

# University of Mohamed Boudiaf - M'sila

FACULTY OF TECHNOLOGY  
DEPARTEMENT OF ELECTRONICS



Serial number.....

Inscription number.....

## Thesis

THESIS SUBMITTED TO THE DEPARTMENT OF ELECTRONICS IN  
CANDIDACY FOR THE DEGREE OF  
**LMD DOCTORATE**

**Field: Electronics**

**Branch: Embedded systems electronics**

## THEME

**Deep learning-based medical data analysis for disease  
prediction and classification**

**Presented By**

**Moustari Mohamed Abderaouf**

**Graduated on: 06/05/2025**

**Members of the judging jury:**

| <u>Name &amp; Surname</u> | <u>Grade</u> | <u>Establishment</u>        | <u>Quality</u> |
|---------------------------|--------------|-----------------------------|----------------|
| SAIGAA Djamel             | Professor    | Univ. Of M'sila             | President      |
| BRIK Youcef               | MCA          | Univ. Of M'sila             | Supervisor     |
| ATTALLAH Bilal            | Professor    | Univ. Of M'sila             | Co-Supervisor  |
| ASBAI Nassim              | MCA          | Univ. Of Bordj Bou Arréridj | Examiner       |
| HAMIDIA Mahfoud           | MCA          | USTHB                       | Examiner       |
| KHENNOUF Salah            | MCA          | Univ. Of M'sila             | Examiner       |
| DJERIOUI Mohamed          | MCA          | Univ. Of M'sila             | Guest          |

**Academic Year: 2024/2025**

To my beloved parents, whose unwavering support, sacrifices, and endless encouragement have been the foundation of my journey. Your love and guidance have shaped me into the person I am today.

To my brothers and family, for their constant belief in me, their encouragement, and for standing by my side through every challenge and triumph.

And to all those who have inspired and supported me along this path, this work is a tribute to your kindness and generosity.

# Acknowledgements

First and foremost, I thank Allah, the Most Merciful, for granting me strength and guidance throughout this journey.

I am deeply grateful to my supervisors, Dr. Brik and Dr. Attallah, for their invaluable support, insightful advice, and encouragement. Their mentorship has been instrumental in shaping this work.

I would like to express my sincere gratitude to the honorable members of the jury before whom I had the privilege of defending my thesis which my supervisors were a part of. My heartfelt thanks go to Professor SAIGAA for presiding over the jury and for their invaluable insights. I am also deeply grateful to Dr. ASBAI, Dr. HAMIDIA, Dr. KHENNOUF, and Dr. DJERIOUI for their time, constructive feedback, and thoughtful questions, which have significantly enriched my work. Thank you all for your support and contributions to this important milestone in my academic journey.

I also extend my sincere appreciation to the professors and researchers who have contributed to my academic growth with their guidance and feedback.

Finally, my heartfelt thanks to everyone who has supported me along this path.

## **Abstract**

The fundus images of patients with Diabetic Retinopathy (DR) often display numerous lesions scattered across the retina. Current methods typically utilize the entire image for network learning, which has limitations since DR abnormalities are usually localized. Training Convolutional Neural Networks (CNNs) on global images can be challenging due to excessive noise. Therefore, it's crucial to enhance the visibility of important regions and focus the recognition system on them to improve accuracy. This thesis investigates two tasks; the first one is a novel two-branch attention-guided convolutional neural network (AG-CNN) with initial image preprocessing for DR classification. The AG-CNN initially establishes overall attention to the entire image with the global branch and then incorporates a local branch to compensate for any lost discriminative cues. The second task is improving diabetic retinopathy classification by combining handcrafted and deep features. We extract LBP, HOG, and GLCM to capture texture patterns and use DenseNet-121 for deep feature extraction. The fusion of these features enables a more comprehensive representation of the retinal images, enhancing the model's ability to discriminate between different severity levels of diabetic retinopathy. We conduct extensive experiments using the APTOS 2019 DR dataset for both tasks.

**Keywords**— Gradient Weighted Class Activation Mapping - Deep Learning - Diabetic Retinopathy Classification - Two-stage System - Feature Extraction - Handcrafted Features - Feature Fusion - Image Pre-processing - Region of Interest Extraction

---

## Résumé

Les images du fond d’œil des patients atteints de rétinopathie diabétique (RD) présentent souvent de nombreuses lésions dispersées sur la rétine. Les méthodes actuelles utilisent généralement l’image entière pour l’apprentissage du réseau, ce qui présente des limites puisque les anomalies de la RD sont généralement localisées. La formation de réseaux neuronaux convolutionnels (CNN) sur des images globales peut être difficile en raison du bruit excessif. Par conséquent, il est crucial d’améliorer la visibilité des régions importantes et de concentrer le système de reconnaissance sur elles pour améliorer la précision.

Cette thèse étudie deux tâches ; la première est un nouveau réseau neuronal convolutionnel guidé par l’attention à deux branches (AG-CNN) avec prétraitement initial de l’image pour la classification de la RD. L’AG-CNN établit d’abord l’attention globale sur l’image entière avec la branche globale, puis intègre une branche locale pour compenser les éventuels indices discriminants perdus.

La deuxième tâche consiste à améliorer la classification de la rétinopathie diabétique en combinant des caractéristiques artisanales et profondes. Nous extrayons LBP, HOG et GLCM pour capturer les motifs de texture et utilisons DenseNet-121 pour l’extraction de caractéristiques profondes. La fusion de ces caractéristiques permet une représentation plus complète des images rétiniennes, améliorant ainsi la capacité du modèle à distinguer les différents niveaux de gravité de la rétinopathie diabétique.

Nous menons des expériences approfondies en utilisant l’ensemble de données AP-TOS 2019 DR pour les deux tâches.

**Keywords**— Cartographie d’activation de classe pondérée par gradient - Apprentissage profond - Classification de la rétinopathie diabétique - Système à deux étapes - Extraction de caractéristiques - Caractéristiques artisanales - Fusion de caractéristiques - Prétraitement d’images - Extraction de régions d’intérêt

## ملخص

غالبًا ما تكشف صور قاع العين لدى مرضى اعتلال الشبكية السكري ( DR ) العديد من الآفات المنتشرة عبر شبكية العين. تستخدم الأساليب الحالية عادةً الصورة بأكملها لتعلم الشبكة، وهو ما له حدود لأن تشوهات اعتلال الشبكية السكري عادة ما تكون موضعية. قد يكون تدريب الشبكات العصبية التلافيفية ( CNNs ) على الصور الشاملة أمرًا صعبًا بسبب الضوضاء المفرطة. لذلك، من الأهمية بمكان تعزيز رؤية المناطق المهمة وتركيز نظام التعرف عليها لتحسين الدقة. تبحث هذه الأطروحة في مهمتين؛ الأولى هي شبكة عصبية تلافيفية موجهة بالانتباه ذات فرعين ( AG-CNN ) جديدة مع معالجة أولية للصورة لتصنيف اعتلال الشبكية السكري. تقوم AG-CNN في البداية بإنشاء انتباه عام للصورة بأكملها بالفرع العام ثم تدجج فرعًا خاصًا للتعويض عن أي إشارات تمييزية مفقودة. المهمة الثانية هي تحسين تصنيف اعتلال الشبكية السكري من خلال الجمع بين الميزات المستخرجة يدويًا والعميقة. نقوم باستخراج LBP و HOG و GLCM لالتقاط أنماط الملمس واستخدام DenseNet-121 لاستخراج الميزات العميقة. يتيح دمج هذه الميزات تمثيلًا أكثر شمولاً لصور الشبكية، مما يعزز قدرة النموذج على التمييز بين مستويات شدة مختلفة من اعتلال الشبكية السكري. تجري تجارب مكثفة باستخدام مجموعة بيانات APTOS

DR 2019 لكلا المهمتين.

### Keywords—

تعيين تنشيط الفئة الموزونة بالتدرج - التعلم العميق - تصنيف اعتلال الشبكية السكري - نظام من مرحلتين - استخراج الميزات - الميزات المصنوعة يدويًا - دمج الميزات - المعالجة المسبقة للصورة - استخراج منطقة الاهتمام

# Table of Contents

|                                                                  |          |
|------------------------------------------------------------------|----------|
| <b>1 Introduction</b>                                            | <b>1</b> |
| 1.1 CNN for diabetic retinopathy detection . . . . .             | 3        |
| 1.1.1 Basic CNN Models . . . . .                                 | 3        |
| 1.1.2 Attention-Driven Models . . . . .                          | 4        |
| 1.1.3 Hybrid Models . . . . .                                    | 5        |
| 1.2 Motivation . . . . .                                         | 5        |
| 1.3 Contribution . . . . .                                       | 6        |
| 1.4 Thesis plan . . . . .                                        | 7        |
| <b>2 Deep learning for medical imaging</b>                       | <b>8</b> |
| 2.1 Introduction . . . . .                                       | 8        |
| 2.2 Machine learning (ML) . . . . .                              | 8        |
| 2.2.1 History . . . . .                                          | 8        |
| 2.2.2 Perceptron . . . . .                                       | 9        |
| 2.2.3 Multilayer perceptron . . . . .                            | 12       |
| 2.2.4 Feed forward networks . . . . .                            | 13       |
| 2.2.5 Transition to deep learning . . . . .                      | 13       |
| 2.2.6 Learning types . . . . .                                   | 14       |
| 2.2.6.1 Supervised learning . . . . .                            | 14       |
| 2.2.6.2 Unsupervised learning: . . . . .                         | 15       |
| 2.2.6.3 Difference: . . . . .                                    | 15       |
| 2.3 Convolutional Neural Networks . . . . .                      | 16       |
| 2.3.1 Definition . . . . .                                       | 16       |
| 2.3.2 The general architecture of a convolutional neural network | 17       |
| 2.3.2.1 Convolution layers . . . . .                             | 17       |

## TABLE OF CONTENTS

---

|          |                                                          |           |
|----------|----------------------------------------------------------|-----------|
| 2.3.2.2  | Pooling layers                                           | 19        |
| 2.3.2.3  | Fully connected layers                                   | 20        |
| 2.3.3    | Types of training                                        | 21        |
| 2.3.3.1  | Training from random initializations                     | 21        |
| 2.3.3.2  | Transfer learning                                        | 22        |
| 2.3.3.3  | Fine Tuning                                              | 22        |
| 2.3.4    | Areas of application of convolutional neural networks in |           |
|          | and medical imaging                                      | 22        |
| 2.3.4.1  | Image Classification                                     | 23        |
| 2.3.4.2  | Object Detection and Localization                        | 28        |
| 2.3.4.3  | Medical Image Segmentation                               | 32        |
| 2.3.4.4  | Image Enhancement and Super-Resolution                   | 36        |
| 2.3.4.5  | Disease Progression Prediction and Prognosis             | 37        |
| 2.4      | Conclusion                                               | 38        |
| <b>3</b> | <b>Deep learning for Diabetic Retinopathy detection</b>  | <b>39</b> |
| 3.1      | Introduction                                             | 39        |
| 3.2      | Generalities on Diabetic Retinopathy                     | 39        |
| 3.3      | Degrees of Diabetic Retinopathy Severity                 | 41        |
| 3.4      | Automatic detection of Diabetic Retinopathy              | 44        |
| 3.5      | Preparing the fundus images                              | 47        |
| 3.5.1    | Acquisition of fundus images                             | 47        |
| 3.5.2    | Fundus image preprocessing techniques                    | 50        |
| 3.6      | Working Fundus images dataset                            | 51        |
| 3.6.1    | Presentation of existing datasets                        | 51        |
| 3.6.1.1  | APTOS 2019                                               | 51        |
| 3.6.1.2  | EyePACS                                                  | 52        |
| 3.6.1.3  | MESSIDOR                                                 | 54        |

## TABLE OF CONTENTS

---

|          |                                                                                               |           |
|----------|-----------------------------------------------------------------------------------------------|-----------|
| 3.6.1.4  | MESSIDOR-2                                                                                    | 55        |
| 3.6.1.5  | IDRiD                                                                                         | 57        |
| 3.6.2    | Choice of working database                                                                    | 58        |
| 3.7      | State of the art of deep learning methods in DR detection                                     | 60        |
| 3.7.1    | Using basic Deep learning models                                                              | 60        |
| 3.7.2    | Using attention guided mechanism                                                              | 64        |
| 3.7.3    | Using deep/handcrafted feature fusion                                                         | 65        |
| 3.8      | Conclusion                                                                                    | 68        |
| <b>4</b> | <b>Attention guided Convolutional Neural Networks for Diabetic Retinopathy classification</b> | <b>69</b> |
| 4.1      | Introduction                                                                                  | 69        |
| 4.2      | Methods                                                                                       | 69        |
| 4.2.1    | DenseNet-121 model                                                                            | 69        |
| 4.2.2    | Laplacian of Gaussian (LoG)                                                                   | 70        |
| 4.2.3    | Contrast Limited Adaptive Histogram Equalization                                              | 72        |
| 4.2.4    | Grad-CAM                                                                                      | 74        |
| 4.3      | Proposed system                                                                               | 76        |
| 4.3.1    | Multilabel setup                                                                              | 77        |
| 4.3.2    | Global and local stages                                                                       | 78        |
| 4.3.3    | Mask inference and ROIs union                                                                 | 78        |
| 4.3.4    | Training strategy of AG-CNN                                                                   | 80        |
| 4.4      | Results and discussion                                                                        | 81        |
| 4.4.1    | Protocols and Setups                                                                          | 82        |
| 4.4.2    | Experiments                                                                                   | 82        |
| 4.4.2.1  | Scenario 1                                                                                    | 83        |
| 4.4.2.2  | Scenario 2                                                                                    | 84        |
| 4.4.2.3  | Scenario 3                                                                                    | 87        |

