People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Yahia Fares University of Médéa
Faculty of Sciences
Department of Mathematics and Computer Science



A Master's Thesis

In Fulfillment Of The Requirements To Obtain A Master's Degree

Computer Science (Systems Engineering And Web Technologies)

Deep Learning Based Diagnostic-Aid System For Ophthalmic (Eye) Diseases

Presented by:
BEKOUACI Mohamed Salim
LOUDJEDI Wail Faycal

Under the Direction of :
Mr. **BOUDJEBBOUR** Karim
Mr. **TCHOKETCH KEBIR** Sofiane

Academic Year: 2023/2024

Acknowledgment

We would like to extend our deepest gratitude to all those who have supported and guided us throughout our journey in pursuing our master's degree. First and foremost, we thank our professors, whose expertise, encouragement, and dedication have been instrumental in our academic growth. Your commitment to excellence and passion for teaching have inspired us to strive for our best.

A special thanks to our Professor **BOUDJEBBOUR Karim** , who introduced us to the world of machine learning in the *Data Mining* class during the first semester of our Master's program. We deeply appreciate his approach of teaching machine learning through mathematical calculations and tackling complex challenges through pen and paper, which significantly strengthened our grasp of the theoretical foundations, methodologies, and mathematical principles underlying this field. Your guidance has been invaluable in shaping our understanding and nurturing our interest in this domain.

We extend our heartfelt gratitude to Professor **TCHOKETCH KEBIR Sofiane** for his invaluable contributions to this thesis. Professor Tchoketch's expertise in image processing, medical image segmentation, and deep learning has profoundly enriched our understanding and shaped the development of our research. His guidance and mentorship have been pivotal in refining our methodologies.

Our sincere appreciation goes to Professor **BOUDOUKHANI Nesrine** for helping us understand and apply Convolutional Neural Networks in the biomedical field, and for interpreting the results with us. Your practical approach and support have been crucial in our academic journey.

Additionally, we express our gratitude to the department staff and administrative personnel. Your hard work behind the scenes has ensured a smooth and enriching academic experience. Your assistance and professionalism have been invaluable, and we are deeply appreciative of your efforts.

We are also grateful to the university as a whole, which has provided a helpful and productive environment for learning and pursuit for more knowledge. From our first year to these final days, the university has been a pillar of support.

To our colleagues and fellow students, your camaraderie, and collaborative spirit have made this journey not only fruitful but also enjoyable. The friendships and experiences we have shared are memories we will cherish forever.

Thank you all for being a part of this incredible journey. Your support has made this achievement possible, and we are forever grateful.

Dedication

I am forever grateful to Allah, the compassionate, for his guidance and support throughout my years of study, which helped me succeed over every challenge and setback, as well as for lighting the way to success in my pursuit. As the final chapter of my academic career comes to an end, I'd like to dedicate this report to everyone who has helped me along the way, including my loved ones Yoomi, all of my colleagues, the faculty, and staff at this university, and especially my family. I extend my heartfelt gratitude to everyone who has placed their trust in me, believing in my capacity for research, creativity, and the pursuit of understanding to foster a better tomorrow.

By LOUDJEDI Wail Faycal

Dedication

I dedicate this thesis to my beloved **Parents** and **Grandparents**, whose unwavering love and support have been my foundation. To my wonderful siblings, my steadfast companions, whose presence is always a source of strength. To my great uncle, **ZOUAOUINE Hicham**, who was a previous mentor and teacher at this university. It is through his guidance in IT and computer science that I found my path. Without his teachings on computers and typing during my childhood, I would not have taken on this career. To my dear friend **Esma**, whose invaluable tips, friendship, and support have greatly contributed to this work. Also to the Biology Professor **HFAIRI Saida** for her invaluable tips and unwavering aid throughout my academic journey at this university, guiding me through its challenges. To all those who have assisted me, near and far, your encouragement and help have been indispensable. Lastly, I dedicate this thesis to everyone who believes in me, both family and friends. Your faith in me has been my greatest strength.

By BEKOUACI Mohamed Salim

Abstract

Diabetic Retinopathy is a global health problem, affecting more than 100 million people globally. It is the most prevalent complication caused by diabetes and has always been one of the leading causes of blindness. Early detection holds significant importance in safeguarding patients from vision loss and irreversible retinal damage by slowing the progression of the disease and treating it in time. With early diagnosis and treatment, the risk of severe vision loss is reduced by 90%. However, early detection remains a challenging task, as patients are often asymptomatic until proliferate stages, and many lesions are often undetectable to the human eye in their early stages and therefore easily overlooked, even by a retinal specialist. In this work, we propose a system that automates the diagnosis of Diabetic Retinopathy using deep learning techniques, particularly convolutional neural networks (CNNs), and aims to identify its presence in its early stages. Additionally, a retinal blood vessel segmentation system using the U-Net architecture was developed to enhance the analysis of the vascular layer of the eye by ophthalmologists and surgeons, aiding in the treatment of Diabetic Retinopathy and other diseases affecting the vascular layer. The detection model achieved up to 93.1% in confidence of correctly grading the stage of Diabetic Retinopathy.

KEYWORDS: Ophthalmic Diseases, Diabetic Retinopathy, Computer Aided Diagnosis, Medical Image classification, Deep Learning, Convolutional Neural Networks, Transfer Learning, Fine Tuning, Bio-medical image segmentation

Resumé

La rétinopathie diabétique est un problème de santé mondial qui touche plus de 100 millions de personnes dans le monde. Il s'agit de la complication la plus répandue du diabète et a toujours été l'une des principales causes de cécité. La détection précoce revêt une importance capitale pour protéger les patients contre la perte de vision et les lésions rétiniennes irréversibles en ralentissant la progression de la maladie et en la traitant à temps. Avec un diagnostic et un traitement précoces, le risque de perte de vision grave est réduit de 90%. Cependant, la détection précoce reste une tâche difficile, car les patients sont souvent asymptomatiques jusqu'aux stades de prolifération, et de nombreuses lésions sont souvent indétectables à l'œil humain et donc facilement négligées, même par un spécialiste de la rétine. Dans ce travail, nous proposons un système qui automatise le diagnostic de la rétinopathie diabétique à l'aide de techniques d'apprentissage profond, notamment les réseaux de neurones convolutifs (CNN), et vise à identifier sa présence à ses débuts. De plus, un système de segmentation des vaisseaux sanguins rétiniens utilisant l'architecture U-Net a été développé pour améliorer l'analyse de la couche vasculaire de l'œil par les ophtalmologistes et les chirurgiens, contribuant ainsi au traitement de la rétinopathie diabétique et d'autres maladies affectant la couche vasculaire. Le modèle de détection a atteint un niveau de précision de 93,1% dans la détection et l'évaluation correcte du stade de la rétinopathie diabétique.

MOTS-CLÉS: Maladies ophtalmologiques, Rétinopathie Diabétique, Diagnostic Assisté Par Ordinateur, Classification des Images Médicales, Apprentissage Profond, Apprentissage par Transfert, Réseaux de Neurones Convolutifs, Segmentation d'Images Médicales.

ملخص

اعتلال الشبكية السكري هو مشكلة صحية عالمية تؤثر على أكثر من 100 مليون شخص. يُعتبر من أكثر المضاعفات شيوعًا الناتجة عن مرض السكري ويُعد دائمًا أحد الأسباب الرئيسية للعمى. للكشف المبكر أهمية كبيرة في حماية المرضى من فقدان البصر والأضرار غير القابلة للعلاج في الشبكية من خلال التحكم في تقدم المرض ومعالجته، حيث يقلل العلاج المبكر من خطر الإصابة بالعمى بنسبة 90%. ومع ذلك، يبقى الكشف المبكر مهمة صعبة، حيث أن العديد من الحالات غالبًا ما تكون غير قابلة للكشف عند الفحص الطبي بالعين المجردة في مراحلها الأولى وبالتالي يتم تجاهلها بسهولة. في هذا العمل، نقترح نظامًا يعتمد على الذكاء الاصطناعي و بالأخص آلية التعلم العمق ليقوم آليا بتشخيص اعتلال الشبكية السكري و تسهيل الكشف عنه في مراحله الأولى. بالإضافة إلى ذلك، تم تطوير نظام باستخدام تقنيات الذكاء الاصطناعي لاستخراج الأوعية الدموية في الشبكية وعزلها لتسهيل وتحسين تحليل شبكات الأوعية الدموية للعين، مما يساعد في اكتشاف اعتلال الشبكية السكري وأمراض أخرى شائعة. حقق نظام الكشف المعتمد على التعلم العميق دقة تصل إلى 93.1% في تشخيص درجة المرض.

<u>الكلمات الرئيسية :</u> أمراض العين, اعتلال الشبكية السكري, التشخيص المساعد بالحاسوب, معالجة الصور بالذكاء الاصطناعي, التعلم العميق

Contents

1 Introduction **15**
Challenges . . . 17
Stages Of Diabetic Retinopathy . . . 21
Objectives . . . 24
1.0.1 Automated Diabetic Retinopathy Detection And Stage Grading System . . . 24
1.0.2 Multi Retinal Blood Vessel Segmentation Systems . . . 25
Literature Review . . . 28
Thesis Structure . . . 30

2 Fundamental Concepts **31**
Artificial Intelligence . . . 31
2.1 Machine Learning (ML) . . . 32
2.2 Deep Learning . . . 35
2.2.1 Artificial Neural Networks . . . 37
2.2.2 Structure of an Artificial Neuron . . . 39
2.2.3 The Multi-Layer Perceptron (MLP) . . . 39
2.2.4 Activation Functions in Neural Networks . . . 40
2.3 How Does the Model Learn and Adjust Itself? . . . 41
2.3.1 Regularisation Methods . . . 42
2.4 Deep Learning Models . . . 43
2.5 U-NET Architecture . . . 51
2.6 Transfer Learning . . . 54
2.6.1 Key Components of Transfer Learning . . . 54
Conclusion . . . 56

3 Conception And Implementation Of The Diagnosis System **57**
3.1 History and Evolution of Image Classification Systems . . . 57
3.1.1 Early Beginnings: Pattern Recognition (1950s-1960s) . . . 58
3.1.2 Introduction of Artificial Neural Networks (1960s-1980s) . . . 58
3.1.3 Handcrafted Feature Extraction (1980s-2000s) . . . 58
3.1.4 Rise of Deep Learning (2010s-present) . . . 59
3.1.5 Beyond Traditional Image Classification . . . 59
3.1.6 Ongoing Developments and Future Directions . . . 59
3.2 Lesions And Medical Context . . . 60
3.3 Data Collection . . . 63
3.4 Data Preprocessing: . . . 66
3.5 Data Augmentation . . . 68

3.6 Implementation And Conception . . . 70
3.6.1 Development Environment . . . 70
3.6.2 Software Specifications . . . 70
3.7 Evaluation Metrics . . . 71
3.7.1 Accuracy . . . 71
3.7.2 Precision . . . 72
3.7.3 Recall (also known as Sensitivity or True Positive Rate In Binary Classifications) . . . 72
3.7.4 F1 Score . . . 72
3.7.5 Cohen Kappa Score . . . 72
3.8 Model Training And Discussion . . . 73
3.8.1 Performance of The Different Models in The Test Dataset . . . 80
3.8.2 Class Metrics Discussion . . . 84
3.9 Architectural Concepts of DenseNet169 . . . 88
3.9.1 Feature Reuse: . . . 90
3.9.2 Dense Blocks . . . 91
3.9.3 Transition Layers . . . 91
3.9.4 Batch Normalisation . . . 91
3.10 Model Inference . . . 93
Conclusion . . . 94

4 Conception And Implementation Of Retinal Blood Vessel Segmentation System 96
4.1 Image Segmentation, Types And Approaches . . . 96
4.1.1 Semantic Segmentation . . . 96
4.1.2 Instance Segmentation . . . 97
4.1.3 Unsupervised Segmentation Approach . . . 97
4.1.4 Object Detection . . . 98
4.1.5 Key challenges In Image Segmentation tasks . . . 99
4.2 Data Collection . . . 99
4.2.1 First Model Dataset (Macula Centered Fundus Images) . . . 99
4.2.2 Second Model Dataset (Optic Nerve Centered Fundus Images) . . . 100
4.3 Additional Evaluation Metrics . . . 101
4.3.1 Dice Coefficient . . . 102
4.3.2 Area Under the Curve (AUC) . . . 102
4.3.3 Specificity . . . 102
4.4 Data Preprocessing . . . 103
4.5 Data Augmentation . . . 103
4.6 Model Training . . . 103
4.7 Results And Discussion Of The First Model (Macula Centred) . . . 106
4.7.1 Visual Segmentation Samples From the First Model . . . 106
4.7.2 Accuracy Clarification . . . 108
4.7.3 Recall And Precision . . . 108
4.7.4 Precision . . . 110
4.7.5 Specificity . . . 110
4.7.6 DICE . . . 111
4.7.7 AUC . . . 111
4.8 Results And Discussion Of The Second Model (Optic Nerve Centered) . . . 113

4.9 Training phase charts of both the models 118
4.9.1 Stabilization of the learning curve 119
Conclusion 120

General Conclusion **122**

Bibliography **133**

List of Figures

1.1 Different fundus cameras used in the screening of the retina 18
1.2 First image of a manual screening of the retina using a fundus camera . . 18
1.3 Second image of a manual screening of the retina using a fundus camera 19
1.4 Inconsistency Of Graders 20
1.5 Simplified illustrative representation of the difference between a healthy eye and an eye of a patient with Diabetic Retinopathy 22
1.6 Fundus view difference between a healthy and an eye with the condition 22
1.7 Retinal fundus images of different stages of Diabetic Retinopathy 23
1.8 Different View Angles of Fundus Imaging 26
1.9 Semantic Segmentation of Retinal Blood Vessels 27

2.1 A Figure of expert base systems 32
2.2 Figure showing a simplified explanation of the difference between AI and Machine Learning 33
2.3 Figure illustrating the different type of machine learning approaches . . . 34
2.4 Venn diagram illustrating how deep learning is a type of representation learning, which in turn is a type of machine learning, with specific use cases 36
2.5 Difference of feature extraction between Deep Learning And Traditional Machine Learning 36
2.6 Illustration of a neural network 37
2.7 Biological Neural Networks vs Artificial Neural Networks 38
2.8 Structure Of The Artificial Neuron 39
2.9 Architecture Of Multi Layer Perceptron 40
2.10 Abstract Illustration of the backpropagation algorithm 41
2.11 Formula of the adjustment of weights during training by the loss function optimizer 42
2.12 Visualization depicting the impact of Dropout regularization technique in neural networks. 43
2.13 Visual illustration of the conv layer taking features from an input image after applying the multiplication of pixel values with the kernel 45
2.14 Evolution of feature representation hierarchy in convolutional layers. . . 46
2.15 Max pooling operation on the feature map 48
2.16 Average pooling operation on the feature map 48
2.17 Illustrative chart of the flatten layer 50
2.18 U-Net architecture 51

3.1 Most important techniques applied on medical imaging 58
3.2 Image indicating the presence of microaneurysm 60
3.3 Image of soft exudates 61

3.4 Image of Hard Exudates . . . 61
3.5 Image of fundus image indicating the presence of hemorrhaging in the eye across different places . . . 62
3.6 Image showing the growth of abnormal new vessels near the optic disc . . 63
3.7 Image of growth of new vessels in the middle of the eye . . . 63
3.8 Distribution of APTOS 2019 Dataset . . . 64
3.9 Distribution of the final dataset . . . 66
3.10 Before and after the application of CLAHE Contrast enhancement . . . 67
3.11 Before and after the application of CLAHE Contrast enhancement . . . 67
3.12 Application of a shearing on the image . . . 69
3.13 Application of a random rotation on the image . . . 70
3.14 Flowchart diagram of the conception of the model . . . 74
3.15 Simplified illustration of transfer learning process . . . 75
3.16 Graph showing an example of local and global minima in function with three variables . . . 76
3.17 Graph showing an example of local and global minima in function with two variables . . . 77
3.18 Normalized Confusion Matrix Of the DenseNet169 Model . . . 81
3.19 Normalized Confusion Matrix Of the ResNet101 Model . . . 82
3.20 Normalized Confusion Matrix Of the InceptionV3 Model . . . 82
3.21 Normalized Confusion Matrix Of the VGG19 Model . . . 83
3.22 Bar Chart Of Performance Metrics Of The Different CNN Models after Fine Tuning . . . 83
3.23 Visual illustration of the way precision and recall are measured . . . 85
3.24 Visual illustration on an example of what can impact the recall and precision 86
3.25 Early stages confusion matrix . . . 88
3.26 Architecture Of Dense Blocks . . . 89
3.27 Specifications of layers used in thearchitecture of DenseNet169 . . . 90
3.28 Batch Normalization in CNN Models . . . 93
3.29 First image of inference . . . 94
3.30 2nd image of inference . . . 94
3.31 3rd image of inference . . . 94
3.32 4th image of inference . . . 94
3.33 Images of the model's inference . . . 94

4.1 An image showing the process of semantic segmentation using encoder decoder neural networks . . . 97
4.2 Figure showing the key difference between instance and semantic segmentation . . . 97
4.3 An example of clustering algorithm results . . . 98
4.4 An example of a simple image where the model was trained to cluster pixels into different categories . . . 98
4.5 Object detection example and differences with other approaches . . . 98
4.6 A sample of the first model's dataset . . . 100
4.7 A sample of CHASE Dataset . . . 101
4.8 Architecture Of the Unet Model . . . 104
4.9 Illustration of the Transposed Convolution Operation . . . 105

4.10 First sample of Retinal vessel segmentation using the Macula centered images . . . 106
4.11 Second sample of Retinal vessel segmentation using the Macula Centered fundus image . . . 107
4.12 3rd sample of Retinal vessel segmentation using the Macula Centered fundus image . . . 107
4.13 4th sample of Retinal vessel segmentation and extraction . . . 107
4.14 5th sample of Retinal vessel segmentation and extraction . . . 108
4.18 A graph showing the ROC curve and AUC of the first model . . . 112
4.19 First example of the segmentation . . . 113
4.20 Second example of the segmentation . . . 113
4.21 AUC ROC Curve of the Second Model . . . 114
4.22 First example of training data of optic disc centered dataset . . . 115
4.23 Second example of training data of optic disc centered dataset . . . 115
4.24 3rd example of training data of optic disc centered dataset . . . 116
4.25 3rd sample of optic disc centered segmentation . . . 116
4.26 4th sample of optic disc centered segmentation . . . 117
4.27 5th sample of optic disc centered segmentation . . . 117
4.28 6th sample of optic disc centered segmentation . . . 117
4.29 7th sample of optic disc centered segmentation . . . 118
4.30 First model training metrics chart during the first iteration . . . 118
4.31 Second model training metrics chart during the first iteration . . . 119
4.32 Loss chart of the secondary iterations with a lower learning rate . . . 120
4.33 Concept of Diabetic Retinopathy grading model that detects the presence of lesions, shows the location in the image and interprets the findings . . . 123
4.34 Concept of instance segmentation of retinal blood vessels, specifying which are the veins and which are the arteries . . . 124

List of Tables

3.1 Distribution of Classes in APTOS dataset 64
3.2 Classes distribution in the final data set 65
3.3 Comparison of Accuracy and F1 Score with and without CLAHE contrast enhancement 68
3.4 Comparison of Model Total Parameters and Accuracy In the ImageNet Benchmark 74
3.5 Training Parameters 78
3.6 Performance Metrics of The Different CNN Models after Fine Tuning . . 80
3.7 Table showing DenseNet169 class metrics 80
3.8 Table showing ResNet101 class metrics 80
3.9 Table showing InceptionV3 class metrics 81
3.10 Classification Report VGG19 class metrics 81
3.11 Overall performance metric on the early stage diagnosis 87
3.12 Early Stages Classification Report 87

4.1 Performance Metrics on the Macula Centered Dataset 106
4.2 Performance metrics for the optic nerve-centered dataset. 113

Lists Of Acronyms And Glossary

AI Artificial Intelligence

ANN Artificial Neural Network

APTOS Asia Pacific Tele-Ophthalmology Society

CAD Computer Aided Diagnosis

CLAHE Contrast Limited Adaptive Histogram Equalization

CNN Convolutional Neural Network

CONV Convolution

DL Deep Learning

DM Diabetes Mellitus

DNN Deep Neural Network

DR Diabetic Retinopathy

ETDRS Early Treatment Diabetic Retinopathy Study

FCL Fully Connected Layer

KNN K-Nearest Neighbors

LR Learning Rate

MA Microaneurysms

ML Machine Learning

MLP Multilayer Perceptron

NPDR Non-Proliferative Diabetic Retinopathy

PDR Proliferative Diabetic Retinopathy

RESNET Residual Network

ReLu Rectified Linear Unit

RVO Retinal Vascular Occlusion

SVM Support Vector Machine

T1D Type 1 Diabetes

T2D Type 2 Diabetes

TL Transfer Learning

VGG Visual Geometry Group

Chapter 1

Introduction

Diabetes mellitus is a chronic, metabolic disorder characterized by elevated blood glucose levels [1]. This can occur due to insufficient insulin production by the pancreas (Type 1 Diabetes) or a lack of response from the body's cells to the insulin that is produced (Type 2 Diabetes). Failure to receive appropriate treatment for diabetes can lead to a variety of complications, of the chronic complications that may occur, cardiovascular diseases, chronic renal failures, and Diabetic Retinopathy which is the main subject of our work.

Diabetic Retinopathy is an eye condition observed in individuals diagnosed with both type 1 diabetes (T1D) and type 2 diabetes (T2D), occurring in 30% to 40% of all diabetic individuals, and the rise in diabetes prevalence is clearly paralleled in DR [2][3]. It occurs due to the long term effects of diabetes. Having Diabetic Retinopathy can result in irreversible vision-threatening damage to the retina, eventually leading to blindness. It is one of the most common causes of severe vision loss among working-age adults.

The condition is characterized by two major forms, non-Proliferative (NPDR) and Proliferative (PDR) [4]. Although both conditions can result in vision loss, NPDR is the early stage of DR. One thing to emphasize is that patients may be asymptomatic or only experience mild symptoms only until the later stages. DR affects up to 80% of patients who have had diabetes for longer than 20 year.

After fifteen years of suffering from the disease, three out of every four diabetics suffer from Retinopathy [5].

Diabetic Retinopathy (DR) is the leading cause of vision loss in adults aged 20–74 years.In the past decade, DR is ranked as the fifth most common cause of preventable blindness and the fifth most common cause of moderate to severe visual impairment [6]

A meta-analysis conducted in the year 2021 estimated that the global prevalence at the time of Diabetic Retinopathy (DR) was 103 million individuals and is projected to rise to 161 million individuals by the year 2040 [7]. Currently, the global prevalence of diabetes mellitus stands at 537 million, with projections indicating an increase to 592 million by 2035 and 789 million by 2045.. From the data for 2022, diabetes was also responsible for 6.7 million deaths in 2021, costing healthcare expenditures of almost 966 billion USD in the 15 years prior to that.

A global study has found The prevalence of Diabetic Retinopathy is 77.3 % in patients with type 1 diabetes and 25.1% in patients with type 2 diabetes. We should expect approximately 25% to 30% of these patients to develop diabetic macular edema, which poses

a risk to their vision [8] [9]. Between 5% and 8% of patients with Diabetic Retinopathy need laser treatment [10], and as many as 5% of patients will require vitrectomy surgery [11].

After fifteen years of suffering from the disease, three out of every four diabetics suffer from Retinopathy. In the year 2020, Diabetic Retinopathy due was the fifth most prevalent reason for premature vision loss and the fifth most pervasive cause of mild to severe vision loss in people aged 50 [5].

Based on epidemiological projections to 2030, the rates of increase in Diabetic Retinopathy (DR) prevalence in middle to low income regions, such as the Western Pacific,the Middle east, and Africa, range from 20.6% to 47.2 %, which by far outnumbers the projected rates in the high income regions, such as North America And Europe [3] [7]. These are also the areas that will see the largest increase in disease burden in absolute terms. Not only is the global prevalence of Diabetes and Diabetic Retinopathy expected to increase, but so is the global prevalence of vision-threatening DR.

Part of the reason why Diabetic Retinopathy is more prevalent in low- to middle-income countries is the limitations of health care access and resources, commonly the lack of awareness among the population, and setting a priority to other matters in life rather than considering spending the money for a fundus examination by a specialist. In developing countries, the private healthcare sector is expensive [12], and it can take months to get your turn in the queue to visit a public health care specialist. The solution we propose in this project aims to decrease the costs and duration of exams, thereby encouraging a greater number of people to adopt preventive measures.

As these countries also have a higher prevalence of DR, and they have an insufficient number of primary physicians and ophthalmic specialists to effectively address the disease. and it is evident in nations like China; in the given region, the ratio is 1.5 ophthalmologists per 50,000 inhabitants. Likewise, the United States has approximately 18,000 ophthalmologists for its population of over 330 million [13]. Another example, there is one ophthalmologist per 107,000 inhabitants in the Indian subcontinent, disregarding the large urban centers, the proportion drops to one ophthalmologist for every 608.000 inhabitants. Another research also found that globally 22.7% of diabetes patients had Diabetic Retinopathy, in Africa it was 36% [14].

Diabetes, along with other "noncommunicable diseases" were traditionally considered predominantly afflictions of high-income, developed countries, but this is no longer true, and in absolute terms as we have seen there are now more individuals with diabetes living in the developing world than established developed countries [15]. All patients should receive affordable treatment. From the numbers, we can infer that numerous patients in many underdeveloped nations do not have access to quality health care or expensive screening. Individuals with diabetes, regardless of type, require yearly retinal screenings on a regular basis in order to detect Diabetic Retinopathy at the proper time in its early stages and receive the appropriate treatment and follow-up.

So in summary, despite the availability of effective treatments, many individuals with Diabetic Retinopathy remain undiagnosed and untreated, particularly in low- and middle-income countries, due to limited access to eye care services and screening programs. This is one of the problems. As the number of diabetics continues to increase exponentially in

the coming years, the number of qualified ophthalmologists needs to increase in order to meet the demand for screening the growing number of diabetic patients. This makes it pertinent to develop ways to automate the detection process for DR.

Challenges

A widely known challenge for DR is that it lacks an early warning sign [16]. And the early identification. Left unattended, DR worsens vision and causes partial or complete blindness. Uncontrolled diabetes can result in hyperglycemia, which is a condition that can have severe impacts on several physiological systems, especially the blood vessels and neurons. The destruction of retinal blood vessels increases the likelihood of fluid loss and hemorrhaging [17]. Medical professionals can help slow down the progression of Diabetic Retinopathy in patients by implementing interventions such as photo-coagulation, intravitreal injections, and controlling blood glucose levels and blood pressure.

The treatment of Diabetic Retinopathy in a critical time window can minimize the risk of visual loss by 90% [18]. Any person with diabetes mellitus should get a retinal fundus examination at the time of diagnosis, and annual follow-ups after. Patients with mild to moderate Non-Proliferative Diabetic Retinopathy (NPDR) will end up needing frequent visits and follow-ups after the first diagnosis of the condition, such as every 6 to 9 months. And those with severe NPDR, or Proliferative Diabetic Retinopathy (PDR) may require examinations every 2 to 4 months, or as recommended by the ophthalmologist.

The conventional method for diagnosis is a manual screening and fundus examination, where an ophthalmologist examines the back of the eye, including the retina, optic disc, and blood vessels. This process involves dilating the pupils using eye drops and then using a specialized device to analyse the retina [19]. The ophthalmologist then looks for signs of Diabetic Retinopathy (DR), such as micro-aneurysms, hemorrhages, and exudates, and changes of the blood vessels like abnormal dilation or constrictions, all indicating vascular abnormalities caused by Diabetic Retinopathy. The conventional solutions for identifying DR include both live manual screening by a well-trained clinician and the assessment of digital color fundus photographs. While effective, both methods are time-consuming and heavily dependent on the expertise of skilled practitioners. The manual classification of Diabetic Retinopathy features, which involves evaluating multiple features and their locations, is particularly labor-intensive, and susceptible to human failure. These limitations highlight the need for more efficient diagnostic approaches. However, once trained, computers can swiftly classify these features, offering the potential to assist clinicians in real-time classification and significantly expediting the process.

In the (Figure 1.1) we'll see the multiple type os fundus cameras commonly used to capture fundus images that we'll be using to train our dataset, and in the (Figure 1.2) and (Figure 1.3) showcase the manual screening process, we can see it is an exhausting process and requires concentration and careful attention to very details of small visibility scope, not only that but the other issue is that during the live screening the ophthalmologist cannot see the entire fundus at once.

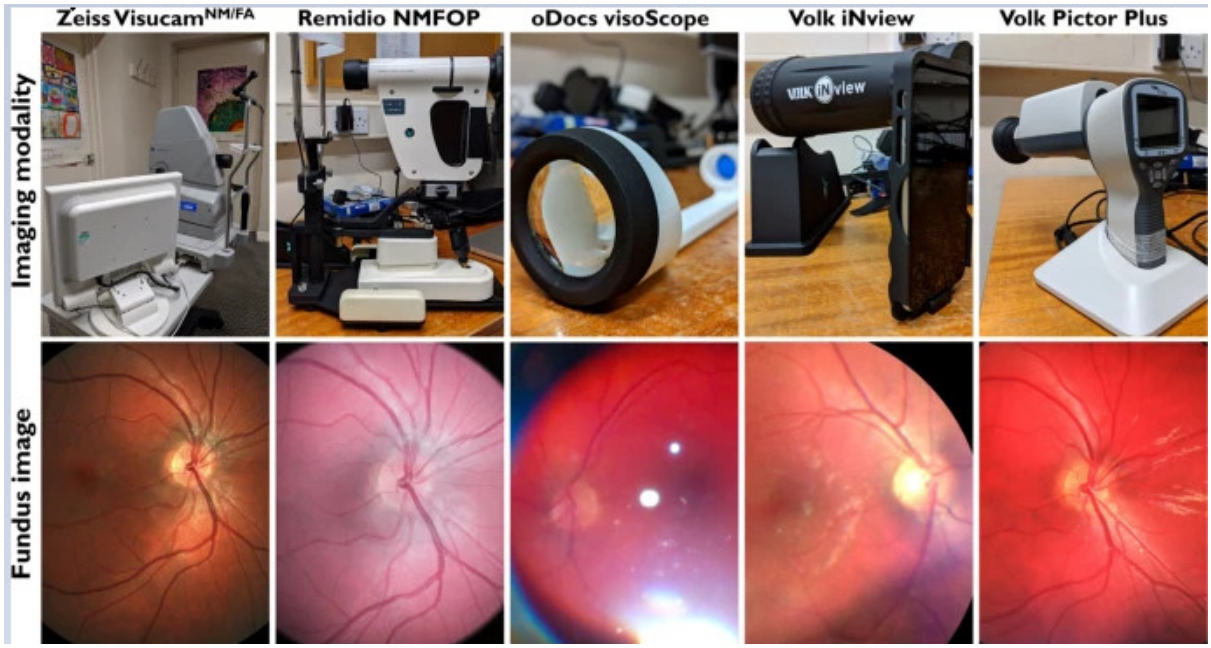


Figure 1.1: Different fundus cameras used in the screening of the retina [20]



Figure 1.2: First image of a manual screening of the retina using a fundus camera

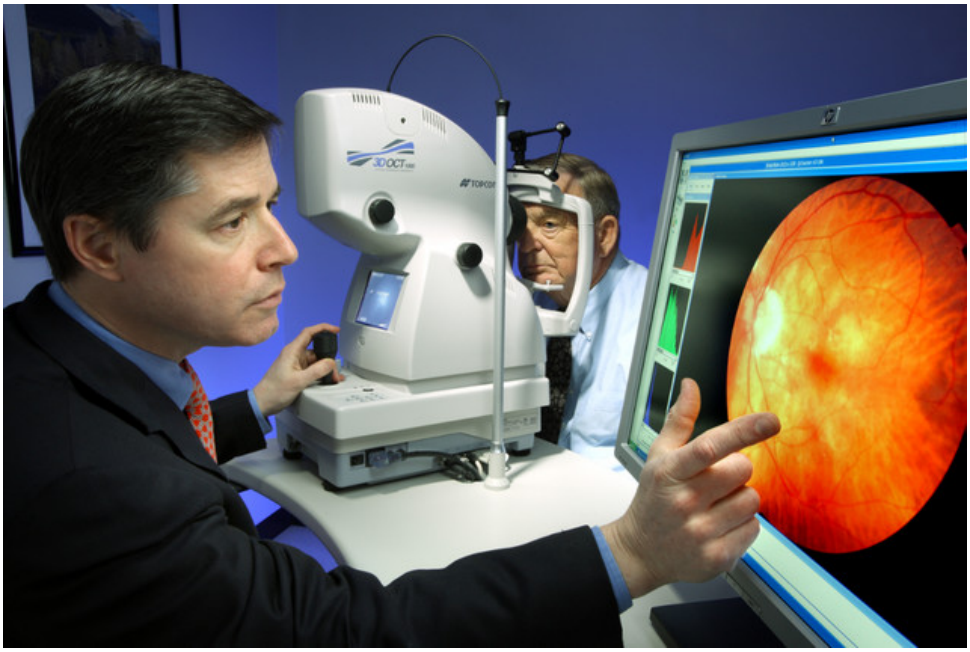


Figure 1.3: Second image of a manual screening of the retina using a fundus camera[21]

Another challenge is that since the diagnosis of Diabetic Retinopathy is conducted by visually inspecting fundus images of the retina, it's a time-consuming process and task, as well as susceptible and prone to human failures, which can be aggravated due to the great workload imposed on the specialists responsible for this analysis. And neither traditional screening methods nor telemedicine can meet the needs of a growing number of patients in the upcoming years, as presented earlier in the introduction.

According to a study conducted by Google, the researchers assembled a panel of their top ophthalmologists to evaluate a set of patient cases. The rows represented individual patients, while the columns corresponded to the different ophthalmologists assessing those cases, The (Figure 1.4) illustrates this.

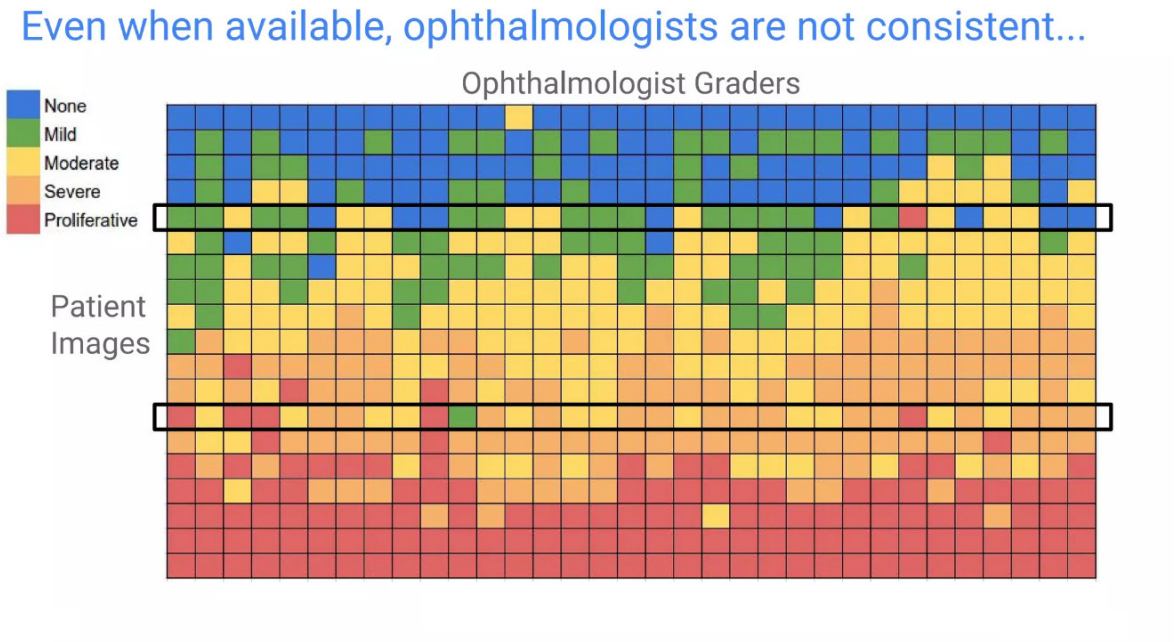


Figure 1.4: Even highly qualified eye specialists find it challenging to accurately diagnose eye diseases. Eye specialists are skilled at identifying both extremely healthy and extremely unhealthy tissue, but they frequently struggle to diagnose tissue that is in between [22]

Even among this best group of their specialists, the researchers found significant variability in the diagnosis of Diabetic Retinopathy. The consistency of diagnoses between ophthalmologists averaged only around 60% across the patient sample. Furthermore, the data revealed that approximately 1 out of every 6 patients received a false diagnosis.

These findings underscore the inherent difficulty in accurately diagnosing Diabetic Retinopathy, particularly in the early stages of the disease. The study highlights how many patients may be missing out on timely treatment due to the challenges clinicians face in making consistent and reliable assessments. This emphasizes the need for more robust and standardized diagnostic tools to improve the early detection and management of this condition.

Certain Diabetic Retinopathy lesions, notably micro-aneurysms and intraretinal microvascular abnormalities (IRMA) [23], can be easily overlooked during the manual screening and evaluation, leading to false negatives and delayed diagnoses. Prolonged manual screening and evaluation of retinal images can lead to fatigue and burnout among ophthalmologists, potentially increasing the risk of diagnostic errors or missed lesions. So it is often a challenge to identify the presence of the condition at an early stage, when the treatment is of the highest benefit and most cost-effective.

Another common issue is the significant delay between the initial screening and the receipt of results from the clinic. In many healthcare settings, patients undergoing DR screenings often face waiting periods that span several weeks before they receive their results. The labor-intensive nature of manual screening, where each fundus image requires examination by highly trained specialists, primarily causes this delay. Compounding this

issue is the volume of screenings required, given the prevalence of diabetes and the necessity for regular monitoring among diabetes affected individuals. Such delays also hinder timely intervention.

The requirement of highly trained and licensed medical professionals incurs substantial expenses [24], Given the circumstances, the solution for the automated techniques to detect Diabetic Retinopathy also serves as a more economically viable substitute for the manual diagnosis of Diabetic Retinopathy conducted by the clinicians who have undergone specialized training. It is imperative that the innovation of new strategies, such as the use of automated diagnostic systems to bridge this gap, create accessible resources to cover this disparity.

Computer-aided diagnosis systems (CAD), featuring an automated diagnostic system that includes our proposed automated solution, have gained significant attention in recent decades. [25] has become in the past decades, one of the major research subjects in medical imaging and diagnostic radiology. With CAD, the ophthalmic healthcare staff can use the computer output as a "second opinion" and make the final decisions.

Previous automated Diabetic Retinopathy detection systems relied on standard Machine Learning algorithms for prediction and manually constructed feature extraction [26]. Which means ophthalmologists or trained clinicians will take an image and manually annotate the abnormal regions pixel by pixel, often on two complete iterations by two specialists to avoid missing out on pixels. Although feature extraction in color fundus images is more challenging than in traditional images for object detection tasks. Furthermore , the sensitivity of these manually crafted features to noise, artifacts, focus angle, and fundus image quality is quite high. Therefore, in order to effectively analyze the subtle features related to the DR detection task, it is imperative to develop an effective automated feature extraction without the intervention of humans due to the limitations of traditional handcrafted features. The technique employed in our automated solution is Deep Learning, which tackles this issue, and we'll be expanding on this in the next chapters.

In our diagnostic system, we will be using state-of-the-art techniques and approaches to tackle this problem and try to achieve state-of-the art specificity and sensitivity in the diagnosis and classification of the severity of the disease. Addressing these bottleneck through innovative solutions, such as the integration of Artificial Intelligence (AI) in the retinal screening process, could aid Diabetic Retinopathy management by providing rapid and confident diagnosis in one hour or two compared to weeks of waiting, enabling prompt treatment initiation and potentially improving patient outcomes.

Stages Of Diabetic Retinopathy

The Treatment of Diabetic Retinopathy Study (ETDRS) classification has been considered the "gold standard" for many years because it was developed and validated on natural history data that demonstrated its ability to prognosticate the risk of progression to Proliferative disease and vision loss [27].
First we'll show the common lesions in the eye of someone diagnosed with Diabetic

Retinopathy and explain the stages in brief, (Figure 1.5) and Figure give an idea.

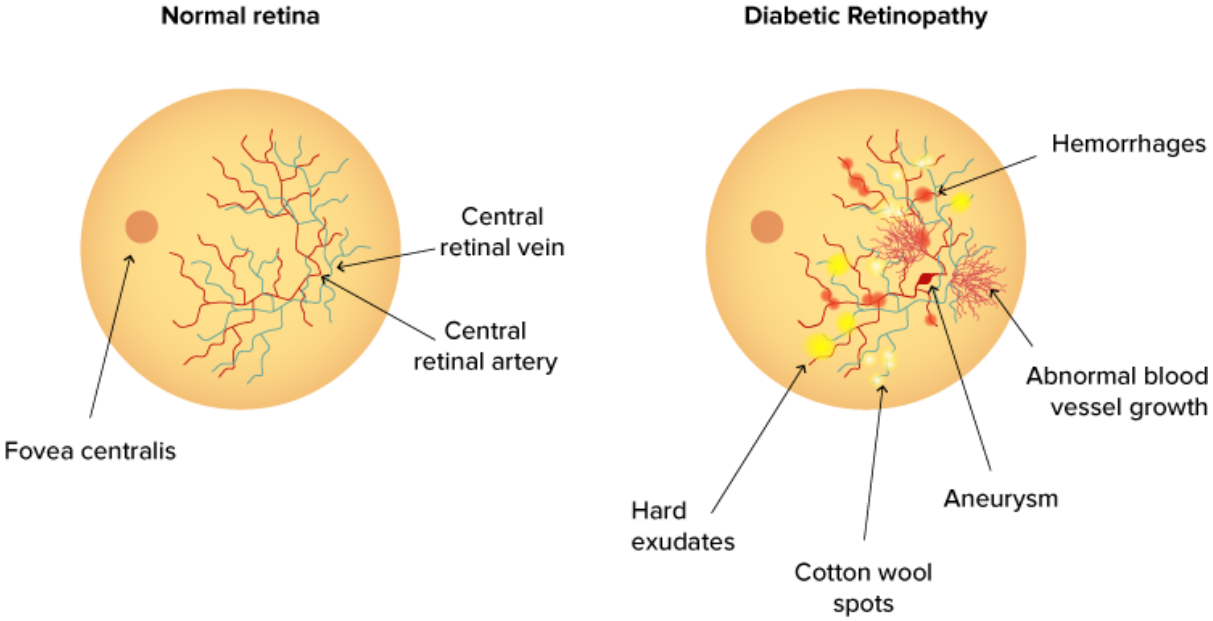


Figure 1.5: Simplified illustrative representation of the difference between a healthy eye and an eye of a patient with Diabetic Retinopathy [28]

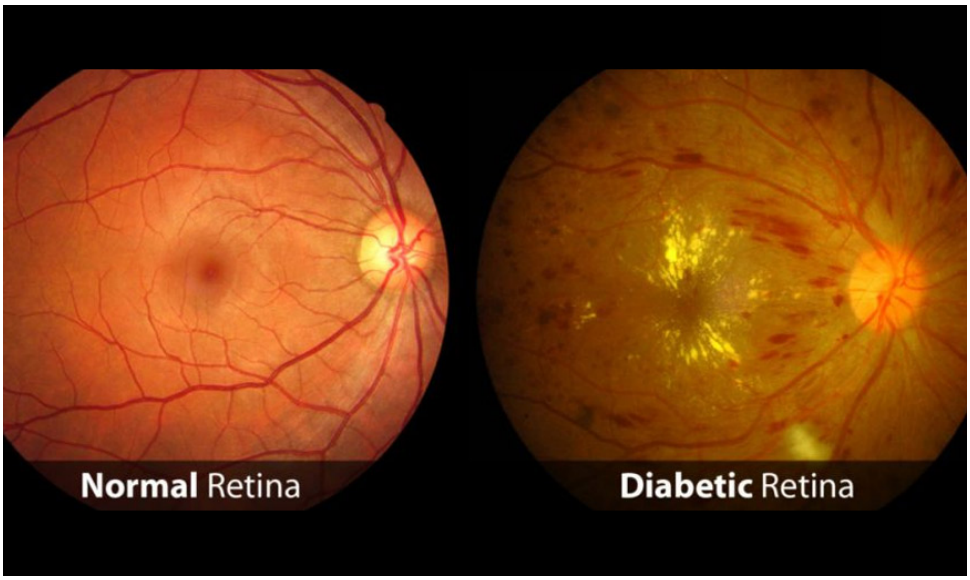


Figure 1.6: Fundus view difference between a healthy and an eye with the condition

According to the ETDRS standard of classification, Diabetic Retinopathy is classified into four stages based on the severity of the retinal damage and the presence of specific ocular findings. These stages provide a framework for understanding the progression of the disease and guiding clinical management. This is just a summary for now, we'll expand even more on the lesions later on.

