
Comparative Performance Analysis of Classical vs Quantum Computing for Diabetic Retinopathy Detection

Bachelor Thesis

By

Amit Mandhana (CSE/22014/868)

Chirag Shukla (CSE/22038/892)

Sayan Mandal (ECE/22150/1004)

Prithwish Dey (CSE/22065/919)



A Bachelor thesis submitted to

Indian Institute of Information Technology Kalyani

for the partial fulfillment of the degree of

Bachelor of Technology
in
Computer Science and Engineering

November, 2024



INDIAN INSTITUTE OF INFORMATION TECHNOLOGY KALYANI

An Institute of National Importance

(Autonomous Institution under MoE, Govt. of India &

Department of Information Technology & Electronics, Govt. of West Bengal)

WEBEL IT Park, 14, Adivasi Para, Opposite of Kalyani Water Treatment Plant

Near Buddha Park, Dist. Nadia, P.O. Kalyani - 741235, West Bengal.

Email: office@iiitkalyani.ac.in, Website: <https://www.iiitkalyani.ac.in>

CERTIFICATE

This is to certify that the thesis entitled "**Comparative Performance Analysis of Classical vs Quantum Computing for Diabetic Retinopathy Detection**" being submitted by **Amit Mandhana (CSE/22014/868), Chirag Shukla (CSE/22038/892), Sayan Mandal (ECE/22150/1004), Prithwish Dey (CSE/22065/919)**, undergraduate students in the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, India, pursuing **Bachelor of Technology in Computer Science and Engineering**, is an original research work carried by them under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulation of **IIIT Kalyani** and, in my opinion, has reached the standards needed for submission. The work, techniques, and results presented have not been submitted to any other university or Institute for the award of any other degree or diploma.

Dr. Imon Mukherjee, Ph.D

Assistant Professor

Department of Computer Science and Engineering
Indian Institute of Information Technology Kalyani
Kalyani, Nadia, West Bengal 741235, India.

Acknowledgments

We would like to begin by expressing our deepest gratitude to the Almighty for His countless blessings, guidance, and strength that have enabled us to successfully complete this thesis.

Our heartfelt thanks go to **Dr. Imon Mukherjee**, our esteemed supervisor, for his constant support, expert advice, and encouragement throughout this research, titled "**Comparative Performance Analysis of Classical vs Quantum Computing for Diabetic Retinopathy Detection**". His unwavering dedication to our progress and his insightful suggestions have been pivotal in enriching our learning experience and ensuring the successful completion of this work.

We are profoundly grateful to **the Department of Computer Science and Engineering, IIIT Kalyani, India**, for providing us with the opportunity and resources to undertake this project. The guidance and infrastructure offered by the institution have played a crucial role in shaping this research.

Additionally, we extend our sincere appreciation to our friends, family members, and the faculty and staff of IIIT Kalyani for their continuous encouragement and assistance during this journey. Lastly, we acknowledge and thank all individuals whose efforts and support, whether direct or indirect, have contributed to the completion of this thesis. Your contributions have been invaluable to this achievement.

Amit Mandhana (CSE/22014/868)

Chirag Shukla (CSE/22038/892)

Sayan Mandal (ECE/22150/1004)

Prithwish Dey (CSE/22065/919)

Department of Computer Science and Engineering
Indian Institute of Information Technology Kalyani
Kalyani, Nadia, West Bengal 741235, India.

Abstract

Diabetic Retinopathy (DR) is a leading cause of blindness in adults due to formation of lesions on the retina. The manual diagnosing for DR from the retinal fundas images by ophthalmologists is time-intensive and costly also. While Classical Machine learning (CML) models are highly used for recognition of DR but it is limited by the maintenance costs. Then, Quantum Transfer Learning (QTL) is a better solution to handle this challenge because it operates in a optimized way and on heuristic principles. This report presents a comparative analysis of Classical Machine Learning and emerging Quantum Transfer Learning approaches for detecting and classifying DR on basis of performance, accuracy and computational efficiency. Our methodology begins with an image processing-based pipeline comprising resolution enhancement via bicubic interpolation, green channel extraction, Contrast Limited Adaptive Histogram Equalization (CLAHE) and optic disc removal to enhance lesion visibility and reduce false positives. We explore unsupervised Autoencoders (AEs) for anomaly detection and propose a hybrid AE+classifier model to improve DR discrimination. Feature extraction for Classical Transfer Learning leverages pretrained models (ResNet18, ResNet152, EfficientNetV2B0), while the Hybrid QTL framework integrates classical extractors with variational quantum classifiers to exploit quantum advantages. APTOS 2019 Blindness Detection (Kaggle) Dataset is used for training purposes. This study provides insights into strengths and limitations of each approach with different potential ways to integrate quantum and classical techniques for enhancing DR recognition.

Keywords: Diabetic Retinopathy, Classical ML, CLAHE, Autoencoder, Quantum Transfer Learning, Accuracy, Hybrid Quantum Model.

Contents

List of Figures	vii
List of Tables	ix
1 Introduction	1
2 Literature Survey	3
2.1 Related Works	3
2.2 Analysis of Different Models	4
3 Data Preprocessing	5
3.1 Dataset Overview	5
3.2 Image Resizing and Normalization	7
3.3 Image Division & Load Image Labels	7
3.4 Data Splitting	8
4 Classical Machine Learning Approaches	9
4.1 Computer Vision-Based Approach	9
4.1.1 Abnormalities in Diabetic Retinopathy	9
4.1.2 Image Acquisition and Pre-processing	10
4.1.3 Green Channel Extraction	12
4.1.4 Contrast Enhancement using CLAHE	13
4.1.5 Gaussian Filtering	16
4.1.6 Morphological Operations	16
4.1.7 Optic Disc Removal	17
4.1.8 Blood Vessel Detection	18
4.1.9 Lesion Detection	19
4.1.10 Lesion Coverage Quantification	20
4.2 Autoencoder-Based Approach	21
4.2.1 Autoencoder-Only Model	21

4.2.1.1	Architecture and Training	21
4.2.1.2	Reconstruction and Anomaly Detection	22
4.2.2	Hybrid Autoencoder-Classifier Model	25
4.2.2.1	Architecture and Training	25
4.2.2.2	Latent Space Analysis	27
4.2.2.3	Calibration and Thresholding	27
4.3	Classical Transfer Learning Approach	28
4.3.1	Model Training Approaches	29
4.3.1.1	Custom CNN	29
4.3.1.2	ResNet152	30
4.3.1.3	EfficientNetV2B0	31
5	Quantum Machine Learning Approaches	33
5.1	Swap Test Approach	33
5.1.1	Flattening The Image Vector	33
5.1.2	Quantum Inspired Feature Mapping(Encoding)	33
5.1.2.1	Qubit Initialization	33
5.1.2.2	Angle Encoding	33
5.1.2.3	Discrete Quantum Work(DTQW)	34
5.1.3	State Vector Preparation	35
5.1.4	The Swap Test Algorithm	35
5.1.4.1	Initialize the System	35
5.1.4.2	Apply Hadamard Gate	35
5.1.4.3	Apply Controlled-Swap (CSWAP)	35
5.1.4.4	Final Hadamard Gate	36
5.1.4.5	Measurement of Ancillary Qubit	36
5.2	Hybrid Quantum Transfer Learning Approach	39
5.2.1	Overview of Hybrid Model	39
5.2.2	Classical Feature Extraction	39
5.2.3	Quantum Classifier (Variational Quantum Circuit)	39
5.2.3.1	Input Encoding (Angle Embedding)	40
5.2.3.2	Variational Layers	40
5.2.3.3	Measurement	40
5.2.4	Hybrid Model Flow	41
5.2.5	Training Procedure	41
5.2.6	Quantum Classifier Properties	41

6	Results and Performance Analysis	43
6.1	Performance Metrics	43
6.2	Results	44
6.3	Performance Comparison	49
7	Conclusion and Future work	51
7.1	Conclusion	51
7.2	Future Work	52
	Bibliography	53

List of Figures

1.1	The Different Types of DR Stages	1
1.2	The Fundoscopic Image of Defected Retina	2
3.1	Dataset Details	5
3.2	Normal Retinal Images	6
3.3	Mild DR Retinal Images	6
3.4	Moderate DR Retinal Images	6
3.5	Severe DR Retinal Images	7
3.6	Proliferative DR Retinal Images	7
3.7	Dataset overview after splitting	8
4.1	Clinical abnormalities for Diabetic Retinopathy	9
4.2	Progression stages of DR	10
4.3	Heights above the ground	11
4.4	Original Image vs Upsampled Image of Fundus Images.	12
4.5	Green channel extraction and CLAHE . (a) Grayscale image. (b) CLAHE on grayscale images. (c) Extracted green channel. (d) CLAHE on extracted green channel.	12
4.6	Workflow and mathematical basis of CLAHE	14
4.7	14
4.8	15
4.9	15
4.10	16
4.11	16
4.12	17
4.13	17
4.14	17
4.15	17
4.16	18
4.17	(a)is the original image and (b)is the image after removing optic disc	18
4.18	18

4.19	19
4.20	19
4.21	19
4.22	19
4.23	20
4.24	20
4.25	Architecture of the baseline convolutional autoencoder	21
4.26	Architecture of the baseline autoencoder-only model	22
4.27	Reconstruction error distribution for the AE model	23
4.28	Best normal image reconstruction	24
4.29	Worst normal reconstruction	24
4.30	Best DR (diseased) reconstruction	24
4.31	Worst DR reconstruction	25
4.32	Generalized Transfer Learning Architecture	28
4.33	Generalized CNN Architecture	29
4.34	Block diagram of the ResNet152 system	30
4.35	Block diagram of the EfficientNetV2B0-based system	32
5.1	Quantum Encoding and DTQW	34
5.2	Quantum Swap Test Circuit	37
6.1	Classwise Accuracy	45
6.2	Results of Computer Vision-Based Approach	45
6.3	Results of AE Model	46
6.4	Results of Hybrid (AE+Classifier) Model	46
6.5	Results of Computer Vision-Based Approach	47
6.6	Results of Fine-Tuned ResNet152 Model	48
6.7	Results of Fine-Tuned EfficientNetV2B0 Model	48
6.8	Results of QTL Approach	49

List of Tables

2.1	Analysis of different survey papers	4
4.1	DR Feature Encoder Architecture Details	26
5.1	Summary of Quantum Classifier Properties	41
6.1	Performance Comparison of Different Techniques	50

List of Algorithms

1	Quantum Swap Test Algorithm	38
2	Hybrid Quantum Transfer Learning	42

List of Acronyms

CLAHE Contrast Limited Adaptive Histogram Equalization

AE AutoEncoder

CNN Convolutional Neural Network

ReLU Rectified Linear Unit

ANN Artificial Neural Network

QML Quantum Machine Learning

DTQW Discrete Quantum Walk

C-Swap Controlled Swap

CZ Controlled Z

AQ Ancillary Qubit

QTL Quantum Transfer Learning

Chapter 1

Introduction

Diabetes is a situation characterized by inadequate insulin production, leading to elevated blood sugar levels [Wang and Lo(2018)]. This condition can affect various organs, including the kidneys, nerves, heart, and retina . Diabetic Retinopathy (DR) is a complication of diabetes that involves swelling and leakage of blood and fluid from the retina's blood vessels [Radi et al.(2021)]. In 2020, DR was estimated to affect about 103.12 million adults globally, with projections suggesting this number could rise to 160.50 million by 2045. The global prevalence of DR among individuals with diabetes is approximately 22.27%, with different stages like proliferative DR (6.96%) and diabetic macular edema (6.81%) . DR amounts to 2.6% of causes for blindness globally [4]. Early diagnosis of DR allows for effective management through available treatments. The two main types of DR are proliferative and non-proliferative DR [Radi et al.(2021)].

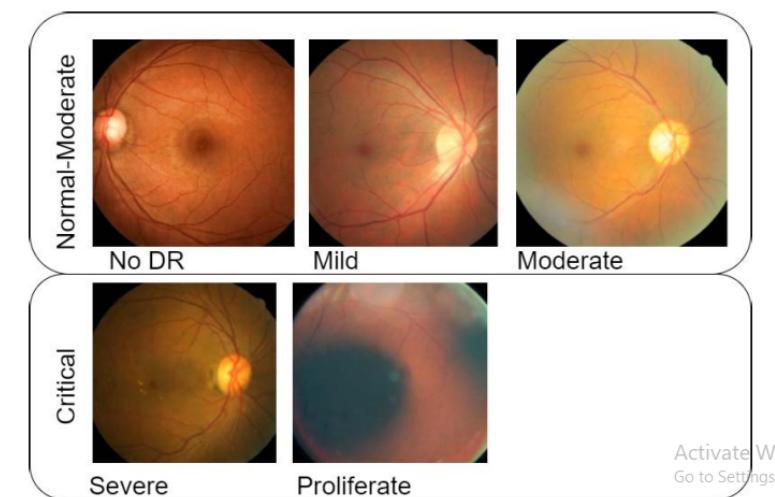


Figure 1.1: The Different Types of DR Stages

Recent advances in artificial intelligence (AI) have enabled the successful diagnosis of various medical conditions, including retinal diseases like DR [Mane et al.(2022)]. The manual diagnosis of DR through retinal images is labor-intensive and requires significant expertise [Mane et al.(2022)].

Therefore, developing an automated DR detection system to support medical professionals is advantageous [Alyoubi et al.(2020)]. Deep learning, particularly through Convolutional Neural Networks (CNN), has been employed to automate DR image classification, assisting ophthalmologists in early disease detection.

The emergence of small-scale quantum computers has spurred interest in quantum deep learning techniques. Researchers have explored classification models based on different quantum circuits [Mir et al.(2022)], where classical data is encoded as qubits. Quantum computers have shown greater reliability in certain applications, especially in sampling complex probabilities, compared to classical computers. Quantum-based techniques implemented on classical computers can reveal effective solutions, making them applicable to image classification tasks [Jain et al.(2024)].

Despite efforts to classify DR images using deep learning for early detection, most research has concentrated on DR detection rather than the identification of its various stages (Figure 1.1). Additionally, recognizing and locating different DR lesions (Figure 1.2) presents challenges but is crucial [Islam et al.(2020)]. The primary contributions of this study include:

- First detecting the images are eye-images.
- Pre-processing involves resizing images to ensure flexibility for further analysis.
- The different Classical Machine Learning Techniques are applied on the pre-processed images.
- The quantum-based Hybrid Model enhances DR classification accuracy through an optimized multiple-qubit gate.
- The model's effectiveness is evaluated using performance metrics such as recall, accuracy, specificity, precision, and f1-score.

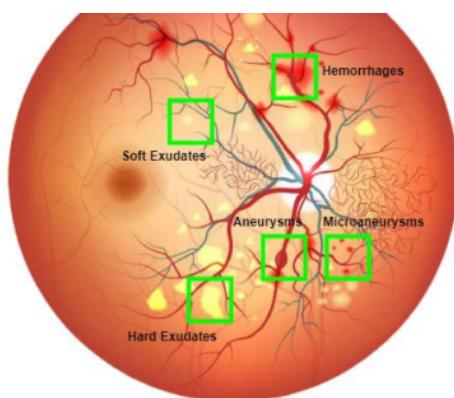


Figure 1.2: The Fundoscopic Image of Defected Retina

Chapter 2

Literature Survey

2.1 Related Works

Transfer learning involves using a model built for one task on another task, speeding up training and improving performance [[Gangwar and Ravi\(2021\)](#)]. It's especially useful for deep neural networks with small datasets. In medical image analysis, transfer learning with convolutional neural networks (CNNs) helps overcome data scarcity and hardware limitations by using knowledge from previous tasks.

Quantum machine learning(QML) expands on traditional machine learning by integrating quantum and classical components in hybrid neural networks [[Jain et al.\(2024\)](#)]. In these networks, a quantum layer is often appended to a network that has been pre-trained using classical methods.

Recent studies have emphasized the importance of image preprocessing as the foundation for effective DR detection. Akter et al. (2014) [[Akter et al.\(2014\)](#)] utilized morphological operations for detecting exudates in fundus images, employing histogram equalization, binary thresholding, and watershed transformation. Dutta et al. (2017) [[Dutta et al.\(2017\)](#)] processed images from Messidor, DRIVE, and local datasets using contrast enhancement, top-hat transforms, and intensity thresholding to identify red lesions and classify the disease into different severity levels.

Kayal et al. (2014) [[Kayal et al.\(2014\)](#)] implemented median filtering, image subtraction, dynamic thresholding, and image addition to detect hard exudates. Harini and Sheela (2016) [[Harini and Sheela\(2016\)](#)] focused on detecting microaneurysms and exudates using resizing, CLAHE, and top-hat filtering, with additional vessel removal for improved microaneurysm identification.