## TABLE OF CONTENTS

---

|                                                                         |           |
|-------------------------------------------------------------------------|-----------|
| 4.4.2.4 Scenario 4 . . . . .                                            | 87        |
| 4.4.3 Discussion . . . . .                                              | 91        |
| 4.5 Conclusion . . . . .                                                | 94        |
| <b>5 Deep/handcrafted feature fusion for DR classification</b>          | <b>95</b> |
| 5.1 Introduction . . . . .                                              | 95        |
| 5.2 Deep feature extraction . . . . .                                   | 95        |
| 5.3 Handcrafted texture feature extraction . . . . .                    | 96        |
| 5.3.1 Local Binary Pattern . . . . .                                    | 96        |
| 5.3.2 Gray-Level Co-occurrence Matrix . . . . .                         | 98        |
| 5.3.3 Histogram of Oriented Gradients . . . . .                         | 100       |
| 5.4 Proposed method . . . . .                                           | 101       |
| 5.4.1 Preprocessing . . . . .                                           | 102       |
| 5.4.1.1 Contrast Limited Adaptive Histogram Equaliza-<br>tion . . . . . | 102       |
| 5.4.1.2 Unsharp Masking . . . . .                                       | 102       |
| 5.4.2 Deep Feature Extraction . . . . .                                 | 103       |
| 5.4.2.1 DenseNet-121 Architecture . . . . .                             | 103       |
| 5.4.2.2 Global Feature Extraction . . . . .                             | 104       |
| 5.4.3 Handcrafted Feature Extraction . . . . .                          | 104       |
| 5.4.3.1 Local Binary Patterns (LBP) . . . . .                           | 104       |
| 5.4.3.2 Gray-Level Co-occurrence Matrix (GLCM) . . . . .                | 104       |
| 5.4.3.3 Histogram of Oriented Gradients (HOG) . . . . .                 | 105       |
| 5.4.4 Feature Fusion . . . . .                                          | 105       |
| 5.4.4.1 Deep and handcrafted Feature Fusion . . . . .                   | 106       |
| 5.4.4.2 Normalization . . . . .                                         | 106       |
| 5.4.5 Classification . . . . .                                          | 106       |
| 5.4.5.1 Support Vector Machine (SVM) . . . . .                          | 106       |

|          |                                              |            |
|----------|----------------------------------------------|------------|
| 5.4.5.2  | K-Nearest Neighbors (KNN)                    | 107        |
| 5.4.5.3  | Multilayer Perceptron (MLP)                  | 107        |
| 5.4.5.4  | Random Forest                                | 107        |
| 5.4.5.5  | 1D Convolutional Neural Network (1D CNN)     | 108        |
| 5.4.6    | Model Evaluation                             | 108        |
| 5.5      | Results                                      | 108        |
| 5.5.1    | Binary classification                        | 110        |
| 5.5.1.1  | Discussion                                   | 116        |
| 5.5.1.2  | Convolutional Neural Networks (CNNs):        | 116        |
| 5.5.1.3  | Feature Integration                          | 116        |
| 5.5.1.4  | Traditional Machine Learning Models results: | 117        |
| 5.5.2    | Multi-class classification                   | 117        |
| 5.5.2.1  | Discussion                                   | 120        |
| 5.5.2.2  | All-Inclusive Performance                    | 121        |
| 5.5.2.3  | Comparison with Binary Classification        | 121        |
| 5.5.2.4  | KNN Analysis                                 | 121        |
| 5.5.2.5  | SVM Analysis                                 | 122        |
| 5.5.2.6  | MLP Analysis                                 | 122        |
| 5.5.2.7  | Random Forest Analysis                       | 122        |
| 5.5.2.8  | 1D CNN Analysis                              | 123        |
| 5.5.3    | Comparaison with other works                 | 123        |
| 5.6      | Conclusion                                   | 125        |
| <b>6</b> | <b>Conclusion</b>                            | <b>126</b> |
| 6.1      | Thesis summary                               | 126        |
| 6.2      | Extracted contributions                      | 126        |
| 6.3      | Future prospects                             | 128        |

# List of Figures

|      |                                                                                                              |    |
|------|--------------------------------------------------------------------------------------------------------------|----|
| 2.1  | real and artificial model of a neuron                                                                        | 9  |
| 2.2  | mathematical representation of a neuron                                                                      | 10 |
| 2.3  | Different activation functions (left to right: logistic, hyperbolic tangent, absolute value, ReLU functions) | 11 |
| 2.4  | Multi-layer perceptron                                                                                       | 12 |
| 2.5  | Hidden layers in a neural network                                                                            | 13 |
| 2.6  | Data flow through non-linear layers                                                                          | 13 |
| 2.7  | Representation of a CNN                                                                                      | 16 |
| 2.8  | the convolution process                                                                                      | 17 |
| 2.9  | weight sharing                                                                                               | 18 |
| 2.10 | different extracted feature maps                                                                             | 18 |
| 2.11 | 3-dimensional convolution                                                                                    | 19 |
| 2.12 | example of different pooling techniques                                                                      | 20 |
| 2.13 | feature maps flattening                                                                                      | 21 |
| 2.14 | DR classification system example                                                                             | 24 |
| 2.15 | skin lesion classification system example                                                                    | 25 |
| 2.16 | COVID-19 classification system example                                                                       | 26 |
| 2.17 | Histopathology image classification system example                                                           | 27 |
| 2.18 | Tumor Detection in MRI and CT Scans                                                                          | 29 |
| 2.19 | Polyp Detection in Colonoscopy images                                                                        | 30 |
| 2.20 | Bone fracture detection in X-Rays                                                                            | 31 |
| 2.21 | Brain tumor segmentation example                                                                             | 33 |
| 2.22 | Lung and Heart Segmentation example                                                                          | 34 |
| 2.23 | Retinal Blood Vessel Segmentation example                                                                    | 35 |
| 2.24 | Noise reduction of MRI and CT scan images                                                                    | 36 |

## LIST OF FIGURES

---

|                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------|----|
| 2.25 medical image supersampling for increasing resolution . . . . .                                | 37 |
| 3.1 The problems present in diabetic retinopathy affected retina ver-<br>sus a normal one . . . . . | 40 |
| 3.2 Stages of diabetic retinopathy compared with a normal retina . . .                              | 42 |
| 3.3 CNN-based diabetic retinopathy classification framework . . . . .                               | 45 |
| 3.4 Fundoscopy procedure for retinal examination . . . . .                                          | 48 |
| 3.5 Fundus images example of the APTOS 2019 dataset . . . . .                                       | 52 |
| 3.6 Fundus images example of the EyePACS dataset . . . . .                                          | 54 |
| 3.7 Images of MESSIDOR database (a) grade 0 (b) grade 1 (c) grade<br>2 (d) grade 3 . . . . .        | 56 |
| 3.8 Fundus images example of the MESSIDOR-2 dataset . . . . .                                       | 57 |
| 3.9 Fundus images example of the IDRiD dataset . . . . .                                            | 58 |
| 4.1 Illustration of the DenseNet-121 architecture . . . . .                                         | 70 |
| 4.2 Standard LoG kernel . . . . .                                                                   | 71 |
| 4.3 Applying LoG filter: Left: original image, Right: filtered image .                              | 71 |
| 4.4 The effect of the CLAHE method: left: input, right: output. . . . .                             | 73 |
| 4.5 Flowchart of the CLAHE technique . . . . .                                                      | 74 |
| 4.6 Explanatory diagram of the Gradient Weighted Class Activation<br>Mapping (Grad- CAM). . . . .   | 75 |
| 4.7 Extracted heatmaps superposed on their corresponding input images.                              | 76 |
| 4.8 Proposed system structure. . . . .                                                              | 77 |
| 4.9 ROI cropping process using mask inference. . . . .                                              | 79 |
| 4.10 Bounding boxes union in case of multiple ROIs. . . . .                                         | 80 |
| 4.11 Confusion matrix . . . . .                                                                     | 82 |
| 4.12 ROC curve of the best performing branch/threshold iteration of<br>scenario 1. . . . .          | 84 |

## LIST OF FIGURES

---

|      |                                                                                                                                                                              |     |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.13 | Diagram of the new datasets generation.                                                                                                                                      | 85  |
| 4.14 | ROC curves of the best performing branch/threshold in scenario 2 with different preprocessing techniques: a) CLAHE, b) LoG, and c) CLAHE+LoG.                                | 86  |
| 4.15 | ROC curves of the scenario 3 with data balancing and three different prepro- cessing: a) CLAHE, b) LoG, and c) CLAHE+LoG.                                                    | 89  |
| 4.16 | ROC curves of the best-performing threshold in scenario 3 for the local branch: a) CLAHE with $\tau = 0.1$ , b) LoG with $\tau = 0.1$ , and c) CLAHE+LoG with $\tau = 0.2$ . | 90  |
| 4.17 | Diagram of the two augmented dataset generations.                                                                                                                            | 91  |
| 4.18 | ROC curves obtained with augmented preprocessing scenario: a) CLAHE and LoG.                                                                                                 | 91  |
| 5.1  | DenseNet for feature extraction.                                                                                                                                             | 96  |
| 5.2  | The Local Binary Pattern method.                                                                                                                                             | 98  |
| 5.3  | Unsharp masking method                                                                                                                                                       | 103 |
| 5.4  | Deep/Handcrafted feature extraction, fusion and classification system.                                                                                                       | 109 |
| 5.5  | ROC curve of SVM classifier (binary classification).                                                                                                                         | 113 |
| 5.6  | ROC curve of MLP classifier (binary classification).                                                                                                                         | 114 |
| 5.7  | ROC curve of KNN classifier (binary classification).                                                                                                                         | 114 |
| 5.8  | ROC curve of Random Forest classifier (binary classification).                                                                                                               | 115 |
| 5.9  | ROC curve of 1D CNN classifier (binary classification).                                                                                                                      | 115 |
| 5.10 | ROC curve of 1D CNN classifier (multiclass classification).                                                                                                                  | 120 |

# List of Tables

|                                                                                                            |     |
|------------------------------------------------------------------------------------------------------------|-----|
| 2.1 Differences between different types of learning . . . . .                                              | 15  |
| 3.1 Class distribution in the APTOS 2019 dataset . . . . .                                                 | 52  |
| 3.2 Class distribution in the EyePACS dataset . . . . .                                                    | 53  |
| 3.3 Class distribution in the MESSIDOR dataset . . . . .                                                   | 55  |
| 3.4 Class distribution in the MESSIDOR-2 dataset . . . . .                                                 | 56  |
| 3.5 Class distribution in the IDRiD dataset . . . . .                                                      | 58  |
| 4.1 DenseNet-121 summary . . . . .                                                                         | 70  |
| 4.2 Evaluation of Multiple Models trained using the APTOS 2019 Dataset . . . . .                           | 83  |
| 4.3 Evaluation metrics for the scenario 1 experiment . . . . .                                             | 84  |
| 4.4 Evaluation metrics for the CLAHE preprocessing process . . . . .                                       | 85  |
| 4.5 Evaluation metrics for the LoG preprocessing approach . . . . .                                        | 85  |
| 4.6 Evaluation metrics for the CLAHE+LoG preprocessing approach . . . . .                                  | 86  |
| 4.7 CLAHE preprocessing effect with data balancing . . . . .                                               | 88  |
| 4.8 LoG preprocessing effect with data balancing . . . . .                                                 | 88  |
| 4.9 CLAHE+LoG preprocessing effect with data balancing . . . . .                                           | 88  |
| 4.10 Results using augmented method with CLAHE preprocessing . . . . .                                     | 89  |
| 4.11 Results using augmented method with LoG preprocessing . . . . .                                       | 90  |
| 4.12 Scenario results summarized . . . . .                                                                 | 93  |
| 4.13 Comparison with the state-of-the-art methods . . . . .                                                | 94  |
| 5.1 performance of the SVM classifier with different feature combination (binary classification) . . . . . | 111 |
| 5.2 performance of the MLP classifier with different feature combination (binary classification) . . . . . | 111 |

## LIST OF TABLES

---

|      |                                                                                                            |     |
|------|------------------------------------------------------------------------------------------------------------|-----|
| 5.3  | performance of the KNN classifier with different feature combination (binary classification)               | 112 |
| 5.4  | performance of the Random Forest classifier with different feature combination (binary classification)     | 112 |
| 5.5  | performance of the 1D CNN classifier with different feature combination (binary classification)            | 113 |
| 5.6  | performance of the SVM classifier with different feature combination (multiclass classification)           | 118 |
| 5.7  | performance of the KNN classifier with different feature combination (multiclass classification)           | 118 |
| 5.8  | performance of the MLP classifier with different feature combination (multiclass classification)           | 119 |
| 5.9  | performance of the Random Forest classifier with different feature combination (multiclass classification) | 119 |
| 5.10 | performance of the 1D CNN classifier with different feature combination (multiclass classification)        | 120 |
| 5.11 | Comparison with the other works                                                                            | 125 |

# List of Abbreviations

|                 |       |                                                  |
|-----------------|-------|--------------------------------------------------|
| <b>2D</b>       | ..... | Two Dimensional                                  |
| <b>AG-CNN</b>   | ..... | Attention Guided Convolutional Neural Network    |
| <b>AI</b>       | ..... | Artificial Intelligence                          |
| <b>APTOS</b>    | ..... | Asia Pacific Tele- Ophthalmology Society         |
| <b>AUC</b>      | ..... | Area Under Curve                                 |
| <b>CLAHE</b>    | ..... | Contrast Limited Adaptive Histogram Equalization |
| <b>CNN</b>      | ..... | Convolutional Neural Network                     |
| <b>CSAM</b>     | ..... | Channel-Spatial Attention Mechanism              |
| <b>CT</b>       | ..... | Computed tomography                              |
| <b>DL</b>       | ..... | Deep Learning                                    |
| <b>DM</b>       | ..... | Diabetes Mellitus                                |
| <b>DR</b>       | ..... | Diabetic Retinopathy                             |
| <b>ELM</b>      | ..... | Extreme Learning Machine                         |
| <b>FC</b>       | ..... | Fully Connected                                  |
| <b>GAN</b>      | ..... | Generative Adversal Networks                     |
| <b>Grad-CAM</b> | ....  | Gradient-Weighted Class Activation Mapping       |
| <b>GLCM</b>     | ..... | Gray-Level Co-occurrence Matrix                  |
| <b>HOG</b>      | ..... | Histogram of Oriented Gradients                  |
| <b>KNN</b>      | ..... | K-Nearest Neighbors                              |

## LIST OF TABLES

---

|                     |                                        |
|---------------------|----------------------------------------|
| <b>MAS</b> .....    | Multi-Agent System                     |
| <b>NPDR</b> .....   | Non-Proliferative Diabetic Retinopathy |
| <b>PDR</b> .....    | Proliferative Diabetic Retinopathy     |
| <b>ResNet</b> ..... | Residual Neural Network                |
| <b>VGG</b> .....    | Visual Geometry Group                  |
| <b>ML</b> .....     | Machine Learning                       |
| <b>ReLU</b> .....   | Rectified Linear Unit                  |
| <b>SLP</b> .....    | Single Layer Perceptron                |
| <b>MLP</b> .....    | Multi-Layer Perceptron                 |
| <b>RGB</b> .....    | Red, Green and Blue                    |
| <b>MRI</b> .....    | Magnetic Resonance Imaging             |
| <b>LoG</b> .....    | Laplacian of Gaussian                  |
| <b>ROC</b> .....    | Receiver-operating characteristic      |
| <b>SGD</b> .....    | Stochastic Gradient Descent            |
| <b>NLB</b> .....    | Non-Local Blocks                       |
| <b>ROI</b> .....    | Region Of Interest                     |
| <b>LBP</b> .....    | Local Binary Patterns                  |
| <b>SVM</b> .....    | Support Vector Machine                 |

## List of notations

|                   |                                     |
|-------------------|-------------------------------------|
| <b>X</b> .....    | Inputs                              |
| <b>Y</b> .....    | Outputs                             |
| <b>W</b> .....    | Weights                             |
| <b>S</b> .....    | Activation function                 |
| <b>F(S)</b> ..... | Output, same as Y                   |
| <b>B</b> .....    | Bias                                |
| <b>R</b> .....    | Size of resulting convolution block |
| <b>H</b> .....    | Image height                        |
| <b>W</b> .....    | Image width                         |
| <b>M</b> .....    | Kernel height                       |
| <b>N</b> .....    | Kernel width                        |
| <b>T</b> .....    | Convolution result                  |
| <b>X</b> .....    | Image region matrix                 |
| <b>W</b> .....    | Synaptic weights matrix             |

# 1 | Introduction

Diseases are disorders in the normal physiological processes that affect living organisms and cause impaired performance and health. Diseases range from acute conditions, such as infections, to chronic disorders, such as cardiovascular diseases and cancer. Diseases may be brought about by one category or another, namely: bacterial, parasitic, viral, or fungal infection; genetic disorders owing to hereditary influences or mutations; or influenced by environmental factors such as poor nutrition, lack of exercise, or exposure to toxic substances. Whatever the cause, however, diseases generally result from a combination of genetic susceptibility, environmental factors, and immune system response.

