

Deep Learning based Optimized Diabetic Retinopathy Detection Platform from Fundus Photographs

by

**Md. Azizul Hakim (191014012)
Tashfat Fatema (191014020)
Md. Mutasim Billah Abu Noman Akanda (192014038)**

*Capstone project (CSE499) submitted in partial fulfillment of the
requirements for the degree of*

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Dr. Mohammed Ashikur Rahman



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF LIBERAL ARTS BANGLADESH**

SUMMER 2022

© *Md. Azizul Hakim, Tashfat Fatema, and Md. Mutasim Billah Abu Noman Akanda*
All rights reserved

DECLARATION

Project Title	Deep Learning based Optimized Diabetic Retinopathy Detection Platform from Fundus Photographs
Authors	Md. Azizul Hakim, Tashfat Fatema, and Md. Mutasim Billah Abu Noman Akanda
Student IDs	191014012, 191014020, and 192014038
Supervisor	Dr. Mohammed Ashikur Rahman

We declare that this capstone project entitled *Deep Learning based Optimized Diabetic Retinopathy Detection Platform from Fundus Photographs* is the result of our own work except as cited in the references. The capstone project has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Md. Azizul Hakim
191014012

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Tashfat Fatema
191014020

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Md. Mutasim Billah Abu Noman Akanda
192014038

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: March 1, 2023



Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Mohammadpur, Dhaka - 1207

CERTIFICATE

This is to certify that the capstone project entitled **Deep Learning based Optimized Diabetic Retinopathy Detection Platform from Fundus Photographs**, submitted by **Md. Azizul Hakim** (191014012), **Tashfat Fatema** (191014020) and **Md. Mutasim Billah Abu Noman Akanda** (192014038) are a undergraduate student of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Dr. Mohammed Ashikur Rahman, Assistant Professor
Dept. of CSE, University of Liberal Arts Bangladesh

Dr. Muhammad Golam Kibria, Professor and Head (Acting)
Dept. of CSE, University of Liberal Arts Bangladesh

Dr. Nafees Mansoor, Associate Professor
Dept. of CSE, University of Liberal Arts Bangladesh

Dr. Shohag Barman, Assistant Professor
Dept. of CSE, University of Liberal Arts Bangladesh

Place: Dhaka

Date: March 1, 2023

ACKNOWLEDGEMENTS

We would like to express our deep and sincere gratitude to our research supervisor, *Dr. Mohammed Ashikur Rahman*, for giving us the opportunity to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh who taught us during the course of our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Md. Azizul Hakim, Tashfat Fatema and Md. Mutasim Billah Abu Noman Akanda

University of Liberal Arts Bangladesh

Date: March 1, 2023

To my beloved mother
 Sadida Hakim,
 father
 Md. Lutful Hakim,
and two lovely sisters
 Safa & Tory.

– *Md. Azizul Hakim*

To my beloved mother
 Neru Shamsun Nahar,
 father
 Md. Nurul Amin Miah,
 and brother
 Md. Tashrif Ul Amin.

– *Tashfat Fatema*

To my beloved mother
 Nasema Begum
 and father
 Md. Mohibullah Akanda.

– *Md. Mutasim Billah Abu Noman Akanda*

ABSTRACT

Diabetic Retinopathy (DR) is primarily an eye disease caused on by persistently high levels of blood glucose. If left untreated, DR can result in lifelong blindness. The fundus images are typically examined with the aid of highly skilled professionals to diagnose these devastating disorders. The manual diagnosis of these disorders, which is often carried out by clinicians, is exceedingly time-consuming, expensive and prone to error. Machine learning can evaluate medical images and has demonstrated its proficiency in evaluating the retinal fundus images. The implementation of deep learning techniques through algorithms has greatly assisted the earlier detection of diabetic retinopathy (DR). In this study, we used the EyePACS dataset, which contains high-resolution annotated images in five distinct classes. The images are produced under a range of imaging conditions and in a variety of dimensions. To detect the stage of diabetic retinopathy, we have experimented with a wide variety of deep learning models. And, we have achieved 89% testing *F1score* in our own model RevampResNet18. We have also developed a web platform to detect the stage of DR from fundus images. Previously, it took weeks to clinically diagnose the stage of DR. But, with our platform, users can find the stage of DR within seconds. People can access the website from anywhere in the world using an active internet connection in the PC or smartphone and a web browser.

Keywords: Deep Learning, Diabetic Retinopathy, CNN, ResNet, EfficientNet, VisionTransformer, RevampResNet18.

Contents

1	Introduction	1
1.1	Overview	1
1.1.1	Diabetic Retinopathy	1
1.1.1.1	Diabetic Eye	2
1.1.1.2	Risk Factors and Symptoms of Diabetic Retinopathy	2
1.1.2	Severity of Diabetic Retinopathy	4
1.1.3	Necessity of an Automated System	5
1.2	Problem Statement	5
1.3	Motivation	6
1.4	Aim and Objectives	7
1.5	Contribution of this Project	7
1.6	Layout of the Report	8
2	Background Study	10
2.1	Related Researches on Diabetic Retinopathy (DR)	10
2.2	Related Works using Deep Learning	12
2.3	Literature Analysis	19
3	Methodology	21
3.1	Dataset Preparation	21
3.1.1	Data Collection	21
3.1.2	Dataset Preprocessing	21
3.2	Custom Convolutional Neural Network (CNN)	25
3.2.1	Neural Networks	25
3.2.1.1	Logistic Regression: In a Neural Network Mindset	26

3.2.1.2	Gradient Descent to Minimize the Cost of the Hypothesis	27
3.2.1.3	Neural Network Representation	27
3.2.1.4	Computing a Neural Network Output	28
3.2.1.5	Forward Propagation	30
3.2.1.6	Backward Propagation	31
3.2.2	Activation Function	31
3.2.3	Convolutional Neural Networks	33
3.2.3.1	The Convolution Operation	34
3.2.3.2	Edge Detection	35
3.2.3.3	Vertical Edge Detection	35
3.2.3.4	Horizontal Edge Detection	36
3.2.3.5	Vertical and Horizontal Edge Detection	36
3.2.3.6	Padding	37
3.2.3.7	Valid & Same Convolutions	37
3.2.3.8	Strided Convolution	38
3.2.3.9	Summary of Convolutions	38
3.2.3.10	Convolution Over Volume	39
3.2.3.11	Use of Multiple Filters	40
3.2.3.12	One layer of a convolutional network	41
3.2.3.13	Analogy with a Neural Network	41
3.2.3.14	Notational Convention	42
3.2.3.15	Simple deep CNN example	42
3.2.3.16	Pooling Layers	44
3.2.3.17	Max Pooling	44
3.2.3.18	Average Pooling	44
3.2.3.19	Deep CNN Example	45
3.3	Residual Networks (ResNets)	45
3.3.0.1	Vanishing/Exploding Gradient	46
3.3.0.2	Skip Connections	46
3.3.0.3	How Residual Block Implements Skip Connections	46
3.3.0.4	Use of Bottleneck Layers in Residual Block	47
3.4	EfficientNet	48
3.4.1	Normal Convolution vs Depth-wise Separable Convolution	48

3.4.2	MobileNet Architecture	49
3.4.3	The Architecture of EfficientNet	49
3.5	Vision Transformer (ViT)	50
4	Experimental Results and Discussion	52
4.1	Experimental Setup	52
4.2	Experimental Result and Analysis	53
4.2.1	The Experimental Standard We Followed	53
4.2.1.1	Evaluation of the Models	53
4.2.2	EfficientNet	54
4.2.3	Vision Transformer (ViT-16)	58
4.2.4	Residual Network (ResNet)	59
4.2.5	RevampResNet-18	61
4.3	Proposed Web Platform	67
4.3.1	Model Deployment	68
4.3.1.1	Saving and Loading a Model	68
4.3.2	Implementation	68
4.3.2.1	Control Flow of the Platform	69
5	Impact on Society, Environment, and Sustainability	74
5.1	Impact on Society	74
5.2	Impact on Environment	74
5.3	Ethical Aspects	76
5.4	Sustainability Plan	77
6	Summary, Conclusion, and Future Scope	78
6.1	Summary of the Study	78
6.2	Conclusion	79
6.3	Implication for Future Study	79
	References	80

List of Figures

1.1	The abnormality of the diabetic eye compared to the healthy eye [<i>Diabetic Retinopathy; American Optometric Association: New York, NY, USA, 2021 (n.d.)</i>]	3
1.2	The abnormality of blood vessels and retina of an eye that has diabetic retinopathy compared to normal eye [<i>Diabetic Retinopathy; Mayo Foundation for Medical Education and Research: Rochester, MN, USA, 2022 (n.d.)</i>]	4
3.1	Data image count in various datasets	23
3.2	Logistic regression algorithm	26
3.3	Neural network representation: Shallow neural network	28
3.4	Neural network representation: Deep neural network	29
3.5	Taking a single node for mathematical computation	30
3.6	Computing the output of a single node	31
3.7	A deep neural network	32
3.8	Forward & backward propagation for the network	33
3.9	Different types of activation functions	33
3.10	The convolution operation: iteration 1	34
3.11	The convolution operation: iteration 2	34
3.12	Convolutions on RGB image	39
3.13	The convolution operation	40
3.14	The convolution operation with multiple filters	40
3.15	Neural network representation	41
3.16	Simple deep CNN example	43
3.17	Simple deep CNN example: feeding into neural network	43
3.18	Deep CNN example	45
3.19	Skip connection using a residual block He et al. (2016)	47

3.20	The architecture of the ResNet-18 model He et al. (2016)	47
3.21	The bottleneck layer	48
3.22	The architecture of EfficientNet Tan & Le (2019)	50
3.23	The architecture of the Vision Transformer Endcoder part Dosovitskiy et al. (2020)	51
4.1	Experimental Result of EfficientNet-B0	57
4.2	Experimental Result of EfficientNet-B1	57
4.3	Experimental Result of ViT-16	60
4.4	Experimental Result of ResNet-18	61
4.5	The Architecture of RevampResNet18	64
4.6	Layerwise comparison between ResNet18 and RevampResNet18	67
4.7	Experimental Result of RevampResNet-18	68
4.8	Block diagram of model deployment	69
4.9	The activity diagram of the proposed web platform	70
4.10	Homepage of our developed proposed platform	71
4.11	Model is analyzing the upload in the backend	72
4.12	The platform is showing the output of the analysis	73
5.1	The impact of Diabetic Retinopathy (DR) patients' consequences on the environment	75

List of Tables

2.1	Summary of the similar research works	20
3.1	Summary of the features of different datasets available	22
4.1	Result summary of EfficientNet and Vision Transformer	59
4.2	Computation Cost Summary of EfficientNet and Vision Transformer	59
4.3	Result Summary of ResNet Family	60
4.4	Computation Cost Summary of ResNet Family	60
4.5	Result summary of RevampResNet-18 and ResNet18	66
4.6	Computation cost summary of RevampResNet-18 and ResNet18 . . .	66

Chapter 1

Introduction

1.1 Overview

1.1.1 Diabetic Retinopathy

One of the most prevalent retinal illnesses, diabetic retinopathy (DR), is a consequence of diabetes that can lead to severe blindness in people. According to data from the World Health Organization (WHO), approximately 422 million people worldwide are diabetic patients. We can infer from these survey results that a sizable population is genuinely impacted by these deadly diseases, which not only cause high blood sugar levels but also cause many people to develop a variety of illnesses as a result of diabetes. People with diabetes began to develop too many different ailments. One of them is diabetic retinopathy (DR), which is a major contributor to blindness worldwide [Naveenkumar et al. \(2022\)](#). Diabetic retinopathy affects more than 93 million (35.4%) people worldwide, with DR threatening the vision of one third of those individuals [Hanco et al. \(2017\)](#). Since DR is a progressive condition, early detection and treatment are crucial. The severity of the condition relies on the number and types of lesions seen in the fundus images. Proliferative diabetic retinopathy and Nonproliferative diabetic retinopathy are the two main phases of DR. Nonproliferative DR, also known as diabetic retinopathy background, develops when the retina's blood vessels become weak due to diabetes and begin to leak blood and fluid onto the retinal surface [Solkar & Das \(2017\)](#). The retina may eventually separate from the back of your eye as a result of scar tissue produced by the development of new blood vessels. The eyeball may become pressurized if the new blood vessels obstruct the usual drainage of fluid from the eye. This

accumulation may harm the optic nerve, which is the nerve that transmits images from the eye to the brain, ultimately leading to blindness.

A sequential approach from no retinopathy through different phases of non-proliferative or proliferative retinopathy to advanced proliferative retinopathy has historically been used to classify and grade the severity of diabetic retinopathy. However, given that macular degeneration with severe vision loss can occur in the midst of modest ophthalmoscopic symptoms, this grade may not correctly reflect functionally serious condition [Lazar & Hajdu \(2012\)](#). According to the presence or absence of abnormal new capillaries, diabetic retinopathy is categorized as follows:

1. *Non-Proliferative Diabetic Retinopathy (NPDR)*: Nonproliferative DR develops when the retina's blood vessels become weak due to diabetes and begin to leak blood and fluid onto the retinal surface [Solkar & Das \(2017\)](#).
2. *Proliferative Diabetic Retinopathy (PDR)*: In proliferative DR, damaged blood vessels close off, causing the growth of new, abnormal blood vessels in the retina.

1.1.1.1 Diabetic Eye

People with diabetes may experience a variety of eye issues known as diabetic eye disease. These ailments include glaucoma, cataracts, diabetic macular edema, and diabetic retinopathy. Diabetes can harm your eyes over time, which could result in blurry vision or even blindness. However, by managing your diabetes, you can take measures to stop diabetic eye disease from occurring or to stop it from growing worse. The abnormality of the diabetic eye compared to the healthy eye is shown in the Figure 1.1. And the Figure 1.2 depicts the abnormality of blood vessels and retina of an eye that has diabetic retinopathy compared to normal eye.

1.1.1.2 Risk Factors and Symptoms of Diabetic Retinopathy

The risk of getting retinopathy increases with the duration of diabetes. Nearly 90% of persons with type 1 diabetes who have had it for longer than 10 years develop diabetic retinopathy to some degree. The percentage of type 2 diabetics with a disease duration of over 10 years who do not use insulin is 67%, whereas the

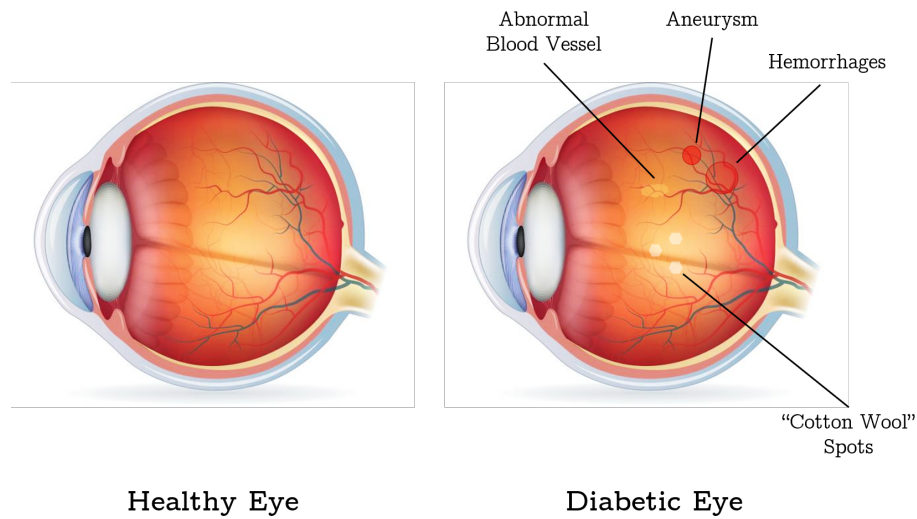


Figure 1.1: The abnormality of the diabetic eye compared to the healthy eye [[Diabetic Retinopathy; American Optometric Association: New York, NY, USA, 2021 \(n.d.\)](#)]

percentage of those with a disease duration of over 10 years who do take insulin is 79%.

Diabetes can cause damage to the tiny blood vessels in the retina, leading to loss of vision. The macula, the region of the retina responsible for sharp central vision, can experience fluid leakage and vision blur. New blood vessels may develop on the surface of the retina in an effort to increase blood flow, but they have the potential to rupture, leaking blood into the back of the eye and obstructing vision. The risk of getting diabetic retinopathy increases with a person's blood sugar level, if their blood glucose level is chronically rising. Hypertensive retinopathy can result from high blood pressure and higher blood sugar levels, and smokers are more likely to develop blood vessel diseases including retinopathy. To reduce the risk, one can control their blood pressure, blood sugar, and cholesterol levels.

Diabetics patient should get a yearly eye examination by an ophthalmologist with expertise in diabetic eye disease to identify early indicators of diabetic retinopathy. People with diabetes should be aware of the signs if they have not had routine eye exams since being diagnosed with the diabetes, if their blood sugar is poorly managed or controlled, or if they have had the ailment for some time. The symptoms are showed as follows:

- Increasing numbers of floaters, which are often described as dots or black

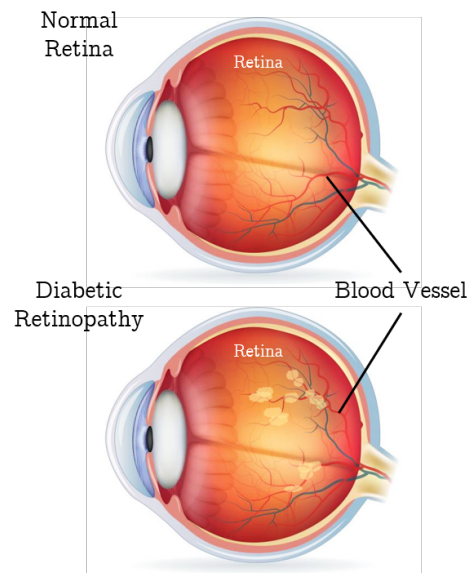


Figure 1.2: The abnormality of blood vessels and retina of an eye that has diabetic retinopathy compared to normal eye [*Diabetic Retinopathy; Mayo Foundation for Medical Education and Research: Rochester, MN, USA, 2022 (n.d.)*]

threads floating around the eye's vision

- Having eyesight which is distorted or blurred
- Frequently fluctuating eyesight
- Seeing a empty or other darkish region in one's vision range
- Having trouble seeing at night
- Often seeing that colors seem washed out or faded
- Difficulty reading or doing technical tasks
- Vision loss

These symptoms should be treated carefully since they might indicate detached retina, advanced stage diabetic retinopathy, or progressive diabetic retinopathy. They generally damage both eyes or occur in both.

1.1.2 Severity of Diabetic Retinopathy

The severity of the condition relies on the number and types of lesions seen in the fundus images. Proliferative diabetic retinopathy and Nonproliferative diabetic retinopathy are the two main phases of DR. Nonproliferative DR, also known as diabetic retinopathy background, develops when the retina's blood vessels become weak due to diabetes and begin to leak blood and fluid onto the retinal surface.

1.1.3 Necessity of an Automated System

There are currently not enough ophthalmologists in this disease-prone zone where the local population's prevalence of diabetes is comparably very high, making the manual approach for DR detection time-consuming. It would also be challenging for them to study and analyze the digital color fundus pictures of the retina in order to identify DR. Once completely programmed, computers using machine learning are able to classify data much more quickly and are better equipped to help clinicians identify diseases more promptly in real-time. Computer imaging research has been focused on the effectiveness of automated grading for DR. A subset of deep learning known as convolutional neural networks (CNNs) has a remarkable track record for applications in image evaluation and interpretation, including medical imaging [Nielsen \(2015\)](#). In our proposed methodology we used the Convolutional Neural Networks (CNNs) architecture to build an automated deep learning model for detecting diabetic retinopathy. We are able to construct a web-based system to detect the stages of diabetic retinopathy using fundus images. We would categorize the detection level of DR into five classifications, including: No DR, mild, moderate, severe, and proliferative DR. This will decrease the expense, time, and effort required for medical experts to diagnose DR, leading to effective deep learning models with realistic clinical potential.

1.2 Problem Statement

Diabetic retinopathy (DR), one of the most common retinal diseases, is a side effect of diabetes that can cause patients to go completely blind. The World Health Organization (WHO) estimates that there are 422 million diabetes individuals globally [Roglic et al. \(2016\)](#). Among these, one-third will eventually develop diabetic

retinopathy [Hou et al. \(2020\)](#). These survey findings suggest that a significant population is indeed affected by the disease, which not only results in high blood sugar levels but also frequently leads to a number of ailments because of diabetes. Diabetes patients started to experience an excessive number of different diseases. One of these is diabetic retinopathy (DR), a leading cause of blindness in the world. The manual approach for DR detection is time-consuming due to the lack of ophthalmologists in this disease-prone region, where diabetes prevalence in the local population is proportionally quite high. To recognize DR, it would also be difficult for them to analyze and evaluate the digital color fundus photographs of the retina. Once fully trained, computers using deep learning are better equipped to assist clinicians in identifying diseases more quickly in real-time and are considerably faster at classifying data. The deep learning based diagnosis process of the disease with the use of state-of-the-art CNN (Convolutional Neural Network) architectures will save time and assist the ophthalmologist in saving time and money.

1.3 Motivation

Diabetes, including type 1 and type 2, is a leading cause of morbidity and death worldwide [Amos et al. \(1997\)](#) [Yau et al. \(2012\)](#). The World Health Organization and the International Diabetes Federation estimate that there were 425 million individuals living with diabetes worldwide in 2017; this number is expected to increase to 629 million by the year 2045 [Federation \(2017\)](#). In 2017, 6.9 million persons in Bangladesh were projected by the International Diabetes Federation to have diabetes; this number is expected to rise to almost 10 million by 2025 [Federation \(2017\)](#).

Retinal screening can significantly lower the chance of going blind since treating diabetic retinopathy at an early stage allows for the preservation of excellent eyesight [Scanlon \(2017\)](#). Diabetic retinopathy is asymptomatic until the later, more severe stages. Nearly all cases of DR-related blindness may be avoided with adequate screening and treatment.