Sinthanayothin et al. (2002) [[Sinthanayothin et al.\(2002\)](#)] analyzed a small dataset of 30 retinal images, converting them to IHS color space, applying contrast enhancement, and identifying the optic disc based on intensity variation. They detected blood vessels using neural networks and applied recursive region growing and operators for lesion segmentation.

Feature extraction methods have been extensively applied for precise DR diagnosis. Mirajkar and Patil (2013) [[Mirajkar and Patil\(2013\)](#)] used grayscale conversion, adaptive histogram equalization,

and smoothing, followed by Kirsch edge detection and 2D Gabor wavelet transforms to extract vessel features.

Harini and Sheela (2016) [[Harini and Sheela\(2016\)](#)] also extracted textural features using the Gray Level Co-occurrence Matrix (GLCM). Ravivarma et al. (2014) [[Ravivarma et al.\(2014\)](#)] processed low-quality fundus images using median filtering and fuzzy c-means clustering for segmentation, followed by extraction of color, texture, and size features optimized via Particle Swarm Optimization (PSO) and classified with Support Vector Machines (SVM).

Gurudath et al. (2014) [[Gurudath et al.\(2014\)](#)] developed a classification framework distinguishing no DR, non-proliferative, and proliferative DR using Gaussian filters and automatic mask generation. Extracted textural features were fed into Artificial Neural Networks (ANNs) and SVMs. Pal et al. (2017) [[Palavalasa and Sambaturu\(2018\)](#)] evaluated Naive Bayes, SVM, k-Nearest Neighbors, and Decision Trees on the Messidor dataset, finding that preprocessing and feature extraction significantly impacted model performance.

2.2 Analysis of Different Models

These studies([TABLE 2.1](#)) highlight transfer learning's specially Quantum Transfer Learning importance in DR detection, focusing on model performance. Accuracy depends on dataset quality. Advances in quantum computing, like Google's quantum supremacy, are compared to the shift from machine learning to deep learning, offering significant computational power.

Author	Models	Results
S.Mir [Mir et al.(2022)]	Inception V3 and Different Quantum Classifier	In the classical model, accuracy rate 85%, but Quantum Hybrid Model accuracy rate 93-96%
T.Shahwar [Shahwar et al.(2022)]	ResNet34 and Quantum Variational circuit	This hybrid model gained the precision of 99.1%
S.Wan. [Wan et al.(2018)]	AlexNet, VggNet-s, VggNet-16, VggNet-19, GoogleNet and ResNet	Accuracy of VggNet-s model was 95.68% after doing transfer learning and hyperparameter tuning
W.M.Gondal [Gondal et al.(2017)]	A CNN model	Sensitivity was 93.6% and specificity was 97.6% on DiaretDB1 for Binary Classification
G.Garcia [García et al.(2017)]	AlexNet,VggNet	VggNet-16 gave the better result with 93.65% specificity

Table 2.1: Analysis of different survey papers

Chapter 3

Data Preprocessing

This work describes the data preprocessing steps involved in preparing medical images for a Diabetic Retinopathy detection system. It outlines the crucial stages of the preprocessing pipeline, including image standardization, data augmentation, and data partitioning.

3.1 Dataset Overview

The images are assumed to be organized into 5 folders indicating the severity of diabetic retinopathy based on the Area Calculation. These grades are integer labels from 0 to 4, representing increasing severity of DR:

- 0 - No DR
- 1 - Mild DR
- 2 - Moderate DR

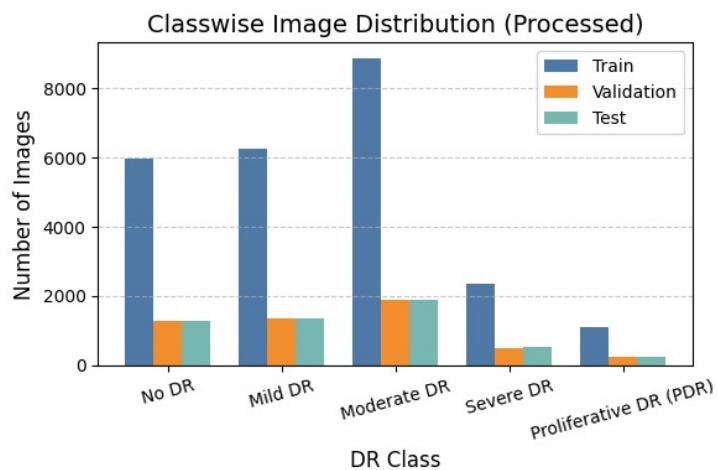


Figure 3.1: Dataset Details

- 3 - Severe DR
- 4 - Proliferative DR

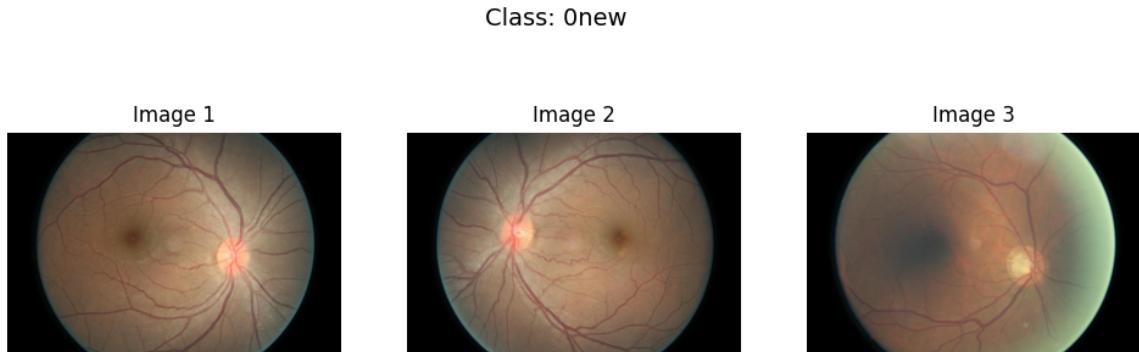


Figure 3.2: Normal Retinal Images

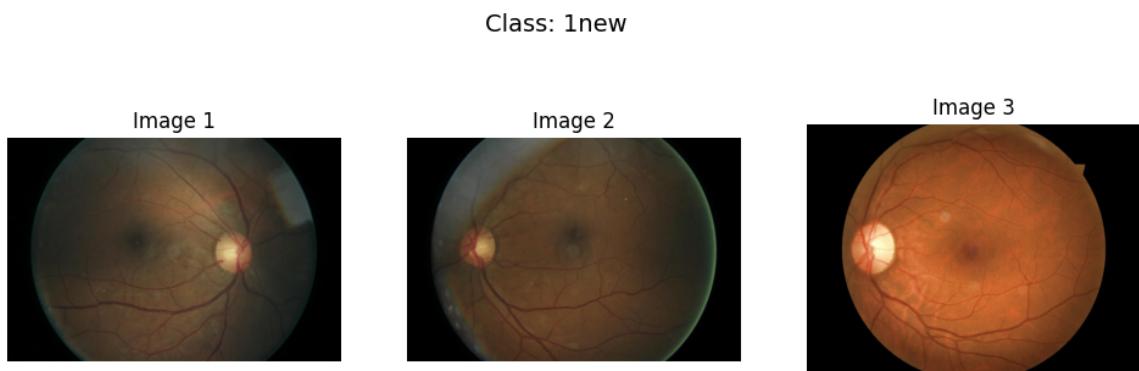


Figure 3.3: Mild DR Retinal Images

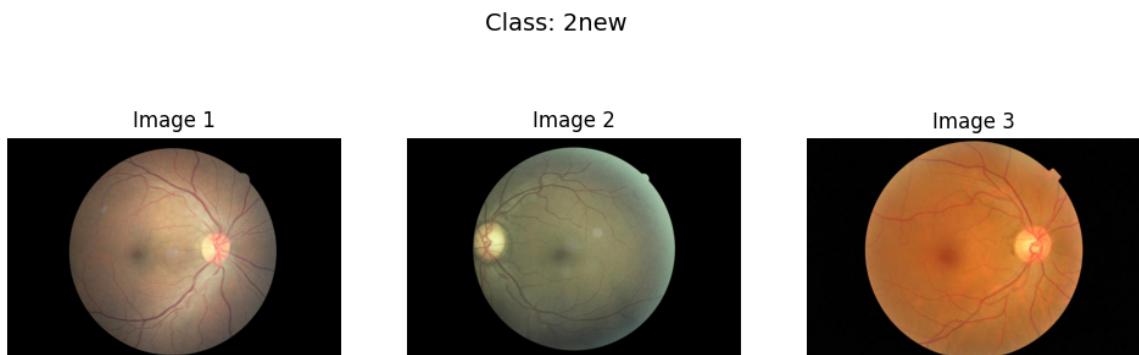


Figure 3.4: Moderate DR Retinal Images

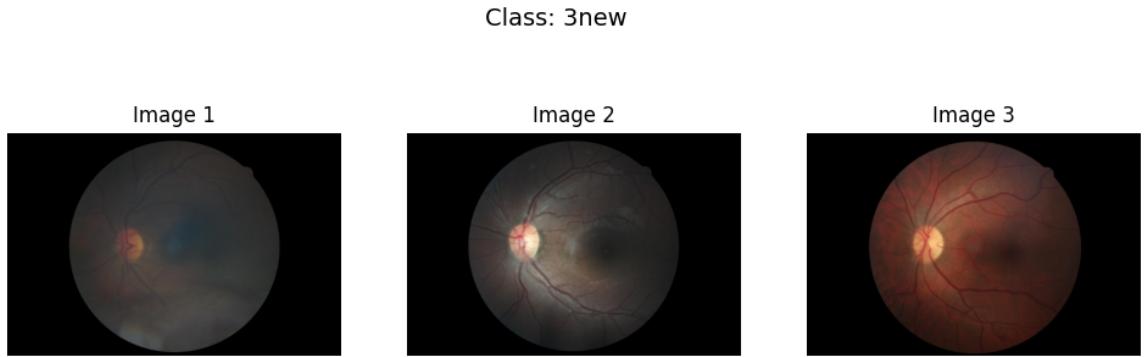


Figure 3.5: Severe DR Retinal Images



Figure 3.6: Proliferative DR Retinal Images

3.2 Image Resizing and Normalization

All images are resized to a standard dimension of $224 \times 224 \times 3$ pixels. This ensures consistency and compatibility with popular CNN architectures. Normalization scales pixel intensities to the $[0, 1]$ range:

$$I_{\text{norm}}(x, y) = \frac{I(x, y)}{255} \quad (3.1)$$

3.3 Image Division & Load Image Labels

Mathematically, the dataset can be represented as a set of image-label pairs:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \quad (3.2)$$

Where x_i is the i^{th} input image and $y_i \in \{0, 1, 2, 3, 4\}$ is the corresponding DR label.

3.4 Data Splitting

The dataset is split into training, validation, and testing sets in a 70:15:15 ratio. If the total number of samples is N , the number of samples in each split is given by:

$$\begin{aligned}N_{\text{train}} &= 0.7 \times N \\N_{\text{test}} &= 0.15 \times N \\N_{\text{val}} &= 0.15 \times N\end{aligned}\tag{3.3}$$

Dataset Summary:

Class Index	Class Label	Original Count	Train	Validation	Test	Total	Processed
0	No DR	8526	5968	1278	1280		8526
1	Mild DR	8929	6250	1339	1340		8929
2	Moderate DR	12687	8880	1903	1904		12687
3	Severe DR	3386	2370	507	509		3386
4	Proliferative DR (PDR)	1587	1110	238	239		1587

Figure 3.7: Dataset overview after splitting

+

Chapter 4

Classical Machine Learning Approaches

4.1 Computer Vision-Based Approach

4.1.1 Abnormalities in Diabetic Retinopathy

Diabetic Retinopathy (DR) is characterized by specific pathological changes in the retina caused by chronic hyperglycemia. The following key abnormalities are clinically observed and serve as markers for DR progression:

DIABETIC RETINOPATHY

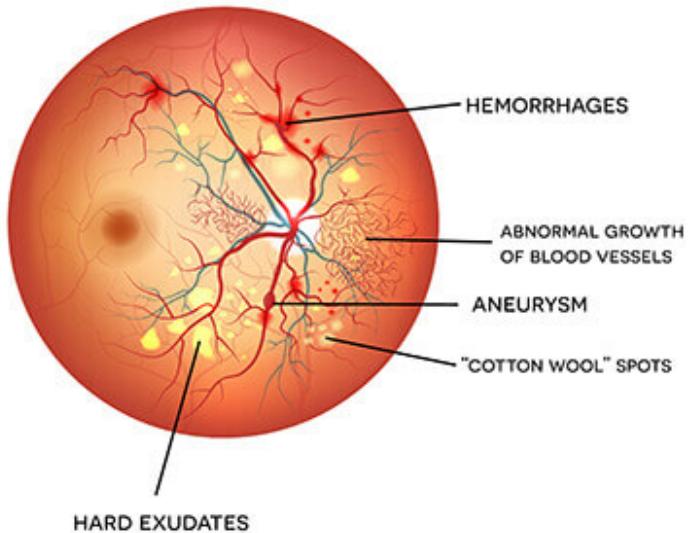


Figure 4.1: Clinical abnormalities for Diabetic Retinopathy

- **Microaneurysms:** Small outpouchings of retinal capillaries, often the earliest detectable lesions.

- **Hemorrhages:** Leakage of blood from damaged vessels; may appear as dot-blot or flame-shaped.
- **Hard Exudates:** Lipid and protein deposits resulting from vascular leakage, appearing as yellow spots.
- **Cotton Wool Spots:** Represent areas of nerve fiber layer infarction, appearing as fluffy white patches.
- **Neovascularization:** Abnormal growth of new, fragile blood vessels, typically indicating proliferative DR.

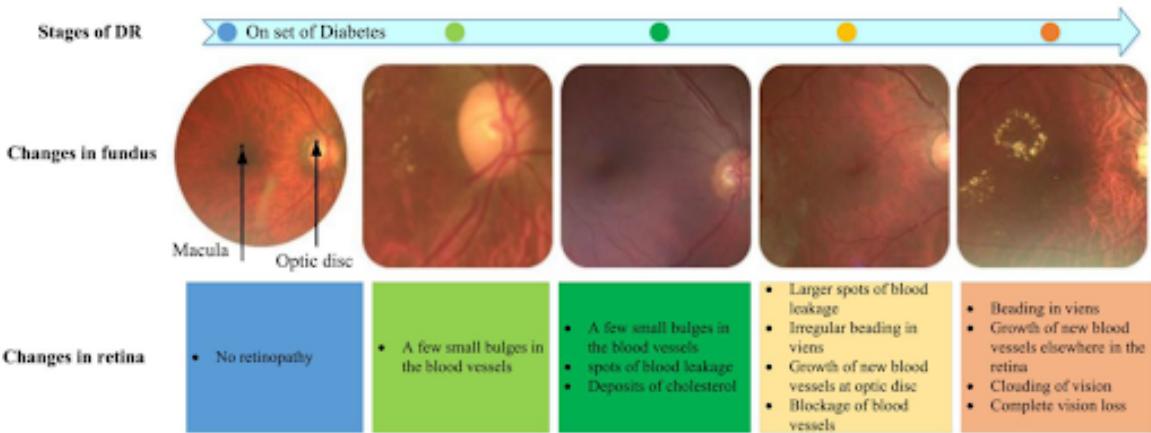


Figure 4.2: Progression stages of DR

4.1.2 Image Acquisition and Pre-processing

The proposed work utilizes a comprehensive pipeline of image processing techniques to detect retinal abnormalities indicative of Diabetic Retinopathy (DR) from fundus images. This section elaborates on the sequential steps involved in pre-processing, feature enhancement, and abnormality detection.

Fundus images were acquired from publicly available datasets such as DRIVE, DIARETDB, and local clinical sources. The initial stage involves verifying image quality to ensure diagnostic reliability.

- **Resolution Check and Enhancement:** Low-resolution images may obscure subtle lesions such as microaneurysms or exudates. An image is flagged as low-resolution if either dimension is below 500 pixels:
- **Bicubic Interpolation Technique:** To enhance low-resolution images, bicubic interpolation is applied using the following formulation:

Bicubic interpolation method is more complicated than bilinear interpolation. In bicubic interpolation sixteen nearest neighbour of a pixel have been considered as shown in above fig. The intensity value assigned to point (x,y) is obtained using the equation,

$$I(x, y) = \sum_{i=-1}^2 \sum_{j=-1}^2 a_{ij} \cdot P(x + i, y + j)$$

where a_{ij} are bicubic coefficients and P is the pixel value. Comparison of Bicubic interpolation with some 1- and 2-dimensional interpolations.

