

# Early Detection of Diabetic Retinopathy using Machine Learning Classifiers



By

Somia Maryam

(Registration No: 00000359400)

Department of Statistics

School of Natural Sciences

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(2024)

# Early Detection of Diabetic Retinopathy using Machine Learning Classifiers



By

Somia Maryam

(Registration No: 00000359400)

A thesis submitted to the National University of Sciences and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Masters of Science in

Statistics

Supervisor: Dr. Tahir Mehmood

School of Natural Sciences

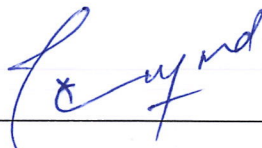
National University of Sciences and Technology (NUST)


Islamabad, Pakistan

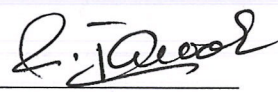
(2024)

## THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis written by Somia Maryam (Registration No. 00000359400), of School of Natural Sciences has been vetted by undersigned, found complete in all respects as per NUST statutes/regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/M.Phil degree. It is further certified that necessary amendments as pointed out by GEC members and external examiner of the scholar have also been incorporated in the said thesis.


Signature:   
Name of Supervisor: Dr. Tahir Mehmood  
Date: 30-05-2024

Signature (HoD):   
Date: 31-5-2024

Signature (Dean/Principal):   
Date: 31.5.2024

**National University of Sciences & Technology****MS THESIS WORK**

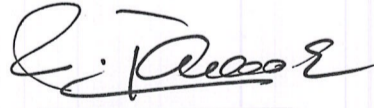
We hereby recommend that the dissertation prepared under our supervision by: Somia Maryam,  
 Regn No. 00000359400 Titled: "Early Detection of Diabetic Retinopathy Using Machine  
 Learning Classifier" accepted in partial fulfillment of the requirements for the award of  
**MS** degree.

**Examination Committee Members**1. Name: DR. SHAKEEL AHMEDSignature: 2. Name: DR. MUHAMMAD ASIF FAROOQSignature: Supervisor's Name: DR. TAHIR MEHMOODSignature: 

  
 Head of Department

31-5-2024  
 Date

**COUNTERSIGNED**Date: 03.05.2024

  
 Dean/Principal

## **DEDICATION**

To my parents for their love, endless support, and encouragement

To my brother for his continuous support and motivation to conquer my life long goals

## ACKNOWLEDGEMENTS

All honor and praise should be given to Allah Almighty, the most merciful and kind, who created the entire universe. He has showered upon me innumerable favors, including the courage and strength to successfully complete my thesis, for which I am incredibly appreciative and thankful to Him. Unquestionably, I want to express my sincere gratitude to my supervisor, Dr. Tahir Mehmood, one of the best professors I've ever had. I owe him a heartfelt thanks for his encouragement, guidance, and, above all, his constant patience during this process. May Allah bless him with His countless blessings. Without his expertise, support, and guidance, this research could not have been completed. Furthermore, because of his efforts and appreciative responses to my questions, I have gained a complete grasp and respect for this field. I also want to express my gratitude to my GEC members Dr. Shakeel Ahmed and Dr. Muhammad Asif Farooq for their support and guidance in completing this thesis. And lastly, a million thanks to my brother and friends, for their support throughout my studies. GOD bless you all for making this difficult journey a success.

# Contents

<b>ACKNOWLEDGEMENT</b>	<b>IV</b>
<b>LIST OF TABLES</b>	<b>V</b>
<b>LIST OF FIGURES</b>	<b>VIII</b>
<b>LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS</b>	<b>VIII</b>
<b>ABSTRACT</b>	<b>IX</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Machine Learning in Medical Imaging . . . . .	1
1.2 An Overview of Diabetic Retinopathy . . . . .	3
1.2.1 Diabetic Retinopathy . . . . .	4
1.2.2 Stages of Diabetic Retinopathy . . . . .	6
1.2.2.1 Mild Non-Proliferative Diabetic Retinopathy . . . . .	6
1.2.2.2 Moderate Non-Proliferative Diabetic Retinopathy . . . . .	7
1.2.2.3 Severe Non-Proliferative Diabetic Retinopathy . . . . .	7
1.2.2.4 Proliferative Diabetic Retinopathy . . . . .	7
1.2.3 Causes of Diabetic Retinopathy . . . . .	8
1.2.4 Symptoms of Diabetic Retinopathy . . . . .	8
1.2.5 Risk Factors . . . . .	8
1.3 Classification . . . . .	9
1.3.1 Binary Classification . . . . .	10
1.3.2 Multi-class Classification . . . . .	11
1.4 K-Nearest Neighbour (KNN) . . . . .	11
1.4.1 How does KNN work? . . . . .	12
1.4.2 How to determine the value of $k$ ? . . . . .	13

1.4.3	Distance Metrics Used in KNN Algorithm	13
1.4.3.1	Euclidean Distance(L2 Distance)	14
1.4.3.2	Manhattan Distance (L1 Distance)	14
1.4.3.3	Hamming Distance	14
1.5	Support Vector Machine (SVM)	14
1.6	Neural Network	17
1.7	Convolutional Neural Network (CNN)	18
1.7.1	Convolutional Layers	19
1.7.1.1	Padding	20
1.7.1.2	Depth (Number of Filters/Kernels)	21
1.7.1.3	Stride	22
1.7.2	Pooling Layer	22
1.7.2.1	Max Pooling	22
1.7.2.2	Average Pooling	23
1.7.3	Fully Connected Layer	23
1.8	Residual Neural Network (ResNet)	23
1.9	Activation Functions	24
1.9.1	ReLU (Rectified Linear Unit)	25
1.9.2	Leaky ReLU	25
1.9.3	Sigmoid	25
1.9.4	Softmax	26
1.9.5	Tanh (Hyperbolic Tangent)	26
1.10	Problem Statement	27
1.11	Objective	27
1.12	Organization of the Thesis	27
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>29</b>
<b>3</b>	<b>MATERIALS AND METHODS</b>	<b>32</b>
3.1	Data Acquisition	32
3.2	Network Architecture for Image Classification	33
3.2.1	K-Nearest Neighbors (KNN)	33
3.2.2	Support Vector Machine (SVM)	34
3.2.3	Convolutional Neural Network (CNN)	35
3.2.4	Residual Neural network (ResNet-50)	36
3.3	Evaluation Metrics for Classification	39



3.3.1	Confusion Matrix . . . . .	39
3.3.2	Accuracy . . . . .	39
3.3.3	Sensitivity . . . . .	39
3.3.4	Specificity . . . . .	40
3.3.5	F1-Score . . . . .	40
3.3.6	Precision . . . . .	40
3.3.7	Error Rate . . . . .	41
<b>4</b>	<b>RESULTS AND DISCUSSION</b>	<b>42</b>
<b>5</b>	<b>CONCLUSION</b>	<b>56</b>

# List of Tables

1.1	Diabetic retinopathy progresses through stages marked by microaneurysms, hemorrhages, venous beading, intraretinal abnormalities, and neovascularization. . . . .	6
3.1	Data distribution summarizes how data points are arranged across different categories depending on the severity of the disease within a dataset. . . . .	32
3.2	ResNet-50 consists of convolutional layers and residual blocks organized into four levels. Each level includes convolutional layers with diverse filter sizes, along with max-pooling and average-pooling operations for feature extraction and spatial reduction. . . . .	38
4.1	Overview of Image Preprocessing and Performance Metrics . . . . .	43
4.2	Comparison of Performance Metrics for Different KNN Configurations for binary classifications . . . . .	44
4.3	SVM Model having binary classification Performance Metrics for Different Kernels . . . . .	47
4.4	Comparison of Performance Metrics having multi-class classification for Different KNN Configurations . . . . .	51
4.5	SVM Model having multi-class classification Performance Metrics for Different Kernels . . . . .	54

# List of Figures

1.1	Diverse applications of medical imaging, including X-rays, MRI, ultrasound, and PET scans, showcasing their pivotal role in diagnosing and treating various medical conditions. . . . .	3
1.2	A Visual Exploration of Eye Complications Caused by Diabetes . . . . .	4
1.3	Variation in blood vessels including dilation, constriction, and irregular branching occur in the retina, resulting in impaired vision and serving as important indicators of disease advancement . . . . .	5
1.4	comparison of the visual acuities of a healthy individual and a patient with diabetic retinopathy . . . . .	5
1.5	Retinal images depicting diabetic retinopathy with varying disease grades, showcasing the spectrum of severity and progression . . . . .	7
1.6	This figure illustrates a gray scale image in a high-dimensional space, where each axis $(x_1, x_2, \dots, x_n)$ reveals detailed layers of grayscale imagery. . . . .	9
1.7	This figure demonstrates the classification of three different images. . . . .	10
1.8	Multiclass classification makes decisions across several categories by navigating through a spectrum of options, whereas binary classification concentrates on differentiating between two classes. . . . .	11
1.9	Before and After KNN . . . . .	12
1.10	Working of KNN . . . . .	13
1.11	A three-class classification scenario in SVM, with distinct data points representing different classes . . . . .	15
1.12	One-to-One SVM method focuses on pairwise class separation, ignoring points from other classes, as depicted by the red-blue line . . . . .	16
1.13	One-to-Rest SVM method employs hyperplanes to simultaneously separate each class from all others, utilizing all points in the classification process, as exemplified by the green line. . . . .	16
1.14	Basic structure of Neural Network . . . . .	17
1.15	An illustration of the fundamental structure of a convolutional neural network . . . . .	19

1.16	Filters move across input images, performing element-wise multiplications and summations to find patterns and enhance feature representation for deep learning models. . . . .	20
1.17	By placing zeros around a picture, padding (zero padding) maintains uniformity in size and speeds up computing processes while ensuring equal dimensions. . . . .	21
1.18	The kernel is a matrix that sequentially moves across the input data, applies the dot product operation to a portion of the data, and subsequently generates a matrix of dot product results. . . . .	21
1.19	In order to preserve spatial sampling and enhance the effectiveness of feature recognition, the filters move throughout the image, pixel by pixel with a stride of one or two pixels at a time with a stride of two. . . . .	22
1.20	Two types of pooling: average pooling, which computes the average to summarise data while minimising spatial dimensions, and maximum pooling, which retains maximum values to capture dominating features. . . . .	23
1.21	Residual unit structure diagram . . . . .	24
1.22	Visual depiction of activation functions . . . . .	27
3.1	The architecture of basic CNN structure, and the various layers that make up the CNN model on the DR image data set . . . . .	36
3.2	Visual architecture of ResNet-50 includes convolutional layers, followed by four levels of residual blocks. . . . .	38
4.1	The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications. . . . .	45
4.2	The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications. . . . .	48
4.3	The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications. . . . .	49
4.4	The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications. . . . .	50
4.5	Confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications. . . . .	53
4.6	The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications. . . . .	55

# LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
DR	Diabetic Retinopathy
DM	Diabetes Mellitus
DED	Diabetic Eye Disease
MA	Microaneurysms
HE	Hard Exudates
NPDR	Non-proliferative Diabetic Retinopathy
PDR	Proliferative Diabetic Retinopathy
KNN	K-Nearest Neighbour
SVM	Support Vector Machine
CNN	Convolutional Neural Network
ResNet	Residual Neural Network
MCC	Mathew's Correlation Coefficient

# ABSTRACT

Diabetic Retinopathy (DR) is a serious complication of diabetes that can lead to blindness if not detected early. Diabetes, along with diabetic retinopathy, is associated with causing harm to the eyes, characterized by the leakage of blood vessels damaging the retina, leading to microaneurysms (MA), hard exudates (HE), and soft exudates (cotton wool spots). As a chronic condition, DR poses a growing challenge worldwide, often resulting in visual impairment or blindness if left undiagnosed. Early detection is crucial, yet challenging, as symptoms manifest only after significant retinal changes occur. This study investigates the effectiveness of a multimodal approach using machine learning classifiers for the early detection of DR, employing both binary and multiclass classification methods. We utilized K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and ResNet-50 algorithms to classify retinal images. In binary classification, designed to distinguish between DR and non-DR images, KNN achieved an accuracy of 88%, SVM reached 92%, and ResNet-50 outperformed both with an accuracy of 98%. These results demonstrate the potential of machine learning, particularly deep learning with ResNet-50, in accurately identifying the presence of DR. For multiclass classification, which aims to classify images into different stages of DR (no DR, mild, moderate, severe, and proliferative DR), KNN achieved an accuracy of 70%, SVM achieved 71%, and ResNet-50 achieved the highest accuracy of 82%. The study's findings highlight the superior performance of ResNet-50 in both binary and multiclass classification tasks, suggesting its suitability for automated DR screening systems. The implementation of such advanced machine learning models can aid in early diagnosis, thus facilitating timely treatment and potentially preventing vision loss in diabetic patients.

**Keywords:** Diabetic retinopathy, classification, machine learning, SVM, ResNet-50, Exudates, microaneurysms, blood vessels.

# Chapter 1

## INTRODUCTION

The eye is unquestionably a significant part of the human body. In the context of human existence, the value of the human eye is enormous. Our eyes, which are the main visual organs, are essential to both our daily activities and general wellbeing. One of our most important senses, vision enables us to interact and perceive the world around us. The eye's relevance extends into disciplines like ophthalmology and vision science, where work is done to improve and preserve its function, as a subject of ongoing research and medical developments.

A critical area of research in contemporary healthcare is the early diagnosis of diabetic retinopathy using machine learning classifiers. The primary factor contributing to blindness in diabetic patients is diabetic retinopathy, and early identification is essential for successful treatments. Automating the detection and classification of diabetic retinopathy from retinal images has shown tremendous potential when combined with modern image processing algorithms. This study not only showcases the significant potential for artificial intelligence in revolutionizing healthcare but also underscores the profound positive influence it can exert on patients well-being by enabling early and precise disease identification and classification.

### 1.1 Machine Learning in Medical Imaging

In the field of medical imaging, machine learning and deep learning have become crucial tools transforming diagnostic and therapeutic procedures. Machine learning automatically extracts complex patterns and data from images generated by modalities like X-rays, CT scans, and MRIs, aiding radiologists and physicians in prompt and accurate diagnoses, especially for diseases like cancer where minor irregularities may be imperceptible. Deep learning, a subset within the realm of machine learning, excels in

medical imaging by employing neural networks with multiple layers, particularly convolutional neural networks (CNNs), which revolutionize tasks like image segmentation, feature extraction, and classification. CNNs are instrumental in tumour identification, organ segmentation, and image generation, enabling unparalleled precision in diagnosis and treatment planning. Despite the remarkable progress, the expansion of data sets in the past decade has strained processors and hardware capacity, impeding machine learning algorithms' processing within projected timelines. These algorithms utilize feature matrices to classify data precisely when supplied with relevant characteristics, particularly in assessing the high-dimensional properties of medical images. The advancement of multimedia data, including image, text, video, and audio, underscores the need for stronger search engines, especially in fields like education, medicine, and entertainment. However, the scarcity of publicly available medical data impedes progress, although recent advancements have updated medical science, facilitating the diagnosis of illnesses like skin, lung, breast, eye, and blood cancer using conventional procedures.

The future of healthcare is quite bright when machine learning and deep learning are combined in medical imaging. These technologies allow for not only quicker and more precise diagnosis but also the possibility of individualised treatment strategies and illness progression prediction modelling. The incorporation of artificial intelligence into medical imaging is poised to improve patient care, outcomes, and drive advancements in the comprehension and management of a range of medical problems as the area continues to develop. A critical area of research in contemporary healthcare is the early diagnosis of diabetic retinopathy using machine learning classifiers. The major cause of blindness in patients with diabetes is diabetic retinopathy, and early identification is essential for successful treatments. Automating the detection and classification of diabetic retinopathy from retinal images has shown tremendous potential when combined with modern image processing algorithms. This study not only showcases the significant potential for artificial intelligence in revolutionizing healthcare but also underscores the profound positive influence it can exert on patients well-being by enabling early and precise disease identification and classification.