With 35% of the population older than 35 obtaining a diagnosis of diabetes or prediabetes, Bangladesh has one of the highest rates of diabetes in the world [Muqit et al. \(2019\)](#) [Becheva & Kirkova-Bogdanova \(2022\)](#). The broad expansion of DR screening programs in Bangladesh's public and commercial healthcare institutions

lacks efficient approaches. There is a lack of the necessary knowledge (ophthalmologist) not only in Bangladesh but also globally in regions where the local population has a high rate of diabetes. DR screening clinics located within eye hospitals cannot adequately handle the high screening demand on their own. The existing system does not effectively spread knowledge or encourage patients to seek out routine DR screening, and hardly any organizations actively engage in DR screening.

We have developed a web-based platform or system that can determine if a patient has diabetic retinopathy or not while keeping all of these risk factors in mind. Additionally, the system is able to indicate what stage the DR is in. Since there are five stages, they are No DR, Mild, Moderate, Severe, and PDR. The patient need not wait several weeks for the results because they may soon find about their eye health with the use of our system. The ineffective ophthalmologist and the amount of time it takes to receive the results of an eye examination were the primary driving forces behind the development of this system. Additionally, the manual process takes huge time to identify DR, but our system can provide the results more quickly, affordably, and efficiently.

1.4 Aim and Objectives

- The main goal of this study is to develop an automated diabetic retinopathy detection system with realistic clinical potential which will reduce the clinical diagnosis time of diabetic retinopathy by ophthalmologists.
- Improving the existing methodologies to reduce computational complexity and improve detection F1 Score
- Developing an automated Deep Learning based diabetic retinopathy detection platform which will result in a faster inference

1.5 Contribution of this Project

We have focused on developing a highly optimized diabetic retinopathy detection platform. As there are a few existing platforms, we tried to incorporate a faster web platform given that the accuracy remains better than the others. From our literature review (Section 2.2), we have found that most of the works were done without

classifying five stages of the DR. That is why we worked on classifying five stages of DR. In term of five-class-classification, we have also get better accuracy (in terms of *F1Scores*). Along with accuracy, we have also considered floating point operations to be minimized. And, we successfully minimized the FLOPs than all the other researches and get better accuracy than other researches (studies that used same dataset and performed 5-class classification) with our own developed deep learning model. From our literature analysis, we have found that there is a necessity of a faster platform of the detection of diabetic retinopathy for the diagnostic centers as they can give a pre-assumption to the patients and the ophthalmologists too.

1.6 Layout of the Report

The chapters outline below provide an overview of the structure of our report.

- *Chapter 1:* This chapter is presented as the report's introduction, focusing primarily on the fundamental principles of diabetic retinopathy, including risk factors, symptoms, and severity. Additionally, the significance of an automated system, the motivation for this project, and the contribution we put forth to improve the effectiveness and efficiency of our system.
- *Chapter 2:* This chapter discusses the background research on relevant studies on diabetic retinopathy (Section 2.1) and related works applying deep learning (Section 2.2). Moreover, the literature analysis is also described in this chapter (Section 2.3).
- *Chapter 3:* This chapter describes the methodology that was applied to the project analysis as a whole. Data collection and data preprocessing have been analysed and discussed in this section of the chapter (Section 3.1). Concepts and information required to create customized CNN, activation functions, residual networks, EfficientNet, and Vision Transformer (ViT) have also been covered in detail (Section 3.2-3.5).
- *Chapter 4:* The main focus of Chapter 4 is on the experimental results and discussion which covers the experimental setup for our system, the results and analysis of the experiment as well as specifics on the model we used for the experiment and the outcomes of all the models. Additionally, the

outcome and architecture of our own custom-built model is also discussed in this chapter (4.2). Besides that, a thorough explanation along with figures of the web platform we propose for our system was also described in this chapter (Section 4.3).

- *Chapter 5:* The main discussion of this chapter is how diabetic retinopathy affects society and the environment. Moreover, this chapter has mentioned the study's sustainability plan and ethical considerations.
- *Chapter 6:* This chapter has discussed the study's summary, conclusion, and future possible implications.

Chapter 2

Background Study

2.1 Related Researches on Diabetic Retinopathy (DR)

[Stitt et al. \(2016\)](#) stated that the most prevalent consequence of diabetes mellitus in most nations is diabetic retinopathy, which continues to be a major global cause of visual loss. Despite decades of intensive research into its genesis and pathophysiology, there are surprisingly few effective treatments. Up to 50% of individuals with diabetic macular edema (DME) do not react to novel therapies, including intravitreal vascular endothelial growth factor inhibitors (anti-VEGFs) and new steroids. Laser photocoagulation is still a go-to treatment for those with proliferative diabetic retinopathy (PDR), despite the fact that it is by nature destructive. The clinical characteristics of diabetic retinopathy and associated risk factors are outlined in this research. It covers specifics of retinal disease, cell culture techniques, and animal models that are employed to imitate its essential elements, enhance knowledge of its pathophysiology, and make it possible to find novel treatment targets. We stress that even while there have been substantial advancements, a deeper understanding of basic processes is still urgently needed to enable the development of trustworthy and rigorous methods to identify patients at highest risk and to effectively intervene before vision loss starts.

[Yau et al. \(2012\)](#) declaim that to investigate the global prevalence and key risk factors for both diabetic retinopathy (DR) and vision-threatening diabetic retinopathy (VTDR) in adults with diabetes. Individual participant data from population-based studies conducted all around the world were combined and analyzed. To find all population-based studies in healthy persons or those with diabetes who have determined DR using retinal images, a systematic review of

the literature was done. Studies provided information on significant systemic risk variables as well as DR end points, including as any DR, proliferative DR, diabetic macular edema, and VTDR. The 2010 World Diabetes Population aged 20–79 years was used to directly age-standardize the pooled prevalence data. Data on 22,896 diabetics were submitted by 35 investigations between 1980 and 2008. The prevalence of any DR was 34.6% (95% CI 34.5-34.8), proliferative DR was 6.96% (6.87-7.04), diabetic macular edema was 6.81% (6.74-6.89), and VTDR was 10.2% (10.1-10.3) overall. All DR prevalence end points were greater in persons with type 1 diabetes compared to type 2 diabetes and increased with diabetes duration, hemoglobin A1c, and blood pressure levels.

[Fong et al. \(2004\)](#) stated that for individuals between the ages of 20 and 74, diabetic retinopathy is the most common cause of new occurrences of blindness. Nearly all type 1 diabetes patients and more than 60% of type 2 diabetic patients develop retinopathy throughout the first two decades of the illness. Legal blindness was present in 3.6% of type 1 diabetes patients with a younger onset and 1.6% of type 2 diabetes patients with an older start in the Wisconsin Epidemiologic Study of Diabetic Retinopathy (WESDR). 86% of blindness in the group with younger onset was caused by diabetic retinopathy. One-third of the instances of legal blindness in the older-onset group, when other eye disorders were prevalent, were brought on by diabetic retinopathy.

[Shukla & Tripathy \(2022\)](#) reported that in Western people of working age, diabetic retinopathy is the most common cause of vision loss. This microvascular consequence of diabetes mellitus is the most frequent. The retina may suffer vision-threatening damage as a result of diabetic retinopathy, which might ultimately result in blindness. To arrest the disease's development, strict glycemic control, early identification, and proper care are essential. The interprofessional team's involvement in diagnosing and treating individuals with diabetic retinopathy is highlighted in this paper, which also discusses the disease's etiology, diagnosis, and treatment. This paper's goals were to define the causes of diabetic retinopathy, list the clinical characteristics that would be seen during an assessment of the condition, and discuss its treatment and prognosis. In addition, the objective was to emphasizing the need of interprofessional cooperation in enhancing patient outcomes in diabetic retinopathy patients.

[Zheng et al. \(2012\)](#) stated that world's health systems are significantly impacted by diabetic retinopathy (DR), a prominent microvascular consequence of diabetes. If immediate action is not done, the number of individuals with DR would increase from 126.6 million in 2010 to 191.0 million by 2030 and the number of people with vision-threatening diabetic retinopathy (VTDR) will rise from 37.3 million to 56.3 million. DR typically results in impaired visual functioning and is the primary cause of blindness in working-age populations, despite mounting evidence of the usefulness of routine screening for DR and early treatment. Many low-income nations, where access to qualified eye care practitioners and tertiary eye care facilities may be insufficient, have disregarded DR in health care research and planning. Services supply and demand may both be problematic. Low rates of adherence to diabetes prescriptions and yearly eye exams may exist, for a variety of complex causes. Innovative and thorough methods are required to lower the risk of vision loss by swiftly diagnosing and treating VTDR.

[Safi et al. \(2018\)](#) reported that one of the main causes of vision impairment globally is diabetic retinopathy (DR). Before the initial clinical symptoms of DR manifest, diabetes mellitus may be linked to neurovascular damage that is ophthalmoscopically imperceptible. Early functional and invisible structural neuroretinal changes have been suggested to be detected by reductions in the inner neuroretinal layer thickness on macular optical coherence tomography, decreased contrast sensitivity primarily at low spatial frequencies, abnormal results in color vision and microperimetry tests, and a prolonged implicit time recorded by multifocal electroretinography. To assess the early stages of DR, vascular anomalies including alterations in retinal artery caliber, architectural indices, and blood flow have been studied. Early indicators of DR have been proposed to include retinal oxygen saturation patterns, optical coherence tomography angiography findings, and high levels of blood markers and cytokines. Rodents have developed light-based molecular imaging to show variations in protein expression in the retinal microvessels as diagnostic indicators. Future clinical research will look at the effectiveness and safety of this method in people. We provide an overview of all research on preclinical DR diagnostics.

2.2 Related Works using Deep Learning

[Qummar et al. \(2019\)](#) told that an eye condition known as diabetic retinopathy (DR) harms the retinal blood vessels. If DR is not identified in its early stages, it can cause vision impairment and ultimately result in blindness. Normal, mild, moderate, severe, and PDR are the five phases or grades of DR (Proliferative Diabetic Retinopathy). The colored fundus photos are often examined by highly qualified professionals to identify this catastrophic condition. It takes time and is prone to error for clinicians to manually diagnose this illness. Determining DR and its various stages automatically from retinal images has so been proposed using a variety of computer vision-based techniques. However, these techniques can only identify DR's many stages with a relatively poor level of accuracy, especially for the early stages, because they are unable to encode the underlying intricate properties. In this study, we trained an ensemble of five deep Convolution Neural Network (CNN) models (Resnet50, Inceptionv3, Xception, Dense121, and Dense169) to encode the rich characteristics and enhance the classification for various phases of DR using the publicly available Kaggle dataset of retina images. The experimental results demonstrate that, in contrast to the state-of-the-art approaches on the same Kaggle dataset, the proposed model detects all stages of DR.

[Khalifa et al. \(2019\)](#) reported that the most prevalent form of diabetic retinopathy (DR) and a major contributor to blindness. There will be more diabetic patients. According to the International Diabetes Federation, to 552 million by 2034. (IDF). Goal: Thanks to developments in computer science techniques like artificial intelligence (AI) and deep learning (DL), there are now more options to identify DR at an early stage. The likelihood of recovery will rise as a result, and patients' chances of losing their vision in the future will decrease. Methods: Deep transfer learning models for medical DR detection were looked into in this work. The 2019 dataset from the Asia Pacific Tele-Ophthalmology Society (APTOS) was used to train and test the DL models. Given that it was recently published in the second quarter of 2019, this study is regarded as one of the first to use the APTOS 2019 dataset. The deep transfer models chosen for this study were VGG16, VGG19, Res-Net18, SqueezeNet, and GoogleNet. These models were chosen because, in contrast to larger models like DenseNet and InceptionResNet, they have fewer layers. Techniques for data augmentation were utilized to strengthen the models and solve the overfitting issue.

Results: To demonstrate the reliability of the chosen models, performance measures including precision, recall, and *F1score* were calculated. The testing accuracy that the AlexNet model attained was the greatest at 97.9%. The acquired performance measures also improved the quality of our output. Additionally, AlexNet contains the fewest possible layers, which reduces both the computational complexity and training time.

[Hemanth et al. \(2020\)](#) stated that this study's goal is to suggest a different, hybrid approach to using retinal fundus pictures to diagnose diabetic retinopathy. In order to achieve better outcomes, the hybrid approach is built on combining deep learning and image processing. Reliable diabetic retinopathy identification from digital fundus images is a known open topic in medical image processing that requires alternative solutions to be developed. In this situation, manual interpretation of retinal fundus images entails a significant amount of labor, skill, and processing time. Doctors therefore require assistance from imaging and computer vision systems, and the deployment of intelligent diagnosis tools is generally considered to be the next stage. The study's suggested approach to solving the problem makes use of image processing with histogram equalization and the contrast restricted methods of adaptive histogram equalization. Next, a convolutional neural network's classification is used to make the diagnosis. 400 retinal fundus photos from the MESSIDOR database used as validation examples for the approach. Accuracy 97%, sensitivity (recall) 94%, specificity 98%, precision 94%, FScore 94%, and GMean 95% were the average values for the various performance evaluation metrics. Along with those findings, a general comparison with some other studies has demonstrated that the newly presented method is sufficiently effective and successful in detecting diabetic retinopathy from retinal fundus pictures. The suggested method and research findings are significant contributions to the related literature since they use deep learning and related image processing techniques to diagnose diabetic retinopathy.

[Nguyen et al. \(2020\)](#) said that an eye condition known as diabetic retinopathy (DR) is linked to long-term diabetes. DR is thought to impact more than 93 million people and is the main cause of blindness in working-age individuals worldwide. If DR is discovered early enough, the progression of vision impairment can be slowed or controlled. However, this can be challenging because the condition frequently exhibits few symptoms until it is too late for effective therapy. Currently, diagnosing DR is a labor-intensive and time-consuming process that calls for an

ophthalmologist or other skilled clinician to look over and assess digital color fundus images of the retina in order to spot lesions connected to the vascular irregularities brought on by the illness. The automated DR screening method will expedite detection and decision-making, which will aid in controlling or control the progression of DR. This study proposes an automated classification system that examines fundus images with varied levels of automation. Using machine learning models like CNN, VGG-16, and VGG-19, it calculates a severity grade for diabetic retinopathy (DR) based on lighting and field of view. When classifying images into two categories, this system obtains 80% sensitivity, 82% accuracy, 82% specificity, and 0.904 AUC.

[Tymchenko et al. \(2020\)](#) said that one of the most dangerous effects of diabetes, diabetic retinopathy can result in total blindness if ignored. Early detection is a major obstacle because it is crucial to the success of treatment. Unfortunately, accurate diagnosis of the diabetic retinopathy stage is notoriously difficult and necessitates skilled human fundus imaging interpretation. Millions of people can benefit from the detection step's simplification, which is essential. Convolutional neural networks (CNN) have been used to diagnose diabetic retinopathy and many other related conditions. However, the performance of these algorithms is hindered by the costly expense of large labeled datasets as well as variation in doctors. Using a single snapshot of the human fundus, we suggest in this study an automated deep-learning-based method for detecting the stage of diabetic retinopathy. We also suggest the multistage transfer learning method, which uses comparable datasets with various labeling. The provided approach, which is ranked 54 out of 2943 rival methods (quadratic weighted kappa score of 0.925466) on the APTOS 2019 Blindness Diagnosis Dataset, can be utilized as a screening method for early detection of diabetic retinopathy (13000 images).

[Gangwar & Ravi \(2021\)](#) stated that one of the main causes of blindness in the population between the ages of 20 and 65 is diabetic retinopathy. In this article, we discuss the issue of automatically detecting diabetic retinopathy and present a unique deep learning hybrid to handle the issue. We Utilize transfer learning on the already trained Inception-ResNet-v2 and add a custom block of CNN layers on top of Inception-ResNet-v2 to create the hybrid model. We APTOS 2019 blindness detection and the Messidor-1 diabetic retinopathy dataset were used to assess how well the suggested model performed (Kaggle dataset). Compared to other published

results, our model performed better. A test accuracy of 72.33% was attained. and 82.18% on the corresponding Messidor-1 and APTOS datasets.

[Rajalakshmi et al. \(2018\)](#) stated the purpose of detecting diabetic retinopathy (DR) and sight-threatening retinopathy (STDR) using fundus photography captured with a smartphone-based equipment and validating it against an ophthalmologist's grading, and the function of artificial intelligence-based automated software. The ophthalmologists evaluated the DR using the International Clinical DR (ICDR) classification system. Proliferative DR, diabetic macular oedema, or severe non-proliferative DR were used to characterize STDR (DME). A proven AI DR screening tool (EyeArt™) that can distinguish between DR, referable DR (moderate non-proliferative DR or worse and/or DME), and STDR was used to examine the retinal images. The automated grading's sensitivity and specificity were evaluated and validated against the grading performed by ophthalmologists. Grading was done on the retinal pictures of 296 individuals. As compared to STDR, which was found in 112 (37.8%) and 146 (49.3%) patients, DR was found in 191 (64.5%) by ophthalmologists and in 203 (68.6%) by AI software.

[Wan et al. \(2018\)](#) declaim that one of the main causes of blindness in the population that is active is diabetic retinopathy (DR), which is a frequent consequence of diabetes. Controlling blood glucose levels and receiving therapy on time can stop many DR problems in their tracks. Due to the diversity and complexity of DR, manual diagnosis takes a lot of time and is very challenging for DR detection. Their goal of this study is to find an automated classification method for a series of fundus photographs. They use convolutional neural networks (CNNs) to tackle the three most challenging problems in DR detection: classification, segmentation, and detection. They adopt AlexNet, VGGNet, GoogleNet, and ResNet and evaluate how well these models perform with the DR image categorization using transfer learning and hyper-parameter tweaking. They also train these models on the publicly accessible Kaggle platform. The results show that CNNs and transfer learning have higher classification accuracy on DR images, with the best accuracy being 95.68%.

[Alyoubi et al. \(2020\)](#) reviewed on their paper that Diabetes mellitus frequently results in diabetic retinopathy (DR), which results in lesions on the retina that impair vision. Blindness may result if it is not caught in time. Unfortunately, there is no cure for DR; treatment only preserves vision. Early diagnosis and treatment

of DR can greatly lower the risk of vision loss. In contrast to computer-aided diagnosis technologies, the manual diagnosis of DR retina fundus pictures by ophthalmologists is costly, time-consuming, and prone to error. Deep learning has recently risen to prominence as one of the most popular methods for improving performance, particularly in the categorization and interpretation of medical images. Convolutional neural networks are more frequently employed as a deep learning technique in the processing of medical images, and they are quite successful. The most cutting-edge ways for classifying and detecting DR color fundus pictures using deep learning techniques have been evaluated and examined for this paper. Additionally, the color fundus retina DR datasets have been examined. There are also discussions on several complex subjects that demand further research.

[Qiao et al. \(2020\)](#) claim that it has been extremely difficult to predict the existence of microaneurysms in fundus pictures and identify diabetic retinopathy in its early stages. Long-term high blood glucose levels cause diabetic retinopathy (DR), which results in microvascular problems and permanent visual loss. The first symptoms of DR are the development of microaneurysms and macular edema in the retina, and prompt detection can lower the chance of developing non-proliferated diabetic retinopathy. Deep learning is quickly advancing, making it an effective method for offering an intriguing answer to difficulties involving medical picture interpretation. The proposed system analyzes the presence of a microaneurysm in a fundus image using convolutional neural network algorithms that incorporate deep learning as a key component and are accelerated by a GPU (Graphics Processing Unit). This will enable high-performance and low-latency inference for the detection and segmentation of medical images. The fundus image is categorized as normal or diseased using the semantic segmentation technique. This gives ophthalmologists an automated technique to help them classify fundus pictures as early, moderate, or severe NPDR. The Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy has been proposed, and it is capable of effectively training a deep convolution neural network for semantic segmentation of fundus images, which can improve the efficiency and accuracy of NPDR (non-proliferated diabetic retinopathy) prediction.

[Atwany et al. \(2022\)](#) on their survey paper stated that high blood glucose levels cause lesions on the retina of the eye, resulting in the degenerative condition known as diabetic retinopathy (DR), which affects the eyes. The majority of diabetes

patients who become blind are thought to have diabetic retinopathy, particularly the working-age population in poor countries. Since the condition is permanent, the goal of treatment is to preserve the patient's existing level of eyesight. Effective maintenance of the patient's vision depends on the early identification of diabetic retinopathy. The fundamental problem with DR detection is that the manual diagnostic method, which entails an ophthalmologist's evaluation of retinal fundus pictures of the eye, is exceedingly time-, money-, and effort-consuming. This process also proved to be more challenging, especially early on in the disease when disease characteristics are less obvious in the photos. Deep learning algorithms have assisted in the early identification of diabetic retinopathy, and machine learning-based medical image analysis has demonstrated its competency in evaluating retinal fundus pictures (DR). This study proposes the categorization and detection of retinal fundus images using cutting-edge deep learning techniques in supervised, self-supervised, and Vision Transformer configurations. Examples include reviews and summaries of the referable, non-referable, and proliferative categories of diabetic retinopathy. The research also examines the retinal fundus datasets for diabetic retinopathy that are readily available and utilized for tasks including detection, classification, and segmentation.

[Ishtiaq et al. \(2020\)](#) stated that Diabetic Retinopathy (DR) is a condition brought on by uncontrolled diabetes that can cause individuals to become legally blind. The benefits of automated DR detection over manual detection have increased due to developments in artificial intelligence. There are now a number of published research on automated deep learning or machine learning DR detection systems. In this paper, DR detection methods are reviewed from five distinct angles: datasets, picture preprocessing methods, machine learning-based approaches, deep learning-based approaches, and performance measurements. Additionally, it includes the authors' observations and the importance of the review's results. We also talk about nine brand-new DR detection research difficulties. For this study, 74 primary papers were chosen from eight academic databases after a careful selection procedure. Numerous public datasets are accessible in the field of DR detection, according on the chosen research. Contrast enhancement and green channel extraction together with picture preprocessing approaches were the most effective in improving classification accuracy. The strongest discriminatory characteristics for DR identification were found to be shape-based, texture-based, and statistical features. Comparing

the Artificial Neural Network to other machine learning classifiers, it was shown to be the best. Convolutional neural networks fared better than other deep learning networks in this area. Finally, criteria for accuracy, sensitivity, and specificity were mainly used to assess categorization performance. The community of scientists working on automated DR detection strategies will find this review, which provides a thorough outline of DR detection techniques, to be helpful.