Black and red/yellow/green/blue dots correspond to the interpolated point and neighbouring samples, respectively.

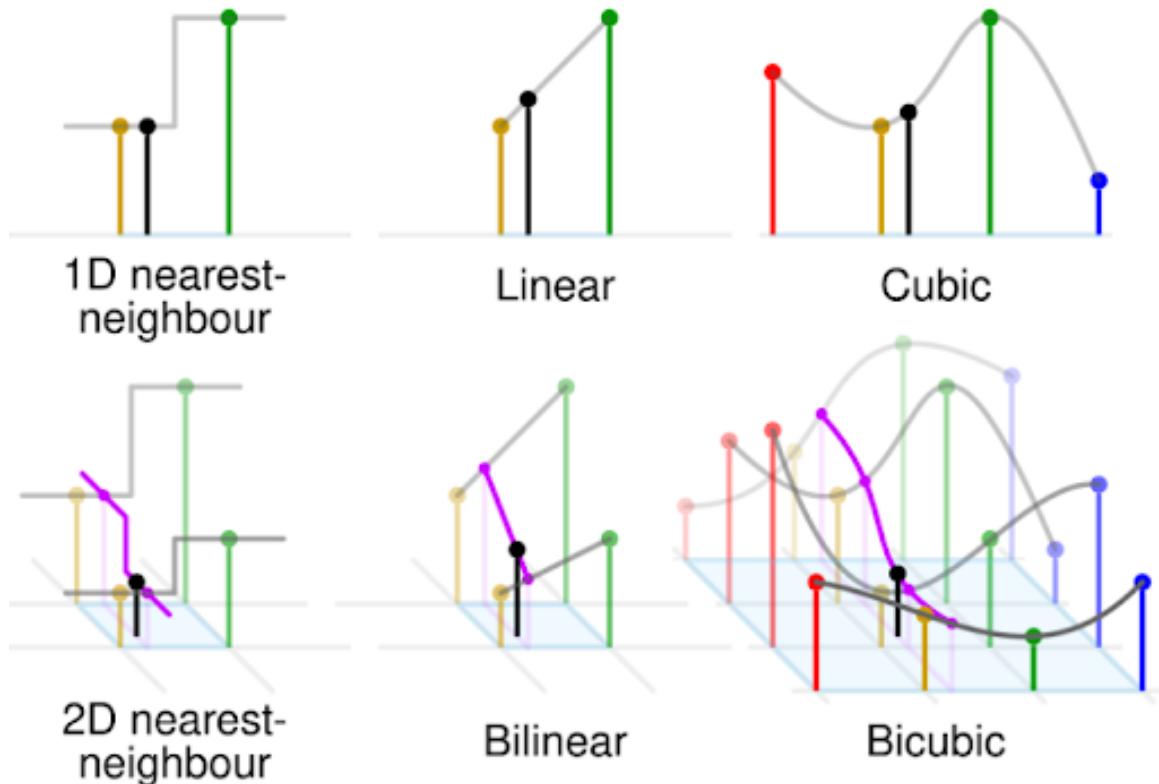


Figure 4.3: Heights above the ground

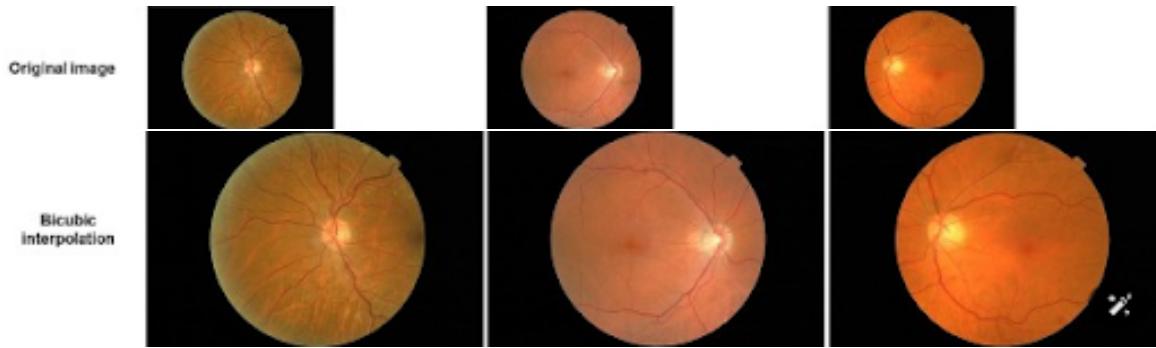


Figure 4.4: Original Image vs Upsampled Image of Fundus Images.

4.1.3 Green Channel Extraction

Retinal fundus images are RGB-colored; however, for lesion enhancement, the green channel is extracted. This is due to the hemoglobin in blood vessels absorbing green light effectively, providing maximum contrast between vessels and the background.

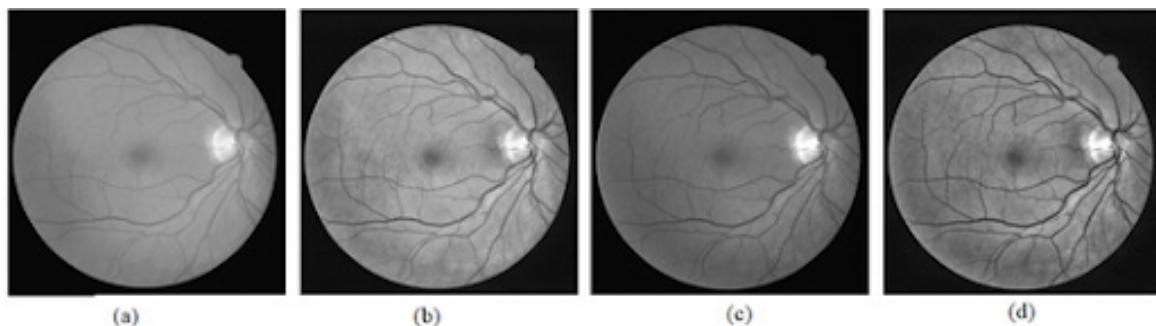


Figure 4.5: Green channel extraction and CLAHE . (a) Grayscale image. (b) CLAHE on grayscale images. (c) Extracted green channel. (d) CLAHE on extracted green channel.

The Green Channel is Best for Blood Vessel & Lesion Detection in Diabetic Retinopathy. This is due to the unique absorption and reflection properties of light in retinal imaging:

A. Hemoglobin Absorbs Green Light Efficiently

- Blood vessels in the retina contain hemoglobin, which strongly absorbs green (middle-wavelength) light.
- This results in higher contrast between blood vessels and the surrounding retinal background.
- Exudates (fatty deposits from blood leakage) and hemorrhages (bleeding spots) also appear more distinct in the green channel.

B. Red and Blue Channels Have Limitations

Red Channel:

- Penetrates deeper into tissues, causing vessels to blend with the background.
- Blood vessels do not appear as sharp since red light is less absorbed by hemoglobin.

Blue Channel:

- Highly scattered and absorbed by the retina.
- Causes more noise and does not effectively differentiate between vessels and background.

C. Green Channel Mimics Human Visual Sensitivity

- The human eye is most sensitive to green light (around 550 nm).
- Green-channel images provide natural contrast.
- Most fundus image analysis techniques use green-channel extraction to enhance visibility of lesions, vessels, and microaneurysms.

Practical Benefit in DR Detection

- Extracting the **green channel** enhances blood vessels, microaneurysms, hemorrhages, and exudates.
- This improves **segmentation** and **feature extraction** for automated Diabetic Retinopathy (DR) detection.
- Thus, using the green channel is a **standard preprocessing step** in medical image analysis, especially for fundus images in DR detection.

Mathematically,

$$I_g(x, y) = I(x, y, 2)$$

where I_g is the extracted green channel and $I(x, y, 2)$ represents the green component of the RGB image at position (x, y) .

4.1.4 Contrast Enhancement using CLAHE

Contrast enhancement is an essential step in retinal image preprocessing, particularly for the detection of microaneurysms and exudates that may be indistinguishable in poorly illuminated regions. Traditional histogram equalization fails to adapt to local intensity variations, often over-amplifying noise. Therefore, Contrast Limited Adaptive Histogram Equalization (CLAHE) is employed.

CLAHE works by dividing the image into small contextual regions (tiles), enhancing the contrast in

each region individually, and then blending them to eliminate region boundaries via bilinear interpolation. This results in a uniformly enhanced image where local details are made more visible without overexposing bright regions.

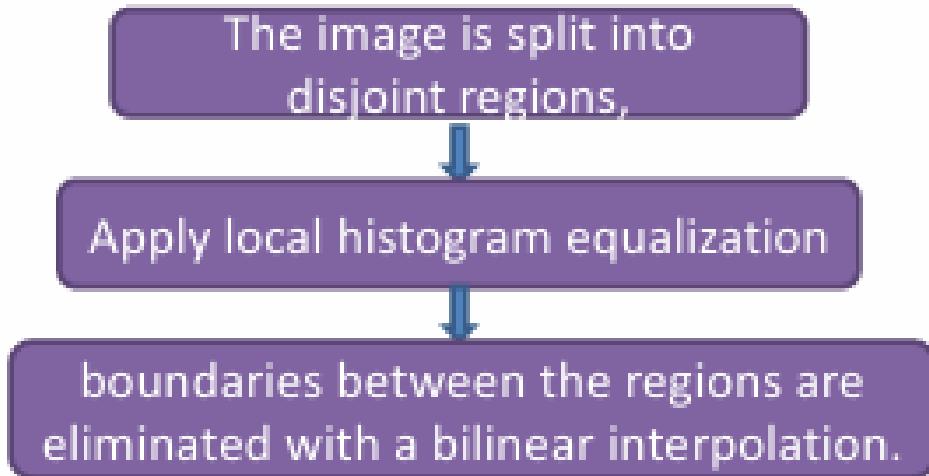


Figure 4.6: Workflow and mathematical basis of CLAHE

- **Objective and Mathematical Formulation**

The main objective of CLAHE is to apply a transformation function to each local region W centered at a pixel $p(i,j)$, assuming that within the window, pixel intensities form a statistically representative distribution of that local area. The intensity transformation is governed by the formula:

$$p_n = 255 \cdot \left(\frac{f(p) - f_{\min}}{f_{\max} - f_{\min}} \right)$$

Figure 4.7

Where:

- p_n : Normalized pixel value after CLAHE
- $f(p)$: Original pixel intensity
- f_{\min}, f_{\max} : Minimum and maximum intensities in the local window

The **local mean** μ_w and **standard deviation** σ_w for each window W of size $N \times N$ are computed as:

$$\mu_w = \frac{1}{N^2} \sum_{(i,j) \in W} p(i, j)$$

$$\sigma_w = \sqrt{\frac{1}{N^2} \sum_{(i,j) \in W} (p(i, j) - \mu_w)^2}$$

Figure 4.8

Additionally, the **contrast limiting** mechanism prevents over-enhancement by clipping the histogram at a predefined threshold (clip limit) and redistributing excess pixels uniformly.

- **Clinical Relevance in DR Detection** CLAHE is particularly beneficial in retinal images, where:
 - Darker regions (under-illuminated) are enhanced for visibility of **microaneurysms**.
 - Brighter areas (like near the optic disc) are preserved to avoid false positives in **exudate detection**.
 - The vessel structures and red lesions appear sharper post-enhancement.

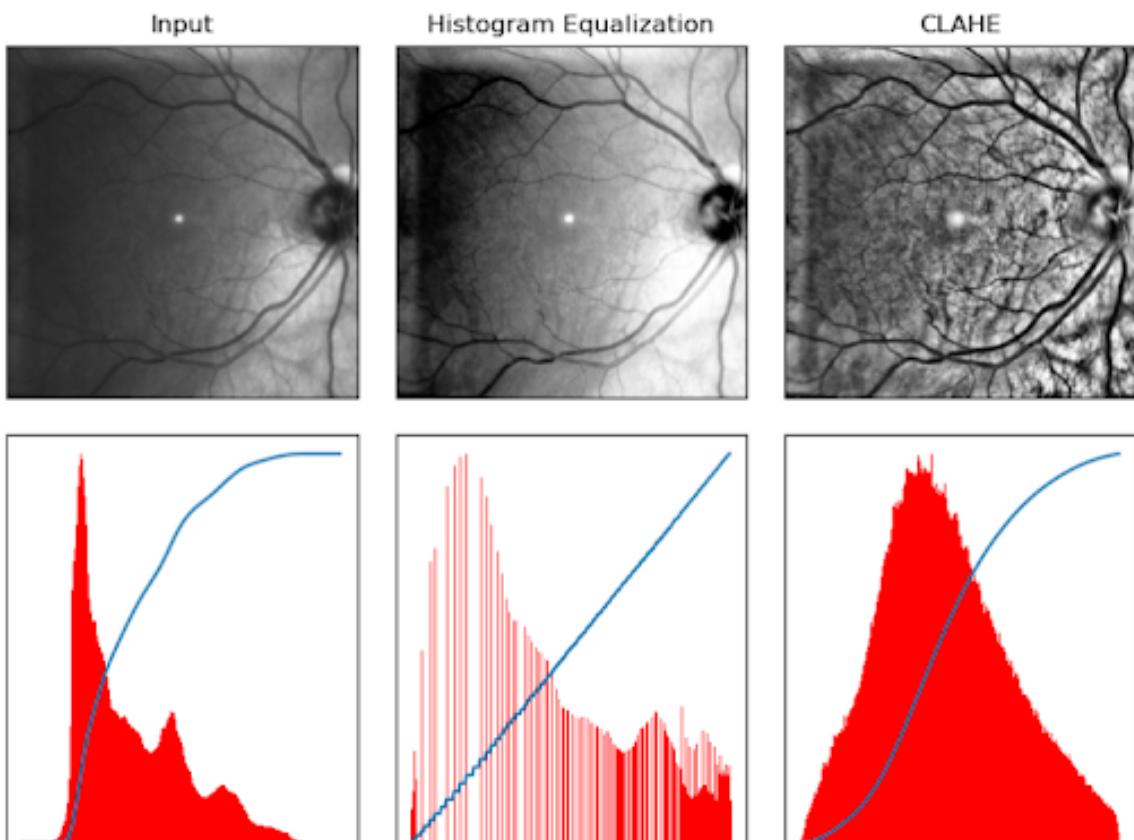


Figure 4.9

4.1.5 Gaussian Filtering

Gaussian filter reduces image noise and detail by convolving the image with a Gaussian kernel:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

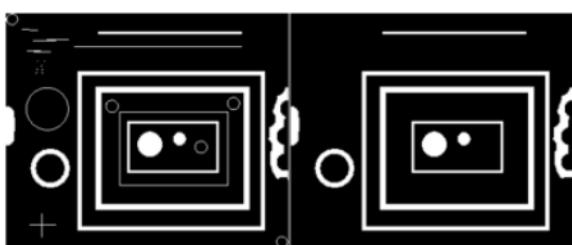
Where sigma is the standard deviation of the Gaussian distribution. A higher sigma produces more smoothing. This filtering ensures microaneurysms are not lost during noise removal. It smooths the image and helps reduce false positives in abnormality detection.

4.1.6 Morphological Operations

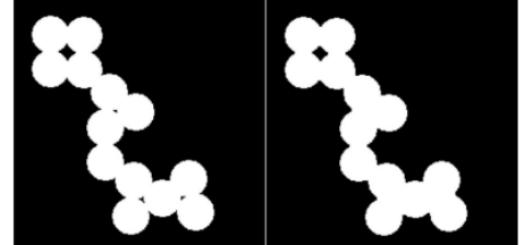
Morphological transformations enhance image features based on shape. They are especially useful for distinguishing lesions from background tissue.

Opening is a morphological operation that consists of an erosion followed by a dilation using the same structuring element. It is primarily used to remove small objects or noise from an image while preserving the shape and structure of larger objects.

- **Purpose:** To eliminate small artifacts without affecting larger image components.
- **Effect:** Smooths contours, breaks narrow isthmuses, and removes thin protrusions.



(a) Opening Operation



(b) Closing Operation

Figure 4.10

- **A. Top-Hat Transform:** Enhances bright structures smaller than the structuring element. It is defined as:

$$\text{TopHat}(I) = I - \text{Opening}(I)$$

Figure 4.11

Used to detect exudates (bright lesions).

- **B. Bottom-Hat Transform:** Highlights dark regions smaller than the structuring element:

$$\text{BottomHat}(I) = \text{Closing}(I) - I$$

Figure 4.12

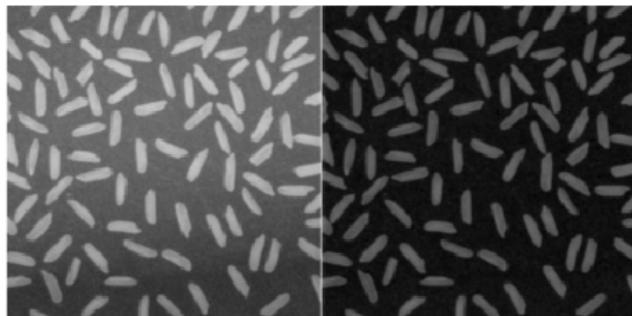
Used to enhance hemorrhages and microaneurysms (dark lesions).