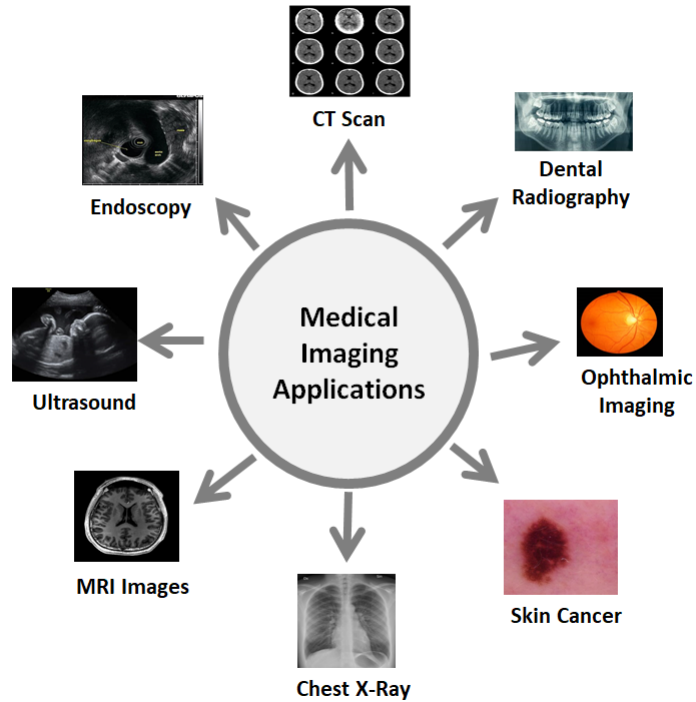


Figure 1.1: Diverse applications of medical imaging, including X-rays, MRI, ultrasound, and PET scans, showcasing their pivotal role in diagnosing and treating various medical conditions.

## 1.2 An Overview of Diabetic Retinopathy

According to the World Health Organization, diabetes affects over 347 million people worldwide and is going to be the seventh dominant cause of mortality by 2030. Diabetes patients tend to develop retinal irregularities over time as a result of a recent problem called diabetic retinopathy. A person over 30 who has had diabetes for more than fifteen years is 78% likely to develop diabetic retinopathy. Diabetic Retinopathy is caused by a long history of diabetic Mellitus. Retinopathy is the term for injury to the retina that results in choking, leaking, and irregular blood vessel growth. Diabetic Retinopathy is asymptomatic; it does not affect vision until it reaches the advanced stage. As a result, screening for diabetic retinopathy is critical.

Diabetic retinopathy, or diabetic eye disease (DED), is a condition where there is damage to the retinal blood vessels of the human eye. The term Retinopathy refers to damage in the blood vessel of the retina, which is the light-receptive membrane of the eye. The most prevalent cause of visual impairment or loss of vision in diabetic people

is diabetic retinopathy, a common illness. It is the third most deadly illness, behind cancer and heart disease. "Diabetic Retinopathy" is one of the effects of diabetes, however it affects most of the body's organs. Diabetic retinopathy affects the patient after damage to the vessels of the retina, as illustrated in Figure 1.2

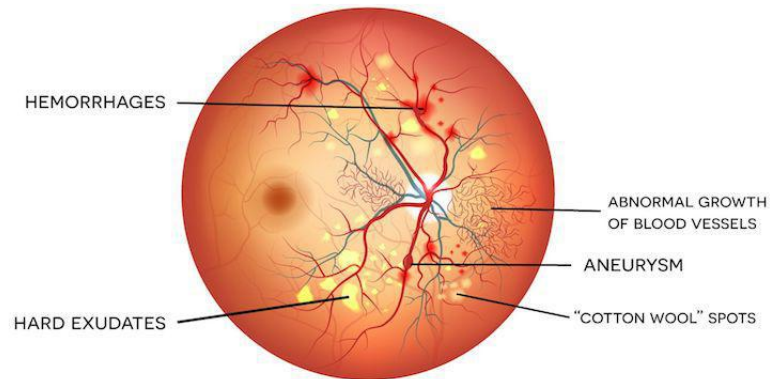


Figure 1.2: A Visual Exploration of Eye Complications Caused by Diabetes

### 1.2.1 Diabetic Retinopathy

Diabetes-related retinal illness in humans is called diabetic retinopathy (DR). It is a microvascular complication of diabetes that results from long-term elevated blood sugar levels harming the retina, which is a crucial component of the eye that provides vision. DR is among the leading factors contributing to blindness in those between the ages of 25 and 74 worldwide. The most common reason for blindness among people of working age in developed nations is diabetic retinopathy. It is intimately correlated with diabetes mellitus and results in structural alterations in the retina that cause irreparable damage. The medical retinal exam is expensive in terms of time and human resources because it must be done on a frequent basis for diabetic patients.

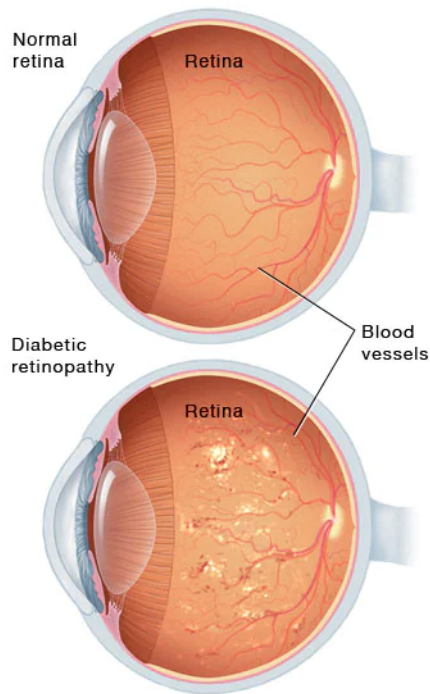


Figure 1.3: Variation in blood vessels including dilation, constriction, and irregular branching occur in the retina, resulting in impaired vision and serving as important indicators of disease advancement



Figure 1.4: comparison of the visual acuities of a healthy individual and a patient with diabetic retinopathy

Diabetes mellitus (DM) is the primary cause of diabetic retinopathy, a type of disease that affects the retina due to elevated blood sugar. The blood supply is cut off as a result of the blood cells' circulation in the retina blocking blood vessels. As a result, the eye creates new blood vessels to nourish the retina, but these vessels grow improperly and soon start to leak, which can potentially cause blindness. Figures 1.3

and 1.4 illustrate the differences between the DR and normal eyes as well as the visual impairments that DR patients face.

Consequently, a request has been made for an automated detection method that would assess the retinal images automatically and send only the patients who were affected to the ophthalmologist. This could lead to quicker patient diagnosis and less effort for the medical staff. The goal of this study is to create a machine learning algorithm-based approach to solve the Kaggle Diabetic Retinopathy Detection challenge, which involves detecting diabetic retinopathy using digital colour fundus photographs of the retina. Clinical studies indicate that DR progresses through five stages. Table 1.1 lists the DR stages in the fundus eye along with associated clinical symptoms.

DR stages	DR Classification	Clinical Signs
0	Normal	No abnormality
1	Mild	Microaneurysms only
2	Moderate	Microaneurysms, hard/soft exudates, hemorrhages
3	Severe	All abnormalities are observed at all four quadrants
4	PDR	Abnormal growth of blood vessels with all abnormalities at all four quadrants

Table 1.1: Diabetic retinopathy progresses through stages marked by microaneurysms, hemorrhages, venous beading, intraretinal abnormalities, and neovascularization.

## 1.2.2 Stages of Diabetic Retinopathy

It is a degenerative eye disorder that occurs in four stages and two types. There are two categories: proliferative and nonproliferative. Nonproliferative disease is in its early stages, while proliferative illness is a more advanced kind.

### 1.2.2.1 Mild Non-Proliferative Diabetic Retinopathy

This is the first stage of diabetic retinopathy, which is identified by tiny blood vessel enlargements in the retina. These enlarged areas are the first stage of microaneurysms, also known as background retinopathy. At this stage, the macula may enlarge due to minute quantities of fluid seeping into the retina. This is quite close to the middle of the retina.

### 1.2.2.2 Moderate Non-Proliferative Diabetic Retinopathy

Another name for the second stage is pre-proliferative retinopathy. The blood supply to the retina is disrupted due to an increase in the size of tiny blood vessels, which complicates the retina's regular feeding process. The macula clogged with blood and other fluids as a result.

### 1.2.2.3 Severe Non-Proliferative Diabetic Retinopathy

Another name for this is proliferative retinopathy. At this point, a greater portion of the retina's blood vessels block, significantly reducing the amount of blood flowing to this region. The body now gets signals to begin producing new blood vessels in the retina.

### 1.2.2.4 Proliferative Diabetic Retinopathy

At this point in the disease's progression, the retina is starting to develop new blood vessels. There is an increased chance of fluid leakage since these blood vessels are frequently brittle. This can lead to a variety of visual issues, including fuzziness, a narrower range of vision, and even blindness. The stages of diabetic retinopathy Normal, Mild, Moderate, Severe, and Proliferative are depicted in Figure 1.5.

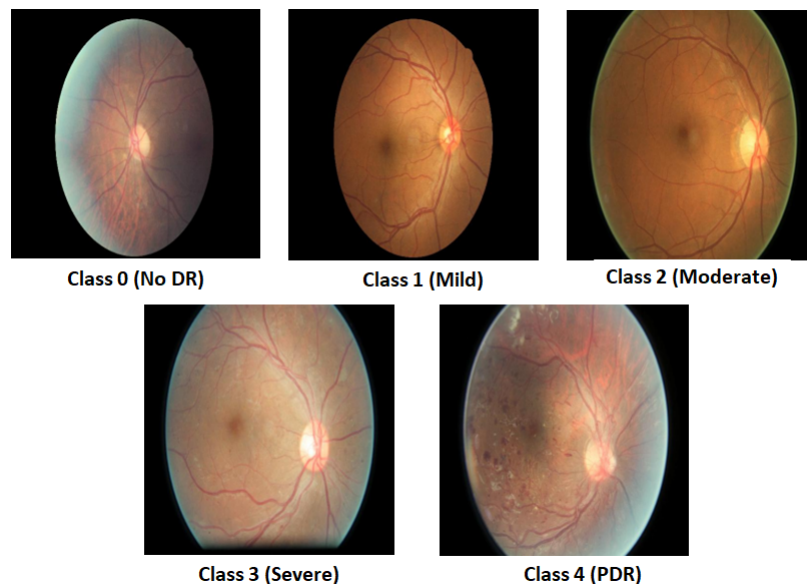


Figure 1.5: Retinal images depicting diabetic retinopathy with varying disease grades, showcasing the spectrum of severity and progression

### **1.2.3 Causes of Diabetic Retinopathy**

Long-term elevated blood sugar levels in diabetics are the cause of diabetic retinopathy. Overconsumption of blood sugar can eventually clog the tiny blood vessels that nourish the retina, preventing blood flow to the area. As a result, the eye tries to grow new blood vessels. However, the improper development and easy leakage of these new blood vessels can result in further loss of vision.

### **1.2.4 Symptoms of Diabetic Retinopathy**

Symptoms of diabetic retinopathy are usually nonexistent in the early stages. When an illness is further advanced, symptoms typically take longer to manifest. Diabetic retinopathy usually affects both eyes. The warning signs and symptoms of this illness may include

1. Blurred or distorted vision
2. Difficulty seeing at night
3. Floaters or spots in the visual field
4. Abrupt loss of eyesight in one or both eyes
5. Changes in color perception
6. Poor color vision
7. Eye pain or pressure
8. Fluctuations in vision

It's important to note that in the early stages of diabetic retinopathy, there may be no noticeable symptoms. That's why it's important for people with diabetes to have regular eye exams with an ophthalmologist or optometrist to check for any signs of diabetic retinopathy or other eye conditions.

### **1.2.5 Risk Factors**

Every diabetic has the potential to develop diabetic retinopathy. The prevalence of the disorder in the eyes may be increased by the following factors:

1. Duration of diabetes
2. Poor control of blood sugar

3. Elevated blood pressure, or hypertension
4. High cholesterol
5. Pregnancy
6. Smoking or using tobacco products
7. Genetics - a family history of diabetic retinopathy or diabetes can increase the risk of developing the condition.
8. Obesity or being overweight

It's essential to keep in mind that while these factors may increase the likelihood of developing diabetic retinopathy, not all people with diabetes will develop this condition. To lower the risk of diabetic retinopathy and other issues connected to diabetes, it is essential to properly control and monitor blood sugar levels, have eye exams on a regular basis, and lead a healthy lifestyle.

### 1.3 Classification

Classification is under the supervised learning method which is concerned with developing a model for variable prediction. The response variable only deals with categorical or qualitative data while the explanatory variables can be quantitative or qualitative. Classification techniques are also named Classifiers.

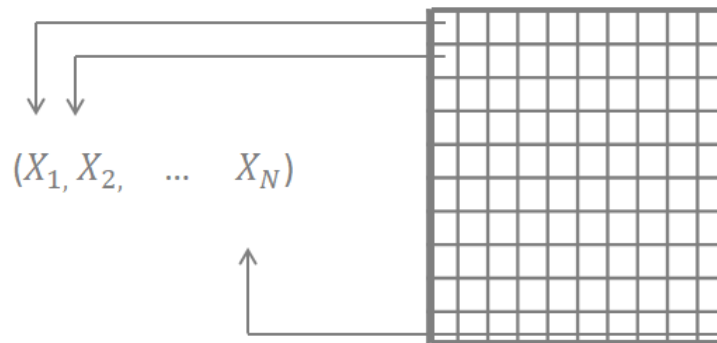


Figure 1.6: This figure illustrates a gray scale image in a high-dimensional space, where each axis  $(x_1, x_2, \dots, x_n)$  reveals detailed layers of grayscale imagery.

Since the classification works based on vectors, so first step is to read the images and transform every image into a vector of pixel values. An image having  $N$  pixels in an  $N$ -dimensional space is described by a vector and also it is assumed that it is a gray scale image as shown in Figure 1.6. The pixel value is the weight of each dimension.

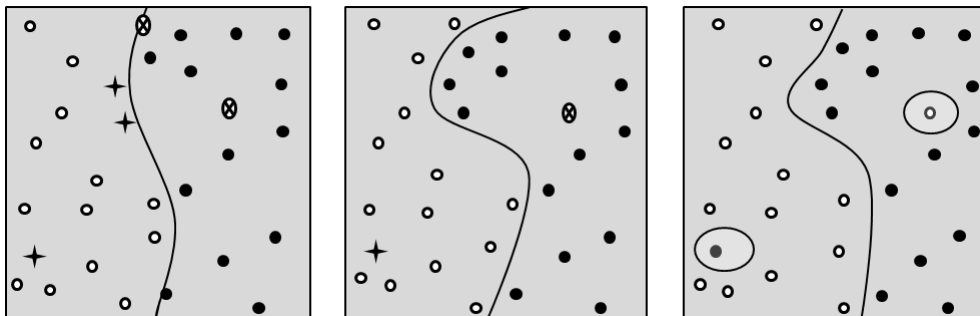


Figure 1.7: This figure demonstrates the classification of three different images.

The goal is to split the filled dots from the unfilled circles in Figure 1.7. The boundary is almost linear on the left side of the figure, but some points are not correctly classified. The right image of the figure has no errors, but the classifier is quite complex. Finally, the middle image is a compromise between both extremes, and the classes are separated efficiently.

Classification requires training datasets for modeling purposes. A model uses that training data set which gives the most accurate representation of the problem to calculate input data to specify class labels. For the classification of disease, a critical area of research in contemporary healthcare is the early diagnosis of diabetic retinopathy, the fundas images are the most demanded medical examination owing to their effectiveness. Classification involves four classes, No DR, Mild DR, Moderate DR, Severe DR and Proliferative. Our research problem is to classify disease and to map unique numeric values for each class. In classification tasks, classes can be divided into different types according to the number and nature of the classes involved. These different types of classes in classification highlight the varying complexities and characteristics of classification tasks. The selection of the appropriate type of classification relies on the nature of the problem, the available data, and the specific goals of the application. The commonly used types of classes in classification are:

### 1.3.1 Binary Classification

In machine learning, binary classification falls under the category of supervised learning since it is supported by high-quality training data that includes examples from two classes and labels each example as either class 0 or class 1 using a set of feature



values. In order to predict the unobserved binary labels of new instances from their seen feature values, a binary decision rule is first built using the training data. A broad class of algorithms that automatically learn prediction rules from training data is represented by binary classification [1]. Classifying incidents into one of two categories, typically expressed as 0 or 1, positive or negative; true or untrue; etc. is the aim. For applications like email spam detection, sentiment analysis, and medical diagnosis, this kind of categorization is employed.