2.3 Literature Analysis

We have showed an overview of the more recent related works on diabetic retinopathy detection using deep learning in the Section 2.2. All of the works has stated the importance of the early detection of this disease. Also, they tried to incorporate the recent deep learning technologies. Their performance varies with the solution they proposed. A summary of all these work has been presented in the Table 2.1. And, the summary of our findings are listed below:

- Transfer learning and ensemble learning are the approaches that researchers mostly followed.
- Transfer learning is a method that employs pieces of a model that has already been trained on a new network to solve a separate but related problem.
- Ensemble learning method combines several machine learning techniques into one predictive model.
- The researchers considered models with a smaller number of hidden layers with importance.
- Use of different datasets on different training, testing, and validation stage varied the accuracy of the papers.

Our examination of the literature revealed that the majority of studies employed baseline deep learning architectures like AlexNet, VGG, ResNet, etc. A few studies have also experimented with ensemble learning architectures. Those that used 2-class classification outperformed studies that used 5-class classification in terms of *F1Scores*. Different studies also employed a wide range of datasets. The accuracy of the research that employed the EyePACS dataset (and 5-class classification)

ranged from 70% to 85%. We learned from this survey that we may modify the architectures to make them better such that the models simultaneously provide greater computational solvency and *F1Scores*.

Table 2.1: Summary of the similar research works

Author(s)	Dataset(s)	Image Preprocessing	Architecture	Result
Qummar et al. (2019)	EyePACS (5 classes)	Image Resizing, Image Cropping, Image Rotation, Image Flipping, Mean Normalization	ResNet50, InceptionV3, Xception, Dense121, Dense169 (Ensemble)	<i>F1score</i> : 53.74%
Khalifa et al. (2019)	APTOS 2019 (5 classes)	Image Augmentation	AlexNet, VGG16, VGG19, ResNet18, SqueezeNet, GoogleNet	95.82%, 95.60%, 95.20%, 84.16%, 82.48%, 92.73% (<i>F1Scores</i>)
Hemanth et al. (2020)	MESSIDOR-1 (5 classes)	Image Resizing, Histogram Equalization, CLAHE	CNN	<i>F1score</i> : 94.00%
Nguyen et al. (2020)	EyePACS (2 classes)	Image Cropping, Image Dropping, Color Normalization	VGG16, VGG19, CNN with customized conv. layer	71%-73%, 80%-83% (ACC)
Tymchenko et al. (2020)	APTOS 2019, IDRiD, MESSIDOR-1	Image Cropping, Image Resizing	EfficientNet-B4, EfficientNet-B5, SE-ResNetXt50 (Ensemble)	<i>F1score</i> : 82.70%
Gangwar & Ravi (2021)	MESSIDOR-1, APTOS 2019	Image Blurring, Image Cropping, Image Resizing	Inception-ResNet-v2 with custom CNN layers	72.33%, 82.18% (ACC)
Abràmoff et al. (2016)	MESSIDOR-2 (4 classes)	Image Augmentation	Various CNNs	Sensitivity 96.8%, Specificity 87%
Pratt et al. (2016)	Kaggle (5 classes)	Color Normalization, Image Resizing	Custom CNN	75% (ACC)
Harangi et al. (2019)	MESSIDOR-1	-	AlexNet	90.07% (ACC)
Li et al. (2019)	IDRiD	-	ResNet50	65.1% (ACC)

Chapter 3

Methodology

3.1 Dataset Preparation

3.1.1 Data Collection

Dataset for this research has been collected from EyePACS. EyePACS a pioneer in the screening program for Diabetic Retinopathy.

- EyePACS has partnership with California Healthcare Foundation in the DR Screening Program which confirms the reliability of its dataset.
- Dataset contains 35,126 high resolution annotated images in 5 different classes.
- The images range in a number of dimensions at a variety of imaging condition. Some common dimensions are: 4752×3168 , 2592×1944 , 4928×3264 , 3888×2592 etc.

A summary of the datasets we found is given in the Table 3.1. The graph in the Figure 3.1 shows the Image Count vs Dataset Size relation.

3.1.2 Dataset Preprocessing

We have followed the conventions given below for our model's (EfficientNet-B0, EfficientNet-B1, ViT-16, and ResNet-18) experiment purpose:

- The image has been resized to 224×224 because it is the standard input size for the pretrained models. These models are designed to work with input

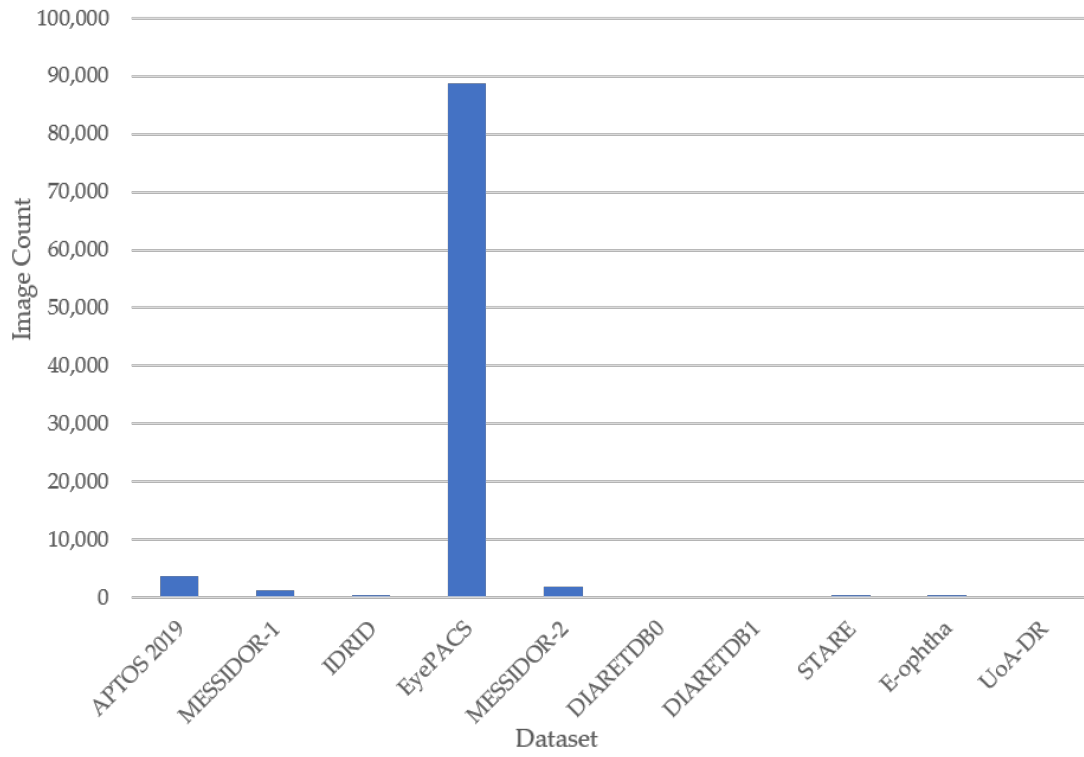


Figure 3.1: Data image count in various datasets

images of a specific size, and resizing the images to this size allows the model to process the images efficiently and effectively.

- Random horizontal flip is used with a probability of 0.4 to increase the diversity of the training data. This helps to reduce overfitting and improve the generalization of the model. Horizontal flipping is used because it preserves the class label of the image.
- Normalizing the images is a common pre-processing step that helps to improve the performance of the model. Normalization scales the pixel values to have zero mean and unit variance, which can help to reduce the internal covariate shift and speed up the convergence of the model.
- Cross-entropy loss is a commonly used loss function for image classification tasks. It measures the dissimilarity between the predicted probability distribution and the true probability distribution. It is a popular choice in image

classification tasks because it can be used to optimize the log-likelihood of the true labels. In all our experiments, we have used either ReLU or LeakyReLU even in the classifier layer instead of Softmax because Cross-entropy loss is already the combination of Negative Log Likelihood loss (NLLL) and Softmax. The softmax function is applied to the output of a neural network to produce probabilities for each class, and the NLLL is used to compare these probabilities with the true labels. The formula for cross-entropy loss for a single sample is defined as:

$$\text{Cross-entropy Loss} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Where,

- y is the true label, a binary value (0 or 1) for binary classification, or a one-hot encoded vector for multiclass classification.
 - $\hat{y}$ is the predicted probability for the positive class.
 - $\log(\cdot)$ is the natural logarithm.
- ReLU (Rectified Linear Unit) is an activation function commonly used in deep learning models. It replaces all negative input values with zero, which helps to introduce non-linearity into the model and improve the representation power of the model. The equation for ReLU is:

$$\text{ReLU}(y) = \max(0, x)$$

Also, we have used LeakyReLU as an activation function and a dropout rate of 0.5 as a regularizer for our newly developed architecture *ResManNet-10* which is given below:

- LeakyReLU is a variant of the Rectified Linear Unit (ReLU) activation function. The standard ReLU function sets negative input values to zero, but the LeakyReLU function instead sets them to a small negative value (a “leak”). The purpose of this is to address the “dying ReLU” problem, where ReLU units can become inactive (i.e. stuck at zero) if they receive negative inputs. By introducing a small gradient for negative inputs, LeakyReLU helps to prevent this from happening. The equation for LeakyReLU is-

$$\text{LeakyReLU}(y) = \max(ax, x)$$

where x is the input and a is a small positive value such as 0.01.

- Dropout is a regularization technique that helps to prevent overfitting by randomly setting a fraction of input units to zero during training. It helps to reduce the co-adaptation of neurons in deep neural networks and improve the generalization of the model.

3.2 Custom Convolutional Neural Network (CNN)

3.2.1 Neural Networks

One of the most widely used algorithms in machine learning is neural networks. Artificial Neural network (ANN) have gained popularity recently and is a useful model for many fields. Everything kind of depends on how it is put into practice with the support of this model. For instance, big data, cloud computing, the internet, and information and communication technologies are all partially related to the concept of the neural network. First, we must understand what a neural network is.

We will be able to comprehend the meaning more clearly if we break down each word separately and attempt to determine its meaning. The word 'neural' derives from the word 'neuron', which basically refers to the human brain's information transmitter. In contrast, the word 'network' refers connection. As a result, if we combine the meanings of these two words, 'neural network', we might be able to interpret their meaning as 'sharing of data' or 'sharing of information'. It mostly covers the range of concepts and methods.

An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

Supervised learning primarily teaches how to perform actions that make sense in neural networks; for instance, we can designate inputs as 'x' and be able to obtain some outputs 'y' for those specific inputs.

Neural network consists of nodes and connection between the nodes. Each connection represent parameter values that were estimated when this neural network

was fit to the data. These parameters estimates to slop and intercept values that solve for when we fit a straight line to data. The connection is build by a spider web connection between each layers of nodes (Figure 2).

The layers of nodes between input and output nodes are called hidden layer. When we build neural networks, one of the first things we decide is how many hidden layers will be there and how many nodes go into each of the hidden layers.

3.2.1.1 Logistic Regression: In a Neural Network Mindset

Logistic Regression is an algorithm for binary classification. Functionally, logistic regression can be thought of as a neural network without any hidden layer. So, before diving into a neural network, let us understand the key concepts of logistic regression. Observe the Figure 3.2. Suppose, we are going to classify an image as a 'DR' or 'No DR' image.

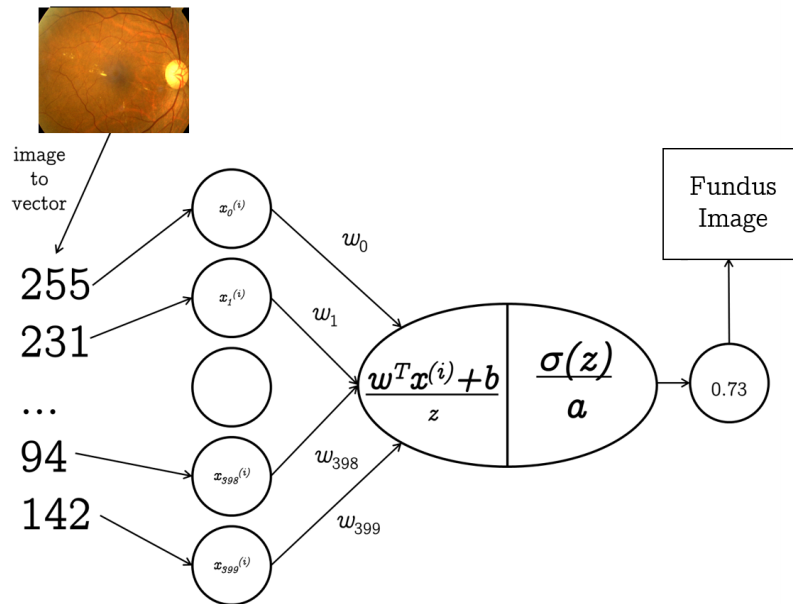


Figure 3.2: Logistic regression algorithm

The algorithm will take one image example. It will create a long vector consisting of three color-channels of the image. It will then calculate the linear sigmoid of the

input which is $\hat{y} = \sigma(w^T x + b)$ ¹, where $0 \leq \hat{y} \leq 1$. If $\hat{y} \geq 0.5$, it will be classified as a 'DR' image otherwise a 'No DR' image.

Mathematically, for one image example $x^{(i)}$:

$$\begin{aligned} z^{(i)} &= w^T x^{(i)} + b \\ \hat{y}^{(i)} &= a^{(i)} = \sigma(z^{(i)}) \\ \iota(a^{(i)}, y^{(i)}) &= -y^{(i)} \log(a^{(i)}) - (1 - y^{(i)}) \log(1 - a^{(i)}) \end{aligned}$$

And, to compute the cost, we need to sum over all the training examples,

$$J = \frac{1}{m} \sum_{i=1}^m \iota(a^{(i)}, y^{(i)})$$

3.2.1.2 Gradient Descent to Minimize the Cost of the Hypothesis

In the section 3.2.1.1, we have seen how to calculate the activation and the cost for image examples. Now, we need to minimize the cost for better accuracy from the learning algorithm. To do so, the two parameters of the algorithm, w and b , need to be better-learned. Here, w is the weights for each of the pixel intensity values for an image example and b is a bias unit which is just a real number. Basically these two parameters will be get updated by performing gradient descent.

For the cost-minimization task, gradient descent is the most popular algorithm. The two formulas that are going to be used as following:

$$\begin{aligned} \frac{\partial J}{\partial w} &= \frac{1}{m} X(A - Y)^T \\ \frac{\partial J}{\partial b} &= \frac{1}{m} \sum_{i=1}^m (a^{(i)} - y^{(i)}) \end{aligned}$$

And, finally, the value of the w and the b will be get optimized by updating the corresponding value using the following formulas:

$$\begin{aligned} w &= w - \alpha \times \frac{\partial J}{\partial w} \\ b &= b - \alpha \times \frac{\partial J}{\partial b} \end{aligned}$$

¹The definition of function σ is given in the section 3.2.2.

Here, α is the learning rate that we are going to use. The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated [Brownlee \(2020\)](#).

3.2.1.3 Neural Network Representation

A single layer neural network also known as Shallow neural network looks something like the Figure 3.3.

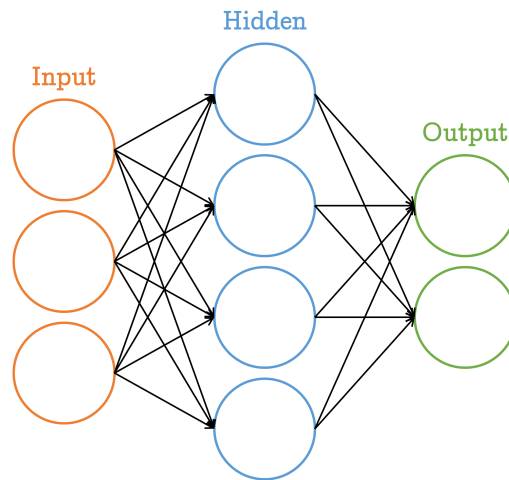


Figure 3.3: Neural network representation: Shallow neural network

The input feature is located on the left most line and is actually piled vertically which is basically the neural network's input layer, which contains the feature that we specified in the starting. On the other hand, the final layer in the neural network is actually referred to as the output layer and is mainly responsible for producing the anticipated/predicted value $\hat{y}$. The hidden layer is the interconnecting layer between the input layer and the output layer. Because the training set does not recognize the true value of these hidden units, it is given the term 'hidden'. When there are multiple hidden layers, such as in the Figure 3.4, a deep neural network is formed.

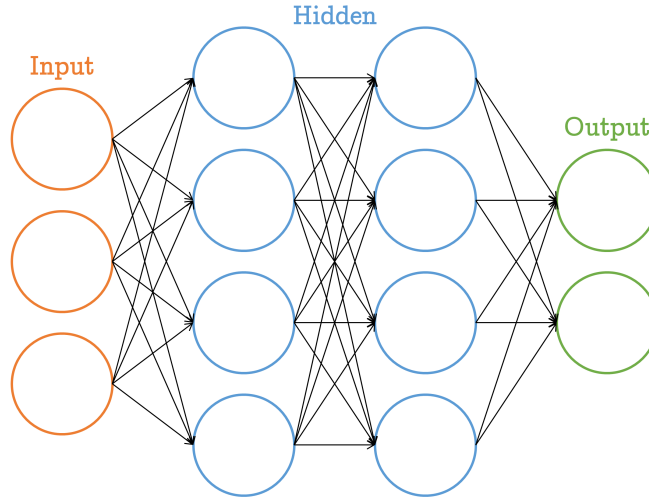


Figure 3.4: Neural network representation: Deep neural network

3.2.1.4 Computing a Neural Network Output

In here, the single hidden layer neural network will be used to establish the basic concept behind generating the output.

As seen in Figure 3.5, we are only using the blue portion that has been highlighted to show the detailed calculations made to produce the output unit.

Here, x_1, x_2 and x_3 are the three inputs that generate the output node by transposing the weight of the input nodes. For a given weight by turning rows into columns and columns into rows. It is particularly helpful in circumstances where it is necessary to determine the inverse and adjoint of weight matrix. Multiplying this transpose weight with the input which is denoted as x and also adding bias b to it.

For our convenience, the entire equation can be written as z which is actually a linear function. On the other hand, performing a sigmoid on the linear function z will ultimately produce the output a which is our neural network's output.

As we have explained only for one particular node, now for some input x we can generate the following neural network.

For each hidden layer with respect to the inputs layer, we get the following linear function and activation functions:

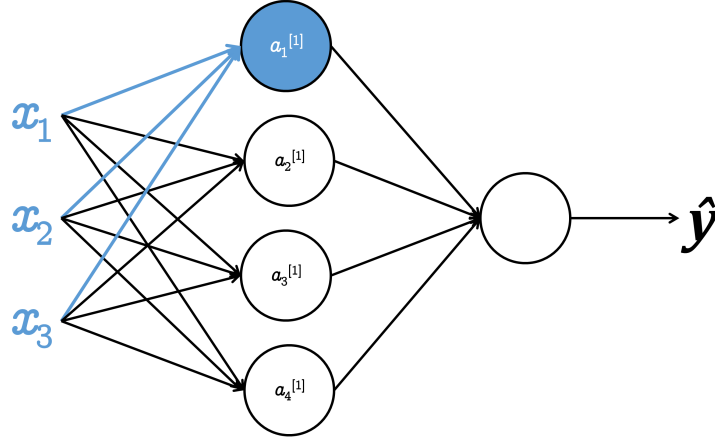


Figure 3.5: Taking a single node for mathematical computation

$$\begin{aligned}
 z_1^{[1]} &= w_1^{[1]T} X + b_1^{[1]}, a_1^{[1]} = \sigma(z_1^{[1]}) \\
 z_2^{[1]} &= w_2^{[1]T} X + b_2^{[1]}, a_2^{[1]} = \sigma(z_2^{[1]}) \\
 z_3^{[1]} &= w_3^{[1]T} X + b_3^{[1]}, a_3^{[1]} = \sigma(z_3^{[1]}) \\
 z_4^{[1]} &= w_4^{[1]T} X + b_4^{[1]}, a_4^{[1]} = \sigma(z_4^{[1]})
 \end{aligned}$$

Let's explain first linear function that is generated. For the first hidden node and first hidden layer, the transpose of weight w_1 is multiplied by X and adding bias b_1 to the layer one, along with activation function of first node and first layer ($a_1^{[1]}$). Ultimately, by doing the sigmoid of the linear function that we get from the first node and first layer the output is generated. And, in this way, all the output is generated for layer one but node one, two, three, and four respectively.

3.2.1.5 Forward Propagation

Let's adopt a more comprehensive strategy and place more emphasis on the fundamental components of deep neural networks. We can refer to a neural network as a deep neural network if it has more than one hidden layer. The figure is a deep neural network with three hidden layers.

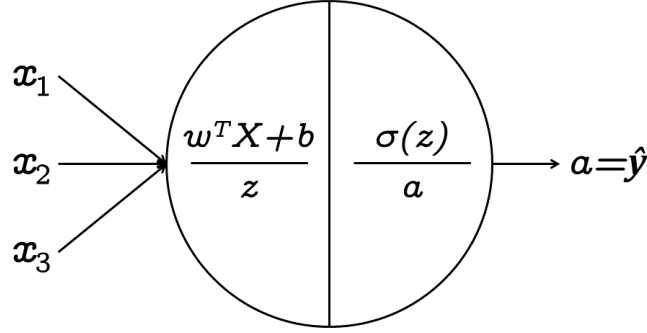


Figure 3.6: Computing the output of a single node

In a single hidden layer, also known as a shallow neural network, we have already discussed how to calculate the linear and activation output of a particular node. For a deep neural network, we must carry out the exact task on each node in each of the hidden layers.

Consider about the example of the single hidden layer or shallow neural network. There, we calculated the linear output for a certain node using the weight matrix and bias unit for that layer. After that, we computed

$$W^T X + b$$

In this case, the X denotes the activation of the first input layer, the following layer.

3.2.1.6 Backward Propagation

We obtain a final output for a particular set of input values after conducting the forward propagation procedure. Because the output is a predicted output, it might be considered a hypothesis function. We also obtain a cost value for that hypothesis, which is the numerical difference between the actual and projected outputs. We must keep that cost to a low. Using a backward propagation approach, we reduce the cost. The algorithm gradually reduces the cost by learning the neural network's parameters or weight values.