Stage 1: Mild Diabetic Retinopathy

indicated by the presence by of microaneurysms, small retinal hemorrhages, and retinal swelling (edema). Often appearing as small red dots [29].

Stage 2: Moderate Diabetic Retinopathy

Progression from mild NPDR with an increased number and size of hemorrhages, as well as the presence of intraretinal microvascular abnormalities (IRMAs) [30].

Stage 3: Severe Diabetic Retinopathy

Marked by extensive retinal hemorrhages, venous beading, and more pronounced IRMAs. The risk of progression to Proliferative Diabetic Retinopathy (PDR) is heightened at this stage.

Stage 4: Proliferative Diabetic Retinopathy (PDR)

The most advanced stage, characterized by the growth of abnormal blood vessels (neovascularization) on the retina or optic disc [31]. These fragile vessels can lead to vitreous hemorrhage or tractional retinal detachment, at a high chance of risk of causing severe visual impairment or blindness if left untreated. The (Figure 1.7) provides a visual representation of common differences between the lesions of each stage of Diabetic Retinopathy.

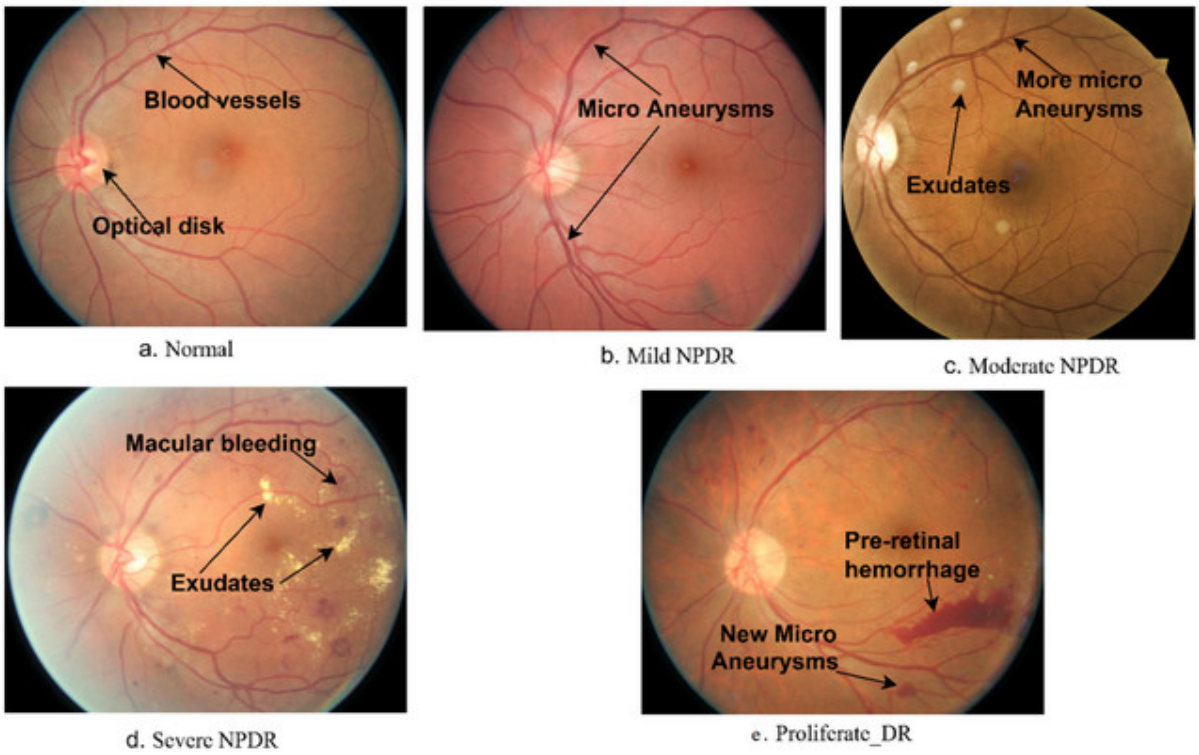


Figure 1.7: Retinal fundus images of different stages of Diabetic Retinopathy [32]

Objectives

From the current challenges and global statistics of the disease, this can be classified as a "pandemic" by the experts. It is pertinent to develop a system that automates the diagnostic and referral process.

Artificial intelligence (AI) is a major field of research in computer science. This technology holds a lot of potential for improvements in healthcare in terms of affordability, accessibility, and quality. In simple terms, artificial intelligence (AI) refers to the ability of a computer or robot to perform a task with minimal human intervention and to guide its functioning. AI is a software or machine-generated simulation of human intelligence. It is the ability of a programmed system to show cognitive capacity and abilities. Just like learning in humans, AI systems need to be trained and put to learn by being exposed to data, which allows them to first "learn" simple targets in regard to a dedicated finding or disease.

The integration of AI into the biomedical field started in the 1970s [33], and since then, several AI systems have been developed. The advancements in computational power and capacity of hardware, the persistent and organized collection of data, and the formulation and advancement of Deep Learning theory have led to significant advancements in the field of intelligent medicine through the implementation of AI systems. The development of AI-assisted medical screening and diagnosis using photos is currently advancing and progressing. The current focus of applying this technology in ophthalmology is primarily on the diseases that commonly have a high incidence, including Diabetic Retinopathy (DR), glaucoma, age-related macular degeneration (ARMD), Retinopathy of prematurity (ROP), age-related or congenital cataract, and retinal vein occlusion (RVO) [34][35]. The automated diagnostic system, especially when trained adequately on large data sets and variations, can offer consistent and objective screening results, reducing the variability associated with human interpretation of retinal fundus images.

The lack of enough annotated training data has been identified as one of the key challenges of applying DL in healthcare, especially in medical images [36][37]. This could be due to a variety of factors, of them due to privacy and security issues with the medical data of patients. Therefore, it is necessary to develop an automated DR detection method that addresses the lack of much annotated data, and Transfer Learning addresses this one obstacle. Transfer Learning approach, Which makes use and leverages Machine Learning models, specific in our case Deep Learning ones that have learned to read and detect the patterns, objects from millions of diverse images and fine-tuned into achieving the highest accuracy possible based on their architectures and programming. The primary objectives we have is to develop Machine Learning models that seamlessly integrate with the systems and hardware utilized by ophthalmologists to diagnose and analyze patients' eyes. Below in the sections, we will discuss the purposes of these models and the solutions we aim they achieve.

1.0.1 Automated Diabetic Retinopathy Detection And Stage Grading System

The CNN architecture models have been successfully applied to automatically classify color fundus retinal photographs into binary (Negative and Positive For DR) and multi-

class categories for the purpose of detecting Diabetic Retinopathy (DR), in our work, the models use state-of-the-art techniques and approaches, and aims to achieve state-of-the-art performance in the detection severity grading of the Diabetic Retinopathy.

The models have exhibited exceptional efficacy in the automated detection and severity grading of Diabetic Retinopathy (DR) in nonclinical environments, even in cases when the dataset or computational resources are limited. especially at the early stages, and minimizes the false negatives.

These models use typical TL approaches to extract numerous important local (textural) properties from retinal fundus images. The goal is to diagnose Diabetic Retinopathy (DR) and predict the severity of the disease, especially as it's the early stages, and determine which patients need to be referred to ophthalmologists. Global image features, together with transfer learning of the CNN pre-trained models, have proved crucial in identifying Diabetic Retinopathy (DR) and classifying the stage of the severity, especially at the early stages. the contour and structural features employed in the identification of retinal abnormalities, including exudates, microaneurysms, and hemorrhages, and blood vessels abnormalities mostly.

When the model is deployed and inferred on the aim of the diagnosis of Diabetic Retinopathy using a fundus eye of a patient, not only the system would inform the prediction it actually made but also the probability and certainty of its prediction since the prediction of models are all based on probabilistic guess from its previous experiences, Almost every Machine Learning methods are fit using Bayesian or approximately Bayesian methods.

The identification of one or more of these lesions can serve as an indicator for the existence of Diabetic Retinopathy (DR), and the patterns and quantity localization they display indicates the severity and progression of the disease. Meaning these global characteristics, commonly referred to as lesion features, are considered to be favorable markers of retinal image defects. But still, the performance result of A.I automated systems depend heavily on the size of training data as much as its quality and the balance of the distribution of its classes.

The employment of highly trained specialists is not a prerequisite for this system. Clinical personnel can be instructed to capture fundus images and execute diagnostic assessments utilizing the user-friendly interfaces of the platform. Also, The screening modality can be more readily accessible within clinical settings, public healthcare facilities, and pharmacies, thereby facilitating timely referral. The proposed approach can enable screening to be completed within an average timeframe of approximately one hour, in contrast to the extended waiting periods of hours or days required for traditional ophthalmologist examination. The system aims to offer increased accessibility for patients, enhanced efficiency in the screening and diagnostic workflow, and improved affordability of treatment for Diabetic Retinopathy.

1.0.2 Multi Retinal Blood Vessel Segmentation Systems

Also in our work we will be developing two distinct models for retinal blood vessel segmentation, each one specified for a specific angle of the fundus photography, one for the fundus images where Macula is centered and the other where the optic nerve is centered

in the image. the two imaging angels are shown in the (Figure 1.8).

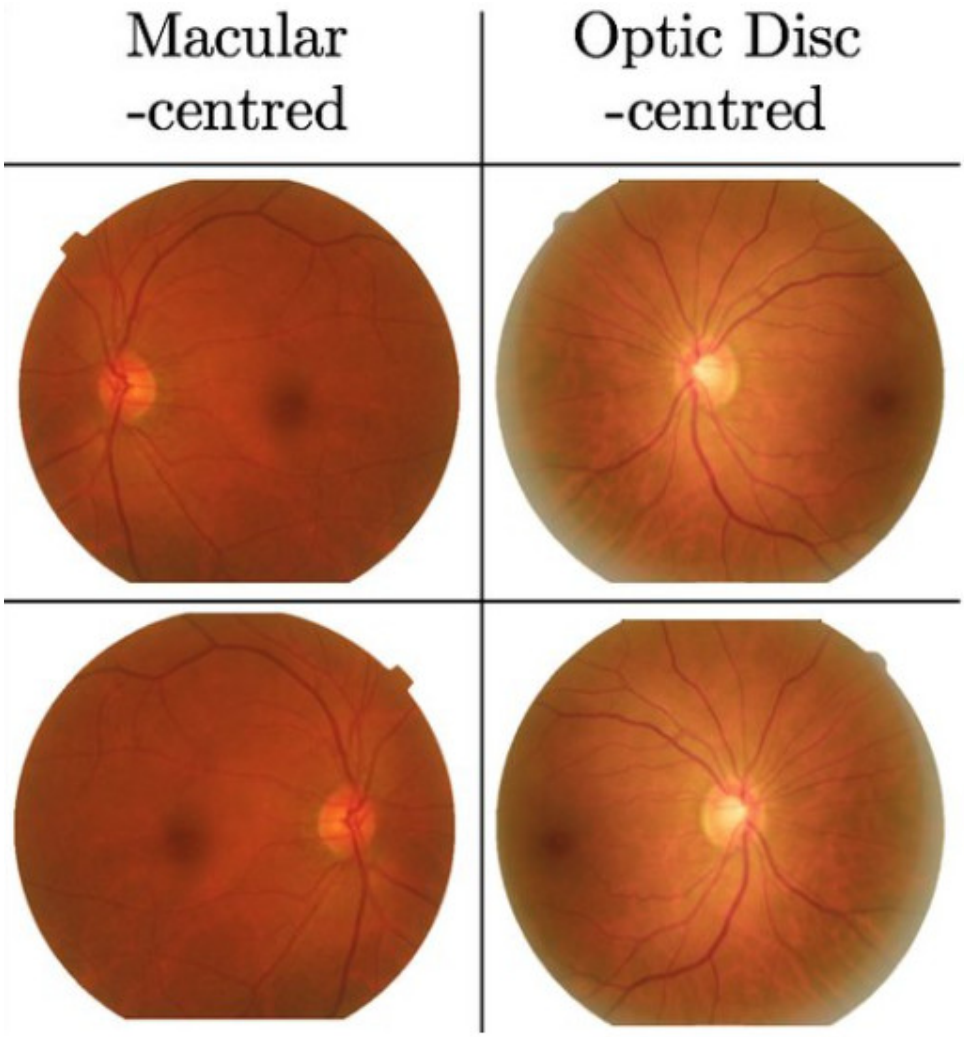


Figure 1.8: Different view angles of fundus imaging [38]

Retinal vessel segmentation is the process of extracting and identifying the blood vessels in the retina of the eye from fundus images, It plays a crucial role in the diagnosis and treatment of various retinal diseases, such as DR (Diabetic Retinopathy), ARMD (age-related macular degeneration), and glaucoma.

Deep Learning (DL) architectures, namely Convolutional Neural Network (CNN) architectures such as U-NET, have been employed in the segmentation of retinal blood vessels. The development of this architecture has facilitated the isolation of pertinent anatomical elements for the purpose of diagnostic analysis. This segmentation methodology provides a supplementary approach to Diabetic Retinopathy detection strategies by displaying and extracting the vascular abnormalities associated with Diabetic Retinopathy.

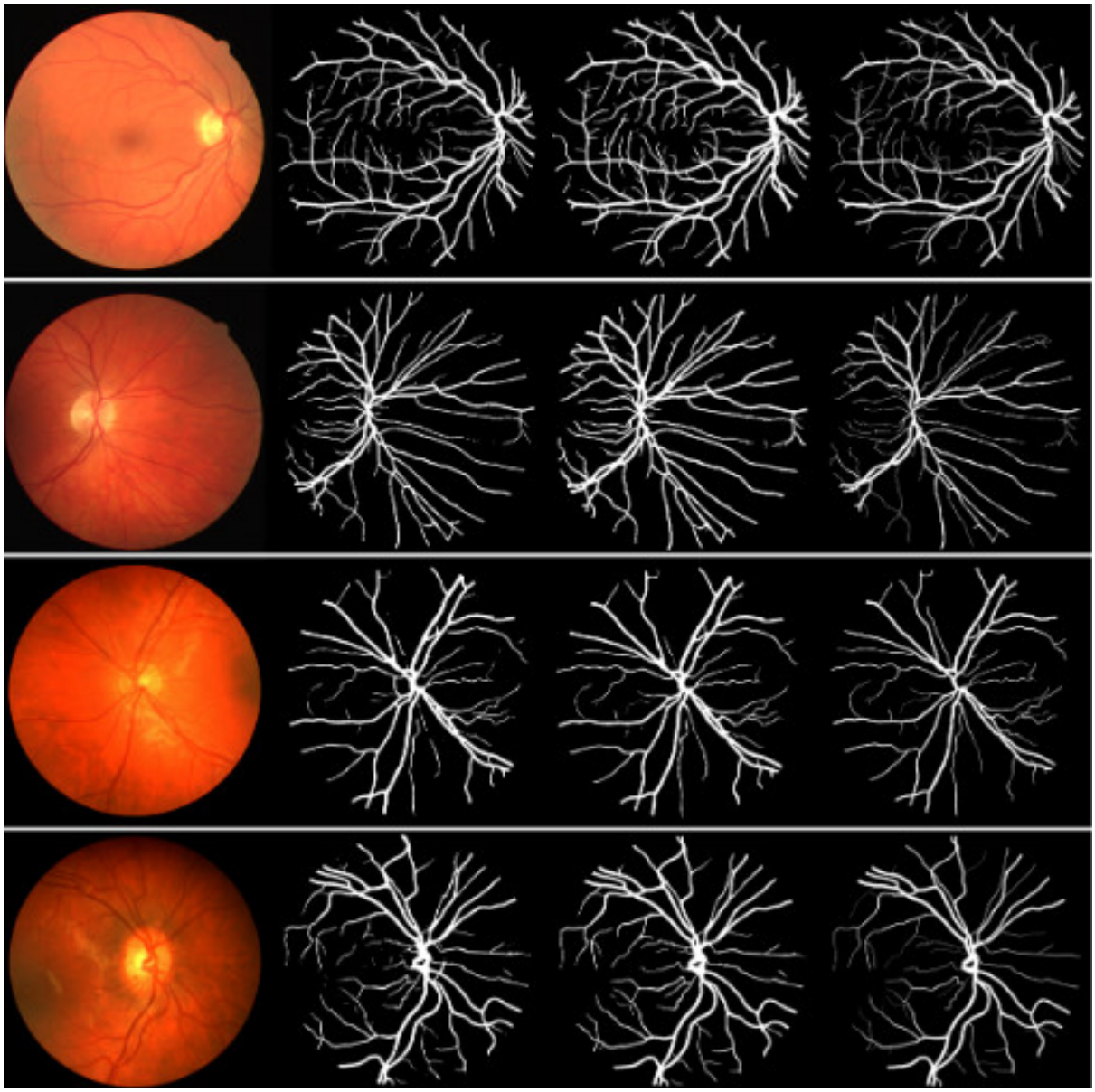


Figure 1.9: Semantic Segmentation of Retinal Blood Vessels

The extraction and segmentation of retinal blood vessels allows ophthalmologists to monitor the progression of diseases over time by comparing segmented images from different visits [39]. Also provides quantitative metrics such as vessel density, tortuosity, and size which can assist in disease characterization and progression assessment.

It can also assist surgeons in planning and executing retinal surgeries by providing detailed preoperative vascular maps [40], it can also enable the discovery of possible new biomarkers associated with retinal diseases by analyzing vessel characteristics across a large dataset of patients, Which can also support epidemiological research by providing insights into the prevalence and distribution of retinal vascular abnormalities in different populations.

Doctors can detect various conditions by examining the dilation and constriction of retinal veins. Changes in the diameter of these vessels can indicate underlying health issues and help in diagnosing and monitoring several diseases:

- Hypertension and Cardiovascular Diseases

Retinal vein dilation can be an early sign of systemic hypertension. Hypertension causes increased pressure in the blood vessels, leading to their dilation [41]. Regular monitoring of these changes can help in assessing the effectiveness of antihypertensive treatments and in predicting cardiovascular events. Other than Diabetic Retinopathy can also help with the diseases

- Retinal Vein Occlusion (RVO):

Dilation of retinal veins is a hallmark of RVO [42], a condition where a blockage occurs in the retinal veins, leading to impaired blood flow. The subsequent increase in venous pressure causes the veins to dilate. Recognizing this can help in the early diagnosis and management of RVO, which is crucial to prevent further vision loss.

- Glaucoma:

Changes in retinal vessel diameter, particularly the narrowing of veins, can indicate glaucoma [43]. Glaucoma causes damage to the optic nerve, often associated with increased intraocular pressure. Monitoring the retinal vessels can assist in early detection and ongoing management of glaucoma.

- Inflammatory Diseases:

Conditions such as uveitis can lead to both dilation and constriction of retinal vessels due to inflammation [44]. Tracking these changes can help in diagnosing and monitoring the progression of inflammatory diseases affecting the retina.

Ultimately, concerning the models we aim to build and develop, it is up to the ophthalmologist to decide which model to use and infer for the specific use case at hand. Each of these models will have distinct inputs and outputs. Therefore, the main objective is to provide automated solutions for the diagnosis and monitoring of ophthalmologic diseases, eventually the system will scale and would have more various models for distinct diseases and use cases expected.

Literature Review

The traditional method of detecting Diabetic Retinopathy involves numerous challenges. The shortage of eye specialists and the high expense of diagnostic tests pose substantial barriers for many individuals in economically disadvantaged regions. Automated processing methods have been used to ensure timely identification and prevention of blindness and irreversible eye damage. These methods improve accessibility and accuracy of diagnostic procedures within a suitable time frame. Machine Learning (ML) systems, utilizing retinal fundus images, have effectively accomplished precise classification of Diabetic Retinopathy (DR). There has been a significant focus on creating affordable and efficient automated algorithms for the examination of retinal images and the prediction of DR. A range of strategies have been employed to address the task of detecting Diabetic Retinopathy (DR).

Numerous innovative approaches have been explored in recent research for the classification and identification of Diabetic Retinopathy (DR). In their study conducted in 2019, [45] Kothare and Malpe utilized a combination of Naive Bayes and Support Vector Machine models to predict the presence of the Diabetic Retinopathy . Their findings revealed that, in comparison to Naive Bayes, SVM outperformed in aspects such as the accuracy, inference time and computational efficiency. In this other work, the authors [46] developed a deep neural network algorithm to identify the presence of hard exudates. Their only limitations was the size of data set 48. In a separate study [47], Fayyaz utilized various Support Vector Machine (SVM) kernels with a meta-heuristic optimization algorithm named Ant Colony System for the selection of features, The study found that with the use of a cubic SVM, they were able to achieve accuracy rates of 92.6% and 93% percent for 250 and 750 features respectively. Data augmentation is a commonly used method to address the issue of unbalanced data in image classification tasks. It's often employed in the context of transfer learning to enhance the categorization of Diabetic Retinopathy (DR) images by generating varied versions of the existing images. This technique helps to increase the diversity and volume of the training data, leading to improved model performance. Eftekheri et al [48] conducted a study on the segmentation of micro-aneurysms from retinal fundus images. They utilized a two-stage Convolutional Neural Network (CNN) to accomplish this task. Furthermore, they explored the correlation between the severity of Diabetic Retinopathy (DR) and the presence of these lesions. This approach allowed them to gain deeper insights into the progression of DR and its relationship with specific retinal abnormalities. Silberman et al [49] suggested an alternate method that involves patch-level detection. They utilized SIFT characteristics and SVM to differentiate exudate patches. Gurcan et al [50] introduced an automated system for classifying Diabetic Retinopathy (DR) using the Inception V3 pre-trained model to extract Diabetic Retinopathy features. The researchers assessed a variety of classifiers, such as Decision Trees, XGBoost, Random Forest, Support Vector Machine (SVM), Logistic Regression. In a distinct study [51], a comparison was made between InceptionV3, ResNet101, and DenseNet121. the performance was best found using the DenseNet121 Model . An CNN model, incorporating Histogram Equalization (HIST) and Contrast Limited Adaptive Histogram Equalization (CLAHE) approaches, achieved a testing accuracy of 97% and an F1-Score of 94% in the classification of Diabetic Retinopathy (DR) [52]. The "AD2Net" system, developed by Z. Qian [53], enhanced the effectiveness of processing and the speed of diagnosis by integrating ResNet and DenseNet with an attention mechanism. This integration resulted in an accuracy rate of 83.2% of 5 class multi classification. A different research [54] conducted a comparison between SVM and KNN classifiers in the detection of micro-aneurysms and lesions, and found that SVM performed better than KNN.In their research, Khalifa et al [55] explored the use of deep transfer learning models for the detection of Diabetic Retinopathy (DR) utilizing the APTOS 2019 dataset. They employed a range of architectures such as AlexNet, ResNet18, GoogleNet AND VGG16, AlexNet had the best peformance despite the low number of layers compared to other networks.In this work, [56] the authors employed an ensemble learning approach by combining several CNN models, including ResNet50, Inception V3, Xception, DenseNet121, and DenseNet169. Ensemble learning involves integrating multiple Machine Learning models to improve predictive performance. This approach mitigates the weaknesses of individual models, leading to more reliable predictions. Techniques such as voting, averaging, or stacking are often employed to aggregate the outputs of the base models. Another contribution, from this article [57] a fuzzy C-means clustering algorithm

and a genetic algorithm were utilized to predict the presence of Diabetic Retinopathy. The developed algorithm was applied to a total of 224 retinal images from patients. The proposed method demonstrated a sensitivity of 0.78%. Qiao et al [58] suggested employing semantic segmentation to detect the presence microaneurysms in fundus images, assisting in the classification of Diabetic Retinopathy especially at the early stages. The authors suggested a U-Net model that combines global and local residual information to overcome the limits of global image pre-processing in retinal vascular segmentation. The model achieved good accuracies on the DRIVE and STARE datasets, as reported in the study by Lian et al. (2015) [59]. A different research [60] introduced a fully convolutional neural network that can directly distinguish arterioles and venules from retinal pictures.

From reading the related articles, a common problem is the limited number of images in the existing datasets, which are often sourced from a particular region or demographic. This results in a lack of diversity and under-representation of other ethnic groups, this limitation can affect the performance of CNN networks that are trained from scratch, as they cannot match the effectiveness of pre-trained CNN models that have been trained on millions of images and consistently achieve high performance scores. Furthermore, manual feature extraction proves to be inadequate due to the variability of the location in lesion locations since they can float around and move in the fundus, also take different shapes and color contrasts. Additionally, not using data augmentation techniques for creating new variations of images from existing ones increases the risk of over-fitting and potentially affects the model performance with unseen data. Training CNN from scratch requires a very large training dataset to make up a result better than Transfer Learning.

Thesis Structure

Chapter 1 was the introduction of our Thesis, detailing the context and motivation behind our work. It discussed the problems we aim to address, the approaches and methodologies commonly used by researchers in the literature, and the proposed solution. The central parts of our work are organized into the following chapters:

- **Chapter 2:** This chapter covers the concepts and details of the techniques used in our work, primarily machine learning, deep learning, and specifically the deep learning architectures we utilized.

- **Chapter 3:** This chapter details the implementation and design of the **Diabetic Retinopathy** detection system. We will discuss the steps and methodology taken to develop the model, including data collection, training, model evaluation, and inference. We conclude with the use case of our model post-construction.

- **Chapter 4:** This chapter focuses on the implementation of the **Retinal Blood Vessel Segmentation** system for two fundus angles, presenting a model for each and considering structural changes.

Chapter 2

Fundamental Concepts

In this chapter, the chapter looks into the historical development of AI and explores the key techniques that are relevant to the field work of this thesis. It also provides a comprehensive overview of the specific techniques used to train and achieve our model, like how A.I, Machine Learning, Deep Learning contribute together to build computer aided diagnostic systems able to process medical images for grading and segmentation.

Artificial Intelligence

Artificial Intelligence (AI) refers to the development of computer systems capable of performing tasks that traditionally require human intelligence, such as visual perception, speech recognition, decision-making, and language translation. The roots of AI can be traced back to the pioneering work of mathematicians like Alan Turing in the 1930s and 1940s [61]. Turing proposed the concept of a "universal machine" that could perform any computation if appropriately programmed.

Alan Turing's pioneering work in computation and artificial intelligence is marked by his invention of the Turing Machine in 1936, a conceptual computing device comprising an infinite tape and a read/write head for symbol manipulation. Demonstrating the universal capability to compute anything algorithmically, Turing extended this concept to propose a Universal Machine, capable of simulating any Turing Machine by reading instructions from a tape. This foundational idea laid the groundwork for modern computers, enabling them to execute various programs. Turing's vision extended further in his 1950 paper, "Computing Machinery and Intelligence," where he introduced the Turing Test. This test remains a benchmark for evaluating whether a machine can engage in human-like conversation.

The term "Artificial Intelligence" was coined in 1956 by John McCarthy at the Dartmouth Conference, a pivotal event that marked the formal establishment of AI as a distinct field of study. During its inception, early AI research centered on knowledge-based systems that aimed to encode human expertise and reasoning into software through explicit rules and logical deductions. These systems aimed to emulate human decision-making processes in domains such as theorem proving, game playing, and language translation using symbolic manipulation. The Dartmouth Conference laid a foundational framework for subsequent advancements in AI, paving the way for the evolution of machine learning, neural networks, and other modern AI techniques that have revolutionized technology and society.

In the 1980s, AI research experienced a resurgence with the development of expert systems [62] - software that could make decisions based on rules extracted from human experts. However, this symbolic approach faced challenges in dealing with uncertainty and integrating common sense reasoning.

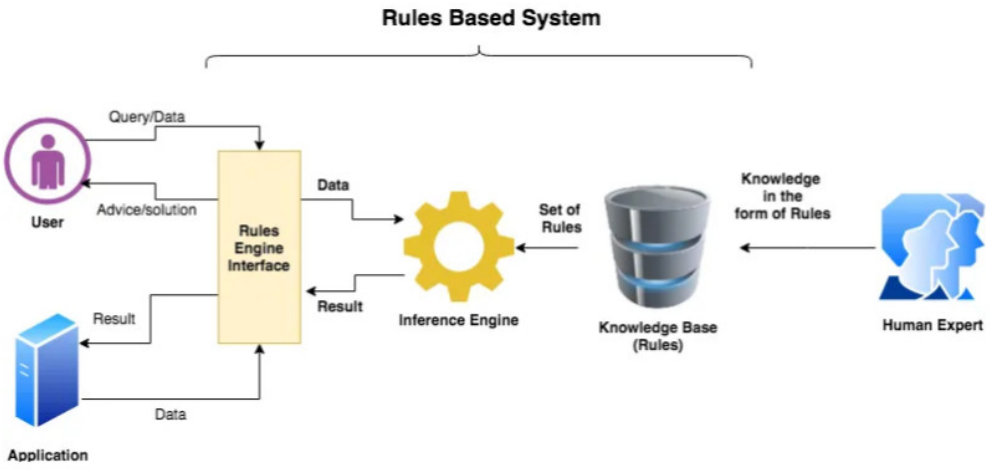


Figure 2.1: A Figure of expert base systems

A major breakthrough occurred in the late 1980s with the introduction of artificial neural networks inspired by biological neural networks in the human brain. This machine learning technique allowed systems to learn from data rather than relying solely on manually encoded rules. Neural networks facilitated breakthroughs in areas like image recognition and speech processing.

The 1990s and 2000s witnessed rapid progress in machine learning, with new techniques like support vector machines and ensemble methods. However, the most transformative development was the revival of neural networks as ”deep learning” models on the back of increasing computational power and availability of big data.

Deep learning has driven remarkable advancements across various AI applications from the late 2000s until today. Convolutional neural networks enabled major leaps in computer vision, while sequence models like LSTMs [63] and Transformers revolutionized natural language processing tasks like translation, summarizing, and question-answering.

Reinforcement learning algorithms that learn from interacting with and receiving feedback from an environment have fueled developments in areas like game-playing systems and robotics control. Generative AI models can now create highly realistic synthetic data like images, audio, text, and even computer code. As AI rapidly advances continued research coupled with accountable development and deployment will be crucial for maximizing its immense potential.

2.1 Machine Learning (ML)

Machine learning is a subdomain of artificial intelligence (AI) that focuses on designing systems that learn or improve performance based on the data they consume . The field of machine learning (ML) has rich theoretical foundations, and it involves using computers

to optimize an information processing model based on certain performance criteria from observations, whether they are example data or past experiences.

Machine learning refers to the development, analysis, and implementation of methods that allow a machine to evolve through a learning process, and thus fulfill tasks that are difficult or impossible to fulfill through more traditional algorithmic means. What is the difference between A.I and Machine learning? A.I is the development of smart machines, while Machine learning is the development of machines that can learn from data, The (Figure 2.2) simplifies the difference.

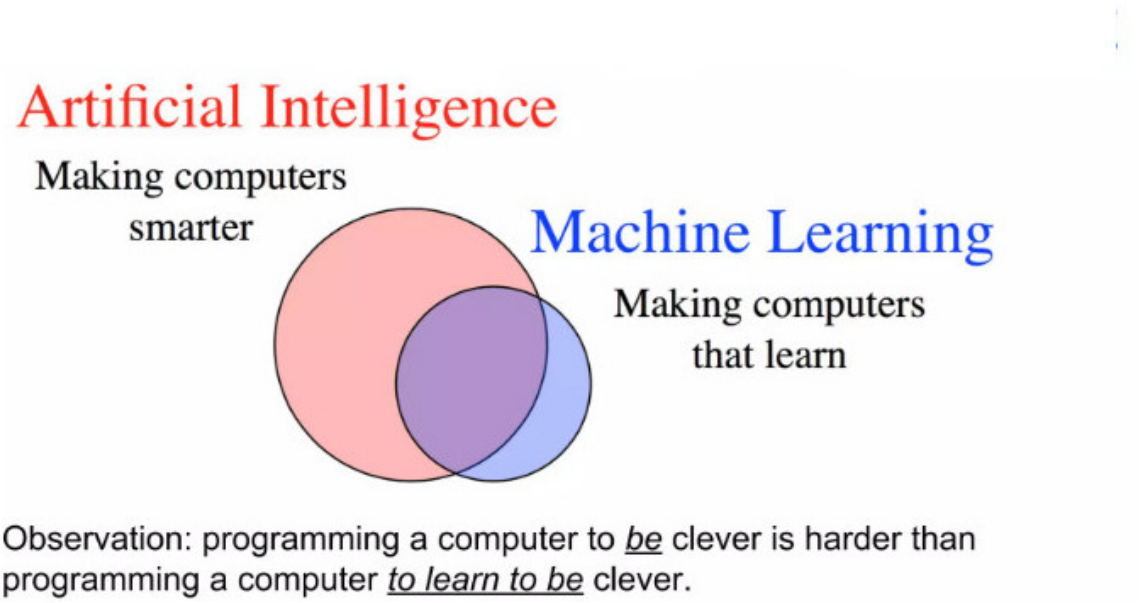


Figure 2.2: Figure showing a simplified explanation of the difference between AI and Machine Learning

As mentioned in several books, including the book on Deep Learning, machine learning was defined by Arthur Samuel in 1959 as the field that gives computers the ability to learn without being explicitly programmed. However, this definition is not very clear. Consequently, it has been reinforced by Tom Mitchell's definition of learning in 1997, which states: "A program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E" [64].

To automatically extract and exploit the information present in a dataset. In machine learning, tasks are generally classified into broad categories. These categories are based on how the learning is received or how feedback on the learning is given to the developed system. All machine learning algorithms fall into one of the following main types of learning:

- Supervised Learning.
- Unsupervised Learning.
- Semi Supervised Learning
- Reinforcement Learning.

In the (Figure 2.3) we can see the primary categories of ML and how they differ.

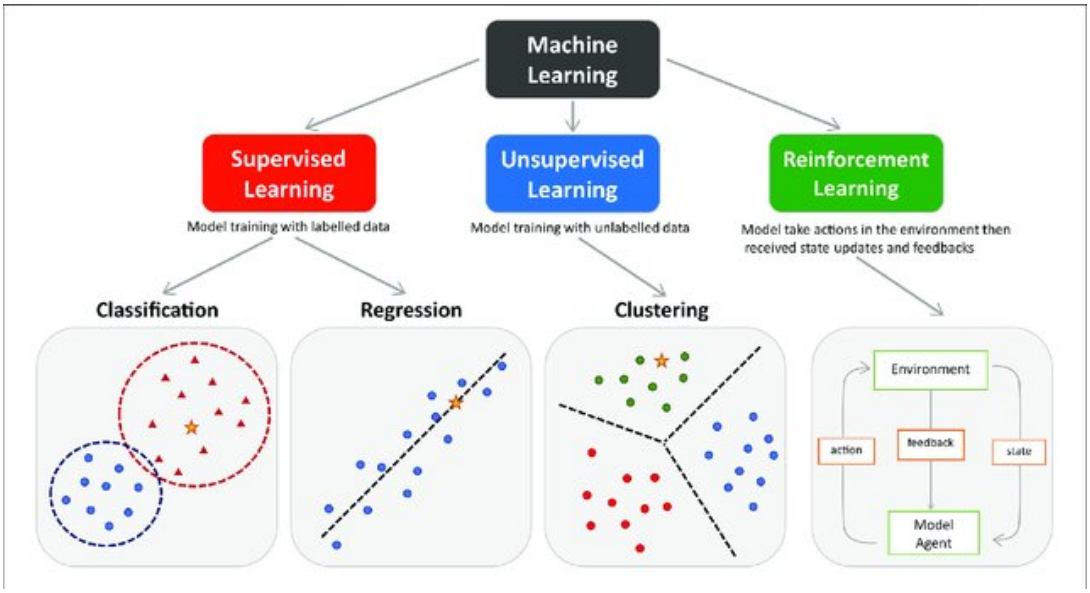


Figure 2.3: Figure illustrating the different type of machine learning approaches

Types of machine learning

This classification is primarily based on the type of input data: labelled or unlabeled, and the objective. We will give a brief overview.

- Supervised Learning

Supervised learning represents the predominant approach in machine learning, wherein algorithms are trained on labeled input data. Here, each example in the training set is paired with a corresponding desired output or label. The primary objective is for the algorithm to iteratively improve its predictions by comparing its generated outputs against the expected labels. Through this iterative process, the model adjusts its parameters to minimize errors, thereby enhancing its ability to generalize and predict accurately to new, unseen data.

- Unsupervised Learning

Contrary to supervised learning, unsupervised learning deals with unlabeled data. The algorithms in unsupervised learning autonomously identify underlying patterns and structures within the input data. This approach allows computers to uncover hidden relationships or groupings without explicit guidance from labeled examples. By exploring the Data's inherent structure, unsupervised learning algorithms can cluster similar data points, reduce dimensionality, or detect anomalies, depending on the specific task and objectives.

- Semi-Supervised Learning

Semi-supervised learning bridges the gap between supervised and unsupervised learning by leveraging both labeled and unlabeled data. In this approach, a small amount of

labeled data is used alongside a larger pool of unlabeled data to improve learning efficiency and accuracy. The primary goal is to utilize the abundant unlabeled data to augment the learning process, eventually reducing the dependency on extensive labeled datasets. Semi-supervised learning algorithms aim to identify patterns and relationships in the unlabeled data and use this information to enhance the model's performance on the labeled data. This method is particularly useful in situations where acquiring labeled data is costly or time-consuming, allowing models to achieve high performance with limited labeled examples.

- Reinforcement Learning

Reinforcement learning introduces an interactive paradigm where an agent learns to make decisions through trial and error within a dynamic environment. Drawing inspiration from behavioral psychology, reinforcement learning models how agents maximize cumulative rewards by exploring actions and learning from their outcomes. Unlike supervised and unsupervised learning, reinforcement learning is concerned with sequential decision-making processes, where actions influence subsequent states and rewards. This approach is particularly suited for scenarios requiring adaptive behavior and continuous learning in dynamic and uncertain environments.

Choosing an approach

Choosing a supervised or unsupervised learning algorithm usually depends on factors related to the structure and volume of our data, and the use case we want to apply it to. Machine learning has gained importance in many sectors, including:

- Customer lifetime value.
- Anomaly detection.
- Dynamic pricing.
- Predictive maintenance.
- Image classification.
- Recommendation systems.

2.2 Deep Learning

The term "Deep Learning" has existed for several decades, in other fields, particularly in education, but it was not until 1986 that it appeared in the field of machine learning. Deep Learning is an artificial intelligence function that imitates the way our brain processes data and creates models to use in decision-making [65]. Deep Learning is a subset of Machine Learning that has networks capable of learning from unstructured or unlabeled data without supervision, when it comes to building the models, it is up for the deep learning to learn the features and data to represent the input. The following Figures, (Figure 2.4) and (Figure 2.5) illustrates the relationship between these different AI disciplines and provides a high-level diagram of how each one works.

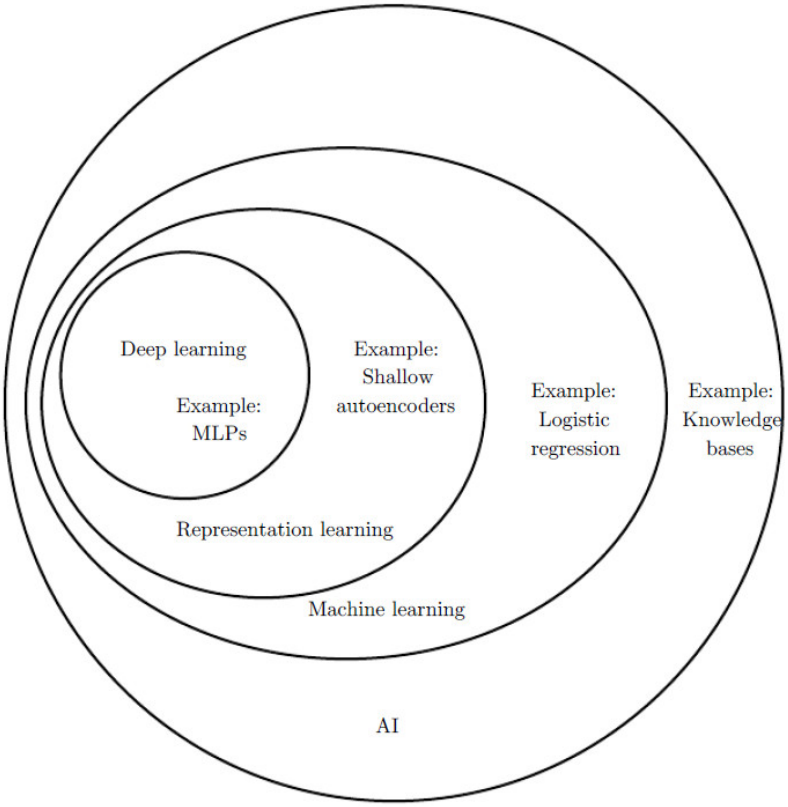


Figure 2.4: Venn diagram illustrating how deep learning is a type of representation learning, which in turn is a type of machine learning, with specific use cases [66]

The approach to feature extraction varies notably between Deep Learning (DL) and Traditional Machine Learning (ML) methods can be seen in (Figure 2.5).