Metabolic disorders are a major concern in chronic diseases, especially in contemporary societies whose lifestyle and dietary habits have changed significantly. One notable example is diabetes mellitus, a disorder that impairs the body's ability to control blood sugar (glucose) levels, leading to persistent high blood glucose levels, a state medically referred to as hyperglycemia. Diabetes is a complex disease that mostly comes in two forms: type 1 diabetes, which results from an autoimmune destruction of the insulin-producing cells in the pancreas, and type 2 diabetes, which is characterized by insulin resistance or too little insulin production in the body. Gestational diabetes is another form of the disease that can develop during the course of pregnancy. Globally, the prevalence of diabetes has surged in recent decades, driven by rising obesity rates and sedentary lifestyles, becoming a major public health concern [1].

Unmanaged diabetes can lead to a number of complications that involve different organs and systems. Among the plethora of complications are diabetic retinopathy, which is one of the leading causes of blindness if left unattended.

The condition results from long-standing high blood sugar levels damaging the blood vessels in the retina, the light-sensitive tissue lining the back of the eye, which sends visual signals to the brain. Diabetic retinopathy is thus thought of as a microvascular complication of diabetes, since it affects the microvasculature, or very small blood vessels [2].

Onset of diabetic retinopathy is gradual, often with four definable stages of progression.

The first stage is called NPDR (Non-Proliferative Diabetic Retinopathy), in which the small blood vessels within the retina begin to weaken and leak blood and other fluids. As the disease worsens, blood vessels serving the retina may become completely blocked, causing a section of the retina to receive less oxygen than required.

In the more advanced stage, PDR (Proliferative Diabetic Retinopathy), new abnormal blood vessels form as a way to replace oxygen loss through the process called neovascularization. These vessels are weak and leaky, can rupture, producing more pronounced vision loss, and may even lead to retinal detachment in the case of the formation of scar tissue.

The complications of diabetic retinopathy can be huge, affecting a patient's quality of life in a big way. There are blurred visions, dark spots, or floaters in the visual field, which may even lead to total loss of vision. Diabetic retinopathy remains a major cause of blindness among adults aged 20-74 years [3]. Despite the fact that diabetic retinopathy is a major and potentially sight-threatening disease, its timely detection and appropriate treatment could reduce the risk of visual loss significantly. Regular eye examinations among those suffering from diabetes is important with respect to early detection of changes in the retina. Treatment modalities like laser photocoagulation and intravitreal injections of anti-vascular

endothelial growth factor medications may be instituted in an attempt to manage the disease process and preserve vision; vitreous surgery is indicated in selected cases [4]. In a nutshell, diabetes is a common chronic disease with serious health consequences; its complications, like diabetic retinopathy, raise the need for thorough management of the disease. Reduction in diabetes risk factors, enabling timely diagnosis and intervention in its complications, is one of the fundamental strategies for reducing the burden of this global health problem [5].

## 1.1 CNN for diabetic retinopathy detection

Diabetic retinopathy (DR) classification using deep learning has made significant strides in recent years. The three primary approaches—basic CNN models, attention-guided networks, and hybrid models combining CNNs with handcrafted features—have each contributed unique advantages to the field. While challenges exist, recent trends have focused on improving accuracy, interpretability, and clinical applicability.

### 1.1.1 Basic CNN Models

Basic CNNs such as Inception, ResNet, and VGG have been wildly effective in the automated classification of diabetic retinopathy from retinal images. This architecture allows for hierarchical feature extraction directly from the unprocessed images, eliminating traditional, hand-engineered feature extraction processes. Initially, CNN-based methods achieved remarkable performances, particularly when using large datasets such as EyePACS and Messidor that provide millions of annotated retinal images. However, enhancements to these models have focused on tackling issues such as overfitting and data imbalance. Transfer learning, where pre-trained models on large datasets are fine-tuned for DR classification, has been

a popular approach to overcome limited dataset sizes. Data augmentation techniques, such as image rotation, scaling, and noise injection, are also employed to create synthetic diversity in the training data, further improving model robustness.

These have helped the CNN models generalize much better, reduce overfitting, and perform better on unseen data.

### 1.1.2 Attention-Driven Models

Attention-guided models are a significant leap in DR classification, as it instructs the model to focus on the most relevant areas of the retina, like microaneurysms, hemorrhages, and exudates. This mimics how an ophthalmologist examines the images of the retina and allows the network to give greater weight to the critical regions.

Recent work on attention-driven models has involved the use of “self-attention mechanisms” and transformer frameworks, which have seen phenomenal success in natural language processing. These attention mechanisms allow the model to dynamically evaluate different parts of the image; in essence, enabling it to focus on both fine-grained local features and larger-range contextual cues, hence, improving classification accuracy across stages of diabetic retinopathy.

Another important development is the use of “multiscale attention”, which allows the models to capture DR features at different resolutions. This helps in identifying smaller lesions, indicative of early DR, while bigger features can be processed that point towards the advanced stages. Attention models also help in improving interpretability by generating saliency maps highlighting the regions of the retina that contribute to the model's prediction, thus providing clinicians with a transparent, interpretable system.

### 1.1.3 Hybrid Models

With integrating Convolutional Neural Networks with Handcrafted Features, a hybrid model combining CNN features with handcrafted features has gradually emerged as a successful strategy to improve DR classification. Besides the CNN-learned features, some handcrafted features, including morphology of blood vessels, exudate size, and texture analysis, may provide complementary information. For instance, blood vessel tortuosity and fractal dimensions are handcrafted features essential for recognizing the advanced stages of DR. Recent models have explored “feature fusion”, combining CNN-learned deep features with domain-specific handcrafted features. This fusion approach has led to improved performance, particularly on imbalanced datasets, where advanced stages of DR are underrepresented. By incorporating both types of features, these models achieve better sensitivity and specificity, especially in distinguishing between different DR stages.

Furthermore, the use of handcrafted features improves model explainability. Since these features are already well-known markers of DR progression, they make the model’s predictions more interpretable to clinicians. This increased transparency is crucial for gaining trust in AI-driven systems in clinical practice.

## 1.2 Motivation

Diabetic retinopathy (DR) is a leading cause of vision loss, requiring early detection and accurate staging for effective treatment. Manual analysis of retinal fundus images is time-consuming, subjective, and unsuitable for large-scale screening. Deep learning systems, particularly CNNs, offer a scalable and efficient solution by learning intricate patterns in retinal images to detect and clas-

sify DR stages accurately. These models enhance diagnostic reliability, support clinicians, and prioritize patients needing urgent care. By integrating with telemedicine, they can improve access to quality eye care, especially in resource-limited areas, helping to reduce the global burden of diabetic blindness.

### 1.3 Contribution

This research offers a significant contribution to diabetic retinopathy (DR) detection and classification work by tackling two important aspects that enhance model performance and interpretability. Firstly, we employed Grad-CAM to develop the performance of deep learning models through visualizing and localizing specific areas in retinal fundus images that affect predictions. This approach gives clinicians better transparency and more trust in the decision-making of the model. For the second case, we combined hand-crafted features like Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gray Level Co-occurrence Matrix (GLCM) with deep features obtained using convolutional neural networks (CNNs). The hybrid method makes use of complementary strengths of both texture-based and deep learning representations such that the model can effectively identify both local and global patterns in retinal images. As a result, we achieved improved classification performance, particularly in the discrimination of subtle DR stages. Not only does our method improve accuracy, but it also improves robustness and reliability, which makes it a valuable tool for automatic DR diagnosis.

## **1.4 Thesis plan**

In order to well present our work, this thesis is structured as follows, chapter 2 presents a general presentation of machine learning and AI for image analysis in the medical field, then, chapter 3 introduces Deep learning for DR detection, after that, chapter 4 presents the attention guided approach for DR classification and chapter 5 presents deep/handcrafted feature fusion for DR classification, and we end is with a conclusion.

## 2 | Deep learning for medical imaging

### 2.1 Introduction

Deep learning has transformed medical image analysis by enabling accurate, automated interpretation of complex patterns in images. This chapter introduces key deep learning concepts and their applications in healthcare, emphasizing advancements in diagnostic accuracy.

### 2.2 Machine learning (ML)

ML develops algorithms that enable computers to learn, closely tied to AI. Initially aimed at AI, it shifted to task-specific solutions due to limitations. Today, ML optimizes functions through trial and error, excelling in many areas but facing challenges in precision-critical fields like language modeling and computer vision. [6].

#### 2.2.1 History

ML, a branch of AI, enables machines to learn from data without explicit programming. AI emerged in the 1950s, with Alan Turing predicting its rise. ML gained traction in the 1970s-80s, while neural networks, first conceived in 1943, resurfaced in the late 80s, laying the foundation for modern ML [7] [8]. Neural networks resurfaced in the late 80s with Backpropagation but faded due to limited computing power. They made a strong comeback in the mid-2000s with Deep Learning [9].

### 2.2.2 Perceptron

The perceptron, shown in Figure 2.1, is the simplest artificial neural network, inspired by biological neurons. A neuron transmits information via synapses in the dendrites [10], while an artificial neuron uses inputs (X) to produce an output (Y). Excitation or inhibition drives this process, akin to the activation function [9], which transforms data and applies thresholding to determine the output [11].

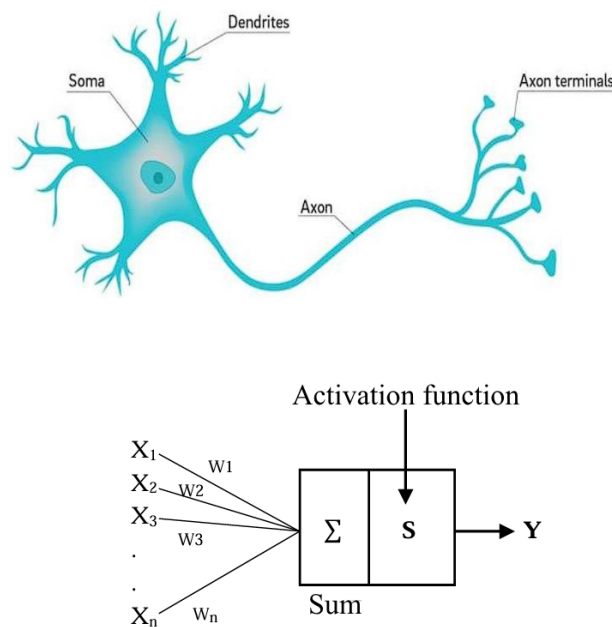


Figure 2.1: real and artificial model of a neuron

Looking carefully at figure 2.2 on the artificial neuron we have a linear combination that operates as follows:

$$S = X_1 \bullet W_1 + X_2 \bullet W_2 + X_3 \bullet W_3 + ..... + X_n \bullet W_n + B \quad (1)$$

$$Y = F(S) \quad (2)$$

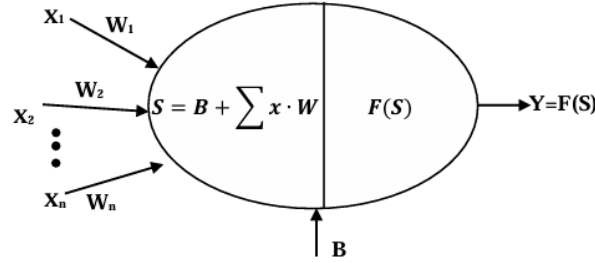


Figure 2.2: mathematical representation of a neuron

From there begins the basic learning principle thanks to the synaptic weights  $W_n$  and the bias  $B$  entering the aggregation phase (Equation (1)) and the activation phase (Threshold).

Each of these two influences the behavior of the perceptron and therefore the output model [12], either by an offset by bias or change of magnitude or even the sign by the synaptic weights, given that we have a kind of linearity that looks like a straight line

$$F(X) = aX + B \quad (3)$$

This influence on the behavior of the model to arrive at the desired output is done by an update of the parameters (Weights update), which is why the parameters  $W_n$  and  $B$  are parameters that enter into the training of the machine or more precisely the learning algorithm which follows the following equation:

$$W = W + \alpha(Y_{ref} - Y)X \quad (4)$$

Through this kind of update, a kind of automatic adjustment is made via this algorithm that allows to reinforce the parameters of the synaptic weights each time an input  $X$  is activated at the same time as the output  $Y$  present in this data

[8, 9].