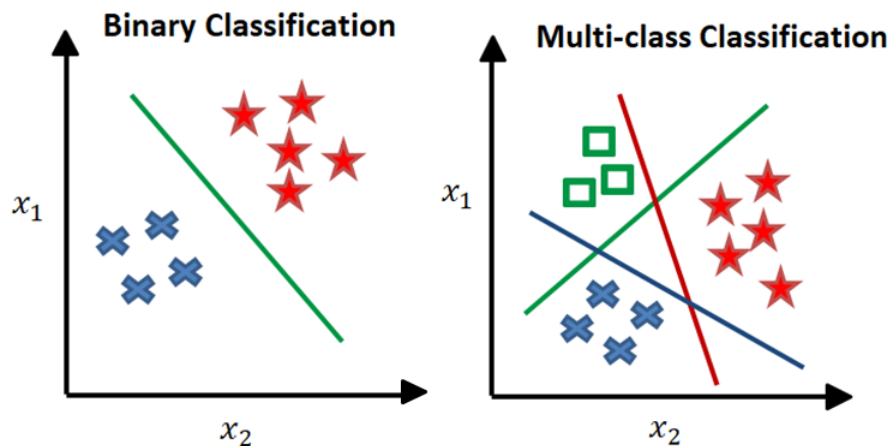


Figure 1.8: Multiclass classification makes decisions across several categories by navigating through a spectrum of options, whereas binary classification concentrates on differentiating between two classes.

### 1.3.2 Multi-class Classification

The term "multi-class classification" describes machine learning classification problems involving more than two classes. Performance measures are very useful for comparing and assessing different machine learning techniques or categorization models [2]. Sorting instances into one of several classes is the aim of this kind of machine learning classification problem. This kind of classification is used for jobs such as photo, text, and audio recognition classification.

## 1.4 K-Nearest Neighbour (KNN)

It is a very simple, easy to understand, and versatile machine learning algorithm. One of the core machine learning techniques is K Nearest Neighbour. Evelyn Fix and Joseph Hodges pioneered the development of the k-nearest neighbors algorithm in 1951, introducing the concept of utilizing the closest neighboring data points to make predictions

or classifications. Machine learning models forecast output values based on a variety of input values.

It is a type of supervised machine learning algorithm used for both regression and classification. However, it is mainly used for classification predictive problems in industry. The classification of the data point depends on the classification of its neighbor. KNN tries to predict the correct class for the test data by measuring the distance between each training point and the test data. The  $K$  locations that are closest to the test data should then be selected.

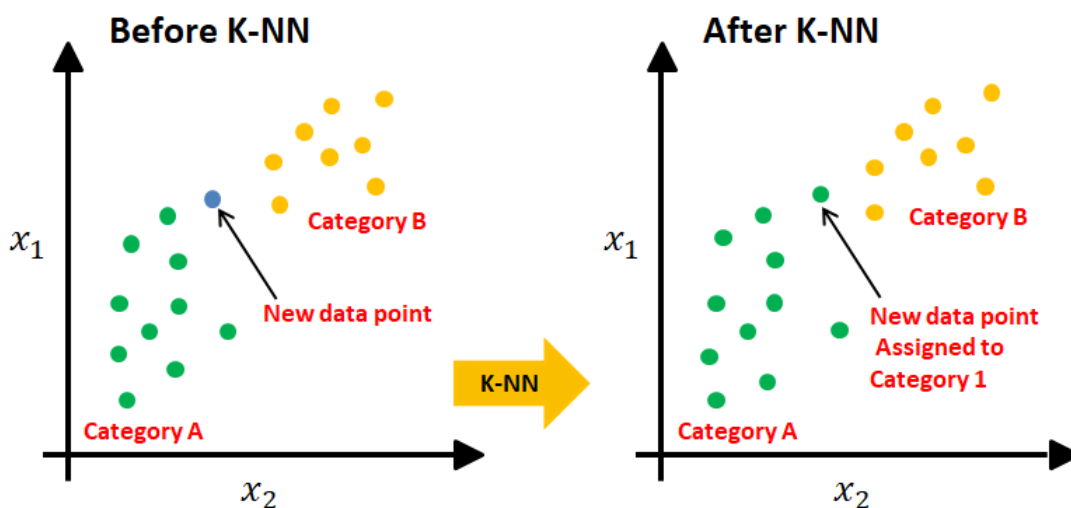


Figure 1.9: Before and After KNN

### 1.4.1 How does KNN work?

It utilizes "feature similarity" to forecast the values of new data points, also indicating that the new data point's value will depend on how similar it resembles the point in the training set. We may comprehend how it functions by taking the subsequent actions. By doing the following, we can understand how it operates.

1. Any method implementation requires a data set. Thus, we also need to load the training and testing data during the first KNN stage.
2. The next step is to find the value of  $k$ , or the closest data points. Any integer  $k$  is possible.

3. To calculate the distance, use any of the following techniques: Manhattan, Euclidean, or Hamming distance is the difference between each row in the test and training data. The most used approach for calculating the distance is Euclidean.
4. For these  $k$  neighbours, find the total number of data points in each category.
5. Allocate the newly acquired data points to the group with the highest neighbour count.
6. The model is ready.

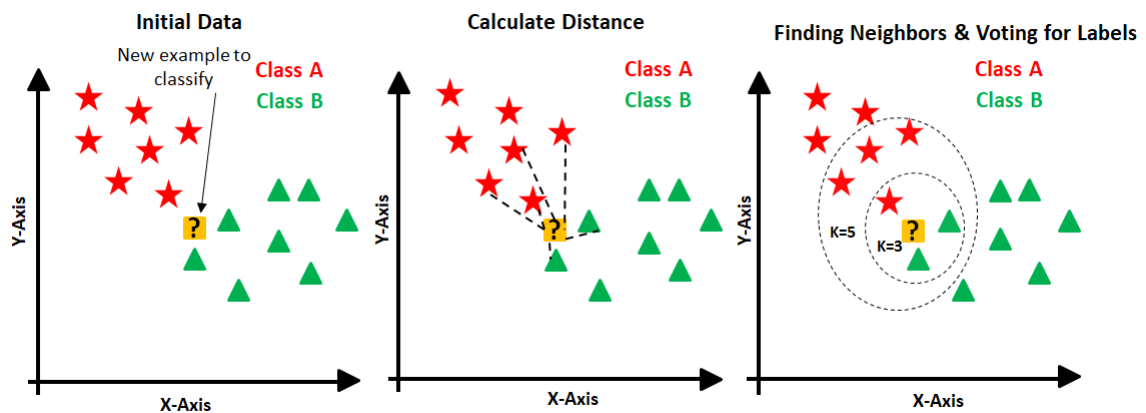


Figure 1.10: Working of KNN

### 1.4.2 How to determine the value of $k$ ?

Selecting the optimal " $k$ " value can be a trial-and-error process since there's no definitive method to determine the best choice. However, a commonly favored value for  $k$  is 5. To determine the suitable  $k$  value for our data, we conduct multiple iterations of the KNN algorithm using different  $k$  values. We evaluate the performance using accuracy as our metric. If accuracy varies in proportion to changes in  $k$ , it indicates a promising choice for our  $k$  value.

### 1.4.3 Distance Metrics Used in KNN Algorithm

Several distance measurements are commonly used in the K-nearest neighbours algorithm to determine the level of similarity or dissimilarity among data points. The choice of a particular distance metric, which is contingent upon the properties of the data and the particular task at hand, can significantly impact the algorithm's performance. In KNN, some of the distance measures that are commonly used are:

### 1.4.3.1 Euclidean Distance(L2 Distance)

In KNN, this is the most often utilised distance metric. In Euclidean space, it determines the straight-line distance between the two points. The Euclidean distance for two points in a two-dimensional space,  $(x_1, y_1)$  and  $(x_2, y_2)$ , is as follows:

$$d(x_1, x_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1.1)$$

### 1.4.3.2 Manhattan Distance (L1 Distance)

It determines the distance by adding the absolute differences between coordinates along each dimension; it is also referred to as the taxicab distance or city block distance. In two dimensions, it is represented by:

$$d(x_1, x_2) = |x_2 - x_1| + |y_2 - y_1| \quad (1.2)$$

### 1.4.3.3 Hamming Distance

Comparing binary or categorical data is the main application for this distance metric. The number of points at which two binary strings diverge is counted. The properties of the data and the specific issue at hand determine which distance measure is best for KNN. Generally, experience in the relevant field and testing lead the selection of the most appropriate distance metric.

## 1.5 Support Vector Machine (SVM)

It is a supervised machine learning algorithm originally proposed by Cortes and Vapnik, are powerful models widely employed for classification and regression tasks. SVM is particularly popular in the field of machine learning because it can handle data that is high-dimensional and effectively classify both linearly and non-linearly separable datasets.

"Multi-class classification" in machine learning refers to classification tasks involving more than two classes. Performance measures are also very helpful for comparing and assessing different machine learning or classification models. [2]. The objective of this sort of machine learning classification issue is to place instances into one of many classes. For tasks like speech recognition, text classification, and image classification, this form of classification is employed. Multi-class classifiers can be easily obtained by using many binary classifiers. There are two well-known methods: "one versus All" and "pairwise classification" (or "one versus one").

1. **One vs. All:** Utilises  $M$  classifiers if the number of classes is  $M$ . Each of these classifiers has been taught to distinguish one class from every other class. The class for which the answer is higher is the ultimate outcome.
2. **One vs. One:** For every pair of classes, a classifier is trained, and the class is chosen using a voting mechanism.

Although the first option appears to be simpler because there are fewer binary classifiers, but this is not always the case. A separation between each pair is easier to find than a separation between one class and all the others. Therefore, even if it must train  $\frac{M(M-1)}{2}$  classifiers in place of  $M$  classifiers, the "one versus one" classifier can still be faster than the "one versus all".

Examine the image below as an illustration of a three-class categorization issue: green, red, and blue.

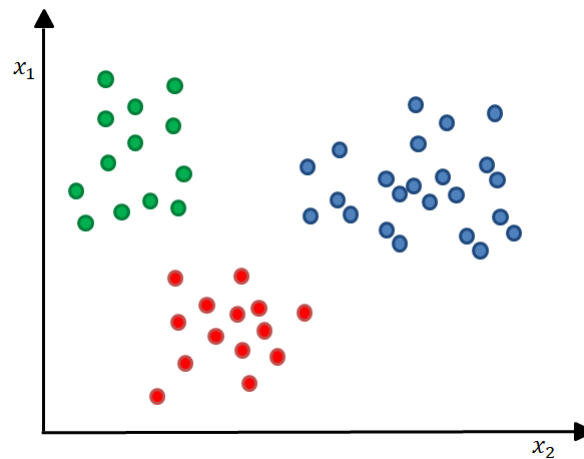


Figure 1.11: A three-class classification scenario in SVM, with distinct data points representing different classes

The following results are achieved when the two approaches are applied to this data set:

A hyperplane is required to divide each class into two according to the One-to-One approach, which ignores the points in the third class. This suggests that the separation takes into account only the points from the two classes in the current split. For instance, the red-blue line aims to maximise the distance between the blue and red points only. It has nothing to do with green points.

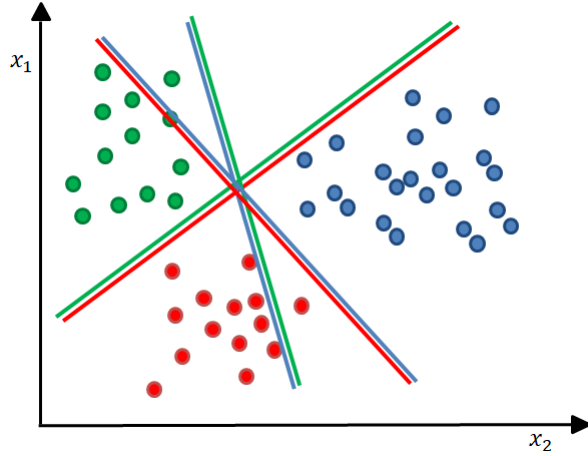


Figure 1.12: One-to-One SVM method focuses on pairwise class separation, ignoring points from other classes, as depicted by the red-blue line

A hyperplane is required in the One-to-Rest method in order to divide a class from all others simultaneously. This indicates that all points are considered in the separation process and are split into two groups: one group is for class points, and the other group is for all other points. The green line, for instance, seeks to maximise the distance between green points and every other point simultaneously.

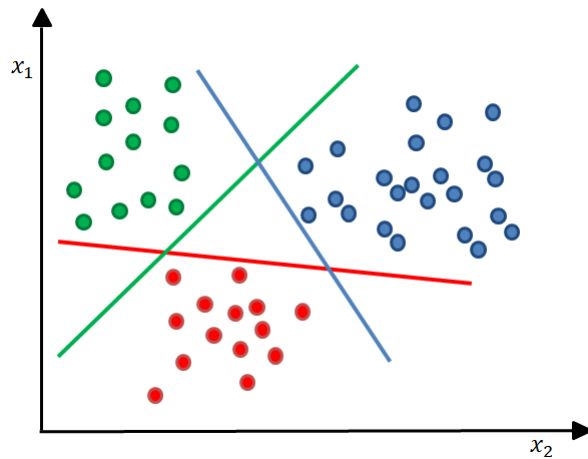


Figure 1.13: One-to-Rest SVM method employs hyperplanes to simultaneously separate each class from all others, utilizing all points in the classification process, as exemplified by the green line.

## 1.6 Neural Network

A neural network, a popular machine learning technique, was created as inspiration for the topologies of biological brain networks. It replicates how the human brain functions by building an artificial neural network [3]. They are composed of interconnected "neurons," or nodes, set up in layers. It provides a wide range of robust brand-new techniques for dealing with pattern recognition, data analysis, and control concerns. A feed-forward neural network is the most basic type of neural network. They consist of a hidden layer or layers, an output layer, and an input layer. FNNs are used for a range of tasks, such as picture classification, regression analysis, and basic pattern recognition [4].

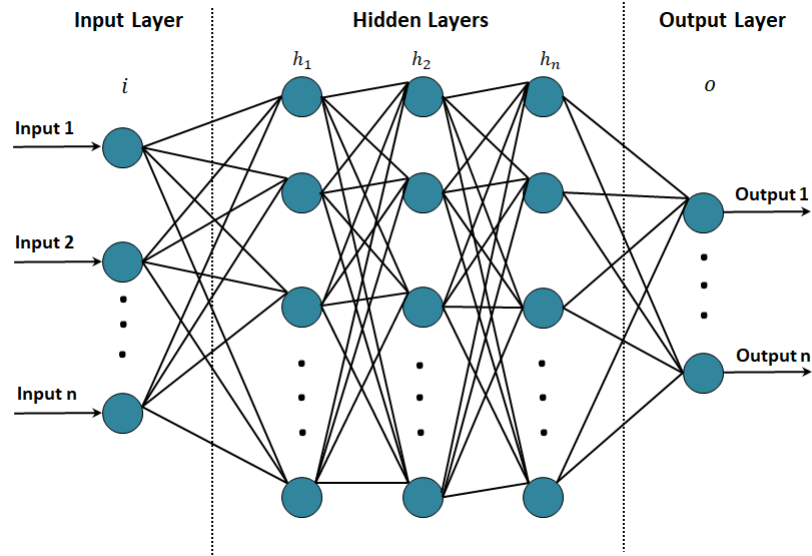


Figure 1.14: Basic structure of Neural Network

In a neural network (NN), a layer called bias can have any number of nodes because the layers are independent of one another. The bias nodes' initial value is always 1. A bias value is significant because it enables the activation function to be moved to the right or left, which may be essential for the success of ANN training. The input and output nodes of a NN used as a classifier will correspond to the input features and output classes. However, when the NN is utilised to approximate a function, it normally contains an input and an output node. There are, however, more essential designed hidden nodes than input nodes.