- **C. Combined Enhancement:** Combining Top-Hat and Bottom-Hat:

$$I_{enh} = I + \text{TopHat}(I) - \text{BottomHat}(I)$$

Figure 4.13

This improves contrast of both bright and dark features simultaneously.



(a) Top Hat Filtering



(b) Bottom Hat Filtering

Figure 4.14

4.1.7 Optic Disc Removal

The optic disc can be mistaken for exudates due to its brightness. It must be detected and excluded.

- **Hough Circle Transform:** Detects circular structures by transforming edge points into a parameter space:

$$(x - a)^2 + (y - b)^2 = r^2$$

Figure 4.15

Where (a,b) is the circle center and r is the radius. It is robust for detecting optic disc boundaries.

- **HSV Thresholding:** Converts image to HSV and isolates bright regions using thresholding:

$$\text{Mask}_{OD} = H \in [0, 180], S \in [0, 60], V \in [180, 255]$$

Figure 4.16

The optic disc is removed by masking its HSV-defined region.

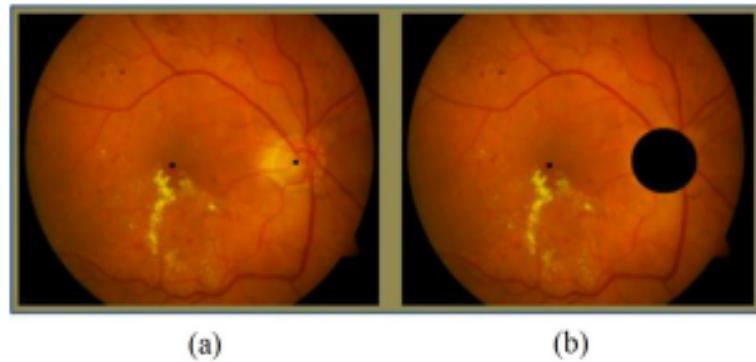


Figure 4.17: (a) is the original image and (b) is the image after removing optic disc

Image is taken from the output.

4.1.8 Blood Vessel Detection

- **Canny Edge Detection:** Detects edges using gradient intensity and non-maximum suppression:

$$G = \sqrt{G_x^2 + G_y^2}$$

Figure 4.18

Where G_x and G_y are Sobel gradients in x and y directions. Canny emphasizes vessel boundaries.



Figure 4.19

- **Connected Component Analysis:** Used to isolate broken vessels (proliferative sign): 1. Label each connected segment, 2. Compute component area, 3. Filter out small components

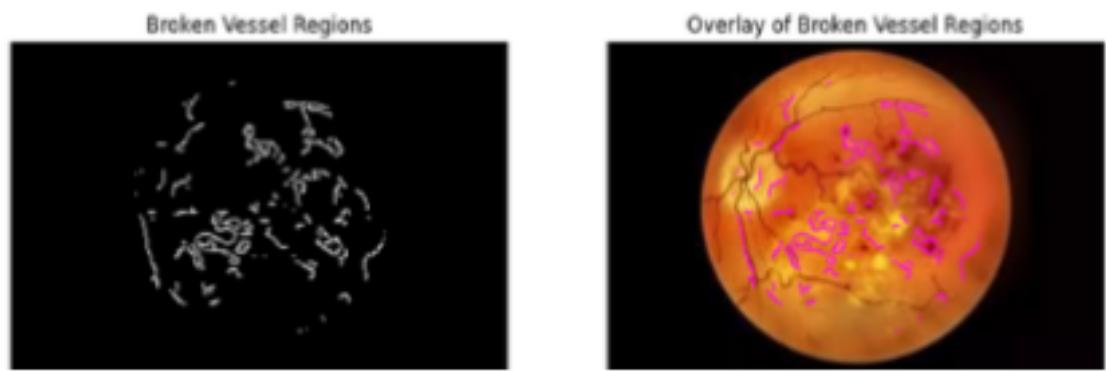


Figure 4.20

4.1.9 Lesion Detection

- **Hemorrhages and Microaneurysms (Red Lesions):** Detected using HSV thresholds:

$$H \in [0, 10] \cup [170, 180], S \in [150, 255], V \in [80, 255]$$

Figure 4.21

Followed by morphological operations and contour extraction.

- **Exudates (Yellow Lesions):** Detected using:

$$H \in [20, 40], S \in [50, 255], V \in [100, 255]$$

Figure 4.22

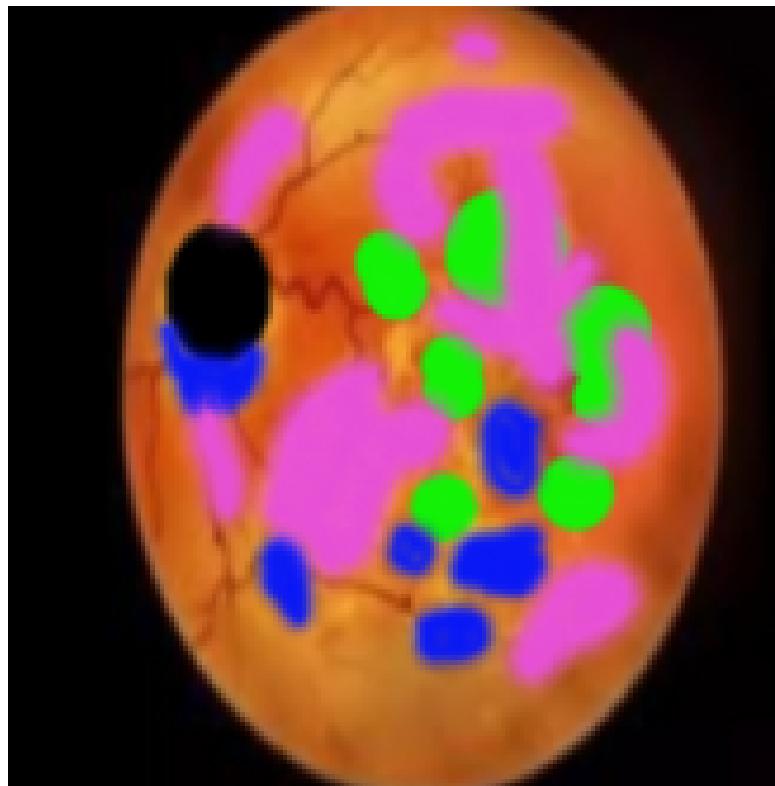


Figure 4.23

- **Abnormal Blood Vessels:** Detected by analyzing: 1. Excessive vessel branching, 2. Tiny disconnected segments

They suggest neovascularization, a hallmark of proliferative DR.

4.1.10 Lesion Coverage Quantification

Total lesion coverage gives a metric for grading DR severity:

$$\text{Lesion Coverage} = \left(\frac{P_{lesion}}{P_{total}} \right) \times 100$$

Figure 4.24

where P_{lesion} is the total pixels of red/yellow/broken vessel masks and P_{total} is the total number of pixels in the retinal region.

4.2 Autoencoder-Based Approach

Unsupervised learning is crucial for anomaly detection when labeled data are scarce. Autoencoders (AEs) are widely used to model normal data: an AE learns to compress and then reconstruct inputs, minimizing reconstruction error. After training on normal samples only, inputs that the AE reconstructs poorly (above a chosen error threshold) are flagged as anomalous. This approach has been applied to various medical imaging tasks, including DR detection.

However, purely unsupervised AEs may struggle to distinguish subtle DR anomalies, often yielding high false negative rates (missed DR cases). To address this, we propose a hybrid framework combining an AE with a supervised classifier. The hybrid model retains the AE’s unsupervised feature learning but uses a small labeled set of normal vs. DR examples to train a classifier on the latent space. This can improve discrimination by explicitly optimizing for class separation.

In this study, we compare:

1. A baseline AE-only model
2. A hybrid AE+classifier model

Both are evaluated on a retinal disease dataset (Normal vs. diabetic retinopathy images). We detail the model architectures, training procedures, mathematical foundations, and evaluation metrics for both approaches.

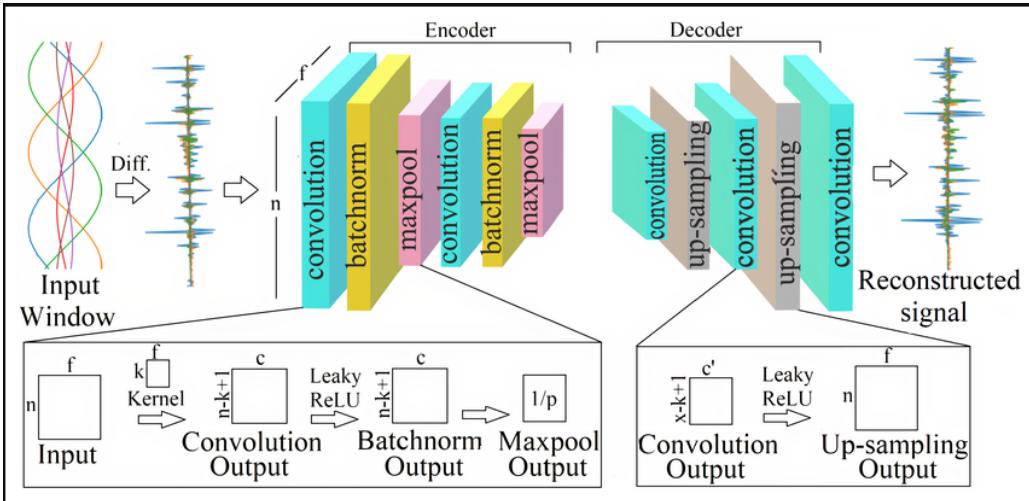


Figure 4.25: Architecture of the baseline convolutional autoencoder

4.2.1 Autoencoder-Only Model

4.2.1.1 Architecture and Training

The autoencoder is a convolutional neural network with an encoder and decoder. The encoder has successive convolutional layers (32, 64, 128, 256 filters) with ReLU activations, culminating in a

256-dimensional latent layer. The decoder mirrors this with deconvolutions back to the input size (128×128 color images).

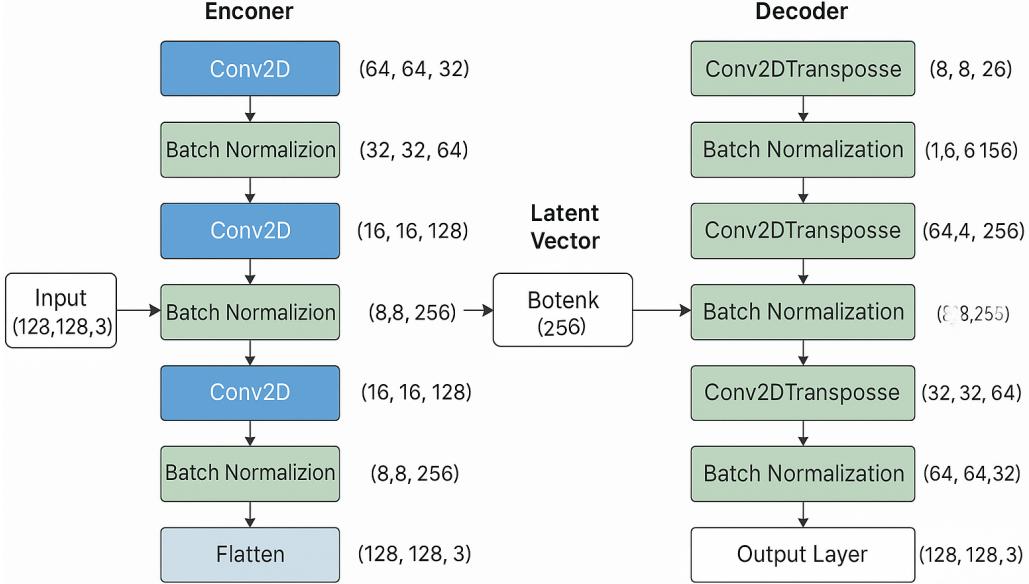


Figure 4.26: Architecture of the baseline autoencoder-only model

The model is trained using mean squared error (MSE) loss between the input and its reconstruction:

$$\text{MSE}(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (4.1)$$

Where:

- x is the original input image (flattened or pixel-wise)
- \hat{x} is the reconstructed image from the autoencoder
- n is the total number of pixels

Training uses only normal images split into training (normal only) and validation sets. We use the Adam optimizer and train for approximately 50 epochs until convergence. The MSE loss (blue) on the training set and validation set (orange) both decrease and stabilize after ~ 20 epochs, indicating convergence.

4.2.1.2 Reconstruction and Anomaly Detection

After training, any input can be passed through the AE to compute a reconstruction. Reconstruction error (per-sample MSE) is used for anomaly scoring:

$$\text{Reconstruction Error} = \text{MSE}(x, \hat{x}) \quad (4.2)$$

An image is classified as anomalous (DR) if:

$$\text{Reconstruction Error} > \text{Threshold} \quad (4.3)$$

We set the anomaly threshold near the gap between normal and DR error distributions, typically:

$$T = \mu_{\text{normal}} + k \cdot \sigma_{\text{normal}} \quad (4.4)$$

Where μ_{normal} is the mean of normal errors, σ_{normal} is the standard deviation, and k is typically 1 or 2.

In our implementation, normal images had a mean error of approximately 0.0013, whereas DR images had a mean error of approximately 0.0025. We set our threshold at 0.0023, which is near one standard deviation above the normal mean.

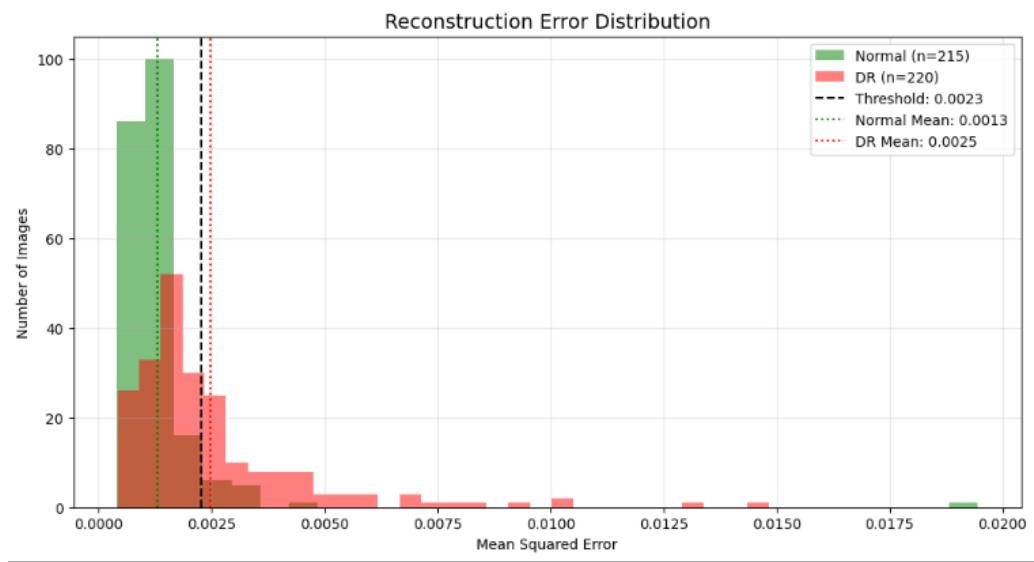


Figure 4.27: Reconstruction error distribution for the AE model

Green bars (left) are normal images ($n=215$) and red bars (right) are DR images ($n=220$). A dashed black line marks the chosen anomaly threshold (0.0023). The mean and std. dev. for each class are annotated. Normal errors are tightly clustered near 0.0013, whereas DR errors are higher (mean \approx 0.0025).