That said, this remains very limited in terms of what the domain of the neural network must provide, which is why the activation functions help to have a model that enters into non-linearity, we have for example very well-known functions such as the sigmoid function for example defined such as:

$$F(S) = \frac{1}{1 + e^{-S}} \quad (5)$$

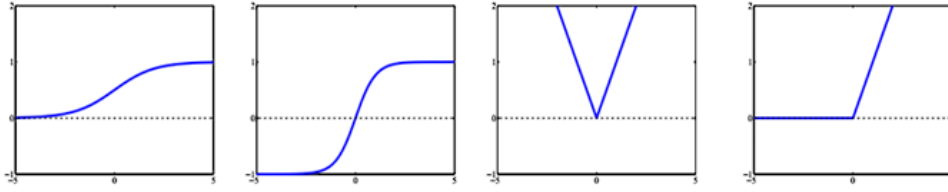


Figure 2.3: Different activation functions (left to right: logistic, hyperbolic tangent, absolute value, ReLU functions)

According to figure 2.3, we can cite the functions starting from the left, the logistic function (equation 5) (Sigmoid), the hyperbolic tangent function (equation 6), the absolute value function (equation 7), and the Rectified Linear Unit (ReLU) function that we will have the opportunity to see in the next chapter which will be used in our work (equation 8).

$$\text{Tanh}(S) = \frac{1 - e^{+2(S)}}{1 + e^{-2(S)}} \quad (6)$$

$$\text{abs}(S) = |S| \quad (7)$$

$$F = \max(0, S) \quad (8)$$

With of course,  $S$  which represents the (weighted sum) that is to say the sum

of the products between the matrix of synaptic weights  $W$  and the network inputs by also adding the bias vector  $B$ .

We use the matrices to simplify the mathematical calculations.

### 2.2.3 Multilayer perceptron

The perceptron's linear constraint, seen in the XOR problem's lack of linear separability [13], limits its use for non-linear phenomena. This issue is resolved by the Multi-Layer Perceptron (MLP), where multiple connected neurons handle complex tasks. A single-layer perceptron (SLP) can include multiple perceptrons to classify different classes by linking inputs to outputs, each with its own bias  $B$  [14]. Figure 2.4 contrasts an SLP, which may have one or several perceptrons in a layer [14], with an MLP, which features multiple layers for enhanced learning. In order to better understand the multi-layer perceptron network, Figure

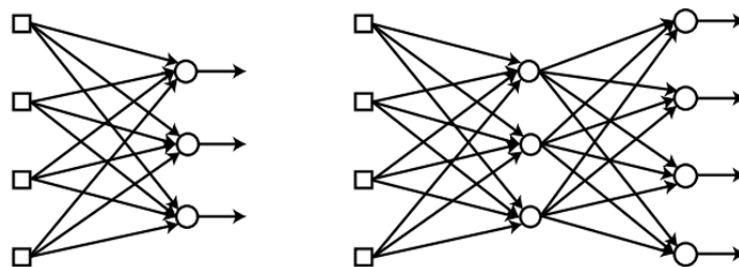


Figure 2.4: Multi-layer perceptron

2.5 illustrates what happens when the data passes through several layers that are processed in a non-linear manner based on the features extracted from the data by the learning algorithm as they approach from one layer to another towards the output which becomes linear [14]. From this idea, we can now understand what the depth of a neural network is.

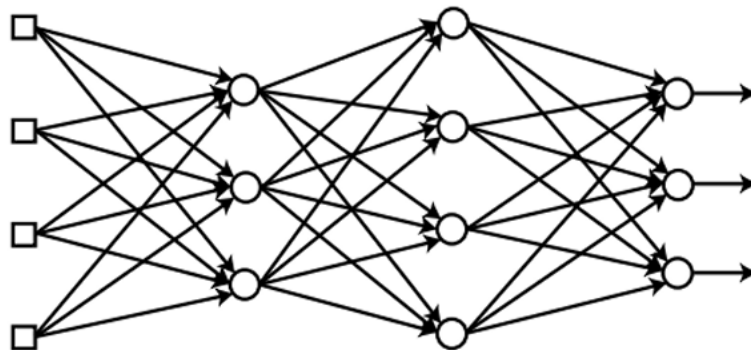


Figure 2.5: Hidden layers in a neural network

### 2.2.4 Feed forward networks

By adding hidden layers between input and output, as shown in Figure 2.6, each neuron receives inputs from the previous layer, forming a feedforward network without feedback [15]. Data flows from input to output through hidden layers. Increasing hidden layers leads to Deep Learning, where deeper architectures enhance data feature extraction [16].

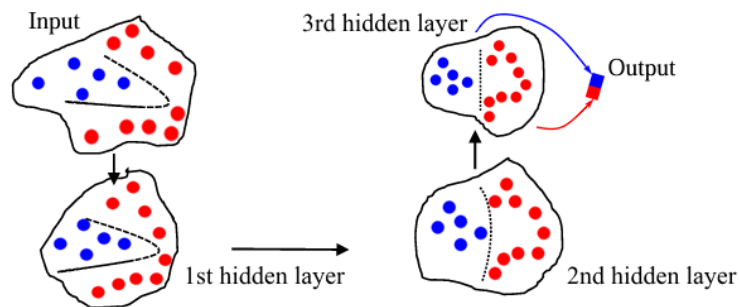


Figure 2.6: Data flow through non-linear layers

### 2.2.5 Transition to deep learning

Deep Learning or DL is a branch of Machine Learning entirely based on artificial neural networks [17]. The concept of deep learning (DL) existed for

years but was initially abandoned due to limited resources. In the mid 2000s, Machine Learning dominated visual recognition competitions until 2012, when DeepMind's deep learning algorithm outperformed all rivals, prompting a shift toward DL. Its success stems from advancements in computing power, storage, and the availability of big data, making DL viable where it was not in the 90s. Pioneers like Geoffrey Hinton, Yoshua Bengio (creators of GANs), and Yann LeCun (developer of CNNs and the LeNet architecture) spearheaded DL's breakthroughs, revolutionizing fields like image recognition. Geoffrey Hinton proved that deep learning could solve problems that were insoluble by other approaches [12].

### 2.2.6 Learning types

Deep learning, a subset of machine learning, trains models to predict or detect patterns from large datasets using supervised and unsupervised learning approaches.

#### 2.2.6.1 Supervised learning

Supervised learning trains models on labeled data to map inputs to outputs and make predictions. It includes classification and regression tasks.

- **Classification** Classification involves predicting categorical outputs, either binary (e.g., 0 or 1) or multiclass (e.g., cat, dog, or bird). The model assigns the correct label to each input.
- **Regression** Regression predicts continuous outputs, aiming for values close to the actual ones. Applications include predicting house or stock prices, with performance measured by the error between predicted and actual values.

#### 2.2.6.2 Unsupervised learning:

Unsupervised learning analyzes unlabeled data to uncover patterns or structure. It is useful for exploring data without predefined labels and is mainly used for clustering and dimensionality reduction.

- **Clustering:** Clustering is an unsupervised learning task that groups data based on similarities, without predefined classes. It divides data into clusters with similar attributes, useful for tasks like customer segmentation based on behavior or image segmentation by pixel similarity.

#### 2.2.6.3 Difference:

The main difference between supervised and unsupervised learning is that supervised learning uses labeled data to map inputs to outputs, while unsupervised learning works with unlabeled data to uncover hidden structures. Other differences include:

Table 2.1: Differences between different types of learning

| Type of learning  | Supervised learning               | Unsupervised learning                |
|-------------------|-----------------------------------|--------------------------------------|
| Input data        | Uses known, labeled data as input | Uses unknown data as input Unlabeled |
| Number of classes | Known                             | Unknown                              |
| Result precision  | Accurate and reliable results     | Moderate accuracy and reliability    |

## 2.3 Convolutional Neural Networks

### 2.3.1 Definition

Convolutional Neural Networks (CNNs), example shown in figure 2.7, introduced in the 1980s by Kunihiko Fukushima [18], are designed for tasks involving spatial hierarchies, making them ideal for image processing, classification, and object detection. CNNs use filters or kernels to extract features from images by sliding across them and learning patterns during training, such as edges or textures [19].

The architecture consists of a convolutional section that extracts features with filters and pooling, and a classification section that uses fully connected layers and softmax to make predictions. This structure enables CNNs to learn complex patterns and achieve high accuracy in areas like medical imaging and face recognition.

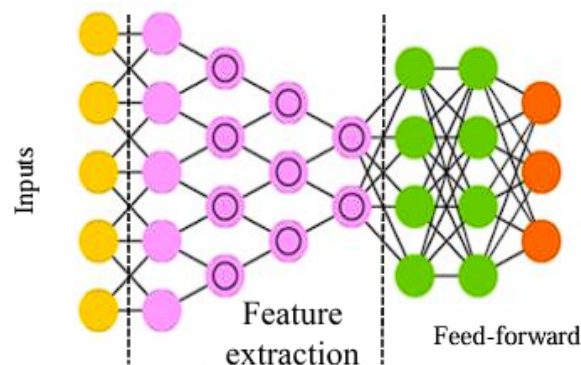


Figure 2.7: Representation of a CNN

## 2.3.2 The general architecture of a convolutional neural network

### 2.3.2.1 Convolution layers

To reduce parameters, neighboring pixels are assumed to contain similar information compared to distant ones [20]. This leads to the use of convolution layers with filters (kernels) of sizes like  $2 \times 2$ ,  $3 \times 3$ ,  $5 \times 5$ , etc., applied to the input image before passing through a feed-forward network. The convolution with the input image block (receptive field) creates a feature map, resulting from scanning the image with the kernel across all pixel blocks. For an input image of size  $(H \times W)$  and a kernel of size  $(M \times N)$  [20], the size of the resulting block R is:

$$R = (H - M + 1) \times (W - N + 1) \quad (9)$$

In Figure 2.8, an input image of size  $7 \times 7$  convolved with a  $3 \times 3$  filter results in an output layer of size  $5 \times 5$ . In practice, the scan involves multiple filters with shared synaptic weights, known as weight sharing. This technique reduces the number of parameters and helps prevent overfitting [16].

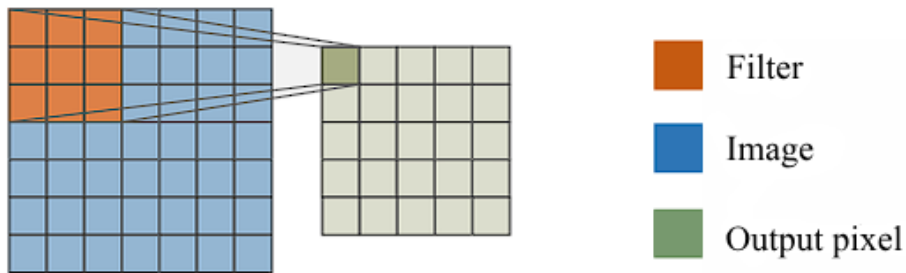


Figure 2.8: the convolution process

Weight sharing, illustrated in Figure 2.9, shows each neuron in the block receiving filtered information with identical parameters (a, b, c, d) for the entire

block, corresponding to the receptive field of size  $5 \times 5$ . These parameters differ for other neuron blocks (m, n, o, p), meaning each block represents a single filter that scans and filters the input image.

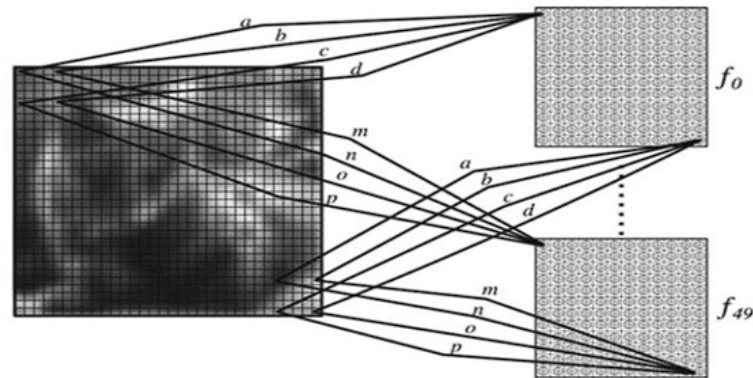


Figure 2.9: weight sharing

What we've seen is a single feature extraction. As shown in the previous figure, weight sharing ensures specific feature extraction. With different filter parameters, multiple features can be extracted [16], as illustrated in Figure 2.10.

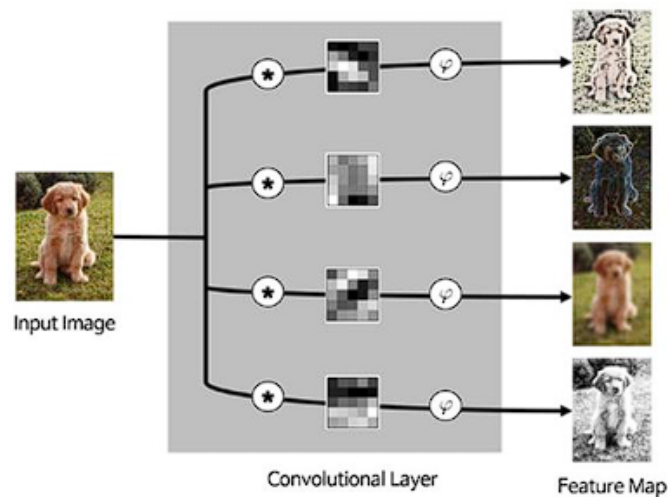


Figure 2.10: different extracted feature maps

Using multiple convolutional filters increases the model's ability to learn and

predict data. Each resulting feature map passes through an activation function, as shown earlier. The applied filters generate several output channels, and through the activation function ( $\varphi$ ), we obtain different image features [8]. Input layers must be 3D, as images can be in RGB, requiring a third dimension for filters instead of using three filters for each color channel (R, G, B), as shown in Figure 2.11.

The input image dimension is  $W \times H \times C$ , where C represents the color channels (R, G, B), or  $W \times H \times 1$  for grayscale images. The convolution operation is:

$$T = X * W + B \quad (10)$$

Where X is the matrix of a region of the image; W the matrix of synaptic weights; B the bias vector.

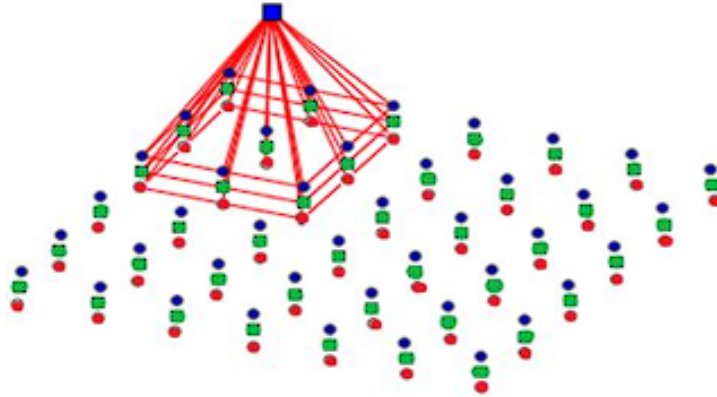


Figure 2.11: 3-dimensional convolution

### 2.3.2.2 Pooling layers

For images of size  $256 \times 256$  convolved with 64 filters of size  $9 \times 9$ , we obtain 64 features of size  $248 \times 248$ , resulting in 3,936,256 parameters. To reduce this complexity, pooling is used. Pooling subsamples the convolution layer's output

blocks by either averaging or taking the max value, depending on the pooling size, as shown in Figure 2.12.