CNNs are capable of processing pictures and other grid-like data. They use convolutional layers in order to automatically learn hierarchical characteristics, They therefore perform exceptionally well in picture segmentation, object detection, and classification [5]. RNNs are designed to handle sequential data, such as time series and natural language. They are excellent for tasks like language modelling and voice recognition because they feature a feedback loop that enables information to be transmitted from one phase of the sequence to the next [6]. Recent studies have concentrated on developing neural networks with fewer parameters to decrease processing costs and boost effectiveness [7]. Many applications, including natural language processing (NLP), audio and speech processing, visual data processing, and others, have shown that the most recent DL techniques perform exceptionally well [8]. Bag of words and the scale-invariant feature transform [9]. A novel feature that is put into use and effectively shown generates a new field of study that is investigated for decades [10]. Deep learning algorithms can currently perform challenging tasks including audio and image identification, natural language processing, and autonomous vehicles. A few well-known deep learning frameworks are TensorFlow, PyTorch, and Keras.

New architectures are always being developed for specific tasks and applications, despite the fact that there are numerous different types of neural networks, each with a unique design and application area. Here are a few illustrations of the different kinds of neural networks we used in this research.

## 1.7 Convolutional Neural Network (CNN)

The Convolutional Neural Network, which takes its name from the "convolution" matrix function in mathematics, is one widely used deep neural network. Convolutional networks are essentially neural networks with at least one layer that substitutes convolution for generic matrix multiplication. CNNs are a type of Artificial Neural Network (ANN) that are frequently used for the analysis of visual content. They are used across multiple domains, such as recommender systems, classification, segmentation, medical image analysis, and brain-computer interfaces. Notably, CNNs have demonstrated great promise in picture-related applications, such as handling a huge image classification dataset (ImageNet), computer vision, and natural language processing (NLP), as indicated by the findings [11].

The CNN architecture is truly impressive. The three essential layers of CNN are shown in Figure 1.15 the input layer, hidden layers, and output layer, also known as the convolutional layer, pooling layer, and fully-connected layer, respectively. The figure demonstrates how the CNN processes the input provided to the model in order to generate evaluated outputs once the model has been trained.



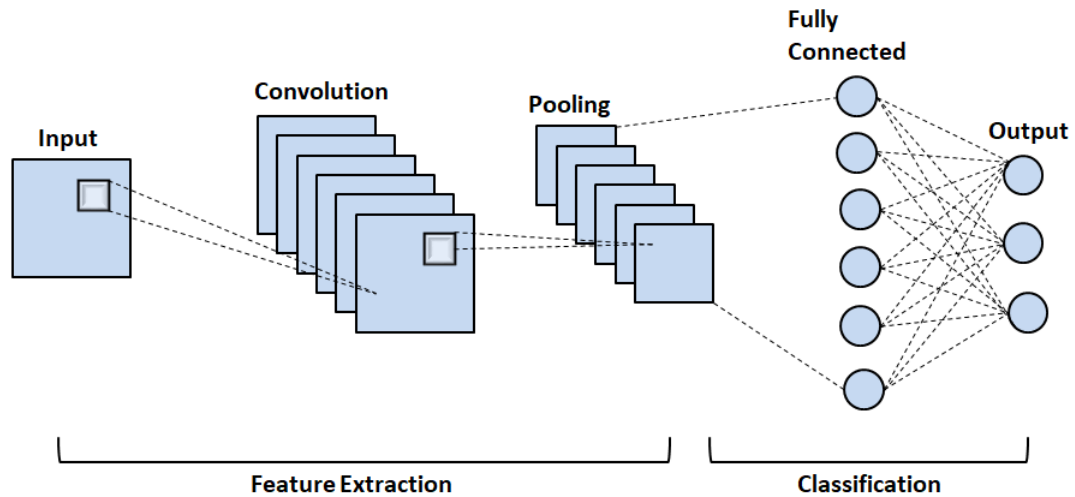


Figure 1.15: An illustration of the fundamental structure of a convolutional neural network

### 1.7.1 Convolutional Layers

The fundamental component of a convolutional network is the Conv layer, sometimes referred to as the convolutional layer, initiating the process of extracting distinctive features from the input images. Convolution is a mathematical process used in this layer to apply a filter of a particular size ( $M \times M$ ) to the input image. While computing the dot product using multiple picture regions based on their respective sizes, the filter moves throughout the input image.

This process results in creating a feature map, which includes important insights about the image, like its corners and edges. Subsequent layers receive this feature map and use it to learn different features from the input image. Three hyperparameters determine the output volume size: stride, padding, and depth. A convolutional layer has the following output size:

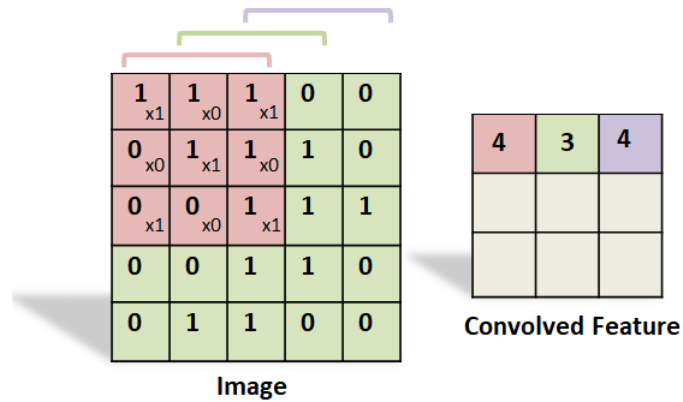


Figure 1.16: Filters move across input images, performing element-wise multiplications and summations to find patterns and enhance feature representation for deep learning models.

### 1.7.1.1 Padding

Before convolutional filters are used, a source image's border pixels can be improved using a technique called padding. It aids in maintaining spatial information and adjusting the duration map of the output feature. Padding comes in two flavours: "valid" padding (no padding) and "same" padding (padding added to maintain output size) [12]. Simply padding our input photographs with layers of zeros will eliminate the problems mentioned above. This prevents shrinkage because, if  $p$  = the number of layers of zeros applied to the image's border, our  $(x * x)$  image becomes  $(x + 2p) * (x + 2p)$  after padding as shown in Figure 1.17.

0	0	0	0	0	0
0					0
0					0
0					0
0					0
0	0	0	0	0	0

**Padding (Zero Padding)**

Figure 1.17: By placing zeros around a picture, padding (zero padding) maintains uniformity in size and speeds up computing processes while ensuring equal dimensions.

### 1.7.1.2 Depth (Number of Filters/Kernels)

A tiny matrix called a kernel, filter or feature detector is used to perform convolution on the input image. By multiplying its values by the appropriate input pixel values and summing the results, it retrieves particular features [13]. When a  $(x \times x)$  image is convolved with a  $(k \times k)$  kernel, the resultant image is typical of size  $(xk + 1) \times (xk + 1)$  and performs better due to the lesser reduction in layer dimensions.

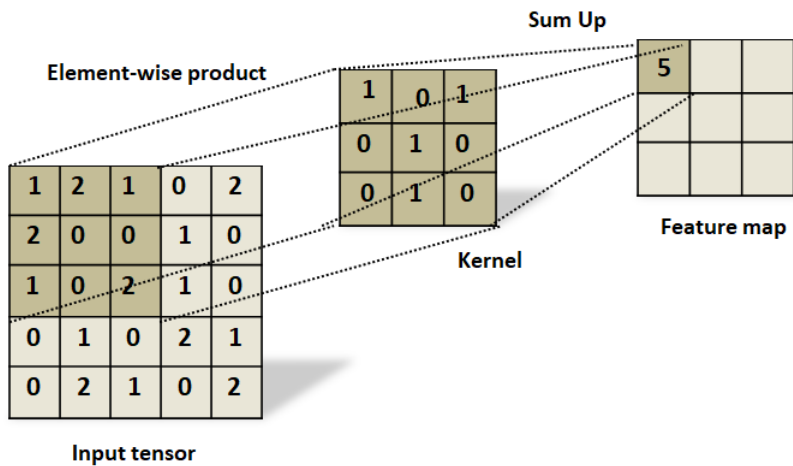


Figure 1.18: The kernel is a matrix that sequentially moves across the input data, applies the dot product operation to a portion of the data, and subsequently generates a matrix of dot product results.

### 1.7.1.3 Stride

The stride of a convolutional filter is the size of its step as it moves across the input image, and it is typically measured in terms of the input image's dimensions. Using a larger stride leads to smaller output feature maps because fewer convolutions are conducted. Conversely, smaller strides can be useful for preserving a greater amount of spatial information [64]. The stride represents the number of pixels by which the input matrix is shifted. The formula for determining the output image dimensions, considering padding ( $p$ ), filter size ( $k*k$ ), input image size ( $x*x$ ), and stride 's' is as follows:

$$((x + 2p - k + 1)/s + 1) * ((x + 2p - k + 1)/s + 1) \quad (1.3)$$

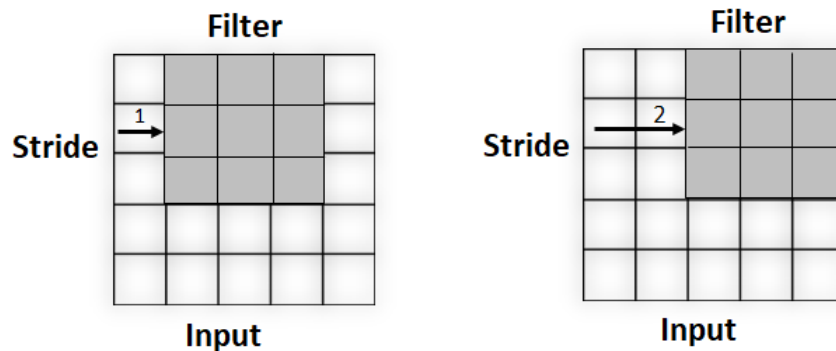


Figure 1.19: In order to preserve spatial sampling and enhance the effectiveness of feature recognition, the filters move throughout the image, pixel by pixel with a stride of one or two pixels at a time with a stride of two.

## 1.7.2 Pooling Layer

The primary purpose of the Pooling Layer, which comes after the Convolutional Layer, is to save computational expenses by decreasing the convolved feature map's size. This reduction is made possible by decoupling layers from one another and processing each feature map separately. The feature maps are downsampled using methods like max pooling and average pooling [14]. Typically, the Pooling Layer serves as the link between the Convolutional Layer and the Fully Connected Layer. Depending on the particular mechanism being used, different sorts of pooling techniques are used.

### 1.7.2.1 Max Pooling

Max pooling is a straightforward method that helps move forward with the image's most crucial attributes by taking the maximum of a region. The brighter pixels in the picture are selected using max-pooling.

### 1.7.2.2 Average Pooling

Average pooling is a technique that uses the average value for feature map patches to build a pooled feature map that has been down-sampled.

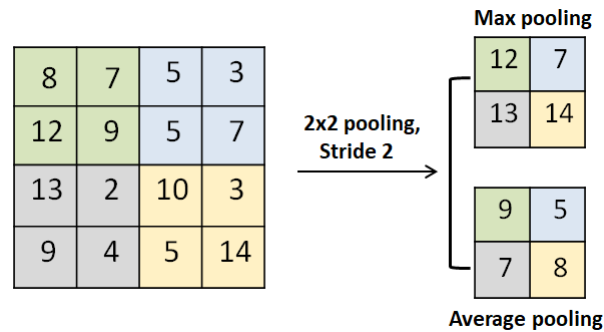


Figure 1.20: Two types of pooling: average pooling, which computes the average to summarise data while minimising spatial dimensions, and maximum pooling, which retains maximum values to capture dominating features.

### 1.7.3 Fully Connected Layer

The FC layer holds the neurons, weights, and biases and links them between two layers. In a CNN architecture, the output layer is often placed before the final few layers. Following this, there are numerous hidden layers that make it challenging to follow the data since various weights are assigned to each neuron's output. This is where all of the computation and data analysis takes place.

At this stage, the input images from the previous layers are flattened and sent to the FC layer. Following that, a few more FC levels get the flattened vector and apply conventional functional math operations to it. This initiates the classifying process.

## 1.8 Residual Neural Network (ResNet)

The Residual Network (ResNet) is the name given by He et al. (2015) to the artificial neural network (ANN). Residual Neural Networks (ResNets), a type of deep neural network architecture, introduced the concept of residual learning. They were developed to address the problem of disappearing gradients and enable the training of extraordinarily deep networks. ResNets are extensively used in many different computer vision applications and hold a significant impact on the deep learning space. It is designed to process sequential data, such as audio signals, text written in natural language, and

time series data. A few of its numerous versions are ResNet16, ResNet18, ResNet34, ResNet50, ResNet101, ResNet110, ResNet152, ResNet164, ResNet1202, and so on.

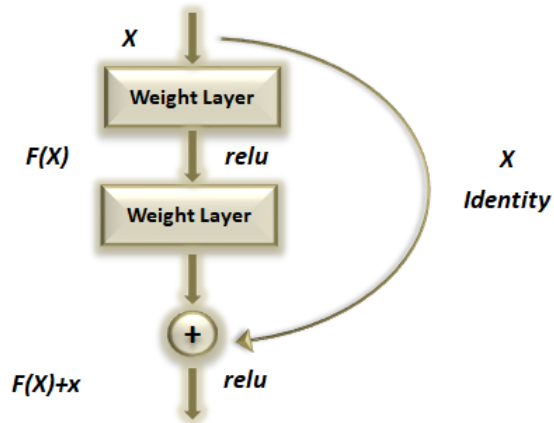


Figure 1.21: Residual unit structure diagram

The residual block is the fundamental concept of ResNets. A residual block learns the residual, or difference, between the input and output rather than the desired output directly. To do this, a "shortcut" or "skip connection" is added, which connects the input directly to the output of one or more layers. Deep networks can be trained more easily as a result of the network's ability to learn the identity function. The direct link, which avoids some triple-layer layers, is what immediately catches our attention. This link is the decoder's connection, also referred to as the core of residual blocks. The output is not pre-trained when there is no skip connection since input 'X' is multiplied by the pre-trained layer before being added to a bias term.

## 1.9 Activation Functions

Activation functions are crucial in transforming artificial neural network output into non-linear outputs because without this non linearity, the network's findings would be less accurate. For back-propagation learning to take place, the activation function needs to be differentiable. When utilizing neural networks to code fractal pictures, the nonlinearity of the activation function is crucial since the coefficients of the Iterated Function System vary depending on the different forms of fractals. The sigmoid function, which yields a positive result, is the most often selected activation function. Other functions that can have positive or negative values based on the input to the network, such as  $\tanh$  or  $\arctan$ , have a tendency to train neural networks more quickly [68]. Some of the most popular activation functions are listed below.

### 1.9.1 ReLU (Rectified Linear Unit)

In deep learning, ReLU has emerged as the most popular activation function. While all negative inputs are changed to zero, all positive inputs stay same. It is a popular activation function in deep learning. This non-linear function keeps positive inputs intact while converting all negative inputs to zero. The ReLU function can be explained as:

$$\max(0, x) = f(x) \tag{1.4}$$

ReLU activation function is a common choice for deep neural network construction since it is simple to implement and computationally efficient. It has proven to be effective across a variety of applications, having natural language processing, picture classification, and audio recognition.

### 1.9.2 Leaky ReLU

A different form of rectified linear unit (ReLU) activation function is known as leaky ReLU. It allows a slight negative slope for negative inputs, whereas the normal ReLU activation function maps all negative inputs to zero. This is the main difference. The Leaky ReLU function is defined as:

$$f(x) = \max(\alpha \cdot x, x) \tag{1.5}$$

where  $\alpha$  is a small positive constant that, when set to 0.01, sets the function's slope for negative inputs. A typical ReLU activation function may result in neurons that never activate and eventually die, but the Leaky ReLU activation function can assist reduce this issue. It is frequently used in a number of deep learning models and has demonstrated success across diverse applications including speech recognition, picture classification, and natural language processing.

### 1.9.3 Sigmoid

The sigmoid function can be used to transform any input into a probability, transferring it to the 0–1 range. The sigmoid activation function is expressed mathematically as follows:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{1.6}$$

where  $e^x$  is the exponential function. The sigmoid function works well for binary classification issues because of its continuous and smooth transition. It does, however, have some shortcomings, like the vanishing gradient issue, which can make training deep neural networks difficult.

### 1.9.4 Softmax

Multi-class classification issues are handled with the softmax activation function. From the inputs, it generates a probability distribution over several classes. The softmax function is defined as follows:

$$f(x_i) = \frac{e^{x_i}}{\sum(e^{x_j})} \quad (1.7)$$

where  $e^{x_j}$  serves as the input for class  $i$  within the function and the total is calculated across all classes  $j$ . The output total of a softmax function is guaranteed to never exceed 1, hence they can be used to depict a probability distribution across several classes. It is usually used to generate a probability distribution over several classes in the output layer of a neural network. The class with the highest probability is then determined by using the final forecast. It is commonly used in a variety of scenarios, including picture classification, speech recognition, and recommendation systems.

