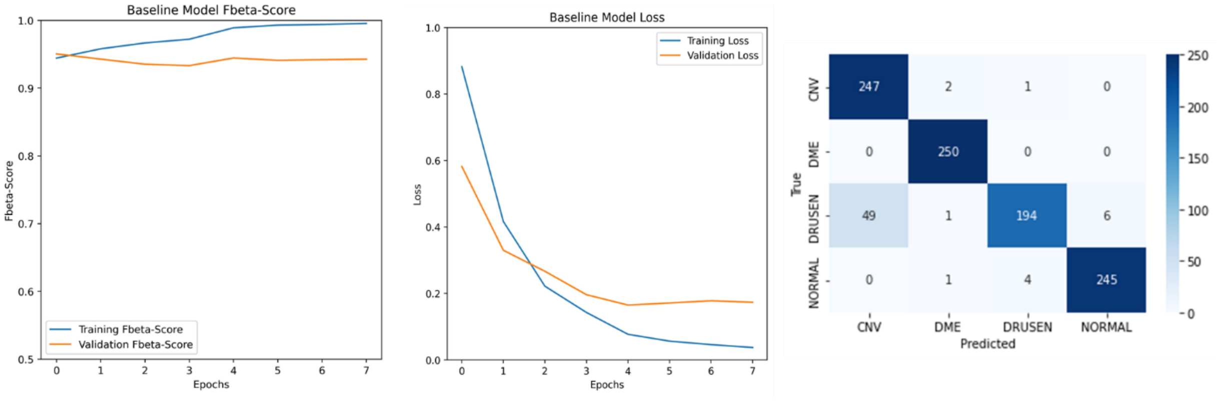
Configuration 1 ((Original images)



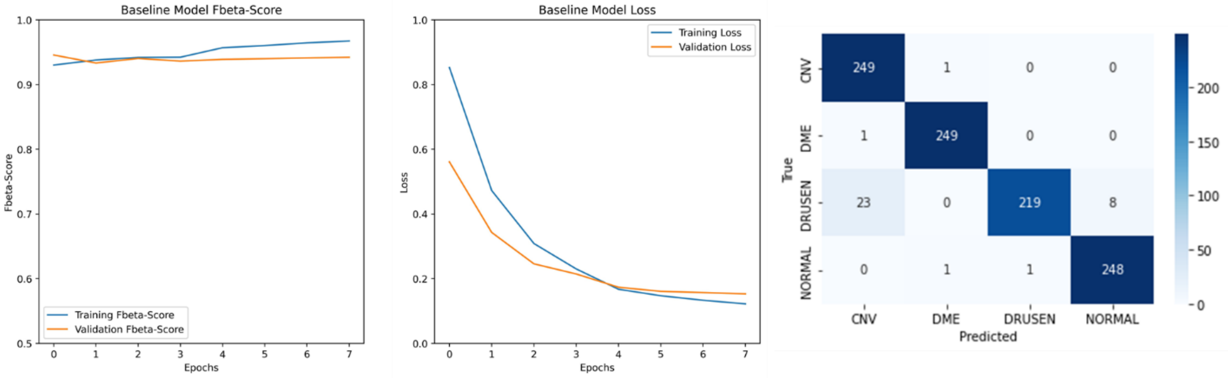
Configuration 2 (No data augmentation + Preprocessing 1)