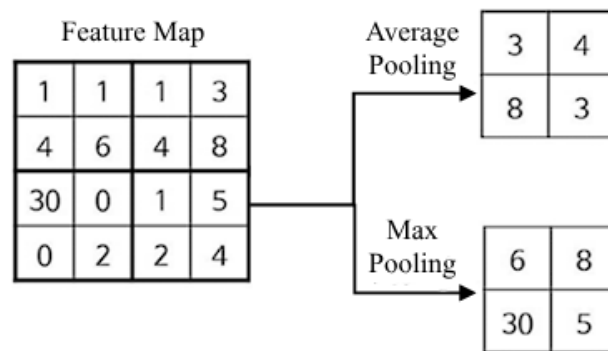


Figure 2.12: example of different pooling techniques

By combining convolution and pooling layers, multiple layers can be stacked while preserving image features. To maintain classification performance, it is best to start with a large first layer, then gradually shrink the layers as the data progresses through them [20].

### 2.3.2.3 Fully connected layers

In the fully connected operation of a neural network, the input representation is flattened into a feature vector and passed through a neural network to predict the output probabilities. figure 2.13 describes the flattening operation.

The lines are concatenated into a long vector, and if multiple input layers exist, they form an even longer feature vector. This vector is then passed through several dense (fully connected) layers, where it undergoes the same operations as in a conventional neural network. The fully connected layer maps the representation between input and output.

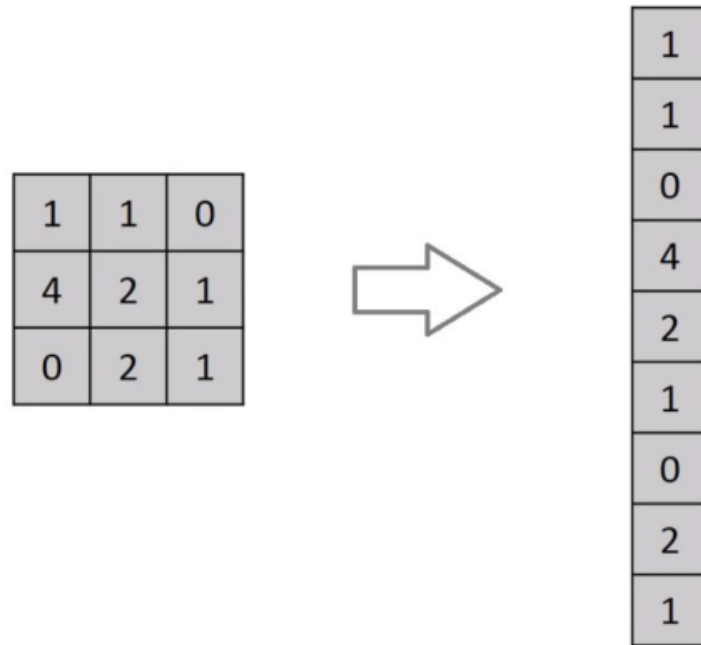


Figure 2.13: feature maps flattening

### 2.3.3 Types of training

Training neural networks can be done in a variety of ways depending on the amount of available data, computational resources, and goal to be achieved. There exist three major approaches for training neural networks: pure training, transfer learning, and fine tuning.

#### 2.3.3.1 Training from random initializations

When training from scratch, all weights are initialized randomly. The model learns by adjusting these weights through multiple iterations, starting with input data and calculating a loss function to measure the difference between predicted and true labels. Optimization methods like gradient descent are used to minimize the loss. This approach works well with large, diverse datasets, allowing the

model to learn intricate patterns. However, it requires significant computational power and may struggle with smaller datasets, as it risks overfitting by memorizing the training data rather than generalizing.

### **2.3.3.2 Transfer learning**

Transfer learning is used when data is limited or training from scratch is too costly. It uses a pre-trained model, such as one trained on ImageNet, which has learned general features like edges and textures. The lower layers, which capture basic patterns, are usually frozen, while the final layers are replaced and trained on the new task. This method is faster and requires less data than training from scratch, and it works best when the new task is related to the original one.

### **2.3.3.3 Fine Tuning**

Fine-tuning builds on transfer learning by offering more flexibility in adapting a pre-trained model. Initially, new layers are trained, like in transfer learning. Then, selected pre-trained layers are unfrozen and further trained on the new dataset, allowing the model to adjust its pre-learned features for the new task. Fine-tuning is especially useful when the new task differs significantly from the original. However, it requires careful monitoring of the learning rate to prevent extreme changes that could degrade performance. When done well, fine-tuning enhances a pre-trained model's performance without risking overfitting.

## **2.3.4 Areas of application of convolutional neural networks in and medical imaging**

Convolutional Neural Networks (CNNs) have become an essential tool in medical imaging due to their ability to extract deep hierarchical features from images.

Their applications span multiple medical domains, from disease classification and detection to segmentation, super-resolution, and disease progression prediction. This document explores the significant applications of CNNs in medical imaging, explaining each use case with real-world examples and technical insights into CNN architectures and methodologies.

### 2.3.4.1 Image Classification

Image classification is one of the fundamental applications of CNNs in medical imaging. In this task, a CNN assigns a medical image to one of several predefined categories based on learned patterns. Traditional classification methods relied on handcrafted features, but CNNs automatically extract relevant features, improving accuracy and reducing the need for manual intervention. CNNs typically consist of convolutional layers, pooling layers, and fully connected layers, allowing them to learn hierarchical representations of medical images.

- **Diabetic Retinopathy Classification:** Diabetic retinopathy (DR) is one of the leading causes of blindness worldwide, particularly among individuals with diabetes, making its early detection and timely treatment essential for preventing vision loss. CNNs have become invaluable tools in automating the classification of DR stages by analyzing fundus images, which capture the condition of the retina. Datasets such as APTOS 2019 and EyePACS are frequently used to train these models, providing a diverse range of retinal images for effective model training. These CNNs have demonstrated high accuracy in distinguishing between different stages of DR, enabling healthcare professionals to assess the severity of the disease and plan appropriate interventions. Transfer learning further enhances the performance of DR classification by leveraging pre-trained models, such as DenseNet, ResNet, and Inception, which have already learned useful features from large, gen-

eral image datasets. By fine-tuning these models on DR-specific data, the classification performance is significantly improved, allowing for more accurate detection and early intervention [21].

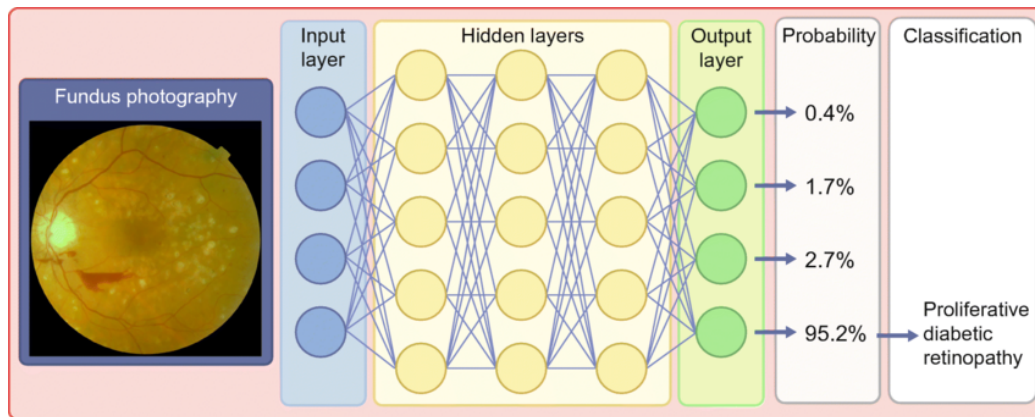


Figure 2.14: DR classification system example

- **Skin Lesion Classification:** Skin lesion classification plays a crucial role in the early detection of skin cancer, helping clinicians differentiate between benign and malignant lesions. CNNs have been extensively used for classifying skin lesions, enabling automated and accurate analysis of dermatological images. The ISIC (International Skin Imaging Collaboration) dataset has been a vital resource for training these models, providing a comprehensive collection of images that include various types of skin lesions, such as melanoma, basal cell carcinoma, and squamous cell carcinoma. With the growing need for reliable and quick diagnosis, CNNs trained on this dataset have shown remarkable performance in identifying and classifying different lesion types. Furthermore, advancements in CNN architectures, including EfficientNet and Vision Transformers (ViTs), have significantly enhanced classification accuracy. EfficientNet, known for its efficiency in balancing model size and performance, and ViTs, which utilize transformer-based architectures to capture global image context, have set new benchmarks in

skin lesion classification [22].

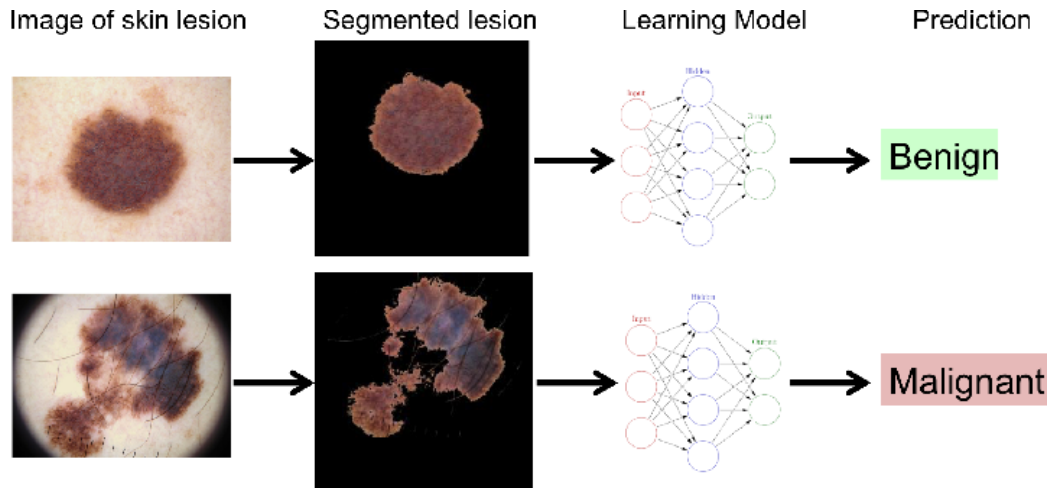


Figure 2.15: skin lesion classification system example

- **Pneumonia and COVID-19 Detection:** Detecting pneumonia and COVID-19 infections through chest X-ray images is critical for timely diagnosis and treatment. CNNs have shown great promise in automating the classification of these conditions, helping healthcare providers identify different types of pneumonia, including bacterial and viral infections like COVID-19. Deep learning models such as VGG16, ResNet, and MobileNetV2 have been widely adopted and fine-tuned for these tasks, leveraging their ability to extract hierarchical features from chest X-rays. These models are trained to distinguish between various categories, including normal, bacterial pneumonia, and viral pneumonia, with a focus on recognizing key patterns associated with COVID-19, such as ground-glass opacities and consolidation. Recent studies have further improved the performance of these models by incorporating attention mechanisms, which enable the model to focus on the most relevant regions of the X-ray, such as areas of infection within the lungs. This integration not only enhances the accuracy of the models but

also improves their interpretability, allowing clinicians to better understand how the model arrives at its predictions. As a result, these CNN-based models assist in the early detection and differentiation of pneumonia and COVID-19 [23, 24].

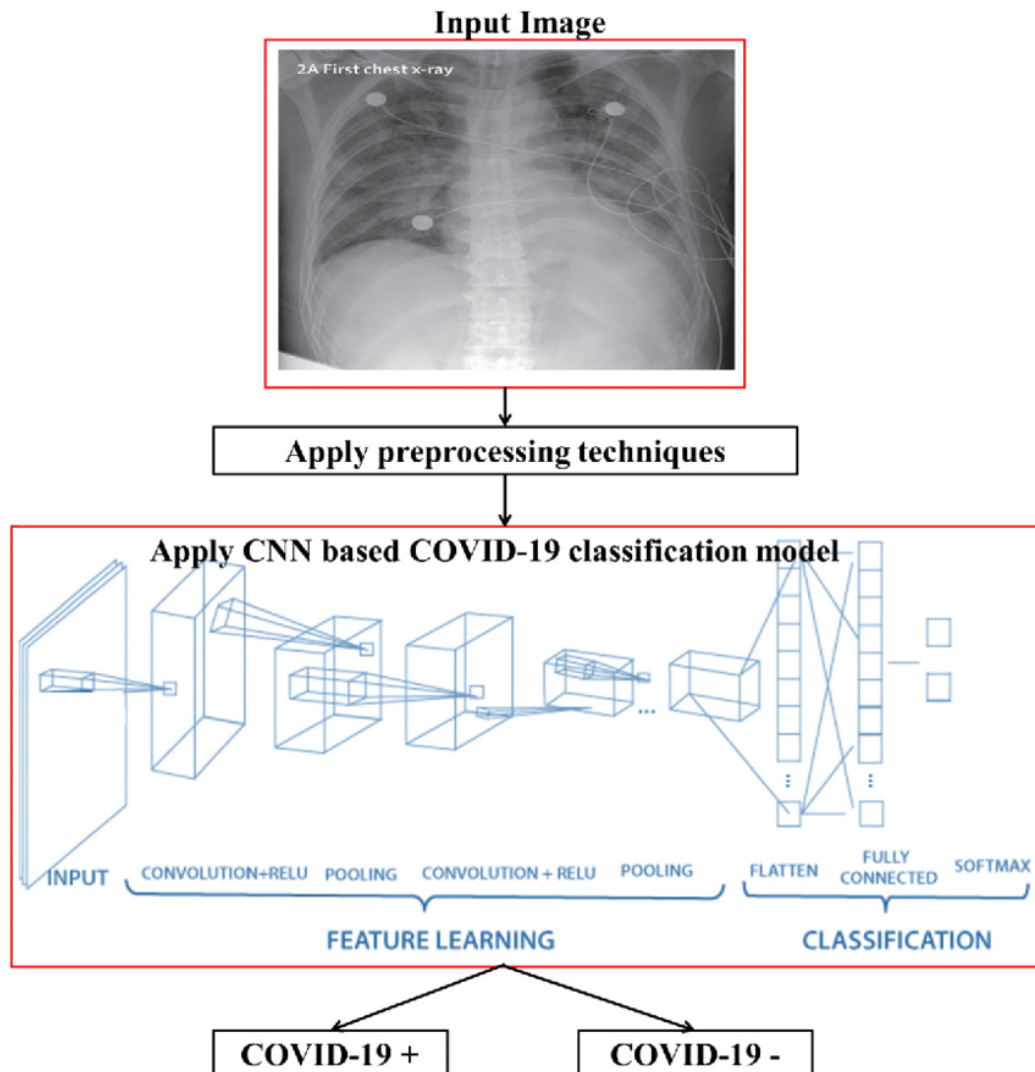


Figure 2.16: COVID-19 classification system example

- **Histopathology Image Classification:** Histopathological slides contain critical information for diagnosing various cancers and other diseases by provid-

ing a detailed view of tissue structures at the cellular level. CNNs have become essential tools for analyzing these images, offering automated methods to classify and detect abnormal tissue features with high accuracy. Models trained on datasets such as PatchCamelyon (PCam) have demonstrated pathologist-level accuracy in tasks like breast cancer classification, enabling early detection of malignant cells. These CNNs can learn complex patterns from the image data, identifying key features such as tumor cells, abnormal cell clusters, and tissue morphology, which are often subtle and difficult to detect manually. In addition to image-level classification, whole-slide image analysis using deep learning pipelines has emerged as a powerful approach, allowing for the examination of entire tissue slides. By automating the detection of malignant cells across large areas, these deep learning models streamline the diagnostic process, reduce the risk of human error, and ultimately aid in the early diagnosis of cancer [25, 26].