### 1.9.5 Tanh (Hyperbolic Tangent)

The tanh activation function accepts inputs between -1 and 1. Like the sigmoid activation function, but with several squashed values. The tanh function can be defined as follows:

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (1.8)$$

Similar to sigmoid, the Tanh activation function translates its inputs to a different range, which is advantageous in some situations. Compared to other activation functions, such the rectified linear unit (ReLU), it can be more challenging to optimise since it can induce saturation at the extremities of the range. In many applications, such as recurrent neural networks, the tanh activation function is employed extensively to describe data sequences. It is frequently employed in CNNs as well, where it is used to give each neuron's output more nonlinearity.



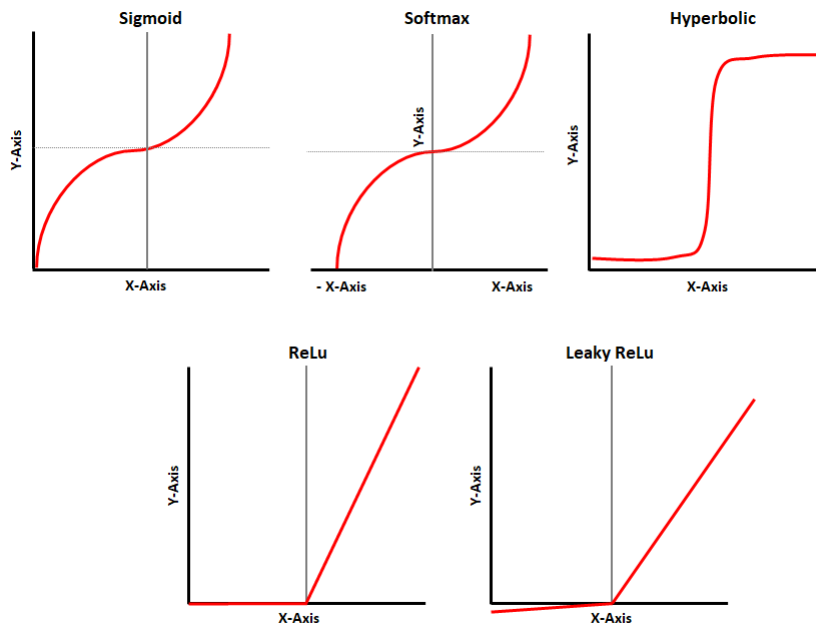


Figure 1.22: Visual depiction of activation functions

## 1.10 Problem Statement

Finding the optimal classification algorithms that can provide a reliable diagnosis.

## 1.11 Objective

The core objective of our study is to contribute towards enhancing the diagnosis of diabetic retinopathy by providing an alternative machine learning methods for early detection of Diabetic Retinopathy.

By achieving this objective, the research can contribute to the development of an accurate, efficient, and reliable method for the early identification and diagnosis of diabetic retinopathy, which can ultimately improve patient outcomes, reduce the risk of blindness in individuals with diabetes, and allow for timely treatment and management

## 1.12 Organization of the Thesis

There are five chapters in this thesis. Following the introduction of machine learning algorithms for picture detection and recognition, we go into each one's types, main

structure, and brief history. We introduce the problem definition of diabetic retinopathy classification. In the second chapter, we summarise several prior investigations on the classification of diabetic retinopathy using various machine learning methods, along with their findings. In chapter three, We elucidate our suggested approach utilising various machine learning algorithms in depth. In chapter four, We present the experimental findings along with the graphical and numerical outcomes of our models. Finally, in chapter five, We present our research's future directions and talk about the findings. The list of references is provided in the Bibliography section.

# Chapter 2

## LITERATURE REVIEW

The previous approaches to the automatic detection problem are compiled in this chapter. Previous researches have used machine learning and various models to perform automated DR screening. We conducted a literature survey to describe DR features and previous work to detect DR to develop our method and analyze the results. The most pertinent published work, which mainly illustrates and represents the suggested study in terms of applied algorithms, datasets used, number of classified stages, and overall obtained accuracy, is introduced in the following section.

On the subject of diabetic retinopathy grading, Jain et al. (2019) [15] devised a framework for detecting and evaluating the severity of diabetic retinopathy by employing various Convolutional Neural Networks (CNNs) architectures, including VGG-16, VGG-19, and Inception v3. The EyePACS dataset, which includes 35,126 images, was split into three sections: 20% reserved for validation, 60% designated for training, and the remaining 20% for classifying the five stages of Diabetic Retinopathy. The images underwent preprocessing methods like data augmentation and normalisation, which involve rotating the training images by  $90^\circ$  and  $270^\circ$ . The authors achieved accuracy rates of 71.7%, 76.9%, and 70.2% for VGG-16, VGG-19, and Inception v3, respectively. Junjun et al. (2018) [16] utilized the Residual Neural Network (ResNet) method on the EyePACS dataset, which contained approximately 35,126 images. They employed 30,000 images during the training phase and approximately 5,000 images for testing purposes. To ensure accuracy, the pictures were scaled to 256 by 256 pixels. To avoid over-fitting, the authors augmented the images of the other classes by flipping and randomly rotating them between  $0^\circ$  and  $360^\circ$  because there are a lot of pictures from class 0. As a result, they achieved a 78.4% accuracy rate for classifying the five stages of diabetic retinopathy. Li et al. (2019) [17] suggested a framework for categorizing the two phases and five distinct stages of diabetic retinopathy by utilizing Deep Convolutional Neural Networks. The researchers utilized a Kaggle dataset that had been split into three parts, with 34,124 training samples, 1,000 validation samples, and 53,572 testing

samples. As a result of their experiments, the authors achieved an 86.17% accuracy rate for the five-class classification and a 91.05% accuracy rate for the binary class. Lian et al. (2018) [18] employed three neural network frameworks (AlexNet, ResNet-50, and VGG-16) to analyze a dataset provided by EyePACS via kaggle. The dataset consisted of 35,126 retinal images that were sorted into five categories: normal, mild, moderate, severe, and proliferative, with each category accounting for 73.46%, 6.69%, 15.06%, 2.50%, and 2.02% of the dataset, respectively. The images were standardized to 256x256 pixels from their original size and overrepresented classes were resampled while underrepresented classes were randomly subsampled. In order to increase the image count and prevent bias, the authors employed spatial translation in both the left and right directions, with a one-pixel horizontal shift. In terms of accuracy, the AlexNet model achieved 73.19%, ResNet-50 achieved 76.41%, while the VGG-16 model had the highest accuracy of 79.04%.

In a separate study conducted by [19], they segmented anatomical structures like blood vessels, exudates, and Mas. They used grey level co-occurrence features to classify diabetic retinopathy (DR) images based on the segmented features. To perform this classification task, they employed the SVM classifier. Garcia et al. (2017) [20] examined the performance of various CNN models using a dataset accessible on Kaggle. They applied VGGnet16 and Alexnet to the left and right eyes in separate pictures. The dataset was preprocessed and augmented in order to improve visual contrast. Their best accuracy was 83.68% when utilising VGG16. Unfortunately, the stages of diabetic retinopathy were not precisely categorised in their studies. On a separate note, Quellec et al. (2010) [21] achieved an AUC of 0.927 by employing a standard KNN algorithm with optimal filters on two classes. Dupas et al. (2010) [22] designed a Computer-Aided Detection technique with a KNN classifier for detecting diabetic retinopathy with an 83.9% sensitivity and a 72.7% specificity. Also, [23] suggested an automated Diabetic Retinopathy detection system based on morphological parameters and attained 80.21% sensitivity and 70.66% specificity, respectively. Priyadarshi et al. [24] proposed an automated methodology based on image processing technology. Simple measurements such as the characteristics from the pictures are extracted, and the images are classified using an SVM classifier based on the training database. In their study, [25] employed soft margin SVM to classify 149 images into Moderate and Severe categories taken from the Messidor dataset. The authors extracted features and properties, including each exudated image's area, perimeter, standard deviation, and energy throughout the feature extraction procedure. They achieved an accuracy of 90.54%.

The authors [26] used MobileNet and MobileNetV2, as well as other transfer learning algorithms. They employed MobileNetV2, which is a faster and more efficient version of the MobileNetV1 architecture that also provided good performance. They

achieved 78.1% accuracy with MobileNetV2 and 58.3% accuracy with MobileNetV1. In their study, Carrera et al. (2017) [27] propose a new computer-assisted diagnosis based on the digital image processing of retinal images. The main objective of this research is to automatically classify the grade of non-proliferative diabetic retinopathy of any retinal image. In order to extract the features, the image processing stage separates the blood vessels, microaneurysms, hemorrhages, and hard exudates which can be used by the SVM classifier to evaluate the grade of each retinal image. A decision tree (DT) classifier is also implemented to compare the results of the SVM base classifier. The study by Harun et al. (2019) [28] aimed to classify two stages of diabetic retinopathy, 1,151 retinal images were utilized, with a data split of 70:30, where 806 pictures were utilised for training while 345 pictures were used for testing. The researchers classified the two classes with an overall accuracy rate of 67.47%, with 66.4% accuracy in identifying DR and 64.48% accuracy in identifying No DR. To accomplish this classification, a Multi-layer Perceptron (MLP) was used, which was trained by Binary relevance with 50 training epochs and 20 hidden layers. Mohammadian et al. (2017) [29] utilized the Inception-V3 technique and fine-tuned the parameters, as well as the Xception pre-trained models, and achieved promising outcomes. Additionally, to balance the dataset, they applied strategies for augmentation of data in order to achieve an accuracy of 87.12% employing the Inception-v3 algorithm. Pratt et al. (2016) [30] utilized 5,000 images for validation, which were then scaled to 512x512 pixels and subjected to color normalization and several modifications, such includes random shifts and flips in both the horizontal and vertical planes, and random rotations between 0 and 90 degrees. For classification, they employed a convolutional neural network and initially trained it on a subset of the dataset, which was later fine-tuned on the entire dataset using Stochastic gradient descent with Nesterov momentum. However, the authors reported limited sensitivity due to the model's inability to differentiate specific traits in the mild and moderate classes. They also discovered an image upgradability issue since over ten percent of the data was not up to UK requirements. Acharya et al. (2009) [31] developed an automated system for recognising the various stages of DR. Using a higher-order spectrum technique, features were extracted from the raw data and supplied into SVM for classification. This approach had an 82% accuracy rate. They also developed a classification method based on the areas of features such as blood vessels, exudate, haemorrhages, and so on. The SVM classifier was fed these features, which were extracted from the raw data., yielding an accuracy of 85.9% and sensitivity of 82%

# Chapter 3

## MATERIALS AND METHODS

In this study, we provide a summary of data set that was implemented in our study in addition to the approach used to include neural networks in the classification task. The methodology discusses the processes needed to create and train the neural networks, while the data set selected for this research serves as the fundamental building block for our investigations.

### 3.1 Data Acquisition

The first and most important thing needed in any machine learning project is a “dataset”. Datasets are the important part in the learning process. The dataset used in this project has been sourced from Kaggle hosted by APTOS. It originates from the Aravind Eye Hospital and comprises a substantial collection of retinal photographs captured by means of fundus photography (a technique for capturing the interior surface of the eye) under various imaging conditions. The dataset contains images with different resolutions. For training and testing, we divided the 3662 picture dataset into an 7:3 ratio. A clinician evaluated each image using a scale from 0 to 4, awarded a severity score for diabetic retinopathy, as shown in Table 3.1.

DR Grade	DR Stage	No. of Examples
0	Normal	1805
1	Mild	370
2	Moderate	999
3	Severe	193
4	PDR	295

Table 3.1: Data distribution summarizes how data points are arranged across different categories depending on the severity of the disease within a dataset.

## 3.2 Network Architecture for Image Classification

### 3.2.1 K-Nearest Neighbors (KNN)

Having a diverse sample dataset that includes images depicting various instances of diabetic retinopathy is crucial. This initial dataset is then used to create subsets for the purposes of both training and testing. Let

$$D = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (3.1)$$

where  $D$  denotes the data set of DR images. Each pair  $(x_i, y_i)$  represents an input image  $x_i$ , and its corresponding label  $y_i$ , which can be 0, 1, 2, 3, or 4 and  $n$  shows the total number of data points in the dataset. Let  $x_{\text{new}}$  be the feature vector of a new, unlabeled data point, and  $y$  be the predicted class label. The equation can be represented as:

$$y = f(x_{\text{new}}) \quad (3.2)$$

where  $f$  is the function that predicts the class label or target value based on the feature vector  $x_{\text{new}}$ .

To quantify the dissimilarity or similarity between pairs of feature vectors  $x_i$  and  $x_j$ , choose a distance metric represented as  $d(x_i, x_j)$ . In practice Euclidean distance and Manhattan distance are mostly used for this purpose. Calculate the distance between  $x_{\text{new}}$  and each data point  $x_i$  in the training dataset  $D$  by using this formula

$$\text{dist}(x_{\text{new}}, x_i) = d(x_{\text{new}}, x_i) \quad \text{for } i = 1, 2, \dots, n \quad (3.3)$$

Select the  $k$  nearest neighbors of  $x_{\text{new}}$  based on the calculated distances. These are the  $k$  data points in  $D$  with the smallest distances to  $x_{\text{new}}$ . Mathematical representation of  $k$ -nearest neighbor is presented as

$$\hat{y}_{\text{new}} = \arg \max_{c \in C} \sum_{i=1}^k I(y_i = c) \quad (3.4)$$

where  $\hat{y}_{\text{new}}$ , represents the predicted class label for new data point  $x$ ,  $y_i$  is the class label of the  $i^{\text{th}}$  neighbor,  $k$  is the number of nearest neighbors to consider, and  $I$  represents the indicator function that returns 1 if the condition inside is true and 0 otherwise.

The above equation calculates the predicted class label  $\hat{y}_{\text{new}}$  for a new data point  $x$  in  $k$ NN classification. It sums up the occurrences of each class label  $c$  among the  $k$ -nearest neighbors of  $x$ , selecting the class label with the highest count as the predicted class label.

### 3.2.2 Support Vector Machine (SVM)

A Support Vector Machine (SVM) for multiclass classification, specifically utilising the One-vs-All (OvA) technique, is mathematically described as extending the formulation of the binary SVM to incorporate multiple classes. A diverse set of sample data, including photos depicting various diabetic retinopathy cases, must be possessed. After then, subsets of this main dataset are created for testing and training. Let's denote our dataset as:

$$D = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (3.5)$$

where  $D$  denotes the data set of DR images,  $x_i$  represents the feature vector of the  $i^{\text{th}}$  sample and  $y_i$  represents its corresponding class label and  $n$  shows the total number of data points in the dataset. Here,  $y_i$  can take values from a set of  $K$  classes:  $1, 2, \dots, k$ . In the OvA strategy, for each class  $k$ , we train a binary SVM where samples from class  $k$  are labeled as positive ( $+1$ ), and samples from all other classes are labeled as negative ( $-1$ ).

Let's denote the decision function of the binary SVM for class  $k$  as  $f_k(\mathbf{x})$ . The prediction for a given input  $\mathbf{x}$  is the class  $k$  for which  $f_k(\mathbf{x})$  is maximized. The decision function  $f_k(\mathbf{x})$  can be represented as:

$$f_k(\mathbf{x}) = \mathbf{w}_k \cdot \mathbf{x} + b_k \quad (3.6)$$

where  $\mathbf{w}_k$  is the weight vector and  $b_k$  is the bias term for class  $k$ . The optimization problem for training the SVM involves finding the optimal weight vector  $w_k$  and bias term  $b_k$  for each class  $k$ . This can be expressed as follows:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w_k\|^2 \\ & \text{subject to} \quad y_i \cdot (w_k \cdot x_i + b_k) \geq 1 \quad \text{for } i = 1, 2, \dots, N \end{aligned}$$

where  $y_i$  represents the label assigned to sample  $i$ . It is set to  $+1$  if the sample  $x_i$  belongs to class  $k$ , and  $-1$  otherwise. The imposed constraints guarantee that all samples are accurately classified with a minimum margin of at least  $1$ . After all  $k$  classes SVMs have been trained, the class prediction for new input  $x$  can be determined by:

$$\arg \max_k f_k(x) \quad (3.7)$$

In other words, the predicted class is determined by selecting the class with the highest decision score out of all the binary classifiers. By training several binary classifiers, each with the goal of differentiating one class from the others, this formulation expands the capabilities of the binary SVM to handle multiclass classification.