The final hidden layer is where the backward propagation begins. It determines the cost function's partial derivative with regard to the activation function for that

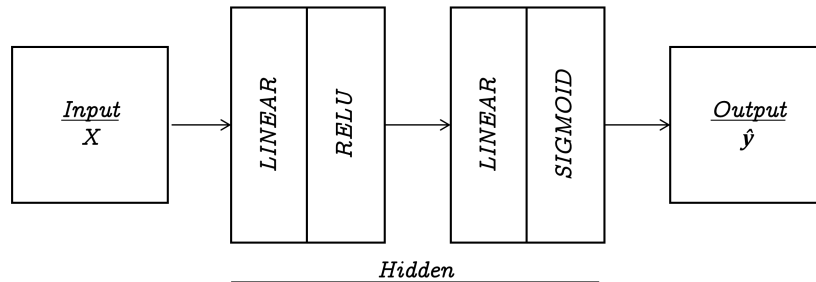


Figure 3.7: A deep neural network

layer. Then it propagated back one layer. It determines the partial derivative of the preceding layer's activation function with respect to the activation function of the current layer. Additionally, it computes the cost function's partial derivative with regard to the layer's activation function. The values are multiplied together. The same method is repeated up until it reaches the input layer. This is a neural network's backward propagation algorithm.

3.2.2 Activation Function

In neural network, activation function mainly describes how a node in a layer of the network transform the sum of transpose weight of the input layer in an output layer.

So far, we have been using the sigmoid activation.

$$\sigma(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

Some other much better options:

- *Tanh*: It stands for hyperbolic tangent functions which is actually a non-linear activation function that almost always works better than the sigmoid function.

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

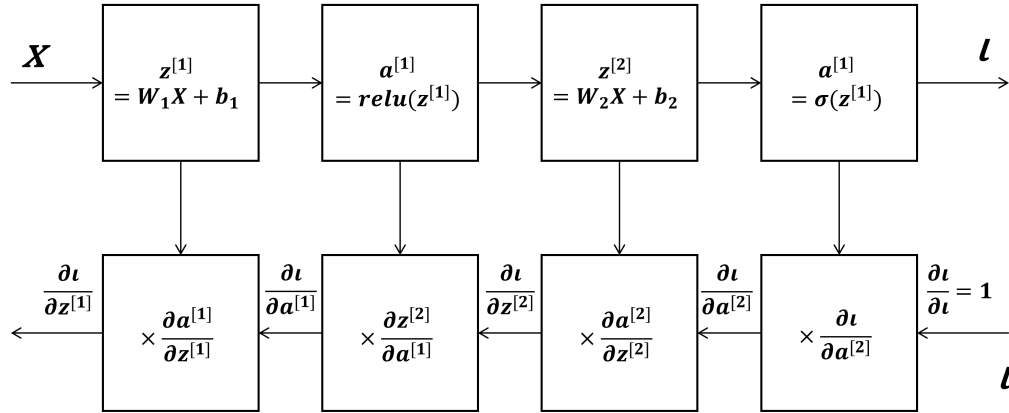


Figure 3.8: Forward & backward propagation for the network

The *tanh* function is really just a shift of the sigmoid function so that it crosses through the origin. Although, the *tanh* activation function usually works better than sigmoid activation function for hidden units because the mean of its output is closer to zero and so it centers the data better for the next layer.

- *ReLU*: It stands for rectified linear unit which mainly an activation function solves the disappearing gradient (here gradient is the slope that we know) problem faced by *tanh* and *sigmoid* activation function. In practice, it also leads to fast learning.

$$ReLU(z) = \max(0, z)$$

Mainly, based on the deep learning calculation, we use the above three activation function according to our necessity of use and choice. Although, easier we used sigmoid function to make the basic learning clear and understandable.

3.2.3 Convolutional Neural Networks

Convolutional networks, also known as convolutional neural networks or CNNs, are a specialized kind of neural network for processing data that has a known, grid-like topology.

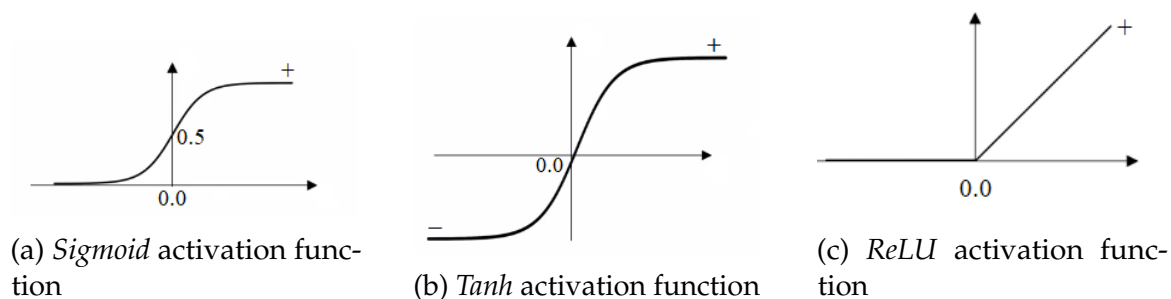


Figure 3.9: Different types of activation functions

Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

3.2.3.1 The Convolution Operation

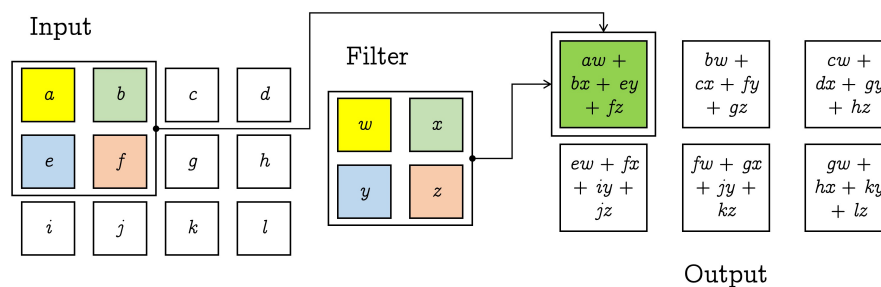


Figure 3.10: The convolution operation: iteration 1

Convolution Operation: We place the first element of the filter on top of the first element of the image. It overlaps the filter with a portion of the image. Then we perform element wise multiplication and get the summation of all the multiplied values. It becomes the first element of the output image. We perform the same task by stepping right and below and create a new output image.

Convolution operation is one of the fundamental building blocks of a convolutional neural network. This operation can be well-demonstrated using an example of edge detection. It is discussed in the Section 3.2.3.2.

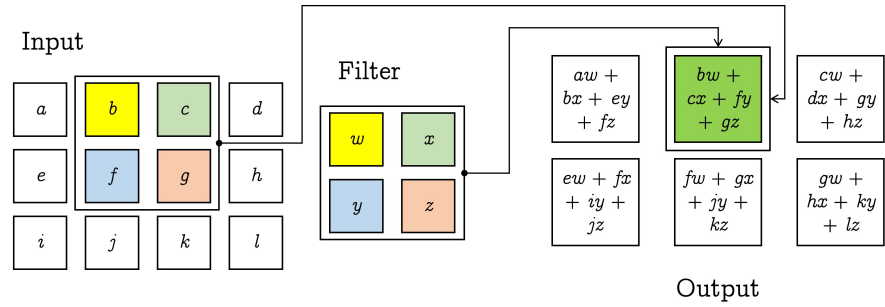


Figure 3.11: The convolution operation: iteration 2

3.2.3.2 Edge Detection

Let's take a sample image as example,

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix}$$

The above gray scale image has a one vertical line at the very center of the image. The color of the left half of the image can be thought as white and the color of the right half of the image can be thought as gray.

The two most common edge detecting filter kernels are: Sobel filter & Scharr filter.

The Sobel filter: $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ & $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$.

And, the Scharr filter: $\begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix}$ & $\begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$.

The convolution operation over the image with a filter kernel can produce an edge detected version of the original image.

3.2.3.3 Vertical Edge Detection

The filter kernel to detect vertical edges is $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$.

We can detect vertical edges by performing convolution operation in the taken image with the filter kernel. The operation would be as follows,

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$

3.2.3.4 Horizontal Edge Detection

The filter kernel to detect horizontal edges is $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$.

We can detect horizontal edges by performing convolution operation in the taken image with the filter kernel. The operation would be as follows,

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Examine the output image. As the image has no horizontal edges, we didn't find any edges in the output image as well!

3.2.3.5 Vertical and Horizontal Edge Detection

Let's take a new image example so that it covers both vertical and horizontal edges. The following image has both vertical and horizontal edges.

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{bmatrix}$$

Let's say we want to detect *horizontal edges* on the above image. So, the convolution operation for the task would be as follows:

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 30 & 10 & -10 & -30 \\ 30 & 10 & -10 & -30 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Similarly, we can detect *vertical edges* from the image as well. The convolution operation for the task would be as follows:

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 10 & 10 & 0 \\ 0 & -10 & -10 & 0 \\ 0 & -30 & -30 & 0 \end{bmatrix}$$

3.2.3.6 Padding

Padding is the process of expanding an image. We use *zero padding* most commonly. It adds layers of zeros around the border of an image.

Let's say, we have this image: $\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$.

A zero padded version of this image would be: $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

Padding resolves two of the very important problems in convolutional neural networks. They are: 1) shrinking output image, and 2) throwing away information from the edges of the image.

The main benefits of padding are:

- It allows you to use a CONV layer without necessarily shrinking the height and width of the volumes. This is important for building deeper networks, since otherwise the height/width would shrink as you go to deeper layers. An important special case is the *same convolution*, in which the height/width is exactly preserved after one layer.

- It helps us keep more of the information at the border of an image. Without padding, very few values at the next layer would be affected by pixels at the edges of an image.

3.2.3.7 Valid & Same Convolutions

Valid Convolutions: If we convolve an image without using any padding on the image, this convolution would be called as *valid convolutions*.

If a $n \times n$ image convolves with a $f \times f$ filter, the output image would be $(n - f + 1) \times (n - f + 1)$.

Same Convolutions: If we convolve an image with such a padding so that the output image has the same height and width as the input size, the convolution would be called as *same convolutions*.

If a $n \times n$ image having a padding, p convolves with a $f \times f$ filter, the output image would be $(n + 2p - f + 1) \times (n + 2p - f + 1)$. And, padding, p would be equal to $\frac{f-1}{2}$.

3.2.3.8 Strided Convolution

Let's examine the following operation. Here we have done our usual convolution operation that we have shown in the above examples. But, instead of taking one step to the right (one step to the below as well), we have taken two steps. And, generated the final output image. The stride in this case would be, $s = 2$. We can calculate the size of the output image using equation (3.1) shown in the Section 3.2.3.9.

$$\begin{bmatrix} 2 & 3 & 7 & 4 & 6 & 2 & 9 \\ 6 & 6 & 9 & 8 & 7 & 4 & 3 \\ 3 & 4 & 8 & 3 & 8 & 9 & 7 \\ 7 & 8 & 3 & 6 & 6 & 3 & 4 \\ 4 & 2 & 1 & 8 & 3 & 4 & 6 \\ 3 & 2 & 4 & 1 & 9 & 8 & 3 \\ 0 & 1 & 3 & 9 & 2 & 1 & 4 \end{bmatrix} * \begin{bmatrix} 3 & 4 & 4 \\ 1 & 0 & 2 \\ -1 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 91 & 100 & 88 \\ 69 & 91 & 117 \\ 44 & 72 & 74 \end{bmatrix}$$

3.2.3.9 Summary of Convolutions

If we want to convolve a $n \times n$ image with a $f \times f$ filter (we are using padding, p and stride, s), the output image size would be,

$$\left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \quad (3.1)$$

The equation (3.1) represents the output image size. The image that we have used in the previous section is of 7×7 dimension ($n = 7$). The filter we have used is of 3×3 dimension ($f = 3$). No padding was used ($p = 0$). And, stride was, $s = 2$. Using equation (3.1), we can calculate the output image size.

$$\begin{aligned} \text{Output image size} &\Rightarrow \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \\ &\Rightarrow \left\lfloor \frac{7 + 2 \times 0 - 3}{2} + 1 \right\rfloor \times \left\lfloor \frac{7 + 2 \times 0 - 3}{2} + 1 \right\rfloor \\ &\Rightarrow \left\lfloor \frac{4}{2} + 1 \right\rfloor \times \left\lfloor \frac{4}{2} + 1 \right\rfloor \\ &\Rightarrow 3 \times 3 \end{aligned}$$

3.2.3.10 Convolution Over Volume

If we want to detect features in RGB image, we have to consider convolution over three dimensional volumes. We will have an image of 3 channels and we will need a filter with 3 channels as well. Consider Figure 3.12.

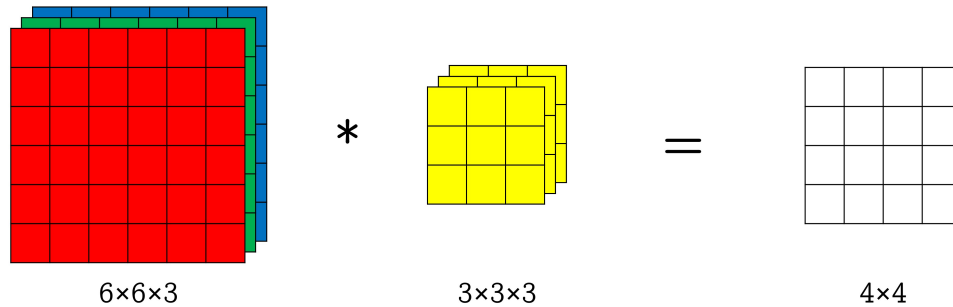


Figure 3.12: Convolutions on RGB image

We have a 6×6 2D image of 3 channels. So, basically, we have a $6 \times 6 \times 3$ volume. And, we have a $3 \times 3 \times 3$ filter to convolve over that 3D image. From the

filter, we are seeing that we have basically 27 values. Similar to the 2D convolution, we can put that filter on top of the first element of the 3D image (Figure 3.13). It will overlap the 3×3 portion of the first matrix. And, as it also has 3 channels, it will automatically overlaps over to all the 3 channels of the original image to match all the 27 values. Then, similar to the 2D convolution, we can perform element wise multiplication and get the summation of those values. This will give us the first element value of the output image. We continue the operation by taking steps to the right and below to tabulate the 4×4 output image. Figure 3.13 summarizes the operation.

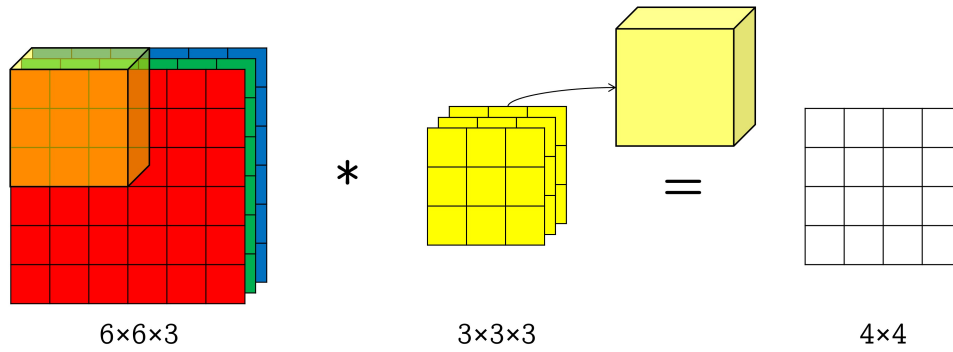


Figure 3.13: The convolution operation

If we convolve a $(n \times n \times n_c)$ image with a $(f \times f \times n_c)$ filter, we will get a $(n - f + 1) \times (n - f + 1)$ output image.

3.2.3.11 Use of Multiple Filters

We can use multiple filters over an RGB image. Consider Figure 3.14, we have two $3 \times 3 \times 3$ filters. If we convolve it with a $6 \times 6 \times 3$ image, we will get a $4 \times 4 \times 2$ volume.

So, to summarize, if we convolve a $(n \times n \times n_c)$ image with n'_c number of $(f \times f \times n_c)$ filters, we will get a $(n - f + 1) \times (n - f + 1) \times n'_c$ output image volume. Here, n_c = number of channels.

3.2.3.12 One layer of a convolutional network

Get back to the Figure 3.14, we have got two 4×4 output matrices (O_1, O_2) after performing convolution with the original image. Now, let's do some more stuff.

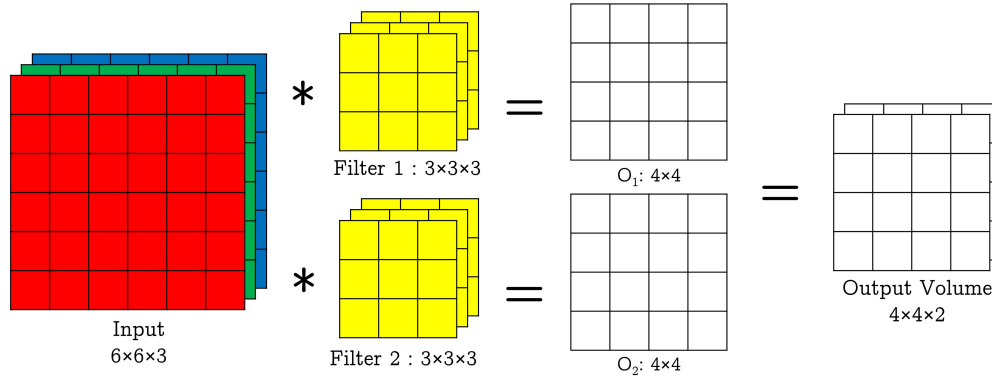


Figure 3.14: The convolution operation with multiple filters

First, add a bias (a real number) to O_1, O_2 . Then, apply a *ReLU* non-linearity to it. So, we are going to get, $ReLU(O_1 + b)$ and $ReLU(O_2 + b)$. And, finally, after stacking the outputs, we will get the final $4 \times 4 \times 2$ output volume. This computational process of getting a $4 \times 4 \times 2$ output from the $6 \times 6 \times 3$ input is one layer of convolutional neural network.

3.2.3.13 Analogy with a Neural Network

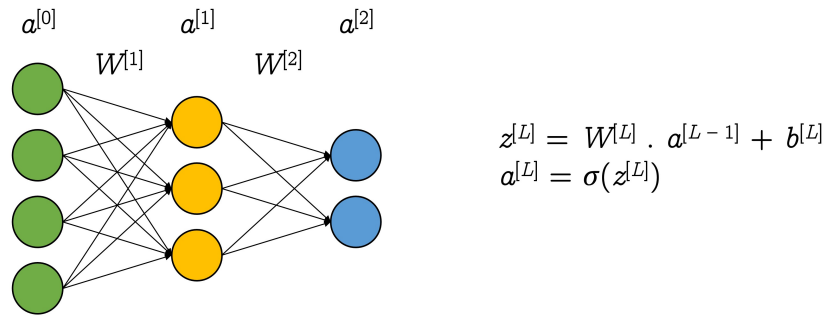


Figure 3.15: Neural network representation

Recall the neural network activation at layer 1 of a neural network (Figure 3.15),

$$z^{[1]} = W^{[1]} \cdot a^{[0]} + b^{[1]}$$

$$a^{[1]} = g(z^{[1]})$$

Here, $W^{[1]}$ is the layer 1 weight matrix, $a^{[0]}$ is the input layer which is equals

to X , and $b^{[1]}$ is the layer 1 bias unit. $g()$ is the sigmoid activation function. Let's compare the representation with the Figure 3.14.

The input volume acts similarly to the $a^{[0]}$ or the input layer of the neural network. Filter 1 and filter 2 combinedly acts as the weight, $W^{[1]}$. And, the output volume is activation of layer 1, $a^{[1]}$. The process of getting the output volume is already described in the previous section. It is completely similar to the sigmoid activation of a neural network.

3.2.3.14 Notational Convention

If layer l is a convolution layer:

$$\begin{aligned}
 \text{Input} &: n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]} \\
 \text{Output} &: n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]} \\
 f^{[l]} &= \text{filter size} \\
 p^{[l]} &= \text{padding} \\
 s^{[l]} &= \text{stride} \\
 n_c^{[l]} &= \text{number of filters} \\
 \text{Each filter is} &: f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \\
 \text{Activations} &: a^{[l]} \rightarrow n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]} \\
 \text{Weights} &: f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]} \\
 \text{Bias} &: n_c^{[l]}
 \end{aligned}$$

3.2.3.15 Simple deep CNN example

Let's take the example in Figure 3.16. In the layer 0 or the input layer ($a^{[0]}$ or X), it has a $39 \times 39 \times 3$ image. So, according to our notational convention, $n_H^{[0]} = n_W^{[0]} = 39, n_c^{[0]} = 3$. To get the first output at layer 1, we have set a filter size of 3, stride size of 1, padding size of 0, and 10 number of filters². So, according to the notational convention, $f^{[1]} = 3, s^{[1]} = 1, p^{[1]} = 0, n_c^{[1]} = 10$.

Using the equation (3.1), we will get the output at layer 1, $\left\lfloor \frac{39+2 \times 0-3}{1} + 1 \right\rfloor \times \left\lfloor \frac{39+2 \times 0-3}{1} + 1 \right\rfloor \Rightarrow 37 \times 37 \times 10$.

²These are hyperparameters for a ConvNet. By definition, hyperparameters are parameters whose values are set before starting the model training process.

So, in the layer 1, $n_H^{[0]} = n_W^{[0]} = 37, n_c^{[0]} = 10$. To get the output at layer 2, we have set the hyperparameters, $f^{[2]} = 5, s^{[2]} = 2, p^{[2]} = 0, n_c^{[2]} = 20$. And, we will get the output using equation (3.1), $a^{[2]} \Rightarrow 17 \times 17 \times 20$. As we have set a stride greater than 1, the dimension shrunk much faster.

Similarly, we have got the layer 3 output, $a^{[3]} \Rightarrow 7 \times 7 \times 40$. It's the final output volume for the example. Finally, to complete the ConvNet, feed the layer 3 output into a logistic regression unit, or a Softmax unit after flattening the output into a $7 \times 7 \times 40 = 1960$ units long vector (Figure 3.17).