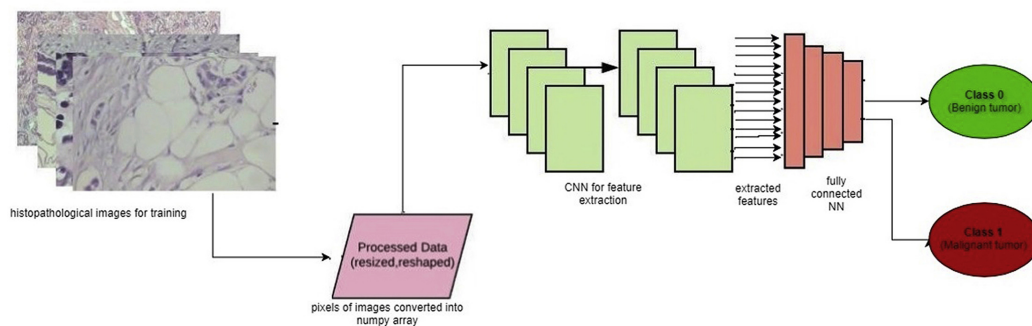


Figure 2.17: Histopathology image classification system example

The advantages of CNN-Based Classification are:

- Automated feature extraction: Reduces the need for handcrafted feature engineering.
- High accuracy in large datasets: CNNs generalize well when trained on

extensive medical image datasets.

- **Reduced need for expert interpretation:** Assists radiologists and pathologists in decision-making.
- **Scalability to different medical domains:** CNNs can be fine-tuned for diverse classification tasks.

### 2.3.4.2 Object Detection and Localization

Unlike traditional image classification, which focuses solely on categorizing images into predefined classes, object detection goes a step further by identifying and localizing specific regions of interest (ROIs) within an image. In the context of medical imaging, this means pinpointing abnormalities such as tumors, lesions, or other pathological features, allowing clinicians to focus on these critical areas for more accurate diagnoses and targeted treatment. CNN-based object detection techniques can be broadly categorized into two types: region-based methods, such as R-CNN and Faster R-CNN, which first propose candidate regions for object detection, and single-shot detectors, like YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector), which perform detection in one step, offering real-time processing capabilities.

- **Tumor Detection in MRI and CT Scans:** Tumor detection and localization in MRI and CT scans are essential for the accurate diagnosis of various types of cancers, including brain tumors, lung nodules, and other abnormalities. Models such as Faster R-CNN, Mask R-CNN, and YOLO are at the forefront of this task, leveraging deep learning to automatically identify and pinpoint tumor regions within medical images. These models have demonstrated significant improvements in diagnostic precision by reducing both false positives and false negatives, making them highly effective in real-

world clinical applications. The ability to detect tumors with high accuracy is especially valuable in complex medical imaging scenarios, where manual detection can be time-consuming and prone to human error. Automated tumor detection also plays a critical role in assisting neurosurgeons and oncologists in treatment planning, providing them with accurate tumor boundaries and locations. This information is invaluable for surgical navigation, enabling surgeons to target tumor areas precisely, while minimizing damage to surrounding healthy tissue, ultimately improving patient outcomes [27, 28].

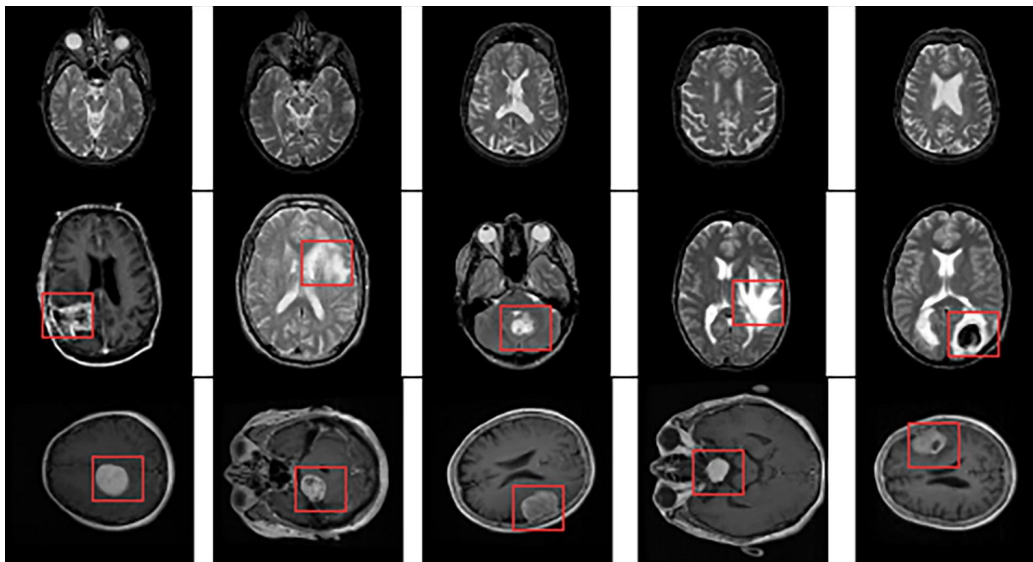


Figure 2.18: Tumor Detection in MRI and CT Scans

- **Polyp Detection in Colonoscopy:** Detecting colorectal polyps from colonoscopy images is a critical step in the early detection of colorectal cancer, one of the leading causes of cancer-related deaths. CNNs play a significant role for automating this process, offering highly accurate detection of polyps within colonoscopy images. These models are trained to recognize both common and atypical polyp shapes, sizes, and textures, improving early-stage diag-

nosis and enabling timely intervention. Real-time polyp detection models, when integrated directly into colonoscopy systems, can alert gastroenterologists to potential abnormalities during the procedure. This not only helps in identifying polyps that may be overlooked by the human eye but also increases the overall accuracy of colorectal screenings. By assisting in the detection process [29].

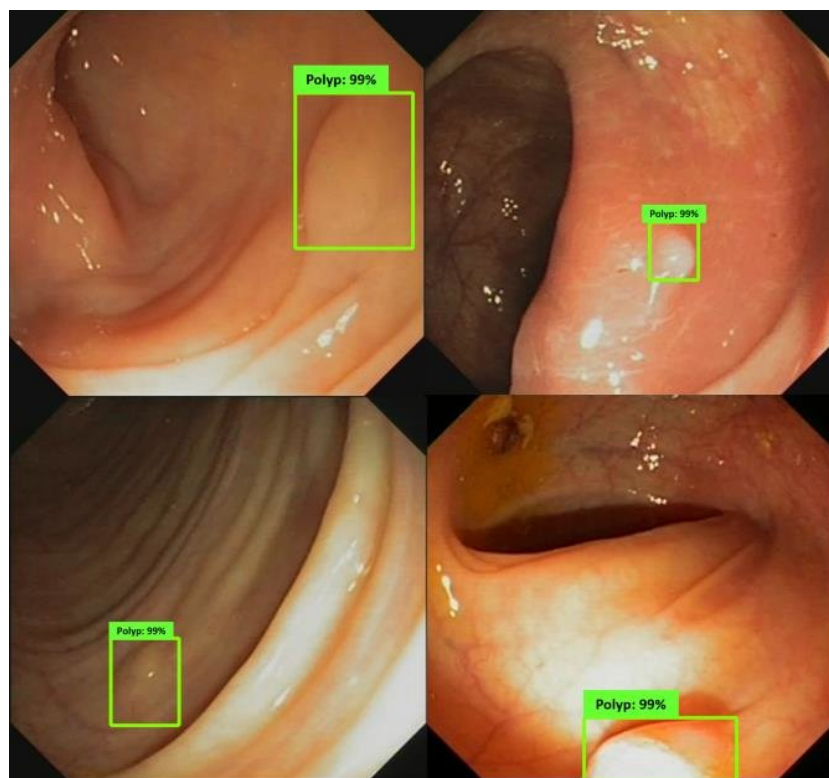


Figure 2.19: Polyp Detection in Colonoscopy images

- **Fracture and Lesion Detection in X-Rays:** Identifying bone fractures and soft tissue lesions in radiographic images is crucial for accurate diagnosis and timely treatment. CNNs are highly effective in automating the detection of these abnormalities, providing accurate and reliable results for radiologists. Advanced deep learning models not only excel in identifying fractures

and lesions but also offer explainability features that allow clinicians to visualize the key regions of concern within X-rays. This transparency helps medical professionals understand the reasoning behind the model's predictions, improving their confidence in the diagnosis. In emergency settings, where rapid triage is essential, AI-assisted fracture and lesion detection can significantly expedite the decision-making process, ensuring that patients receive appropriate care without delay. By assisting radiologists in detecting even subtle or complex fractures and lesions [30].

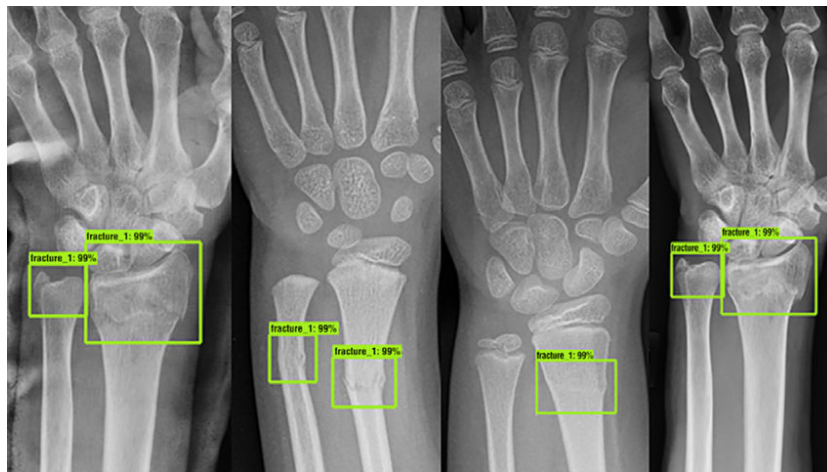


Figure 2.20: Bone fracture detection in X-Rays

The advantages of CNNs in Object Detection are:

- Faster and more accurate detection: Reduces diagnostic delays.
- Helps reduce human error in diagnosis: Provides a second opinion for radiologists.
- Enables real-time analysis for quick decision-making: Critical in emergency and surgical settings.

### 2.3.4.3 Medical Image Segmentation

Medical image segmentation is a crucial task that involves dividing an image into distinct, meaningful regions based on the structures or abnormalities it contains. For example, it can be used to separate tumors from surrounding healthy tissue, or to distinguish between different anatomical structures in medical scans. Accurate segmentation plays a vital role in diagnosing and planning treatment for various medical conditions. CNN-based architectures such as U-Net, DeepLabV3+, and SegNet are commonly employed for these tasks due to their ability to learn complex patterns in medical images. U-Net, for instance, is well-known for its encoder-decoder structure, which captures both fine details and high-level features. DeepLabV3+ utilizes atrous convolutions to capture multi-scale information, making it effective for images with varying object sizes. SegNet is known for its robust feature extraction and segmentation capabilities. Deep learning-based approaches, particularly those leveraging CNNs, significantly outperform traditional image segmentation methods by effectively utilizing spatial relationships and contextual information within the image. These deep models can automatically learn the features that are most important for accurate segmentation, providing superior performance compared to classical techniques like thresholding, region growing, or edge detection.

- **Brain Tumor Segmentation:** U-Net and its variants, such as 3D U-Net, are widely used for segmenting brain tumors in MRI scans, providing precise boundary delineation between the tumor and surrounding healthy tissue. These models excel in capturing fine-grained details, making them essential for accurately identifying complex tumor structures, even in cases where tumors are small or irregularly shaped. This level of precision is crucial for clinical applications such as surgical planning, where it enables surgeons to

effectively navigate the tumor's location and remove it with minimal damage to healthy tissue. Additionally, accurate tumor segmentation plays a key role in radiation therapy, ensuring that the radiation is precisely targeted at the tumor, improving treatment outcomes and reducing the risk of harm to surrounding healthy brain tissue. The Brain Tumor Segmentation (BraTS) dataset is a widely used resource for benchmarking CNN-based brain tumor segmentation models, offering a large, annotated collection of MRI scans. By utilizing this dataset, researchers can evaluate and compare the performance of different models, contributing to advancements in segmentation accuracy for clinical use [31, 32].

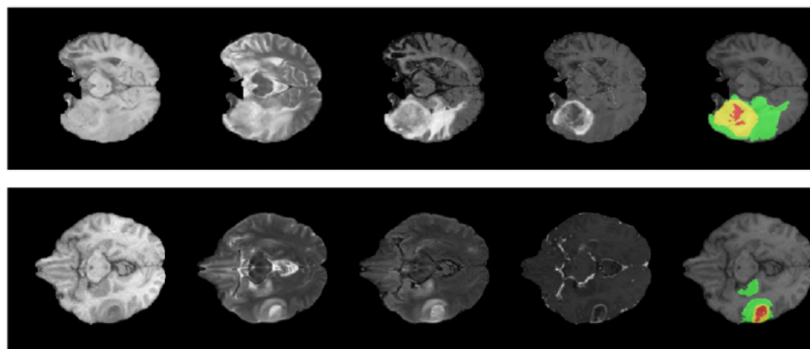


Figure 2.21: Brain tumor segmentation example

- **Lung and Heart Segmentation:** Segmentation of lung fields and heart structures in CT scans is crucial for the accurate diagnosis and assessment of respiratory and cardiovascular diseases. By isolating the lungs and heart from surrounding tissues, this process helps physicians evaluate conditions such as Chronic Obstructive Pulmonary Disease (COPD), lung cancer, heart failure, and other related diseases. AI-assisted segmentation has become an integral part of clinical practice, as it offers enhanced accuracy and efficiency compared to manual methods. In particular, CNN models, often integrated with advanced techniques like transformers, have shown signif-

icant improvements in segmentation performance, especially in challenging cases where the boundaries between organs and surrounding tissues are unclear. These hybrid models can capture both spatial and contextual information, leading to more robust results in scenarios with varying lung and heart anatomy, and even in cases with significant pathologies. Such advancements are transforming the way healthcare professionals approach diagnosis and treatment planning, especially in complex conditions where precise organ delineation is critical [33].