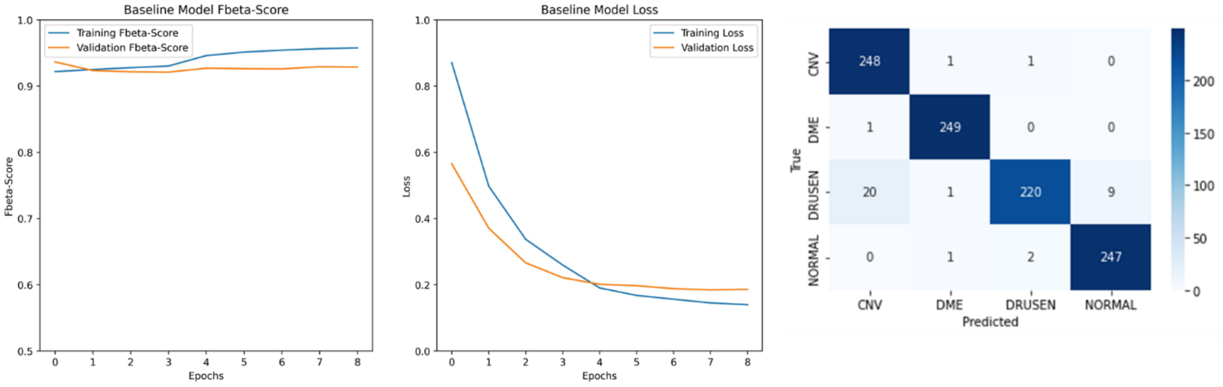
Configuration 3 (No data augmentation + Preprocessing 2)



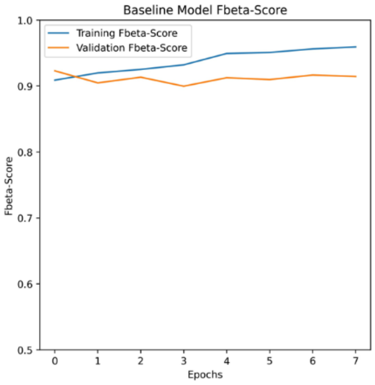
Configuration 4 (Data augmentation + Original Images)

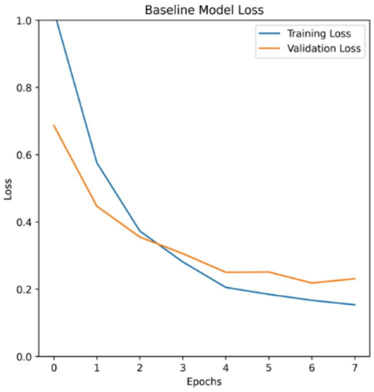


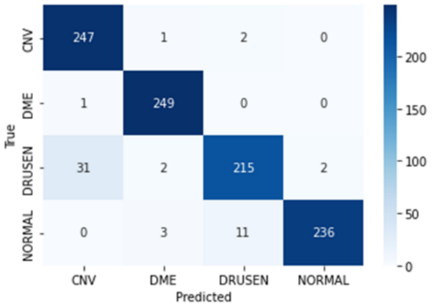
Configuration 5 (Data augmentation + Preprocessing 1)



Configuration 6 (Data augmentation + Preprocessing 2)