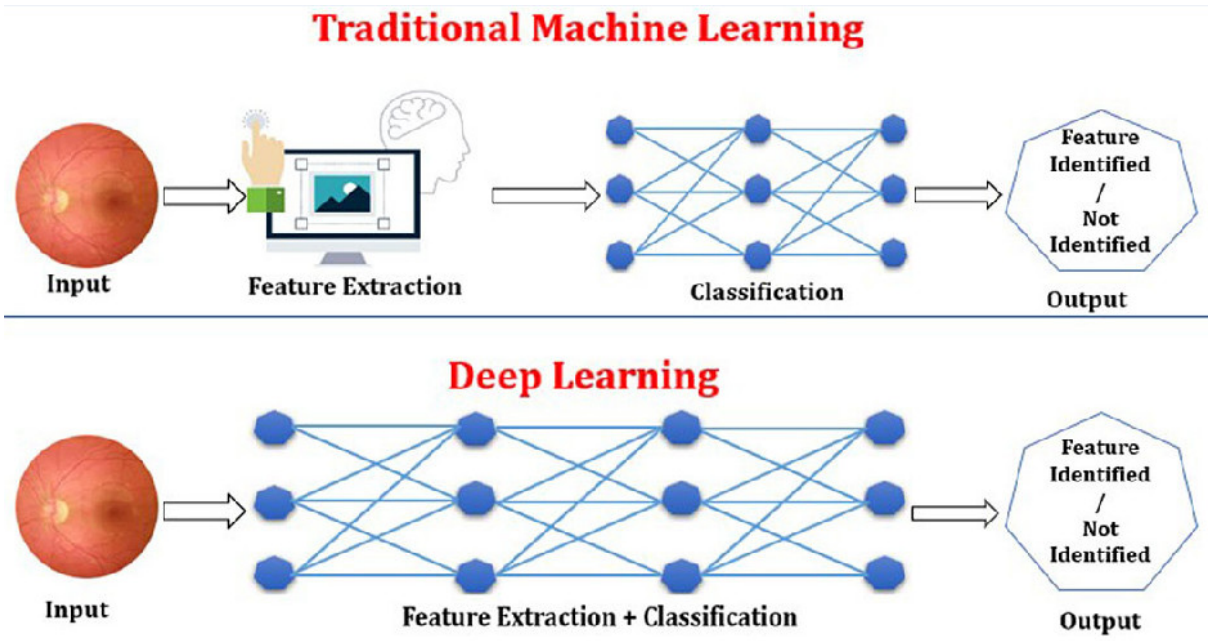


Figure 2.5: Difference of feature extraction between Deep Learning And Traditional Machine Learning

2.2.1 Artificial Neural Networks

An artificial neural network is a flexible mathematical structure capable of identifying complex nonlinear relationships between sets of input and output data. Artificial neural network models have proven to be useful and effective, especially in problems where the characteristics of the processes are difficult to describe using physical equations . The development of artificial neural networks stemmed from the attempt to simulate biological nervous systems by combining many simple computational elements (neurons) into a highly interconnected system, and hoping that complex phenomena such as intelligence would emerge from self-organization or learning. The illustration in (Figure 2.6) depicts a simple neural network.

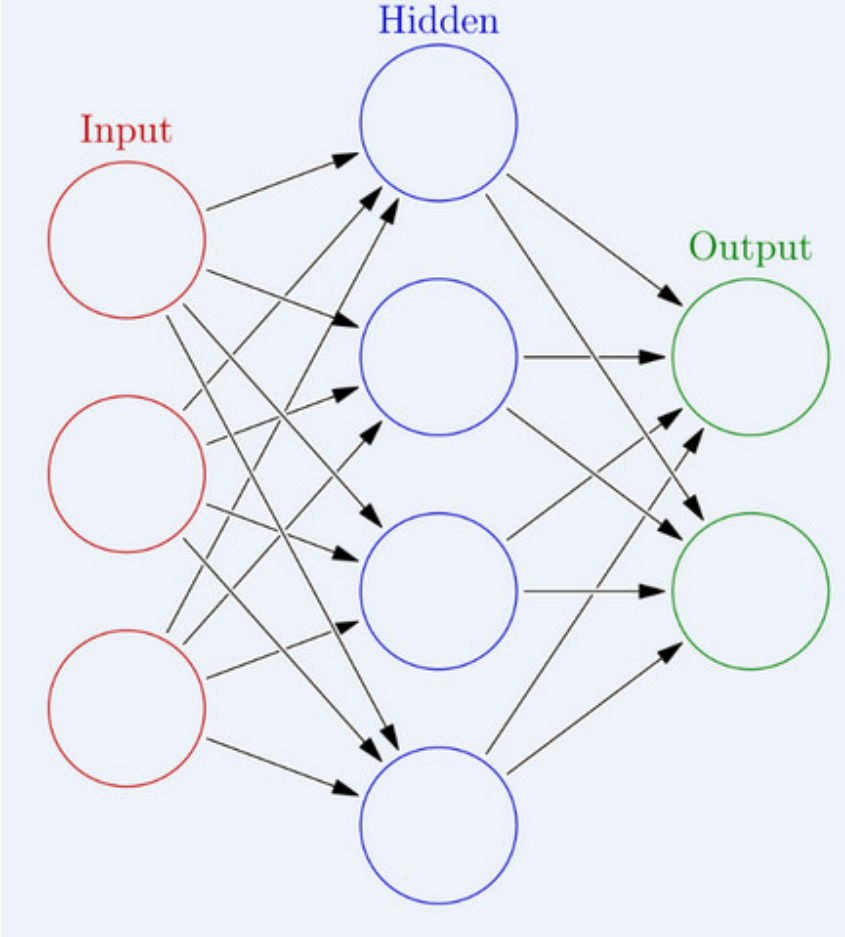


Figure 2.6: Illustration of a neural network

We can also use this (Figure 2.7) to see how neural networks reseemble biological neural networks.

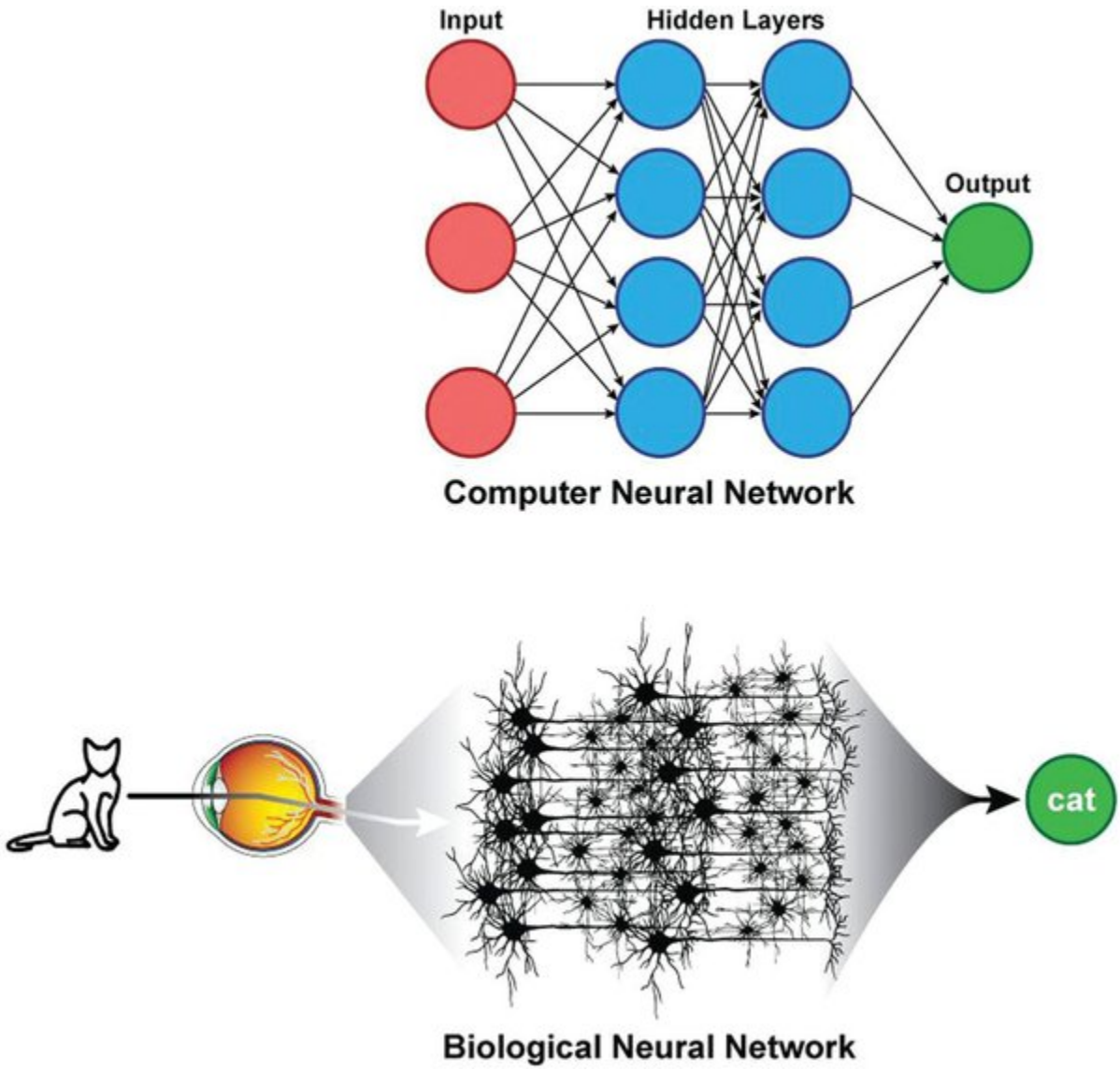


Figure 2.7: Biological Neural Networks vs Artificial Neural Networks

In machine learning, there is a theorem called the "No Free Lunch" theorem. In summary, it states that no algorithm works best for every problem, and this is particularly relevant for supervised learning (i.e., predictive modeling)

We cannot say that neural networks are always better than decision trees or vice versa. Many factors are at play, such as the size and structure of the data source. Therefore, it is recommended to try many algorithms for a given problem, while using a "test set" to evaluate performance and select the most performing model. Certainly, the algorithms must be adapted to the problem, i.e., choosing the right machine learning task. Deep Learning models can achieve a very high level of accuracy, which depends on several factors, such as the size and structure of the data source and the architecture of the network and/or the machine. The quality of the data has a direct impact on the effectiveness of the learning.

2.2.2 Structure of an Artificial Neuron

A perceptron represents a processing unit that receives input data, in the form of a vector, and produces a real output A neuron is characterized by an internal state and an activation function such that illustrated by the equation.

$$\text{Net} = b + \sum_{i=0}^{n} w_i x_i \tag{2.1}$$

Where b represents the bias, the bias is a measure of how easily the perceptron is triggered. The bias allows the network to try new interpretations or behaviors.

The weights (wi) associated with each neuron in a neural network play a crucial role by modifying the signal entering each neuron. These weights act as coefficients that either amplify or diminish the input trigger, and they are adjusted iteratively during the learning process to improve the network's performance. (Figure 2.8) illustrates the structure of an artificial neuron, depicting its connections and the role of weights in signal processing. Initially conceptualized in the form of perceptrons capable of approximating basic Boolean functions like AND and OR, neural networks evolved to include multilayer perceptrons (MLPs) to handle more complex tasks, such as approximating XOR and other nonlinear functions.

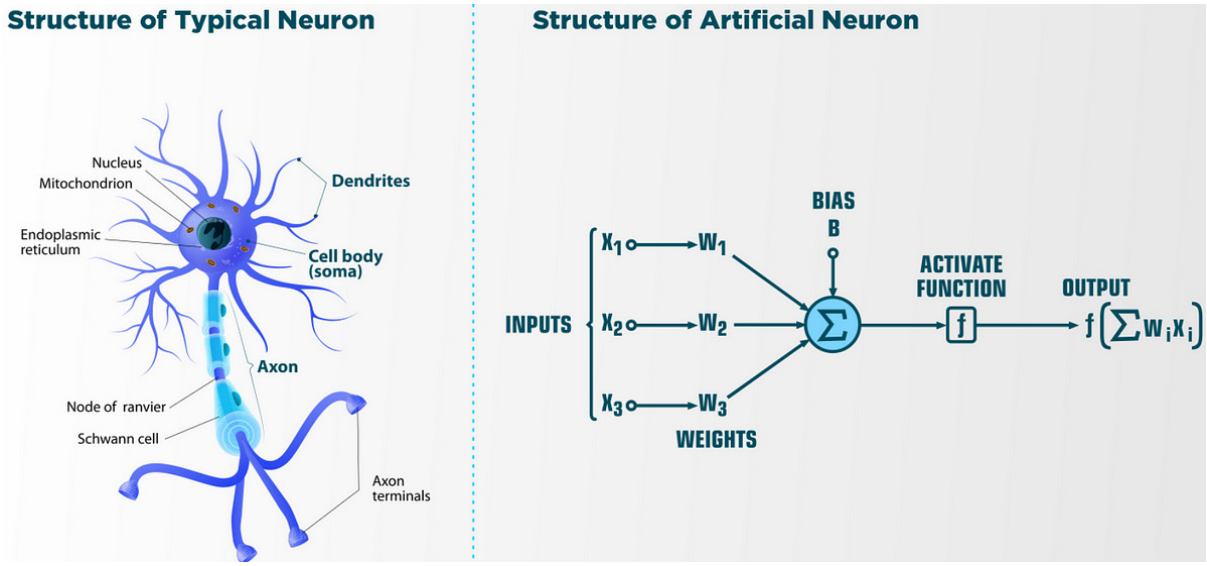


Figure 2.8: Structure Of The Artificial Neuron [67]

2.2.3 The Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) is a neural network-based classifier that is organized into multiple layers, with each layer made up of one or more formal neurons. When used in a supervised learning case, it uses the backpropagation algorithm. Multi-layer perceptrons are capable of processing data that is non-linearly separable. With the advent of backpropagation algorithms, they become the most widely used type of neural networks. MLPs are generally organized into three layers: the input layer, the intermediate (hidden) layer, and the output layer.

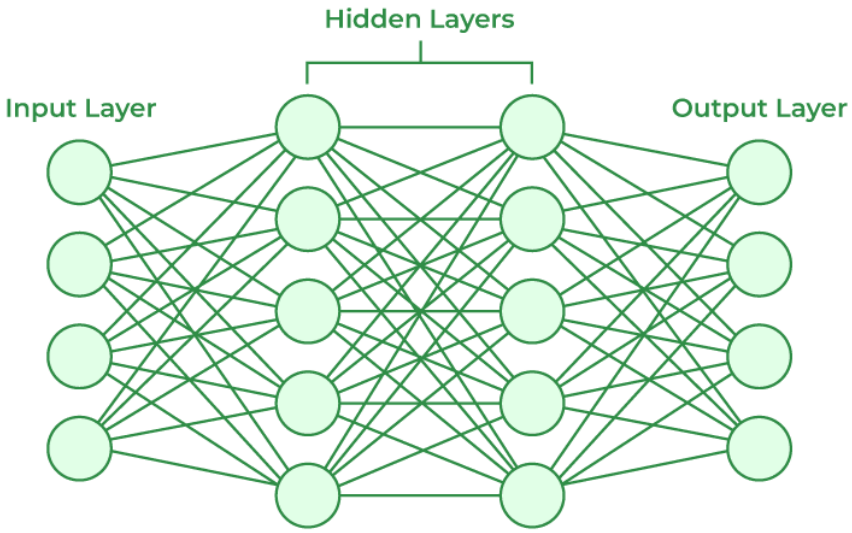


Figure 2.9: Architecture Of Multi Layer Perceptron

2.2.4 Activation Functions in Neural Networks

Activation functions differ depending on the objective of the layer. Activation functions transform the combination of inputs, weights, and biases. Among the most commonly used functions, we have:

- The linear activation function

$$f(x) = w \cdot x \tag{2.2}$$

The linear transfer function calculates the neuron's output by simply returning the value that was passed to it, it is mainly used regression tasks.

- The Sigmoid Function

$$\sigma(x_i) = \frac{1}{1 + e^{-x_i}} \tag{2.3}$$

It takes a real input and scales it between 0 and 1. This output will be a continuous value between 0 and 1.

- SoftMax Function

This is a generalization of the Sigmoid function. It takes a vector as input and gives a vector as output. It is often used in the output layer of a classifier.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2.4}$$

2.3 How Does the Model Learn and Adjust Itself?

In the process of learning, the Neural networks adjust its neurons weights and biases by leveraging gradients computed during backpropagation. Initially, the network makes predictions based on its current, often randomly initialized weights. These predictions are compared to the actual target values using a chosen loss function, generating an error signal. Through backpropagation, this error signal is propagated backward layer by layer, computing gradients of the loss function with respect to each weight and bias. These gradients indicate the direction and magnitude of adjustments needed to minimize the error. The weights and biases are then updated using an adequate optimization algorithm such as Adam, SGD optimizer, which are famous optimization algorithms for neural networks that combine elements of adaptive learning rates and momentum. Adam adjusts the learning rate for each parameter individually. This iterative process continues over multiple epochs, gradually refining the model's parameters until convergence, where further adjustments yield minimal improvement in the loss function. Meaning, by iteratively updating parameters based on computed gradients with the Adam optimizer, the MLP learns to better fit the training data and generalize to unseen examples.

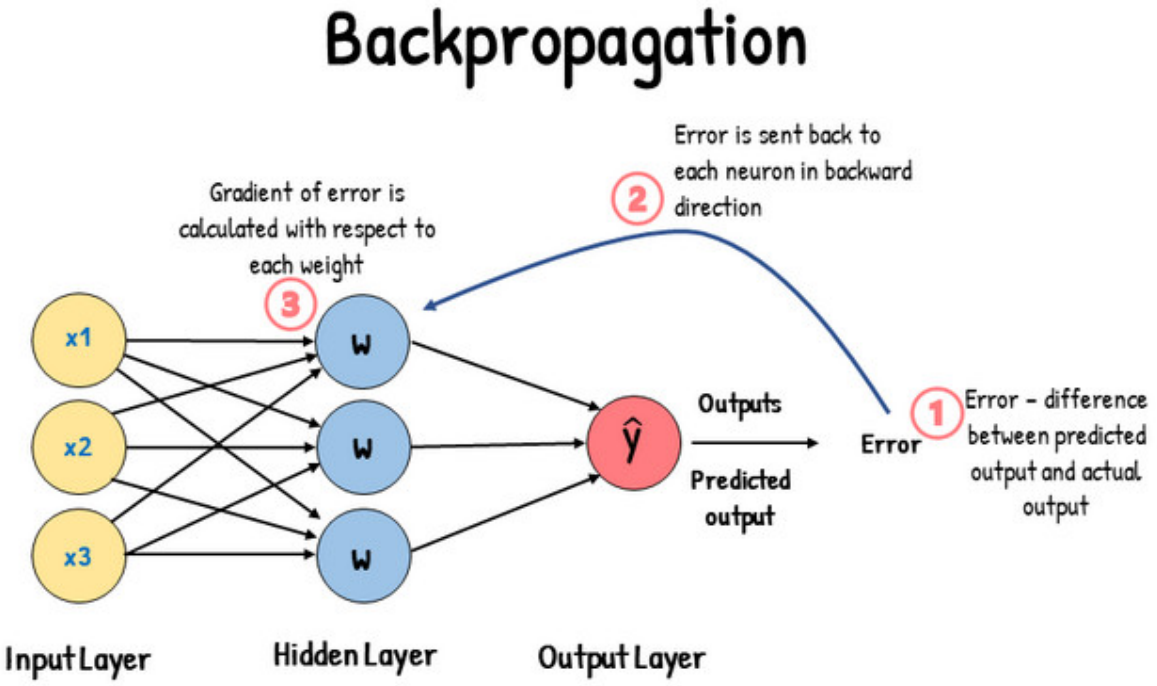


Figure 2.10: Abstract Illustration of the backpropagation algorithm [68]

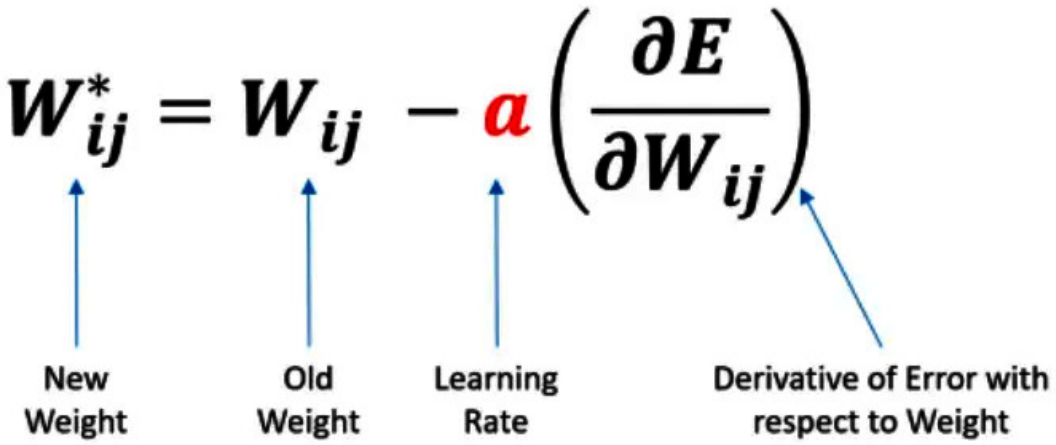


Figure 2.11: Formula of the adjustment of weights during training by the loss function optimizer [69]

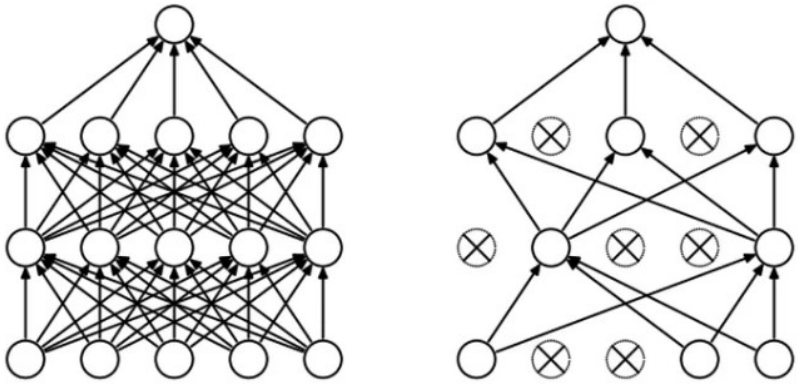
2.3.1 Regularisation Methods

When models accumulate a large number of parameters and increase in complexity, they often exhibit elevated weights. However, such models are susceptible to poor generalizability, as they may learn noise data instead of capturing relevant patterns [70]. To mitigate this, regularization techniques are commonly used in the literature. These methods introduce penalty terms into the loss function, aiming to restrain excessively large weights. The aim of regularization is to discourage the model from overly relying on dominating weights by penalizing their magnitudes. Two prominent formal regularization methods, L1 and L2, they differ mainly in their penalty terms.

In addition to L1 and L2 regularization, another widely used technique is dropout. Dropout involves randomly deactivating a fraction of neurons during training, thereby reducing the model's reliance on specific neurons and preventing overfitting. This technique has been instrumental in improving the generalization performance of complex models.

- Dropout

Dropout is a regularization method which is used in artificial neural networks in order to prevent overfitting by introducing randomness into the learning process [71]. During training, a certain percentage of neurons, for example 50%, are "dropped out," meaning they are temporarily deactivated and their contributions to the activation of further neurons are disabled. This results in the creation of a different network architecture for each training step, effectively simulating an ensemble of neural networks. By doing so, dropout helps to mitigate the risk of the network becoming overly dependent on specific features or patterns, the (Figure 2.12) shows what happens at the level of the network once dropout is used.



(a) Standard Neural Net (b) After applying dropout.

Figure 2.12: Visualization depicting the impact of Dropout regularization technique in neural networks.

- L1 Regularization

L1 regularization constraints weight values towards zero by addition the loss function with the sum of their absolute magnitudes [70]. The value of a determines the degree of the penalty, so the larger the value of a, the higher the constraint to the weights, Usually $0 < \alpha < 1$. L1 regularization can be formally defined as

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \alpha \sum_{j=1}^{m} |w_j| \tag{2.5}$$

$L(y_i, \hat{y}_i)$ represents the loss function used to measure the discrepancy between the predicted output $\hat{y}_i$ and the actual label y_i.

n is the number of training examples, m denotes the number of weights, wj indicates the weight associated with neuron j, yi stands for the corresponding label, and a is the regularization coefficient.

- L2 regularization

L2 regularization addresses the problem of large weights in the network by addition of the sum the squares of the weights to the loss function. The degree of the penalty is also determined by the same value of a.

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \alpha \sum_{j=1}^{m} w_j^2 \tag{2.6}$$

$L(y_i, \hat{y}_i)$ represents the loss function used to measure the discrepancy between the predicted output $\hat{y}_i$ and the actual label y_i.

n is the number of training examples, m denotes the number of weights, wj indicates the weight associated with neuron j, and a is the regularisation coefficient.

2.4 Deep Learning Models

There are several neural network architectures, each with their own strengths and weaknesses, and they can be combined to optimize results. Most are derived from certain

original parent architectures. The choice of architecture is thus crucial and is mainly determined by the objective. These are the ones centered at our work for today:

Convolutional Neural Networks

Convolutional neural networks are currently the most powerful models for image classification [72]. Designed with the acronym CNN, Convolutional Neural Network, they are a type of neural network specialized for processing data with a grid-like topology. CNNs have proven to be highly effective in identifying faces, objects, traffic signs, and autonomous car driving. They have also been effective in various natural language processing tasks (such as sentence classification).

Convolutional neural networks are based on the multilayer perceptron (MLP), and their operation is inspired by biological processes, more precisely, the behavior of the visual cortex of vertebrates. While effective for image processing, MLPs struggle to handle large images, due to the exponential growth in the number of connections with the size of the image.

Regular neural networks do not adapt well to all kinds of images. For example, if an image is only 32x32x3 (32 wide, 32 high, 3 color channels), a single fully connected neuron in the first hidden layer of a normal neural network would have 32*32*3=3072 weights. This amount still seems manageable, but it's clear that this fully connected structure does not adapt to larger images. For example, a more respectable image size, e.g., 200x200x3, would lead to neurons with 200*200*3=120,000 weights. Moreover, we would almost certainly want to have several of these neurons, so the parameters would quickly add up, Clearly, this complete connectivity is a waste [73].

A CNN architecture is formed by a stack of independent processing layers

- The convolutional layer (CONV) that processes data from a receptive field.
- The rectification layer (ReLU), often referred to as 'ReLU' in reference to the activation function (Rectified Linear Unit).
- The pooling layer (POOL), which compresses the information by reducing the size of the intermediate image.
- The "fully-connected" layer (Fully-connected layer FL) is a fully connected layer of classification, a neural layer.

There is also a batch normalization layer which we will be mentioning and expanding on later on in the third chapter, the reason being is that not all CNN architectures and models use this.

- The Convolutional Layer (CONV)

The purpose of the convolution operation is to extract high-level features such as edges from an input image. The convolutional layer receives various input images and then calculates the convolution of each of them with each filter. To apply a good convolution, the filters must correspond nearly exactly to the features we want to find in the images. The convolutional layer applies a series of filters (also known as kernels) to the

input image or feature map when the convolutional layer is in the middle of the network. Each filter is a small matrix of weights that slides over the input image and the outputs of the consequent convolutional layers.

The filter moves to the right with a certain step value until it analyses the full width. Continuing, it jumps back to the beginning (left) of the image with the same step value and repeats the process until the entire image is covered.

For each pair (image, filter), we obtain an activation map, or feature map, which indicates where the features are located in the image: the higher the value, the more the corresponding location in the image resembles the feature.

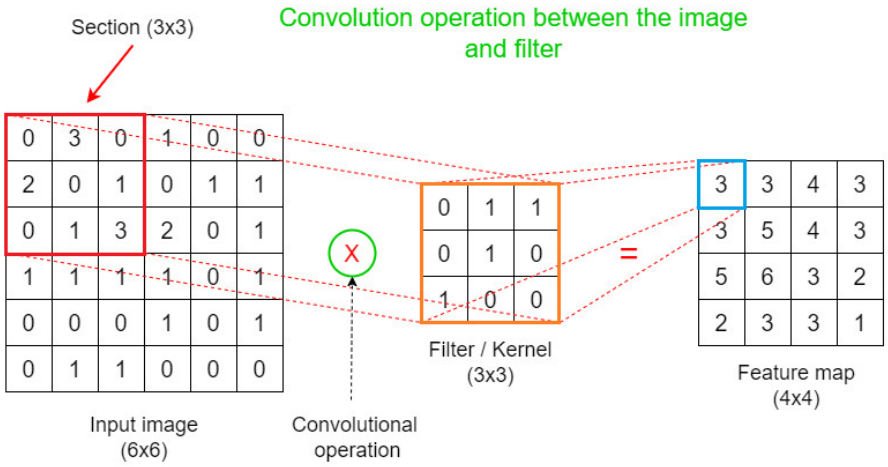


Figure 2.13: Visual illustration of the conv layer taking features from an input image after applying the multiplication of pixel values with the kernel [74]

There is also the stride parameter, which refers to the step size at which the filter moves across the matrix of the input image, from left to right, a smaller stride means the filter covers less ground per step, leading to a larger output feature map but potentially missing finer details. Conversely, a larger stride means the filter skips more pixels, resulting in a smaller output feature map but potentially capturing broader patterns.

Feature Selection

The features are not predefined according to a particular formalism, but are learned by the network during the training phase. The weights of the filters used the convolutional layers are initialized randomly and then updated by back propagating the gradient.

In the case of images with multiple channels (e.g., RGB), the kernel has the same depth as the input image. Matrix multiplication is performed between the kernel stack *Kn* and the input *In ([K1, I1]; [K2, I2]; [K3, I3])*, and all the results are summed with the bias to give us a convoluted output function of a squeezed channel depth. Also, the first convolutional layers usually capture a more simplistic high level and generalist features like shapes, edges, corners but as the convolutional layers their ability to capture features becomes more sophisticated and detailed. The (Figure 2.14) gives a simple idea

of how at works.

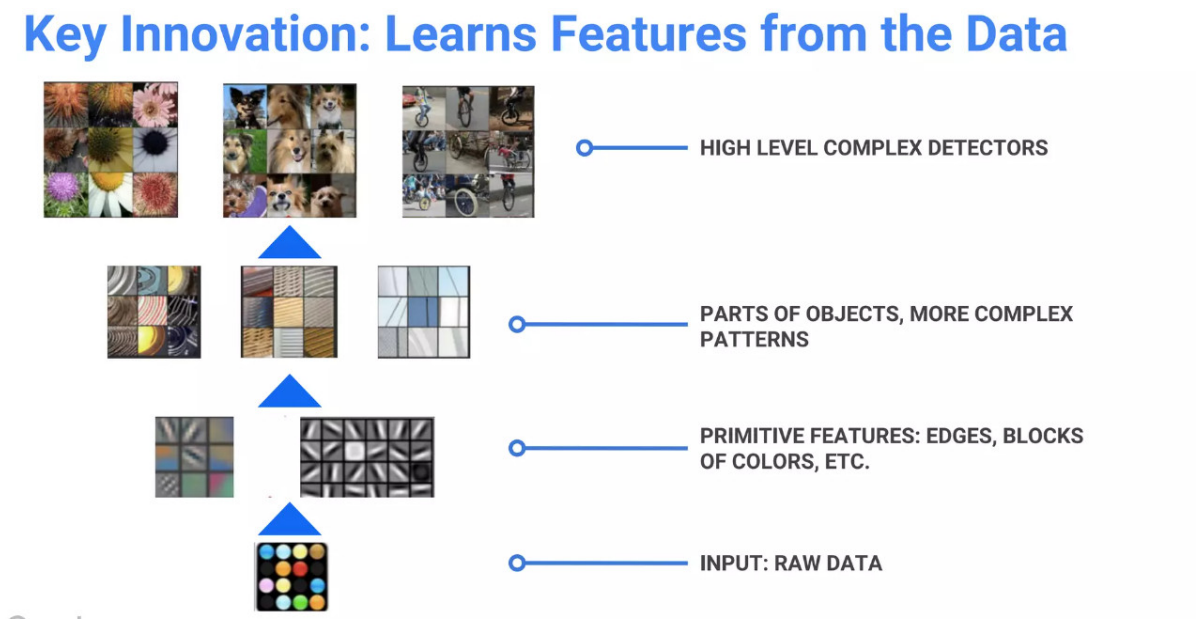


Figure 2.14: Evolution of feature representation hierarchy in convolutional layers.

The formula of convolutional operation is formally defined [75] as:

$$x(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.7}$$

- $x(i,j)$ is the output of the pixel at position (i,j) in the feature map.
- I refers to the region of the image under processing, or the input matrix.
- K is the convolution kernel or filter that is applied to the input.
- $\sum_m \sum_n$ represents the sum over all valid positions of m and n where the kernel K is applied to the input I.
- $I(m,n)$ denotes the value of the input pixel located at the position (m,n).
- $K(i-m, j-n)$ is the convolution kernel's value at the position determined based on the offset applied from the output position. (i,j).

- The ReLU Rectification Layer

ReLU (Rectified Linear Units) refers to the real non-linear function defined by:

$$ReLU(x) = x^+ = \max(0, x) \tag{2.8}$$

This function forces the neurons to return positive values. The ReLU rectification layer thus replaces all negative values received at the inputs with zeros. It acts as an activation function after applying the convolution operation to produce the final feature map.

- The LeakyReLU Rectification Layer

Some Convolutional Neural networks can opt to A leaky rectified linear activation layer is a non-saturated activation function that allows some negative gradients to pass. It is used to reduce the effect of the negative gradients by a specific factor.

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases} \tag{2.9}$$

- The Pooling Layer

This type of layer is often placed between two convolutional layers, or multiple convolutional layers. Similar to the convolutional layer, the pooling layer is responsible for reducing the spatial size of the convolved entity, the feature maps. The goal is to decrease the computational power required to process the data by reducing dimensionality. Furthermore, it is useful for extracting dominant features that are invariant to rotation and position, keeping the model's learning process efficient. there are numerous types of pooling used.

Max Pooling: This operation selects the maximum of values in the specified region.

$$\text{MaxPooling}(x_{i,j}) = \max_{(p,q)\in R_{i,j}} x_{p,q} \tag{2.10}$$

Average Pooling: This operation calculates the average of the values in a specified region.

$$\text{AveragePooling}(x_{i,j}) = \frac{1}{|R_{i,j}|} \sum_{(p,q)\in R_{i,j}} x_{p,q} \tag{2.11}$$

Sum Pooling: This operation calculates the sum of values in the specified region.

$$\text{SumPooling}(x_{i,j}) = \sum_{(p,q)\in R_{i,j}} x_{p,q} \tag{2.12}$$

- Min Pooling: This operation selects the minimum of values in the specified region.

$$\text{MinPooling}(x_{i,j}) = \min_{(p,q)\in R_{i,j}} (x_{p,q}) \tag{2.13}$$

Below are Figures showing the mainly used pooling methods, max and average pooling.

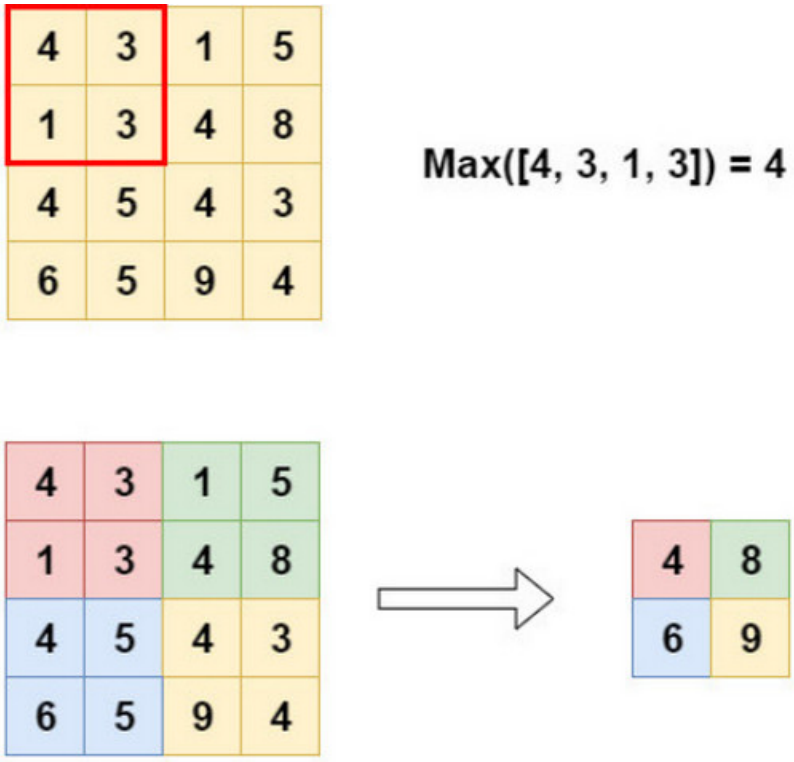


Figure 2.15: Max pooling operation on the feature map

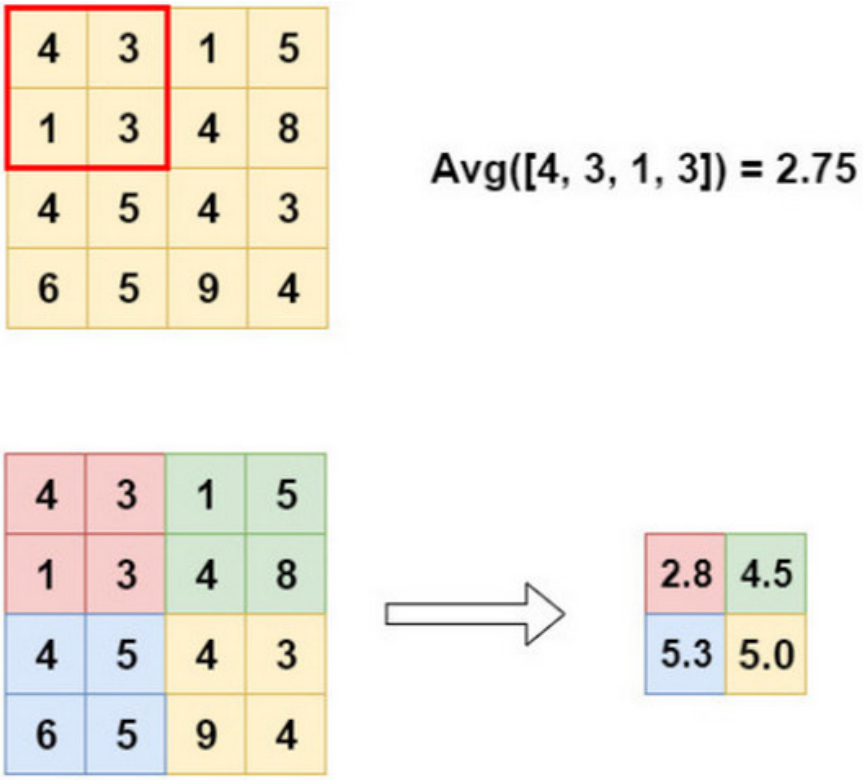


Figure 2.16: Average pooling operation on the feature map

Similar to the convolutional layer, the pooling layer uses a certain size (e.g., 2x2) and stride (how much the pooling window moves across the feature map) to perform the

pooling operation. Smaller sizes and strides lead to more aggressive downsampling.

- Fully Connected Layer

After the convolutional and pooling layers extract the relevant features from the input images, the final stages of a typical convolutional neural network (CNN) involve using those extracted feature maps for classification. This is accomplished by attaching either a single perceptron or a multi-layer perceptron (MLP) to the end of the network architecture. A perceptron is the most basic form of an artificial neural network unit. It takes the flattened feature maps from the previous convolutional/pooling layers as input and produces a vector output of size N, where N represents the number of classes in the image classification problem at hand. Each element of this output vector indicates the probability or "score" for the input image belonging to a particular class.

For instance, in a binary classification problem to distinguish between images of cats and dogs, the final output vector would have a size of 2. The first element would represent the probability of the input image belonging to the "cat" class, while the second element would be the probability for the "dog" class. Essentially, each element in the flattened feature map input "votes" in favor of a specific class, with learned weights determining the importance or contribution of each vote. However, for more complex classification problems involving numerous classes, it is common to use a multi-layer perceptron (MLP) instead of a single perceptron at the end of the CNN architecture. An MLP consists of multiple fully-connected layers, each composed of artificial neurons that combine the inputs from the previous layer using learnable weights and non-linear activation functions.

In this case, the flattened feature maps from the convolutional/pooling layers would serve as the input to the first fully-connected layer of the MLP. The subsequent layers of the MLP would then process and transform this information further, with the final layer producing the output vector of size N, where N is the number of classes in the classification problem. The flatten layer is a crucial component in convolutional neural networks (CNNs) that is used to transition between the convolutional/pooling layers and the fully-connected layers.In a CNN architecture, the convolutional and pooling layers are designed to extract features from the input images, These layers output 3D volumes of feature maps, where each 2D feature map represents different aspects of the input image, such as edges, textures, or patterns.

However, the fully-connected layers in a neural network expect a 1D vector as input. This is where the Flatten layer comes into play. Its primary function is to reshape the 3D output from the convolutional/pooling layers into a 1D vector, also known as a feature vector.

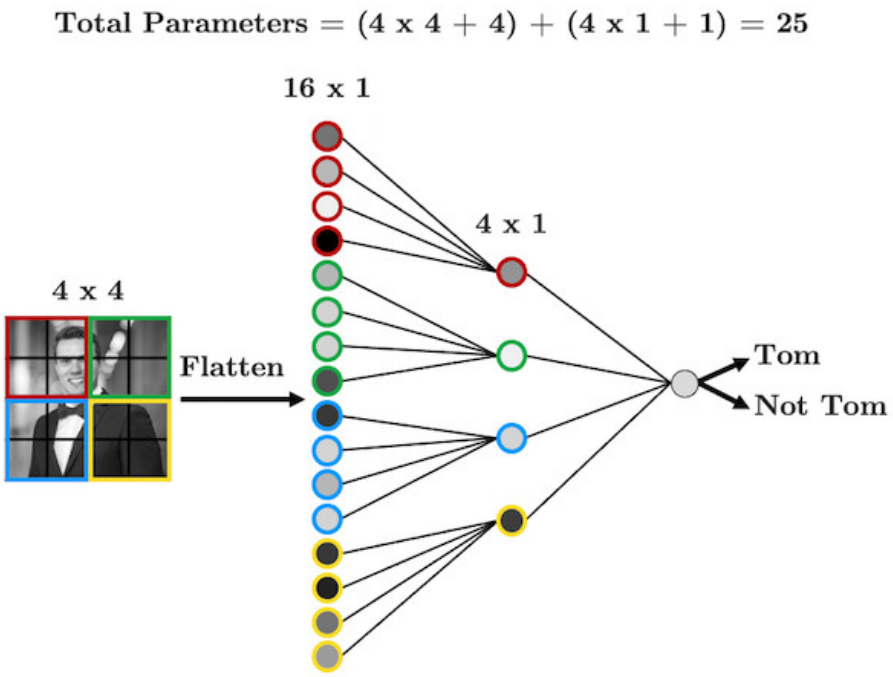


Figure 2.17: Illustrative chart of the flatten layer [76]

the flatten layer takes the 3D output from the previous convolutional or pooling layer as input. Let's assume the input shape is (batch size, height, width, channels), where batch size is the number of images in the current batch, height, and width represent the spatial dimensions of the feature maps, and channels is the number of feature maps. The flatten layer "flattens" or "unrolls" the 3D input tensor into a 2D tensor with shape (batch size, height, width, channels). This is achieved by concatenating all the feature maps along their spatial dimensions (height and width) for each image in the batch. The resulting 2D tensor is a 1D vector for each image in the batch, where the vector length is equal to (height X width X channels). This 1D vector, also known as a feature vector, contains all the relevant features extracted by the convolutional and pooling layers, but without any spatial information.

The purpose of the flatten layer is to prepare the data for the fully-connected layers, which operate on 1D vectors. These fully-connected layers can then learn complex combinations of the extracted features and perform classification or regression tasks based on the flattened feature vectors. It's important to note that while the flatten layer discards the spatial information present in the feature maps, the convolutional and pooling layers earlier in the network have already captured and encoded this spatial information in the form of learned feature representations.

If the task is to classify images into 5 different classes, the final output vector from the MLP would have a size of 5. Each element in this vector would represent the probability or "score" of the input image belonging to a specific class. During training, the network learns to adjust the weights and biases of the MLP layers, as well as the convolutional layers, to optimize these class probability scores based on the provided labeled training data.