### 3.2.3 Convolutional Neural Network (CNN)

Having a diverse sample dataset that includes images depicting various instances of diabetic retinopathy is crucial. This initial dataset is then used to create subsets for the purposes of both training and testing. Let

$$F = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (3.8)$$

where  $F$  denotes the dataset of DR images. Each pair  $(x_i, y_i)$  represents an input image  $x_i$ , and its corresponding label  $y_i$ , which can be 0, 1, 2, 3, or 4 and  $n$  shows the total number of images. The output matrix computation is described as follows:

$$F_i = f \left( \sum_{i=1}^N I_i \cdot k + B_j \right) \quad (3.9)$$

First, a kernel matrix  $K$  corresponding to each input matrix  $I_i$  is convolved. Then, each member of the resulting matrix is given a bias value  $B_j$ , and the sum of all the convoluted matrices is calculated. One output matrix  $F_i$  is generated by applying a non-linear activation function  $f$  to every component of the previous matrix.

The formula for calculating the number of output features  $n_o$  is:

$$n_o = \frac{n_i + 2p - k}{s} + 1 \quad (3.10)$$

$n_o$  displays the number of output features,  $n_i$  reveals the number of input characteristics,  $p$  shows padding size,  $s$  shows stride size and  $k$  is the number of convolution filter sizes. Analyze the effectiveness of the train model using a different testing data set after training. Let  $F_{test}$  denote the testing data set, where  $m$  is the number of test images.

$$F_{test} = (x_{test_1}, y_{test_1}), (x_{test_2}, y_{test_2}), \dots, (x_{test_m}, y_{test_m}) \quad (3.11)$$

Predict the labels for the test images using the trained model

$$\hat{y}_{test^i} = F(x_{test^i}) \quad (3.12)$$

Compare the predicted labels  $\hat{y}_{test^i}$  with the true labels  $y_{test^i}$  to compute the evaluation of the model.

The training procedure entails feeding the training images and labels into the model repeatedly. Any required hyperparameter modifications are made once the validation set performance of the model was assessed for over-fitting.

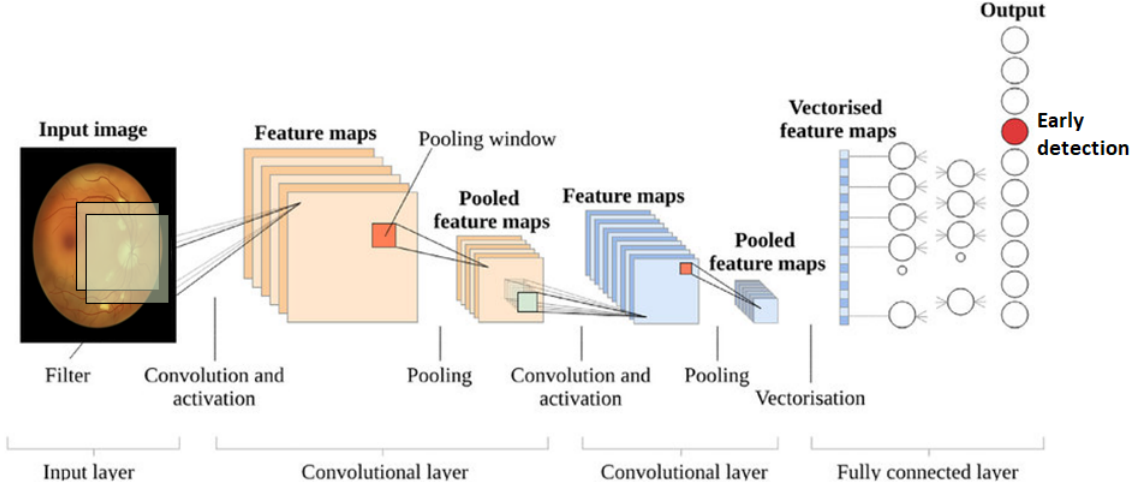


Figure 3.1: The architecture of basic CNN structure, and the various layers that make up the CNN model on the DR image data set

### 3.2.4 Residual Neural network (ResNet-50)

ResNet-50 is a popular CNN architecture known for its deep residual learning. The architecture has 50 layers, including residual blocks, convolutional layers, batch normalization techniques, and Rectified Linear Unit (ReLU) activation functions. Notably, in order to improve computational performance and make training deeper networks easier, ResNet-50 incorporates a bottleneck design into its residual blocks. To get the expected result  $\hat{y}$ , propagate an input image  $x_i$  through the ResNet-50 model. For each layer in the ResNet-50 architecture, compute the activation value  $a_i$  as follows: For a convolutional layer,

$$a_i = \text{Convolve}(x_i, W_i) + b_i \quad (3.13)$$

where  $W_i$  represents weights of the layer and  $b_i$  represents bias term. For each activation in a given channel apply batch normalisation, BN indicates the same normalisation.

$$BN_{b,c,x,y} = \gamma_c * \frac{I_{b,c,x,y} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} + \beta_c \quad (3.14)$$

In this case, BN subtracts the mean activation  $\mu_c$  from all input channel  $c$  activations, where  $\beta$  is the sum of all channel  $c$  activations for every attribute  $b$  in the whole mini-batch and for all spatial  $x, y$  positions. Then, following a similar strategy, BN divides the centered activation by the standard deviation  $\sigma_c$  (plus  $\epsilon$  for numerical stability). Testing is conducted using running mean and variance averages, followed by an affine transformation channel-wise that is parametrized by  $\gamma_c$  and  $\beta_c$ , which are discovered during training.

Apply the ReLU activation function to introduce non-linearity and enhance feature representation.

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & otherwise \end{cases} = \max(0, xi) \quad (3.15)$$

For each residual block, compute the output by passing the activation value  $a_i$  across a sequence of convolutional layers and adding the input to the output, creating a skip connection. Apply global average pooling lastly to get a compact representation of the characteristics.

$$GAP = \frac{1}{X_k} \left( \sum_{x \in X_K} xp_k \right) p_k \quad (3.16)$$

When a parameter is  $p > 0$ . The pooled feature map's contrast is increased and the image's salient features are brought into sharper focus when this exponent is set to  $p > 1$ . The expected probability distribution over the class labels can be obtained by running the pooled features through a fully connected layer with softmax activation.

$$p_i = \frac{e^{z_i}}{\sum_j^J e^{z_j}} \quad (3.17)$$

where the softmax input is  $z_i$  and the number of categories is  $j$ . A probability distribution with a range of  $(0,1)$  and a sum of  $1$  can be created from the output value of a multi-classification using the softmax function  $p_i$ . Then, the distance between the actual and desired output may be calculated using the cross-entropy loss function of Softmax, and the formula is described as:

$$L = \sum_j^J y_j p_i \quad (3.18)$$

where the real tag is represented by  $y_j$ . The loss function for the classification issue represents the discrepancy between expected and actual outcomes. Finding a set of optimal solutions in the parameter space to reduce  $L$  can be thought of as the network's training process as a parameter optimization process.

Layer	Filter Size	Resolution	Activation Function
Input	-	224x224x3	-
Conv1	7x7	112x112x64	Relu
MaxPool	3x3	56x56x64	-
Conv2	1x1	56x56x64	Relu
Conv2	3x3	56x56x64	Relu
Conv2	1x1	56x56x64	Relu
Residual	-	56x56x64	
Conv3	1x1	28x28x128	Relu
Conv3	3x3	28x28x128	Relu
Conv3	1x1	28x28x512	Relu
Residual	-	28x28x512	
Conv4	1x1	14x14x256	Relu
Conv4	3x3	14x14x256	Relu
Conv4	3x3	14x14x1024	Relu
Residual	-	14x14x1024	
Conv5	1x1	7x7x512	Relu
Conv5	3x3	7x7x512	Relu
Conv5	1x1	7x7x2048	Relu
AvgPool	-	1x1x2048	-
FC	-	1x1x1000	Softmax

Table 3.2: ResNet-50 consists of convolutional layers and residual blocks organized into four levels. Each level includes convolutional layers with diverse filter sizes, along with max-pooling and average-pooling operations for feature extraction and spatial reduction.

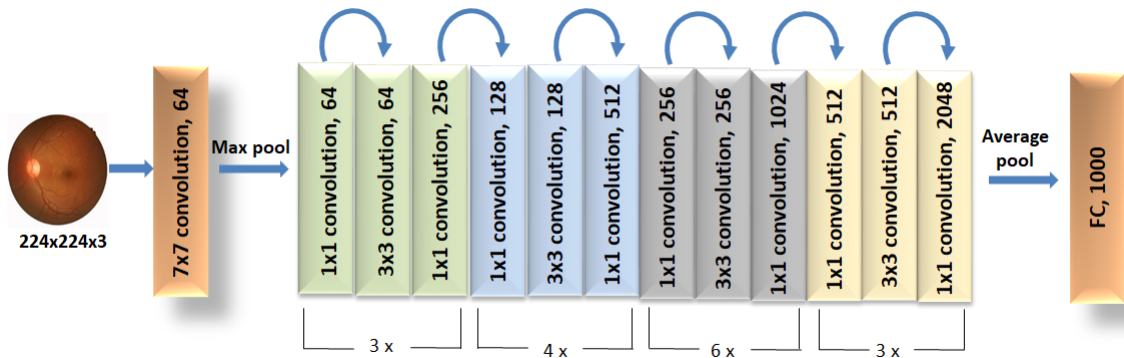


Figure 3.2: Visual architecture of ResNet-50 includes convolutional layers, followed by four levels of residual blocks.

### 3.3 Evaluation Metrics for Classification

Measurement approaches are employed to evaluate the effectiveness and performance of categorization models by comparing predicted class labels with actual class labels. The particular requirements of the classification problem and the intended trade-offs between various performance indicators determine which assessment metrics should be used. With the aid of the confusion matrix, a machine-learning concept that provides information on actual and anticipated classifications generated by a classification system, we may provide a variety of frequently used assessment metrics for classification.

#### 3.3.1 Confusion Matrix

In a confusion matrix, the classes that an item truly belongs to and the class that serves as the anticipated class are both used as indexes. Events that were improperly predicted are known as false positives (FP), events that were correctly anticipated as true negatives (TN), and events that were incorrectly projected as false negatives (FN). The following table shows the numbers with corresponding labels.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)
Total		TP + FN	FP + TN

#### 3.3.2 Accuracy

One measure used to assess a classification model's overall effectiveness is accuracy. It calculates the percentage of accurate forecasts—both true positives and true negatives—among all the forecasts made. The definition of accuracy is:

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

To put it another way, accuracy is the proportion of the model's predictions that came true, taking into account both positive and negative examples.

#### 3.3.3 Sensitivity

Sensitivity is a statistic used to assess a classification model's effectiveness, especially in binary classification issues. It is often known as recall or the true positive rate. It quantifies the percentage of real positive examples that the model accurately recognises. A definition of sensitivity is:

$$\text{Accuracy} = \frac{TP}{TP+FP}$$

To put it another way, sensitivity indicates how well the model detects positive cases. When you want to make sure that the model captures as many positive instances as feasible, and the cost of missing positive examples (false negatives) is large, this is a crucial measure.

### 3.3.4 Specificity

A metric used to assess a classification model's effectiveness, especially within the framework of binary classification, is specificity, often referred to as the true negative rate. It quantifies the proportion of actual negative cases that the model accurately detects. The definition of specificity is:

$$\text{Accuracy} = \frac{TN}{TN+FP}$$

In simple terms, specificity indicates the model's capability to detect negative situations. This is an important step to ensure the model accurately detects negative cases when there is a significant cost associated with false positives.

### 3.3.5 F1-Score

The F1-Score is a metric that's used to evaluate how well a classification model performs, especially when dealing with binary classification jobs. It provides a single statistic that balances recall and precision: the harmonic mean of these two measurements. When there is an unequal class distribution or when the costs of false positives and false negatives fluctuate, the F1-Score can be helpful in balancing the trade-off between precision and recall. The definition of an F1-Score is:

$$\text{F1-Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

where recall is the percentage of real positive cases that the model correctly identifies, and precision is the percentage of true positive forecasts among all positive predictions. The F1-Score is a numerical value ranging from 0 to 1, used to compare recall and precision merits, allowing for a more balanced approach.

### 3.3.6 Precision

A statistic called precision is used to assess a classification model's accuracy, especially when it comes to binary classification. Out of all the model's positive predictions, it determines the proportion of true forecasts. The definition of precision is:

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

In other words, accuracy shows the percentage of objects that are actually categorised as positive. This is an important statistic to ensure that the positive predictions are as precise as feasible and that the cost of false positives is as high as possible.

### 3.3.7 Error Rate

One way to measure a model or system's correctness is to look at its error rate. It shows the percentage of mistakes made relative to all predictions or operations. The error rate is computed as follows in the context of statistics and machine learning:

$$\text{Error Rate} = \frac{FP+FN}{TP+TN+FP+FN}$$

While a larger error rate denotes more frequent errors and worse performance, a lower error rate shows that the model is more accurate. Though it can be applied to many situations where accuracy and performance are evaluated, error rate is frequently employed in classification tasks.

# Chapter 4

## RESULTS AND DISCUSSION

In this section, we delve into an analysis of the metrics incorporated into this tool, examining their characteristics and aligning them with classification properties and algorithmic requirements. Drawing conclusions from this exploration guides the selection of optimal metrics tailored to specific image data and classification tasks. We evaluated several machine learning and deep learning techniques, such as K-Nearest Neighbours, Support Vector Machine, and ResNet-50, an architecture specifically designed for Convolutional Neural Networks (CNNs). These models were applied to the task of diabetic retinopathy detection using an image dataset. Throughout our experiments, all models underwent training on the designated training set and subsequent testing on an independent testing set. Our investigation primarily centered on prevalent neural network paradigm ResNet-50, showcasing the robust performance on this specific image dataset while demonstrating the generalizability across diverse domains. The ability of these models to extrapolate observed patterns from training data to unobserved images is pivotal in their proficiency to convert colored images to grayscale. This adaptability empowers the network to effectively represent intensity information inherent in RGB images through grayscale outputs.

The preprocessing pipeline applied to the dataset played a crucial role in enhancing the model's robustness and ability to generalize for detecting diabetic retinopathy. To improve the model's robustness and accurate diagnosis of diabetic retinopathy, the dataset has to be preprocessed, especially to address uneven data distribution. The model's ability to identify different severity levels of diabetic retinopathy was significantly improved by the preprocessing procedures used, as shown in Table 4.1. These included RGB channel standardization of the image size to 224x224 pixels and the application of various augmentations such as flips, rotations, and contrast modifications. Additionally, efforts were made to balance the dataset by employing techniques to address the inherent imbalance among different severity levels of diabetic retinopathy. With this approach, the model was exposed to different image orientations, lighting



conditions, and spatial transformations in an effort to diversify the dataset and mitigate the impact of class imbalance. As such, during training, the model showed better generalization and was resilient to overfitting tendencies. These preprocessing techniques, combined with measures to handle data imbalance, were effective in enabling the correct classification of diabetic retinopathy severity levels, as demonstrated by a comprehensive set of evaluation metrics, which included Accuracy, Sensitivity, Specificity, F1-score, Error Rate, Mathew’s Correlation.

<b>Parameter</b>	<b>Description</b>
Batch size	32
Image Size	224x224x3
Shift Scale Rotation	Rotate limit to 20
Augmentations	Horizontal flip, Random rotate 90 degrees, random brightness and contrast
Evaluation Metrics	Accuracy, Sensitivity, specificity, F1-score, error rate, Mathew’s correlation

Table 4.1: Overview of Image Preprocessing and Performance Metrics

To identify diabetic retinopathy, we used a multimodal strategy that included binary and multiclass classification methods. Our approach, which made use of machine learning classifiers, employed two different strategies to deal with the task’s complexity. First, to determine whether diabetic retinopathy was present in retinal pictures, we used binary classification models to separate aberrant from normal images. Subsequently, we utilised multiclass classification techniques to further classify the disease’s severity levels, enabling a more sophisticated comprehension of its evolution. Our goal was to improve the diagnostic framework’s accuracy and robustness by merging these two methods, which should help with diabetic retinopathy screening and management.