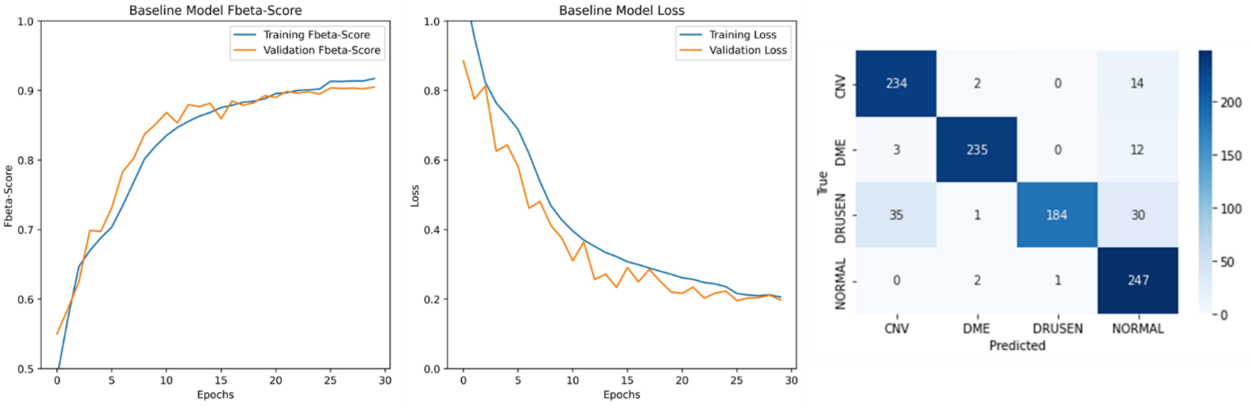




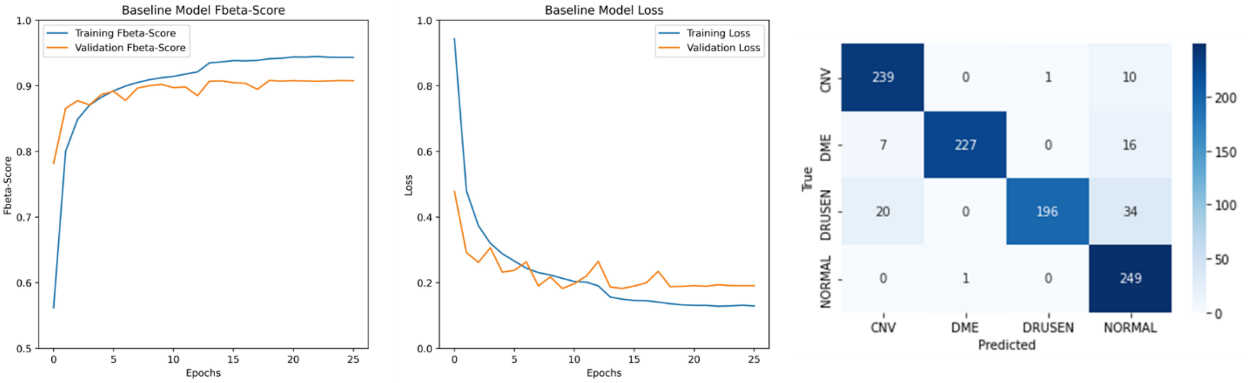
Appendix E: Proposed Model: Learning curves and Test Confusion Matrix