We have seen convolution type of convolutional layer in this example. There can be several types of layers that can be used in a Convolutional Neural Network:

- Convolution layer (*CONV*)
- Normalization layer (*NORM*)
- Pooling layer (*POOL*)
- Activation layer
- Fully connected layer (*FC*)
- Dropout layer

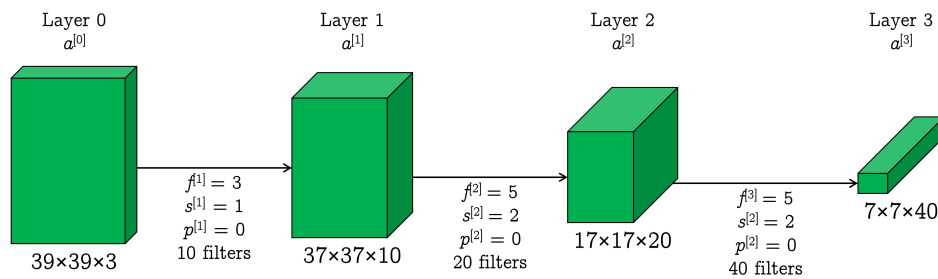


Figure 3.16: Simple deep CNN example

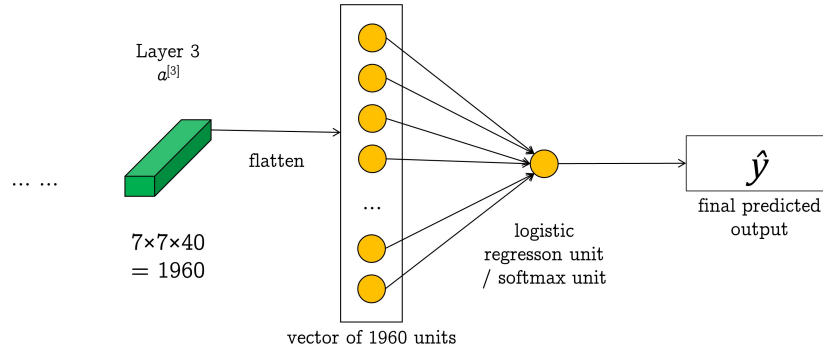


Figure 3.17: Simple deep CNN example: feeding into neural network

3.2.3.16 Pooling Layers

The pooling (POOL) layer reduces the height and width of the input. It helps reduce computation, as well as helps make feature detectors more invariant to its position in the input.

ConvNets often use pooling layers to reduce the size of the representation, to speed the computation, as well as make some of the features that detects a bit more robust.

3.2.3.17 Max Pooling

Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

It slides an $f \times f$ window over the input and stores the max value of the window in the output. From the following 4×4 image, we performed max pooling considering a filter size, $f = 2$, and stride size, $s = 2$. The operation takes the maximum value from a 2×2 region. Using the equation (3.1), we get the output size would be, 2×2 .

$$\begin{bmatrix} 1 & 3 & 2 & 1 \\ 2 & 9 & 1 & 1 \\ 1 & 3 & 2 & 3 \\ 5 & 6 & 1 & 2 \end{bmatrix} \xrightarrow[f=2, s=2]{\text{max pooling}} \begin{bmatrix} 9 & 2 \\ 6 & 3 \end{bmatrix}$$

In the following example, we have a 5×5 image. If we perform max pooling considering filter, $f = 3$, stride, $s = 1$ (zero padding), we will get a $\frac{5+0-3}{1} \times \frac{5+0-3}{1}$

or 3×3 output. We will perform max pooling in the 3×3 window of the image and take one step left and one step below.

$$\begin{bmatrix} 1 & 3 & 2 & 1 & 3 \\ 2 & 9 & 1 & 1 & 5 \\ 1 & 3 & 2 & 3 & 2 \\ 8 & 3 & 5 & 1 & 0 \\ 5 & 6 & 1 & 2 & 9 \end{bmatrix} \xrightarrow[f=3, s=1]{\text{max pooling}} \begin{bmatrix} 9 & 9 & 5 \\ 9 & 9 & 5 \\ 8 & 6 & 9 \end{bmatrix}$$

3.2.3.18 Average Pooling

Average pooling is not very commonly used pooling. It performs the same operation stated in the earlier section. It takes the average of that 2×2 window instead of taking the maximum value.

$$\begin{bmatrix} 1 & 3 & 2 & 1 \\ 2 & 9 & 1 & 1 \\ 1 & 3 & 2 & 3 \\ 5 & 6 & 1 & 2 \end{bmatrix} \xrightarrow[f=2, s=2]{\text{average pooling}} \begin{bmatrix} 3.75 & 1.25 \\ 4 & 2 \end{bmatrix}$$

3.2.3.19 Deep CNN Example

The Figure 3.18 shows a complex CNN-based digit recognizer. It was a proposed model by [LeCun et al. \(1998\)](#). It shows how few convolution, pooling, and fully-connected layers can extract the features of handwritten digits to learn the trainable parameters so that it can further predict any digit from an image. The following model has 2 conv-layer, and 2 pooling layer. And, it has 3 fully-connected layer. The final fully-connected layer connects with a *softmax* activation function to predict the digit from an image.

3.3 Residual Networks (ResNets)

ResNets are derived from native CNNs. ResNets learn residual functions with reference to the layer inputs instead of learning unreferenced functions; which makes it different from other CNNs. With residual nets, these layers can fit a residual mapping as opposed to expecting that each of the few stacked layers directly matches the intended underlying mapping. They stack residual blocks on top of each other to form network. This architecture has been proposed by [He et al. \(2016\)](#) to solve image classification problem.

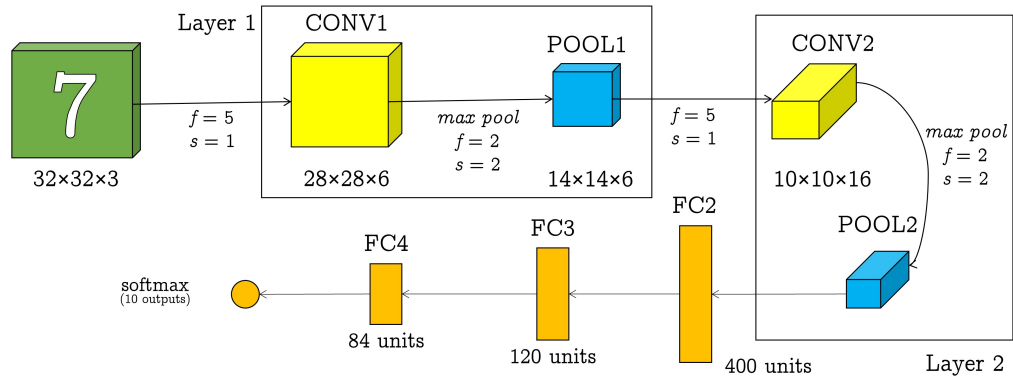


Figure 3.18: Deep CNN example

3.3.0.1 Vanishing/Exploding Gradient

In a network of n hidden layers, n derivatives will be multiplied together. If the derivatives are large then the gradient will increase exponentially as we propagate down the model until they eventually explode, and this is what we call the problem of *exploding gradient*. Alternatively, if the derivatives are small then the gradient will decrease exponentially as we propagate through the model until it eventually vanishes, and this is the *vanishing gradient* problem.

3.3.0.2 Skip Connections

Skip connections (or shortcut connection) allows to take the activation from one layer and feed that output as the input of another layer of a deeper neural network. Information can move more readily up the network with the use of skip connections, which let layers link to levels higher up the network and skip layers in between. The skip connection method addresses the exploding/vanishing gradient problem. As it has the ability go further the neural network, it can descent to its gradient more quickly than expected.

Deep neural networks have the advantage over shallow neural networks in that they can learn complex functions more quickly. As the depth of the architecture is increased during deep learning, the model's performance degrades. This issue is referred to as Degradation Problem. The main reason of this degradation problem is the problem of overfitting. The model typically overfit as depth increases, but not in this instances. The shallower model performs better than the deeper one.

3.3.0.3 How Residual Block Implements Skip Connections

The skip connection bypasses some levels in between to link layer activations to subsequent layers. This creates a leftover block. These leftover blocks are stacked to create ResNets. The benefit of including this kind of skip link is that regularization will skip any layer that degrades architecture performance. As a result, training an extremely deep neural network is possible without encountering issues with vanishing or expanding gradients. Skip Connections were developed to address various issues in various architectures. Skip connections allowed us to address the degradation issue in the case of ResNets, while it ensured feature reusability in the case of DenseNets.

In ResNets, matrix addition is used to transmit data from initial layers to deeper layers. The output from the preceding layer is added to the layer ahead, therefore this operation doesn't require any further arguments. A single residual block has been shown in the Figure 3.19. If we denote the desired underlying mapping of one layer of convolutions as $\mathcal{H}(x)$, the non-linear layers which are stacked to the architecture maps as $\mathcal{F}(x) := \mathcal{H}(x) - x$. Here, the $\mathcal{F}(x)$ forms the residual block. The figure of the residual block has been shown in the Figure 3.19. The architectural diagram of a ResNet-18 model has also be shown in the Figure 3.20.

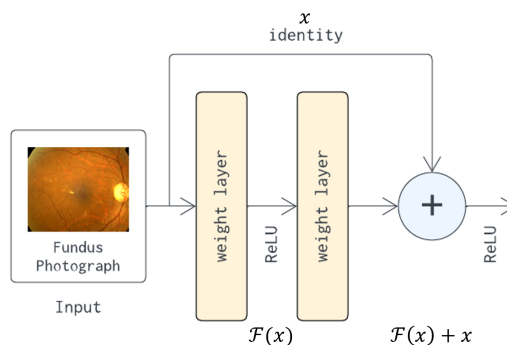


Figure 3.19: Skip connection using a residual block [He et al. \(2016\)](#)

3.3.0.4 Use of Bottleneck Layers in Residual Block

The bottleneck in a neural network is a layer with fewer neurons than the layer below or above it. It can be use to get a representation of the input that has fewer dimensions. Bottleneck residual blocks are used for deeper ResNets as it reduces

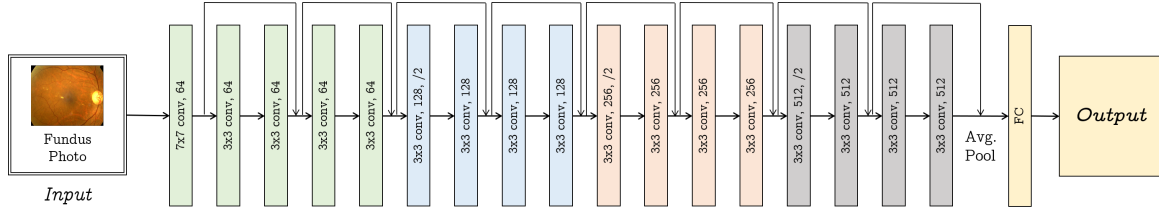


Figure 3.20: The architecture of the ResNet-18 model [He et al. \(2016\)](#)

the computational complexity. In a CNN, bottleneck layers are added to reduce the number of feature maps or channels in the network which tend to increase in each layer. This is achieved by using 1×1 convolutions with fewer output channels than input channels. Because of the bottleneck layers in between of the other layers, the dimensions of the other layers get skewed; makes the training time more inexpensive. The representation of a bottleneck layer has been shown in the Figure 3.21. Bottleneck layer uses depth-wise separable convolution which is described in the Section 3.4.1.

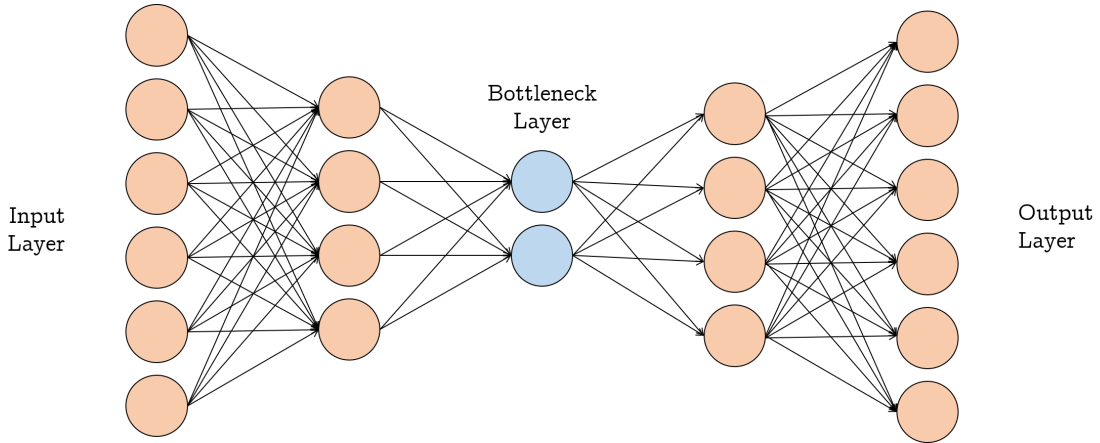


Figure 3.21: The bottleneck layer

3.4 EfficientNet

EfficientNet is a convolutional neural network design and scaling technique that uses a compound coefficient to consistently scale all depth, breadth, and resolution dimensions. The EfficientNet scaling method evenly scales network breadth, depth,

and resolution using a set of present scaling coefficients, in contrast to standard practice, which scales these variables arbitrarily.

3.4.1 Normal Convolution vs Depth-wise Separable Convolution

We have described the normal convolution operation in the Section 3.2.3.1. Normal convolution has a very high computational complexity. [Howard et al. \(2017\)](#) had proposed depth-wise convolution to overcome this problem. And, this depth-wise convolution is the heart of MobileNet and EfficientNet architecture.

Depth-wise Separable Convolution: Depth-wise Seperable Convolution can be computed in two parts. They are *depth-wise convolutions* and *point-wise convolutions*.

- *Depth-wise Convolution:* Unlike the normal convolution, convolution is applied to a single channel at a time in depth-wise convolution. So, if an $n \times n \times n_c$ image convolves with n_c number of $f \times f$ filters, the output of the convolution will be $n_{out} \times n_{out} \times n_c$. And, the computational cost will be = dimension of filter parameters $\times$ dimension of filter positions $\times$ number of filters.
- *Point-wise Convolution:* In point-wise convolution, a 1×1 conv layer convolves with the input image. So, if an $n_{out} \times n_{out} \times n_c$ image have n'_c channels, the filter size will be $1 \times 1 \times n'_c$. And, the output of the convolution will be $n_{out} \times n_{out} \times n'_c$. Therefore, the computational cost will be = dimension of filter parameters $\times$ dimension of filter positions $\times$ number of filters.

Therefore, the computation cost in depth-wise separable convolution is $\frac{1}{n'_c} + \frac{1}{f^2}$ times cheaper than normal convolution.

3.4.2 MobileNet Architecture

MobileNet architecture was proposed by [Howard et al. \(2017\)](#) which was initially called as MobileNetV1. The foundation of this architecture was based on a streamlined architecture using depth-wise separable convolution operation. It ensures to run on a computationally restrained devices like mobile or any other embedded devices. That MobileNetV1 had the implementation of depth-wise separable convolution only in its convolutional network. That network was further redesigned to get more computational efficiency. In the MobileNetV2 architecture, [Sandler et al.](#)

(2018) introduced the concept of bottleneck layer by integrating residual connection with the depth-wise separable convolution operation.

3.4.3 The Architecture of EfficientNet

EfficientNet is a convolutional neural network design and scaling technique that uses a compound coefficient to consistently scale all depth, breadth, and resolution dimensions. The EfficientNet scaling method evenly scales network breadth, depth, and resolution using a set of preset scaling coefficients, in contrast to standard practice, which scales these variables arbitrarily. For instance, in order to use 2^n more computational resources, we need to increase the depth of the network by α^n , width by β^n , and image dimension by γ^n . The constant coefficients (α, β, γ) are determined by a grid search on the base model. To uniformly scale network width, depth, and resolution, a compound coefficient ϕ has been used by the EfficientNet architectures. The rationale behind the *Compound Scaling* method is that larger input images require more layers in order to expand the network's receptive field and more channels in order to capture more fine-grained patterns on the larger picture. In addition to squeeze-and-excitation blocks, the foundational EfficientNet-B0 network is built upon the MobileNetV2 inverted bottleneck residual blocks. On the CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, EfficientNets likewise transfer well and achieve state-of-the-art accuracy while using orders of magnitude fewer parameters. The architecture of the baseline EfficientNet architecture has given in the Figure 3.22.

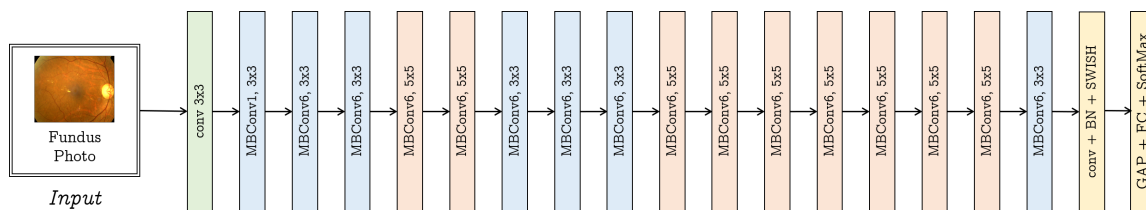


Figure 3.22: The architecture of EfficientNet Tan & Le (2019)

3.5 Vision Transformer (ViT)

A methodology for classifying images called the Vision Transformer, or ViT, applies a Transformer like design to selected areas of the image. A sequence of vectors is

created by dividing a picture into fixed-size patches, linearly embedding each one, adding position embeddings, and then feeding the assembled vectors to a conventional Transformer encoder. The traditional method of performing classification involves including an extra learnable ‘classification token’ in the sequence. The architecture of the Transformer Encoder of the Vision Transformer model is shown in the Figure 3.23. The further details of the Vision Transformer architecture is given in the Section 4.2.3.

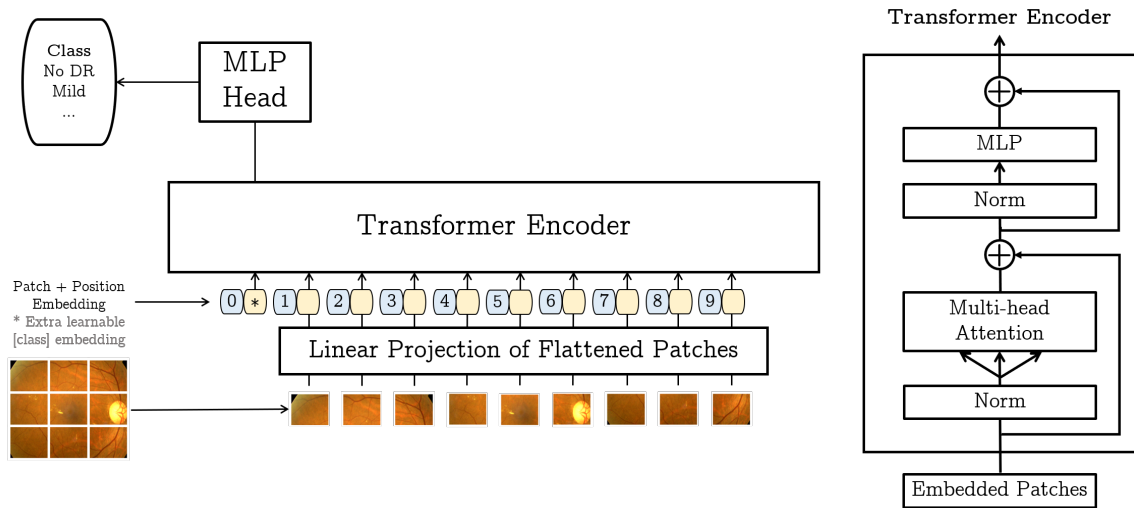


Figure 3.23: The architecture of the Vision Transformer Endcoder part [Dosovitskiy et al. \(2020\)](#)

Table 3.1: Summary of the features of different datasets available

Datset	Image Count	Resolution	Label Count	Train + Val/Test Size
APTOS (2019) Tymchenko et al. (2020)	3662	Varies	0: 1805, 1: 370, 2: 999, 3: 193, 4: 295	—
MESSIDOR-1 Gangwar & Ravi (2021)	1200	1440×960, 2240×1488, 2304×1536	—	—
IDRID Porwal et al. (2018)	516	4288×2848	—	413 + 103
EyePACS Saxena et al. (2020)	88702	1440×960, 2240×1488, 2304×1536, 4288×2848	0: 25810, 1: 2443, 2: 5292, 3: 873, 4: 708	35126 + 53576
MESSIDOR-2 Abràmoff et al. (2016)	1748	1440×960, 2240×1488, 2304×1536	—	—
DIARETDB0 Kauppi et al. (2006)	130	1500×1152	No DR: 20, DR: 110	Varies
DIARETDB1 Gondal et al. (2017)	89	1500×1152	No DR: 5, DR: 84	28 + 61
STARE Fu et al. (2016)	400	605×700	—	—
E-ophtha Wankhede & Khanchandani (2020)	381	2048×1360	—	—
UoA-DR Abdulla & Chalakkal (2018)	200	2124×2056	—	—

Chapter 4

Experimental Results and Discussion

4.1 Experimental Setup

We have developed a proper experimental setup to experiment with our machine learning models. As a computer vision problem with a dataset of very high-resolution images, we have required a robust hardware setup so that we can train our model properly. We required a Hard Disk (HDD) with a large amount of storage as we have a huge dataset to experiment with. We also used a Solid State Drive (SSD) to speed up our operations faster. Before diving into Graphics Processing Unit (GPU) requirements, we had worked to make our PC works faster. For that, we have used a 32 Gigabytes Random Access Memory (RAM). RAM was so necessary for handling our PC faster. Then, we fall into trouble in choosing our required CPU as both Intel and Ryzen have their own positive sides in the CPU industry. After performing research on CPUs, we have found that an Intel Core i7 CPU will handle much of our requirement in handling deep learning models. And, the final thing we chose was the Graphics Processing Unit (GPU). We chose a GPU that supports the CUDA environment. CUDA is such a platform that supports parallel computing. We used the PyTorch framework for developing our deep-learning models. PyTorch supports the CUDA environment to speed up computational operations. So, these are things we have focused on to develop our experimental setup. We have used Google Colab and Kaggle to quickly train and run our model.