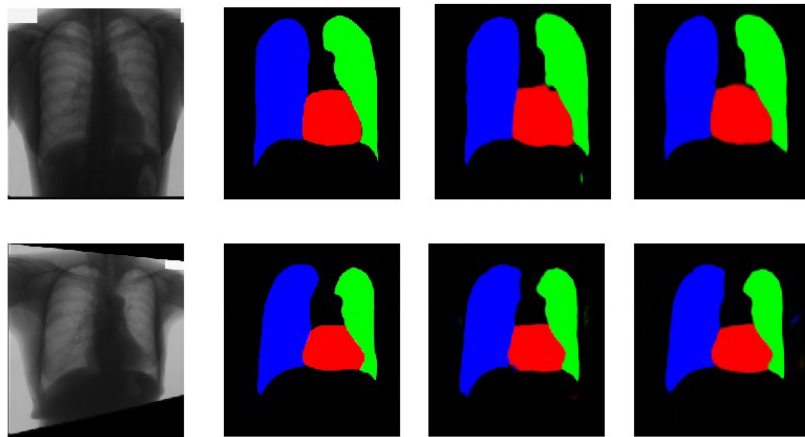


Figure 2.22: Lung and Heart Segmentation example

- **Retinal Blood Vessel Segmentation:** Segmenting retinal blood vessels from fundus images is a crucial step in diagnosing eye-related diseases such as diabetic retinopathy and glaucoma. Accurate segmentation allows healthcare professionals to assess the condition of blood vessels in the retina, which can reveal early signs of disease, including microvascular changes that are indicative of diabetes-related complications or glaucoma-related damage. CNN models excel at extracting fine-grained vessel structures from fundus images, improving the early diagnosis of these conditions by providing highly detailed and accurate segmentation. These models are capable of

identifying intricate vessel patterns even in challenging images with poor contrast or abnormalities. Additionally, deep learning pipelines that incorporate graph-based learning techniques have further enhanced segmentation accuracy. By modeling the relationships between adjacent vessels and incorporating contextual information, these advanced approaches improve the precision of vessel boundary detection, even in regions with noise or overlapping structures, ultimately leading to better clinical outcomes in the early detection and treatment of retinal diseases [34].

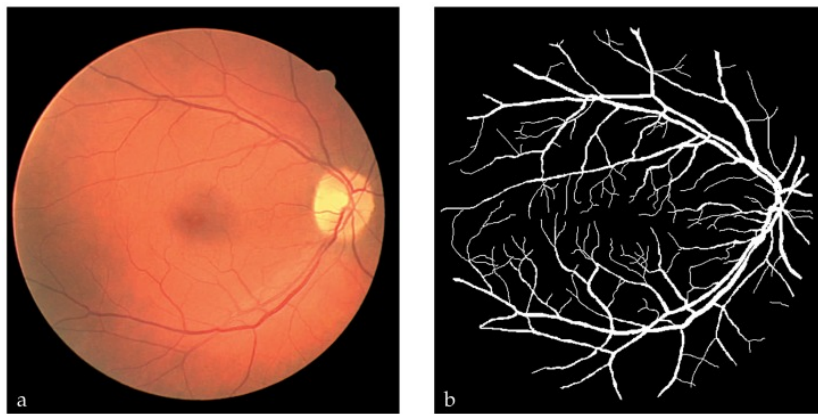


Figure 2.23: Retinal Blood Vessel Segmentation example

The advantages of CNN-Based Segmentation are:

- High accuracy in delineating structures: Essential for precise treatment planning.
- Reduces manual annotation time: Saves radiologists' efforts in labor-intensive tasks.
- Improves treatment planning precision: Provides accurate measurements for clinical applications.

#### 2.3.4.4 Image Enhancement and Super-Resolution

Medical images often suffer from noise and low resolution. CNNs are used to enhance image quality, making details more visible for accurate diagnosis. Super-resolution techniques based on deep learning reconstruct high-resolution images from low-quality inputs.

- Noise Reduction in MRI and CT Scans: CNNs remove artifacts and noise, improving image clarity while preserving diagnostic information. Deep denoising autoencoders enhance MRI images affected by motion artifacts [35].

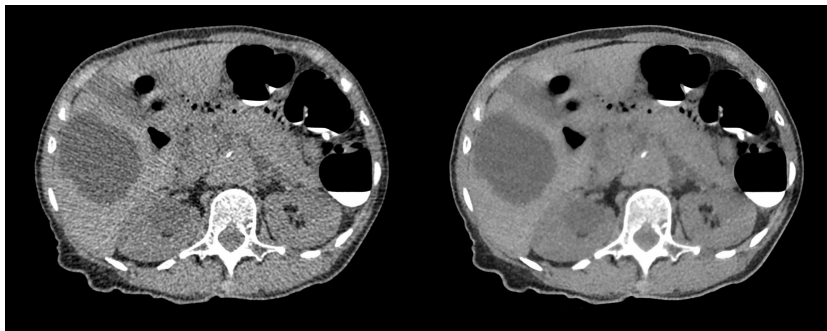


Figure 2.24: Noise reduction of MRI and CT scan images

- Super-Resolution for Medical Imaging: CNNs and Generative Adversarial Networks (GANs) increase image resolution, allowing finer details to be observed without additional scans. ESRGAN (Enhanced Super-Resolution GAN) improves diagnostic quality in medical imaging [36].

The benefits of CNN-Based Image Enhancement are:

- Improves visibility of anatomical structures: Enhances diagnostic interpretation.
- Reduces the need for repeated imaging: Lowers patient radiation exposure.

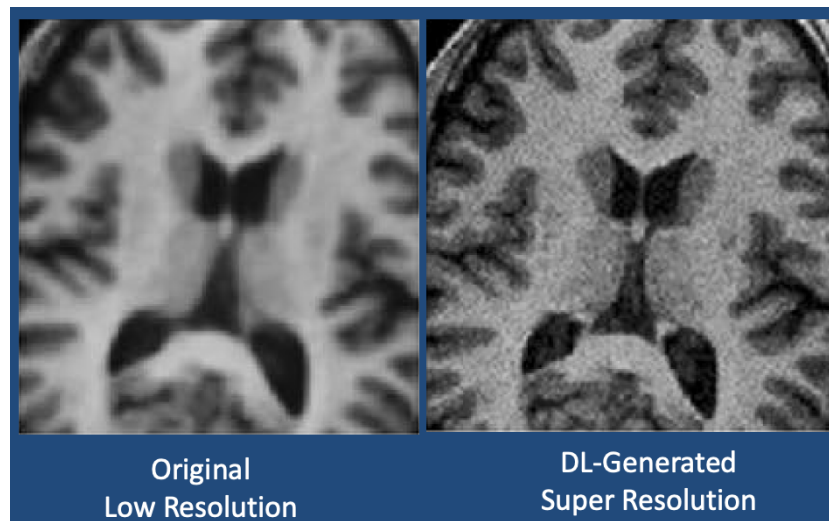


Figure 2.25: medical image supersampling for increasing resolution

- Enhances remote diagnosis capabilities: Supports telemedicine applications.

#### 2.3.4.5 Disease Progression Prediction and Prognosis

Beyond diagnosis, CNNs contribute to predicting disease progression and prognosis.

- Alzheimer's Disease Prediction: CNNs analyze brain MRI scans to predict the progression of Alzheimer's disease, distinguishing between mild cognitive impairment and advanced stages.
- Cancer Prognosis: Deep learning models analyze histopathological images to predict patient survival rates and tumor aggressiveness.

Predictive models assist clinicians in personalized treatment planning and early intervention strategies.

## 2.4 Conclusion

After highlighting the diverse applications of CNNs in medical imaging, including classification, detection, and segmentation tasks across various diseases, we are focusing our research on diabetic retinopathy (DR). Given that DR is a leading cause of blindness worldwide, its early detection is crucial for preventing vision loss. CNN-based models have demonstrated remarkable accuracy in DR classification using fundus images, making them an ideal choice for automated screening. By leveraging deep learning techniques, our study aims to enhance DR detection and classification, contributing to more effective and accessible diagnostic solutions.

## 3 | Deep learning for Diabetic Retinopathy detection

### 3.1 Introduction

Diabetic retinopathy (DR) is a leading cause of vision impairment globally, making early detection crucial for effective treatment. Deep learning has emerged as a powerful tool for automated DR detection, enabling rapid, accurate analysis of retinal images. This chapter explores the prevalence of DR, its stages, and how deep learning is transforming screening and diagnosis.

### 3.2 Generalities on Diabetic Retinopathy

Diabetes, or diabetes mellitus (DM), is a global health crisis, striking individuals of every demographic and region. It's characterized by chronic hyperglycemia—elevated blood glucose levels—due to either the body's inability to produce insulin (Type 1 diabetes) or its failure to respond adequately to insulin (Type 2 diabetes). The World Health Organization has projected for the world: by the year 2030, the number of people living with diabetes will climb to 360 million globally [37]; even more recent estimates predict higher numbers as socioeconomic, dietary, and lifestyle changes contribute to an accelerated increase in cases around the world.

One of the most serious complications of DM is diabetic retinopathy, presented in Figure 3.1, a condition that progresses to cause severe vision impairment or even blindness if it remains untreated. DR occurs as a consequence of the state of prolonged hyperglycemia damaging the blood vessels inside the retina, which eventually leads to microaneurysms, hemorrhages, lipid exudates, and potentially

retinal detachment. It has been estimated that more than one-third of people with diabetes develop DR to some degree and the rate is higher in those with long-standing diabetes or poor glycemic control. In addition, since it often has no initial symptoms, regular monitoring is necessary for proper intervention.

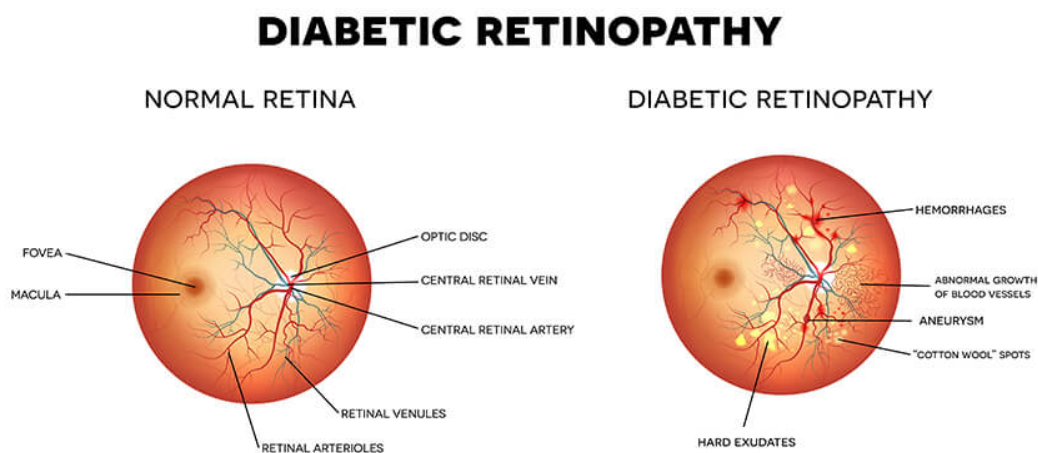


Figure 3.1: The problems present in diabetic retinopathy affected retina versus a normal one

The current gold standard for the diagnosis of DR is fundus photography, a non-invasive imaging technique that retains detailed images of the retina. Ophthalmologists manually examine these images for early signs of DR, such as microaneurysms, hemorrhages, and exudates. However, it is a very laborious, time-consuming process subject to wide inter-observer variability. Differences in the experience levels and judgment among ophthalmologists may give rise to inconsistency in DR severity assessment—implying poor patient outcomes. There is, therefore, a critical need for a more efficient and standardized approach to the diagnosis of DR.

In combating such challenges, researchers have focused on automated DR detection and severity classification systems. These systems, by using state-of-the-

artificial intelligence, especially in the areas of deep learning and computer vision, analyze fundus images to allow for the possibility of detecting signs of DR and accurately predicting the severity of the disease. Such automated systems have the potential to revolutionize DR management by enabling faster, more reproducible, and objective assessments. Implemented in health settings, especially those that are under-resourced, these tools would enable the health providers to raise the chances of early diagnosis and facilitate timely treatment, thereby averting most cases of vision loss associated with DR.

With the expected rise in diabetes cases worldwide, the development and deployment of autonomous DR screening systems could be life-transforming with regard to one of the most debilitating complications of diabetes. Increasing access to and lowering costs of regular DR screenings, these technologies have the potential to bring a profound effect on public health in general, improving quality of life for millions and subsequently easing the economic burden due to diabetes-related visual impairment and blindness.

### **3.3 Degrees of Diabetic Retinopathy Severity**

Diabetic retinopathy (DR) and diabetic maculopathy are important to assess in the stages of severity for treatment and visual outcomes in patients with diabetes. Several studies have revealed variable concordance of the severity stages of DR assigned to screening photographs versus standardized reference photographs, with agreement for DR at 60.6%–75.8% and for macular edema at 93.2%–94.6%. This variability suggests good concordance for macular edema evaluations but less so for DR, possibly supporting the fact that DR grading can be more subject to interpretative variations. Mistakes in severity staging are a common occurrence, with overestimation being a fact in cases where artifacts or lesion sites mainly at the

posterior pole or specific microvascular changes are misinterpreted as neovascularization. The same can be said for epithelial changes, which can simulate, in some instances, dry exudates. In contrast, underestimations occur mainly because retinal hemorrhages sometimes lie outside the screened field and do not get picked up during the initial assessment process.