E.1 Models

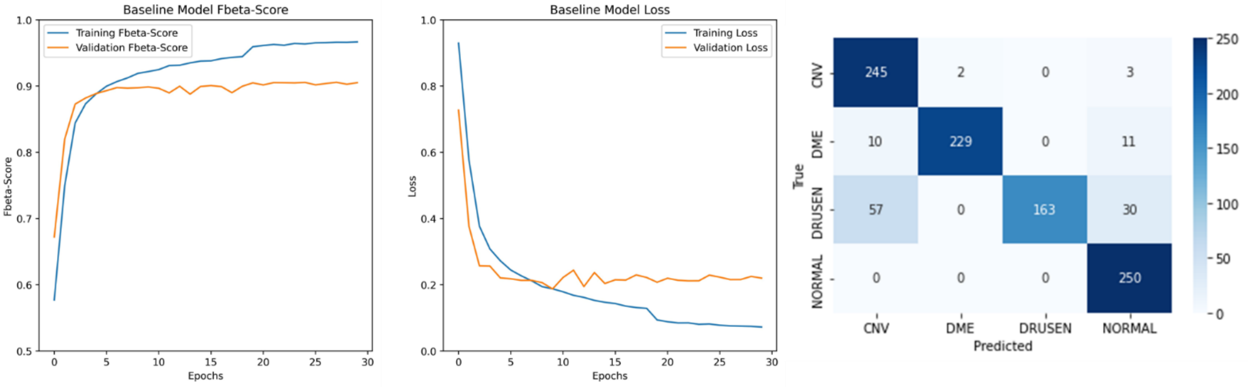
Model 1



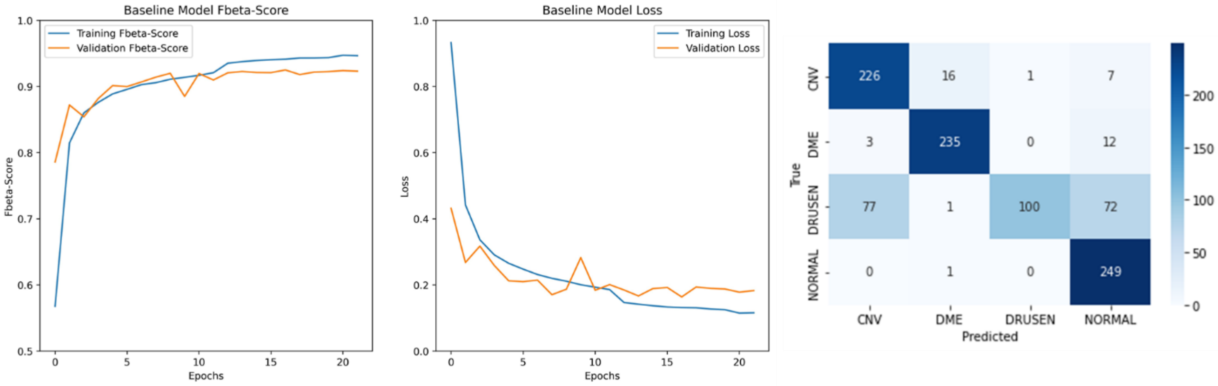
Model 2



Model 3

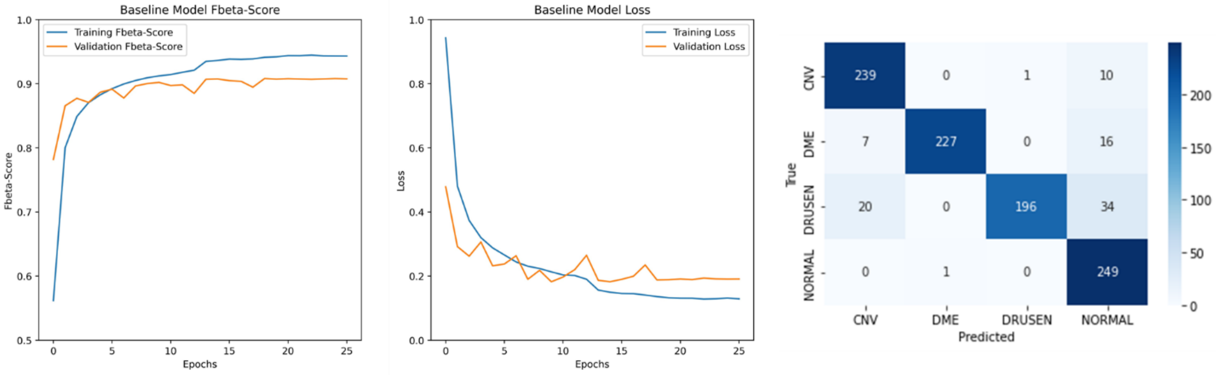


Model 4

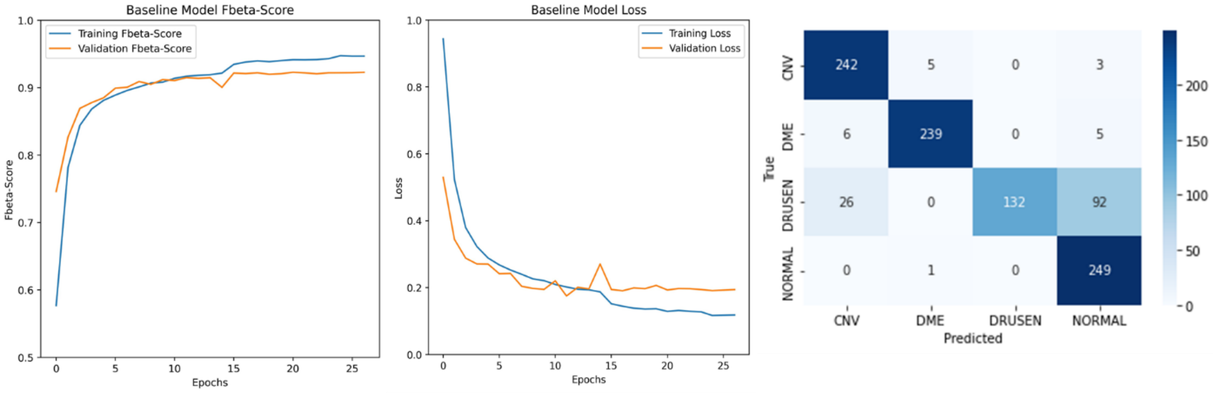


E.2 Configurations