4.2 Experimental Result and Analysis

Our objective was to develop a deep-learning model with better accuracy and fewer Floating Point Operations (FLOPs) for deploying the model more efficiently. That is why we started our model experiment with EfficientNet-B0 as it has fewer trainable parameters than other state-of-the-art (SOTA) CNN architectures [Tan & Le \(2019\)](#).

We used the 'Early Stopping' regularization technique for all the experiments given below to prevent overfitting and save computational resources whenever we noticed that the test performance has reached a plateau or stabilized means the model is no longer improving with further training [Prechelt \(2012\)](#).

4.2.1 The Experimental Standard We Followed

4.2.1.1 Evaluation of the Models

We have used the *F1score* instead of accuracy for evaluating the performance of the model which is the harmonic mean of precision and recall [Yacouby & Axman \(2020\)](#). Precision is the number of true positive predictions divided by the total number of positive predictions, while recall is the number of true positive predictions divided by the total number of actual positive samples. For medical datasets, it's important to use the *F1score* instead of accuracy because medical datasets often have an imbalanced class distribution, meaning there are far fewer positive samples than negative samples. In this case, accuracy can be misleading because a model could achieve high accuracy by simply predicting most samples as negative. The *F1score* takes both precision and recall into account, providing a more balanced evaluation of the model's performance. In medical applications, it's crucial to have both high precisions (to avoid false positive predictions) and high recall (to ensure that all positive cases are detected), and the *F1score* provides a single metric that balances both of these requirements [Ramana & Boddu \(2019\)](#). The equations of all the evaluation matrices are given below:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{False Negatives} + \text{False Positives} + \text{True Negatives}}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$F1score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where,

- True Positives (TP) are the number of instances correctly predicted as positive.
- True Negatives (TN) are the number of instances correctly predicted as negative.
- False Positives (FP) are the number of instances incorrectly predicted as positive.
- False Negatives (FN) are the number of instances incorrectly predicted as negative.

4.2.2 EfficientNet

We have used EfficientNet-B0 because it is a state-of-the-art CNN architecture (achieved best result on CIFAR-10 [Kondratyuk et al. \(2020\)](#)) that has been designed to achieve high performance with a relatively small number of parameters and it is the base version of the EfficientNet family.

The EfficientNet model uses a combination of depth-wise separable convolutions and global average pooling, which makes the model more efficient and accurate. The depth-wise separable convolutions allow the model to perform convolution on each channel separately, which reduces the computation and memory requirements. The global average pooling is used to reduce the spatial dimensions of the feature map and make the model more robust to translation [Hsiao et al. \(2019\)](#). EfficientNet also uses a weight scaling technique to balance the depth and width of the network and a compound scaling method to scale the resolution, width and depth together. The compound scaling method allows the model to be more efficient and accurate while using fewer parameters.

We have pre-processed the images by resizing them to 224x224, applying random horizontal flipping with a probability of 0.4 as image transformation, and normalizing the pixel values using the mean and standard deviation (mean = 0.485,

0.456, 0.406, std = 0.229, 0.224, 0.225) of the ImageNet benchmark dataset [Krizhevsky et al. \(2017\)](#). We have split the dataset into a training set (80%) and a test set (20%).

We have used the EfficientNet-B0 model and modified the fully connected layer to have 5 output features, corresponding to the 5 classes in our dataset. However, we have trained the model from scratch without using the ImageNet weights.

While studying the hyperparameters before training the models, we found some valuable insights based on our literature review which are given below:

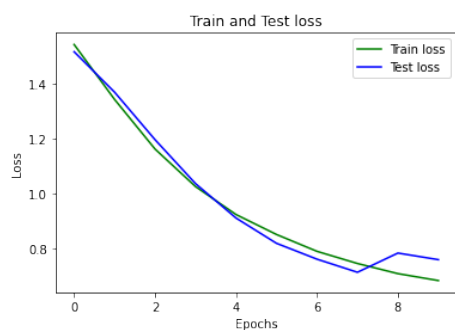
- Since the batch size determines the regularization strength of batch normalization, a small batch size can result in high variance that leads to the overfitting issue [Yong et al. \(2020\)](#). On the other hand, a large batch size can result in a more stable model, but it may also lead to underfitting if the model is not able to learn the fine-grained details of the data. So, after rigorous experiments, we decided to use batch size 64 as it was more stable than batch size 32, and 128 while training.
- We have used Cross-entropy loss as the loss function because it performs comparatively well in multi-class classification as it is a convex function, which makes it easier to optimize and reduces the risk of getting stuck in poor local minima [Semenov et al. \(2019\)](#).
- We have used Adam optimizer as it uses adaptive learning rates, which means that the learning rates are adjusted for each parameter based on the historical gradient information [Bae et al. \(2019\)](#). This allows the optimizer to respond to changes in the gradient information and makes it more robust to noisy gradients.
- We have used the ReLU activation function as it sets all negative activations to zero, leading to many zero activations in the model. This results in reduced computation and improved model interpretability [Agarap \(2018\)](#).
- We have used the dropout regularization technique to prevent overfitting and improve the generalization ability of the model. This is achieved by randomly dropping out (setting to zero) a certain percentage of neurons during training [Gal & Ghahramani \(2016\)](#).

- Momentum is a hyperparameter used in optimization algorithms like Adam. It helps to accelerate the convergence of the optimizer by adding a fraction of the update vector of the past time step to the current update vector [Sutskever et al. \(2013\)](#). After a rigorous experiment, we decided to use a momentum of 0.01 as it was performing better than 0.1 and 0.001.
- Learning rate is a hyperparameter that controls the step size of the optimizer. It determines the amount by which the weights are updated in the gradient direction. A smaller learning rate will cause the model to converge slowly but with a lower risk of overshooting the optimal solution. In contrast, a larger learning rate will cause the model to converge quickly but with a higher risk of overshooting the optimal solution [Wu et al. \(2019\)](#). We have used a learning rate of 0.0001 after passing through a number of trials and errors.
- We have also used a learning rate scheduler with a step size of 5 and a gamma of 0.1 that means after every 5 epochs, the initial learning rate will be updated by multiplying with 0.1. The idea behind using the learning rate scheduler is that during the initial stages of training, a high learning rate allows the model to quickly explore the loss landscape and make significant changes to the parameters. However, as training progresses, the learning rate should be reduced to allow for fine-tuning of the parameters and prevent the model from overshooting the optimal solution [Lewkowycz \(2021\)](#).

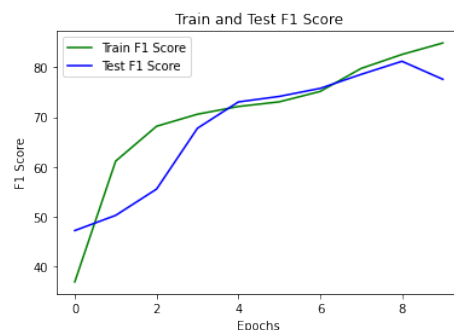
To summarize, for training and testing the model, we have used a batch size of 64, Cross-entropy loss as loss function, Adam optimizer, ReLU activation function, a dropout rate of 0.2 for regularization, and a momentum of 0.01. The initial learning rate was set to 0.0001, and the model was trained for 10 epochs.

The model achieved an *F1score* of 84% on the training set and 77.5% on the test set. The model has a total of 4 million trainable parameters and 0.39 billion FLOPs (floating-point operations).

However, we also implemented EfficientNet-B1 (a larger version of the EfficientNet base model) which has 6 million trainable parameters and 0.6 billion FLOPs but noticed that using this lightweight model, we could not achieve a significantly better result (*F1score* of 87% for the train set and 79% for the test set) as it has fewer trainable parameters to learn patterns.

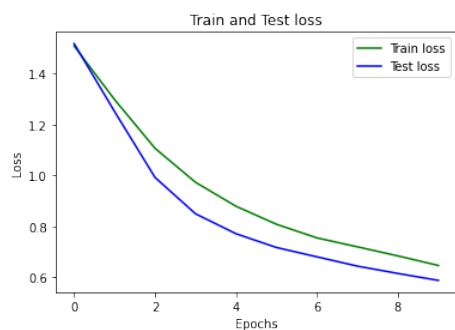


(a) Plot of Train and Test Loss vs Epoch in EfficientNet-B0

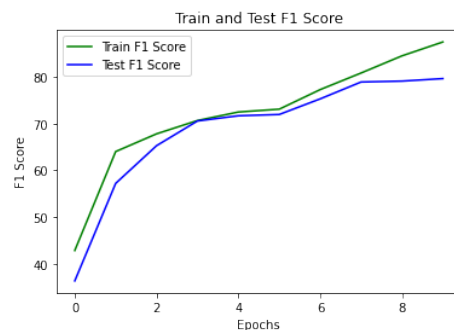


(b) Plot of Train and Test $F1score$ vs Epoch in EfficientNet-B0

Figure 4.1: Experimental Result of EfficientNet-B0



(a) Plot of Train and Test Loss vs Epoch in EfficientNet-B1



(b) Plot of Train and Test $F1score$ vs Epoch in EfficientNet-B1

Figure 4.2: Experimental Result of EfficientNet-B1

4.2.3 Vision Transformer (ViT-16)

We have also implemented Vision Transformer (ViT-16) for experiment purposes which is a transformer-based deep learning architecture for image classification tasks. It is a variant of the standard transformer architecture that was introduced by Google in the paper "An Image is Worth 16x16 Words" for natural language processing (NLP) tasks but has been adapted for computer vision tasks by using patches of image data as input tokens [Dosovitskiy et al. \(2020\)](#).

ViT-16 is a specific implementation of the Vision Transformer architecture, where "16" refers to the size of the patches used as input tokens. In this implementation, the input image is divided into a grid of patches (also known as embeddings), each of which is treated as a separate token. These tokens are then processed by the transformer encoder to extract high-level features and make predictions after passing through the decoder.

Here, the encoder is responsible for transforming the input image patches into a set of feature representations that capture the relationships between the image patches. The encoder typically consists of a series of self-attention and feedforward layers, which are designed to process the input data in parallel and capture both local and global dependencies in the input data.

The decoder, on the other hand, is responsible for taking the encoded features from the encoder and transforming them into the final prediction for the image. The decoder typically consists of a series of feedforward layers, which are used to aggregate the encoded features and make a prediction for the image class.

Together, the encoder and decoder form the complete ViT model, which can be trained end-to-end on an image classification task. During the training process, the encoder and decoder learn to extract high-level features from the input image patches and use these features to make accurate predictions for the image class.

However, while using the ViT-16 for our dataset, we preprocessed the dataset as stated earlier and changed the fully connected layer to 5 (corresponding to our 5 classes).

We used batch size 64, patch size 16 as mentioned previously, embedding size 768, cross-entropy loss as loss function, Adam optimizer, GeLU (Gaussian Error Linear Unit) as activation function that introduces non-linearity by modeling the input as a random variable with a Gaussian distribution and computing its

cumulative distribution function [Hendrycks & Gimpel \(2016\)](#), and learning rate 0.00003.

$$GeLU(y) = 0.5x \left(1 + \tanh \left(\sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right) \right)$$

We achieved an 84% *F1score* on the training set and 81% on the test set with 10 epochs. The model has 85M trainable parameters and 33.01 billion FLOPs which is very expensive in terms of computational resources. But we believe if we can increase the computational budget, we can come up with better results with fine-tuning and some architectural changes in the future.

Before finalizing the base model for our dataset in the next section, a brief summary of the EfficientNet family and Vision Transformer (ViT-16) implementation discussed so far is given below:

Table 4.1: Result summary of EfficientNet and Vision Transformer

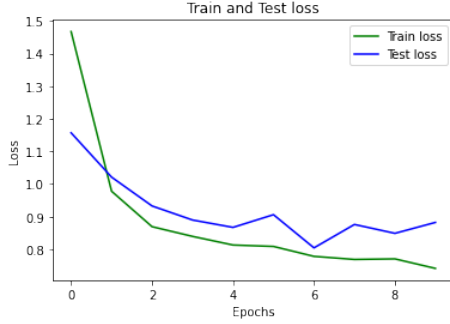
Models	Training <i>F1score</i>	Testing <i>F1score</i>	Epochs
EfficientNet-B0	84%	77.5%	10
EfficientNet-B1	87%	79%	10
ViT-16	84%	81%	10

Table 4.2: Computation Cost Summary of EfficientNet and Vision Transformer

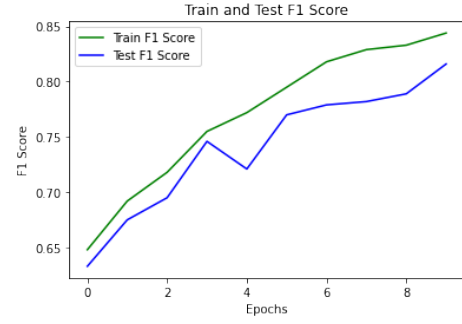
Models	Trainable Parameters (in million)	FLOPs (in billion)
EfficientNet-B0	4	0.4
EfficientNet-B1	6	0.70
ViT-16	85	33.01

4.2.4 Residual Network (ResNet)

As our objective was to develop a CNN architecture with fewer FLOPs, we moved towards the skip connection concept which is the main technique of ResNet [He et al. \(2016\)](#). One important point to be noted here is we implemented ResNet152, ResNet101, ResNet34, and ResNet18 in the same experimental setup, and finally, we decided to use ResNet 18 for further modification as we achieved a significant result along with the least number of FLOPs at this point. However, here we will discuss the implementation of the original ResNet18 architecture for our dataset.



(a) Plot of Train and Test Loss vs Epoch in ViT-16



(b) Plot of Train and Test *F1score* vs Epoch in ViT-16

Figure 4.3: Experimental Result of ViT-16

We preprocessed the dataset as stated earlier and changed the output features of the fully connected layer from 1000 to 5 as we have 5 classes (No DR, mild, moderate, severe, and proliferative) in our dataset.

We used batch size 64, Cross-entropy loss as loss function, Adam optimizer, ReLU as activation function, learning rate 0.0001, and trained the model for 30 epochs.

A brief summary of the ResNet family implementation for our dataset is given below:

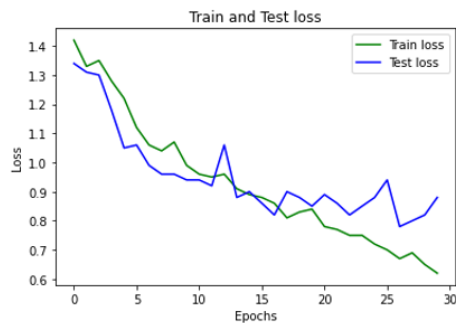
Table 4.3: Result Summary of ResNet Family

Models	Training <i>F1score</i>	Testing <i>F1score</i>	Epochs
ResNet-152	97%	85%	30
ResNet-101	89%	78%	30
ResNet-34	96%	83%	30
ResNet-18	94%	83%	30

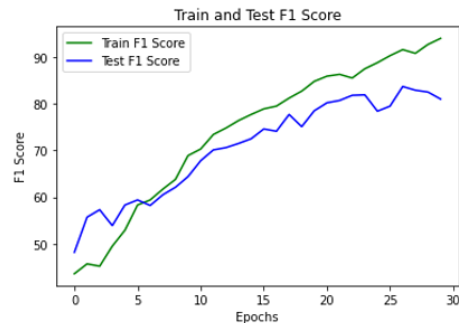
Table 4.4: Computation Cost Summary of ResNet Family

Models	Trainable Parameters (in million)	FLOPs (in billion)
ResNet-152	58	12
ResNet-101	43	8
ResNet-34	21	4
ResNet-18	10.9	1.79

When we compared ResNet-18 with ResNet-152, we noticed that ResNet-152 cost much more computation resources ($12B - 1.79B = 10.21B$) than ResNet-18,



(a) Plot of Train and Test Loss vs Epoch in ResNet-18



(b) Plot of Train and Test *F1score* vs Epoch in ResNet-18

Figure 4.4: Experimental Result of ResNet-18

but the performance difference between the two is only 2% in terms of *F1score*. So, after all these experiments, we have decided to use ResNet-18 as our base model for further modification to improve performance and efficiency.

4.2.5 RevampResNet-18

ResNets learn residual functions with reference to layer inputs instead of unreferenced functions, allowing them to fit a residual mapping [He et al. \(2016\)](#). Exploding gradient and vanishing gradient are two types of optimization problems in a convolutional neural network. The skip connection in a residual block helps to overcome these problems. Skip connections allow information to move more quickly up a neural network, addressing the exploding/vanishing gradient problem. Deep neural networks have the advantage of learning complex functions quickly, but as depth increases, their performance degrades due to overfitting. Skip connections bypass layer activations to create ResNets, providing regularization to skip any layer that degrades architecture performance. To reduce the computational complexity of a CNN architecture, the bottleneck layers are useful. The bottleneck in a neural network is a layer with fewer neurons than the layer below or above it, reducing computational complexity and making training time more inexpensive.

This time we have resized the images to 512 instead of 224 as 224 pixels are less effective for highlighting the dominating features in an image [Qin et al. \(2020\)](#) and normalized the images using the mean and standard deviation of ImageNet.

Our main challenge was to decrease the FLOPs along with improving accuracy. Moreover, the challenge became more difficult as we used an image shape of 512 instead of 224 which resulted in more computational cost.

Generally, we know that increasing the layers of a neural network tends to perform better in terms of accuracy because of increased parameters (until the vanishing gradient occurs) [Khojasteh et al. \(2018\)](#). Even if it performs better in terms of accuracy, it will require more computational resources hence a trade-off between accuracy and efficiency. Computational resources should be a matter of fact for ensuring production-capable DCNN model along with performance. So it was a challenge for us how we can decrease the FLOPs keeping the layers and trainable parameters unchanged.

So, we started experimenting with the width (kernel and stride size) of the original ResNet18 architecture instead of the depth (layers) and found mathematically that FLOPs are highly dependent on the width of any architecture. We performed some mathematical calculations on hand and the findings are given below.

In the original ResNet18 architecture, the kernel size is 7, stride size is 2, and padding size is 3 in the 1st convolutional layer. So the output of this layer can be calculated using Equation 3.1.

$$\begin{aligned}
 \text{Output (1st Conv2D)} &= \left\lfloor \frac{n + 2p - f}{s} \right\rfloor + 1 \\
 &= \left\lfloor \frac{512 + 2 \times 3 - 7}{2} \right\rfloor + 1 \\
 &= \left\lfloor \frac{512 + 6 - 7}{2} \right\rfloor + 1 \\
 &= \left\lfloor \frac{511}{2} \right\rfloor + 1 \\
 &= \lfloor 255.5 \rfloor + 1 \\
 &= 255 + 1 \\
 &= 256
 \end{aligned}$$

In this equation,

Image width, $n = 512$

Kernel Size, $f = 7$

Padding Size, $p = 3$

Stride Size, $s = 2$

We increased the kernel size to 8 from 7 to increase the number of trainable parameters and stride size to 3 from 2 to decrease the feature maps and calculated the output for the 1st convolution layer again.

$$\begin{aligned}\text{Output (1st Conv2D)} &= \left\lfloor \frac{n + 2p - f}{s} \right\rfloor + 1 \\ &= \left\lfloor \frac{512 + 2 \times 3 - 8}{3} \right\rfloor + 1 \\ &= \left\lfloor \frac{512 + 6 - 8}{3} \right\rfloor + 1 \\ &= \left\lfloor \frac{510}{3} \right\rfloor + 1 \\ &= \lfloor 170 \rfloor + 1 \\ &= 171\end{aligned}$$

In this equation,

Image width, $n = 512$

Kernel Size, $f = 8$

Padding Size, $p = 3$

Stride Size, $s = 3$

We noticed that the output features of the 1st convolution layers decreased by almost 33% ($\frac{171}{256} \times 100$). Thus, we decreased the FLOPs from 1.79 billion to only 88 million (decreased by 44.4% compared to the original ResNet18 architecture) keeping the number of layers unchanged with more trainable parameters than ResNet18 (10.9 million).

However, we added 3 additional fully connected classifier layers (FC1 (256), FC2 (128), FC3(5)) on top of the feature extraction layers (512) as we used a larger image

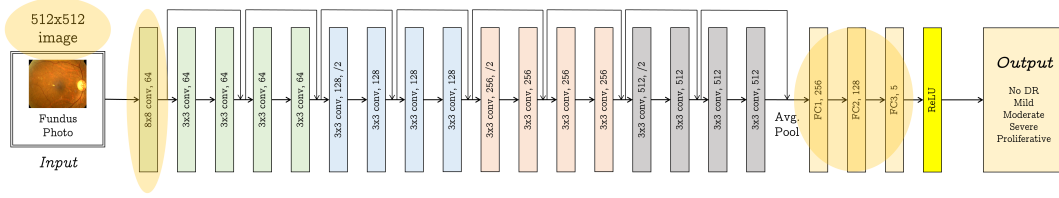


Figure 4.5: The Architecture of RevampResNet18

size (512 instead of 224) so that the model can learn more complex patterns from the images. A figure of the modified architecture is given in Figure 4.5.

Lei et al. (2018) stated that the underlying issue with deep learning is that there has not been any theoretical analysis of its internal operations, and there is no universally acknowledged theoretical explanation for the so-called “black box” phenomenon [Shwartz-Ziv & Tishby \(2017\)](#). However, we can mathematically establish how we have decreased the FLOPs of RevampResNet18 layer by layer by comparing with the original ResNet18 architecture.

We used batch size 32, weighted cross-entropy loss to solve the data imbalance issue instead of data augmentation [Phan & Yamamoto \(2020\)](#), Adam optimizer, ReLU activation function, a dropout rate of 0.5 for regularization, weight decay (L2 regularization) of 0.0001 as an additional regularizer, and learning rate 0.00001 to avoid overshooting while converging to global minima.

For the mathematical comparison of the layers between ResNet18 and RevampResNet18 model, we have used some formulas Paoletti et al. (2020) which is shown as follows:

1. No. of trainable parameters = $n_f^{(l-1)} \times n_f^{(l)} \times n_k^{(l)} \times n_k^{(l)}$
2. Convolutional layers FLOPs = $n_f^{(l-1)} \times n_f^{(l)} \times n_k^{(l)} \times n_k^{(l)} \times m^{(l)} \times m^{(l)}$
3. Pooling layer FLOPs = $m_h \times m_w \times m_c$
4. Fully connected layer FLOPs = FC parameters = $I_n \times O_n$

The notational conventions are as follows:

- n = Input image shape
- f = Filter size
- p = Padding size
- s = Stride size
- $n_f^{(l-1)}$ = No. of filters in the previous layer
- $n_f^{(l)}$ = No. of filters in the current layer
- $n_k^{(l)} \times n_k^{(l)}$ = Filter/kernel shape
- m_l = Output (conv2d)
- m_h = Input height
- m_w = Input width
- m_c = Input channel
- I_n = Input neuron
- O_n = Output neuron

In the original Resnet18 architecture, the 1st convolution layer includes 3 input channel, 64 output channel, 7 filter size, 2 stride size, and 3 padding size.