The performance of a K-Nearest Neighbors (KNN) model across three different values of K (K=25, K=30, and K=35) for binary classification tasks. Across all scenarios, the model demonstrates consistent and robust performance, with high accuracy, precision, recall, and F1 score. Specificity and sensitivity remain stable as well. The error rate and Matthews correlation coefficient also show consistent values. Although slight variations are observed in the confusion matrices, the overall pattern indicates the model's reliability in classifying instances within the dataset. Further analysis and experimentation may be needed to optimize the model for specific applications, but these results underscore the effectiveness of KNN in this context.

<b>Metrics</b>	<b>k=25</b>	<b>k=30</b>	<b>k=35</b>
Validation Set Accuracy	0.8788	0.8807	0.8807
Overall Accuracy	0.88	0.88	0.88
Precision	0.95	0.95	0.95
Recall	0.81	0.81	0.81
F1-Score	0.87	0.88	0.88
Specificity	0.95	0.95	0.95
Sensitivity	0.81	0.81	0.81
Error rate	0.12	0.12	0.12
Mathew's Correlation	0.77	0.77	0.77

Table 4.2: Comparison of Performance Metrics for Different KNN Configurations for binary classifications

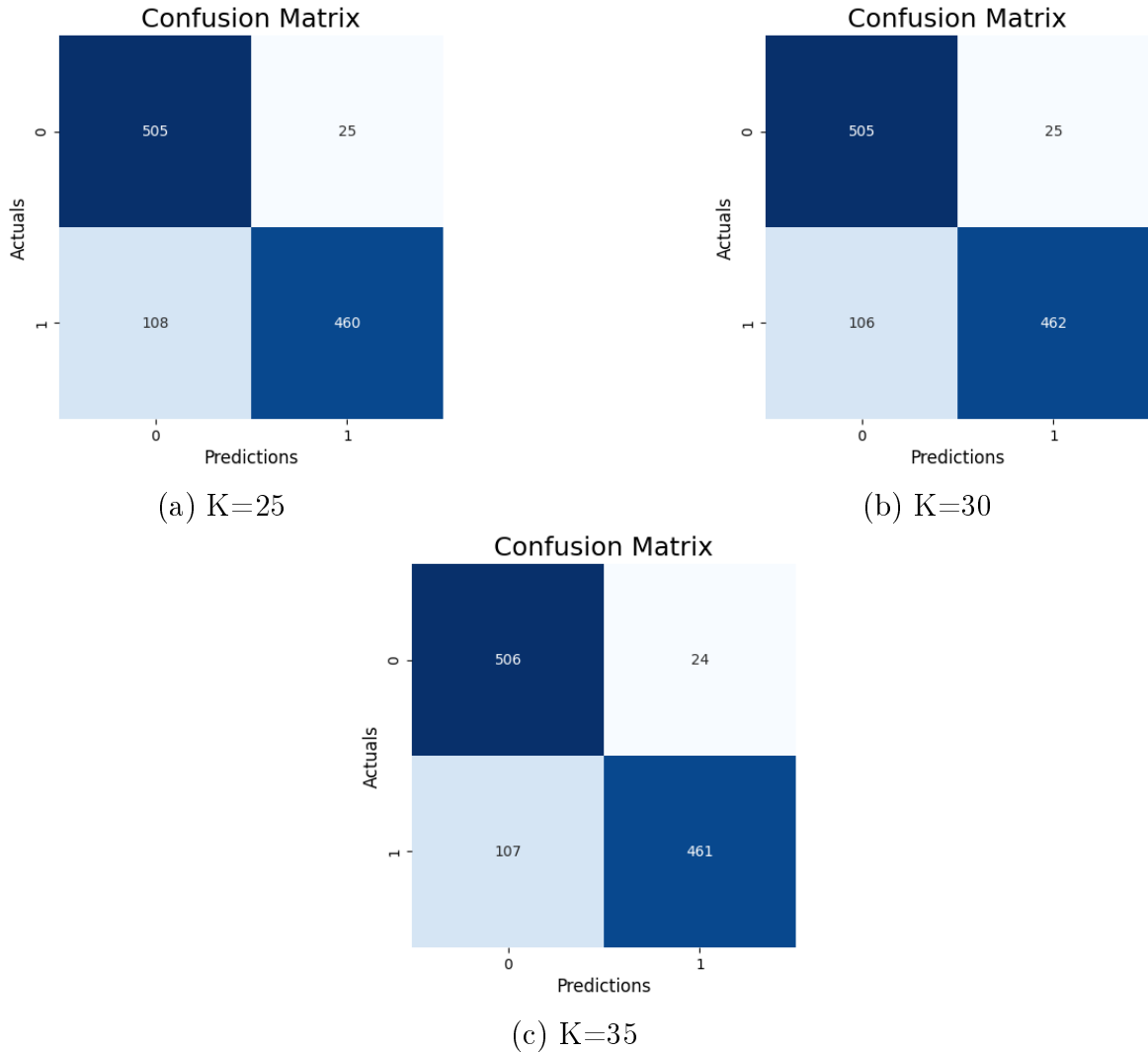


Figure 4.1: The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications.

The SVM model, applied to a binary classification task, provide valuable insights into its performance across different kernel functions. For the linear kernel, the model's accuracy of 0.92 indicates that it can classify occurrences accurately. Precision and recall metrics are also notably high, with values of 0.96 and 0.88, respectively. A balanced indicator of recall and precision, the F1 score, is calculated at 0.92. Specificity and sensitivity, which are essential for binary classification, are robust at 0.96 and 0.88, respectively. The low error rate of 0.08 suggests minimal misclassifications. Furthermore, the MCC, capturing the relationship between the projected and observed classifications, is computed at 0.84. The confusion matrix provides additional insight into the model's

performance, with 507 true negatives, 23 false positives, 68 false negatives, and 500 true positives.

For the polynomial kernel, the SVM model achieves an accuracy of 0.89. Precision remains high at 0.97, demonstrating the model's capacity to accurately determine positive instances. However, recall slightly decreases to 0.81, resulting in an F1 score of 0.88. Specificity and sensitivity remain balanced at 0.97 and 0.81, respectively. The error rate is calculated at 0.11, indicating a moderate level of misclassifications. The Matthews correlation coefficient is computed at 0.79, indicating a strong correlation between observed and predicted classifications. The confusion matrix highlights 516 true negatives, 14 false positives, 107 false negatives, and 461 true positives.

Similarly, for the radial basis function (RBF) kernel, the SVM model achieves an accuracy of 0.88. Precision remains high at 0.98, while recall decreases to 0.79, resulting in an F1 score of 0.88. Specificity and sensitivity are balanced at 0.98 and 0.79, respectively. The error rate is calculated at 0.12, indicating a relatively higher rate of misclassifications compared to the linear kernel. The Matthews correlation coefficient remains strong at 0.78. The confusion matrix illustrates 521 true negatives, 9 false positives, 119 false negatives, and 449 true positives.

The sigmoid kernel, on the other hand, performs worse, with an accuracy of 0.46. Significantly lower precision, recall, and F1 score values suggest difficulties correctly classifying cases. The sensitivity is significantly lower at 0.18 than the specificity, which is 0.76. There is insufficient agreement between the observed and predicted classifications, as indicated by the high error rate of 0.54 and the negative Matthews Correlation value of -0.07. The sigmoid kernel's shortcomings for this binary classification task are further highlighted by the confusion matrix, which shows significant misclassifications across classes.

The linear kernel outperforms the other SVM kernels that have been studied, achieving the highest accuracy and a well-balanced precision-recall trade-off. Conversely, the polynomial and RBF kernels exhibit competitive performance, each with unique benefits and drawbacks.

Performance Metrics	SVM Kernels			
	Linear	Poly	RBF	Sigmoid
Accuracy	0.92	0.89	0.88	0.46
Precision	0.96	0.97	0.98	0.45
Recall	0.88	0.81	0.79	0.18
F1-score	0.92	0.88	0.88	0.26
Specificity	0.96	0.97	0.98	0.76
Sensitivity	0.88	0.81	0.79	0.18
Error Rate	0.08	0.11	0.12	0.54
Matthew's Correlation	0.84	0.79	0.78	-0.07

Table 4.3: SVM Model having binary classification Performance Metrics for Different Kernels

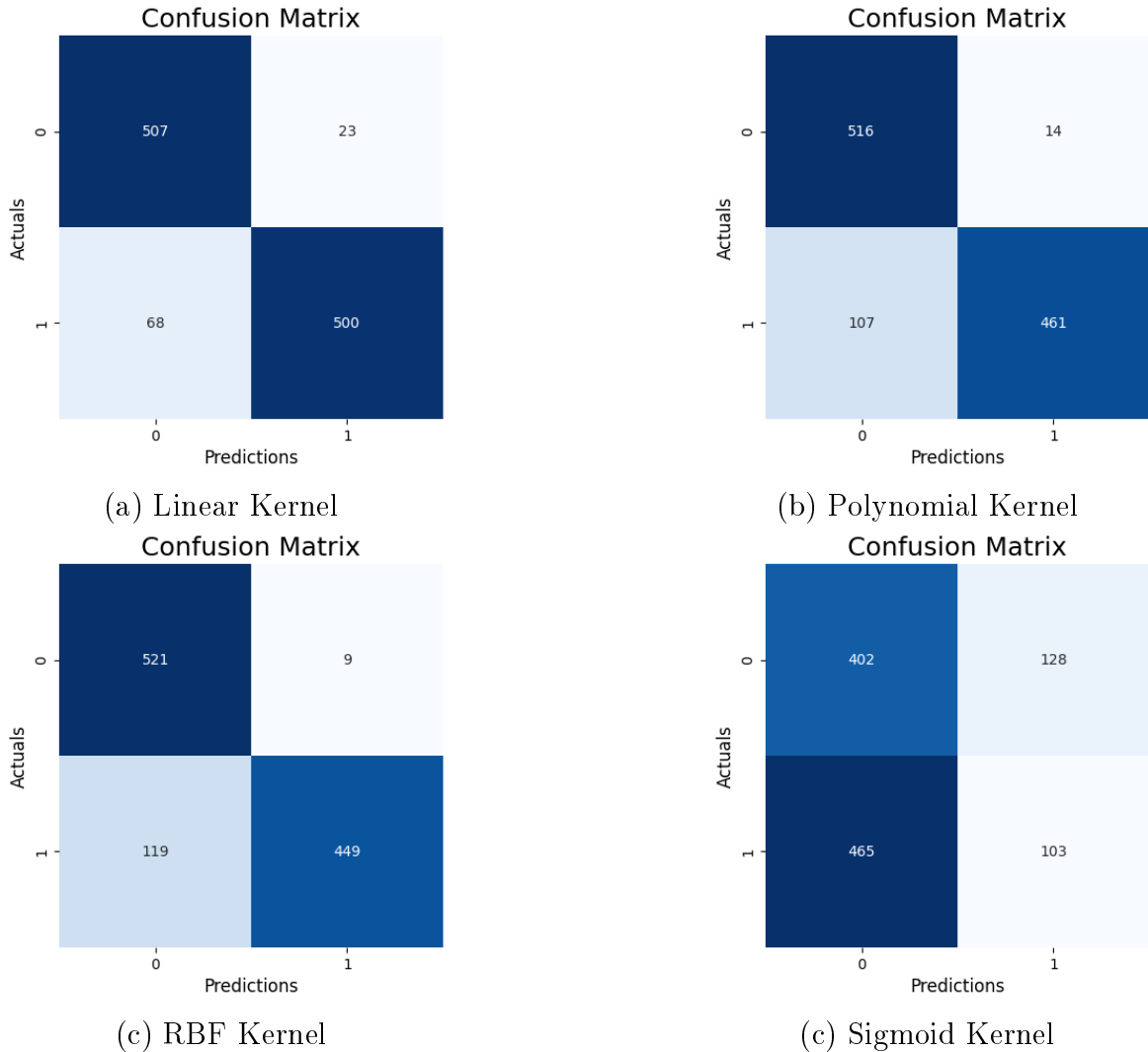


Figure 4.2: The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications.

The ResNet-50 model for binary classification demonstrate exceptional performance across various evaluation metrics. The model achieves an impressive accuracy of 0.98, demonstrating its capacity for accurate instance classification within the dataset. Precision and recall metrics also exhibit outstanding values, with precision at 0.99 and recall at 0.98. A harmonic mean of recall and precision, the F1 score, attains a high value of 0.99, reflecting the model's balanced performance in accurately recognizing both positive and negative examples. Specificity and sensitivity, crucial for binary classification tasks, demonstrate robust values of 0.98 each, indicating the model's capability to accurately identify actual negatives and actual positives, respectively. The

low error rate of 0.02 highlights the model’s minimal misclassification rate, further emphasizing its reliability in making accurate predictions. Additionally, the Matthews correlation coefficient, computed at an impressive 0.97, demonstrates a good level of concordance between the model’s predictions and the actual classifications. The confusion matrix, which includes 522 true negatives, 8 false positives, 9 false negatives, and 559 true positives, provides a thorough overview of the model’s performance. Overall, these outcomes highlight the effectiveness of the ResNet-50 model in binary classification tasks, with high accuracy, precision, recall, and F1 score, showcasing its potential for real-world applications where accurate classification is paramount.

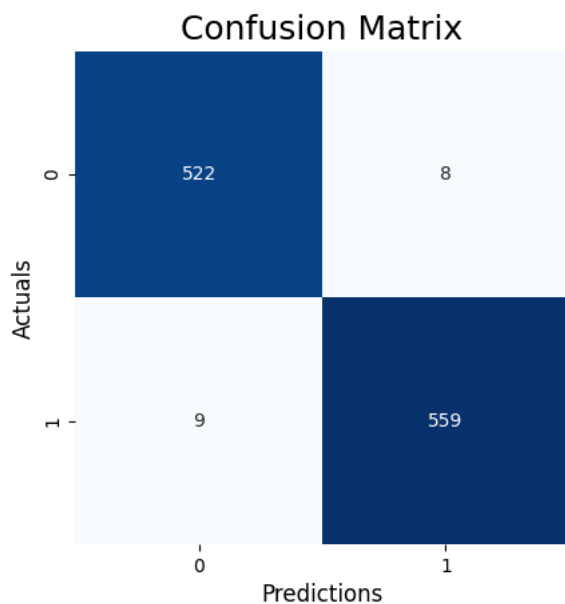


Figure 4.3: The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications.

The K-Nearest Neighbors (KNN) model undergoes evaluation through three distinct configurations. In Figure 4.4, The confusion matrices for the predicted classes are displayed, with the x-axis denotes the predicted labels and the y-axis denotes the true labels of the classes. These matrices are used to evaluate the performance of classification models on a given test dataset. From the confusion matrix, various metrics like accuracy and precision can be calculated to assess the model. The real and predicted labels span from 0 to 4 classes.