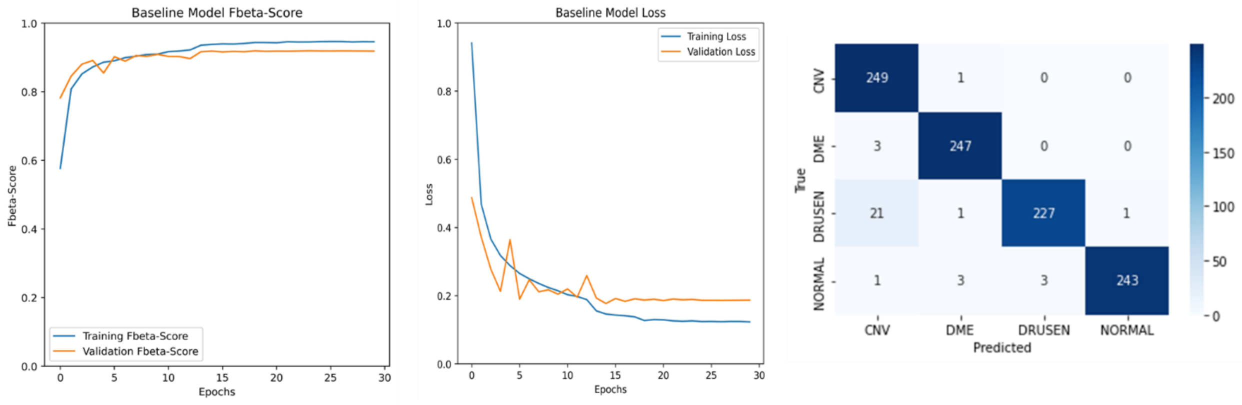
Configuration 1 (Original images)



Configuration 2 (No data augmentation + Preprocessing 1)

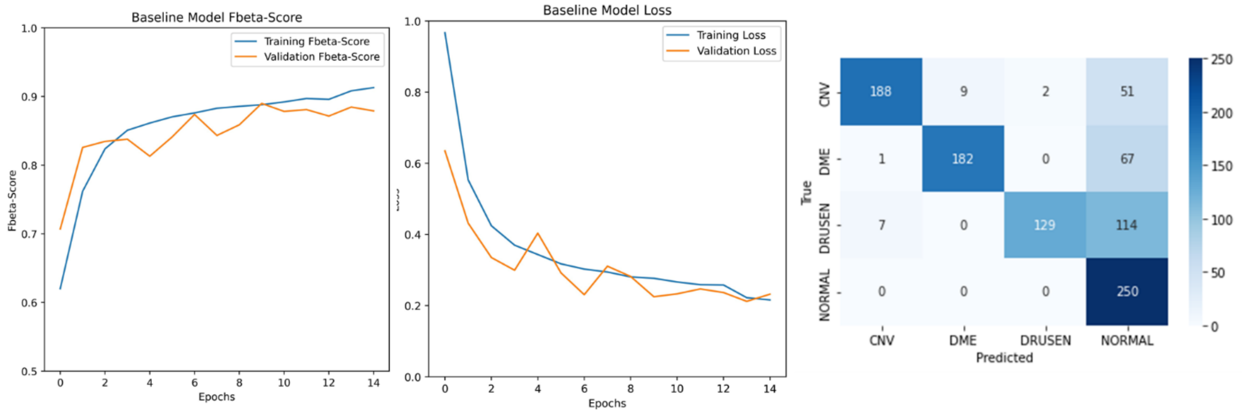


Configuration 3 (No data augmentation + Preprocessing 2)