Using the equations above, Output (conv2d) = $\frac{224-7+2 \times 3}{2} + 1 = 112$.

No. of trainable parameters = $3 \times 64 \times 7 \times 7 \times 7 = 9408$.

Convolutional layer FLOPs = $3 \times 64 \times 7 \times 7 \times 7 \times 112 \times 112 = 118013952$.

In RevampResNet18, we have changed the kernel size to 8 and stride size to 3 in the 1st convolution layer. After analyzing the fundus photographs from our dataset, we detected a few characteristics of the images. As we need to find the features of the abnormal blood vessels, aneurysms, hemorrhages, and Cotton Wool spots to detect the DR stage from a fundus image, we found that these abnormalities get surrounded through a region. For this we do not need to analyze every pixel, rather we need to analyze through a region.

Using the equations above, Output (conv2d) = $\frac{224-8+2 \times 3}{3} + 1 = 75$ (feature maps decreased by 33%).

No. of trainable parameters = $3 \times 64 \times 7 \times 7 \times 7 = 12288$ (parameters increased by 23%).

Convolutional layer FLOPs = $3 \times 64 \times 7 \times 7 \times 7 \times 112 \times 112 = 69120000$ (FLOPs decreased by 41%).

We trained the model for 30 epochs and achieved a 99% train *F1score* and 89% test *F1score*. We are claiming that this is the best model so far for Diabetic Retinopathy Detection on the EyePacs dataset for multiclass classification using one single architecture with the least number of FLOPs.

We have calculated layerwise trainable parameters and FLOPs for both the original ResNet18 and RevampResNet18 following the same calculations given above which is demonstrated in Table 2. In the conv layer, the trainable parameters get decreased so feature extraction also gets decreased. But, as stated earlier, from the nature of the fundus photographs for DR, analyzing complex patterns with fully connected layers is more important than feature extraction. And, as the output feature maps of a layer are used as the input for the following layer of our modified version of the model, the FLOPs get significantly decreased for the model.

A brief summary of our best model RevampResNet-18 and the ResNet18 implementation for our dataset is given below:

Table 4.5: Result summary of RevampResNet-18 and ResNet18

Models	Train <i>F1score</i>	Test <i>F1score</i>	Epochs
ResNet-18	94%	83%	30
RevampResNet-18	98%	89%	30

Table 4.6: Computation cost summary of RevampResNet-18 and ResNet18

Models	Trainable Parameters (in million)	FLOPs (in Billion)
ResNet-18	10.9	1.79
RevampResNet-18	11.2	0.88

However, this performance can be increased by doing further research for ensuring better accuracy with fewer computation resources for multiclass classification on the EyePacs dataset.

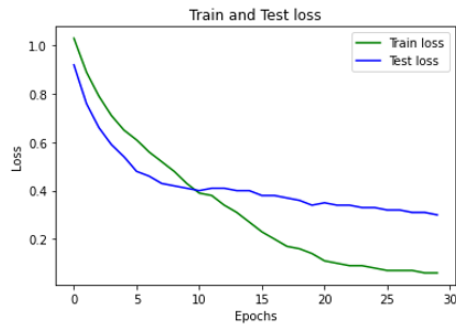
ResNet-18						Revamp-ResNet-18					
k = 7	Conv2D	Input	224, 224, 3	Parameters	9408	k = 8	Conv2D	Input	224, 224, 3	Parameters	12288
s = 2, p = 3		Output	112, 112, 64	FLOPs	118,013,952	s = 3, p = 3		Output	75, 75, 64	FLOPs	69,120,000
k = 3	Maxpool	Input	112, 112, 64	Parameters	-	k = 3	Maxpool	Input	75, 75, 64	Parameters	-
s = 2, p = 1		Output	56, 56, 64	FLOPs	802816	s = 2, p = 1		Output	38, 38, 64	FLOPs	360000
k = 3	Conv2D	Input	56, 56, 64	Parameters	36864	k = 3	Conv2D	Input	38, 38, 64	Parameters	36864
s = 1, p = 1		Output	56, 56, 64	FLOPs	115,605,504	s = 1, p = 1		Output	38, 38, 64	FLOPs	53,231,616
k = 3	Conv2D	Input	56, 56, 64	Parameters	36864	k = 3	Conv2D	Input	38, 38, 64	Parameters	36864
s = 1, p = 1		Output	56, 56, 64	FLOPs	115,605,504	s = 1, p = 1		Output	38, 38, 64	FLOPs	53,231,616
k = 3	Conv2D	Input	56, 56, 64	Parameters	36864	k = 3	Conv2D	Input	38, 38, 64	Parameters	36864
s = 1, p = 1		Output	56, 56, 64	FLOPs	115,605,504	s = 1, p = 1		Output	38, 38, 64	FLOPs	53,231,616
k = 3	Conv2D	Input	56, 56, 64	Parameters	36864	k = 3	Conv2D	Input	38, 38, 64	Parameters	36864
s = 1, p = 1		Output	56, 56, 64	FLOPs	115,605,504	s = 1, p = 1		Output	38, 38, 64	FLOPs	53,231,616
k = 3	Conv2D	Input	56, 56, 64	Parameters	73728	k = 3	Conv2D	Input	38, 38, 64	Parameters	73728
s = 2, p = 1		Output	28, 28, 128	FLOPs	57,802,752	s = 2, p = 1		Output	19, 19, 128	FLOPs	26,615,808
k = 3	Conv2D	Input	28, 28, 128	Parameters	147456	k = 3	Conv2D	Input	19, 19, 128	Parameters	147456
s = 1, p = 1		Output	28, 28, 128	FLOPs	115,605,504	s = 1, p = 1		Output	19, 19, 128	FLOPs	53,231,616
k = 3	Conv2D	Input	28, 28, 128	Parameters	147456	k = 3	Conv2D	Input	19, 19, 128	Parameters	147456
s = 1, p = 1		Output	28, 28, 128	FLOPs	115,605,504	s = 1, p = 1		Output	19, 19, 128	FLOPs	53,231,616
k = 3	Conv2D	Input	28, 28, 128	Parameters	147456	k = 3	Conv2D	Input	19, 19, 128	Parameters	147456
s = 1, p = 1		Output	28, 28, 128	FLOPs	115,605,504	s = 1, p = 1		Output	19, 19, 128	FLOPs	53,231,616
k = 3	Conv2D	Input	28, 28, 128	Parameters	294912	k = 3	Conv2D	Input	19, 19, 128	Parameters	294912
s = 2, p = 1		Output	14, 14, 256	FLOPs	57,802,752	s = 2, p = 1		Output	10, 10, 256	FLOPs	29,491,200
k = 3	Conv2D	Input	14, 14, 256	Parameters	589824	k = 3	Conv2D	Input	10, 10, 256	Parameters	589824
s = 1, p = 1		Output	14, 14, 256	FLOPs	115,605,504	s = 1, p = 1		Output	10, 10, 256	FLOPs	58,982,400
k = 3	Conv2D	Input	14, 14, 256	Parameters	589824	k = 3	Conv2D	Input	10, 10, 256	Parameters	589824
s = 1, p = 1		Output	14, 14, 256	FLOPs	115,605,504	s = 1, p = 1		Output	10, 10, 256	FLOPs	58,982,400
k = 3	Conv2D	Input	14, 14, 256	Parameters	589824	k = 3	Conv2D	Input	10, 10, 256	Parameters	589824
s = 1, p = 1		Output	14, 14, 256	FLOPs	115,605,504	s = 1, p = 1		Output	10, 10, 256	FLOPs	58,982,400
k = 3	Conv2D	Input	14, 14, 256	Parameters	1179648	k = 3	Conv2D	Input	10, 10, 256	Parameters	1179648
s = 2, p = 1		Output	7, 7, 512	FLOPs	57,802,752	s = 2, p = 1		Output	5, 5, 512	FLOPs	29,491,200
k = 3	Conv2D	Input	7, 7, 512	Parameters	2359296	k = 3	Conv2D	Input	5, 5, 512	Parameters	2359296
s = 1, p = 1		Output	7, 7, 512	FLOPs	115,605,504	s = 1, p = 1		Output	5, 5, 512	FLOPs	58,982,400
k = 3	Conv2D	Input	7, 7, 512	Parameters	2359296	k = 3	Conv2D	Input	5, 5, 512	Parameters	2359296
s = 1, p = 1		Output	7, 7, 512	FLOPs	115,605,504	s = 1, p = 1		Output	5, 5, 512	FLOPs	58,982,400
k = 3	Conv2D	Input	7, 7, 512	Parameters	2359296	k = 3	Conv2D	Input	5, 5, 512	Parameters	2359296
s = 1, p = 1		Output	7, 7, 512	FLOPs	115,605,504	s = 1, p = 1		Output	5, 5, 512	FLOPs	58,982,400
	Avg Pool	Input	7, 7, 512	Parameters	-		Avg Pool	Input	5, 5, 512	Parameters	-
		Output	512	FLOPs	25088			Output	512	FLOPs	12800
	FC	Input	512	Parameters	2560		FC1	Input	512	Parameters	131072
		Output	5	FLOPs	2560			Output	256	FLOPs	131072
							FC2	Input	256	Parameters	32768
								Output	128	FLOPs	32768
							FC3	Input	128	Parameters	640
								Output	5	FLOPs	640

* k = kernel size, s = stride size, p = padding size, conv = convolutional layer, FC = fully-connected layer

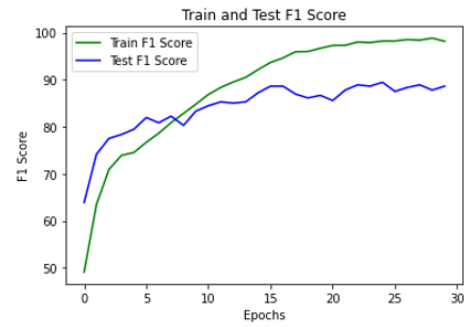
Figure 4.6: Layerwise comparison between ResNet18 and RevampResNet18

4.3 Proposed Web Platform

We have developed a web platform where users can upload their own fundus photograph to find the stage of DR in that fundus image. Previously, it took weeks to clinically diagnose the stage of DR. But, with our platform, users can find the stage of DR within seconds. People can access the website from anywhere in the world using an active internet connection and a web browser.



(a) Plot of Train and Test Loss vs Epoch in RevampResNet-18



(b) Plot of Train and Test $F1score$ vs Epoch in RevampResNet-18

Figure 4.7: Experimental Result of RevampResNet-18

4.3.1 Model Deployment

To develop a deep learning based web platform where users can upload his/her own image to get the predicated output by the model, we needed to deploy our model. A deep learning model is deployed after it has been completed and is ready to be utilized in a real-world setting for the intended use.

4.3.1.1 Saving and Loading a Model

To get a saved model which is ready for inference, we need a learned weight matrix and a model class. We get a learned weight matrix by training our dataset to a model. We bind the learned weight matrix and the model class. To make inference, we upload the image to the saved model. It passes through the class via the learned weights to predict the final output. A block diagram of the model deployment process is shown in the Figure 4.8.

4.3.2 Implementation

To develop the web platform where one can upload his/her own fundus photograph to find the stage of DR, we have used Python's Streamlit framework. Streamlit framework gives a wide variety of functionality to develop a website. This open-source framework is being widely used by various machine learning and data science teams. Our front-end was designed using this framework. We have hosted our platform as a public space at Huggingface. People can use their PCs as well as smartphones to navigate to the Huggingface web platform.

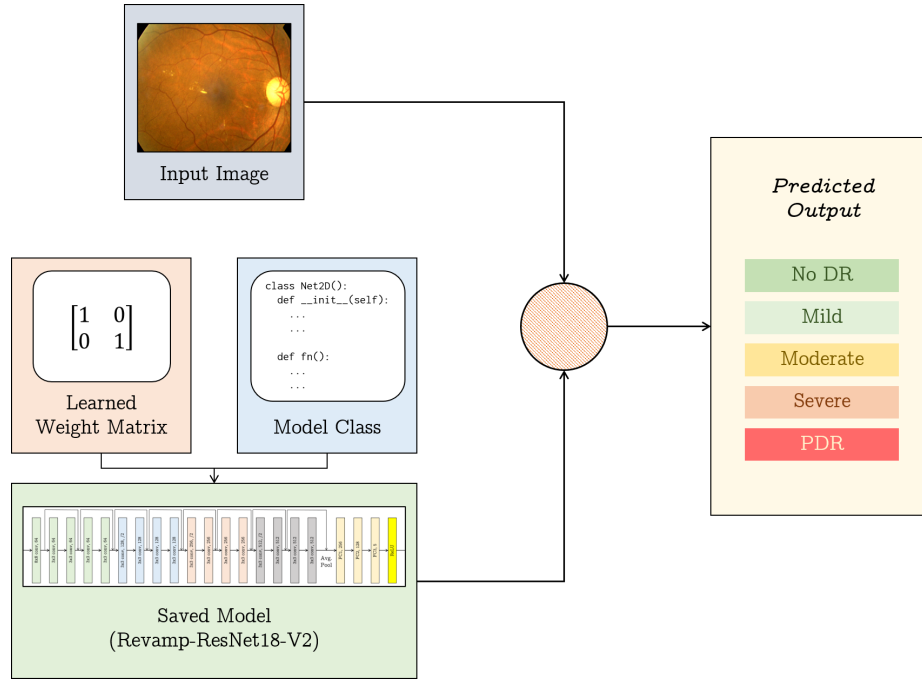


Figure 4.8: Block diagram of model deployment

4.3.2.1 Control Flow of the Platform

The landing page of the web platform has a file-picker to upload a fundus photograph to the database of the site. *The Platform*, *Prevention Guide*, and *Credits* are three of its tabs. The file uploading part of *The Platform* tab allows users to upload their own fundus photographs. Following image upload, the model will begin image analysis and generate the output in accordance with the features of the images. The expected stage as well as the percentages of each step are displayed when the output has been prepared. A short note on the detected stage will also be shown in the drop-down expander. On the other hand, the *Prevention Guide* tab contains instructions on how to reduce one's chance of developing diabetic retinopathy. On that tab, there is also a blog that may encourage individuals to take their health and ongoing lifestyle seriously. Additionally, the *Credits* tab displays the names of the platform creators together with the name of their supervisor. The overall control flow has been shown in the activity diagram of the Figure 4.9.

Figure 4.10 shows the homepage of our platform. The platform's emblem and the name of the proposed system that we develop are both shown on the homepage. There are three tabs below that, the first of which is *The Platform*. The user can

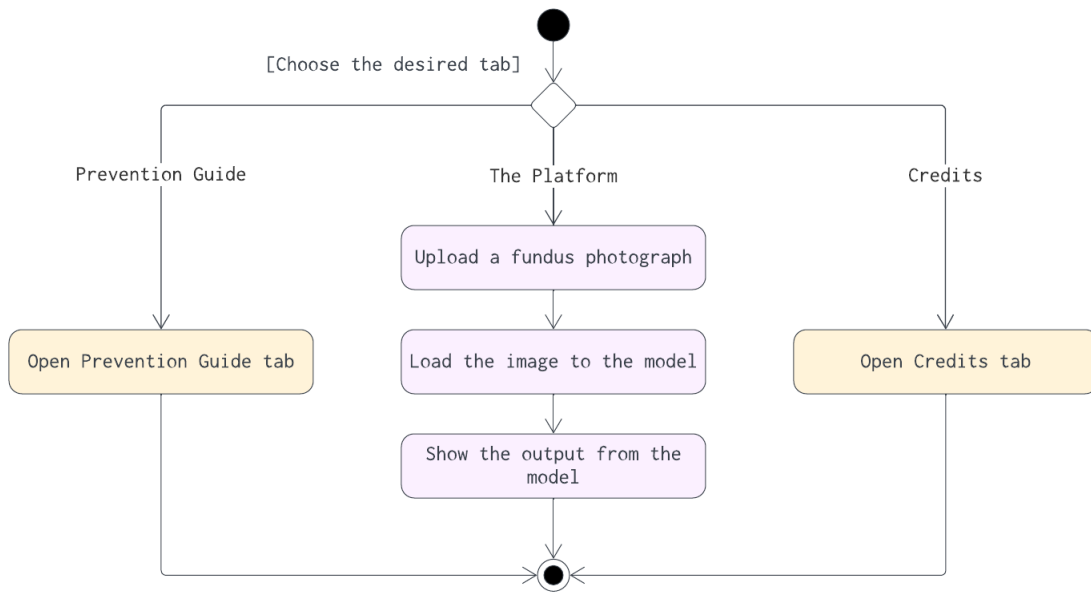


Figure 4.9: The activity diagram of the proposed web platform

quickly upload the fundus photograph using the drag and drop feature in that. The photo will appear on the page after the uploading process completes.

Figure 4.11 shows the model-analyzing process. The image loads into the model after the user uploads the fundus photo to the system. The model begins its prediction process after it has been loaded in order to identify the precise features.

Figure 4.12 shows detailed prediction result. When the model has finished analyzing the image, a detailed result is displayed. The predicted stage of diabetic retinopathy is displayed first, followed by all the specific probabilities for each of the five stages of DR. The drop-down expander will also display a brief description of the identified stage.

DRDetect_

A Diabetic Retinopathy (DR) Stage Detecting Platform

[The Platform](#) [Prevention Guide](#) [Credits](#)

Upload a fundus photograph to find the stage of DR



Drag and drop file here
Limit 200MB per file

Browse files

Figure 4.10: Homepage of our developed proposed platform

DRDetect_

A Diabetic Retinopathy (DR) Stage Detecting Platform

[The Platform](#) [Prevention Guide](#) [Credits](#)

Upload a fundus photograph to find the stage of DR

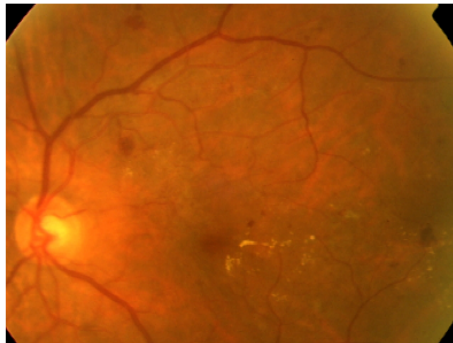


Drag and drop file here
Limit 200MB per file

Browse files



class - moderate - 2.png 202.0KB



Model is analyzing the upload...

Figure 4.11: Model is analyzing the upload in the backend

DRDetect

A Diabetic Retinopathy (DR) Stage Detecting Platform

[The Platform](#) [Prevention Guide](#) [Credits](#)

Upload a fundus photograph to find the stage of DR



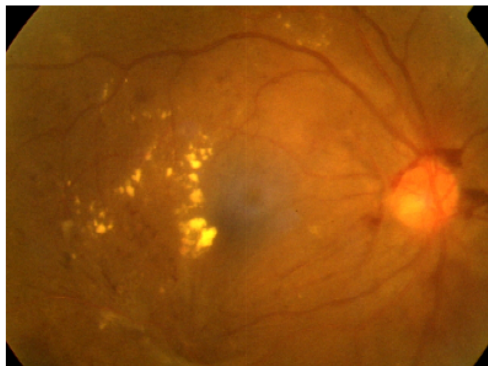
Drag and drop file here

Limit 200MB per file

Browse files



class - proliferative - 4.png 224.7KB



Predicted Stage of DR: Proliferative

Detailed Probabilities

No DR: 0.0%

Mild: 0.1%

Moderate: 2.2%

Severe: 1.9%

Proliferative: 95.8%

Prediction completed!

Health Advisory for Proliferative Stage of DR

At this point, the disease has advanced significantly and is very threatening to one's vision. Because of additional damage to the eye's blood vessels, there is worsening circulation inside the eye. In response, the retina then grows new blood vessels; however, they are abnormal and can cause severe damage, possibly resulting in vision loss and potentially blindness.

Figure 4.12: The platform is showing the output of the analysis

Chapter 5

Impact on Society, Environment, and Sustainability

5.1 Impact on Society

People who have diabetic retinopathy, which is harmful to their vision, are likely to feel more social pressure. Significantly, people who have both psychosocial issues and vision-threatening diabetic retinopathy may perform at considerably lower than others [Kaminsky et al. \(2014\)](#). This may lead to poor adherence, heightened family functioning demands, worsened diabetes control and faster progression of diabetic retinopathy. These effects have a number of negative effects on the management of the disease, health outcomes, and healthcare costs [Fenwick et al. \(2011\)](#). If someone has symptoms of diabetic retinopathy and their retinal scans are of high quality, they may use our proposed system to accurately determine what stage of the disease they are in. As we develop a web platform for each user that may produce a result based on the data provided to the system. Since, ophthalmologists typically spend a lot of time diagnosis the retinal picture and examining it in the research laboratories before they can generate a report for a specific patient, which makes it difficult for them to deliver their report on time. Therefore, Our entire system is faster and more accurately designed to benefit society as a whole.