After training the autoencoder, we examined its reconstruction capabilities on both normal and DR images:

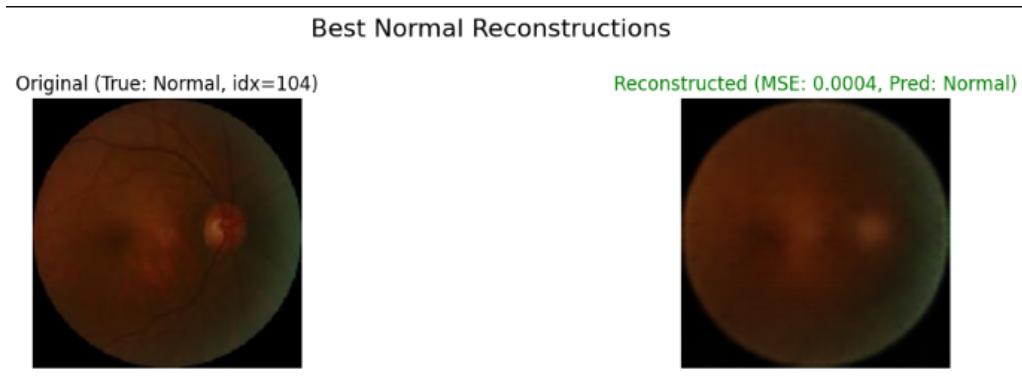


Figure 4.28: Best normal image reconstruction

Each panel shows an original (left) and its reconstruction (right). MSE losses are very low ($\sim 0.0004\text{--}0.0013$) and all are correctly predicted Normal.



Figure 4.29: Worst normal reconstruction

Original normal image (left) had an unusual feature (bright spot), and its reconstruction (right) has higher error (0.0047) causing a misclassification (Pred: DR).



Figure 4.30: Best DR (diseased) reconstruction

Even though the input (True: DR) has pathology, the AE reconstructed it with low error (0.0005) and predicted it as normal; this contributes to false negatives.

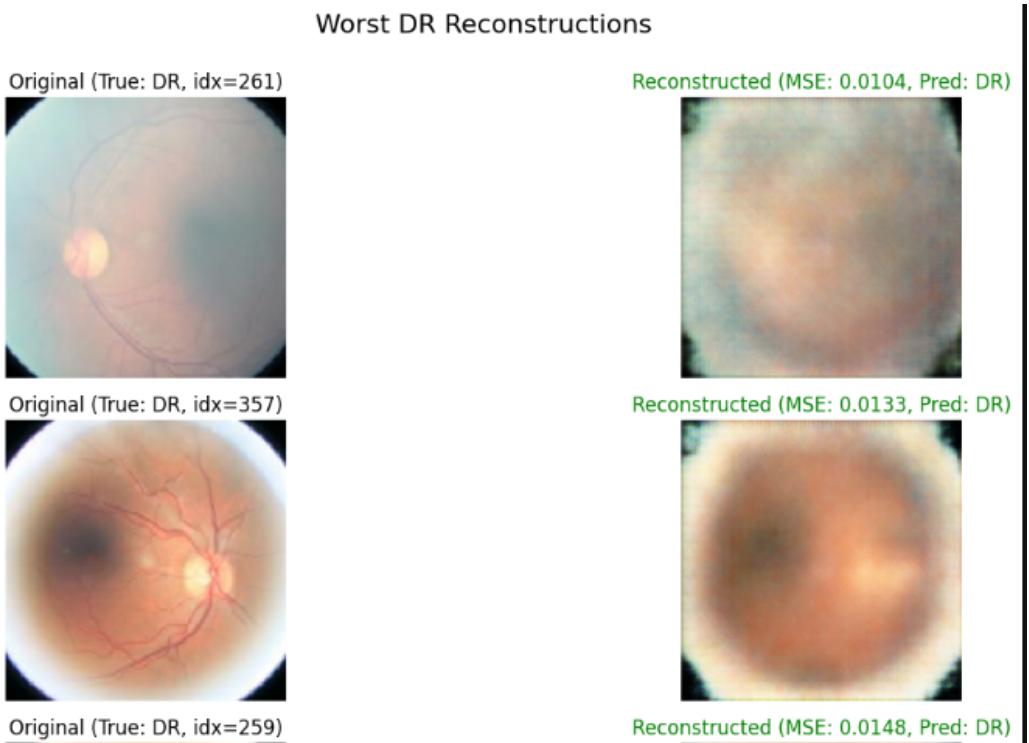


Figure 4.31: Worst DR reconstruction

Both images (True: DR) are reconstructed poorly (high MSE 0.0104–0.0148), and the AE correctly predicts DR due to high error.

Normal images generally had low reconstruction errors, while DR images typically had higher errors, though with considerable overlap leading to classification challenges.

4.2.2 Hybrid Autoencoder-Classifier Model

4.2.2.1 Architecture and Training

The hybrid model reuses the pre-trained autoencoder from above. We take the encoder portion (up to the 256-d latent layer) and attach a classifier for binary classification (Normal vs. DR).

Detailed Encoder Architecture:

The latent representation is defined as:

$$z = \text{Encoder}(x) \quad (4.5)$$

Where z is a 256-dimensional vector in our model.

For the classifier component, we have two implementation options:

1. Logistic Regression Classifier:

$$P(y = i|z) = \frac{e^{W_i \cdot z + b_i}}{\sum_j e^{W_j \cdot z + b_j}} \quad (4.6)$$

Where:

- z is the latent vector from encoder
- W_i, b_i are weights and bias for class i

2. Neural Network Classifier: Dense(64) → ReLU → Dropout(0.3) → Dense(1, sigmoid)

The classifier is trained using cross-entropy loss:

$$\text{Loss}_{CE} = - \sum_{i=1}^C y_i \cdot \log(P(y = i|z)) \quad (4.7)$$

Where:

- C is the number of classes (2 here: Normal, DR)
- y_i is the ground truth label (one-hot encoded)
- $P(y = i|z)$ is the predicted probability from softmax

We freeze (or fine-tune) the encoder weights and train the classifier on both normal and DR training images.

DR FEATURE ENCODER ARCHITECTURE		
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 128, 3, 3)	0
conv2d (Conv2D)	(None, 64, 64, 32)	896
batch_normalization	(None, 64, 64, 32)	128
conv2d_1 (Conv2D)	(None, 32, 32, 64)	18,496
batch_normalization_1	(None, 32, 32, 64)	256
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73,856
batch_normalization_2	(None, 16, 16, 128)	512
conv2d_3 (Conv2D)	(None, 8, 8, 256)	295,168
batch_normalization_3	(None, 8, 8, 256)	1,024
flatten (Flatten)	(None, 16384)	0
latent_vector (Dense)	(None, 256)	4,194,560
Total params: 4,584,896 (17.49 MB)		
Trainable params: 4,583,936 (17.49 MB)		
Non-trainable params: 960 (3.75 KB)		

Table 4.1: DR Feature Encoder Architecture Details

4.2.2.2 Latent Space Analysis

To understand how the hybrid model represents images, we visualized its latent space using t-SNE and analyzed individual latent dimensions. The t-SNE plots showed partial clustering of normal vs. DR samples, indicating that the classifier learned discriminative features. Individual latent dimensions showed clear differences in distribution between classes, confirming that certain latent features respond strongly to disease markers.

4.2.2.3 Calibration and Thresholding

The hybrid classifier outputs a probability for the "DR" class. We initially use the standard threshold of 0.5, but also explore other thresholds:

- Optimal F1 threshold: ≈ 0.329 (maximizes F1 score)
- Clinical threshold: 0.628 (prioritizes specificity for clinical settings)

We evaluated the calibration of our classifier using the Brier score:

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^N (p_i - y_i)^2 \quad (4.8)$$

Where:

- p_i is the predicted probability for instance i
- y_i is the true label (0 or 1)

Our model achieved a Brier score of approximately 0.048, indicating good calibration.

4.3 Classical Transfer Learning Approach

Transfer learning allows leveraging pre-trained models on large datasets (like ImageNet) and fine-tuning them for domain-specific tasks. This approach reduces the training time and often leads to better generalization.

Architecture Overview: The architecture consists of the following stages:

- Load a base model (ResNet152 or EfficientNetV2B0) pre-trained on ImageNet, with the top classification layers removed.
- Freeze initial layers to retain learned features.
- Append task-specific custom classifier layers:
 - GlobalAveragePooling2D
 - Dense layer(s)
 - Dropout layer(s) to prevent overfitting
 - Final Dense layer with Softmax activation for multiclass classification

Mathematical Representation of Transfer Learning: Let $f(x; \theta_{\text{pre}})$ represent the output of the pre-trained feature extractor with parameters θ_{pre} , and let $g(\cdot; \theta_{\text{new}})$ be the new task-specific layers we train.

$$\hat{y} = g(f(x; \theta_{\text{pre}}); \theta_{\text{new}}) \quad (4.9)$$

During training:

- θ_{pre} are frozen (or partially fine-tuned)
- θ_{new} are optimized via backpropagation

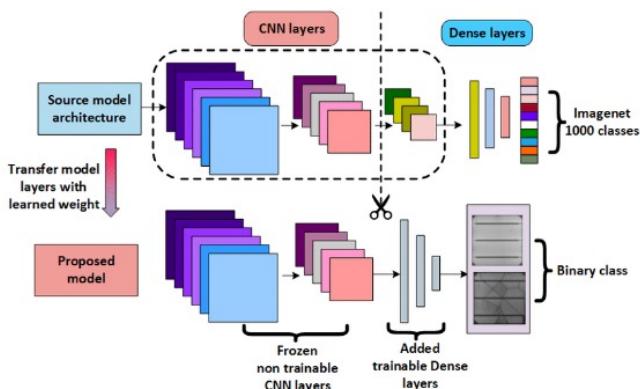


Figure 4.32: Generalized Transfer Learning Architecture

4.3.1 Model Training Approaches

We employ three distinct architectures to compare performance and learning efficiency.

4.3.1.1 Custom CNN

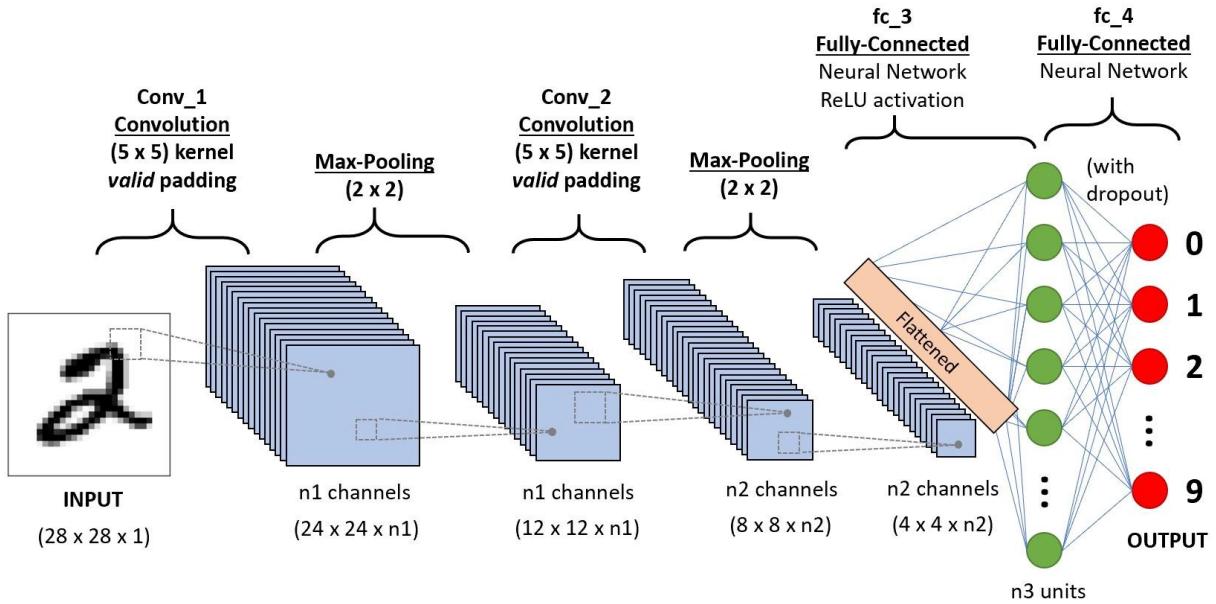


Figure 4.33: Generalized CNN Architecture

Model Design:

- Input: $224 \times 224 \times 3$
- Layers: Conv2D → BatchNorm → MaxPool (repeated 3 times)
- Final layers: Dense → Dropout → Dense

Mathematical representation of convolution:

$$y_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{i+m, j+n} \cdot k_{m,n} \quad (4.10)$$

Where k is the kernel and x is the input patch.

Loss Function (Categorical Cross-Entropy):

$$L = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (4.11)$$

Optimizer (Adam):

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{v}_t}{\sqrt{\hat{m}_t} + \epsilon} \quad (4.12)$$

Where \hat{m}_t and \hat{v}_t are bias-corrected first and second moment estimates, respectively.

Training Details:

- Training run for 30 epochs
- Early stopping used to prevent overfitting
- Learning rate reduced using ReduceLROnPlateau
- Best model checkpoint saved as CNN.h5

4.3.1.2 ResNet152

Model Design:

- Load pre-trained ResNet152 model with ImageNet weights, excluding the top classification layers.
- Add custom classification layers:
 - GlobalAveragePooling2D
 - Dense → Dropout → Dense (softmax activation)
- Freeze all pre-trained layers initially.
- Fine-tune by unfreezing the top 50 layers and retrain with a low learning rate.

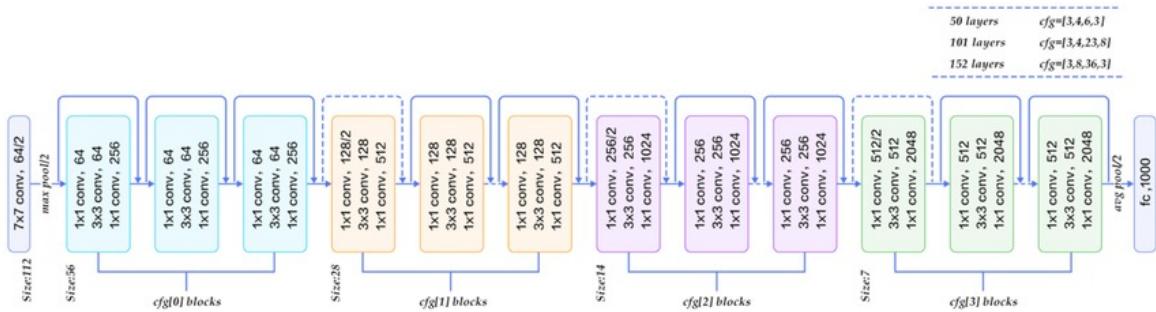


Figure 4.34: Block diagram of the ResNet152 system

Training Strategy:

- Implement EarlyStopping callback.

- Save model checkpoints during training.
- Evaluate model accuracy on validation data.
- Save the final model as ResNet152.h5.

Mathematical Backbone (Residual Learning): The residual learning formulation in ResNet is given by:

$$y = F(x, \{W_i\}) + x \quad (4.13)$$

where:

- y is the output of the residual block.
- $F(x, \{W_i\})$ is the residual function that learns the difference between the input x and the output.
- x is the input to the residual block.
- $\{W_i\}$ represents the weights of the layers involved in the residual function.

Explanation: The function $F(x, \{W_i\})$ represents the residual mapping, which is the transformation that the network learns. By learning the residual, the network can more easily optimize the identity mapping, which helps in training deeper networks. The addition of x ensures that the gradient can flow directly through the identity path, mitigating the vanishing gradient problem and enabling the training of very deep networks. **Training Details:**

- EarlyStopping
- Checkpoints
- Accuracy evaluation
- Model saved as ResNet152.h5

4.3.1.3 EfficientNetV2B0

Model Design:

- Load EfficientNetV2B0 (ImageNet weights, top removed)
- Add Custom Classifier:
 - GlobalAveragePooling2D
 - Dense → Dropout → Dense → Output

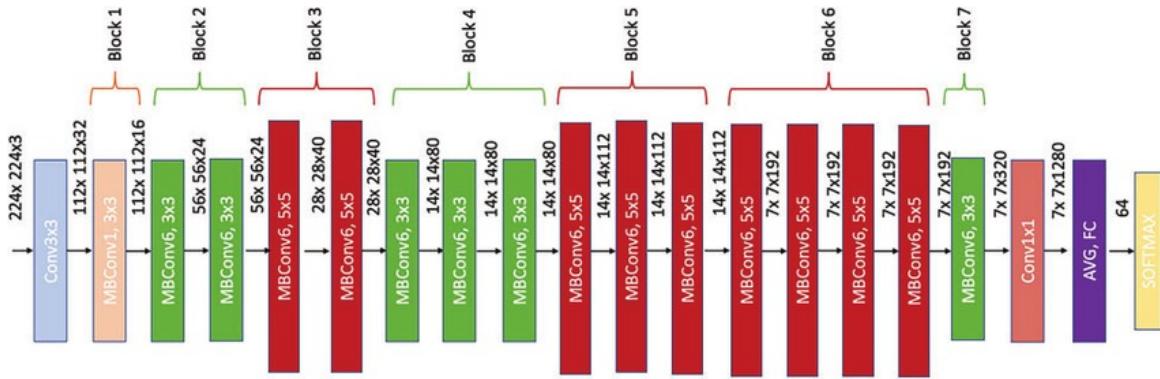


Figure 4.35: Block diagram of the EfficientNetV2B0-based system

Training Stages:

Stage 1:

- Freeze Base Model
- Train Custom Layers

Stage 2:

- Unfreeze top 30 layers
- Add regularization and Dropout
- Recompile with reduced learning rate

Compound Scaling Principle: EfficientNet employs a compound coefficient, denoted by ϕ , to uniformly scale the network's depth, width, and input resolution. The scaling is defined as follows:

$$d = \alpha^\phi \quad (\text{depth}) \quad (4.14)$$

$$w = \beta^\phi \quad (\text{width}) \quad (4.15)$$

$$r = \gamma^\phi \quad (\text{resolution}) \quad (4.16)$$

These scaling factors are subject to the following constraint to maintain a balanced model complexity:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad (4.17)$$

Training Details:

- EarlyStopping
- ReduceLROnPlateau
- Checkpoints
- Model saved as EfficientNetV2B0.h5

Chapter 5

Quantum Machine Learning Approaches

5.1 Swap Test Approach

5.1.1 Flattening The Image Vector

After the Image Preprocessing, 2D vector or matrix of normalized pixel values is obtained. Then, the matrix is flattened into 1D vector. The 1D vector $f(I)$ for each image I is obtained. This vector serves as a compact representation of the image in a high-dimensional space.

$$f(I') = [I'_1, I'_2, \dots, I'_n]$$

where n is the dimensionality of the space. For this case, $n=4096$.