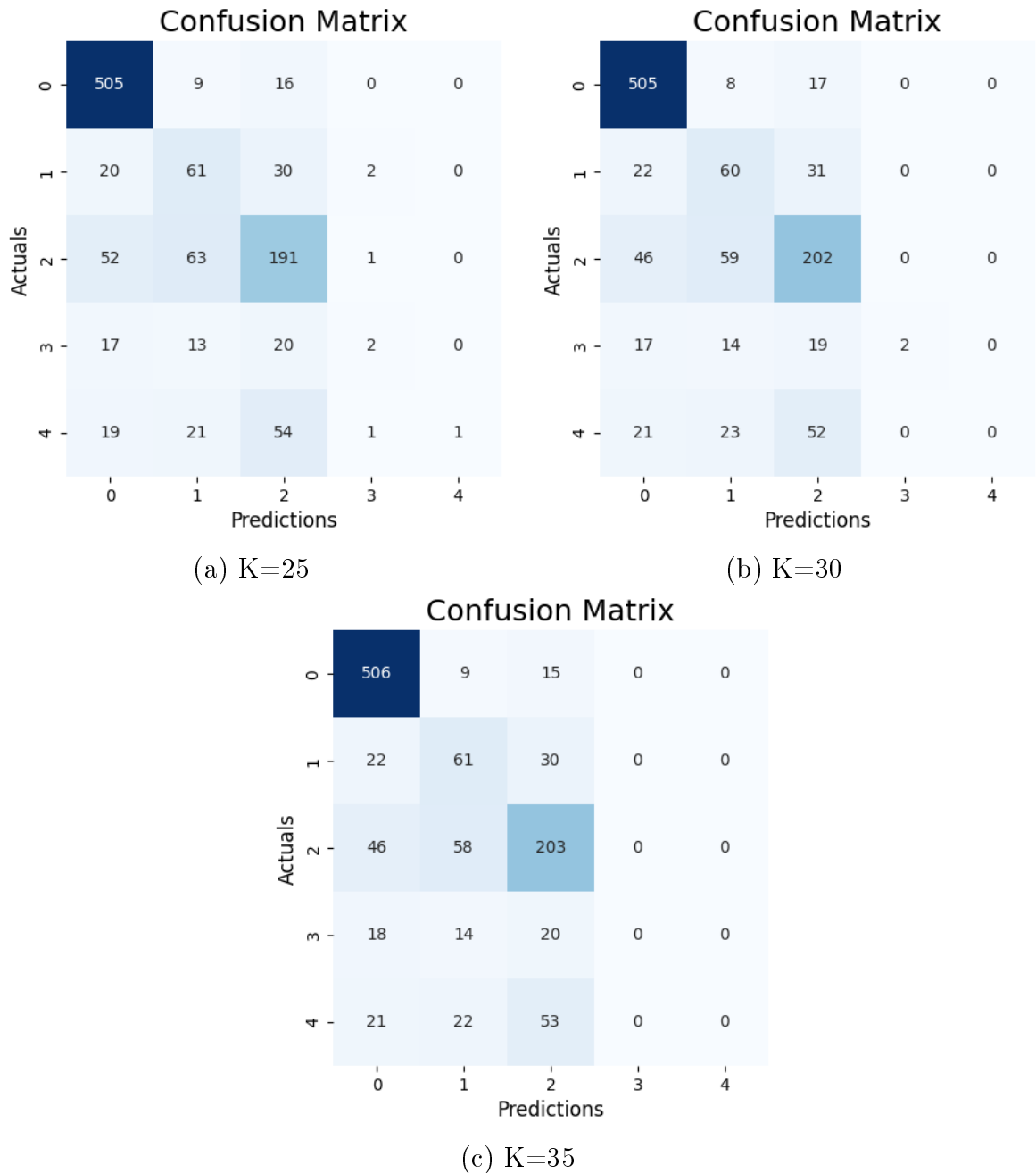


Figure 4.4: The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications.

In the first configuration, a validation set accuracy of 0.692 is achieved alongside an overall accuracy of 0.69. Precision stands at 0.71, recall at 0.69, and an F1 score



of 0.65. Specificity is high at 0.909, while sensitivity is lower at 0.433. The error rate calculates to 0.31, alongside a Matthews Correlation Coefficient of 0.52. The confusion matrix indicates variable performance across classes with varying misclassifications. Similarly, the second configuration produces a validation set accuracy of 0.700, mirroring the first closely. The accuracy remains at 0.70, with precision at 0.66 while recall maintains at 0.70. The F1 score shows a slight improvement at 0.66. Specificity and sensitivity values show comparable patterns to the first configuration, with specificity at 0.911 and sensitivity at 0.436. The error rate remains stable at 0.30, correlating again with the Matthews Correlation Coefficient of 0.54. The confusion matrix reveals a similar pattern of misclassifications to the previous configuration. The third configuration demonstrates a validation set accuracy of 0.701, in line with the other trials. The accuracy holds steady at 0.70, with precision at 0.61 and recall at 0.70. The F1 score remains consistent at 0.65. Specificity and sensitivity values closely resemble those of the prior configurations, indicating robustness in these metrics, with specificity at 0.912 and sensitivity at 0.431. The error rate remains at 0.30, in line with the Matthews Correlation Coefficient of 0.54. The confusion matrix reveals a consistent pattern of misclassifications across the classes, similar to previous evaluations.

The KNN model exhibits consistency in its performance metrics, such as accuracy, precision, recall, and F1 score, even with slight differences across various parameter values. The model's inconsistent performance in class predictions, as illustrated by the confusion matrices, points to the necessity of more research to increase the precision of classification for particular classes. Table 4.4 below provides a thorough comparison of the confusion matrices and performance indicators for each of the three configurations.

<b>Metrics</b>	<b>k=25</b>	<b>k=30</b>	<b>k=35</b>
Validation Set Accuracy	0.692	0.700	0.701
Accuracy	0.69	0.70	0.70
Precision	0.71	0.66	0.61
Recall	0.69	0.70	0.70
F1-Score	0.65	0.66	0.65
Specificity	0.909	0.911	0.912
Sensitivity	0.433	0.436	0.431
Error rate	0.31	0.30	0.30
Mathew's Correlation	0.52	0.54	0.54

Table 4.4: Comparison of Performance Metrics having multi-class classification for Different KNN Configurations

Different kernels are used to execute the Support Vector Machine model, producing

a range of performance measures. When applying the Support Vector Machine (SVM) model to a multiclass classification task, its performance varies depending on the kernel function. The model obtains an accuracy of 0.71 for the linear kernel, and its F1 score, precision, and recall are 0.68, 0.71, and 0.69, respectively. With robust values of 0.919 and 0.469 for specificity and sensitivity, respectively, the model can reliably distinguish between positive and negative examples. With a Matthews Correlation of 0.55, the computed error rate is 0.29. The confusion matrix illustrates the range of misclassifications that occur in various classes, showing the model's strong and weak points.

When switching to the 'poly' kernel, the SVM model obtains an accuracy of 0.70, with values for precision, recall, and F1 score of 0.63, 0.70, and 0.65, respectively. At 0.908 and 0.413, respectively, specificity and sensitivity are comparable to the linear kernel. The error rate stays at 0.30, which is consistent with a 0.52 Matthews Correlation Coefficient. The misclassification pattern in the confusion matrix is comparable to the linear kernel configuration.

Using the 'rbf' kernel, the SVM model also attains an accuracy of 0.70. The F1 score, recall, and precision values are consistent with polynomial kernel at 0.63, 0.70, and 0.66 respectively. The values for specificity and sensitivity, which are 0.910 and 0.444, respectively, indicate minor differences. Consistent with the 0.53 Matthews Correlation Coefficient, the error rate stays at 0.30. The confusion matrix shows a pattern of misclassifications that is similar to the linear and polynomial kernel configurations in terms of consistency across classes.

The sigmoid kernel, on the other hand, performs worse, with an accuracy of 0.42. The values of F1 score, recall, and precision are significantly lower at 0.34, 0.42, and 0.30, respectively. The sensitivity is much lower at 0.190 than the specificity, which is 0.793. With a negative Matthews Correlation Coefficient of -0.04, the error rate is high at 0.58. The sigmoid kernel's confusion matrix reveals significant misclassifications across classes.

In the SVM model for multiclass classification, the linear kernel exhibits almost higher accuracy and Matthews Correlation than the other investigated kernel functions. However, all three kernels offer competitive performance when the proper parameters are adjusted, suggesting that SVM is a good choice for workloads involving multiclass classification.



Figure 4.5: Confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications.

Performance Metrics	SVM Kernels			
	Linear	Poly	RBF	Sigmoid
Accuracy	0.71	0.70	0.70	0.42
Precision	0.68	0.63	0.63	0.30
Recall	0.71	0.70	0.70	0.42
F1-score	0.69	0.65	0.66	0.34
Specificity	0.919	0.908	0.910	0.793
Sensitivity	0.469	0.413	0.444	0.190
Error Rate	0.29	0.30	0.30	0.58
Matthew's Correlation	0.55	0.52	0.53	-0.04

Table 4.5: SVM Model having multi-class classification Performance Metrics for Different Kernels

The performance evaluation of the ResNet-50 model in multiclass classification reveals significant insights into its effectiveness. With an overall accuracy of 0.82, the model demonstrates a balanced performance across classes, as reflected in consistent precision, recall, and F1 score metrics, each at 0.82. Specificity is notably high at 0.953, indicating the model's proficiency in identifying negative instances, while sensitivity stands at 0.626, showcasing its ability to identify positive instances. The error rate of 0.18 suggests relatively few misclassifications. Furthermore, the Matthews Correlation coefficient, computed at 0.73, indicates a strong correlation between observed and predicted classifications. A thorough breakdown is provided by the confusion matrix of the model's performance across different classes, illustrating both its strengths in accurate classification and areas where improvements may be needed.

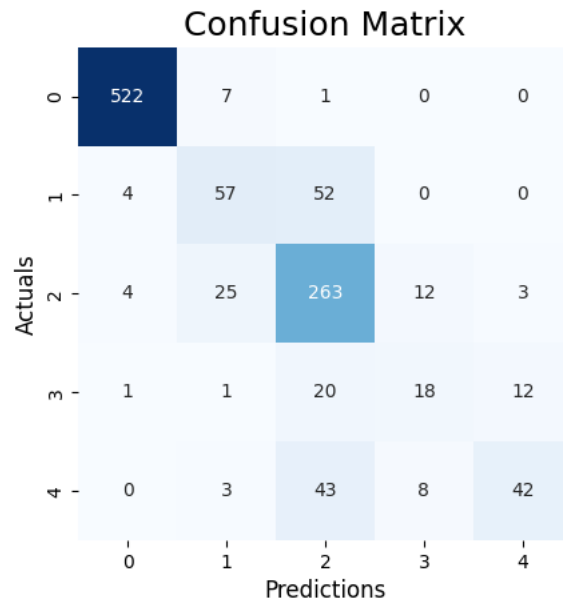


Figure 4.6: The confusion matrix shows correct and incorrect classifications made by the classifier, where diagonal elements represents right predictions and non-diagonal elements denotes misclassifications.

# Chapter 5

## CONCLUSION

To sum up, this research examined the performance of KNN, SVM, and ResNet-50 in binary and multiclass classification tasks, revealing notable differences among the models. ResNet-50 consistently outperformed KNN and SVM across various evaluation metrics due to its deep design, which enables it to extract complex features from images, making it highly effective for both binary and multiclass classification. However, each model presents trade-offs, such as computational complexity and generalization capability. Despite ResNet-50's impressive results, including a 79% accuracy compared to 70% for KNN and 72% for SVM, future research should address scalability issues and explore hybrid approaches to leverage the strengths of different algorithms. This study provides valuable insights into the comparative performance of machine learning and deep learning models in image classification, particularly in distinguishing between Normal, Mild, Moderate, Severe, and Proliferative DR phases, and paves the way for advancements in medical imaging, object recognition, and other domains requiring accurate classification.

# Bibliography

- [1] J. J. Li and X. Tong, “Statistical hypothesis testing versus machine learning binary classification: Distinctions and guidelines,” *Patterns*, vol. 1, no. 7, 2020.
- [2] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” *arXiv preprint arXiv:2008.05756*, 2020.
- [3] Z. B. Kizilkan, M. S. Sivri, I. Yazici, and O. F. Beyca, “Neural networks and deep learning,” in *Business Analytics for Professionals*, pp. 127–151, Springer, 2022.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [6] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [7] J. Frankle and M. Carbin, “The lottery ticket hypothesis: Finding sparse, trainable neural networks,” *arXiv preprint arXiv:1803.03635*, 2018.
- [8] A. Adeel, M. Gogate, and A. Hussain, “Contextual deep learning-based audio-visual switching for speech enhancement in real-world environments,” *Information Fusion*, vol. 59, pp. 163–170, 2020.
- [9] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.
- [10] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions,” *Journal of big Data*, vol. 8, pp. 1–74, 2021.

- [11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, pp. 1–6, Ieee, 2017.
- [12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [14] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 international joint conference on neural networks*, pp. 2809–2813, IEEE, 2011.
- [15] A. Jain, A. Jalui, J. Jasani, Y. Lahoti, and R. Karani, "Deep learning for detection and severity classification of diabetic retinopathy," in *2019 1st International Conference on Innovations in Information and Communication Technology (ICICT)*, pp. 1–6, IEEE, 2019.
- [16] P. Junjun, Y. Zhifan, S. Dong, and Q. Hong, "Diabetic retinopathy detection based on deep convolutional neural networks for localization of discriminative regions," in *2018 International Conference on Virtual Reality and Visualization (ICVRV)*, pp. 46–52, IEEE, 2018.
- [17] Y.-H. Li, N.-N. Yeh, S.-J. Chen, Y.-C. Chung, *et al.*, "Computer-assisted diagnosis for diabetic retinopathy based on fundus images using deep convolutional neural network," *Mobile Information Systems*, vol. 2019, 2019.
- [18] C. Lian, Y. Liang, R. Kang, and Y. Xiang, "Deep convolutional neural networks for diabetic retinopathy classification," in *Proceedings of the 2nd International Conference on Advances in Image Processing*, pp. 68–72, 2018.
- [19] D. Selvathi, N. Prakash, and N. Balagopal, "Automated detection of diabetic retinopathy for early diagnosis using feature extraction and support vector machine," *Int J Emerg Technol Adv Eng*, vol. 2, no. 11, pp. 762–767, 2012.
- [20] G. García, J. Gallardo, A. Mauricio, J. López, and C. Del Carpio, "Detection of diabetic retinopathy based on a convolutional neural network using retinal fundus images," in *Artificial Neural Networks and Machine Learning–ICANN 2017: 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11-14, 2017, Proceedings, Part II 26*, pp. 635–642, Springer, 2017.



- [21] G. Quellec, S. R. Russell, and M. D. Abràmoff, “Optimal filter framework for automated, instantaneous detection of lesions in retinal images,” *IEEE Transactions on medical imaging*, vol. 30, no. 2, pp. 523–533, 2010.
- [22] B. Dupas, T. Walter, A. Erginay, R. Ordonez, N. Deb-Joardar, P. Gain, J.-C. Klein, and P. Massin, “Evaluation of automated fundus photograph analysis algorithms for detecting microaneurysms, haemorrhages and exudates, and of a computer-assisted diagnostic system for grading diabetic retinopathy,” *Diabetes & metabolism*, vol. 36, no. 3, pp. 213–220, 2010.
- [23] C. Sinthanayothin, V. Kongbunkiat, S. Phoojaruenchanachai, and A. Singalavanija, “Automated screening system for diabetic retinopathy,” in *3rd International Symposium on Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the*, vol. 2, pp. 915–920, IEEE, 2003.
- [24] S. Priyadarshi, A. Kumar, and M. Tech, “A review on diabetic retinopathy and its early detection techniques,”
- [25] H. Tjandrasa, R. E. Putra, A. Y. Wijaya, and I. Ariesianti, “Classification of non-proliferative diabetic retinopathy based on hard exudates using soft margin svm,” in *2013 IEEE International Conference on Control System, Computing and Engineering*, pp. 376–380, IEEE, 2013.
- [26] R. Sarki, S. Michalska, K. Ahmed, H. Wang, and Y. Zhang, “Convolutional neural networks for mild diabetic retinopathy detection: an experimental study,” *bioRxiv*, p. 763136, 2019.
- [27] E. V. Carrera, A. González, and R. Carrera, “Automated detection of diabetic retinopathy using svm,” in *2017 IEEE XXIV international conference on electronics, electrical engineering and computing (INTERCON)*, pp. 1–4, IEEE, 2017.
- [28] N. H. Harun, Y. Yusof, F. Hassan, and Z. Embong, “Classification of fundus images for diabetic retinopathy using artificial neural network,” in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 498–501, IEEE, 2019.
- [29] S. Mohammadian, A. Karsaz, and Y. M. Roshan, “Comparative study of fine-tuning of pre-trained convolutional neural networks for diabetic retinopathy screening,” in *2017 24th National and 2nd International Iranian Conference on Biomedical Engineering (ICBME)*, pp. 1–6, IEEE, 2017.
- [30] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, “Convolutional neural networks for diabetic retinopathy,” *Procedia computer science*, vol. 90, pp. 200–205, 2016.

- [31] U. R. Acharya, C. M. Lim, E. Y. K. Ng, C. Chee, and T. Tamura, “Computer-based detection of diabetes retinopathy stages using digital fundus images,” *Proceedings of the institution of mechanical engineers, part H: journal of engineering in medicine*, vol. 223, no. 5, pp. 545–553, 2009.