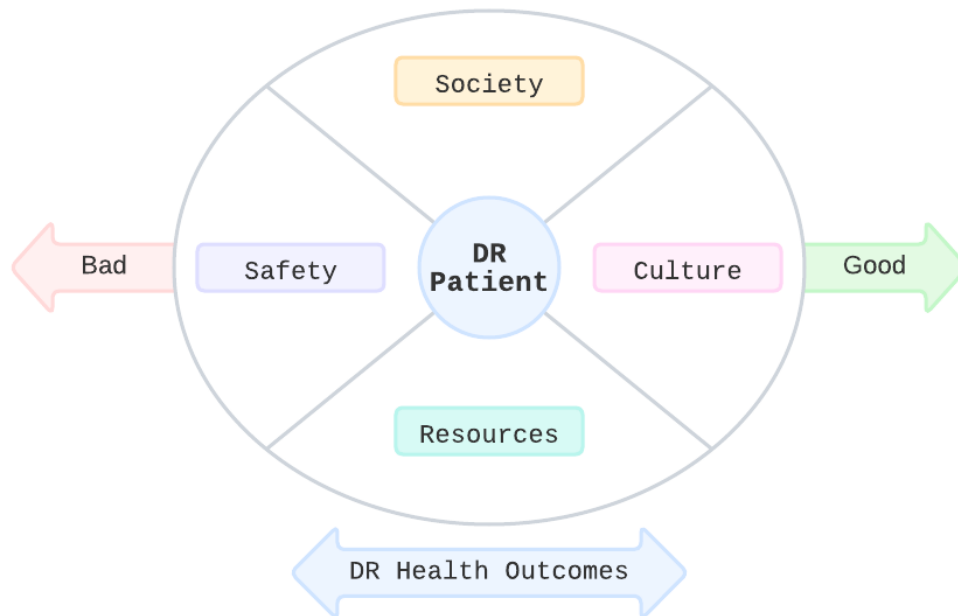


Figure 5.1: The impact of Diabetic Retinopathy (DR) patients' consequences on the environment

5.2 Impact on Environment

The pathophysiology of diabetic retinopathy which is a multi-factorial syndrome, is not well understood. Glycosylated hemoglobin may be responsible for 11% of the risk of developing micro-vascular problems associated with diabetes [Hirsch & Brownlee \(2010\)](#). Despite having a confirmed, efficient cure, DR continues to be the largest cause of blindness in the worldwide. Patients with low socioeconomic level are more prone to develop diabetic eye disease [Flynn & Smiddy \(2000\)](#). Individual risk factors have been identified, but it has also been discovered that patient community features have an impact on overall health outcomes and diabetes outcomes in particular [Brown et al. \(2004\)](#). Blood glucose levels and glycosylated hemoglobin (HbA1c) were found to be higher in patients from underprivileged neighborhoods [11]. Although patients are at the focus of healthcare treatments, the places they live in have an influence on their health outcomes. Communities and neighborhoods must be taken into account while making interventions to enhance diabetes management, as this is the focus of our system. According to Figure 5.1, both a

patient's own systemic traits and those present in their surroundings may cause them to shift along the health-care continuum outcomes. There are a number of risk factors for diabetes and its consequences, as well as environmental risk factors for diabetic retinopathy, which have already covered in subsection ???. The figure makes it simple to understand how the environment has an influence. Additionally, by providing a very robust network for the inhabitants of the environment, our system may play a very significant function. and be able to do much more for our surrounding.

5.3 Ethical Aspects

To prevent vision loss and blindness, diabetic retinopathy (DR) patients must be sent to an ophthalmologist as soon as possible [Vujosevic et al. \(2020\)](#). The development of deep learning aided diagnostics has sped up access to ophthalmic specialists and cut healthcare expenses [Vujosevic et al. \(2020\)](#). Although the concept of automated DR screening dates back to the late 1990s, significant advancements in processing power in the early 2010s allowed for the handling of the large volume of retinal pictures [Ting et al. \(2019\)](#). Since 2018, commercially feasible DR deep learning solutions for primary care settings are available [Fda \(2018\)](#).

Our system uses deep learning concept to evaluate eye image data and primarily relies on an architecture that increases accuracy. It is a suggested system to offer an autonomous online platform that delivers enhanced accuracy and a less error-free system, and the output of the system is an advice to refer patients to an eye care expert when it recognizes the stages of the DR. In terms of ethics, a novel and distinctive feature of this deep learning-aided system necessitates a critical evaluation that the DR purposefully produces an autonomous recommendation that functions as a diagnosis without requiring the supervision of a physician, even though ophthalmologist are generally in charge of making diagnosis and treatment.

There are two implications for general practitioners: First, in order to appropriately advise patients, ophthalmologists need to be more knowledgeable about the ethical implications of cutting-edge technology. Second, it is possible to reduce patients' worries or overconfidence by explaining the dangers, restrictions, and potential advantages of diagnostic Deep Learning systems. Patients have a better

chance of receiving a timely diagnosis and treatment if they accept and are aware of the limits of deep learning assisted diagnosis.

5.4 Sustainability Plan

Rates of diabetic retinopathy are increasing globally. These ailments are among the main causes of blindness globally, along with a number of chronic diseases linked to poor nutrition. Societies and healthcare systems are heavily impacted by this epidemic [Galaz et al. \(2012\)](#), particularly in nations with some of the lowest incomes in the world. The demand for health sector will continue to grow faster than people's means of financing it. 80% of diabetes patients live in low- and middle-income countries, where 20% of global diabetes health sector expenditures were made [Abate & Chandalia \(2001\)](#). Therefore, persons with diabetes who make up 80% of the population have a possibility to develop diabetic retinopathy at some time during their lives. The World Bank reports that during 2007 and 2009, Bangladesh's per-capita health sector spending remained stable at about 3% of GDP, with no change in trend [Group \(2013\)](#). To put it another way, our suggested system mainly focuses on a computer-based autonomous system for detecting whether a patient has diabetic retinopathy and, if so, what stage it is in. Since it takes a month to receive the results from the ophthalmologists, knowing that makes it simple to make the diagnosis in a timely manner. Numerous reasons, such as inadequate understanding of diabetic eye illness, economic issues, a low referral rate from doctors, and limited mobility among the elderly, contributed to the lack of desire for eye care. Our approach will enable them to navigate with ease and gain knowledge on diabetic retinopathy. Our system will be crucial in maintaining the project's long-term sustainability. To guarantee suitable systems, meetings with health management teams to give technical assistance have to be held.

Chapter 6

Summary, Conclusion, and Future Scope

6.1 Summary of the Study

The effects of retinal abnormalities are often progressive in nature, demonstrating the need for a reliable approach for abnormality diagnosis. Retinal abnormality is one of them. Diabetes-related issues can lead to a condition known as diabetic retinopathy (DR), which damages the retina of the human eye. One of the primary causes of visual loss is DR, which is becoming more common. A common diabetic retinal ailment called diabetic retinopathy is brought on by blood vessels in the retina changing from their original configuration. Both of a person's eyes are often affected by diabetic retinopathy. The majority of ophthalmologists rely on visual interpretation to determine the different sorts of disorders. However, an inaccurate diagnosis can alter the planned course of therapy, which can be devastating. In this study, we sought to develop an automated system that produces highly accurate findings while being bias-free, more accurate, less error-prone, and easy to maintain. We used the EyePACS dataset, which includes high-resolution annotated pictures in five different groups, to categorize the various stages of DR. The photos are created in a variety of dimensions and imaging parameters. To detect the stage of diabetic retinopathy, we have experimented with a wide variety of deep learning models. We have experimented with both Convolutional Neural Network (CNN) and attention mechanism-based deep learning architectures such as ResNet (152, 101, 34, 18), EfficientNet (B0, and B1), and VisionTransformer. The best performing models in our experiment were EfficientNetB1, VisionTranformer16, ResNet18,

etc. Additionally, we have tested our own model RevampResNet18 with an 89% *F1score*. It is possible to obtain the minimum number of FLOPs with more trainable parameters than the base ResNet-18 model by using our model, RevampResNet18. The total FLOPs that are minimized across the entire model are 0.88 billion, down from 1.79 billion in ResNet-18. We have developed a web platform where users can upload their own fundus photograph to find the stage of DR in that fundus image. Previously, it took weeks to clinically diagnose the stage of DR. But, with our platform, users can find the stage of DR within seconds. People can access the website from anywhere in the world using an active internet connection in the PC or smartphone and a web browser.

6.2 Conclusion

In this study, we have developed a web platform where one can detect the stage of DR by uploading a fundus photograph. This will save the time of ophthalmologists in clinical diagnosis of diabetic retinopathy detection. We have used our modified model called RevampResNet18 which has achieved 89% *F1score* in the test set. Using our model, we get minimum number of FLOPs with more trainable parameters than base ResNet-18 model.

6.3 Implication for Future Study

We aspire to enhance our model in the future for the convenience of our suggested system. Additionally, we desire a responsive web platform where users may undertake fundus photography and learn about the key stages of diabetic retinopathy. Since we currently have a dataset at our access, we wish to further enhance it by gathering user data on the web platform. However, it will be a major issue because we are dealing with the user data, therefore we will preserve these data with additional security and also with the consent of the user concern. We want to train those data as set to our model to make our model function more efficiently.

References

- Abate, N., & Chandalia, M. (2001). Ethnicity and type 2 diabetes: focus on asian indians. *Journal of Diabetes and its Complications*, 15(6), 320–327.
- Abdulla, W., & Chalakal, R. J. (2018). University of auckland diabetic retinopathy (uoa-dr) database-end user licence agreement.
- Abràmoff, M. D., Lou, Y., Erginay, A., Clarida, W., Amelon, R., Folk, J. C., & Niemeijer, M. (2016). Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. *Investigative ophthalmology & visual science*, 57(13), 5200–5206.
- Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.
- Alyoubi, W. L., Shalash, W. M., & Abulkhair, M. F. (2020). Diabetic retinopathy detection through deep learning techniques: A review. *Informatics in Medicine Unlocked*, 20, 100377.
- Amos, A. F., McCarty, D. J., & Zimmet, P. (1997). The rising global burden of diabetes and its complications: estimates and projections to the year 2010. *Diabetic medicine*, 14(S5), S7–S85.
- Atwany, M. Z., Sahyoun, A. H., & Yaqub, M. (2022). Deep learning techniques for diabetic retinopathy classification: A survey. *IEEE Access*.
- Bae, K., Ryu, H., & Shin, H. (2019). Does adam optimizer keep close to the optimal point? *arXiv preprint arXiv:1911.00289*.
- Becheva, M. S. V., & Kirkova-Bogdanova, A. G. (2022). Prophylactics of type 2 diabetes and diabetic foot. *Iranian Journal of Public Health*, 51(10), 2370.
- Brown, A. F., Ettner, S. L., Piette, J., Weinberger, M., Gregg, E., Shapiro, M. F., ... others (2004). Socioeconomic position and health among persons with diabetes mellitus: a conceptual framework and review of the literature. *Epidemiologic reviews*, 26(1), 63–77.

- Brownlee, J. (2020). *Machine learning mastery: Understand the impact of learning rate on neural network performance*.
- Diabetic retinopathy; american optometric association: New york, ny, usa, 2021. (n.d.).*
- Diabetic retinopathy; mayo foundation for medical education and research: Rochester, mn, usa, 2022. (n.d.).*
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... others (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Fda, U. (2018). Fda permits marketing of artificial intelligence-based device to detect certain diabetes-related eye problems. *Silver Spring, DM, Department of Health and Human Services*.
- Federation, I. D. (2017). Idf diabetes atlas 8th edition. *International diabetes federation, 905–911*.
- Fenwick, E., Pesudovs, K., Rees, G., Dirani, M., Kawasaki, R., Wong, T., & Lamoureaux, E. (2011). The impact of diabetic retinopathy: understanding the patient's perspective. *British Journal of Ophthalmology, 95*(6), 774–782.
- Flynn, H. W., & Smiddy, W. E. (2000). *Diabetes and ocular disease: past, present, and future therapies* (No. 14). Oxford University Press.
- Fong, D. S., Aiello, L., Gardner, T. W., King, G. L., Blankenship, G., Cavallerano, J. D., ... Association, A. D. (2004). Retinopathy in diabetes. *Diabetes care, 27*(suppl.1), s84–s87.
- Fu, H., Xu, Y., Wong, D. W. K., & Liu, J. (2016). Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. In *2016 ieee 13th international symposium on biomedical imaging (isbi)* (pp. 698–701).
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050–1059).
- Galaz, V., Biermann, F., Folke, C., Nilsson, M., & Olsson, P. (2012). *Global environmental governance and planetary boundaries: an introduction* (Vol. 81). Elsevier.
- Gangwar, A. K., & Ravi, V. (2021). Diabetic retinopathy detection using transfer learning and deep learning. In *Evolution in computational intelligence* (pp. 679–689). Springer.

- Gondal, W. M., Köhler, J. M., Grzeszick, R., Fink, G. A., & Hirsch, M. (2017). Weakly-supervised localization of diabetic retinopathy lesions in retinal fundus images. In *2017 IEEE International Conference on Image Processing (ICIP)* (pp. 2069–2073).
- Group, W. B. (2013). *Global financial development report 2014: Financial inclusion* (Vol. 2). World Bank Publications.
- Hancho, N., Kirigia, J., & Claude, K. (2017). IdF diabetes atlas. vol. 8. *International Diabetes Federation*, 1–150.
- Harangi, B., Toth, J., Baran, A., & Hajdu, A. (2019). Automatic screening of fundus images using a combination of convolutional neural network and hand-crafted features. In *2019 41st annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 2699–2702).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Hemanth, D. J., Deperlioglu, O., & Kose, U. (2020). An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. *Neural Computing and Applications*, 32(3), 707–721.
- Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.
- Hirsch, I. B., & Brownlee, M. (2010). Beyond hemoglobin a1c—need for additional markers of risk for diabetic microvascular complications. *Jama*, 303(22), 2291–2292.
- Hou, Y., Cai, Y., Jia, Z., & Shi, S. (2020). Risk factors and prevalence of diabetic retinopathy: A protocol for meta-analysis. *Medicine*, 99(42).
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Hsiao, T.-Y., Chang, Y.-C., Chou, H.-H., & Chiu, C.-T. (2019). Filter-based deep-compression with global average pooling for convolutional networks. *Journal of Systems Architecture*, 95, 9–18.
- Ishtiaq, U., Abdul Kareem, S., Abdullah, E. R. M. F., Mujtaba, G., Jahangir, R., & Ghafoor, H. Y. (2020). Diabetic retinopathy detection through artificial intelligent techniques: a review and open issues. *Multimedia Tools and Applications*, 79(21), 15209–15252.

- Kaminsky, T. A., Mitchell, P. H., Thompson, E. A., Dudgeon, B. J., & Powell, J. M. (2014). Supports and barriers as experienced by individuals with vision loss from diabetes. *Disability and rehabilitation*, 36(6), 487–496.
- Kauppi, T., Kalesnykiene, V., Kamarainen, J.-K., Lensu, L., Sorri, I., Uusitalo, H., ... Pietilä, J. (2006). Diaretdb0: Evaluation database and methodology for diabetic retinopathy algorithms. *Machine Vision and Pattern Recognition Research Group, Lappeenranta University of Technology, Finland*, 73, 1–17.
- Khalifa, N. E. M., Loey, M., Taha, M. H. N., & Mohamed, H. N. E. T. (2019). Deep transfer learning models for medical diabetic retinopathy detection. *Acta Informatica Medica*, 27(5), 327.
- Khojasteh, P., Aliahmad, B., Arjunan, S. P., & Kumar, D. K. (2018). Introducing a novel layer in convolutional neural network for automatic identification of diabetic retinopathy. In *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)* (pp. 5938–5941).
- Kondratyuk, D., Tan, M., Brown, M., & Gong, B. (2020). When ensembling smaller models is more efficient than single large models. *arXiv preprint arXiv:2005.00570*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- Lazar, I., & Hajdu, A. (2012). Retinal microaneurysm detection through local rotating cross-section profile analysis. *IEEE transactions on medical imaging*, 32(2), 400–407.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Lewkowycz, A. (2021). How to decay your learning rate. *arXiv preprint arXiv:2103.12682*.
- Li, X., Hu, X., Yu, L., Zhu, L., Fu, C.-W., & Heng, P.-A. (2019). Canet: cross-disease attention network for joint diabetic retinopathy and diabetic macular edema grading. *IEEE transactions on medical imaging*, 39(5), 1483–1493.
- Muqit, M. M., Kourgialis, N., Jackson-deGraffenried, M., Talukder, Z., Khetran, E. R., Rahman, A., ... others (2019). Trends in diabetic retinopathy, visual acuity, and treatment outcomes for patients living with diabetes in a fundus photograph-based diabetic retinopathy screening program in bangladesh. *JAMA network open*, 2(11), e1916285–e1916285.
- Naveenkumar, M., Srithar, S., Maheswaran, T., Sivapriya, K., & Brinda, B. (2022). Diabetic retinopathy disease classification using efficientnet-b3. In *Innovative data communication technologies and application* (pp. 809–821). Springer.

- Nguyen, Q. H., Muthuraman, R., Singh, L., Sen, G., Tran, A. C., Nguyen, B. P., & Chua, M. (2020). Diabetic retinopathy detection using deep learning. In *Proceedings of the 4th international conference on machine learning and soft computing* (pp. 103–107).
- Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). Determination press San Francisco, CA, USA.
- Phan, T. H., & Yamamoto, K. (2020). Resolving class imbalance in object detection with weighted cross entropy losses. *arXiv preprint arXiv:2006.01413*.
- Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabuddhe, V., & Meriaudeau, F. (2018). Indian diabetic retinopathy image dataset (idrid): a database for diabetic retinopathy screening research. *Data*, 3(3), 25.
- Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). Convolutional neural networks for diabetic retinopathy. *Procedia computer science*, 90, 200–205.
- Prechelt, L. (2012). Early stopping—but when? *Neural networks: tricks of the trade: second edition*, 53–67.
- Qiao, L., Zhu, Y., & Zhou, H. (2020). Diabetic retinopathy detection using prognosis of microaneurysm and early diagnosis system for non-proliferative diabetic retinopathy based on deep learning algorithms. *IEEE Access*, 8, 104292–104302.
- Qin, J., Pan, W., Xiang, X., Tan, Y., & Hou, G. (2020). A biological image classification method based on improved cnn. *Ecological Informatics*, 58, 101093.
- Qummar, S., Khan, F. G., Shah, S., Khan, A., Shamshirband, S., Rehman, Z. U., ... Jadoon, W. (2019). A deep learning ensemble approach for diabetic retinopathy detection. *Ieee Access*, 7, 150530–150539.
- Rajalakshmi, R., Subashini, R., Anjana, R. M., & Mohan, V. (2018). Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence. *Eye*, 32(6), 1138–1144.
- Ramana, B. V., & Boddu, R. S. K. (2019). Performance comparison of classification algorithms on medical datasets. In *2019 IEEE 9th annual computing and communication workshop and conference (ccwc)* (pp. 0140–0145).
- Roglic, G., et al. (2016). Who global report on diabetes: A summary. *International Journal of Noncommunicable Diseases*, 1(1), 3.
- Safi, H., Safi, S., Hafezi-Moghadam, A., & Ahmadi, H. (2018). Early detection of diabetic retinopathy. *Survey of ophthalmology*, 63(5), 601–608.

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4510–4520).
- Saxena, G., Verma, D. K., Paraye, A., Rajan, A., & Rawat, A. (2020). Improved and robust deep learning agent for preliminary detection of diabetic retinopathy using public datasets. *Intelligence-Based Medicine*, 3, 100022.
- Scanlon, P. H. (2017). The english national screening programme for diabetic retinopathy 2003–2016. *Acta diabetologica*, 54(6), 515–525.
- Semenov, A., Boginski, V., & Pasiliao, E. L. (2019). Neural networks with multidimensional cross-entropy loss functions. In *Computational data and social networks: 8th international conference, csonet 2019, ho chi minh city, vietnam, november 18–20, 2019, proceedings 8* (pp. 57–62).
- Shukla, U. V., & Tripathy, K. (2022). Diabetic retinopathy. In *Statpearls [internet]*. StatPearls Publishing.
- Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.
- Solkar, S. D., & Das, L. (2017). Survey on retinal blood vessels segmentation techniques for detection of diabetic retinopathy. *Diabetes*.
- Stitt, A. W., Curtis, T. M., Chen, M., Medina, R. J., McKay, G. J., Jenkins, A., ... others (2016). The progress in understanding and treatment of diabetic retinopathy. *Progress in retinal and eye research*, 51, 156–186.
- Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139–1147).
- Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114).
- Ting, D. S., Peng, L., Varadarajan, A. V., Keane, P. A., Burlina, P. M., Chiang, M. F., ... others (2019). Deep learning in ophthalmology: the technical and clinical considerations. *Progress in retinal and eye research*, 72, 100759.
- Tymchenko, B., Marchenko, P., & Spodarets, D. (2020). Deep learning approach to diabetic retinopathy detection. *arXiv preprint arXiv:2003.02261*.
- Vujosevic, S., Aldington, S. J., Silva, P., Hernández, C., Scanlon, P., Peto, T., & Simó, R. (2020). Screening for diabetic retinopathy: new perspectives and challenges. *The Lancet Diabetes & Endocrinology*, 8(4), 337–347.

- Wan, S., Liang, Y., & Zhang, Y. (2018). Deep convolutional neural networks for diabetic retinopathy detection by image classification. *Computers & Electrical Engineering*, 72, 274–282.
- Wankhede, P., & Khanchandani, K. (2020). Automated microaneurysms detection from retinal fundus images using pixel intensity rank transform. *Biomedical & Pharmacology Journal*, 13(1), 47–54.
- Wu, Y., Liu, L., Bae, J., Chow, K.-H., Iyengar, A., Pu, C., ... Zhang, Q. (2019). Demystifying learning rate policies for high accuracy training of deep neural networks. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 1971–1980).
- Yacouby, R., & Axman, D. (2020). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of nlp systems* (pp. 79–91).
- Yau, J. W., Rogers, S. L., Kawasaki, R., Lamoureaux, E. L., Kowalski, J. W., Bek, T., ... others (2012). Global prevalence and major risk factors of diabetic retinopathy. *Diabetes care*, 35(3), 556–564.
- Yong, H., Huang, J., Meng, D., Hua, X., & Zhang, L. (2020). Momentum batch normalization for deep learning with small batch size. In *Computer vision—eccv 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part xii 16* (pp. 224–240).
- Zheng, Y., He, M., & Congdon, N. (2012). The worldwide epidemic of diabetic retinopathy. *Indian journal of ophthalmology*, 60(5), 428.

Deep Learning based Optimized Diabetic Retinopathy Detection Platform from Fundus Photographs

by

**Md. Azizul Hakim (191014012)
Tashfat Fatema (191014020)
Md. Mutasim Billah Abu Noman Akanda (192014038)**

*Capstone project (CSE499) submitted in partial fulfillment of the
requirements for the degree of*

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Dr. Mohammed Ashikur Rahman



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF LIBERAL ARTS BANGLADESH**

SUMMER 2022