It's worth noting that the final output scores from the perceptron or MLP are typically passed through a softmax activation function, which normalizes the scores to represent

proper probability distributions that sum to 1 across all classes. This allows the network to output well-calibrated class probabilities for decision-making and evaluation purposes.

2.5 U-NET Architecture

The U-Net architecture is a convolutional neural network (CNN) commonly used in biomedical image segmentation tasks. Developed by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015 [77], U-Net has since become a notable model in the field of medical image analysis due to its effectiveness in segmenting structures of interest from complex backgrounds. It is particularly adept at segmenting biomedical images where the size of the training dataset is limited.

The U-Net architecture derives its name from its U-shaped architecture, consisting of a contracting path followed by an expansive path. This architecture enables the network to effectively capture both local features and global context, making it well-suited for semantic segmentation tasks.

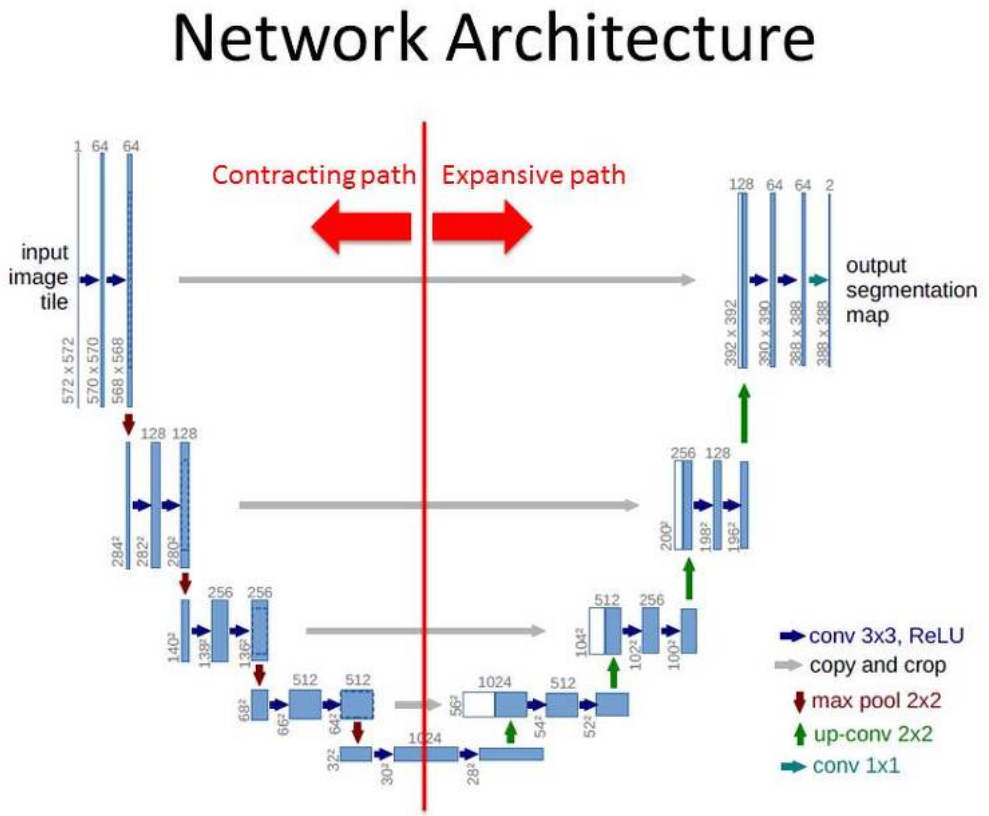


Figure 2.18: U-Net architecture [77]

- Contracting Path (Encoder)

The contracting path in the U-Net architecture is crucial for the initial analysis of the input image, aiming to identify key features and patterns. This section of the network employs a sequence of operations designed to progressively decrease the spatial dimensions of the image while simultaneously expanding the depth of feature channels. At the heart

of this process are convolutional layers, which apply a set of learnable filters to the image. These filters are capable of extracting a wide range of features, from low-level details like edges and textures to high-level patterns such as object parts.

Each step in the contracting path typically involves two 3x3 convolutional layers, each followed by a rectified linear unit (ReLU) activation function to introduce non-linearity into the model. This allows the network to learn complex patterns beyond simple linear relationships. After passing through these convolutional layers, the feature maps undergo max-pooling, a downsampling strategy that selects the maximum value from a small region of the image. This operation reduces the spatial dimensions of the feature map, focusing on retaining the most prominent features while discarding less significant details.

By iterating this process of convolutional operations followed by max-pooling across several steps, the contracting path effectively captures a hierarchical representation of the input image. It extracts detailed features at various scales, from fine-grained textures at the early stages to more abstract representations towards the deeper layers. This hierarchical feature extraction is essential for the subsequent expansive path, where these features are upsampled and combined to produce precise segmentations.

- Expansive Path (Decoder)

Unlike the traditional convolutional layers found in the contracting path, the expansive path, or decoder, of the U-Net architecture employs a distinct mechanism to generate the final segmentation mask. This section of the network is tasked with the process of incrementally restoring the spatial dimensions of the feature maps to match the original input image's resolution. To achieve this, the expansive path incorporates a sequence of upsampling techniques paired with convolutional operations, which serve to expand the feature maps while concurrently diminishing the depth of feature channels.

The upsampling process is crucial as it enhances the spatial resolution of the feature maps, providing the network with the capability to reconstruct the segmentation mask with exceptional detail and precision. This is a departure from the conventional use of convolutional layers alone, as seen in the encoding phase, highlighting the U-Net's innovative approach to handling spatial information.

A defining characteristic of the U-Net architecture is its utilization of skip connections that link the contracting and expansive paths. These connections function as conduits, facilitating the integration of high-resolution feature maps obtained from the contracting path with the corresponding layers in the expansive path. This strategic fusion of features from both pathways empowers the decoder to pinpoint and recognize features with enhanced accuracy and reliability. Consequently, this integration leads to the creation of a segmentation mask that is not only highly precise but also rich in detailed information, underscoring the effectiveness of the U-Net in image segmentation tasks.

- Skip Connections

Skip connections are for direct transmission of information from the contracting to the expansive path. These connections merge feature maps derived from the contracting path with those of corresponding spatial resolutions in the expansive path. This integration preserves intricate spatial details from earlier layers, compensating for the loss incurred during downsampling. Consequently, segmentation accuracy is enhanced as spatial intricacies are retained.

- Final Layer

The final layer of the U-Net architecture typically consists of a 1x1 convolution followed by a softmax or sigmoid activation function. For binary segmentation tasks, such as foreground-background segmentation, a sigmoid activation function is used to produce a probability map indicating the likelihood of each pixel belonging to the foreground class. - For multi-class segmentation tasks, a softmax activation function is employed to compute the probability distribution over multiple classes for each pixel.

Training Principle

- Forward Pass:

During the forward pass, the input image first goes through the contracting path, where a series of convolutional and max-pooling operations are applied to extract hierarchical features from the image. Next, in the expansive path, the feature maps from the contracting path are upsampled and combined with the corresponding feature maps from the contracting path using skip connections. As the feature maps progress through the decoder, the network gradually reconstructs the segmentation mask.

- Loss Computation:

After the segmentation mask has been produced, it undergoes evaluation against the ground truth mask, which serves as the benchmark for precision established during the training phase. The comparison leverages appropriate loss functions adequate to the goal of the segmentation task at hand. For binary segmentation, where the objective is to distinguish between foreground and background, binary cross-entropy loss is commonly employed. In scenarios involving multi-class segmentation, where the challenge lies in differentiating among multiple objects or categories, categorical cross-entropy loss is utilized for example. Through this comparative analysis, the training process aims to adjust the network's parameters via backpropagation, with the ultimate goal of minimizing any mistakes between the generated segmentation mask and the intended ground truth mask. This iterative refinement seeks to generate the most effective and precise filters within the network, achieving the desired level of segmentation accuracy.

- Inference

during inference (when the trained model is used on new, unseen images), the input image is passed through the trained U-Net model. The model then generates a segmentation mask, indicating the regions of interest or different objects within the image.

The U-Net architecture is notable in the field of deep learning due to its multitude of benefits. The capacity to efficiently extract features is especially essential in applications such as biomedical image analysis, where the identification of small details within complex images is important. The capability of these architectures is enhanced by its outstanding parameter efficiency, which is achieved through the use of skip connections. The relationships in U-Net enable the reuse of feature maps from previous layers, so improving both the efficiency of training and deployment, particularly in contexts with

limited resources. Moreover, U-Net remains robust in situations where there is a limitation of data, which is often the case in biomedical imaging. Its adaptability extends to other fields beyond the segmentation of biomedical images, including satellite imaging, microscopy, and industrial quality control. U-Net's flexible structure makes it well-suited for a wide range of segmentation tasks in diverse fields.

2.6 Transfer Learning

Infinite numbers of real-world applications use Machine Learning (ML) techniques to develop potentially the best data available for the users. Transfer learning (TL) [78], one of the categories under ML, has received much attention from the research communities in the past few years.

Traditionally, ML algorithms perform under the assumption that a model uses limited data distribution to train and test samples. These conventional methods predict target tasks undemanding and are applied to small data distribution. However, this issue conceivably is resolved using Transfer Learning. Transfer Learning is acknowledged for its connectivity among the additional testing and training samples, resulting in faster output with efficient results.

People can hardly afford the luxury of investing resources in data gathering in today's world since they are rare, inaccessible, often expensive, and difficult to compile. As a result, most people found a better means of data collection: one of the ways is to transfer knowledge between the tasks [. This has inspired Transfer Learning (TL): to improve data gathering and learn in machine learning (ML) using the data compiled before it has been introduced. Most of the algorithms of ML are to predict future outcomes, which are traditionally in the interest of addressing tasks in isolation Whereas Transfer Learning does the otherwise, it bridges the data from the source and targets the task to find a solution, perhaps a better one.

The pre-trained models are usually trained on massive datasets that are a standard benchmark in the computer vision frontier. These models can be used directly in making predictions on new tasks, or integrated into the process of training a new model. Including the pre-trained models in a new model leads to lower training time and lower generalization error.

2.6.1 Key Components of Transfer Learning

- Source Task

The task for which a model is initially trained with abundant labelled data. For instance, training a model to classify images using a large dataset like ImageNet. Which contains more than 1 million and 200 thousand images, and 1000 object classes.

- Target Task

The new task for which the pre-trained model's knowledge is transferred into. This task typically has a smaller dataset or different data distribution compared to the source task.

- Model Architecture

The neural network architecture used for the pre-trained model. Common architectures include Convolutional Neural Networks (CNNs) for image-related tasks and Recurrent Neural Networks (RNNs) for sequential data tasks. Techniques such as feature extraction, fine-tuning, and domain adaptation are used to transfer knowledge effectively between tasks and domains. The advantages of Transfer Learning are numerous, it reduces data requirements. By leveraging pre-trained models, transfer learning reduces the need for large labelled datasets, making it effective to train models in data-scarce domains. Also training a model from scratch can be computationally intensive and costly and very much time-consuming. Transfer learning speeds up training by starting with a model that has already learned basic features from vast datasets. There is also an improved generalization because pre-trained models often capture generic features from data, which can improve model generalization and performance on unseen data in the target task.

In the feature extraction part, the pre-trained model is used as a fixed feature extractor, meaning its weights are frozen and not updated during training on the new task. The pre-trained model is typically composed of two parts, a feature extractor and a classifier. In the transfer learning we will be making our own customized fully connected layers and classifiers that suit our task, which will be a deep neural network, or in other terms multi perceptron neural network.

The feature extractor extracts meaningful representations or features from the input data, leveraging the knowledge learned during pre-training. The extracted features are then used as input to a new, task-specific classifier or regressor, which is trained from scratch or with a different initialization on the new task's dataset.

Fine-tuning by unfreezing layers in the layers of the architecture of the pretrained model involves a strategic adjustment of the training process to enhance the model's performance on a specific task. Initially, during the "warm-up" phase, all layers except the final fully connected (FC) layers are frozen, meaning their weights are not updated during the training phase. This approach ensures that the features learned by the pretrained convolutional layers remain intact, while the newly added fully connected layers adapt to the task at hand. After the Fully connected layers have started to learn patterns from the dataset, the training process pauses, and the previously frozen layers are unfrozen. These layers are then trained with a very small learning rate to prevent major alterations to the convolutional filters. This gradual unfreezing and retraining of layers allows the model to leverage both the general knowledge encoded in the pre-trained layers and the task-specific adaptations in the FC layers, leading to improved performance on the target task. This method is particularly effective for adapting pre-trained CNNs to classify images in custom datasets, offering a powerful alternative to traditional transfer learning approaches.

Conclusion

In summary, this chapter has provided a comprehensive overview of the fundamental concepts and techniques that underpin the deep learning models employed in this thesis work. We have explored the historical development of artificial intelligence and machine learning, tracing their evolution from early symbolic approaches to the rise of powerful neural network architectures driven by increasing computational capabilities and data availability. Central to our work are convolutional neural networks (CNNs), which have demonstrated remarkable success in computer vision tasks by automatically learning hierarchical feature representations from raw image data. We have gone into the architectural details of CNNs, elucidating the roles of convolutional, pooling, and fully-connected layers in extracting and aggregating multi-level features for image understanding. Furthermore, we have examined the U-Net architecture, a specialized CNN tailored for precise image segmentation tasks. Its unique encoder-decoder structure, coupled with skip connections, enables the model to capture both local and global contextual information, making it well-suited for our medical image analysis objectives. To enhance model performance and generalization, we have also discussed regularization techniques such as dropout, L1, and L2 regularization, which mitigate overfitting by introducing controlled noise and parameter constraints during training. Finally, we have introduced the concept of transfer learning, a powerful approach that leverages knowledge from models pre-trained on large datasets, enabling efficient adaptation to new tasks with limited data. By applying techniques like feature extraction and fine-tuning, we can capitalize on the rich representations learned from diverse domains, leading to improved performance and reduced training requirements. With this solid theoretical foundation, we are well-equipped to explore the practical implementation and evaluation of deep learning models for medical image grading and segmentation tasks, as detailed in the subsequent chapters of this thesis.

Chapter 3

Conception And Implementation Of The Diagnosis System

In the first parts of this chapter, we will explain in brief the task and process of image classification and then provide fundamental concepts as the background information on our approach to developing a model that can classify the fundus images of diabetes patients and grade the Diabetic Retinopathy in case of its presence. Next, we'll go over the common lesions found in the different stages of diabetic retinopathy. Following that, we will provide an explanation of the procedures involved in the training and building of these models. After that, we'll go into the details of data collection, the training of the models, interpreting the results, explaining the architecture of the best-performing model, and, along with that, showcase a sample of the model inference for the diagnosis.

3.1 History and Evolution of Image Classification Systems

Image classification, the task of assigning labels or categories to digital images based on their content, is a part of the computer vision field. This section goes into the brief history and evolution of image classification techniques, tracing the key milestones and developments that have shaped this field over the past decades. Image classification is one of the many techniques used in medical imaging and CAD systems, In the figure we can see it is used with other techniques which will expand later in Chapter 4.

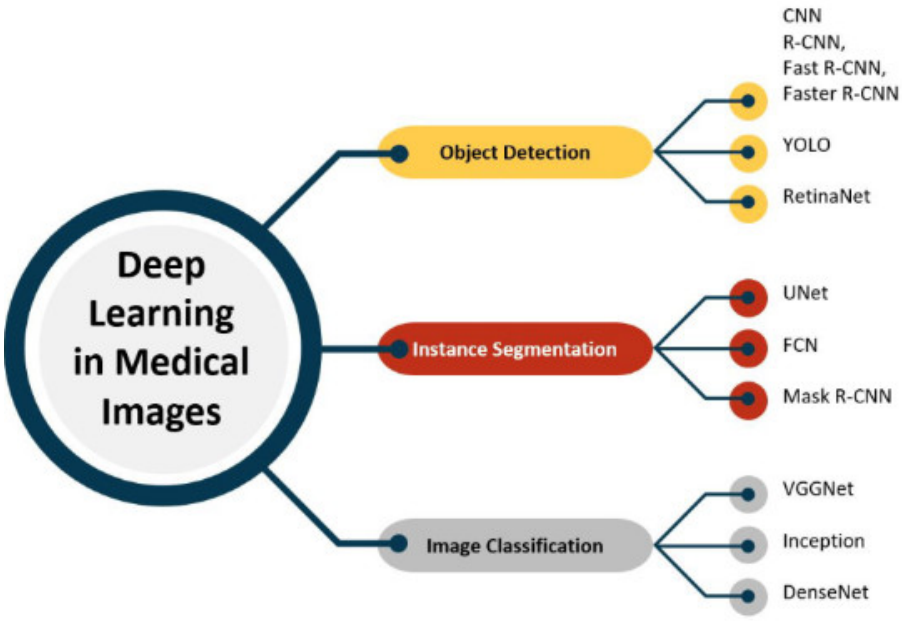


Figure 3.1: Most important techniques applied on medical imaging [79]

3.1.1 Early Beginnings: Pattern Recognition (1950s-1960s)

The roots of image classification can be traced back to the early days of artificial intelligence and pattern recognition research. In the 1950s and 1960s, the concept of pattern recognition emerged, laying the foundation for image classification tasks. Early work in this era focused on developing simple pattern recognition techniques, such as template matching and decision trees. These methods relied heavily on manually engineered features and rules, requiring significant domain expertise and human intervention.

3.1.2 Introduction of Artificial Neural Networks (1960s-1980s)

Researchers in the 1960s and 1980s explored the potential of artificial neural networks (ANNs) for pattern recognition tasks, including image classification. One of the earliest neural network models, the perceptron, was developed in the late 1950s by Frank Rosenblatt. Despite their promising start, the limitations of early neural networks, such as the inability to effectively train deep architectures, hindered their widespread adoption for image classification tasks during this time period.

3.1.3 Handcrafted Feature Extraction (1980s-2000s)

During the 1980s to 2000s, researchers focused on developing robust handcrafted features for image classification in the era of digital cameras and increasing image data availability. Key methods included SIFT for scale-invariant feature detection, HOG for capturing gradient orientations to describe shape and edge information, and LBP for encoding local texture patterns based on binary comparisons of pixel intensities. These methods computed specific descriptors that characterized images in terms of key points, gradient distributions, or texture patterns. These descriptors were crucial for tasks like object recognition and classification, often combined with traditional machine learning algorithms such as SVMs and random forests. They provided a mathematical foundation

to extract and utilize discriminative visual information from images effectively before the rise of deep learning techniques that now largely automate feature extraction and classification processes.

3.1.4 Rise of Deep Learning (2010s-present)

The resurgence of deep learning in the 2010s, facilitated by the availability of large datasets, increased computational power, and algorithmic advancements, revolutionized the field of image classification. Convolutional Neural Networks (CNNs), were specifically designed for image processing, became the dominant approach for image classification tasks.

Influential CNN architectures, such as AlexNet, VGGNet, GoogLeNet, and ResNet, achieved breakthrough performance on large-scale image classification benchmarks like ImageNet. The introduction of these powerful models, combined with techniques like transfer learning, where pre-trained models are fine-tuned on specific datasets, became a common practice to leverage the knowledge learned from large datasets.

3.1.5 Beyond Traditional Image Classification

As the field progressed, researchers started exploring more advanced tasks beyond traditional image classification, such as object detection, semantic segmentation, and instance segmentation. Techniques like region proposal networks, attention mechanisms, and transformer architectures were introduced to handle these more complex visual understanding tasks, they can have demonstrated reasonable efficacy in the interpretation of their decisions and focusing on the regions of the interest.

The field expanded to address challenges like domain adaptation, and interpretability of deep learning models for image classification. Researchers aimed to develop models that could effectively learn from limited data, adapt to new domains and types of dataset, and provide insights into their decision-making processes, which is always having a problem called the Black Box in deep neural networks.

3.1.6 Ongoing Developments and Future Directions

The field of image classification continues to evolve rapidly, with researchers pushing the boundaries by exploring novel architectures, efficient training techniques, and the incorporation of multimodal data (e.g., text and images). Moreover, the integration of image classification with other AI disciplines, such as natural language processing and reasoning, is enabling the development of more advanced and comprehensive visual understanding systems. These systems aim to not only classify images accurately but also to provide insights, explanations, and context-aware predictions.

As computational resources and data availability continue to grow, image classification will likely play a crucial role in various applications, including computer vision, robotics, healthcare, and beyond. The ongoing research efforts in this field are driven by the pursuit of more accurate, efficient, and interpretable image classification models that can adapt to the ever-increasing complexity and diversity of visual data.

3.2 Lesions And Medical Context

The lesions are the features that our model will aim to detect as the pattern and classify the stage and severity of Diabetic Retinopathy based on, though we will not be crafting the features on our own, thanks to the CNN architecture and deep learning the using the artificial neural networks algorithm will learn the pattern and features on its own.

Microaneurysms

Microaneurysms are most frequently the first indicators identified in the fundus examination of individuals diagnosed with Diabetic Retinopathy. These structures show a remarkable visual aspect, indicated by their diminutive size, vivid red color, and precise circular shape. When vascular edema develops and pressure builds in the vascular network, blood vessels may rupture, leading to the development of microaneurysms. This medication has the potential to reduce the appearance of these symptoms, especially during the initial phases of Diabetic Retinopathy, as well as in individuals with hypertension. (Figure 3.2) shows the presence of microaneurysms in the retina.

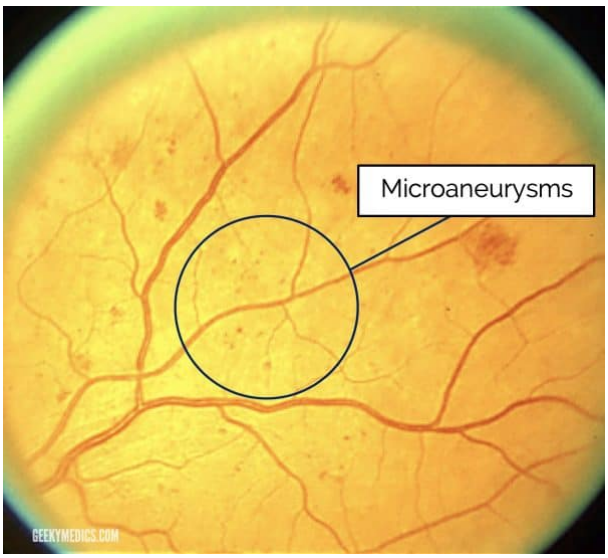


Figure 3.2: Image indicating the presence of microaneurysm [80]

Soft exudates

Soft exudates, sometimes referred to as cotton wool spots, are characterized by their perceptible appearance on the retinal surface. The typical visual characteristics include a cloud-like appearance, a greyish-white or yellowish white color. These substances induce inflammation in the retina and may not result in severe vision impairment in diabetic individuals, but only if they are located outside the fovea, which is the center region of the eyes responsible for sharp and focused vision. Swelling occurs as a result of ischemia, which is caused by a decrease in axonal transport inside the blood vessels. The presence of soft exudates in the retina is seen in (Figure 3.3).

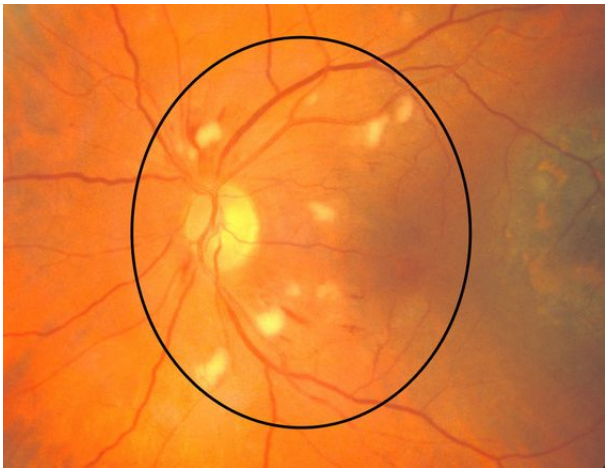


Figure 3.3: Image of soft exudates [81]

Hard exudates

The growth of new capillaries leads to suffocation in the blood vessels, causing a shortage of oxygen. This suffocation causes pressure and breaking in the blood vessels, resulting in the production of white or yellow spots by the deposition of proteins and lipids into the retina. These spots have sharper borders compared to soft exudates. In the retina, they may be organized in the form of patches, dots. Depositions of these substances may also be seen in the vicinity of the retinal veins. . (Figure 3.4) illustrates the presence of hard exudates.

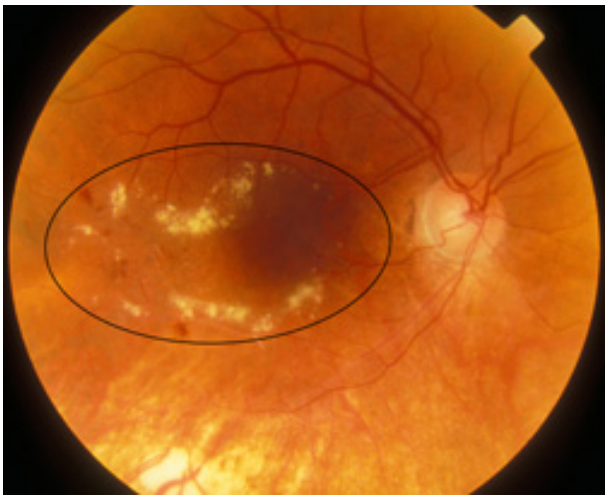


Figure 3.4: Image of Hard Exudates [82]

Hemorrhages

The accumulation of pressure within the blood vessel network, caused by oxygen deprivation and elevated glucose levels in the bloodstream, leads to the development of new, fragile capillaries. This occurs because the existing blood vessels are unable to transmit signals to the brain, prompting the brain to respond by forming new vessels that are of lesser strength. Due to their fragility, these capillaries leak into the retina, leading to the formation of flame-shaped hemorrhage or dot or blot-shaped hemorrhages.

Dots are often linked to Diabetic Retinopathy, whereas flames occur as a result of the rupture of retinal nerve fibers. The items in the retina have irregular shapes or clear edges. (Figure 3.5) shows the presence of severe hemorrhages inside the retinal fundus image.

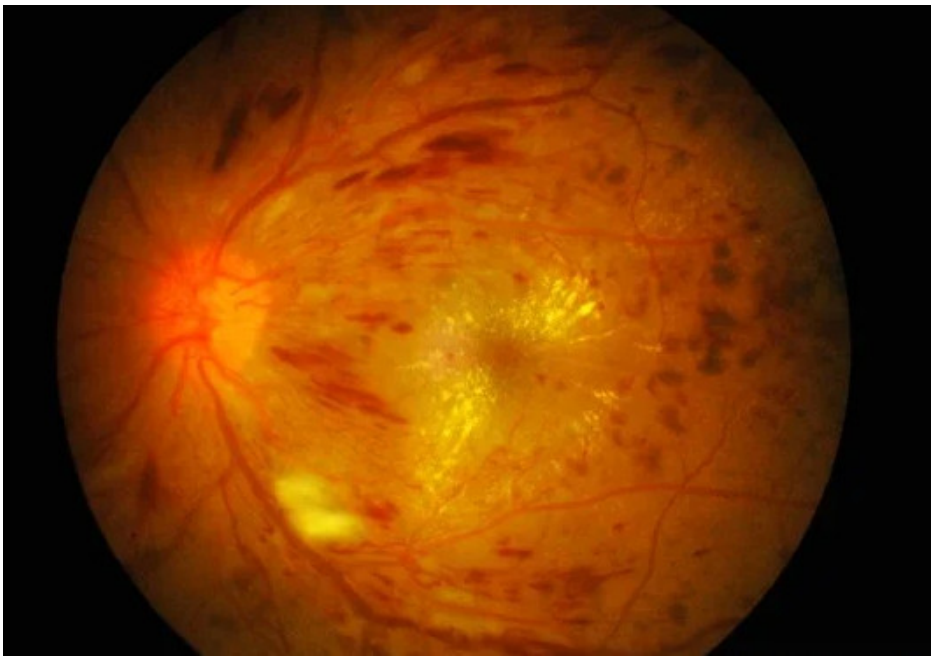


Figure 3.5: Image of fundus image indicating the presence of hemorrhaging in the eye across different places [83]

Neovascularization:

Neovascularization lesions come at the latter, proliferative stages, which is defined by the abnormal proliferation of new blood vessels in response to ischemia circumstances inside the retina. These little abnormalities indicate a crucial phase in conditions including Diabetic Retinopathy and retinal vein blockage. Neovascularization lesions often present as vasculature that exhibits fragility and infiltration into the retinal tissue, frequently extending into the vitreous humor. In contrast to typical blood vessels, these vessels have a deficiency in structural development, making them susceptible to leakage and bleeding. Furthermore, their atypical development patterns may result in problems such as the proliferation of fibrovascular tissue, bleeding in the vitreous, and detachment of the retina due to traction. The identification of neovascularization lesions highlights the need for prompt therapeutic care in order to minimize the potential for retinal complications.

In short, the identification and treatment of neovascularization lesions in retinal disease are crucial for preserving visual acuity and preventing irreversible damage to the eye. And the detection at early stages could minimise the risk of it progressing to this stage.

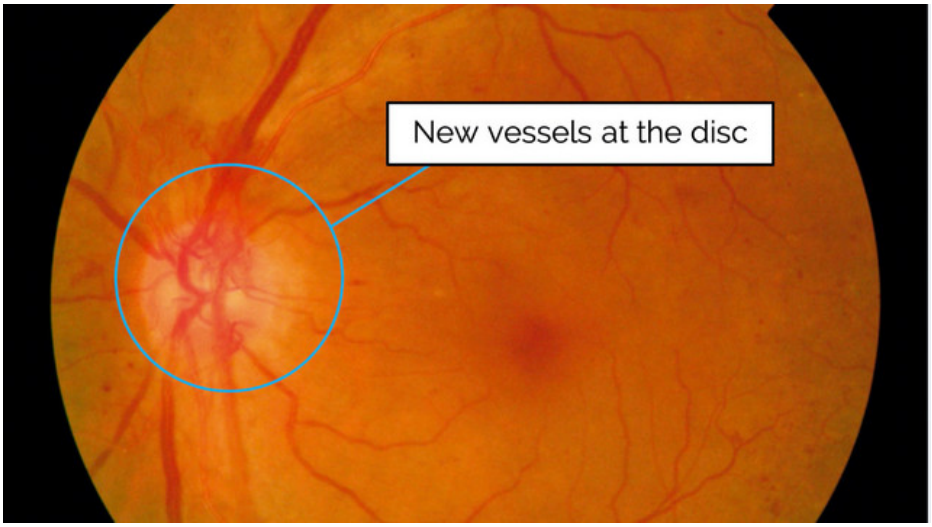


Figure 3.6: Image showing the growth of abnormal new vessels near the optic disc [84]

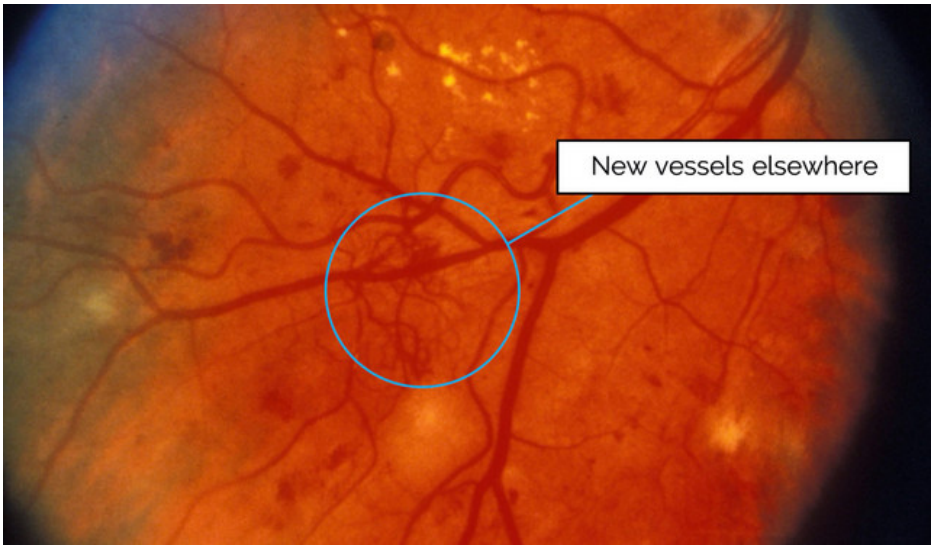


Figure 3.7: Image of growth of new vessels in the middle of the eye [84]

3.3 Data Collection

The first phase in the development of AI systems for DR screening from retinal photographs is to establish a sufficiently large dataset of photographs of patients with DM, with each photograph marked for the severity of DR and the presence of DME.In this work, the retinal fundus images of the Asia Pacific Tele - Ophthalmology Society 2019 Blindness Detection (APTOS 2019 BD) dataset were used. APTOS 2019 BD is an open Diabetic Retinopathy research dataset. The associated dataset contains 3,662 retinal images collected from numerous subjects living in the rural areas of India. The images were put together by Aravind Eye Hospital, India, with the purpose of building an ML model to detect blindness autonomously without manual medical screening. These Red-Green-Blue (RGB) images were captured using Fundus photography. The samples were then labeled by trained doctors who categorized the extent of blindness of a person into five different levels – no DR, mild DR, moderate DR, severe DR, and proliferative DR. The

images were captured in various conditions and environments over an extended period.

The APTOS (Asia Pacific Tele-Ophthalmology Society) 2019 Diabetic Retinopathy dataset focuses on the classification of Diabetic Retinopathy (DR), encompassing five distinct severity levels manually annotated by medical specialists, mostly ophthalmologists. These levels span from 0 indicating the absence of DR to 4 indicating Proliferative DR. The dataset categories DR severity as follows: 0 for No DR, 1 for Mild DR, 2 for Moderate DR, 3 for Severe DR, and 4 for Proliferative DR. Composed predominantly of retinal images acquired through fundus photography across diverse imaging conditions, each image has undergone assessment by clinicians to determine the severity of DR on the aforementioned scale. In total, the dataset consists of 5900 retinal images, split into 3662 training images and 1928 testing images, The test phase data has no specific labels provided. The images were gathered from multiple clinics. The images were taken with diverse fundus cameras and under different conditions.

Class	**Label Count**	**Percentage**
No DR	1,805	50.01%
Mild DR	999	27.6%
Moderate DR	370	10.25%
Severe DR	295	8.17%
Proliferative DR	139	3.85%
Total	**3,608**	**100%**

Table 3.1: Distribution of Classes in APTOS dataset

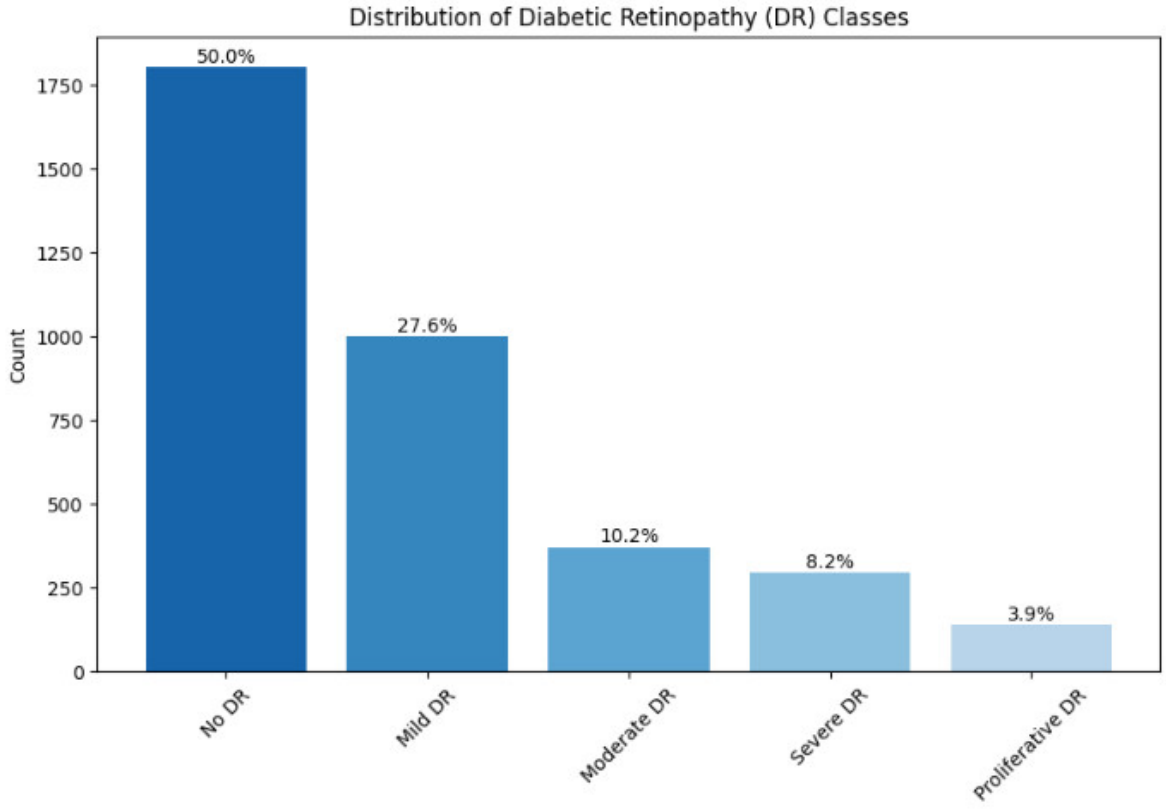


Figure 3.8: Distribution of APTOS 2019 Dataset

The remaining part of the data set will be collected through the EyePACS platform, which stands for "An Adaptable Telemedicine System for Diabetic Retinopathy Screening." This Web-based DRS solution is licence-free and intended to streamline the image collection, transmission, and review process. [85] for EYEPACS,The system offers a flexible system for medical professionals to collaborate on Diabetic Retinopathy.

The EyePACS program provides training and certification to primary clinic professionals, such as nurses, technicians, and administrative staff, on how to get retinal pictures using common digital retinal cameras. Following certification via the EyePACS Retinopathy Grading System, images are analyzed by either local ophthalmologists or professional EyePACS reviewers. Retinopathy level reports together with suggestions for referrals are sent back to primary care physicians via the EyePACS website or interfaces between EyePACS and electronic health records or chronic illness registries that comply with Health Level 7.In California, the EyePACS DRS program documented 3562 contacts between 2005 and 2006. Between 2006 and 2009, EyePACS was extended to more than 120 primary care locations in California and other states. Referral rates overall are 7.83% for various disorders (such as glaucoma and cataracts) and 8.21% for sight-threatening retinopathy. The 35,126 total photos are categorized and labeled in the same manner as the APTOS 2019 (Asia Pacific Tele-Ophthalmology Society) first data set. according to the ETDRS categorization criteria. we weren't to take all of it so we opted for the whole first APTOS 2019 dataset and a fraction of EYEPACS dataset. and took 1600 images for testing, which is a custom dataset.

Class	**Label Count**	**Percentage (%)**
No DR	3300	29.35%
Mild DR	3247	28.82%
Moderate DR	3000	26.68%
Severe DR	1008	8.96%
Proliferative DR	687	6.11%
Total	**11242**	**100.00%**

Table 3.2: Classes distribution in the final data set

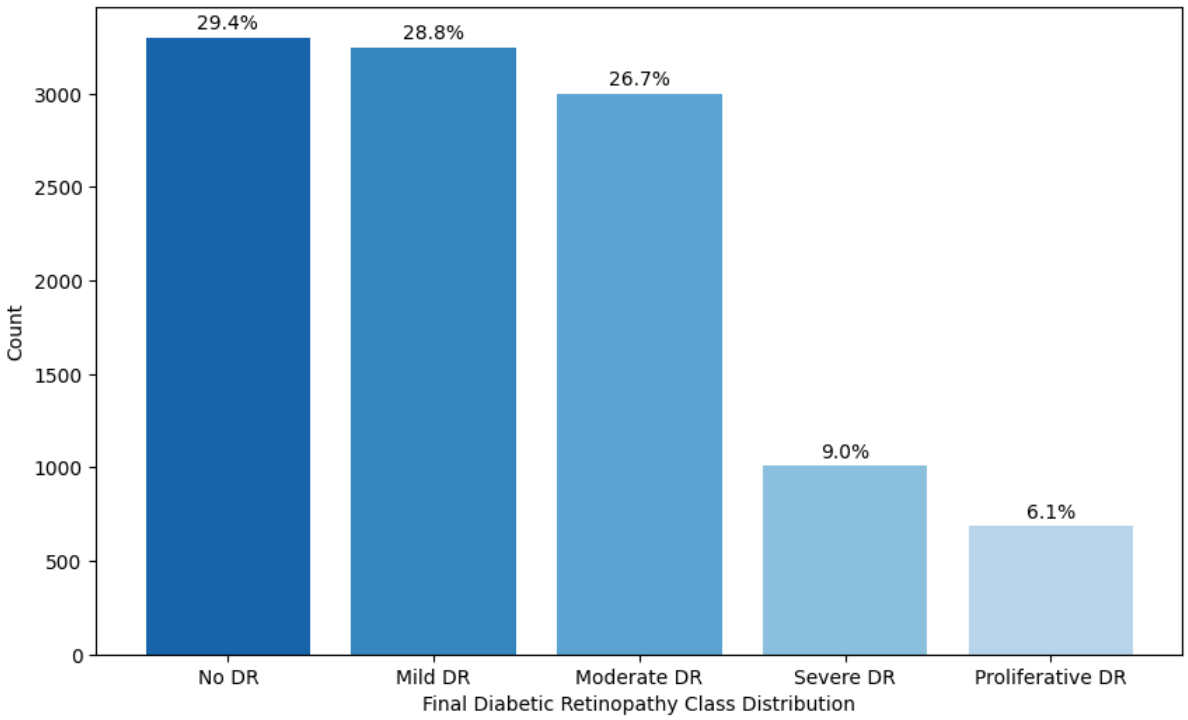


Figure 3.9: Distribution of the final dataset

3.4 Data Preprocessing:

Since in the images there will be a lot of exposure and lighting variation over the original fundus images, we took several preprocessing steps to try to standardize the image conditions and enhance contrast. All of the images taken as inputs will be resized in accordance to the pretrained model they're fed into for features extraction The pretrained models are designed with fixed input layers, and feeding images of different sizes can result in dimensional mismatches, causing the model to not extract the features at its fully potential, The convolutional layers in these models have learned specific features (edges, textures, patterns etc…) from images of a certain size during their training on large datasets like ImageNet. By resizing the images to the expected input size, we can ensure that these learned features can be effectively utilized for our specific task, leveraging the advantages of transfer learning, The image dimensions are 244x244 for 3 models and 299x299 for one.

Secondly, we explicitly converted the data type of the pixel arrays to float. This conversion is crucial for several reasons. Floating-point numbers provide a more precise representation of the scaled pixel values compared to integers, which is particularly important when dealing with the subtle variations in pixel intensity that can significantly impact model performance. Additionally, most deep learning frameworks and libraries expect input data to be in float format, as it allows for efficient computation and integration with various optimization algorithms used during model training.

We also divided all pixel values by 255. This operation scales down the range of pixel intensities from the standard 8-bit representation (0-255) to a floating-point range between 0 and 1. This scaling is essential because it helps the model converge faster during training by ensuring that the input features have a consistent scale.

we will also be applying CLAHE (Contrast Limited Adaptive Histogram Equalization) which changes the contrast of the image and points out details that aren't clearly visible from the default image [86], which eases the identification of small details both to the naked eye and CNN models. It mostly operates on the small regions of the image, and can enhance the models performance. It can reduce the blur, low contrast, and uneven lighting problems commonly found in the in fundus images dataset.