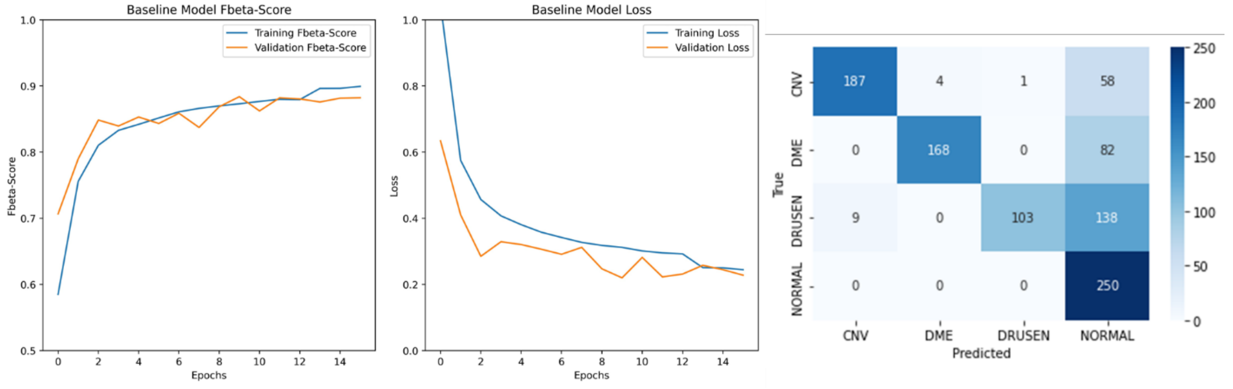
True \ Predicted	CNV	DME	DRUSEN	NORMAL
CNV	249	1	0	0
DME	3	247	0	0
DRUSEN	21	1	227	1
NORMAL	1	3	3	243

Configuration 4 (Data augmentation + Original Images)



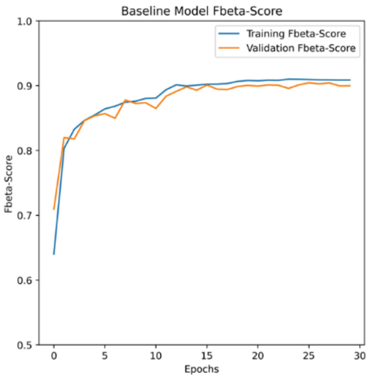
True \ Predicted	CNV	DME	DRUSEN	NORMAL
CNV	188	9	2	51
DME	1	182	0	67
DRUSEN	7	0	129	114
NORMAL	0	0	0	250

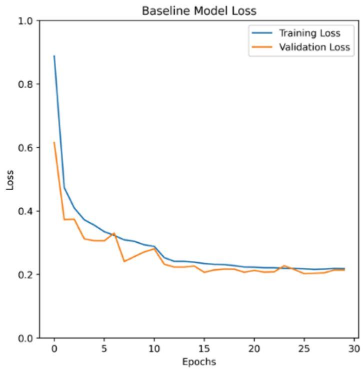
Configuration 5 (Data augmentation + Preprocessing 1)

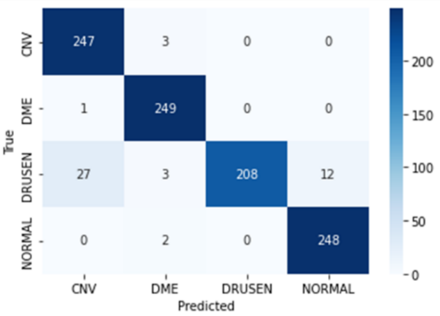


True \ Predicted	CNV	DME	DRUSEN	NORMAL
CNV	187	4	1	58
DME	0	168	0	82
DRUSEN	9	0	103	138
NORMAL	0	0	0	250

Configuration 6 (Data augmentation + Preprocessing 2)


Baseline Model Fbeta-Score
Training Fbeta-Score
Validation Fbeta-Score
Fbeta-Score
Epochs


Baseline Model Loss
Training Loss
Validation Loss
Loss
Epochs


CNV
DME
DRUSEN
NORMAL
True
CNV
DME
DRUSEN
NORMAL
Predicted