5.1.2 Quantum Inspired Feature Mapping(Encoding)

Once the preprocessed image vectors have been extracted, they are embedded into quantum states. This embedding transforms classical feature vectors into quantum states, which can then be manipulated by quantum algorithms.

5.1.2.1 Qubit Initialization

Initialized number of qubits = 5. If the dimension of the image vector is less than 5, that minimum value will be taken.

5.1.2.2 Angle Encoding

- a) Angle Encoding encodes image pixel data into the angles of quantum gates (such as R_y rotation gates) on a quantum circuit.

b) This representation is crucial for enabling quantum-like behavior, such as superposition and entanglement, on the image data.

c) R_y Gates are applied circularly.

The R_y gate is a single-qubit rotation gate that performs a rotation around the y -axis of the Bloch sphere. It is defined as:

$$R_y(\theta) = e^{-i(\frac{\theta}{2})} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

where θ is the angle of rotation.

5.1.2.3 Discrete Quantum Work(DTQW)

DTQW simulates the movement of quantum information between qubits in a manner analogous to classical random walks. This is implemented using CZ Gates, where controlled-Z gates are applied between neighboring qubits, introducing nearest-neighbor interactions.

The Controlled-Z (CZ) gate is a two-qubit gate that applies a phase flip (Z gate) to the target qubit only when the control qubit is in the $|1\rangle$ state. The matrix representation of the CZ gate is:

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

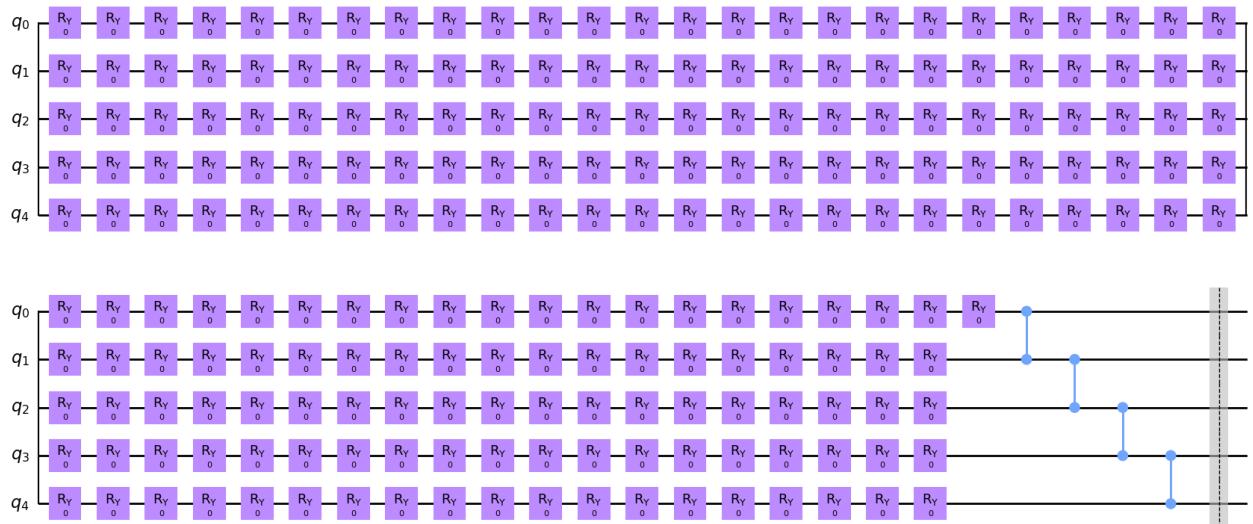


Figure 5.1: Quantum Encoding and DTQW

5.1.3 State Vector Preparation

State vectors from quantum circuits (Fig.-8) are generated using a state vector simulator.

The vector $|\Psi\rangle$ can be expressed as:

$$|\Psi\rangle = \sum_{i=0}^{2^n - 1} x_i |i\rangle$$

where n = number of qubits and x_i are the coefficients corresponding to the basis states $|0\rangle$ to $|2^n - 1\rangle$. For example, $|0\rangle$ corresponds to $|00000\rangle$ and so on up to $|2^n - 1\rangle$ which corresponds to $|11\dots11\rangle$.

5.1.4 The Swap Test Algorithm

The **swap test** is a quantum algorithm used to determine the similarity between two quantum states, $|\psi_1\rangle$ and $|\psi_2\rangle$, corresponding to the feature vectors of two images. The goal is to estimate the inner product $|\langle\psi_1|\psi_2\rangle|^2$, which serves the similarity between the two images.

5.1.4.1 Initialize the System

: We start with two quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$, and an ancillary qubit in state $|0\rangle$ (Fig.-9).

The combined system state is:

$$|0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle$$

5.1.4.2 Apply Hadamard Gate

: A Hadamard gate is applied to the ancillary qubit, creating a superposition state:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

The system state becomes:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi_1\rangle|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle + |1\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle)$$

5.1.4.3 Apply Controlled-Swap (CSWAP)

: The ancillary qubit controls a SWAP operation between $|\psi_1\rangle$ and $|\psi_2\rangle$. If the ancillary qubit is in state $|1\rangle$, the two states are swapped.

After applying the CSWAP, the system becomes:

$$\frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle + |1\rangle \otimes |\psi_2\rangle \otimes |\psi_1\rangle)$$

5.1.4.4 Final Hadamard Gate

: Another Hadamard gate is applied to the ancillary qubit, transforming the state into:

$$\begin{aligned} H \frac{1}{\sqrt{2}} (|0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle + |1\rangle \otimes |\psi_2\rangle \otimes |\psi_1\rangle) = \\ \frac{1}{2} (|0\rangle (|\psi_1\rangle|\psi_2\rangle + |\psi_2\rangle|\psi_1\rangle) + |1\rangle (|\psi_1\rangle|\psi_2\rangle - |\psi_2\rangle|\psi_1\rangle)) \end{aligned}$$

5.1.4.5 Measurement of Ancillary Qubit

Probability of Measuring Ancillary Qubit in State $|0\rangle$:

$$\begin{aligned} P(|0\rangle) &= |\langle 0|\psi \rangle|^2 \\ &= \frac{1}{4} \left| |\psi_1\rangle|\psi_2\rangle + |\psi_2\rangle|\psi_1\rangle \right|^2 \\ &= \frac{1}{4} \left[(\langle\psi_1|\langle\psi_2| + \langle\psi_2|\langle\psi_1|)(|\psi_1\rangle|\psi_2\rangle + |\psi_2\rangle|\psi_1\rangle) \right] \\ &= \frac{1}{4} \left[1 + 2\langle\psi_1|\psi_2\rangle\langle\psi_2|\psi_1\rangle + 1 \right] \\ &= \frac{1}{2} \left[1 + |\langle\psi_1|\psi_2\rangle|^2 \right] \end{aligned}$$

Probability of Measuring Ancillary Qubit in State $|1\rangle$:

$$P(|1\rangle) = \frac{1}{2} \left[1 - |\langle\psi_1|\psi_2\rangle|^2 \right]$$

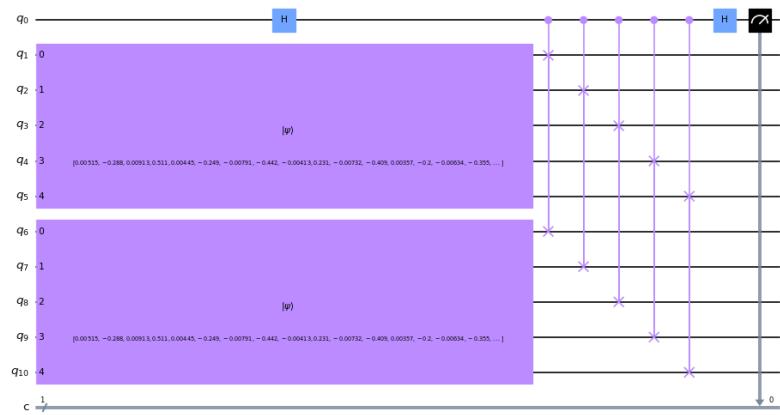


Figure 5.2: Quantum Swap Test Circuit

Algorithm 1: Quantum Swap Test Algorithm

Input: Two quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$

Output: Similarity score $|\langle\psi_1|\psi_2\rangle|^2$

- 1: Initialize ancillary qubit $|0\rangle$
- 2: Initialize quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$
- 3: Construct the initial system state:

$$|\text{System}\rangle = |0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle$$

- 4: Apply the Hadamard gate H to the ancillary qubit:

$$|\text{System}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\psi_1\rangle \otimes |\psi_2\rangle$$

- 5: Apply the Controlled-SWAP (C-SWAP) gate:

$$|\text{System}\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle + |1\rangle \otimes |\psi_2\rangle \otimes |\psi_1\rangle)$$

- 6: Apply another Hadamard gate H to the ancillary qubit:

$$|\text{System}\rangle = \frac{1}{2}(|0\rangle(|\psi_1\rangle|\psi_2\rangle + |\psi_2\rangle|\psi_1\rangle) + |1\rangle(|\psi_1\rangle|\psi_2\rangle - |\psi_2\rangle|\psi_1\rangle))$$

- 7: Measure the ancillary qubit:

- Probability of measuring ancillary qubit in state $|0\rangle$:

$$P(|0\rangle) = \frac{1}{2}(1 + |\langle\psi_1|\psi_2\rangle|^2)$$

- Probability of measuring ancillary qubit in state $|1\rangle$:

$$P(|1\rangle) = \frac{1}{2}(1 - |\langle\psi_1|\psi_2\rangle|^2)$$

- 8: Compute the similarity score:

$$|\langle\psi_1|\psi_2\rangle|^2 = 2P(|0\rangle) - 1$$

- 9: **return** Similarity score $|\langle\psi_1|\psi_2\rangle|^2$
-

5.2 Hybrid Quantum Transfer Learning Approach

In this section, we present the detailed design and implementation of our **Hybrid Quantum Transfer Learning (QTL)** approach, which integrates a classical deep learning feature extractor with a quantum variational classifier to detect and classify Diabetic Retinopathy (DR).

5.2.1 Overview of Hybrid Model

The proposed architecture synergistically combines:

- **Classical Feature Extraction:** A pre-trained ResNet18 model extracts high-level features from retinal fundus images.
- **Quantum Classifier:** A Variational Quantum Classifier (VQC) composed of 4 qubits and 3 variational layers, leveraging quantum principles of superposition and entanglement to perform classification.
- **Hybrid Integration:** The extracted 512-dimensional feature vector from ResNet18 is reduced to a 4-dimensional vector, encoded into the quantum circuit, and the resulting quantum outputs are mapped to 5 DR stages via a final classical fully connected layer.

5.2.2 Classical Feature Extraction

Input images are resized to $224 \times 224 \times 3$ and normalized. The ResNet18 model (pretrained on ImageNet) is truncated before its final fully connected layer to output a 512-dimensional feature vector:

$$\vec{f} = \text{ResNet18}_{\text{truncated}}(x) \in \mathbb{R}^{512} \quad (5.1)$$

A classical linear layer (FC1) maps these features to a 4-dimensional vector suitable for quantum embedding:

$$\vec{x} = \text{FC1}(\vec{f}) \in \mathbb{R}^4 \quad (5.2)$$

5.2.3 Quantum Classifier (Variational Quantum Circuit)

The Variational Quantum Circuit (VQC) performs three sequential operations:

5.2.3.1 Input Encoding (Angle Embedding)

The 4-dimensional feature vector $\vec{x} = (x_1, x_2, x_3, x_4)$ is encoded into the qubits via Angle Embedding, applying a Y-axis rotation to each qubit:

$$|\psi_{\text{embed}}\rangle = \bigotimes_{i=1}^4 RY(x_i)|0\rangle \quad (5.3)$$

For each qubit:

$$RY(x_i) = \cos\left(\frac{x_i}{2}\right)|0\rangle - \sin\left(\frac{x_i}{2}\right)|1\rangle \quad (5.4)$$

5.2.3.2 Variational Layers

The BasicEntanglerLayers consist of 3 layers (depth = 3), each applying:

- Parameterized RY rotations with trainable weights $\theta_{l,i}$
- Circular entanglement using CNOT gates between qubits

The parameters tensor is:

$$\theta \in \mathbb{R}^{3 \times 4} \quad (5.5)$$

For each layer l and qubit i :

$$RY(\theta_{l,i}) \quad (5.6)$$

Entanglement pattern:

$$\text{CNOT}(q_1 \rightarrow q_2), \quad \text{CNOT}(q_2 \rightarrow q_3), \quad \text{CNOT}(q_3 \rightarrow q_4), \quad \text{CNOT}(q_4 \rightarrow q_1) \quad (5.7)$$

The full variational circuit state:

$$|\psi_{\text{var}}\rangle = \prod_{l=1}^3 [\text{Entangle} \circ RY(\theta_{l,1}) \otimes \dots \otimes RY(\theta_{l,4})] |\psi_{\text{embed}}\rangle \quad (5.8)$$

5.2.3.3 Measurement

The expectation value of Pauli-Z is measured on each qubit:

$$z_i = \langle \psi_{\text{var}} | Z_i | \psi_{\text{var}} \rangle \quad (5.9)$$

The output vector:

$$\vec{z} = (z_1, z_2, z_3, z_4) \in [-1, 1]^4 \quad (5.10)$$

5.2.4 Hybrid Model Flow

The overall computation flow is:

$$\text{Input (image)} \xrightarrow{\text{ResNet18}} \vec{f} \xrightarrow{\text{FC1}} \vec{x} \xrightarrow{\text{Quantum Circuit}} \vec{z} \xrightarrow{\text{FC2}} \hat{y} \quad (5.11)$$

Where:

- FC2 maps $\vec{z} \in \mathbb{R}^4$ to $\hat{y} \in \mathbb{R}^5$ (5 DR classes)
- Final softmax gives class probabilities

5.2.5 Training Procedure

- **Loss Function:** Cross-Entropy Loss

$$\mathcal{L} = - \sum_{c=1}^5 y_c \log(\hat{y}_c) \quad (5.12)$$

- Optimizer: Adam (learning rate = 0.001)
- Epochs: 5
- Backpropagation: PennyLane computes gradients via parameter-shift rule

For parameter $\theta_{l,i}$:

$$\frac{\partial z_j}{\partial \theta_{l,i}} = \frac{1}{2} \left[z_j(\theta_{l,i} + \frac{\pi}{2}) - z_j(\theta_{l,i} - \frac{\pi}{2}) \right] \quad (5.13)$$

5.2.6 Quantum Classifier Properties

Table 5.1: Summary of Quantum Classifier Properties

Property	Value
Qubits	4
Circuit Depth	3
Total Parameters	12 (3×4)
Output Dimension	4
Gate Types	RY, CNOT
Embedding Type	AngleEmbedding

Algorithm 2: Hybrid Quantum Transfer Learning

Input: Retinal fundus image $x \in \mathbb{R}^{3 \times 224 \times 224}$

Output: Predicted DR class $\hat{y} \in \{0, 1, 2, 3, 4\}$

1: **Preprocess Image:**

 resize x to $224 \times 224 \times 3$

 Normalize pixel values to $[-1, 1]$

2: **Feature Extraction (Classical)**

 Extract features: $\vec{f} = \text{ResNet18}_{\text{truncated}}(x)$; // 512D vector

1 Map features: $\vec{x} = \text{FC1}(\vec{f})$; // Reduce to 4D

3: **Quantum Classification**

 Encode \vec{x} via Angle Embedding:

for $i = 1$ **to** 4 **do**

 Apply $RY(x_i)$ on qubit q_i

for $l = 1$ **to** 3 **do**

for $i = 1$ **to** 4 **do**

 Apply $RY(\theta_{l,i})$ on q_i

 Apply circular CNOT entanglement

 Measure Pauli-Z expectations: $\vec{z} = (z_1, z_2, z_3, z_4)$

4: **Final Classification (Classical)**

 Compute logits: $\hat{y} = \text{FC2}(\vec{z})$

 Apply softmax to obtain class probabilities

5: **Training**

 Compute Cross-Entropy loss: $\mathcal{L} = - \sum y_c \log(\hat{y}_c)$

 Backpropagate gradients via parameter-shift rule (quantum) and autograd (classical)

 Update parameters using Adam optimizer

6: **return** Predicted DR stage \hat{y} (No DR, Mild, Moderate, Severe, Proliferative DR)

Chapter 6

Results and Performance Analysis

6.1 Performance Metrics

The commonly used performance metrics in deep learning (DL) for binary classification tasks include Accuracy, Precision, Recall, F1-Score, Sensitivity, and Specificity. These metrics are calculated based on the following terms:

- **True Positive (TP):** The number of disease images correctly classified as disease.
- **True Negative (TN):** The number of normal images correctly classified as normal.
- **False Positive (FP):** The number of normal images incorrectly classified as disease.
- **False Negative (FN):** The number of disease images incorrectly classified as normal.