Figure 3.10: Before and after the application of CLAHE Contrast enhancement

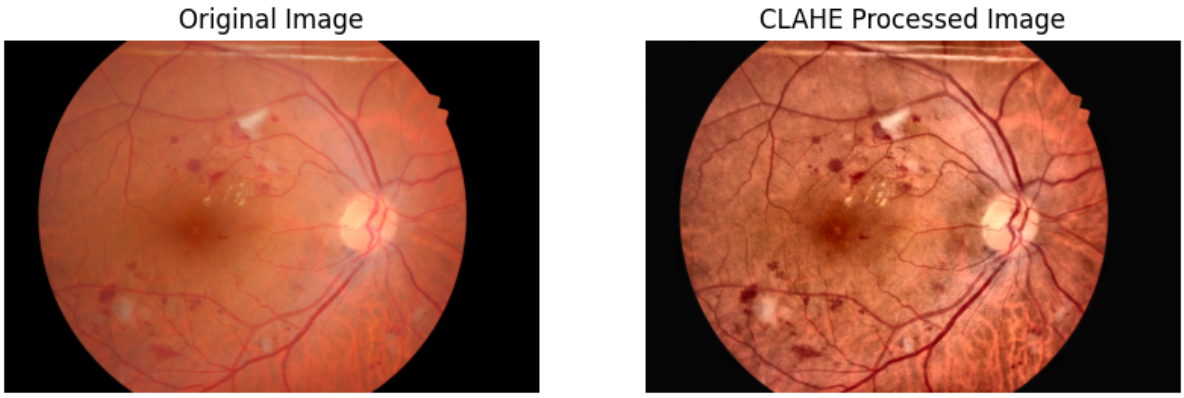


Figure 3.11: Before and after the application of CLAHE Contrast enhancement

In the context of Contrast Limited Adaptive Histogram Equalization (CLAHE), the process begins by dividing the image into smaller segments known as tiles or blocks. It's important to note that each tile operates independently of the others. Following this division, the histogram equalization procedure is applied to each individual tile. This process involves adjusting the pixel values within each tile to enhance the contrast. However, if not carefully managed, this adjustment can lead to over-amplification of the image, particularly in areas with low contrast or high noise levels. This very often a standard approach and have been demonstrated to augment the performance of nearly all CNN networks trained to detect the presence of lesions in Diabetic Retinopathy, in many stud-

ies [87].

To mitigate this issue, the "clip.limit" parameter plays a crucial role. By setting an appropriate clip limit, it is possible to prevent excessive amplification in regions of the image that are either very dark or very bright, thereby reducing the risk of creating an unnatural appearance due to extreme intensity variations. The clip limit essentially ensures that the redistribution of pixel intensities during the equalization process does not result in overly dramatic changes, thus maintaining a more balanced and visually appealing outcome.

Our experiments have shown that incorporating the CLAHE technique for contrast enhancement significantly improves the performance of our model. To validate these findings, we utilized the APTOS 2019 Dataset, allocating 20% of the data for validation purposes.

Preprocessing	**Accuracy (%)**	**F1 Score (%)**
Without CLAHE	85.62	84.05
With CLAHE	86.07	85.67

Table 3.3: Comparison of Accuracy and F1 Score with and without CLAHE contrast enhancement

Just like the literature and in many works from the literature, contrast enhancement, allowing the model to capture more details by making patterns more apparent that weren't statistically distinguished from the first place.

3.5 Data Augmentation

Deep learning models typically require large amounts of labeled data to achieve high performance. However, acquiring a sufficient number of labelled fundus images for training can be challenging, as medical image datasets are often limited in size like mentioned in the first chapter, and computation resources can be limited. Data augmentation techniques can be used to artificially generate new training samples from the existing data. This effectively increases the size and diversity of the training dataset, which can lead to improved model generalization and robustness.

Also, overfitting occurs when the model performs well on the training data but fails to generalize to new, unseen data. Data augmentation can help us mitigate this issue by introducing more variations into the training data, forcing the model to learn more generalizable features that are invariant to these transformations [88].

By exposing the model to a wider range of variations in the training data, positions ,angels data augmentation can help the model learn more robust and generalizable features. This can be particularly beneficial for the classification of Diabetic Retinopathy, where the appearance of lesions and other pathological features may vary across different patients, imaging conditions, and fundus camera settings.

In a multi-class classification task like the Diabetic Retinopathy at hand, the distribution of samples across different classes (severity levels) may be imbalanced, some classes can have a distribution as low as 2 and 3 percent in the dataset, therefore the parameters and weights of the model will be much more favored into the majority classes. Data augmentation can be used to selectively and randomly generate more samples giving

more probabilistic and augmented chance to be more often seen during training, helping to achieve a more balanced class representation during training. This can improve the model's ability to learn discriminate features for all classes, leading to better overall classification performance.

Data augmentation methods are numerous and come in various techniques to manipulate the image by altering the pixels [89], nevertheless, geometric transformation is the most well-liked strategy [90]. It has therefore been applied to the chosen datasets in this work to address the class imbalance challenge. Different transformations have been used, such as Vertical Flipping, Horizontal flipping, random zooming, and random shearing, and random cropping within a limited range, and we will use the filling mode to nearest in the instances the image shift its out of its original borders.

- **Vertical Rotation**: We applied vertical rotations to the images.
- **Horizontal Rotation**: Horizontal rotations were performed on the images.
- **Zooming**: The images were randomly zoomed, with magnification levels adjusted between 115% and 130%.
- **Shearing**: A shearing transformation was applied to the images, with angles ranging from 30° to 60°.
- **Cropping**: Random cropping was executed on the images, reducing their sizes to between 80% and 90% of their original dimensions.
- **Height and Width Shift**: We used a height and width shift adjustment between 10% and 25%.
- **Pixel Translation**: Pixels within the images were randomly moved around the canvas, changing the positions of the content without modifying the overall image size.



Figure 3.12: Application of a shearing on the image

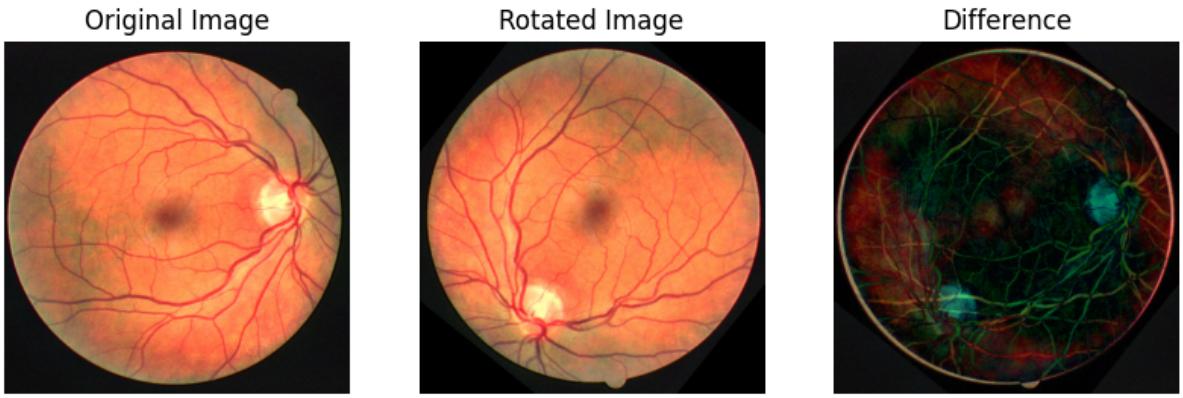


Figure 3.13: Application of a random rotation on the image

Part of the reason Data augmentation proves particularly advantageous for images used in Diabetic Retinopathy and general eye disease detection due to the dynamic nature and varied locations of lesions. Lesions such as microaneurysms, hemorrhages, and exudates containing lipids and proteins can appear at different XY coordinates across the eye. This variability underscores the importance of augmenting data to encompass diverse positional scenarios, enhancing the robustness and effectiveness of detection models, since on the validation and unseen data. we can almost never know where the lesions are located.

3.6 Implementation And Conception

3.6.1 Development Environment

To implement our solution and deploy it, we need a specific set of hardware and software tools. In this section, we will present the different hardware and software specifications required to accomplish the implementation task.

Computing power is a crucial factor for the acceleration as well as for the precision of the models. This requires the use of parallel architectures (CUDA, OpenCL, ...) and GPUs (NVIDIA, AMD, ...). This is why we have opted for the Cloud Platform Of kaggle

- Kaggle

his is a free and open-source platform under Google LLC, designed for developers of data science applications in general and machine learning applications in particular. Kaggle offers the ability to write and execute Python code without any required configuration, as well as easy-to-use implementation management.

3.6.2 Software Specifications

- Python

Python is an interpreted, multi-paradigm, and cross-platform programming language designed for various purposes and domains, including data science. It is open-source and offers simple and easy-to-use syntaxes as well as high-level tools to optimise development productivity.

- TensorFlow/Keras

- TensorFlow is a widely used tool in AI and machine learning, developed by Google, and provides a stable API in Python and other languages. - Keras is an open-source library written in Python. It allows interaction with deep neural network algorithms and machine learning, particularly TensorFlow.

Other Libraries Used

- keras-metrics:

This package provides additional metrics for evaluating classification models in Keras.

- tensorflow-addons:

This is a repository of contributions that implement new features not available in the core TensorFlow library.

- numpy:

A fundamental scientific computing library for Python, used for linear algebra, random number generation, and more.

- pandas:

A data manipulation and analysis tool for Python, useful for tasks like reading files and creating DataFrames.

- sklearn:

A machine learning library for Python, used for data exploration, analysis, and implementing various algorithms.

3.7 Evaluation Metrics

Accuracy, Precision, Recall, and F1-Score, Cohen Kappa are common statistical measures, metrics commonly used as a benchmark evaluation metrics for multi- classification models.

Here below are their definitions and mathematical formulas:

3.7.1 Accuracy

The proportion of correctly predicted instances out of the total number of instances. This one is straightforward and is a very intuitive indicator of the model's performance. Accuracy gives a straightforward measure of how often the classifier is correct. It's the ratio of correctly predicted instances to the total instances

$$Accuracy = \frac{TP + TN}{TotalPredictions} \tag{3.1}$$

3.7.2 Precision

The proportion of true positive predictions out of the total positive predictions. When it comes to this one, the metric will compute the average precision for each class we have 5.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

For example, suppose we consider calculating the precision of Class number 2 which is moderate Diabetic Retinopathy, the divides the accurate predictions of class 2 by the false classifications of when the model returns Class 2.

3.7.3 Recall (also known as Sensitivity or True Positive Rate In Binary Classifications)

The Recall metric for a particular class measures the ability of a classifier to correctly identify instances of that class among all true instances of that class in the dataset, For example given we give the system 100 images of stage 3 Diabetic Retinopathy, how likely out of 100 images is it going to return stage 3.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

3.7.4 F1 Score

The F1 Score, representing the harmonic mean of Precision and Recall, offers a balanced assessment of a model's performance by equally weighting the model's ability to correctly identify relevant instances (Precision) and its capacity to retrieve all relevant instances (Recall).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.4}$$

3.7.5 Cohen Kappa Score

The Cohen Kappa Score measures the agreement between two raters (or classifiers) who each classify N items into C mutually exclusive categories. It adjusts for the possibility of the agreement occurring by chance. A higher Kappa score indicates a higher degree of agreement beyond what would be expected by chance alone. The Cohen Kappa Score is particularly valuable because it quantifies the level of agreement achieved by a model beyond what could be expected by mere chance. Unlike metrics such as accuracy, which can sometimes reflect favorable outcomes due to random guessing rather than genuine predictive power, the Cohen Kappa Score accounts for the base rate of agreement in the population [91]. This makes it a more reliable measure of a model's predictive accuracy,

especially in scenarios where the distribution of classes is uneven or where there is a high risk of chance agreements. Essentially, a higher Cohen Kappa Score indicates that the model's predictions are probabilistically more accurate, demonstrating a strong ability to distinguish between classes rather than merely benefiting from the imbalance in the dataset or the simplicity of the classification task.

$$Cohen'sKappa = \frac{p_o - p_e}{1 - p_e} \tag{3.5}$$

Where: - p_o represents the observed agreement between the model predictions and the actual class labels.
- p_e is the expected agreement purely by chance, which represents the proportion of times the models predict the same class by random chance.

The Kappa score is especially useful in situations where the classes are imbalanced, providing a more informative measure than simple accuracy.

3.8 Model Training And Discussion

In our project, we'll be focusing on a specific dataset and task and training data to understand how different pre-trained architectures, such as VGG19 [92], ResNet101 [93], InceptionV3 (GoogleNet) [94], DenseNet169 [95], perform in extracting features that are most useful for our purposes. We'll be experimenting with these architectures to see which ones can extract the most relevant features. The (Figure 3.14) displays a simplified flowchart of the training process.

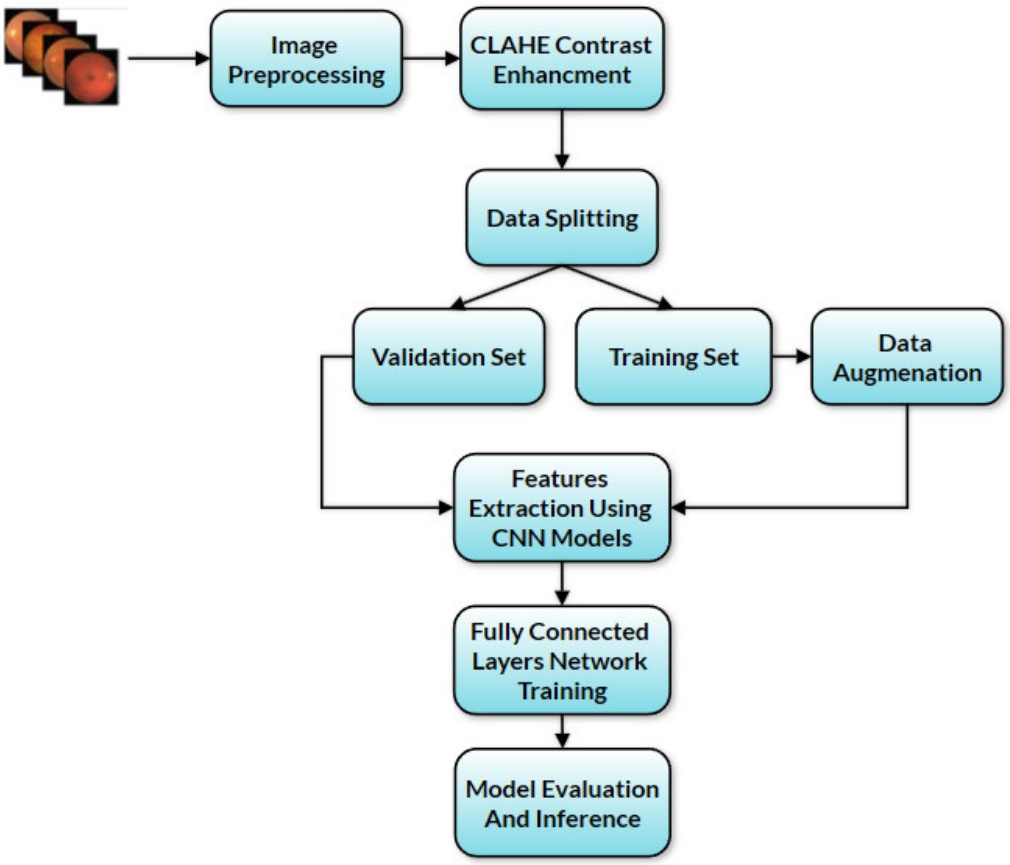


Figure 3.14: Flowchart diagram of the conception of the model

Once we identify the models that perform the best, we'll use the one with the highest performance as the feature extractor in a multi-layer neural network that we'll take as our most performing model. The (Table 3.4) shows the parameter size of each network and their ImageNet benchmark accuracies

Model	**Parameters**	**Top-1 Accuracy (%)**	**Top-5 Accuracy (%)**
ResNet101	44,549,152	77.374	93.546
VGG19	138,357,094	71.2	89.9
DenseNet169	14,699,840	76.5	93.7
Inception V3	23,851,784	77.5	93.6

Table 3.4: Comparison of Model Total Parameters and Accuracy In the ImageNet Benchmark

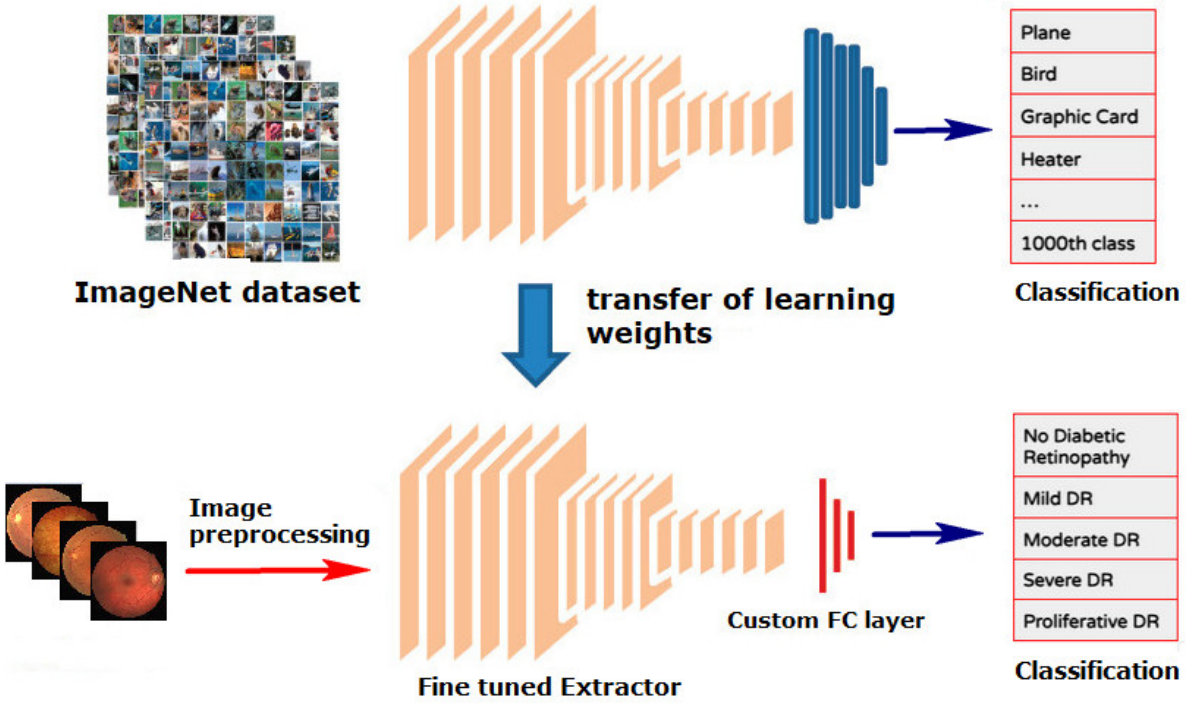


Figure 3.15: Simplified illustration of transfer learning process

In this part, we outline the optimization strategy for our classification neural network designed for the prediction of Diabetic Retinopathy stage, focusing on the transition from initial setup through fine-tuning and monitoring. The optimization process employs the Adam optimizer with default parameters to enhance model performance and update the parameters of the network in accordance of how it is performing on the validation dataset. Initially, the model undergoes 2 warm up epochs during which the base model layers are frozen to stabilize training. A high initial learning rate of 0.0005 is applied during these warmup epochs to expedite convergence and prepare the model for more nuanced adjustments.

After the warm up phase, the final layers of the pretrained model are unfrozen, allowing for fine-tuning of parameters to better align with the specifics of the dataset. Many researchers in several studies have proposed that the lower layers [96] at the bottom mostly see generic main characteristics such as edges and circles, while the higher layers at the final levels perceive more particular elements related to the dataset. Due to this rationale, the researchers advocate just adjusting the uppermost layers.

The learning rate is reduced during the fine-tuning phase to 0.0001 to facilitate precise adjustments to model weights. Additionally, a dynamic learning rate scheduling strategy is implemented where the learning rate is decreased by a factor of 0.5 if the validation losses does not improve, ensuring optimal training efficiency down to a minimum of 0.00001, the model always stops at its best weights in the sequence of training and resumes training with a lower learning rate.

The reason we should be careful with the learning rate it controls the adjustment

of weights as they're being optimized after the gradient calculating to minimize the loss function, considering earlier the formula of gradient adjustment. Meaning at some point a learning rate can bring down the loss function into a local minimum, and it stays stuck there unable to go out, so the model should restore it's best weights at that specific learning rate and then resume training with a lower learning rate that carefully adjusts the weights, the (Figure 3.16) and (3.17) below gives an idea of a loss function moving direction in a 3D and 2D Plane respectively.

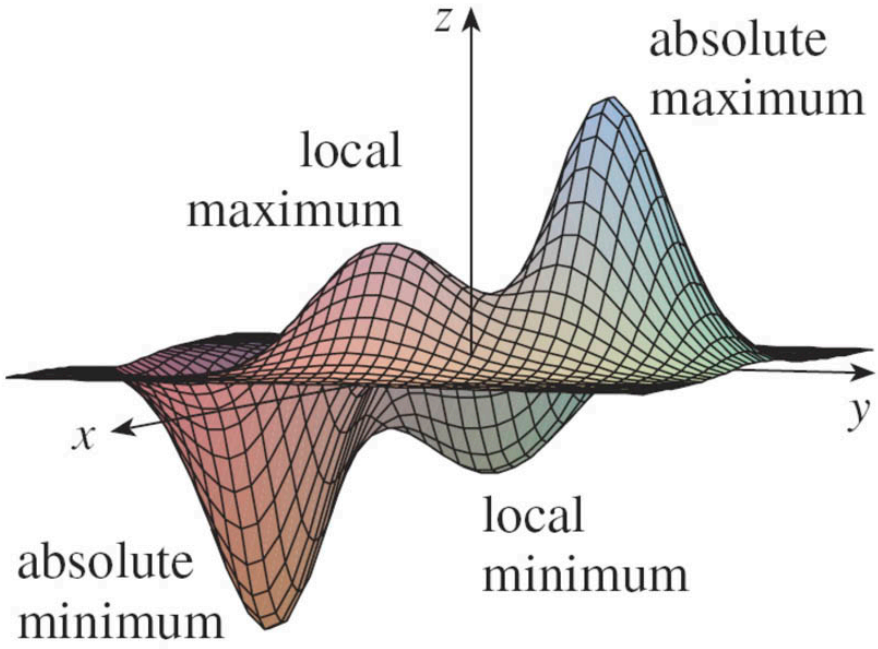


Figure 3.16: Graph showing an example of local and global minima in function with three variables

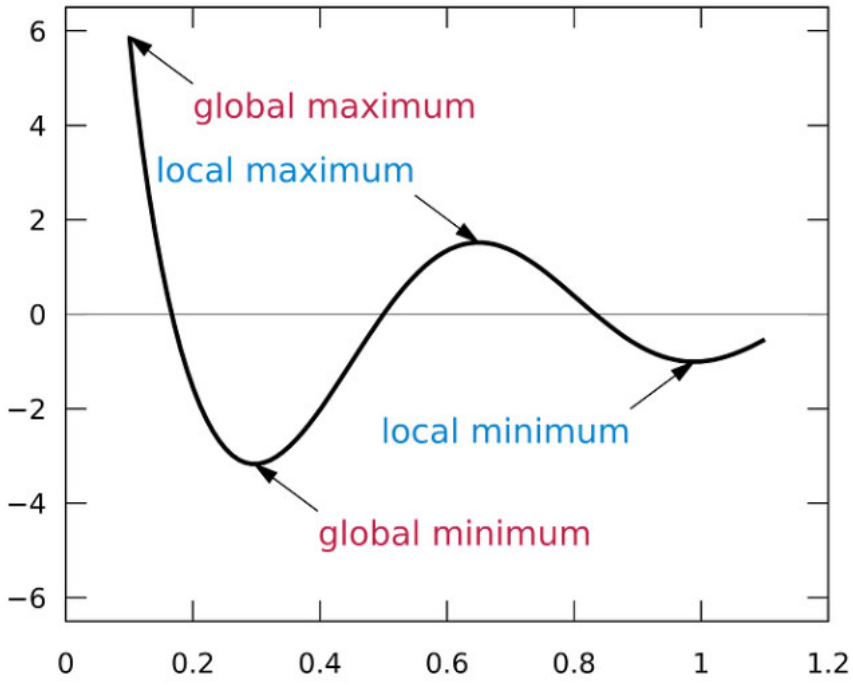


Figure 3.17: Graph showing an example of local and global minima in function with two variables

We took validation loss for monitoring during training, chosen over accuracy for its sensitivity to model certainty in predictions, is monitored throughout training. This metric provides a clearer indication of how well the model distinguishes between classes at the probabilistic level, crucial for medical image classification tasks. The choice of a batch size of 16 is informed by statistical insights from relevant literature, balancing computational efficiency with model stability.

The architecture of the neural network includes dense layers with 2048 neurons each across 2 dense layers. Flatten is employed after the convolutional layers, enhancing feature extraction by aggregating spatial information globally. Regularization techniques such as dropout with a rate of 0.4 and L2 regularization with a parameter of 0.0001 are applied to mitigate overfitting and enhance generalization capability. The loss function employed is categorical cross-entropy, suitable for multi-class classification tasks like grading the stage of Diabetic Retinopathy.

The formula for Categorical Cross-Entropy is given by:

$$\text{Categorical Cross-Entropy} = -\sum_{i} y_{\text{true},i} \log(y_{\text{pred},i}) \tag{3.6}$$

Where:

- $y_{\text{true},i}$ is the true probability distribution of the i-th class (often one-hot encoded).
- $y_{\text{pred},i}$ is the predicted probability of the i-th class output by the model.
- log denotes the natural logarithm function.

Given the variability in convergence times and performance thresholds, training is equipped with early stopping criteria based on validation loss metrics to prevent overfitting and ensure optimal model generalization. This comprehensive approach to neural network optimization aims to leverage both theoretical insights and empirical data to achieve a pertinent performance in the diagnosis of Diabetic Retinopathy and grading its stage.

Parameter	**Value**
Dense Layer Neurons	2048
Dense Layers Count	2
Warmup Epochs	2
Warmup Learning Rate	0.0005
Tuning Learning Rate	0.0001
Minimum Learning Rate	10^{-6}
Learning Rate Drop Value	0.5
Monitoring	Validation Loss
Loss Function	Categorical Cross Entropy
Batch Size	16
Dropout Rate	0.4
L2 Regularization Parameter	0.0001
Total Epochs	Early stopping

Table 3.5: Training Parameters

It's important to note that these models have significant architectural differences, such as weights, dimensions, numbers of filters, their dimensions, and how they prioritize computational efficiency versus feature richness. For example, DenseNet focuses on reusing features from layers at different levels, while GoogleNet's inception modules prioritise computational efficiency with parallel convolutional pathways, which can affect the parameters and complexity of the features extracted compared to models with more sequential architectures.

Regularization techniques during training, like batch normalization, ,and weight decay, can vary between models and significantly influence the robustness and generalization of the feature representations. Architectural choices, such as skip connections in ResNet and DenseNet, which help mitigate the gradient vanishing, play a crucial role in how well features are learned and transferred into the deeper layers of the [97]. Additionally, different initialization strategies for weights and biases in pre-trained models can lead to varied starting points in the feature space, which can affect the convergence speed and the final quality of the features.

The architectural choices and hyperparameters in Convolutional Neural Networks (CNNs) play a pivotal role in feature extraction, significantly impacting the performance and capabilities of models like VGG19, GoogLeNet, ResNets, and DenseNets. This discussion below will provide a simple overview into how various architectural design decisions influence feature extraction in these models.

Max Pooling vs. Average Pooling

Pooling methods are crucial for Downsampling feature maps and introducing translation invariance. Max pooling selects the maximum value within each pooling window, emphasizing the most prominent features. This approach is advantageous for capturing high-frequency details and enhancing robustness to small distortions. However, it can also lead to the loss of important contextual information, as less dominant features are discarded.

In contrast, average pooling computes the mean value of the features within each pooling window, preserving a broader spectrum of features, including low-frequency components and contextual information. This method can smooth out the feature maps, potentially diluting salient features but offering a more holistic representation. The choice between max pooling and average pooling can thus influence the balance between capturing fine details and preserving contextual information.

Batch Normalization

Batch normalization (BN) is a technique used to stabilize and accelerate training by normalizing the inputs of each layer. The degree of batch normalization, controlled by hyperparameters like momentum and epsilon, can significantly impact the learned features. Higher momentum values and lower epsilon values can lead to more stable and consistent feature representations by reducing internal covariate shift.

However, excessive normalization can constrain the model's capacity to learn complex, non-linear patterns, potentially limiting its expressiveness. Balancing BN involves tuning these hyperparameters to ensure effective gradient flow and feature learning without overly constraining the network's flexibility.

Network Depth and Complexity

The depth of a CNN architecture, defined by the number of layers, directly influences the complexity and abstraction level of the extracted features. Shallow networks capture basic, low-level features like edges and textures, while deeper networks, such as ResNet and DenseNet, progressively build upon these initial features to capture more abstract and high-level representations.

Deeper architectures enable the learning of hierarchical feature representations, crucial for complex tasks. However, they are also prone to the vanishing gradient problem, where gradients diminish as they propagate back through the layers, hindering effective feature learning in the deepest layers. Techniques like skip connections (ResNet) and dense connections (DenseNet) are employed to mitigate this issue, facilitating better gradient flow and feature reuse.

Skip and Dense Connections

Skip connections in ResNet introduce identity mappings that bypass one or more layers, allowing gradients to flow directly through the network, alleviating the vanishing gradient problem. This enables deeper networks to be trained effectively, fostering the learning of richer feature representations. Skip connections also allow for the combination of low-level and high-level features, enhancing the model's capacity to capture diverse patterns.

Dense connections in DenseNet further extend this concept by connecting each layer to every other layer in a feed-forward fashion. This ensures maximum information and gradient flow between layers, promoting feature reuse and reducing the number of param-

eters compared to traditional architectures. Dense connections facilitate the learning of complex, highly discriminative feature representations by leveraging multi-scale feature fusion.

In conclusion, the interplay of pooling methods, batch normalization, network depth, and architectural innovations profoundly influences feature extraction in CNNs. A deep understanding of these elements and their interdependencies is crucial for designing effective CNN This part provided a more technical and detailed exploration of the various factors that influence feature extraction in CNNs.

3.8.1 Performance of The Different Models in The Test Dataset

At first in these tables below such as (Table 3.6) and (Table 3.7), the total classification metrics and then the specific class metrics are displayed, for all the CNN models how they managed to extract the features, after that we'll discuss and interpret in details the results of the metrics, and the differences and what is the clinical importance of what is achieved here.

Metric	**VGG19**	**DenseNet169**	**InceptionV3**	**ResNet101**
Accuracy (%)	85.12	**89.75**	86.38	87.84
F1 Score (%)	85.09	**89.70**	86.11	87.22
Precision (%)	85.61	**90.25**	87.00	87.66
Recall (%)	85.12	**89.75**	86.38	87.84
Cohen's Kappa (%)	88.33	**93.09**	89.91	91.78
Validation Loss	0.312	**0.239**	0.286	0.247

Table 3.6: Performance Metrics of The Different CNN Models after Fine Tuning

Class	**Precision** (%)	**Recall** (%)	**F1-Score** (%)
No DR	98.82	97.33	98.07
Stage 1	82.18	90.67	86.21
Stage 2	81.86	90.25	85.85
Stage 3	94.12	74.67	83.27
Proliferative	90.68	71.33	79.85

Table 3.7: Table showing DenseNet169 class metrics

Class	**Precision** (%)	**Recall** (%)	**F1-Score** (%)
No DR	98.00	98.17	98.08
Stage 1	85.82	80.67	83.16
Stage 2	77.28	91.00	83.58
Stage 3	86.44	68.00	76.12
Proliferative	78.91	67.33	72.66

Table 3.8: Table showing ResNet101 class metrics

Class	Precision (%)	Recall (%)	F1-Score (%)
No DR	96.89	98.67	97.77
Stage 1	87.15	72.33	79.05
Stage 2	76.21	90.50	82.74
Stage 3	73.94	81.33	77.46
Proliferative	89.00	59.33	71.20

Table 3.9: Table showing InceptionV3 class metrics

Class	Precision (%)	Recall (%)	F1-Score (%)
NO DR	96.11	94.67	95.38
Stage 1	81.29	75.33	78.20
Stage 2	74.63	88.25	80.87
Stage 3	86.55	68.67	76.58
Proliferative	80.58	74.67	77.51

Table 3.10: Classification Report VGG19 class metrics



Figure 3.18: Normalized Confusion Matrix Of the DenseNet169 Model



Figure 3.19: Normalized Confusion Matrix Of the ResNet101 Model

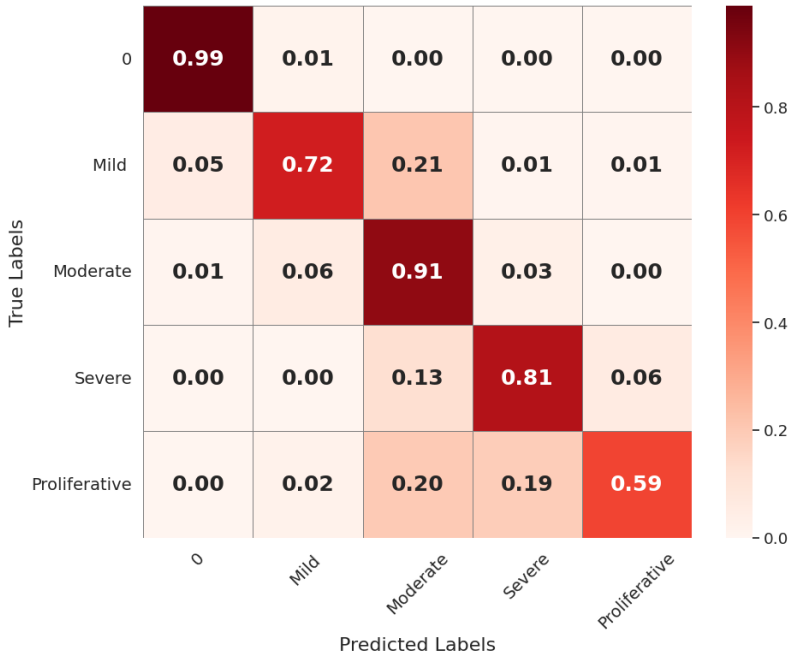


Figure 3.20: Normalized Confusion Matrix Of the InceptionV3 Model



Figure 3.21: Normalized Confusion Matrix Of the VGG19 Model

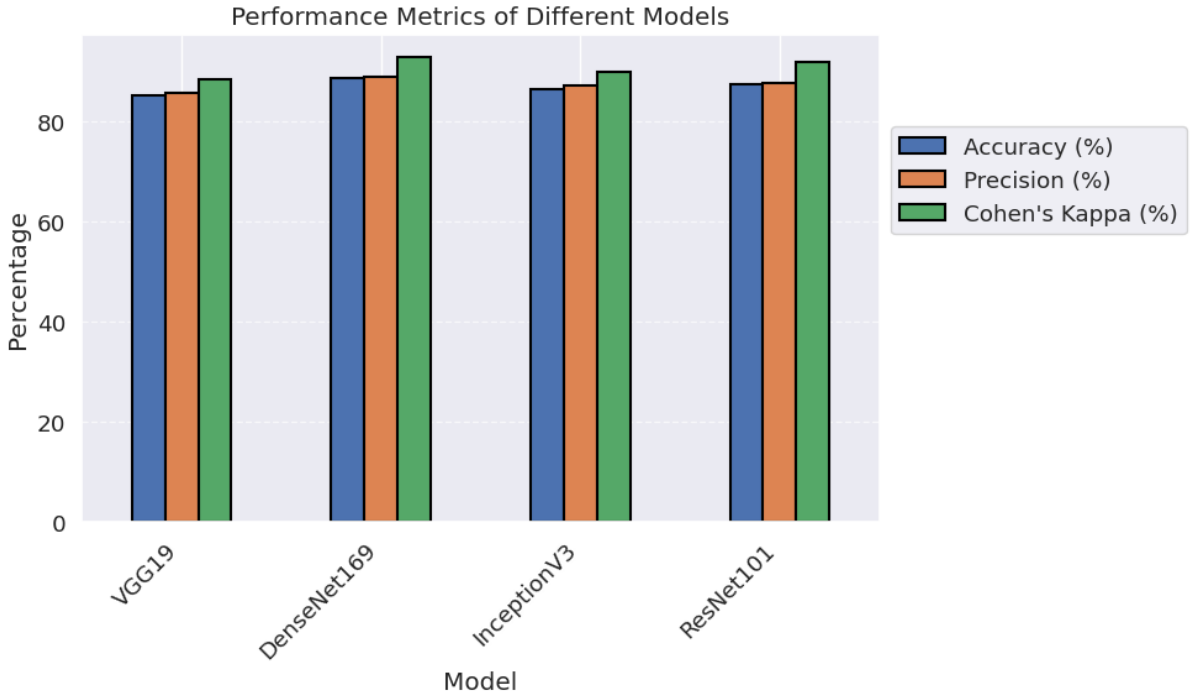


Figure 3.22: Bar Chart Of Performance Metrics Of The Different CNN Models after Fine Tuning

From the results displayed on (Table 3.6) we'll start by the primary metric which is the accuracy of total predictions, the highest accuracy of 89.75%, followed by ResNet101 at 87.38%, InceptionV3 at 86.38%, and VGG19 at 85.12%. The superior accuracy of DenseNet169 and ResNet101 can be attributed to their advanced architectural features, such as dense connections and residual blocks, which facilitate more efficient feature propagation and learning. The accuracy means how many predictions are true? but is it is

not an enough metric to judge since we can say model predictions can be out of "luck" or guess, since the highest probability takes the decision.

From the performance of all the models on the testing dataset, the results indicate they all near equally perform and work well on this task, with the accuracy of total and distinguishing and telling the differences between the classes and what to predict minimum 85%, and the precision of the negative cases returned is up to in the most performing model at 98.82% and 96.11% in the VGG19 model.

For the Cohen's Kappa score, which is a statistical measure used to assess the agreement between predicted and actual labels, considering agreement that could occur by chance. Ranging from -1 to 1, where 1 signifies perfect agreement and 0 indicates agreement equal to chance alone, In the context of Diabetic Retinopathy classification models, DenseNet169 achieved the highest Cohen's Kappa score of 93.09%, indicating excellent agreement beyond chance. ResNet101 followed closely with a score of 91.78%, demonstrating strong agreement. InceptionV3 and VGG19 achieved scores of 89.91% and 88.33%, respectively, reflecting different level of agreements that parallel the accuracy of predictions. These scores highlight DenseNet169 as particularly reliable and consistent in its predictions, essential for accurate medical diagnostics where precision is crucial.

We should consider keeping a model like Inception V3 which has a low number of layers at 48 and only 23 million parameters, The main aim of this is to create CNN model which keeps costs and complexity low, making it easier to implement and use for detection and classification on mobile device or other biomedical devices with constrained resources like memory and energy. Low complexity models can benefit biomedical devices and resource-constrained clinics and hospitals developing nations. It demonstrates potency in extracting features, segmentation, and object detection.

Since DenseNet has deeper networks due to the number of layers, uses batch normalization, offers the communication of features information across all layers over the network, and ensures gradients do not vanish and contribute to the adjustment of loss function. We can definitely say for this task the DensNet169 model is more robust and superior to other models in our current dataset and also data set of similar size. especially in timely diagnosis, meaning the prediction of inputted images at a clinical setting will take less time. This model presents an advantage over the other models both in terms of training and validation performance.

3.8.2 Class Metrics Discussion

As for the discussion of the Precision and F1, Recall score considering those are the averages taken of each class, and the results are paralleled through all the models considering the recall values of the earliest stages are at the highest compared to the severe stages of 3 and 4, the model is doing its purpose at the early stage classification.

Our focus will be directed towards the DenseNet169 Model, which accurately distinguished between classes and demonstrated superior recall values for the initial stages of Diabetic Retinopathy.

The Precision of Class 0 (Healthy Eye) is the primary key metric in this diagnostic task at peak on our custom data set is 98.88%, as it is crucial to minimize false negatives

in the diagnosis of Diabetic Retinopathy. While recall measures the model's ability to detect all negative observed cases correctly, the harm caused by false positives in this context is minimal, particularly when using CAD systems. This is because Stage 1 and Stage 2 of Diabetic Retinopathy do not require treatment or major intervention, and these systems primarily serve as a second opinion [98] and clarify their certainty, and the system is on time of false predictions only falsely taking the negative class as Stage 1 and Stage 2, we can expect this from the main reason people at early stages can only have tiny lesions like Micro-aneurysms or tiny bulges of the vessels.

However, the situation becomes different when we shift our focus to the positive cases. Consider class 1 of mild Diabetic Retinopathy and class 2 of moderate Diabetic Retinopathy, which are both early stages and challenging to visually diagnose, even for an ophthalmologist. In this context, an important metric to consider is the Recall Aka sensitivity. The recall for Stage 1 is 90.97% and for Stage 2 is 90.25%. These Figures below should give a general idea of how the metric of Recall and Precision works.

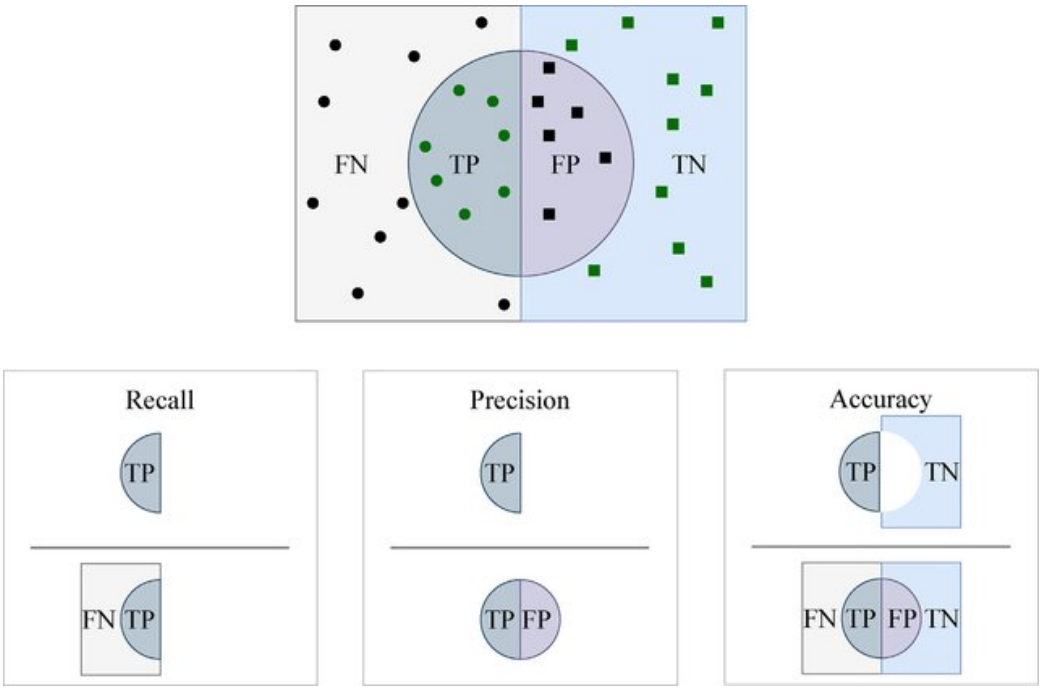


Figure 3.23: Visual illustration of the way precision and recall are measured [99]

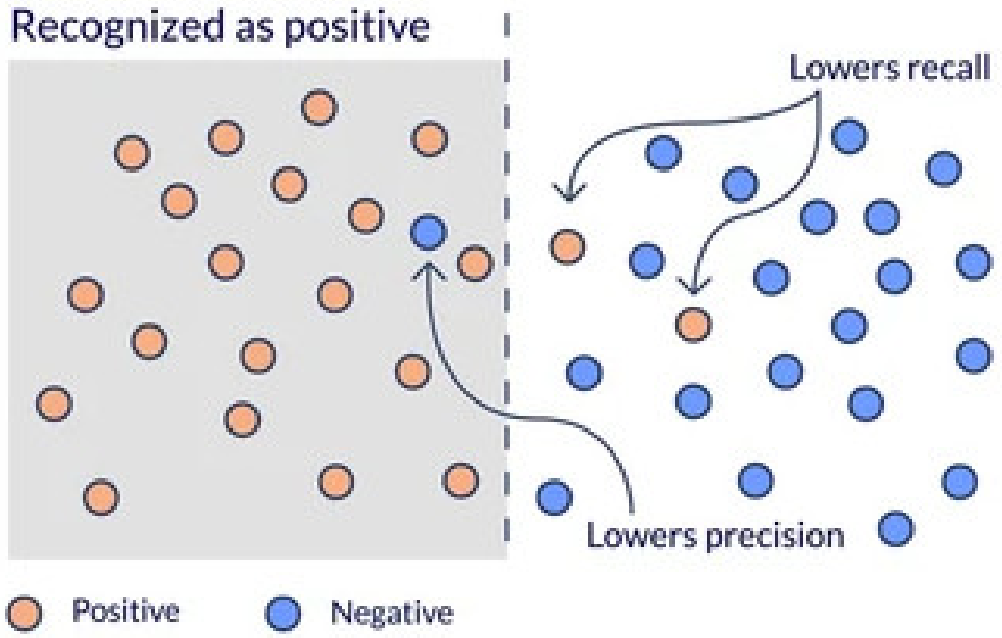


Figure 3.24: Visual illustration on an example of what can impact the recall and precision

The recall of these values, if we give the model a task of classifying a Diabetic Retinopathy at the early stages, what is the confidence our model will classify them correctly in comparison of all other classes of retinopathy including the lack of it? it is basically the confidence of the model to correctly identify the presence of the disease in the early stages if a patient of Diabetic Retinopathy stage 1 or 2 is presented.

And on the other hand, the precision of two first positive classes tells us how accurately will our model take a set of images that can vary in all the possibilities and classes and precisely classify that as Stage 1 and 2? Which is less relevant, but we'll expand on that later, We can also consider the F1 score which is the balance between both. But augmenting recall of these classes is actually the solution to the current problematic with the diagnosis of Diabetic Retinopathy both from previous automated models that use classical ML algorithms and handcrafted features, but also the difficulty of the conservative solution of the diagnosis by the manual screening and using the fundus imaging to analyze and look for small lesions that can easily hide when the lightning or angle is not adequate [100], So the *Standard Benchmark* for the evaluation of models working on this type of Disease is to consider the sensitivity rate which is calculated taking the precision metric consideration of only one class which is the healthy eye, class 0 and the recall of the early stages, And the recall of it's early stages.

And as of the precision for the first stages, we should also note something, if the precision metric of class 1 (Stage 1 Mild Diabetic Retinopathy) is a bit decreased, we should look for what stages are being misclassified as 1 ? Obviously it can be stage 0, but that isn't problematic earlier since our concerns are minimizing false negatives primarily, but as we see from the (Figure 3.18), the false grading of stage 1 are almost all mainly

coming from the model interpretation when given class 2 being diagnosed as 1, barely any instances from class 3 and 4 which require immediate interventions and are often easily detectable with the human eye and easily doubted after the trained clinicians examine the patient with questionnaire about their experience and symptoms, and these stages will only come after many years of having the disease, so healthcare staff also consider this when deciding how much do they trust the diagnosis made by the model, the healthcare staff and the patient should all be aware of the probabilistic strength of the model and consider it as a second opinion, So it is fair to say the model at the current state is on the right path for our local dataset and size of it.

Early Stages Performance of the Model

And in summary this below is the confusion matrix if we crop out the confusion matrix parts of stage 3 and 4 and calculate the scores, note this is not definite but provides an estimate of the model state dealing with a set of images given the first stages since the probabilistic contribution of class 3 and class 4 is also included in the final testing confusion matrix.

Metric	**Value**
Accuracy	93.98%
F1 Score	94.03%
Precision	94.14%
Recall	93.15%

Table 3.11: Overall performance metric on the early stage diagnosis

Class	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Class 0	98.82	97.33	98.07
Class 1	85.27	90.97	88.03
Class 2	93.77	91.16	92.45

Table 3.12: Early Stages Classification Report

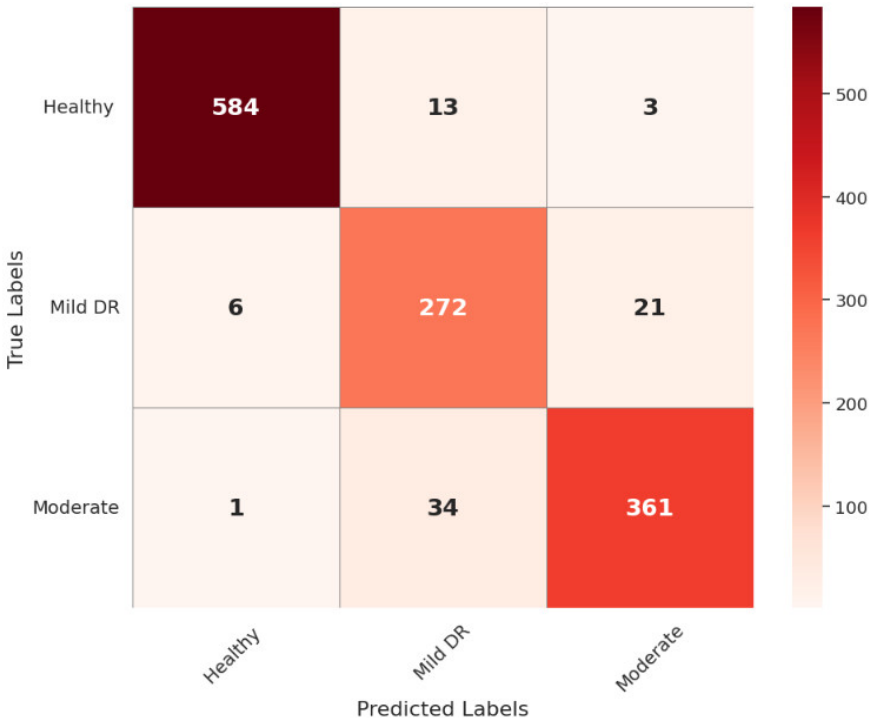


Figure 3.25: Early stages confusion matrix

This provides a general overview of the results provided by the models and how it dealt with the distinguishing between a healthy retina and mild and moderate Diabetic Retinopathy, and a severe proliferative stage. it remains an issue in deep learning the black box problem which is the non interpret able nature of the models, it is often difficult to train a model that interprets why he made such a decision, what pixels were the decision point between class 2 and 3 ? one thing to also note this is the performance of the model on our own custom dataset which is a part of the EYEPACS dataset, it is better if we had access to standard benchmarks and were able to train models to match.

3.9 Architectural Concepts of DenseNet169

DenseNet, short for Densely Connected Convolutional Networks, represents a significant advancement in the field of deep learning, particularly in the design of convolutional neural networks (CNNs). The architecture, introduced by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger in their paper "Densely Connected Convolutional Networks," published in 2016, addresses several key challenges faced by traditional CNNs, such as the vanishing-gradient problem, parameter inefficiency, and overfitting. DenseNet169 is a specific model of the DenseNet family, characterised by its depth of 169 layers, making it one of the deeper variants within this architecture in the (Figure 3.26) we see how the dense blocks are programmed in the DenseNet architectures, and in the 5Figure 3.27) we get detailed information about the layers concerning the DenseNet169 model.

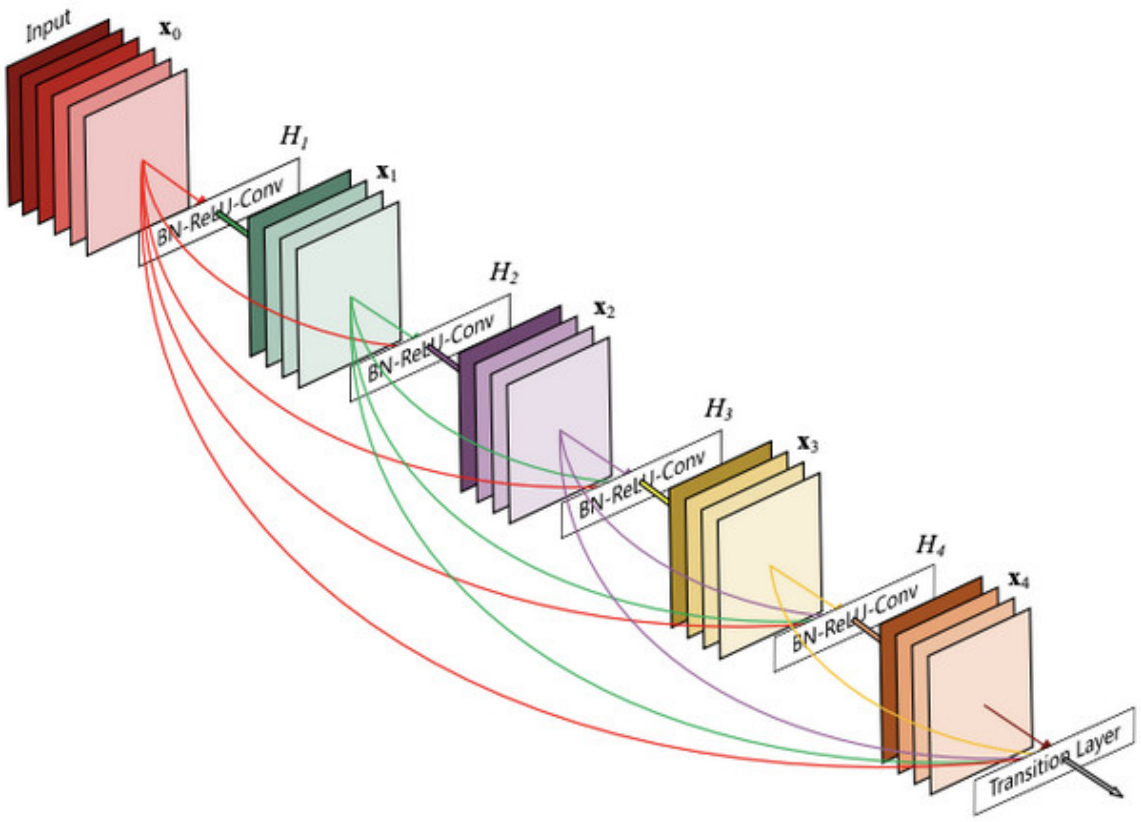


Figure 3.26: Architecture Of Dense Blocks [77]

Layers	Output Size	DenseNet 169
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Transition Layer (3)	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool
	1000	1000D fully-connected, softmax

Figure 3.27: Specifications of layers used in thearchitecture of DenseNet169

The fundamental idea behind DenseNet is the introduction of dense connectivity patterns. Unlike traditional CNNs where layers are connected sequentially, DenseNet connects each layer to every other layer in a feed-forward manner. This dense connectivity pattern has several implications, we'll expand on them on the sections after.

3.9.1 Feature Reuse:

Each layer in DenseNet produces a set of feature maps, which are then used as inputs to all subsequent layers. This approach encourages feature reuse, as features learned early in the network are directly accessible to later layers, potentially improving the quality of higher-level representations. By reusing features across layers, DenseNet reduces redundancy in the learned features. Since each layer adds new features to the collective knowledge of the network, the overall model becomes more efficient in terms of parameter usage.

The direct connections from early layers to later layers facilitate the propagation of gradients during backpropagation. This dense connectivity helps mitigate the vanishing gradient problem, allowing for the effective training of very deep networks. In contrast to conventional convolutional networks, this dense connection design has the advantage of requiring fewer parameters. This is because there is no need to relearn redundant feature mappings, which is an effect that may appear to be counterintuitive.

DenseNet169 consists of four main components: the initial convolution layer, three dense blocks, transition layers, and the final classification layer. The network starts with a standard convolution layer, followed by batch normalization and a rectified linear unit

(ReLU) activation function. This layer serves to transform the input images into a set of feature maps that can be processed by the subsequent dense blocks.

3.9.2 Dense Blocks

The core of DenseNet169 is composed of dense blocks. Each dense block is a collection of layers, where each layer takes all preceding feature maps as input and produces a fixed number of output feature maps. The outputs of each layer are concatenated with the inputs to form the next layer's input. This process continues, adding more feature maps as the network progresses through the dense block. These constitute the core building blocks of DenseNets. A dense block comprises a series of convolutional layers interwoven with dense connections Each layer within the dense block takes the concatenated feature maps from all preceding layers within the block as input, along with its own feature maps. This concatenation operation enables feature reuse and encourages information exchange across layers.

3.9.3 Transition Layers

Transition layers are an essential component in the architecture of DenseNet, serving to manage the growth of the network's width between dense blocks. These layers are strategically placed between dense blocks and are responsible for controlling the dimensionality and size of feature maps. A transition layer is composed of a batch normalization layer, a 1X1 convolution layer, and a 2x2 average pooling layer. The batch normalization layer and 1 X 1 convolution layer help to reduce overfitting and improve the generalization ability of the network. The 2 X 2 average pooling layer reduces the spatial dimensions of the feature maps, allowing the network to learn more abstract features in the subsequent dense block.

By introducing a degree of downsampling, transition layers effectively control the growth of feature maps, similar to traditional CNN architectures. This downsampling process helps to reduce the computational complexity of the network and prevent overfitting. Overall, transition layers play a crucial role in the efficient and effective operation of DenseNet, enabling the network to learn more abstract and complex features.

The Advantages of DenseNet169 architecture is efficient use of parameters, due to its dense connectivity pattern, DenseNet169 can achieve high levels of accuracy with fewer parameters compared to CNN models of similar depth. The direct connections between layers in DenseNet169 facilitate better gradient flow during training, mitigating the vanishing gradient problem and enabling the training of deeper networks.

Since its introduction, DenseNet169 and its variants have been widely adopted in various computer vision tasks, including image classification, object detection, and semantic segmentation. The architecture's efficiency and effectiveness have made it a popular choice for researchers and developers alike.

3.9.4 Batch Normalisation

Batch Normalisation (BN) is a novel technique that has significantly impacted the training and performance of deep neural networks, it is used across most of the distinct layers

of DenseNet Architectures, from the input, to the dense blocks, to transition layers. Batch Normalisation, introduced by Sergey Ioffe and Christian Szegedy in 2015 [101], BN addresses a fundamental challenge in training deep networks, the problem of internal covariate shift. Internal covariate shift refers to the change in the distribution of each layer's inputs as the network learns, which can lead to unstable and inefficient training dynamics. This instability becomes more pronounced as the depth of the network increases, making it increasingly challenging to train very deep networks effectively.

The core idea behind BN is to normalize the inputs to each layer by adjusting them to have zero mean and unit variance. This normalization is performed separately for each mini-batch of data during training, which means the term "batch normalization." By ensuring that the inputs to each layer have a consistent distribution, BN stabilizes the learning process, allowing for higher learning rates and faster convergence. This stabilization is crucial because it enables the network to learn more effectively from the data, reducing the need for manual tuning of hyperparameters and speeding up the training process.
the steps of batch normalization in standard neural networks are as below.

1. **Input to BatchNorm:** we suppose we have a mini-batch of inputs $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, where $x^{(i)}$ is the input to the layer for the i-the example in the mini-batch.

2. **Compute Mean and Variance:**

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, \tag{3.7}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_B)^2. \tag{3.8}$$

3. **Normalize:** Normalize each input $x^{(i)}$ in the mini-batch using the mean and variance calculated:

$$\hat{x}^{(i)} = \frac{x^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{3.9}$$

   where ϵ is a small constant (typically 10^{-5}) added for numerical stability.

4. **Scale and Shift:** Introduce two learnable parameters, γ (scale) and β (shift), for each feature dimension:

$$y^{(i)} = \gamma\hat{x}^{(i)} + \beta. \tag{3.10}$$

5. **Output:** The final output Y of the BatchNorm layer for the mini-batch is:

$$Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\}. \tag{3.11}$$

6. **During Training (Normalization Across Mini-batches):** Update the moving averages of μ and σ^2 for use during inference, which are computed across multiple mini-batches.

In convolutional neural networks (CNNs), Like DenseNet, batch normalization (BN) operates uniquely to normalize activations across each channel's spatial dimensions within mini-batches of 3D inputs like RGB images. This approach differs from fully connected

layers by calculating mean and variance separately for each channel.

As we have seen earlier, BN introduces two learnable parameters for each feature in the layer, a scale factor (γ) and a shift (β). These parameters allow the network to learn the optimal scale and offset for each feature, further enhancing the model's ability to generalize from the training data. The scale factor (γ) controls the amount of scaling applied to the normalized features, while the shift (β) allows for translating the normalized features, effectively restoring the representational power lost during normalization.

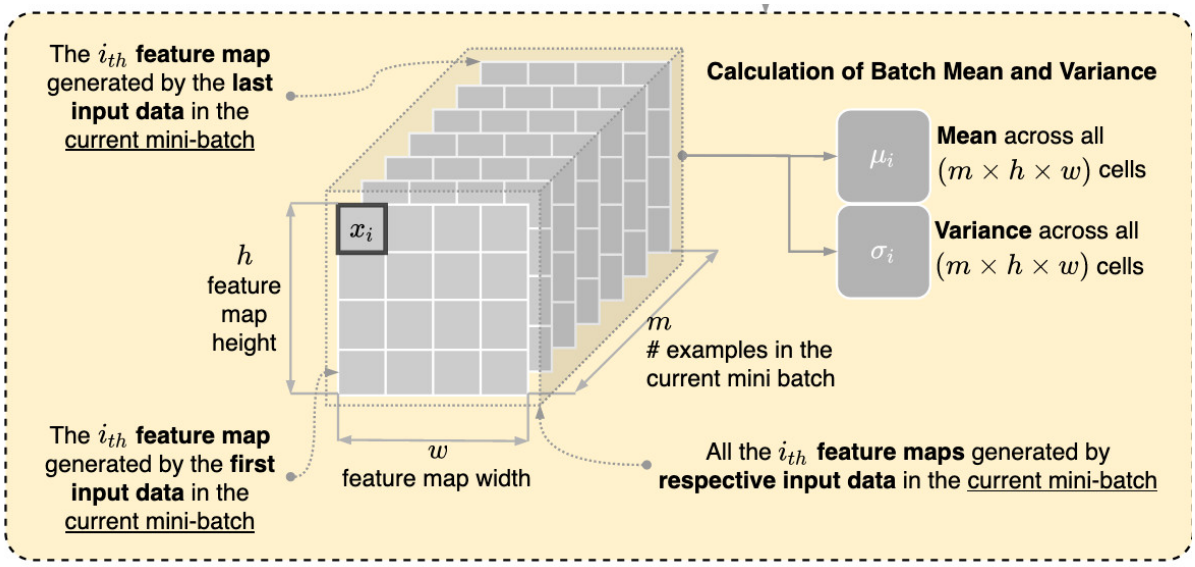


Figure 3.28: Batch Normalization in CNN Models [101]

However, during inference or evaluation, when the full dataset is available, Batch Normalization switches to using the population mean and variance, which are computed over the entire training dataset. This switch ensures that the normalization remains consistent between training and evaluation phases, preventing any discrepancies that could arise from using batch-specific statistics.

3.10 Model Inference

This example demonstrates a sample of the inference of our model in the diagnosis and grading of Diabetic Retinopathy using retinal fundus imaging input. The model processes the input image and provides a diagnostic assessment along with a corresponding confidence level of the classification. Our observations indicate a positive correlation between image quality—specifically, optimal lighting and contrast—and the accuracy of the model's output. Consequently, it is imperative that ophthalmic and healthcare professionals adhere to protocols ensuring the capture of high-quality retinal fundus images.

The CAD system, functioning as a reliable second opinion, enhances diagnostic reliability by providing supplementary assessments based on quantitative image analysis. Deployable on any system that supports Python or through cloud-based platforms, the model offers flexibility in integration across healthcare settings. It operates seamlessly to augment clinical decision-making, particularly in cases where diagnostic certainty is crucial. In the figures below are the testing interface we made for our system.

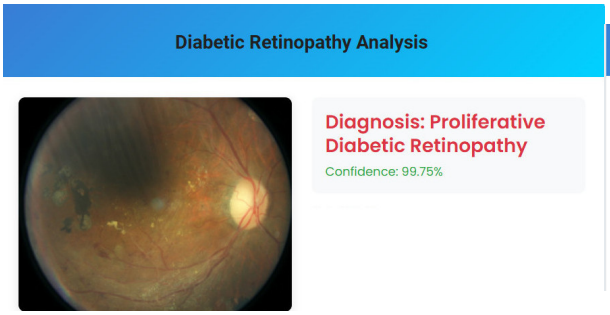


Figure 3.29: First image of inference

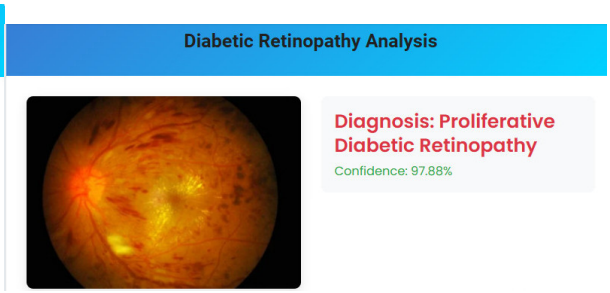


Figure 3.30: 2nd image of inference

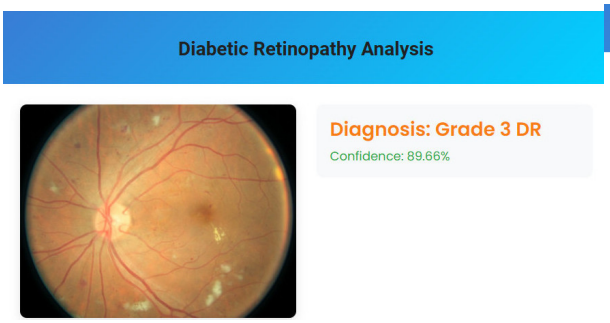


Figure 3.31: 3rd image of inference

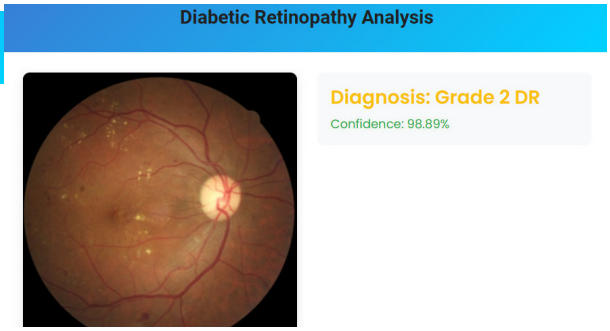


Figure 3.32: 4th image of inference

Figure 3.33: Images of the model's inference

It is noteworthy that false diagnoses consistently exhibit low confidence after assessment by the system. Clinicians should also take into consideration additional patient-specific factors, such as medical history and how the long the patient had diabetes for, to contextualize the model's predictions effectively. This comprehensive approach enables a nuanced evaluation of the CAD system's diagnostic recommendations, thereby optimizing patient care and management strategies.

Conclusion

In summary, this chapter has explored the essential elements of creating a classification model that is effective for Diabetic Retinopathy. We initiated our review by exploring the historical development of image classification, defining its principles and the diverse methodologies that have impacted the field. We highlighted the importance of diverse and well-annotated datasets in the future, emphasizing the effectiveness of transfer learning with advanced CNN architectures like ResNet101 and DenseNet169. The conversation encompassed augmentation techniques, which are essential for improving model generalization in medical imaging applications. We assessed critical metrics, including accuracy, precision, recall, F1-score, and Cohen's Kappa, and interpreted their performance in relation to the various stages of Diabetic Retinopathy classification. Furthermore, the study investigated optimization strategies, such as dynamic learning rate scheduling and fine-tuning of the CNN layers in accordance to our dataset, to underscore their critical role in improving the stability and performance of the model. The development of this classification model for Diabetic Retinopathy is not just a technical achievement, but a

significant step towards improving patient care. Early diagnosis is crucial in managing this condition, yet many patients remain untreated and suffer vision loss due to late detection. Considering that the consistency of grading early stages of Diabetic Retinopathy by human experts is only about 60%, there's a clear need for more reliable diagnostic tools. This deep learning-based system addresses this need by potentially improving diagnostic accuracy, especially in early stages. Moreover, by automating the initial screening process, this system makes the treatment more affordable, accessible, and efficient. It can help healthcare providers identify at-risk patients earlier, enabling timely interventions that could prevent or slow the progression of vision loss. Thus, this model not only advances the technical field of medical image classification but also has the potential to significantly impact patient outcomes and public health.

This chapter functions as a fundamental guide for the utilization of Deep Learning, CNN networks in specific to medical imaging to facilitate the accurate diagnosis and grading of Diabetic Retinopathy.

Chapter 4

Conception And Implementation Of Retinal Blood Vessel Segmentation System

In this chapter, we commence by over-viewing the concept of image segmentation, including its historical context and various techniques. Subsequently, we systematically lay down the groundwork to comprehensively detail the implementation process of our model. We go over into the nature of our datasets and justify the necessity for employing distinct models, considering that fundus images can be obtained from two different perspectives relative to the optic nerve. We outline the data preprocessing procedures, the methodology for data augmentation, the training strategy, and the selection of parameters. Additionally, we discuss how we monitored our model to optimize performance, leveraging both our training regimen and available data to its fullest extent.

Following these discussions, we proceed to analyze the model results and elaborate their implications for the task of segmenting and extracting Retinal blood vessels.

4.1 Image Segmentation, Types And Approaches

Image segmentation refers to the computational process of dividing a digital image into multiple segments or sets of pixels, where each segment corresponds to a distinct object or part within the image. This technique plays a pivotal role across numerous domains, including but not limited to, object recognition, scene understanding, and medical imaging diagnostics. The primary objective of image segmentation is to simplify and/or change the representation of an image into something more meaningful and easier to analyze. It essentially aids in extracting useful information from complex visual data by isolating objects of interest from the background or other objects. Various methodologies exist for achieving image segmentation, ranging from traditional thresholding and edge detection to more advanced machine learning-based approaches, each tailored to address specific challenges and requirements in image analysis.

4.1.1 Semantic Segmentation

Semantic segmentation is a type of image segmentation where the goal is to assign a semantic label or category to each pixel in the image. This technique is particularly

useful for scene understanding and object recognition tasks. For example, in an image of a street scene, semantic segmentation would classify pixels into categories such as "car," "pedestrian," "road," and "building." Popular deep learning architectures for semantic segmentation include Fully Convolutional Networks (FCNs) [102], U-Net, and DeepLab [103].

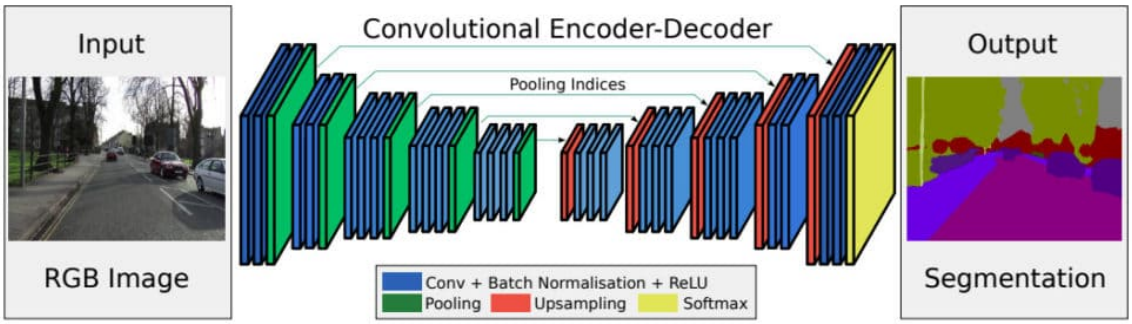


Figure 4.1: An image showing the process of semantic segmentation using encoder decoder neural networks

4.1.2 Instance Segmentation

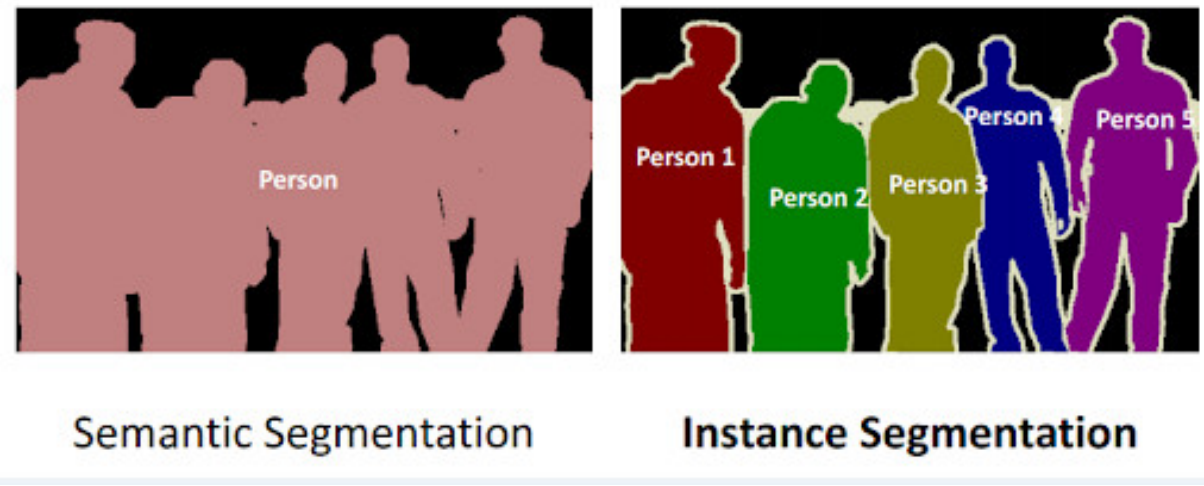


Figure 4.2: Figure showing the key difference between instance and semantic segmentation [104]

4.1.3 Unsupervised Segmentation Approach

Unsupervised systems generally operate on many types of data, images are one of them, Unsupervised segmentation techniques aim to partition an image into meaningful regions without relying on labeled training data or prior knowledge about the image content [105]. These methods typically rely on low-level image features, such as color, texture, or edge information, to group pixels into coherent regions as best as possible. Common well known unsupervised segmentation algorithms include k-means clustering, which is a popular choice mean-shift, and graph-based methods like normalized cuts. The core idea

behind clusters is to give unlabeled data to the model and have it learn patterns from them and draw the divide line and boundary that separates the classes, for the case of image segmentation, if we take a pixel which group of pixels it is most likely a like?

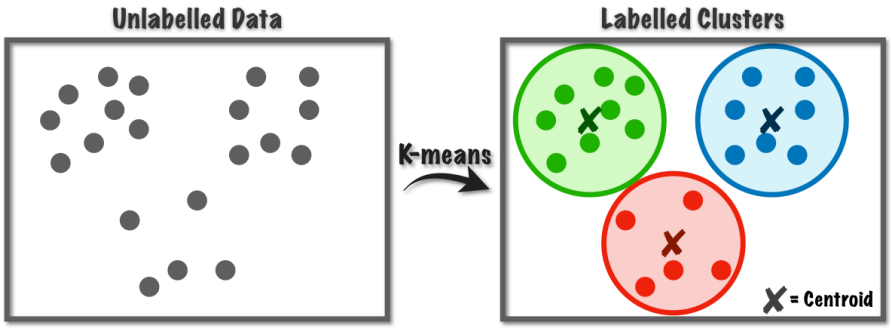


Figure 4.3: An example of clustering algorithm results



Figure 4.4: An example of a simple image where the model was trained to cluster pixels into different categories [106]

4.1.4 Object Detection

Object detection is a branch task in computer vision that involves identifying and localizing multiple objects within an image or video. Unlike image segmentation, which focuses on dividing an image into semantic or instance-based regions, object detection specifically aims to detect and classify individual objects while also providing their precise spatial coordinates.

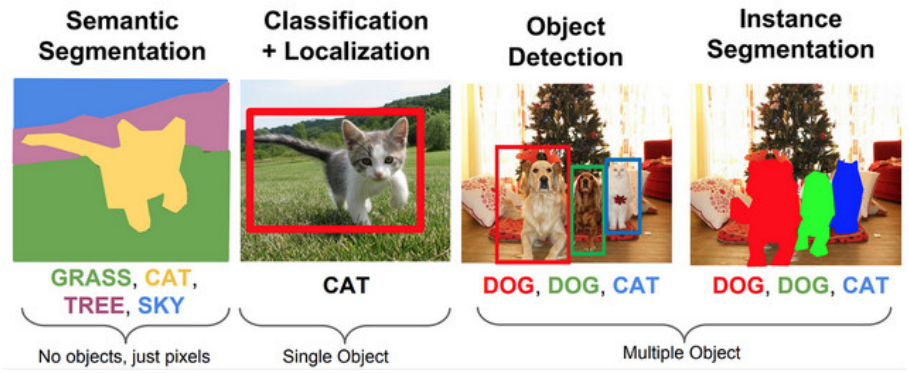


Figure 4.5: Object detection example and differences with other approaches [107]

4.1.5 Key challenges In Image Segmentation tasks

- **Scale Variation**: Objects may appear at different scales within images, requiring segmentation algorithms to be robust to size variations, the solution to this is often more data.
- **Occlusion**: Objects can be partially obscured and cut of in the view by other objects or background elements, making their accurate segmentation challenging.
- **Complex Backgrounds**: Images with cluttered or complex backgrounds can interfere with segmentation accuracy by blending object boundaries with background elements.
- **Object Class Variability**: Variations in object appearance within the same class (e.g., different types of vehicles) can challenge segmentation algorithms.
- **Robustness to Image Quality**: Variations in image quality due to factors such as lighting conditions, resolution, noise, or compression artifacts can affect the performance of image classifiers.

4.2 Data Collection

4.2.1 First Model Dataset (Macula Centered Fundus Images)

The DRIVE (Digital Retinal Images for Vessel Extraction) dataset, introduced by Jooes Staal et al, in their influential publication "Ridge-based vessel segmentation in color fundus images of the retina," is a fundamental resource for the development of Retinal vessel segmentation solution systems [108]. This collection of photos and accompanying segmentation data is specifically setup to aid in the process of Retinal vascular layer segmentation, which is essential for the identification and monitoring of medical disorders like Diabetic Retinopathy, and other common diseases that affect the vascular layer. The dataset consists of 20 color fundus photos, including seven cases that display aberrant pathologies. These images were obtained from a Dutch Diabetic Retinopathy screening program. The imaging procedure was performed utilizing a Canon CR5 non-mydriatic 3CCD camera, guaranteeing the acquisition of high-quality images. The dataset includes additional manual segmentations conducted by a skilled ophthalmologist for the training subset. For the test subset, two different observers performed manual segmentations, with the primary observer's delineations considered as the standard for evaluating model performance. Each image in these subsets is accompanied by a circular field of view (FOV) mask. The dataset's significance and relevance to the field of Retinal vascular segmentation are emphasized by the careful curation and annotation methods, which are supervised by ophthalmology professionals. The DRIVE dataset has a community of researchers and developers focused on medical images segmentation. The dataset is accessible to the public with specific conditions. It is specifically designed to prioritize Retinal vascular segmentation, which helps in the progress of diagnosing and treating eye illnesses. This (Figure 4.6) shows a sample of the DRIVE Dataset, images and their corresponding masks.



Figure 4.6: A sample of the first model's dataset

4.2.2 Second Model Dataset (Optic Nerve Centered Fundus Images)

The CHASEDB1 dataset, created by researchers at Kingston and St. George's Universities in London, is a valuable resource for developing and evaluating Retinal vessel segmentation algorithms [109]. It's a subset of Retinal images from the Child Heart and Health Study in England (CHASE) focusing on 14 individuals from diverse ethnicities. CHASEDB1 offers 28 high-resolution color fundus images, each captured from one of the individual's eyes. Notably, each image has two corresponding "ground truth" images created by independent experts. These ground truth images act as a reference, where every pixel is labeled as either "1" (part of a blood vessel) or "0" (not part of a vessel). By making this data publicly available, researchers can train and test their vessel

segmentation algorithms on a standardized dataset, allowing for direct comparisons of performance between different approaches.

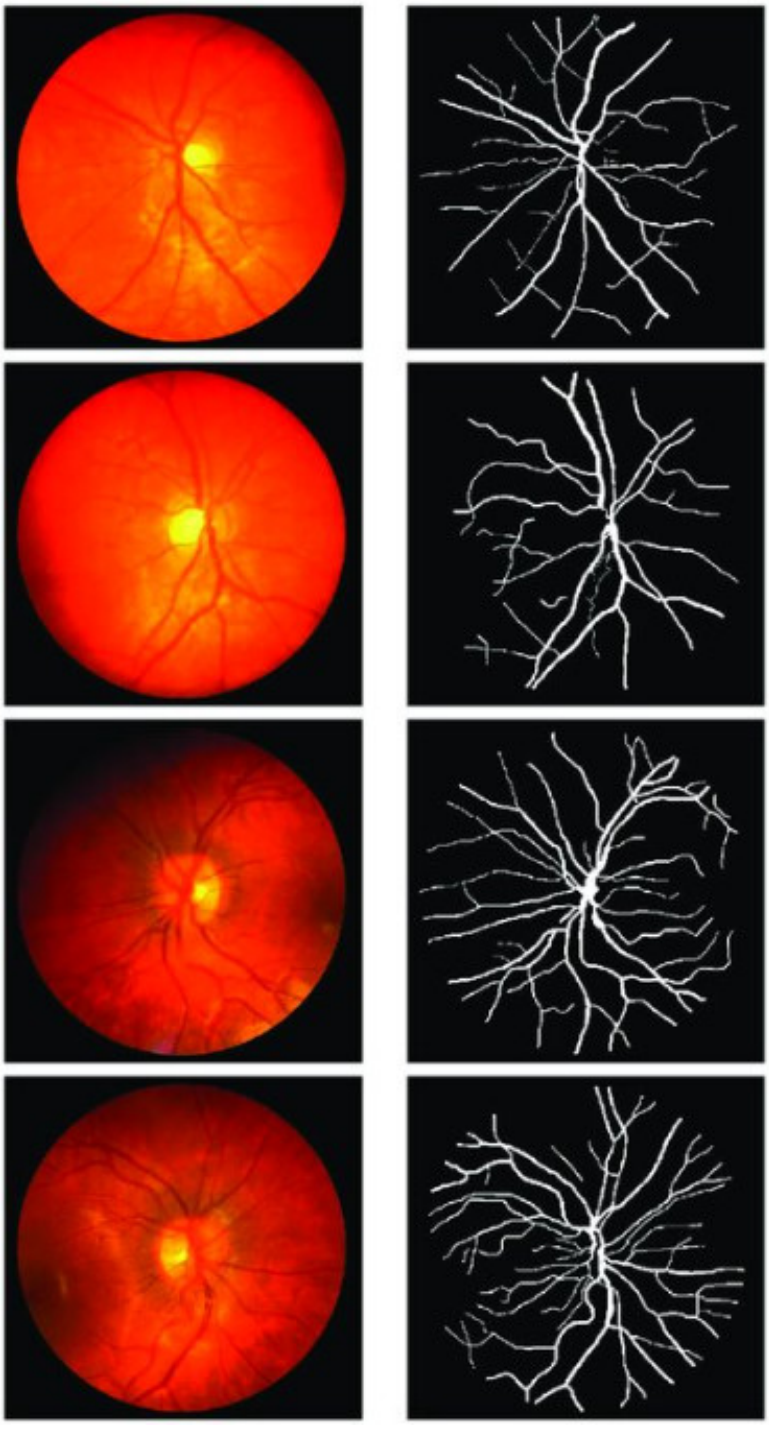


Figure 4.7: A sample of CHASE Dataset

4.3 Additional Evaluation Metrics

In this section, we describe the additional evaluation metrics used to assess the performance of our Retinal vessel segmentation model. These metrics are crucial for understanding how well the model differentiates between vessel and non-vessel pixels.

4.3.1 Dice Coefficient

The Dice coefficient, also known as the Dice similarity index, is used to gauge the overlap between the segmented output and the ground truth. It is particularly useful for evaluating segmentation algorithms because it takes both false positives and false negatives into account[110]. The Dice coefficient is defined as:

$$\text{Dice Coefficient} = \frac{2 \cdot |A \cap B|}{|A| + |B|} \tag{4.1}$$

where:

- A is the set of pixels identified as vessels by the segmentation model.
- B is the set of ground truth vessel pixels.
- $|A \cap B|$ is the number of pixels correctly identified as vessels by the model (True Positives).

The Dice coefficient ranges from 0 (no overlap) to 1 (perfect overlap). A higher Dice coefficient indicates better segmentation performance.

4.3.2 Area Under the Curve (AUC)

The Area Under the Curve (AUC) refers to the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings [111]. AUC provides a single scalar value that summarizes the performance of the binary classifier across all thresholds. It is defined as:

$$\text{AUC} = \int_0^1 \text{ROC}(t)\, dt \tag{4.2}$$

where:

- $\text{ROC}(t)$ is the ROC curve as a function of the threshold t.

The AUC value ranges from 0 to 1, with a value closer to 1 indicating a better performing model. An AUC of 0.5 suggests no discriminative power, equivalent to random guessing.

4.3.3 Specificity

Specificity, also known as the true negative rate, measures the proportion of actual negatives (non-vessel pixels) that are correctly identified as such by the segmentation model. It is an important metric for ensuring that non-vessel regions are not incorrectly classified as vessels, which can reduce false positives. Specificity is defined as:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.3}$$

4.4 Data Preprocessing

In this part of the work, preprocessing plays a pivotal role in optimizing the dataset for effective model training, as highlighted in our previous chapter. The initial steps involve resizing images to a standardized dimension of 576x576 pixels, ensuring uniformity in input size across the dataset. Subsequently, pixel values are converted from their original uint8 format to float and normalized within the range of 0 to 1, aligning them with the requirements of pretrained models. Moreover, contrast and brightness adjustments are applied to enhance the visibility of vascular patterns and their edges, crucial for accurate feature extraction during training. Augmentation techniques further augment the dataset by introducing variations that enhance the model's ability to generalize from diverse inputs. Additionally, special attention is given to masks, ensuring binary pixel representation where all non-zero values are set to 1, facilitating precise delineation of features, particularly in grayscale images.

4.5 Data Augmentation

For this part of the project while working on the model that segments Retinal images, We only performed **horizontal segmentation**, part of the reason is that on the training data set, so we stay within the anatomical expectations of vascular layer of the eye, generally vessel are often always taking the same path spreading from the primary artery and vein so for the sake of this model we expect of it to be used on a limited fundus imaging zoom and angle on clinical practice We only performed horizontal augmentation, taking the fundus images and applying the geometric horizontal flip, and the matching mask exactly the same.