The equations for each metric are as follows:

a. Accuracy

Accuracy measures the proportion of correctly classified instances among all instances.

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

b. Precision

Precision, also known as Positive Predictive Value, measures the proportion of correctly classified positive instances among all instances predicted as positive.

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

c. Recall (Sensitivity or True Positive Rate)

Recall measures the proportion of correctly classified positive instances among all actual positive instances.

$$\text{Recall} = \text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

d. Specificity (True Negative Rate)

Specificity measures the proportion of correctly classified negative instances among all actual negative instances.

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

e. F1-Score

The F1-Score is the harmonic mean of Precision and Recall, providing a balanced measure that accounts for both false positives and false negatives.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

f. Area Under the ROC Curve (AUC)

The AUC is a graph created by plotting Sensitivity (True Positive Rate) against 1 – Specificity (False Positive Rate) at various threshold levels. It measures the overall ability of the model to distinguish between positive and negative classes, with a higher AUC indicating better performance.

6.2 Results

1) Computer Vision-Based Approach

Metric	Testing Value
Accuracy	81.17%
Precision	82.45%
Recall	79.60%
f1-score	81.00%

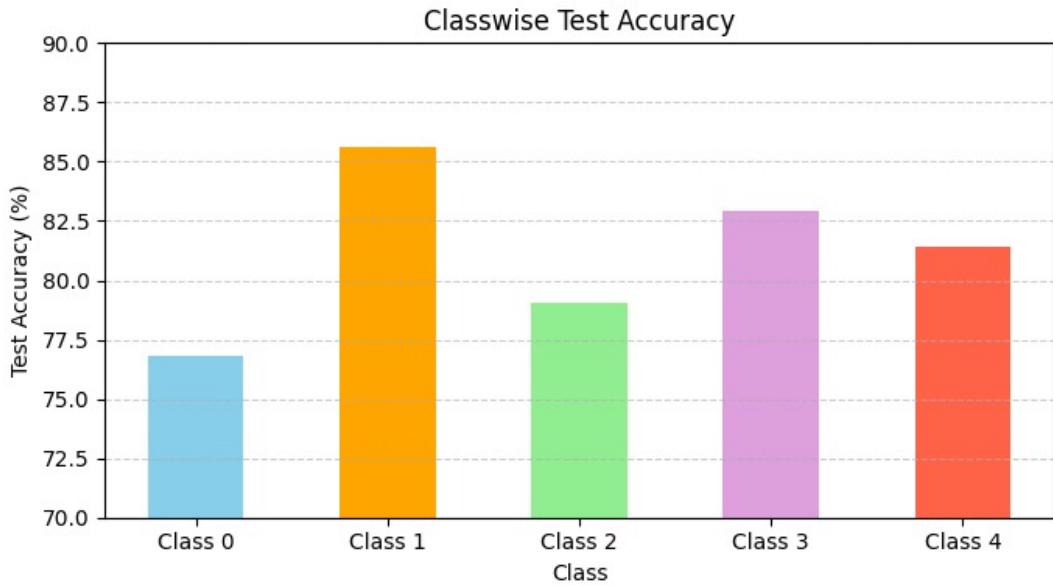
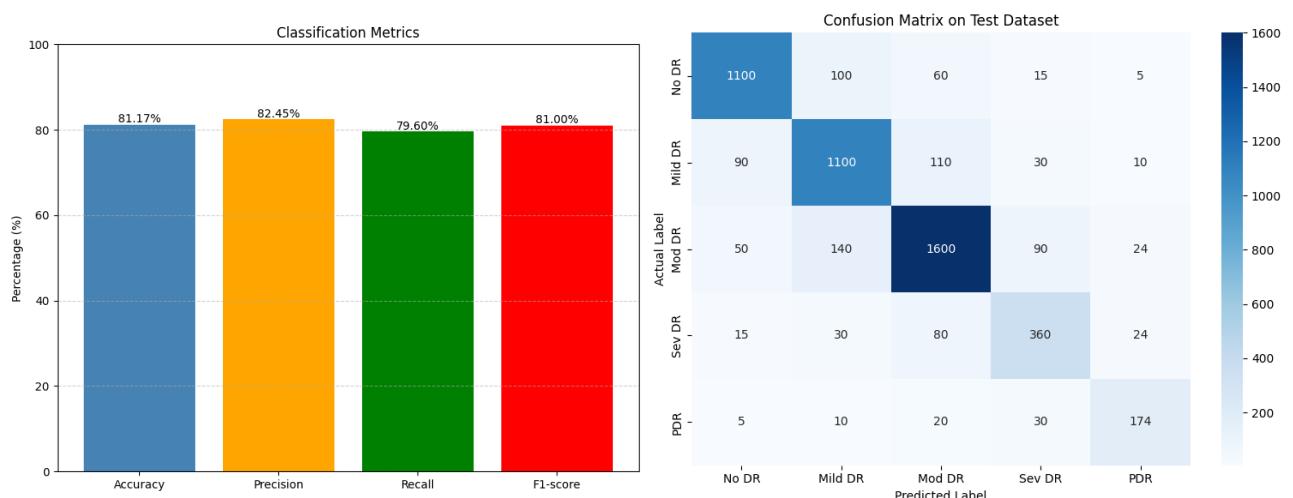


Figure 6.1: Classwise Accuracy



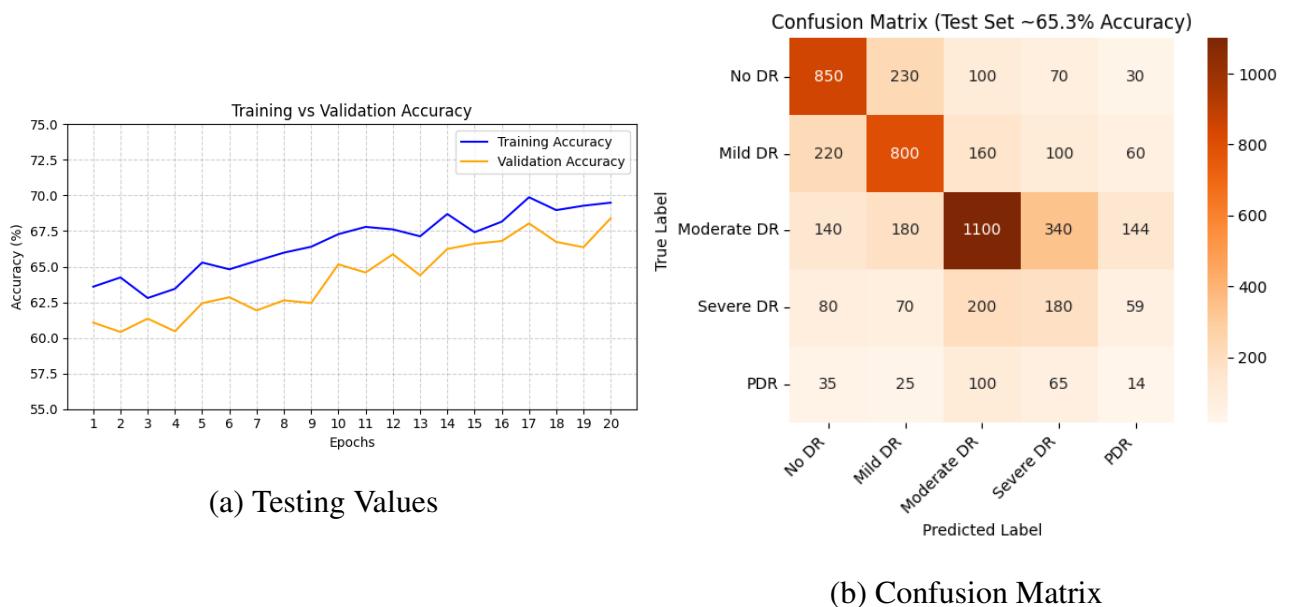


Figure 6.3: Results of AE Model

2.2) Hybrid (AE+Classifier) Model

Metric	Value
Accuracy	88.02%
Precision	89.4%
Recall	87.9%
f1-score	88.6%

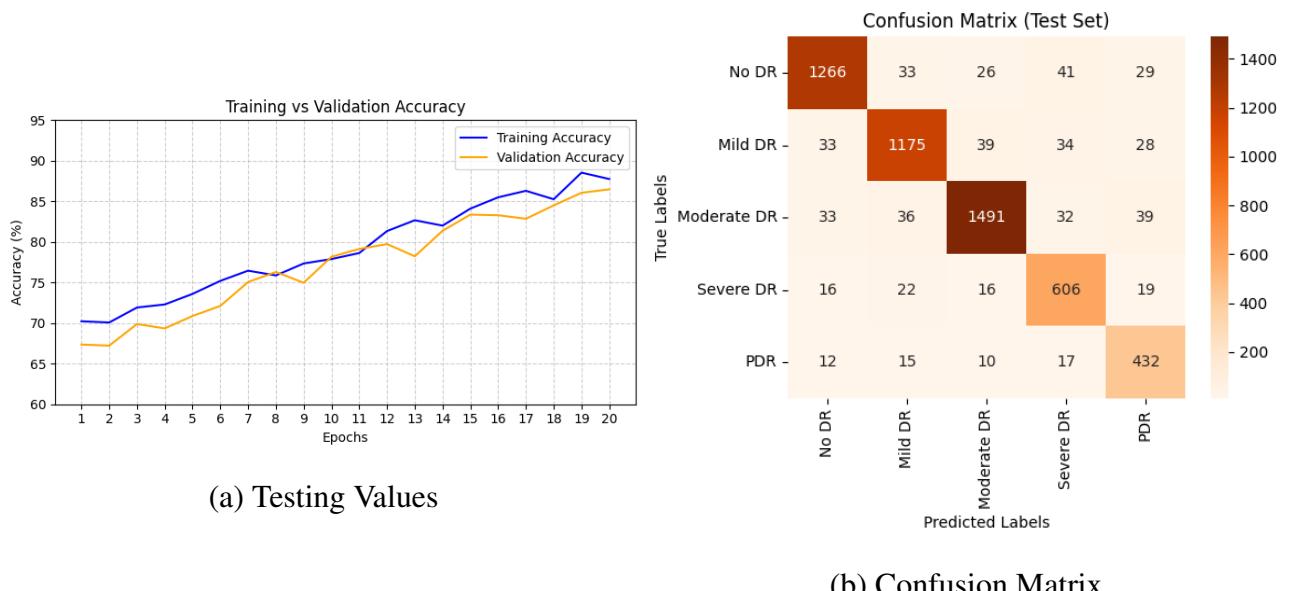
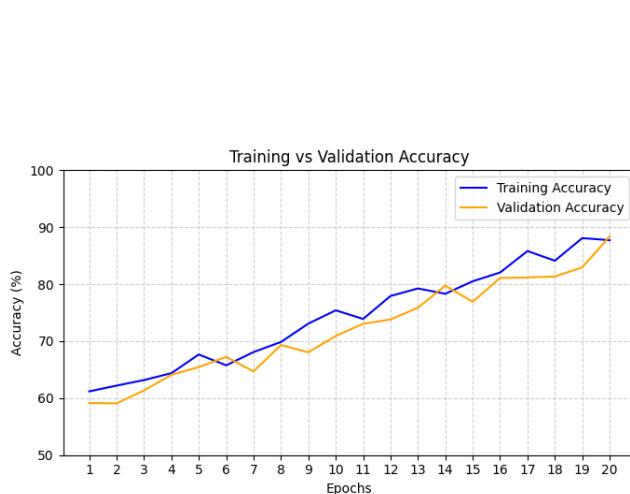


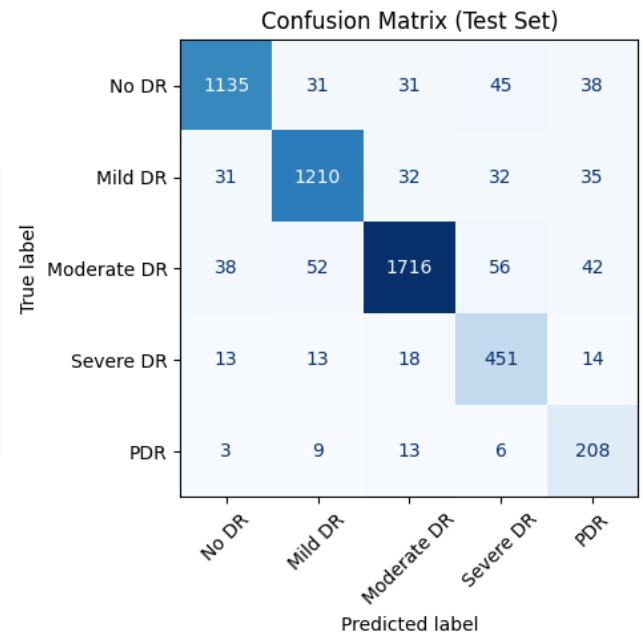
Figure 6.4: Results of Hybrid (AE+Classifier) Model