Additionally, patch-level data augmentation is a widely recognized technique in segmentation tasks [112]. Future research and development should explore this approach to further enhance the model's performance.

4.6 Model Training

In this section, we go over the methodological framework for training our U-NET based model. The model's convergence was achieved through an iterative process, necessitating hundreds of epochs to achieve minimal optimization. Two distinct models were developed and evaluated: Model 1, utilizing the DRIVE dataset with macular-centered images, comprised 40 images for training, 10 for validation, and the standard 20 benchmark images for testing. Model 2 incorporated a total of 28 images, with 7 reserved for validation. The remaining 21 images underwent horizontal augmentation, resulting in a cumulative training set of 42 images. The training was initiated with a learning rate of 1×10^{-4} , employing an early stopping mechanism with a patience parameter to mitigate premature convergence to local minima. Upon observing enhanced loss function performance, the learning rate was decremented by 90% iteratively, with a minimum of 1×10^{-7} learning rate to facilitate refined gradient adjustments during the latter stages of training. A batch size of 8 was selected to balance computational constraints and training stability.

For the encoder component, we leveraged the DenseNet169 architecture, which has demonstrated superior feature extraction capabilities compared to ResNet50 and VGG16 in our preliminary experiments. The dense connectivity pattern in DenseNet169 allows for

improved information flow and feature reuse, critical for capturing the intricate vascular structures in Retinal images. The decoder segment utilized a modified U-Net architecture, chosen for its efficacy in medical image segmentation tasks. The U-Net's skip connections facilitate the fusion of low-level features with high-level semantic information, crucial for accurate vessel extraction. In the (Figure 4.8) is the representation of u-net architecture.

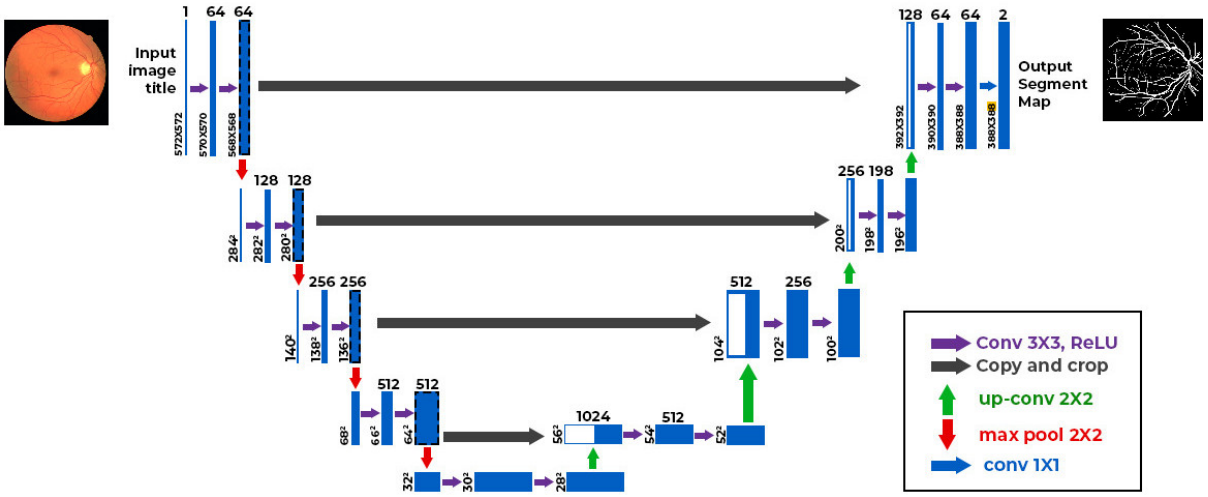


Figure 4.8: Architecture of the UNet Model[77]

The unique aspect of U-Net when it starts generating features in the decoder is its use of transposed convolutions (also known as upconvolutions or deconvolutions) instead of traditional convolution layers. Unlike standard convolutions that typically reduce the spatial dimensions of the input, transposed convolutions increase the spatial resolution of feature maps. This property is crucial for U-Net's ability to reconstruct high-resolution output images from lower-dimensional representations.

Transposed convolutions work by inserting zeros between the elements of the input feature map, and then applying a standard convolution operation with learned filters. This process effectively upsamples the feature map, allowing the network to generate higher-resolution outputs. The U-Net architecture specifically uses a 2x2 transposed convolution with stride 2 to upscale both the spatial resolution and channel depth during the decoding phase 1. The (Figure 4.9) gives us a simplified illustration of the upconv process.

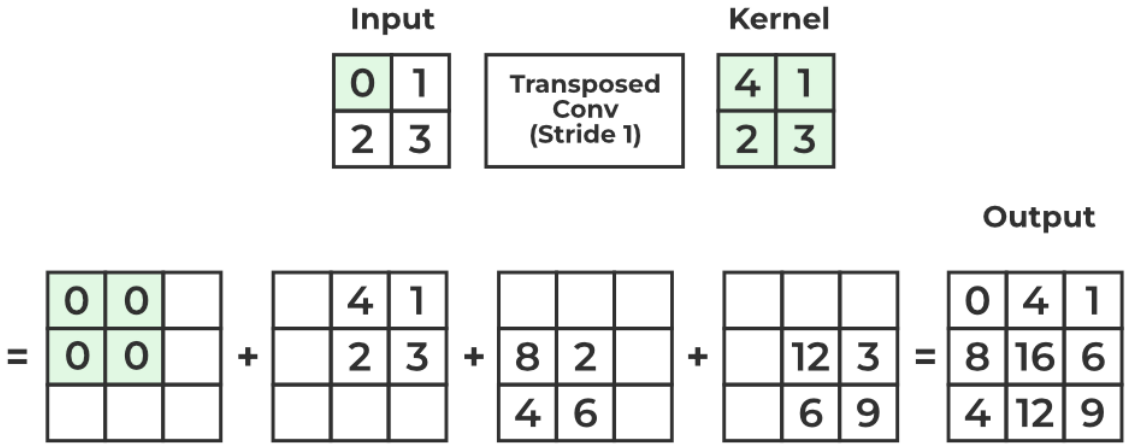


Figure 4.9: Illustration of the Transposed Convolution Operation

The choice of up-sampling method in the decoder is significant because it affects the quality and accuracy of the reconstructed image. Transposed convolution offers a learnable approach that can adapt to the specific characteristics of the data. This makes it particularly effective for tasks requiring precise reconstruction of details, such as image segmentation in medical imaging applications like our case.

Moreover, the use of transposed convolutions in conjunction with skip connections allows U-Net to combine low-level features with high-level semantic information effectively. This combination helps in preserving fine-grained details and improving the overall performance of the segmentation task

The interface between the encoder and decoder was optimized through feature map alignment and channel reduction techniques to ensure seamless information transfer. This encoder-decoder system allowed for robust feature extraction and precise segmentation map generation. Binary cross-entropy was employed as the loss function, appropriate for the binary nature of vessel segmentation. The output layer utilized a **sigmoid** activation function at the level of pixels, aligning with the pixel-wise binary classification nature of Retinal blood vessel segmentation. Notably, both the DenseNet169 backbone and U-Net decoder layers were unfrozen during training, allowing for end-to-end fine-tuning specific to our Retinal vessel segmentation task and customized dataset.

The model's training was monitored through the validation loss metric, ensuring optimal performance and generalization capacity. Additional metrics such as Dice coefficient and Jaccard index are also often used in segmentation tasks to provide a comprehensive assessment of segmentation accuracy.

Here below is the (Equation 4.4) of the Loss Function.

$$\text{Binary Cross Entropy} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \tag{4.4}$$

- N: Total number of samples in the dataset.

- i: Index representing each individual sample from 1 to N.
- y_i: True label of the i-th sample, where $y_i = 1$ indicates the sample belongs to the positive class and $y_i = 0$ indicates the sample belongs to the negative class.
- p_i: Predicted probability that the i-th sample belongs to the positive class.
- $\log(p_i)$: Natural logarithm of p_i, penalizes the model if predicted probability p_i is low when $y_i = 1$.
- $\log(1-p_i)$: Natural logarithm of $1-p_i$, penalizes the model if predicted probability $1-p_i$ is low when $y_i = 0$.

4.7 Results And Discussion Of The First Model (Macula Centred)

Metric	**Value (%)**
Accuracy	96.55
Precision	82.19
Recall / Sensitivity	77.53
Specificity	98.39
DICE Coefficient	79.61
Area Under the Curve (AUC)	97.07

Table 4.1: Performance Metrics on the Macula Centered Dataset

4.7.1 Visual Segmentation Samples From the First Model

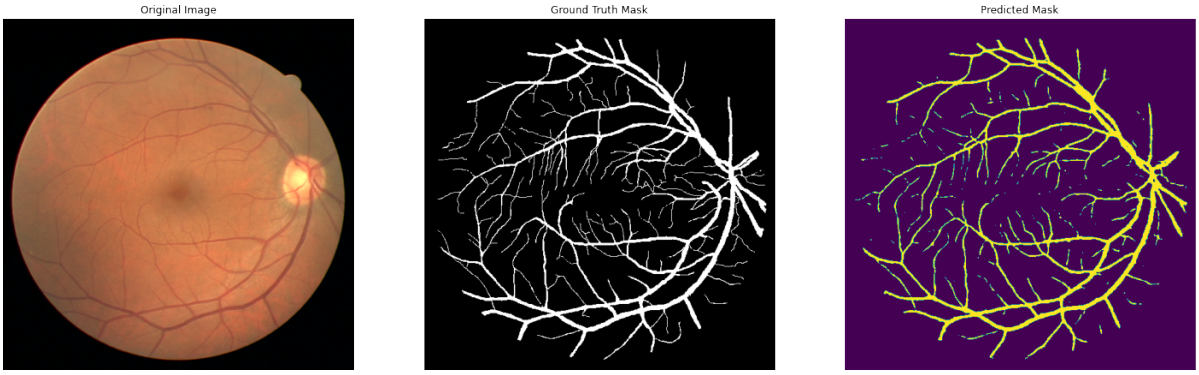


Figure 4.10: First sample of Retinal vessel segmentation using the Macula centered images

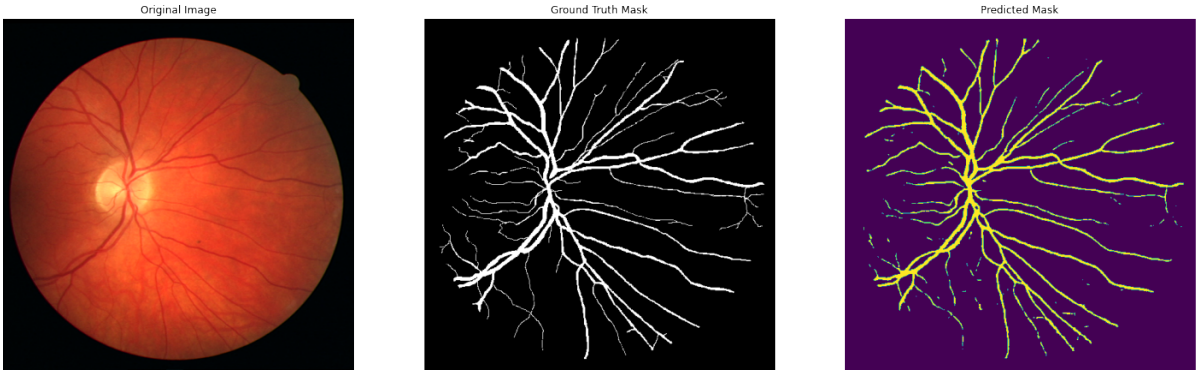


Figure 4.11: Second sample of Retinal vessel segmentation using the Macula Centered fundus image

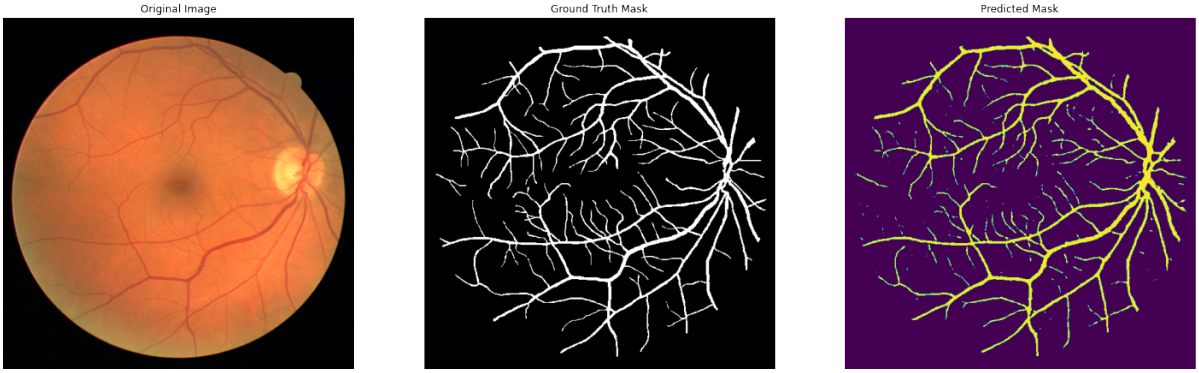


Figure 4.12: 3rd sample of Retinal vessel segmentation using the Macula Centered fundus image

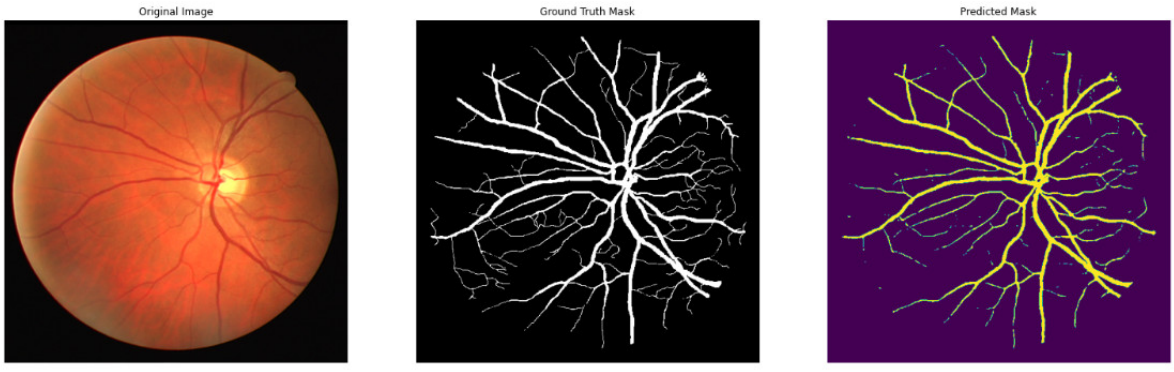


Figure 4.13: 4th sample of Retinal vessel segmentation and extraction

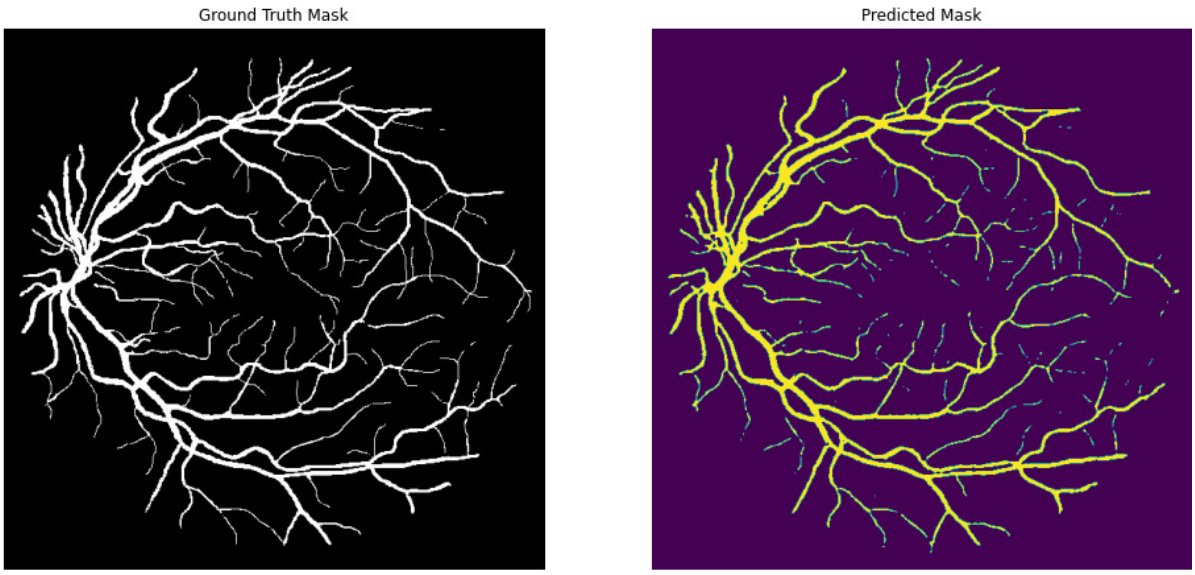


Figure 4.14: 5th sample of Retinal vessel segmentation and extraction

4.7.2 Accuracy Clarification

Our focus will be mainly on the clinical relevance of our model results, and the discussion extensively will be only mainly on the First Model operating on the DRIVE (Digital Retinal Images for Vessel Extraction) dataset, considering both models operate on the same basis and anatomical structures detection, we'll discuss the first model in length since it has more intense training to converge and more variance in the data. From the (Table 4.1) Our Retinal vessel segmentation of the DRIVE data set model achieved 96.55% accuracy, which tells us how accurate our predictions of classifying the right pixels are. while this is a general measure metric, it can be tricky and provide false confidence on the model, there is a catch, it is actually it is very easy for the segmentation model to predict the negative class pixels in the background and learn their patterns.

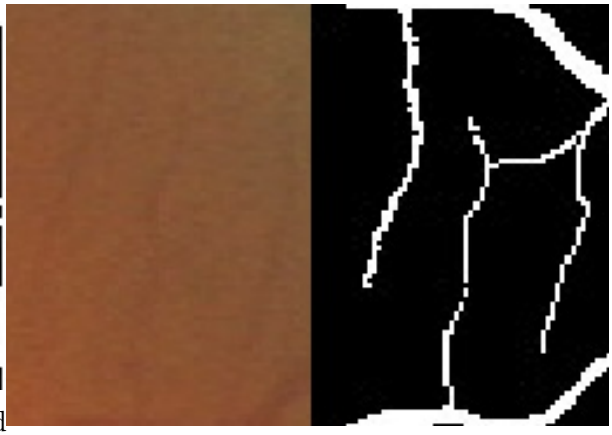
4.7.3 Recall And Precision

The important metric for the evaluation here is the Recall / Sensitivity, and which stands at 77.53%, it provides a good confidence on the model's ability to extract as many Retinal vessel pixels as possible, this is a semantic segmentation task, and is binary classification in specific, but we should also note that vessels are of many categories and sizes as we know anatomically, there are the main arteries and veins and there are other vessels that branch out from the bigger ones, the reason the recall value is brought down here is the models' inability to accurately identify the very tiny and thin blood vessels, the recall tells us given all the vessels presence in the image, how many can you accurately identify the presence of and segment?

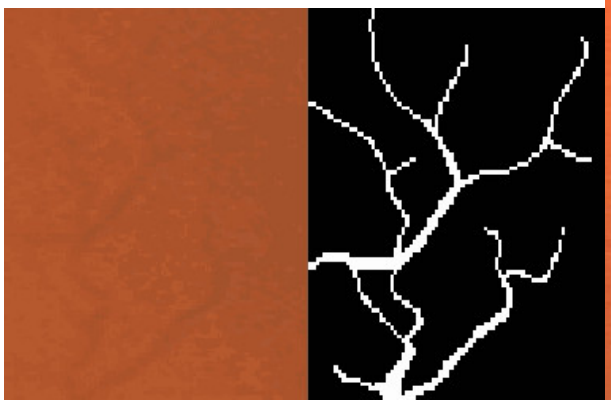
The pictures below will display and illustrate how the tiny vessels are represented in the majority of the training data of the model and the nature of funds images leading to this metric, and we can intuitively grasp an idea why the model is struggling at this task.



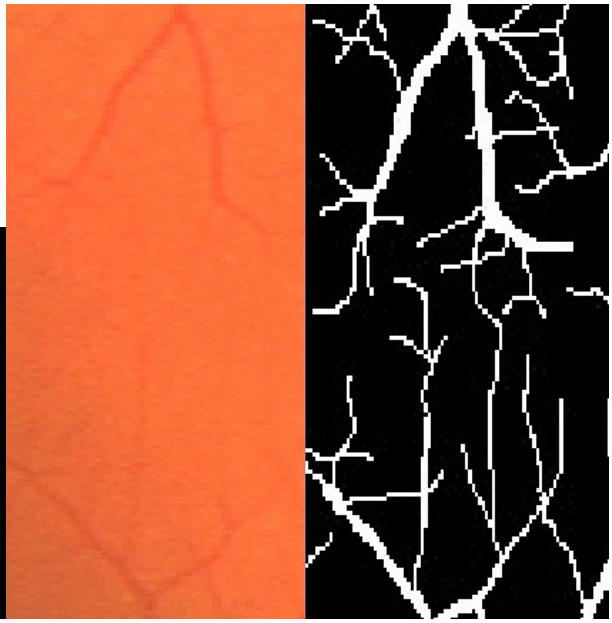
(a) First image of training data vessels and ground truth mask



(b) Second example



(a) 3rd example



(b) 4th example



(a) 5th example



(b) 6th example

The provided images illustrate why the model struggles with detecting tiny blood vessels. While the model effectively extracts and identifies primary, secondary, and 3rd vessels in the branching hierarchy, it consistently fails with extracting the full size of the smallest vessels. This issue is understandable given the images' characteristics.

The primary challenge lies in the minimal contrast between the tiny vessels and the background, making differentiation difficult for the model. Our current dataset, which lacks specific lighting and balanced pixel intensities, further complicates the task. High-quality images that highlight these small blood vessels are challenging to collect in sufficient quantities.

Convolutional Neural Networks (CNNs) struggle in such scenarios because they rely on local feature extraction. When the pixel intensity of tiny vessels closely matches the background, the CNN's convolutional filters may fail to detect subtle differences. This is particularly problematic for smaller, less prominent features, as the CNN might not generate sufficient feature maps to accurately differentiate these vessels from the background. The lack of contrast and insufficient training data impede the model's ability to learn and generalize the detection of these tiny structures.

4.7.4 Precision

The precision is at 82.19%, the remaining percentage until 100% means the model is falsely classifying background as positive Retinal vessel pixel presence. We can attribute this to the pixel, similarly to the very tiny vessels and the background, often the vessels are nearly visible due to the noise present in the dataset. False positives here are often just tiny positive dots that are very detectable and obvious as false positives visually to the eye of an experienced ophthalmologist, easily discarded and often irrelevant considering Ophthalmologists work with the larger vessels at the higher branches.

4.7.5 Specificity

A specificity of 98.39% indicates that the model is highly effective in correctly identifying background pixels, thereby minimizing false positives where non-vessel pixels are erroneously segmented as retinal blood vessels. This elevated specificity can be attributed to the model's proficiency in avoiding the misclassification of background as blood vessels, with a particular emphasis on the accurate segmentation of larger primary and secondary blood vessels. These vessels occupy a substantial portion of the image, facilitating their correct identification by the model.

The model demonstrates near-perfect accuracy in segmenting these larger vessels, ensuring reliable detection of primary arteries and veins. Nevertheless, the residual gap in specificity arises from the model's occasional misclassification of background pixels as retinal blood vessels, the tiny blood vessels in specific. These smaller vessels present a greater challenge for detection and are sometimes misclassified due to their subtle features and varying orientations in the retinal fundus images.

4.7.6 DICE

The DICE coefficient, measuring 79.61% in our model, can be pivotal in evaluating the accuracy of pixel classification for Retinal vessel segmentation. This metric is intuitive and explains how it quantifies the overlap between the model's predicted segmentation and the ground truth, offering insight into how well the model delineates vessel boundaries and excludes. A higher DICE score signifies a stronger agreement between predicted and actual segmentation, indicating superior pixel-level classification performance.

4.7.7 AUC

The AUC (Area Under the ROC Curve) serves as a measure for assessing the performance of binary classification models, providing a comprehensive evaluation of their ability to differentiate between classes. In our context of Retinal vessel segmentation, achieving an AUC of 97.07% signifies. This metric represents the probability that the model will rank a randomly chosen positive instance (Retinal vessel pixel) higher than a randomly chosen negative instance (background pixel). In clinical applications, a high AUC indicates that the model effectively detects Retinal vessels with high sensitivity while maintaining a low false positive rate.

From a statistical perspective, an AUC of 97.07% indicates that the model is doing well separates Retinal vessels from background pixels and does so with high certainty and not by chance, minimizing the likelihood of false classifications. By adjusting classification thresholds, the model optimizes its predictions to approach values closer to 1 for positive instances (Retinal vessels) and closer to 0 for negative instances (background). This adjustment minimizes the chance of misclassification and enhances the model's reliability in clinical practice.

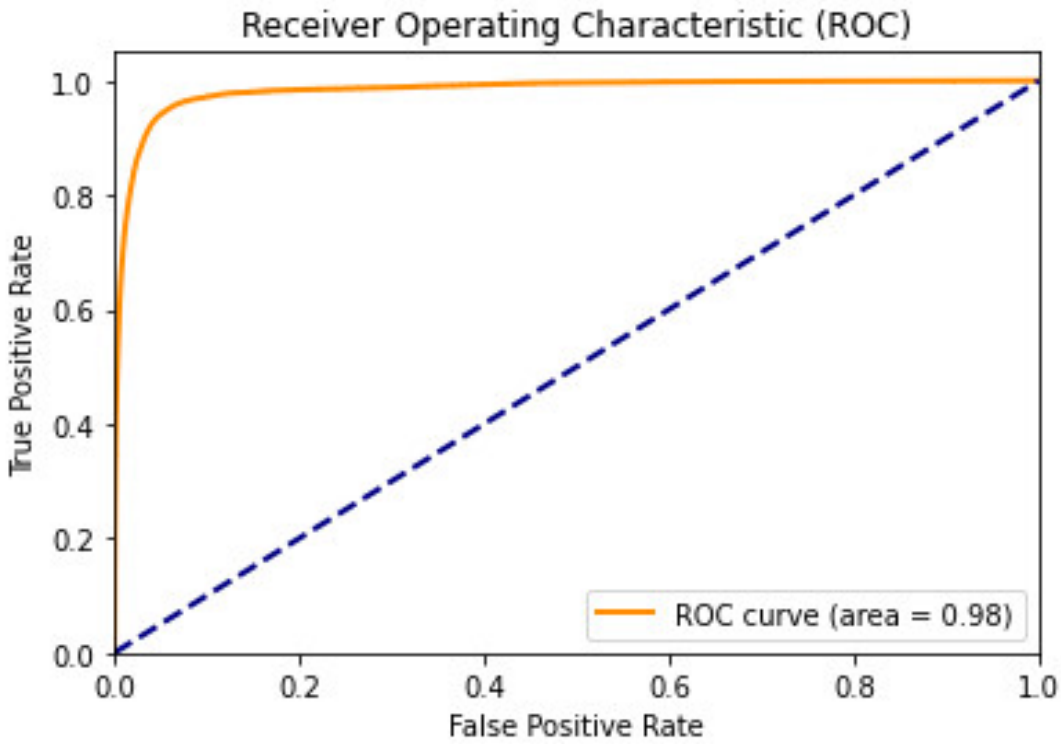


Figure 4.18: A graph showing the ROC curve and AUC of the first model

The graph on (Figure 4.18) displays the various thresholds the model will still manage to classify the positive classes, a perfect performing model will have the curve as close as possible to the coordinate of 1 Y and 0 Axis. Our model employs a default classification threshold of 0.5, which is an optimal balance between sensitivity and specificity for Retinal vessel segmentation. However, it is worth noting that even if we were to lower the threshold, the model would still maintain its ability to detect the presence of positive pixels, especially the first and secondary, 3rd vessels in the hierarchy. This adaptability highlights the robustness of our model in accurately identifying Retinal vessels across various threshold settings, further emphasizing its potential as a valuable clinical tool for diagnosing and monitoring Retinal diseases.

4.8 Results And Discussion Of The Second Model (Optic Nerve Centered)

Metric	Value (%)
Accuracy	96.83
Precision	82.27
Recall (Sensitivity)	75.40
Dice Coefficient	78.68
Specificity	98.63
Area Under the Curve (AUC)	95.35

Table 4.2: Performance metrics for the optic nerve-centered dataset.

Considering we have already explained and interpreted where the metrics are stemming from in the previous model, this time we'll briefly overview them considering they work on the same principle and equivalent anatomical structure, just except this model has less small blood vessels showing. First in the figures below we'll see examples of how it managed to extract blood vessels from the unseen data, and then we'll discuss the metrics achieved and displayed on the (Table 4.2).

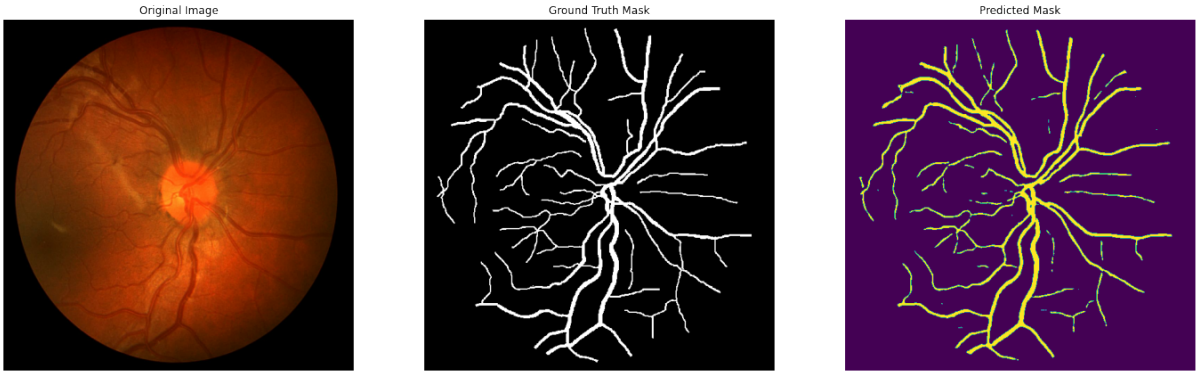


Figure 4.19: First example of the segmentation

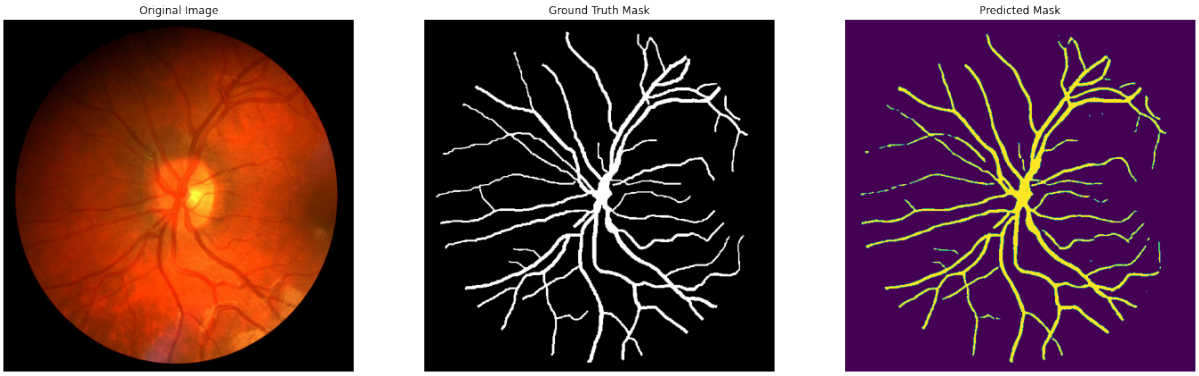


Figure 4.20: Second example of the segmentation

From the (Table 4.2) we notice our model achieved similar results to the first model 96.83% in accuracy of predictions which isn't enough evaluation for the task at hand, Precision of positive predictions at 82.27%, which means how precise were the pixels the model correctly identified as the presence of pixels, it is due the same reason of false positives caused by the similarity of the background with the small blood vessels in the representation, Recall in the model was at 75.40% the model again managed to capture the primary vessels almost completely successfully which is the most clinically important and missed on segments of the tiny blood vessels, The DICE coefficient of 78.68% indicating the overlap of the ground truth and predict mask. Specificity similarly at 98.63%, indicating how most of the background is classified correctly, minimizing the false negatives. AUC ROC score this time came at 95.14%.which represents to us the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at different decision thresholds.

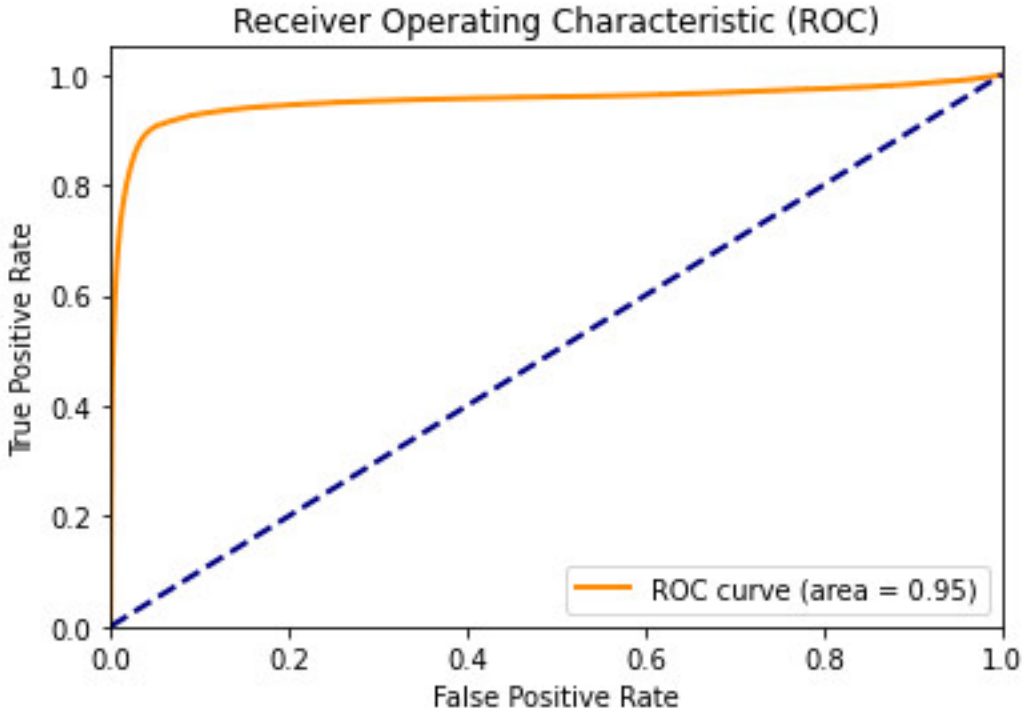


Figure 4.21: AUC ROC Curve of the Second Model

The model the same way as the first, even if it fails segmenting the vessels, it has shown the same pattern of displaying the tiny blood vessels cropped in the model which to the eye of trained Retinal specialists can be an enough segmentation, as they can tell easily how the rest will fill out on a perfect scenario, the same thing for false positives, a specialist in his analysis can tell the false positives and ignores them. And objectively this data has issues, considering most of the images have a lot of noise and are lacking good contrast between the blood vessels and the background, The figures below can show as an example.

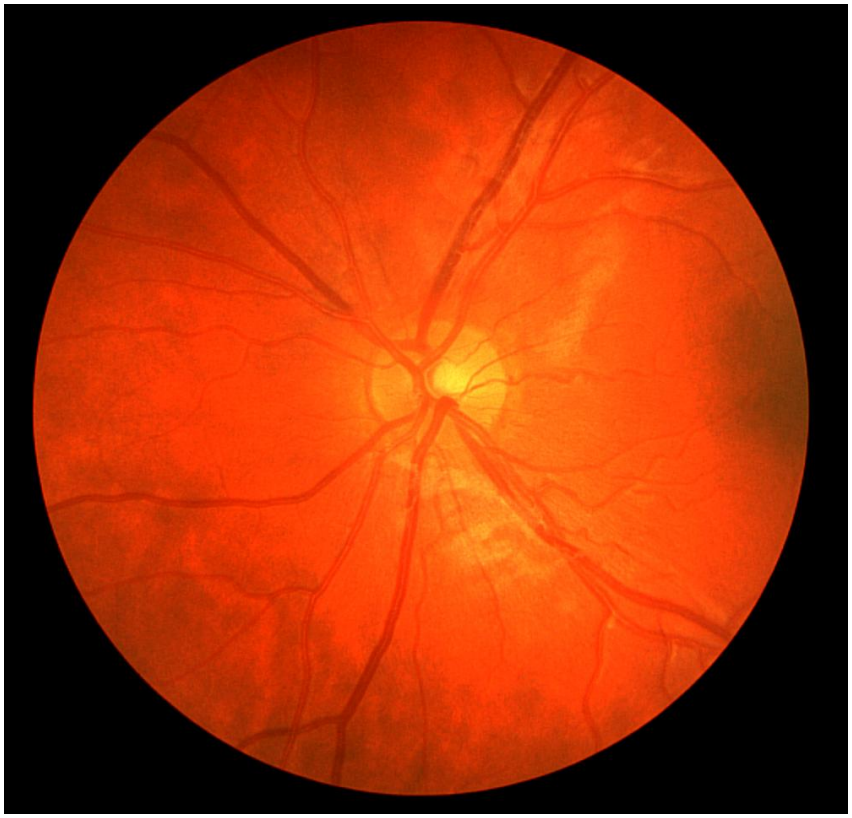


Figure 4.22: First example of training data of optic disc centered dataset

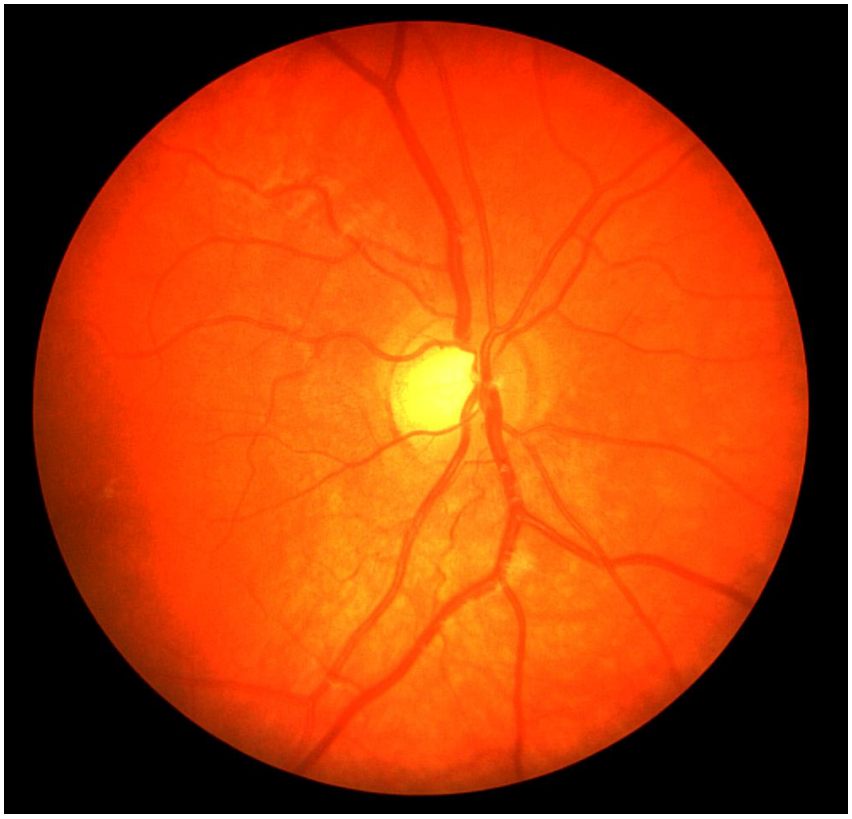


Figure 4.23: Second example of training data of optic disc centered dataset



Figure 4.24: 3rd example of training data of optic disc centered dataset

And in the Figures below are remaining results of the Retinal vessel segmentation in the unseen data.

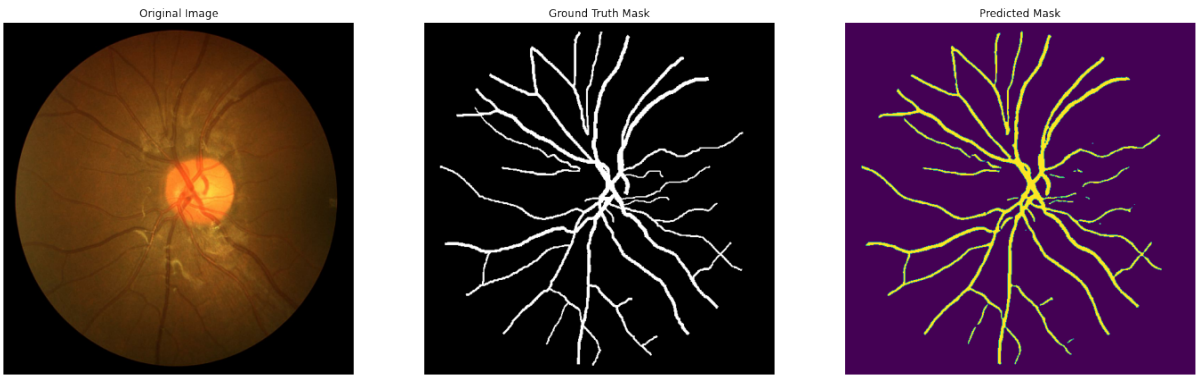


Figure 4.25: 3rd sample of optic disc centered segmentation

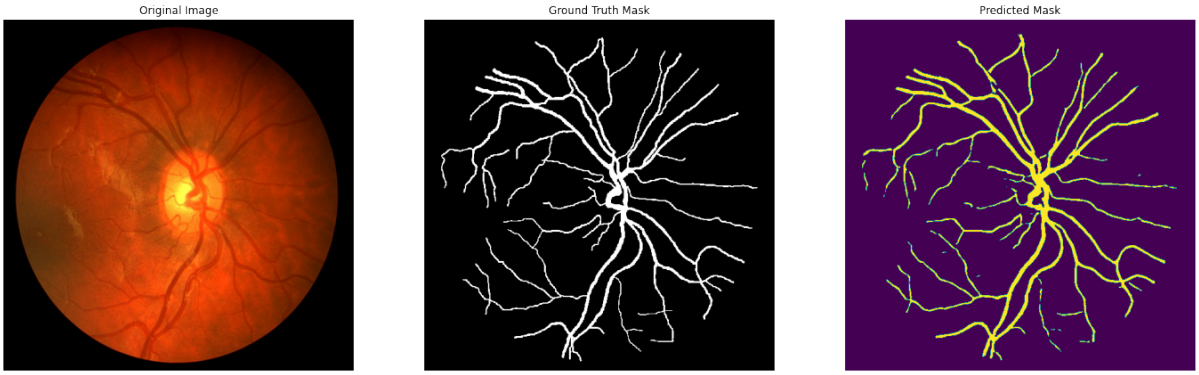


Figure 4.26: 4th sample of optic disc centered segmentation

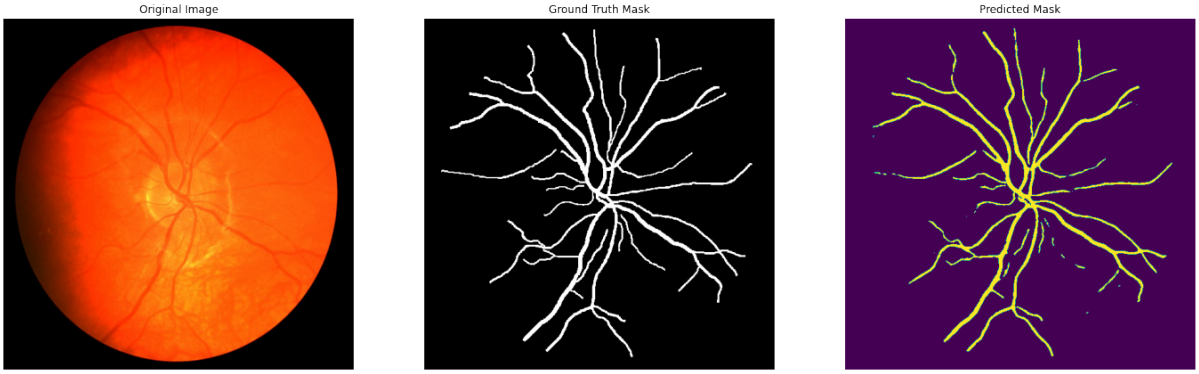


Figure 4.27: 5th sample of optic disc centered segmentation

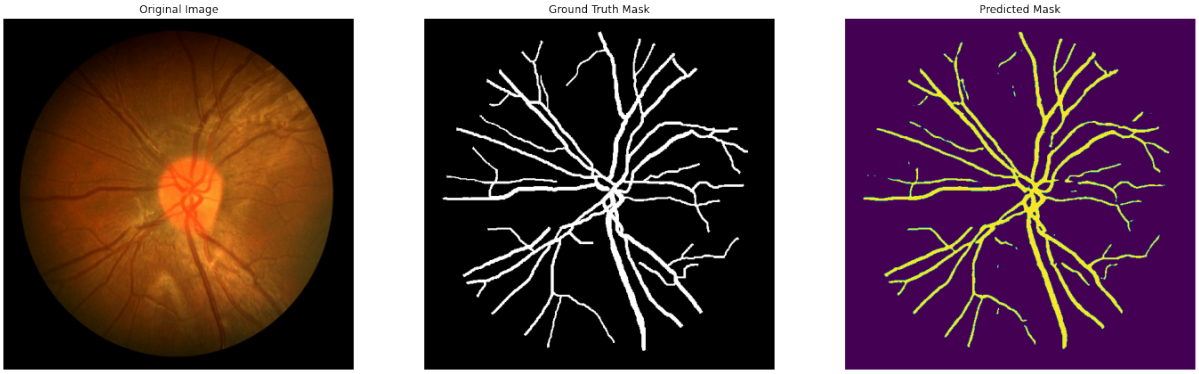


Figure 4.28: 6th sample of optic disc centered segmentation

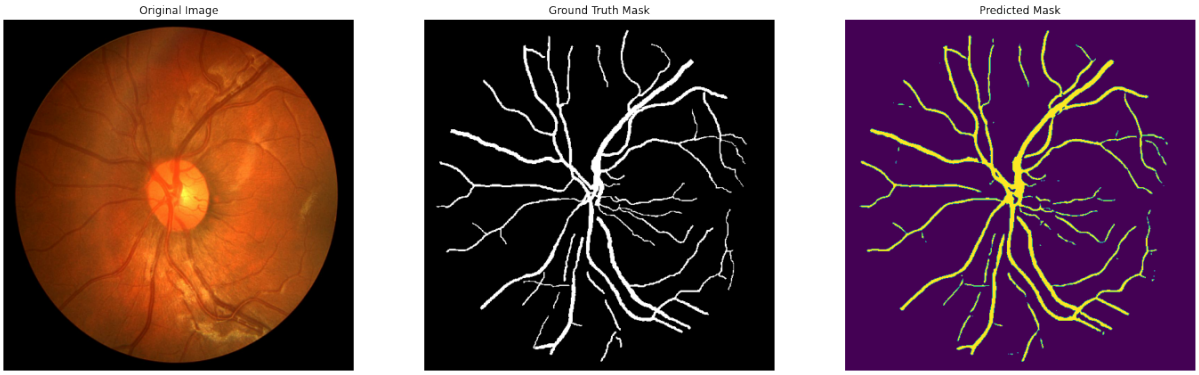


Figure 4.29: 7th sample of optic disc centered segmentation

4.9 Training phase charts of both the models

Below in both the figures we'll see the training charts of the models, and then comment on them.

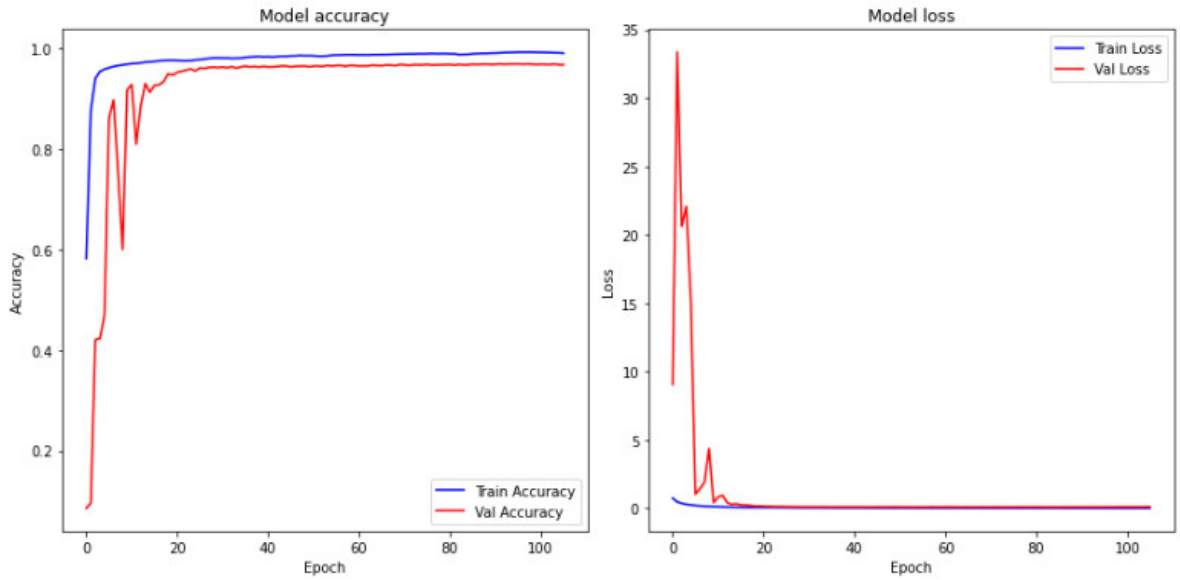


Figure 4.30: First model training metrics chart during the first iteration

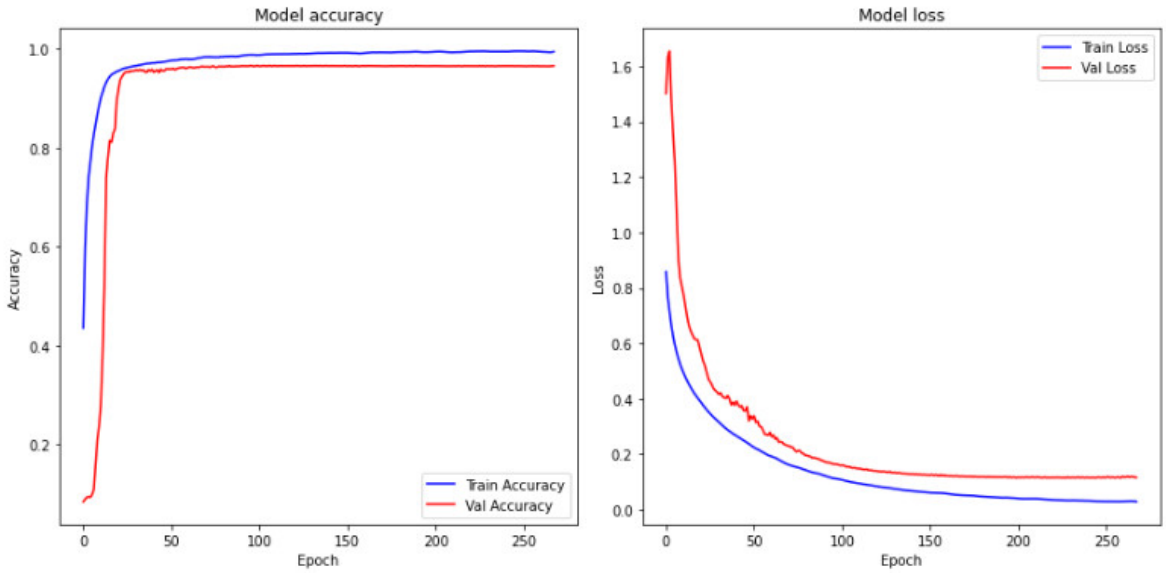


Figure 4.31: Second model training metrics chart during the first iteration

These two graphs in (Figure 4.30) and Figure (4.31) illustrate the training histories of both models. The x-axis represents the number of epochs, which are iterations over the entire training dataset. The y-axis denotes the loss value, a measure of how well the model's predictions align with the actual outcomes, or accuracy, which indicates the proportion of correctly predicted instances.In the first graph (Figure 4.30), we observe the model's performance with a higher learning rate, initially, during the early epoch iterations, the model struggles to converge due to the aggressive learning rate causing instability. Conversely, in the second graph in (Figure 4.31), the model is trained on a simpler dataset with less variance. This results in a smoother and quicker convergence, as depicted by a rapid decrease in the loss value or a swift increase in accuracy. The comparison highlights that the fundus images with a Macula-Centered focus in the first model contain significantly more variance, necessitating extensive optimization and gradient adjustments to achieve stability. In contrast, the second model, dealing with a less complex dataset, finds it easier to stabilize and converge.

4.9.1 Stabilization of the learning curve

This in the (Figure 4.32) demonstrates the model's loss curve during the final iterations of training with a decreased learning rate at 1×10^{-8}, suggesting that the model's adjustments to its loss function are slowly progressing. This indicates that the model is approaching a state of near stability, at which point it can no longer extract additional useful features from the training data to effectively handle unseen data.

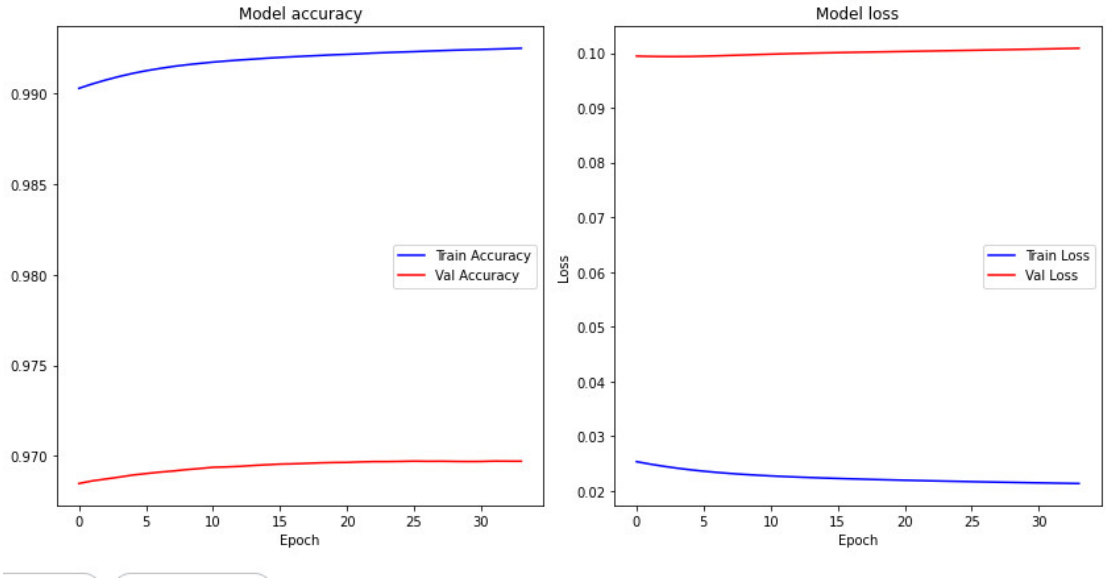


Figure 4.32: Loss chart of the secondary iterations with a lower learning rate

Conclusion

This chapter has presented a comprehensive overview of our work aiming for the segmentation and extraction of Retinal blood vessels from fundus images. The importance of accurate vessel segmentation lies in its ability to aid in the diagnosis and monitoring of various eye diseases, such as Diabetic Retinopathy, and other diseases that affect the vascular layer.

Accurate Retinal vessel segmentation, especially for the central Retinal veins and arteries and the next ones that branches out from them, plays a crucial role in diagnosing and monitoring a wide range of ocular and systemic conditions. It aids in the detection of diseases such as Retinal vein occlusion (RVO), glaucoma, Retinal hypertension, and various inflammatory disorders. Moreover, it facilitates the tracking of disease progression over time, enabling more precise quantitative analysis. By clearly delineating vessel structures, it allows for easier identification of vessel dilations and constrictions, which are key indicators of vascular health and disease states Two distinct datasets, DRIVE and CHASEDB1, were utilized to train and evaluate separate models specified to the specific fundus photography view angles employed during image acquisition. data preprocessing steps, including resizing, normalization, were undertaken to optimize the image data for model training. The model architecture used DenseNet169 as the backbone and the U-Net decoder, demonstrating improved convergence compared to other architectures after testing. Binary cross-entropy was employed as the loss function, aligning with the binary classification nature of the segmentation task. The training process involved careful monitoring of the loss metric and learning rate adjustments to ensure optimal model performance. Thorough evaluations were conducted using various metrics, including accuracy, precision, recall, specificity, Dice coefficient, and Area Under the Curve (AUC). The models achieved promising results with the primary blood vessels of larger sizes, However, the analysis revealed that the models faced challenges in accurately detecting and segmenting the smallest blood vessels due to their minimal contrast with the background and the inherent limitations of the datasets. This limitation highlights the need for further data collection and the acquisition of higher-quality datasets to improve

the models' ability to segment even the tiny Retinal vessels accurately. Despite these constraints, the developed models demonstrated robust performance in segmenting the primary and secondary blood vessels, which are crucially important for clinical assessment and diagnosis. The high specificity and AUC values suggest that the models can effectively differentiate Retinal vessels from the background. Overall, this chapter explored Retinal vessel segmentation techniques, offering insights into the strengths and limitations of current approaches. These remarks can pave the way for future research aimed at enhancing model performance, particularly in the segmentation of tiny vessels.

General Conclusion

Early diagnosis and proper management of Diabetic Retinopathy are essential for preserving vision and improving patient outcomes. In this work, we describe the development and evaluation of a convolutional neural network (CNN) model that can effectively detect and assess the severity of Diabetic Retinopathy using retinal fundus images.

Utilizing retinal fundus images, we have implemented an effective system for diagnosing Diabetic Retinopathy and grading its severity. The model employed Convolutional Neural Networks (CNNs), achieving relevant performance despite limitations in data and computational resources. Prior to training and inputting images into the CNN models, we carefully preprocessed the data to ensure the correct format to input into the neural network for training. Data augmentation techniques were applied to enhance model generalizability and improve lesion detection across various locations and sizes. Among the four pretrained models evaluated (DenseNet169, ResNet101, VGG19, Inception V3), DenseNet169 demonstrated the highest performance and diagnostic confidence. It effectively minimized false negatives and exhibited high sensitivity in classifying the early stages of Diabetic Retinopathy. This work underscores the efficacy of CNNs in leveraging retinal fundus images for accurate diagnosis and severity assessment of Diabetic Retinopathy, highlighting the importance of preprocessing techniques and model selection in enhancing diagnostic outcomes.

Our model can be of great aid in the early detection and treatment of Diabetic Retinopathy, enabling timely intervention to slow the progression of the disease. By precisely identifying subtle changes in retinal fundus images, it supports early diagnosis and prompt medical intervention, reducing the likelihood of vision loss by up to 90%. The model's ability to accurately classify with good confidence the severity of Diabetic Retinopathy into different stages, such as mild, moderate, and severe, is particularly valuable for guiding appropriate treatment interventions.

Furthermore, our segmentation model enhances the analysis of retinal vascular diseases, offering detailed insights into disease progression and severity. This supports informed clinical decision-making and optimizes patient care strategies. Future enhancements in our system in future work could address the interpret-ability issues inherent in Deep Learning models, often characterized by the "black box" problem—wherein decisions lack transparency. This could be mitigated by integrating attention mechanisms or CAM (Classification Activation Maps) that interpret the predictions and highlight regions of interest pivotal for classification thresholds between different stages of Diabetic Retinopathy. This approach enhances not only diagnostic accuracy but also provides insights into the rationale behind model decisions. The (Figure 4.33) shows what we hope to achieve.

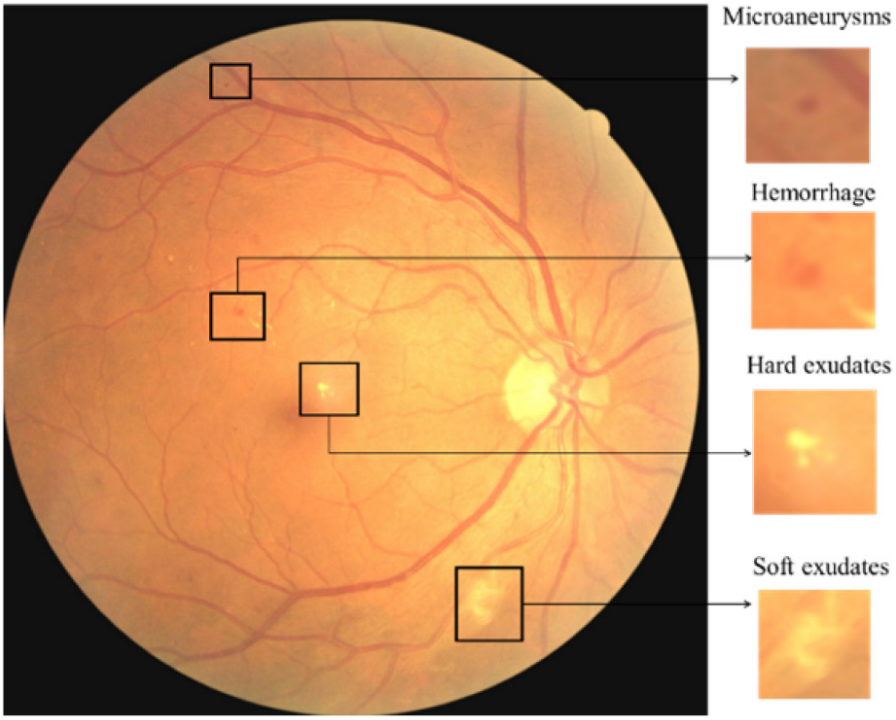


Figure 4.33: Concept of Diabetic Retinopathy grading model that detects the presence of lesions, shows the location in the image and interprets the findings

Moreover, expanding our dataset and computational resources would allow us to further refine and generalize our model. By training on diverse datasets, we can improve robustness and adaptability across varied clinical scenarios. Additionally, integrating and training additional models for lesion detection within the same framework could provide comprehensive insights into both disease diagnosis and lesion characterization, including their spatial boundaries.

Ensemble learning presents another avenue for enhancing model performance. This technique involves training multiple neural networks on the same data and aggregating their outputs to improve predictive accuracy and reliability. By leveraging ensemble methods, our system could achieve higher confidence in predictions and resilience to noise or variability in input data.

Looking forward, we hope to be able to extend our system to encompass a multi-disease classification framework. Beyond Diabetic Retinopathy, our goal is to develop a comprehensive system capable of detecting and classifying other prevalent ophthalmological conditions such as cataracts, age-related macular degeneration, glaucoma, and macular edema. This approach aims to provide clinicians with a unified platform for early detection, precise diagnosis, and tailored treatment strategies across a spectrum of eye diseases. When inferred for the diagnostic aid The model has a higher level of uncertainty. for the miss-classified images, which is clear from the probabilistic result of the classifier indicating how it distinguishes between the classes when observing an input image, meaning it can be used as an indicator to drive referral and more examination of the retina of the patient. A false negative result may provide a pseudo sense of security about the Retinopathy status. For the present, a comprehensive dilated eye ex-

amination by a well-trained and experienced ophthalmologist remains the gold standard of screening and cannot be replaced with this device until appropriately proven otherwise.

The segmentation system managed to successfully extract fully the primary and secondary to the third vessels in the hierarchy and struggled with the full identification of tiny blood vessels, but still managed to draw an abstract hint of the path they're taking and their diameter, in clinical practice the primary vessels investigated for the identification of vascular problems are the primary ones for quantitative and visual investigation.

The model can be proved with more dataset of high quality and lightning conditions, minimizing the noise as much as possible. Also, another area of improvement could be training the model for more sophisticated instance segmentation, by training the model on different pixel intensities between the arteries and veins, veins as blue and arteries as red, the concept of this is shown in the (Figure 4.34) below.

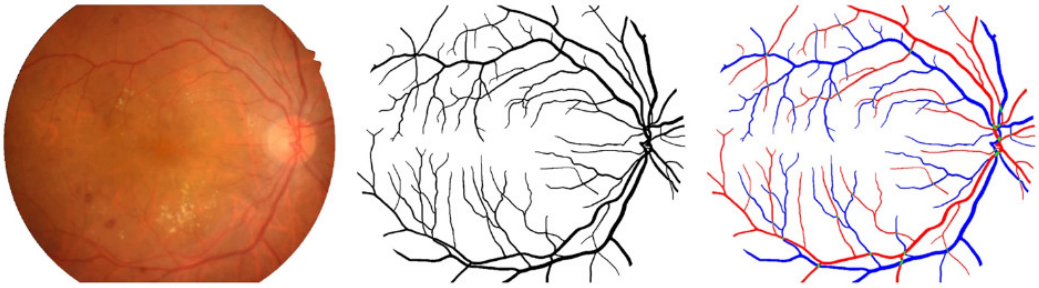


Figure 4.34: Concept of instance segmentation of retinal blood vessels, specifying which are the veins and which are the arteries

This system exhibits high flexibility and adaptability, offering multiple integration methods into various computing environments. A primary approach involves localizing the model and configuring the input data preprocessing pipeline according to model expectations. It supports integration across diverse platforms capable of executing Python code, including embedded systems, mobile devices, ophthalmic equipment, and computers.

Deploying the model can vary in complexity, making cloud platforms like Google Cloud and AWS a recommended choice. Utilizing these platforms allows seamless deployment via APIs, leveraging predefined functions on the host machine. This streamlined approach minimizes the need for software engineers to extensively review model documentation to comprehend preprocessing requirements, thus simplifying implementation and maintenance processes.

We expect the clinical staff of ophthalmology to carefully capture the fundus images under appropriate lighting conditions and contrast, and we can even experiment with different model settings. That often yields the best model results.

The contents of this thesis can provide assistance to future students who are considering working in a similar field or area. The thesis presents well-researched and experimented-with methodologies and approaches for preprocessing and training the models, which are cited throughout the chapters.

Interdisciplinary collaboration between machine learning researchers, ophthalmolo-

gists, and healthcare providers is essential for getting the most out of computer-aided diagnostic systems in the field of ophthalmology. By combining expertise in ML with clinical insights and patient care needs, collaborative efforts can ensure the development of robust diagnostic tools that enhance accuracy and accessibility. This partnership fosters innovation, bridging the gap between technology and clinical practice to improve outcomes for patients with Diabetic Retinopathy.

This work and experience has been transformative, allowing us to go further into the intricacies of neural network training and hyperparameter optimization appropriate to our specific tasks. We've augmented our research skills and developed a robust methodology through this project, learning to organize our work efficiently and embrace an iterative, collaborative approach. This project pushed us to explore the boundaries of our capabilities in machine learning, particularly in its application to biomedical image analysis using deep learning techniques.

We feel privileged to have worked on a topic with such potential for real-world impact, hoping that our efforts will contribute to improving people's lives and their eye health. The process has been incredibly rewarding, and we're grateful for the supportive environment and guidance provided by our teachers throughout this journey.

This experience has undoubtedly shaped us, enhancing our skills and broadening our perspectives. We eagerly anticipate future opportunities to engage in similar projects within this nurturing academic setting. The lessons learned and memories formed during this time will remain valuable to us, serving as a foundation for our future endeavors in the field.

Bibliography

[1] Mujeeb Z et al Banday. Pathophysiology of diabetes: An overview. *Avicenna journal of medicine*, 10(4):174–188.

[2] Joanne W.Y. Yau, for the Meta-Analysis for Eye Disease (META-EYE) Study Group, and Others. Global prevalence and major risk factors of diabetic retinopathy. *Diabetes Care*, 35(3):556–564, 2012.

[3] Tien-En Tan and Tien Yin Wong. Diabetic retinopathy: Looking forward to 2030. *Frontiers in Endocrinology*, pages 3–5, 2012.

[4] M Kropp, O Golubnitschaja, A Mazurakova, and et al. Diabetic retinopathy as the leading cause of blindness and early predictor of cascading complications-risks and mitigation. *EPMA Journal*, 14(1):21–42, 2023.

[5] S. Gupta and A. R. Thool. A narrative review of retinopathy in diabetic patients. *Cureus*, 16(1), 2024.

[6] M Kropp, O Golubnitschaja, A Mazurakova, and et al. Diabetic retinopathy as the leading cause of blindness and early predictor of cascading complications-risks and mitigation. *EPMA Journal*, 14(1):21–42, 2023.

[7] Zhen Ling et Al Teo. Global prevalence of diabetic retinopathy and projection of burden through 2045: Systematic review and meta-analysis. *Ophthalmology*, 128(11):3–5, 2021.

[8] U. V. Shukla and K. Tripathy. Diabetic retinopathy. 2024. [Updated 2023 Aug 25].

[9] Wilkinson-Berka JL and Miller AG. Update on the treatment of diabetic retinopathy. *Scientific World Journal*, 8:98–120, 2008.

[10] Moutray T, Evans JR, Lois N, Armstrong DJ, Peto T, and Azuara-Blanco A. Different lasers and techniques for proliferative diabetic retinopathy. *Cochrane Database of Systematic Reviews*, 3(3), 2018.

[11] V Gupta and JF Arevalo. Surgical management of diabetic retinopathy. *Middle East African Journal of Ophthalmology*, 20(4):283–292, 2013.

[12] Faizan Fazal et al. The rising cost of healthcare and its contribution to the worsening disease burden in developing countries. *Annals of Medicine and Surgery*, 82:104683, 2022.

[13] S Dang, H Pakchanian, E Flynn, and R Raiker. Estimating patient demand for ophthalmologists in the united states using google trends. *Investigative Ophthalmology and Visual Science*, 2021.

[14] Zhen Ling Teo, Yih-Chung Tham, M Yu, et al. Global prevalence of diabetic retinopathy and projection of burden through 2045: Systematic review and meta-analysis. *Ophthalmology*, 128(11):1580–1591, 2021.

[15] H Sun, P Saeedi, S Karuranga, and et al. Idf diabetes atlas: Global, regional and country-level diabetes prevalence estimates for 2021 and projections for 2045. *Diabetes Research and Clinical Practice*, pages 109–119, 2022.

[16] NEM Khalifa, M Loey, MHN Taha, and HNET Mohamed. Deep transfer learning models for medical diabetic retinopathy detection. *Acta Informatica Medica*, 27(5):6, 2019.

[17] S.R. Flaxman and et al. Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis. *The Lancet Global Health*, 5(12), 2017.

[18] SE Mansour, DJ Browning, K Wong, HW Jr Flynn, and AR Bhavsar. The evolving treatment of diabetic retinopathy. *Clinical Ophthalmology*, 14:656, 2020.

[19] H. Schneiderman. The funduscopic examination. In H.K. Walker, W.D. Hall, and J.W. Hurst, editors, *Clinical Methods: The History, Physical, and Laboratory Examinations*, chapter 117. Butterworths, Boston, 3rd edition, 1990.

[20] Huma Naz, Neelu J.Ahuja, and Rahul Nijhawan. Clinical utility of handheld fundus and smartphone-based camera for monitoring diabetic retinal diseases: a review study.

[21] Johns Hopkins Medicine. Leading the ai ophthalmology revolution. `https://clinicalconnection.hopkinsmedicine.org/news/leading-the-ai-ophthalmology-revolution`, 2023.

[22] Borys Tymchenko, Philip Marchenko, and Dmitry Spodarets. Deep learning approach to diabetic retinopathy detection. 03 2020.

[23] A. Sami Memon et al. Role of optical coherence tomography angiography to differentiate intraretinal microvascular abnormalities and retinal neovascularization in diabetic retinopathy. *Pakistan journal of medical sciences*, 38(1):57–61, 2022.

[24] YC Chung, T Xu, TH Tung, M Chen, and PE Chen. Early screening for diabetic retinopathy in newly diagnosed type 2 diabetes and its effectiveness in terms of morbidity and clinical treatment: A nationwide population-based cohort. *Frontiers in Public Health*, 10, 2022.

[25] K Doi. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computers and Graphics*, 31(4-5):198–211, 2007.

[26] Nathan Silberman, Kristy Ahlrich, Rob Fergus, and Lakshminarayanan Subramanian. Case for automated detection of diabetic retinopathy. In *Artificial Intelligence for Development - Papers from the AAAI Spring Symposium, Technical Report*, AAAI Spring Symposium - Technical Report, pages 85–90. AI Access Foundation, 2010.

[27] Z Yang, TE Tan, Y Shao, TY Wong, and X Li. Classification of diabetic retinopathy: Past, present and future. *Frontiers in Endocrinology (Lausanne)*, 13:2, 2022.

[28] A Family Optician. Diabetic retinopathy. `https://afamilyoptician.co.uk/diabetic-retinopathy/`.

[29] D An, B Tan, DY Yu, and C Balaratnasingam. Differentiating microaneurysm pathophysiology in diabetic retinopathy through objective analysis of capillary nonperfusion, inflammation, and pericytes. *Diabetes*, 71(4):733–746, 2022.

[30] R Singh, K Ramasamy, C Abraham, V Gupta, and A Gupta. Diabetic retinopathy: an update. *Indian Journal of Ophthalmology*, 56(3):201, 2008.

[31] P.A. Campochiaro. Ocular neovascularization. *Journal of Molecular Medicine (Berlin)*, 91(3):311–321, 2013.

[32] Mohammed Alshahrani et al. Hybrid methods for fundus image analysis for diagnosis of diabetic retinopathy development stages based on fusion features. `https://www.mdpi.com/2075-4418/13/17/2783`, 2023.

[33] Vimla L Patel, Edward H Shortliffe, Mario Stefanelli, et al. The coming of age of artificial intelligence in medicine. *Artificial Intelligence in Medicine*, 46(1):1–5, 2009.

[34] K. Doi. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computers, Medical Imaging and Graphics*, 31:198–211, 2007.

[35] K. Doi. Diagnostic imaging over the last 50 years: Research and development in medical imaging science and technology. *Physics in Medicine and Biology*, 51:R5–27, 2006.

[36] Riccardo Miotto, Fei Wang, Shwetak Wang, Xiaoqian Jiang, and Jared T. Dudley. Deep learning for healthcare: Review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6):1236–1246, 2017.

[37] Muhammad Imran Razzak, Syed Naz, and Arslan Zaib. Deep learning for medical image processing: Overview, challenges and the future. In *Classification in BioApps*, pages 323–350. Springer, Cham, 2018.

[38] Valentina Et Al Bellemo. Artificial intelligence using deep learning in classifying side of the eyes and width of field using retinal fundus photographs. 12 2018.

[39] Niranjan R. Panda and Ashok Kumar Sahoo. A detailed systematic review on retinal image segmentation methods. *Journal of Digital Imaging*, 35(5):1254, 2022.

[40] Jair Cervantes, Jared Cervantes, Farid García-Lamont, Arturo Yee-Rendon, Josué Espejel Cabrera, and Laura Domínguez Jalili. A comprehensive survey on segmentation techniques for retinal vessel segmentation. *Neurocomputing*, 556:130, 2023.

[41] P. Modi and T. Arsiwalla. Hypertensive retinopathy. *StatPearls [Internet]*, Jan 2024. Updated 2023.

[42] Francesco Romano, Francesca Lamanna, Pierre-Henry Gabrielle, Kelvin Teo, Maurizio Parodi, Pierluigi Iacono, Samantha Fraser-Bell, Elisa Cornish, Marco Nassisi, Francesco Viola, Aniruddha Agarwal, Anindya Samanta, Jay Chhablani, Giovanni Staurenghi, and Alessandro Invernizzi. Update on retinal vein occlusion. *Asia-Pacific journal of ophthalmology (Philadelphia, Pa.)*, Publish Ahead of Print, 02 2023.

[43] Wanshu Zhou and Bernhard A Sabel. Vascular dysregulation in glaucoma: retinal vasoconstriction and normal neurovascular coupling in altitudinal visual field defects. *The EPMA journal*, 14(1):87–99, Feb 2023.

[44] A Duplechain, CD Conrady, BC Patel, et al. Uveitis. *StatPearls [Internet]*, Jan 2024. Updated 2023 Aug 8.

[45] K.S. Kothare and K. Malpe. Design and implementation of inspection model for knowledge patterns classification in diabetic retinal images. 2019.

[46] M.A. Hussain, S.O.B. Islam, M. Tiwana, and W. Qureshi. Detection and classification of hard exudates with fundus images complements and neural networks. 2019.

[47] A.M. Fayyaz, M.I. Sharif, S. Azam, A. Karim, and J. El-Den. Analysis of diabetic retinopathy (dr) based on the deep learning. *Information*, 14, 2023.

[48] N. Eftekhari, H.-R. Pourreza, M. Masoudi, K. Ghiasi-Shirazi, and E. Saeedi. Microaneurysm detection in fundus images using a two-step convolutional neural network. *Biomedical Engineering Online*, 18(1):67, 2019.

[49] N. Silberman, K. Ahrlich, R. Fergus, and L. Subramanian. Case for automated detection of diabetic retinopathy. 2010.

[50] OF Gurcan, OF Beyca, and O. Dogan. A comprehensive study of machine learning methods on diabetic retinopathy classification. *International Journal of Computational Intelligence Systems*, 14(2), 2021.

[51] CY et al. Tsai. Necessity of local modification for deep learning algorithms to predict diabetic retinopathy. *International Journal of Environmental Research and Public Health*, 19(3):1–12, 2022.

[52] D.J. Hemanth, O. Deperlioglu, and U. Kose. An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. *Neural Computing and Applications*, 32:707–721, 2020.

[53] Z. Qian, C. Wu, H. Chen, and M. Chen. Diabetic retinopathy grading using attention based convolution neural network. In *Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 2652–2655, Chongqing, China, March 12–14 2021.

[54] J. Lachure, A.V. Deorankar, S. Lachure, S. Gupta, and R. Jadhav. Diabetic retinopathy using morphological operations and machine learning. In *2015 IEEE International Advance Computing Conference (IACC)*, Bangalore, India, 2015.

[55] N.E.M. Khalifa, M. Loey, M.H.N. Taha, and H.N.E.T. Mohamed. Deep transfer learning models for medical diabetic retinopathy detection. *Acta Informatica Medica*, 27, 2019.

[56] Sehrish Qummar, Fiaz Gul Khan, Sajid Shah, Ahmad Khan, Shahaboddin Shamshirband, Zia Ur Rehman, Iftikhar Ahmed Khan, and Waqas Jadoon. A deep learning ensemble approach for diabetic retinopathy detection. *IEEE Access*, 7:150530–150539, 2019.

[57] Saeid Jafarzadeh Ghoushchi et al. An extended approach to predict retinopathy in diabetic patients using the genetic algorithm and fuzzy c-means. *BioMed research international*, 2021:5597222, 2021.

[58] L. Qiao, Y. Zhu, and H. Zhou. Diabetic retinopathy detection using prognosis of microaneurysm and early diagnosis system for non-proliferative diabetic retinopathy based on deep learning algorithms. *IEEE Access*, 8, 2020.

[59] S. Lian, L. Li, G. Lian, X. Xiao, Z. Luo, and S. Li. A global and local enhanced residual u-net for accurate retinal vessel segmentation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14, 2015.

[60] X. Xu, R. Wang, T. Tan, and F. Xu. An improved u-net architecture for simultaneous arteriole and venule segmentation in fundus image. *Medical Image Understanding and Analysis*, 2018.

[61] Amisha, P Malik, M Pathania, and V K Rathaur. Overview of artificial intelligence in medicine. *J Family Med Prim Care*, 8(7):2328–2331, 2019.

[62] T. L. Isenhour and J. C. Marshall. Expert systems and robotics. *Journal of Research of the National Bureau of Standards*, 93(3):209–212, 1988.

[63] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[64] Tom M Mitchell et al. *Machine Learning*, volume 45. McGraw Hill, Burr Ridge, IL, 1997.

[65] I.H. Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(420), 2021.

[66] Analytics Yogi. Ai – three different types of machine learning algorithms. `https://vitalflux.com/ai-types-machine-learning-algorithms`.

[67] Sedat Parlak. Foundation of neural networks. `https://medium.com/@sedatparlak1953/foundation-of-neural-networks-a68925aa7e2`, 2023.

[68] Aravind Reigns. Neural networks. `https://medium.com/@aravindreigns/neural-networks-d67f6dc24dd9`, 2019.

[69] MrBam44. The dying relu problem: Causes and solutions. `https://medium.com/@MrBam44/the-dying-relu-problem-causes-and-solutions-1a970f177c`, 2020.

[70] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. *Computer Science Department, Stanford University*, 2004.

[71] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[72] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. 2015.

[73] CS231n. Convolutional neural networks. `https://cs231n.github.io/convolutional-networks/`, 2024.

[74] Cyrill Ugod. Filtering and morphological operations. `https://cyrillugod.medium.com/filtering-and-morphological-operations-990662c5bd59`, 2020.

[75] S. T. Kebir, F. Berrhail, and F. Didi. An efficient tomato diseases detection and classification methodology using cnn deep learning network. *Brazilian Journal of Technology*, 7(2):e69316, Apr. 2024.

[76] Tomas Beuzen. Deep learning with pytorch: Convolutional neural networks (part 1). `https://www.tomasbeuzen.com/deep-learning-with-pytorch/chapters/chapter5_cnns-pt1.html`, 2020.

[77] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 2015.

[78] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2020.

[79] Bader Aldughayfiq, Farzeen Ashfaq, Noor Jhanjhi, and Mamoona Humayun. Yolo-based deep learning model for pressure ulcer detection and classification. *Healthcare*, 11:1222, 04 2023.

[80] Medischola. Funduscopy findings. `https://medischola.org/topics/fundoscopy-findings/`.

[81] University Of Iowa. Cotton wool spots. `https://webeye.ophth.uiowa.edu/dept/coms/grading/cotton-wool-spots.htm`.

[82] University of Iowa Health Care Ophthalmology and Visual Sciences. Hard exudates (lipid). `https://webeye.ophth.uiowa.edu/dept/coms/grading/hard-exudates.htm`.

[83] Richard Kirk. Diabetic retinopathy therapy innovations: An interview with richard kirk, ceo of polyphotonix. `https://www.news-medical.net/news/20151019/Diabetic-retinopathy-therapy-innovations-an-interview-with-Richard-Kirk-CEO-of-aspx`, 2015.

[84] GeekyMedics. Fundoscopic appearances of common retinal pathologies. `https://geekymedics.com/fundoscopic-appearances-of-retinal-pathologies/`.

[85] Cuadros J and Bresnick G. Eyepacs: an adaptable telemedicine system for diabetic retinopathy screening. *Journal of Diabetes Science and Technology*, 3(3):509–516, 2009.

[86] Purnawarman Musa, Farid Rafi, and Missa Lamsani. A review: Contrast-limited adaptive histogram equalization (clahe) methods to help the application of face recognition. 2018.

[87] Mira Hayati, Kahlil Muchtar, Roslidar, Novi Maulina, Irfan Syamsuddin, Gregorius Natanael Elwirehardja, and Bens Pardamean. Impact of clahe-based image enhancement for diabetic retinopathy classification through deep learning. *Procedia Computer Science*, 216:57–66, 2023. 7th International Conference on Computer Science and Computational Intelligence 2022.

[88] Oscar A. Montesinos López, Ana Montesinos López, and José Crossa. Overfitting, model tuning, and evaluation of prediction performance. *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, 2022.

[89] Z. Mungloo-Dilmohamud, M. Heenaye-Mamode Khan, K. Jhumka, B. N. Beedassy, N. Z. Mungloo, and C. Peña-Reyes. Balancing data through data augmentation improves the generality of transfer learning for diabetic retinopathy classification. *Appl. Sci.*, 12, 2022.

[90] S. Agarwal and A. Bhat. A survey on recent developments in diabetic retinopathy detection through integration of deep learning. *Multimed. Tools Appl.*, 82:17321–17351, 2022.

[91] Mary L. McHugh. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3):276–282, 2012.

[92] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[93] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[94] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[95] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[96] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks. pages 3320–3328, 2014.

[97] Giuseppe Pio Cannata. Vanishing gradient in deep neural networks, 2021.

[98] Frank Ursin et al. Diagnosing diabetic retinopathy with artificial intelligence: What information should be included to ensure ethical informed consent? *Frontiers in Medicine*, 8:695217, July 2021.

[99] Sai Manoj Yarlagadda. Understanding sensitivity, specificity, precision, recall, f1 score, and confusion matrix in binary classification, 2021.

[100] Chin Sheng Teoh et al. Variability in grading diabetic retinopathy using retinal photography and its comparison with an automated deep learning diabetic retinopathy screening software. *Healthcare (Basel, Switzerland)*, 11(12):1697, Jun 2023.

[101] AI Content Lab. Batch normalization (bn) and its effects. `https://www.ai-contentlab.com/2022/12/batch-normalization-bn-and-its-effects.html`, 2022.

[102] Hironobu Fujiyoshi, Tsubasa Hirakawa, and Takayoshi Yamashita. Deep learning-based image recognition for autonomous driving. *IATSS Research*, 43, 12 2019.

[103] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 06 2016.

[104] Folio3 AI. Semantic segmentation vs instance segmentation. `https://www.folio3.ai/blog/semantic-segmentation-vs-instance-segmentation/`, n.d.

[105] KDnuggets. Introduction to image segmentation with k-means clustering. `https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html`, 2019.

[106] Vivienne Sze. Energy-efficient machine learning for mobile and iot devices. `https://eems.mit.edu/wp-content/uploads/2019/04/2019_mars_sze-compressed.pdf`, 2019.

[107] Asmita Sinha. Yolact: Real-time instance segmentation. `https://medium.com/@asmitasinha/yolact-real-time-instance-segmentation-1-af0c27ee0bbc`, 2020.

[108] Nasser Tamim, Mohamed Elshrkawey, Gamil Abdel Azim, and Hamed Nassar. Retinal blood vessel segmentation using hybrid features and multi-layer perceptron neural networks. *Symmetry*, 12:894, 06 2020.

[109] Gendry Alfonso, Jesus Carlos Pedraza Ortega, Marco Aceves-Fernandez, and Saúl Tovar-Arriaga. Chaining a u-net with a residual u-net for retinal blood vessels segmentation. *IEEE Access*, PP:1–1, 02 2020.

[110] Ng Hi Huynh. Understanding evaluation metrics in medical image segmentation, 2023.

[111] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[112] Hechen Yang, Xin Zhao, Tao Jiang, Jinghua Zhang, Peng Zhao, Ao Chen, Marcin Grzegorzek, Shouliang Qi, Yueyang Teng, and Chen Li. Comparative study for patch-level and pixel-level segmentation of deep learning methods on transparent images of environmental microorganisms: From convolutional neural networks to visual transformers. *Applied Sciences*, 12(18), 2022.