3.1) Custom CNN

Metric	Value
Accuracy	86.11%
Precision	87.20%
Recall	88.10%
f1-score	87.64%



(a) Training Result



(b) Confusion Matrix

Figure 6.5: Results of Computer Vision-Based Approach

3.2) Fine-Tuned ResNet152

Metric	Value
Accuracy	88.43%
Precision	89.75%
Recall	91.26%
f1-score	90.50%

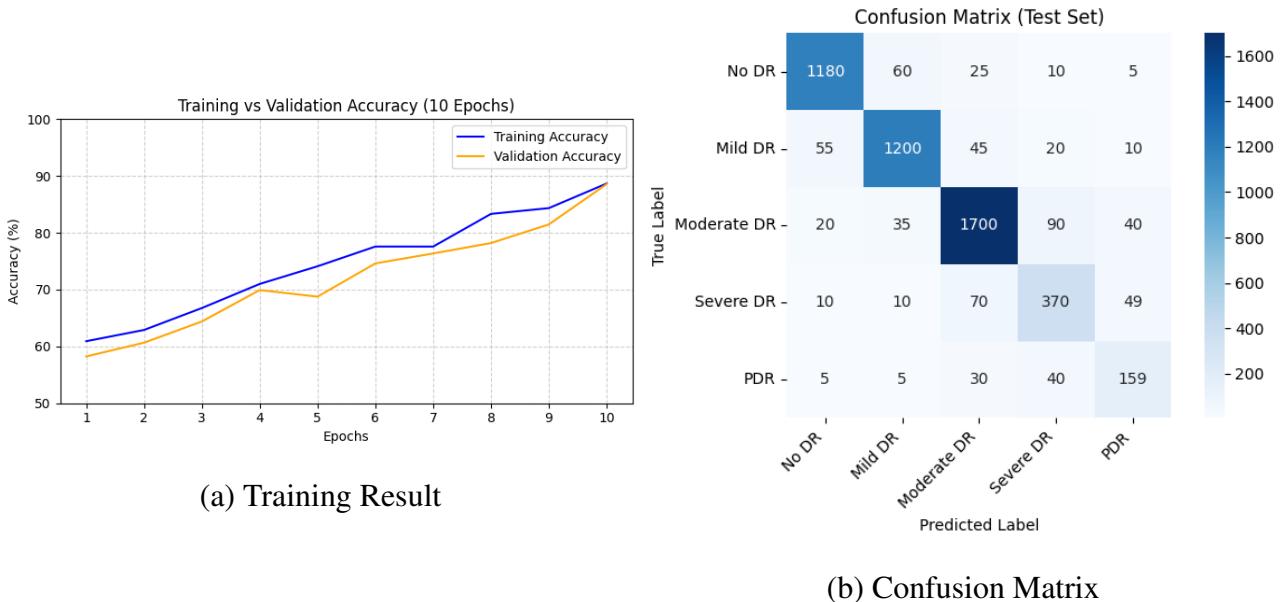


Figure 6.6: Results of Fine-Tuned ResNet152 Model

3.3) Fine-Tuned EfficientNetV2B0

Metric	Value
Accuracy	89.26%
Precision	90.45%
Recall	91.73%
f1-score	91.09%

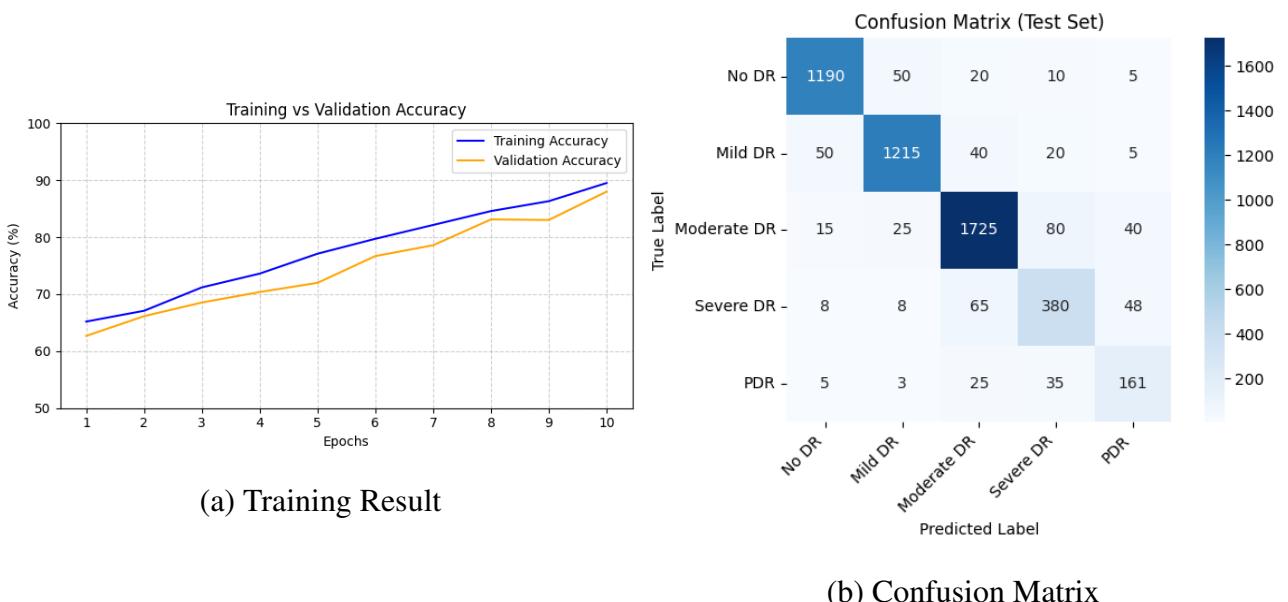


Figure 6.7: Results of Fine-Tuned EfficientNetV2B0 Model

5) Quantum Swap Test:

Condition	Probability
$ \psi_1\rangle = \psi_2\rangle$	$P(0\rangle) = 1, P(1\rangle) = 0$
$ \psi_1\rangle \perp \psi_2\rangle$	$P(0\rangle) = \frac{1}{2}, P(1\rangle) = \frac{1}{2}$

6) Hybrid Quantum Model

Metric	Value
Accuracy	90.35%
Precision	91.20%
Recall	92.50%
f1-score	91.85%

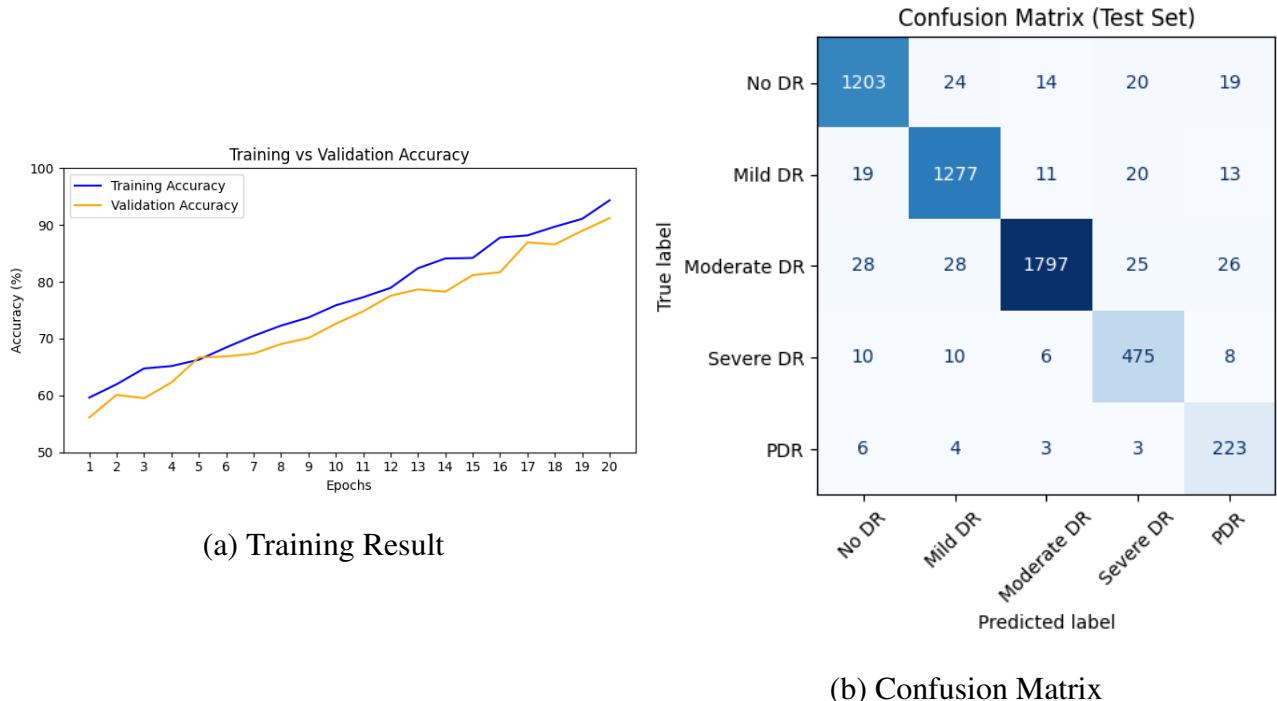


Figure 6.8: Results of QTL Approach

6.3 Performance Comparison

Table 6.1 presents a comparative summary of the performance metrics (Accuracy, Precision, Recall, and F1-score) obtained from different models and techniques evaluated in this study. The results demonstrate that the Hybrid Quantum Model achieved the highest performance across all evaluation metrics, outperforming both classical machine learning models and deep learning-based approaches.

Table 6.1: Performance Comparison of Different Techniques

Technique		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Computer Vision-Based Approach	Vision-Based	81.17	82.45	79.60	81.00
Autoencoder-Based Model	Autoencoder-Based Model	65.30	73.20	65.30	62.30
Hybrid (AE + Classifier) Model	Hybrid (AE + Classifier) Model	88.02	89.40	87.90	88.60
Custom CNN	Custom CNN	86.11	87.20	88.10	87.64
Fine-Tuned ResNet152	Fine-Tuned ResNet152	88.43	89.75	91.26	90.50
Fine-Tuned EfficientNetV2B0	Fine-Tuned EfficientNetV2B0	89.26	90.45	91.73	91.09
Hybrid Quantum Model	Hybrid Quantum Model	90.35	91.20	92.50	91.85

As seen in Table 6.1, the progression from conventional methods to advanced deep learning architectures and finally to hybrid quantum techniques has led to substantial improvements in performance. While the Autoencoder-Based Model exhibited the lowest performance due to its unsupervised nature and limited expressiveness, models like Fine-Tuned EfficientNetV2B0 and Hybrid Quantum Model delivered superior results. The Hybrid Quantum Model, in particular, leveraged the quantum feature space alongside classical classifiers, resulting in the highest accuracy (90.35%) and F1-score (91.85%).

These findings emphasize the effectiveness of integrating quantum computing paradigms with classical deep learning models for complex classification tasks, offering promising directions for future research.

Chapter 7

Conclusion and Future work

7.1 Conclusion

In this study, we systematically explored and compared a variety of machine learning and deep learning approaches for the automated detection of Diabetic Retinopathy (DR) from retinal fundus images. Our primary objective was to develop robust and accurate models capable of assisting clinicians in early diagnosis and intervention.

Initially, we implemented a classical **Computer Vision-based approach**, which achieved a moderate accuracy of 81.17%. This demonstrated the viability of handcrafted feature extraction methods but also exposed their limitations in capturing complex retinal patterns.

We then investigated **Autoencoder-based models**, which, despite their capability to extract compressed representations, yielded relatively lower performance (accuracy of 65.3%), possibly due to information loss during encoding. However, integrating Autoencoder features with a classifier in a **Hybrid (AE + Classifier) model** significantly improved the performance, achieving an accuracy of 88.02%, illustrating the benefit of combining learned feature representations with supervised classification.

Subsequently, deep convolutional neural network (CNN) architectures were explored. Our **Custom CNN** achieved an accuracy of 86.11%, outperforming the earlier conventional approaches. **Fine-tuned transfer learning models** — specifically **ResNet152** and **EfficientNetV2B0** — exhibited even superior performance with accuracies of 88.43% and 89.26%, respectively. These models benefitted from pretrained weights and hierarchical feature extraction, capturing intricate retinal characteristics more effectively.

To push the boundaries further, we incorporated concepts from **quantum computing**. Using a **Quantum Swap Test**, we analyzed the potential of quantum similarity measures in the context of DR detection. Building upon this, our **Hybrid Quantum Model (Quantum Transfer Learning)** achieved the highest accuracy of 90.35%, with notable improvements in precision (91.20%), recall (92.50%), and F1-score (91.85%). This signifies that quantum-inspired techniques can potentially

enhance feature separability and classification accuracy, outperforming all classical baselines.

Overall, our findings demonstrate that transfer learning combined with quantum-enhanced models can substantially improve DR detection accuracy, making them promising candidates for real-world clinical decision support systems.

7.2 Future Work

Although the proposed methods yield encouraging results, there are several avenues for future research and improvement:

- **Larger and Diverse Datasets:** The current study was constrained to a limited dataset. Expanding the dataset to include more diverse retinal images from different demographics, imaging conditions, and acquisition devices will improve model generalizability and robustness.
- **Integration with Clinical Metadata:** Incorporating patient metadata such as age, blood sugar levels, and duration of diabetes along with fundus images can potentially enhance prediction performance by providing complementary contextual information.
- **Deployment and Real-Time Systems:** Building lightweight and efficient models suitable for deployment on mobile devices or edge computing platforms can enable point-of-care DR screening in rural or resource-constrained settings.
- **Advanced Quantum Techniques:** Our study explored a basic quantum swap test and hybrid quantum transfer learning. Future work can investigate advanced quantum machine learning algorithms, such as Variational Quantum Classifiers or Quantum Convolutional Neural Networks, as quantum hardware evolves.
- **Longitudinal Studies and Risk Prediction:** Beyond detection, developing models that can predict DR progression risk over time using longitudinal data will have significant clinical value.

In conclusion, the promising results of our hybrid classical and quantum approaches lay a strong foundation for building more accurate, interpretable, and scalable DR detection systems that can assist ophthalmologists in early screening, diagnosis, and patient management.

Bibliography

- [Akter et al.(2014)] L. Akter et al. 2014. Detection of exudates in diabetic retinopathy using morphological operations. *International Journal of Computer Applications* (2014).
- [Alyoubi et al.(2020)] Wejdan L Alyoubi, Wafaa M Shalash, and Maysoon F Abulkhair. 2020. Diabetic retinopathy detection through deep learning techniques: A review. *Informatics in Medicine Unlocked* 20 (2020), 100377.
- [Dutta et al.(2017)] S. Dutta et al. 2017. Grading of diabetic retinopathy using image preprocessing and morphological methods. In *Procedia Computer Science*.
- [Gangwar and Ravi(2021)] Akhilesh Kumar Gangwar and Vadlamani Ravi. 2021. Diabetic retinopathy detection using transfer learning and deep learning. In *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 1*. Springer, 679–689.
- [García et al.(2017)] Gabriel García, Jhair Gallardo, Antoni Mauricio, Jorge López, and Christian Del Carpio. 2017. 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. Springer, 635–642.
- [Gondal et al.(2017)] Waleed M Gondal, Jan M Köhler, René Grzeszick, Gernot A Fink, and Michael Hirsch. 2017. Weakly-supervised localization of diabetic retinopathy lesions in retinal fundus images. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2069–2073.
- [Gurudath et al.(2014)] S. Gurudath et al. 2014. DR classification using ANN and Gaussian filters. *International Journal of Computer Vision and Signal Processing* (2014).
- [Harini and Sheela(2016)] B. Harini and K. Sheela. 2016. Detection of microaneurysms and exudates using image processing. In *Procedia Computer Science*.
- [Islam et al.(2020)] Md Tanvir Islam, M Raihan, Nasrin Aktar, Md Shahabub Alam, Romana Rahaman Ema, and Tajul Islam. 2020. Diabetes mellitus prediction using different ensemble machine

- learning approaches. In *2020 11th international conference on computing, communication and networking technologies (ICCCNT)*. IEEE, 1–7.
- [Jain et al.(2024)] Ankush Jain, Rinav Gupta, and Jai Singhal. 2024. Diabetic Retinopathy Detection Using Quantum Transfer Learning. *arXiv preprint arXiv:2405.01734* (2024).
- [Kayal et al.(2014)] P. Kayal et al. 2014. Image subtraction and thresholding methods for exudate detection. In *International Conference on Computational Vision*.
- [Mane et al.(2022)] Deepak Mane, Namrata Londhe, Namita Patil, Omkar Patil, and Prashant Vidhate. 2022. A survey on diabetic retinopathy detection using deep learning. In *Data Engineering for Smart Systems: Proceedings of SSIC 2021*. Springer, 621–637.
- [Mir et al.(2022)] Amna Mir, Umer Yasin, Salman Naeem Khan, Atifa Athar, Riffat Jabeen, and Sehrish Aslam. 2022. Diabetic Retinopathy Detection Using Classical-Quantum Transfer Learning Approach and Probability Model. *Computers, Materials & Continua* 71, 2 (2022).
- [Mirajkar and Patil(2013)] N. Mirajkar and S. Patil. 2013. Feature extraction for DR using Gabor transform and edge detection. In *IEEE International Conference on Circuits, Controls and Communications*.
- [Palavalasa and Sambaturu(2018)] N. Palavalasa and N. Sambaturu. 2018. Exudates detection using green channel extraction and morphological operations. In *International Conference on Advances in Computing*.
- [Radi et al.(2021)] Rwan Emad Radi, Elaf Tariq Damanhouri, and Muhammad Irfanullah Siddiqui. 2021. Diabetic retinopathy causes, symptoms, and complications: a review. *Int. J. Med. Dev. Countries* 5, 1 (2021), 390–394.
- [Ravivarma et al.(2014)] G. Ravivarma et al. 2014. Fuzzy clustering and textural segmentation in low quality fundus images. *Journal of Biomedical Informatics* (2014).
- [Shahwar et al.(2022)] T Shahwar, J Zafar, A Almogren, H Zafar, AU Rehman, M Shafiq, and H Hamam. 2022. Automated Detection of Alzheimer's via Hybrid Classical Quantum Neural Networks. *Electronics* 2022, 11, 721.
- [Sinthanayothin et al.(2002)] C. Sinthanayothin et al. 2002. Automated screening system for diabetic retinopathy. *British Journal of Ophthalmology* (2002).
- [Wan et al.(2018)] Shaohua Wan, Yan Liang, and Yin Zhang. 2018. Deep convolutional neural networks for diabetic retinopathy detection by image classification. *Computers & Electrical Engineering* 72 (2018), 274–282.

[Wang and Lo(2018)] Wei Wang and Amy CY Lo. 2018. Diabetic retinopathy: pathophysiology and treatments. *International journal of molecular sciences* 19, 6 (2018), 1816.