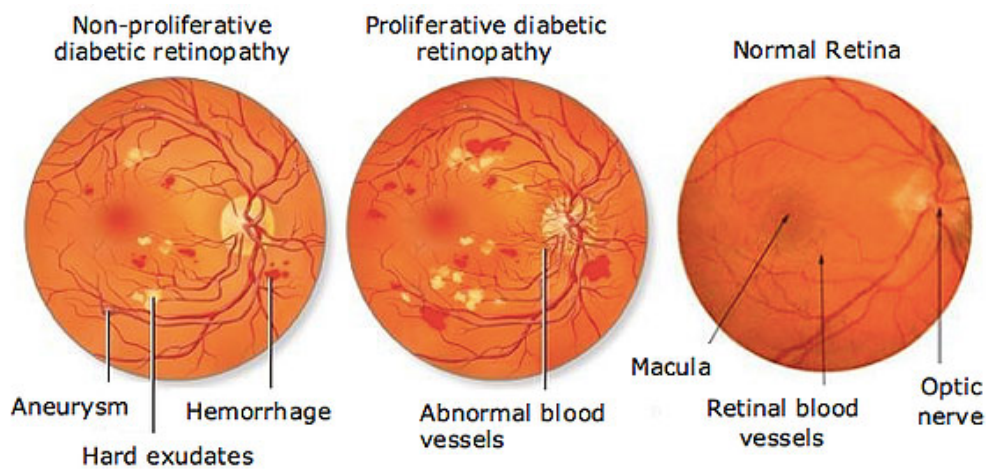


Figure 3.2: Stages of diabetic retinopathy compared with a normal retina

These results underline the need for very serious training in the interpretation of images from the fundus, especially because it may be difficult to interpret the DR stages. The training of the physician increases consistency in the diagnosis and avoids misclassification, meaning that patients will receive appropriate care. These findings are also useful in developing screening programs based on lesion location, particularly in the case of maculopathies, where lesion location with respect to the center of the macula is crucial. The maculopathy screening classification is especially helpful in streamlining vision-threatening types of DR that are referred to ophthalmologists in clinical practice.

This approach not only optimizes resources but also prioritizes patients who are at higher risk of vision loss, which is in line with the objective of reducing

preventable blindness in diabetic populations. Given the variability in interpretation and impact of accurate staging on patient outcomes, both standardized image reader training and simplified classification frameworks can augment quality diabetic eye care in both clinical and screening settings. More emphasis on accurately diagnosing vision-threatening conditions may help streamline the referral pathways of patients with advanced DR or maculopathy, ensuring timely specialist intervention [38].

Retinal angiography is often performed to confirm the severity of retinal ischemia.

- **Absence of diabetic retinopathy:** normal retina
- **Non-proliferative diabetic retinopathy**
  - Minimal non-proliferative diabetic retinopathy: We will thus find a small number of micro-aneurysms and punctate hemorrhages during clinical examinations.
  - Moderate non-proliferative diabetic retinopathy: From a clinical point of view, the patient will present [39]:
    - \* Microaneurysms,
    - \* And/or “flame” or punctiform hemorrhages,
    - \* And/or cotton wool nodules,
    - \* And/or intraretinal microcirculation abnormalities,
    - \* And/or “spot” retinal hemorrhages in less than four quadrants of the retinal periphery.
- **Severe non-proliferative diabetic retinopathy:** This stage presents a significant risk of progression to neovascularization. Its definition corresponds

to one of the following three criteria:

- Severe hemorrhages in four quadrants of the retinal periphery,
- And/or moniliform venous anomalies in two quadrants,
- And/or numerous intraretinal microcirculation anomalies (AMIR) in one quadrant of the retinal periphery[39].

- **Proliferative diabetic retinopathy**

### 3.4 Automatic detection of Diabetic Retinopathy

The diagnosis and classification of diabetic retinopathy (DR) involve examining fundus images to identify disease characteristics and grade its severity. Typically, ophthalmologists analyze these images to identify early signs of DR, such as microaneurysms, hemorrhages, and hard exudates, which help them determine the type and progression stage of the disease. However, with advancements in artificial intelligence, machine learning (ML) and deep learning (DL) methods are increasingly being used as supplementary tools to assist doctors in diagnosing and classifying DR with enhanced accuracy and efficiency[40].

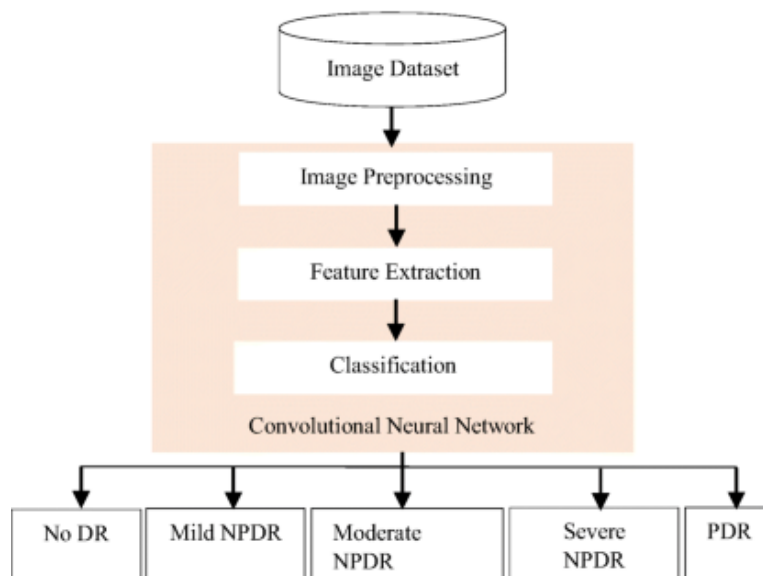


Figure 3.3: CNN-based diabetic retinopathy classification framework

In recent years, intelligent computer-aided systems have emerged as powerful tools for DR detection. These systems use ML techniques to train models on large datasets of labeled fundus images, enabling automated detection and classification of the disease. Traditional ML models and more complex DL models have both been applied to DR detection, with each type of model offering unique benefits and limitations. For example, traditional ML models may use predefined features extracted from images, such as color histograms or texture measures, while DL models, particularly CNNs, are able to automatically learn and extract relevant features directly from raw images during training. Each model type has shown promise in DR detection, but they vary in terms of accuracy, computational requirements, and adaptability to different datasets. By leveraging the strengths of different models, researchers have been able to achieve highly accurate classification results across various clinical datasets and contexts[41].

One of the essential steps in developing these computer-aided systems for DR

detection is image preprocessing. This step involves segmenting and processing the raw fundus images to enhance the features used for training and classification. Image preprocessing may include tasks like noise reduction, contrast enhancement, and color normalization, all of which help improve the clarity of features that are essential for distinguishing between different stages of DR. Additionally, segmentation techniques may be applied to isolate specific regions of interest within the image, such as the optic disc or blood vessels, which can provide crucial information for accurate DR diagnosis[42]. By optimizing these preprocessing steps, the quality and reliability of feature extraction are significantly improved, which in turn enhances the performance of the classification model.

Once the images have been preprocessed, specific features are extracted for use in training the model. These features might include spatial characteristics, intensity values, or texture descriptors that capture variations in image patterns associated with DR. During the training phase, a selected ML or DL model is exposed to this set of processed images along with their corresponding labels, enabling the model to learn and distinguish patterns that correlate with different stages of DR. To validate the model's performance, a separate test dataset is used, allowing researchers to assess the model's ability to generalize to new, unseen images from the same or similar databases[41].

In advanced ML and DL systems, subsets of features from the extracted feature vectors can be used to refine model training further. These feature subsets, which may represent critical indices or high-dimensional vectors, are combined with their labels to build a trained model. This trained model can then be used to classify new images, providing a prediction of the DR stage based on previously learned patterns. By iterating on this process and fine-tuning the model's parameters, researchers can create highly accurate, reliable diagnostic tools for DR detection, which hold great potential for integration into clinical practice. These

systems not only offer high accuracy but also promise to reduce the time and resources required for DR screening, making them especially valuable for regions with limited access to specialist care[41].

Ultimately, ML and DL-based DR detection systems represent a transformative step forward in the field of ophthalmology. By automating the diagnostic process, these technologies not only augment the capabilities of healthcare professionals but also open new avenues for preventive care and early intervention. The widespread adoption of such systems could lead to significant improvements in global DR management, helping reduce the burden of vision impairment and blindness associated with diabetic complications.

### **3.5 Preparing the fundus images**

#### **3.5.1 Acquisition of fundus images**

The fundus images, essential for diagnosing and monitoring various eye conditions, are acquired through a procedure known as “funduscopy” or “ophthalmoscopy”. This is an ophthalmic examination that enables a detailed view of the eye structures located behind the lens, including the vitreous body, retina, macula, retinal vessels, and optic nerve head, also known as the papilla. Funduscopy plays a vital role in assessing the health of the retina, macula, and the eye’s vascular system, helping to identify early signs of conditions like diabetic retinopathy, age-related macular degeneration, and glaucoma.

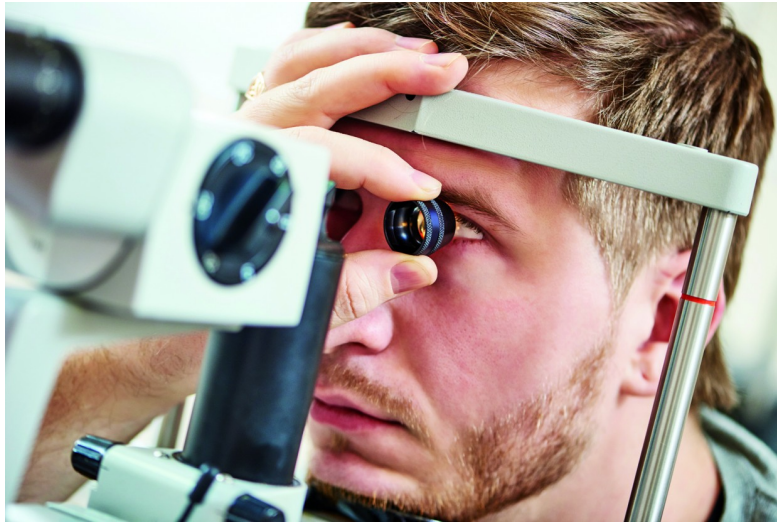


Figure 3.4: Fundoscopy procedure for retinal examination

The retina itself is a delicate layer of nerve fibers and is densely packed with photoreceptors, which are cells responsive to light. These photoreceptors are crucial for visual function as they capture light entering the eye and convert it into neural signals that are then sent to the brain via the optic nerve. The retina thus functions as a sensory membrane lining the back of the eye, positioned behind the iris and lens. At the center of the retina is the macula, an area critical for sharp, detailed vision, which is why damage to the macula often results in the loss of central vision. It is within this area that fine visual tasks, like reading or recognizing faces, are processed in detail due to the concentration of cone cells, which are photoreceptors specialized for color vision and high spatial acuity [\[43\]](#).

Fundoscopy, often performed by an ophthalmologist, varies slightly depending on the area of the fundus being examined and the need for pupil dilation. For example, if the ophthalmologist intends to focus primarily on the central part of the fundus—where the macula and optic nerve are located—the examination may be performed without pupil dilation, allowing a quick and minimally inva-

sive procedure with no special preparation needed. This type of examination is generally brief, painless, and can be completed using a lens with a mirror placed directly on the eye or by employing contact-free techniques, such as an indirect ophthalmoscope or biomicroscopy with a biconvex lens.

To gain a more comprehensive view of the retinal periphery, however, pupil dilation is often necessary. This involves administering mydriatic eye drops to enlarge the pupil, typically requiring a short wait of 30 to 45 minutes to take effect. Dilation enables the ophthalmologist to inspect the entire fundus, including its outer edges, thereby improving the detection of peripheral retinal diseases or tears. The decision to dilate the pupil depends on the patient's clinical needs and is made based on a thorough assessment. It's worth noting that dilation can cause temporary side effects, including a mild burning sensation, a dry mouth, and occasionally an unpleasant taste as the drops reach the throat through the nasolacrimal duct.

If the examination requires a closer look with a contact lens, the ophthalmologist may first apply a numbing eye drop to anesthetize the cornea, a thin, transparent layer that covers the front of the eye. This anesthetic ensures that the procedure is completely painless and comfortable for the patient, as the effect of the numbing drop lasts only a few minutes and allows the direct contact required for some types of funduscopy. The use of this lens provides the ophthalmologist with a high-resolution view of the retina and optic nerve, facilitating a detailed assessment and enabling the early detection of potential eye disorders, which can often be asymptomatic in the early stages[44].

Funduscopy is thus an indispensable tool in eye care, as it provides a non-invasive way to monitor the health of the retina, macula, and optic nerve while ensuring that any signs of disease are detected promptly. The combination of

advanced imaging and skilled interpretation by an ophthalmologist can lead to timely treatments and improved outcomes for patients experiencing visual issues, underlining the importance of this procedure in routine and specialized eye examinations.

### 3.5.2 Fundus image preprocessing techniques

Preprocessing is a way to correct some defects in images for better learning as well as the augmentation of the number of images to improve the robustness and generalization of the CNN by exposing it to a wider range of possible variations in the data. In general, this results in the application of image filters or operations on the image histogram followed by augmentation transformations.

The methods adopted for image preprocessing to obtain the characteristics of diabetic retinopathy:

- Gaussian filter,
- Anisotropy filter.
- RGB color space decomposition,
- Histogram expansion,
- Histogram equalization (CLAHE).
- Median filtering.
- Morphological operations.

The geometric methods adopted for data augmentation are:

- Rotation
- Translation

- Scaling/Zooming
- Flipping
- Shearing

## **3.6 Working Fundus images dataset**

### **3.6.1 Presentation of existing datasets**

#### **3.6.1.1 APTOS 2019**

The data used in this work is a collection of eye fundus images obtained using fundus imaging. This process uses reflected light to get a two-dimensional (2D) representation of semi-transparent retinal tissues in 3D that is projected onto the image plane. We use in this work the APTOS 2019 DR Kaggle dataset [45] to develop and validate the proposed system. The APTOS 2019 DR Kaggle benchmark dataset is a part of the blindness detection challenge of the same name, is a sizable dataset of retinal photos obtained with a fundus camera under various imaging circumstances. To identify various DR severity levels, the images undergo manual grading on a scale from 0 to 4, where 0 represents no DR, 1 for mild, 2 for moderate, 3 for severe, and 4 for proliferative DR. The distribution of the classes in the APTOS 2019 DR dataset is shown in Table 1, while Figure 3.5 illustrates some DR images extracted from different APTOS 2019 classes.

Table 3.1: Class distribution in the APTOS 2019 dataset

| Scale | Severity level   | # Samples |
|-------|------------------|-----------|
| 0     | No DR            | 1805      |
| 1     | Mild DR          | 370       |
| 2     | Moderate DR      | 999       |
| 3     | Severe DR        | 193       |
| 4     | Proliferative DR | 295       |
| Total |                  | 3662      |

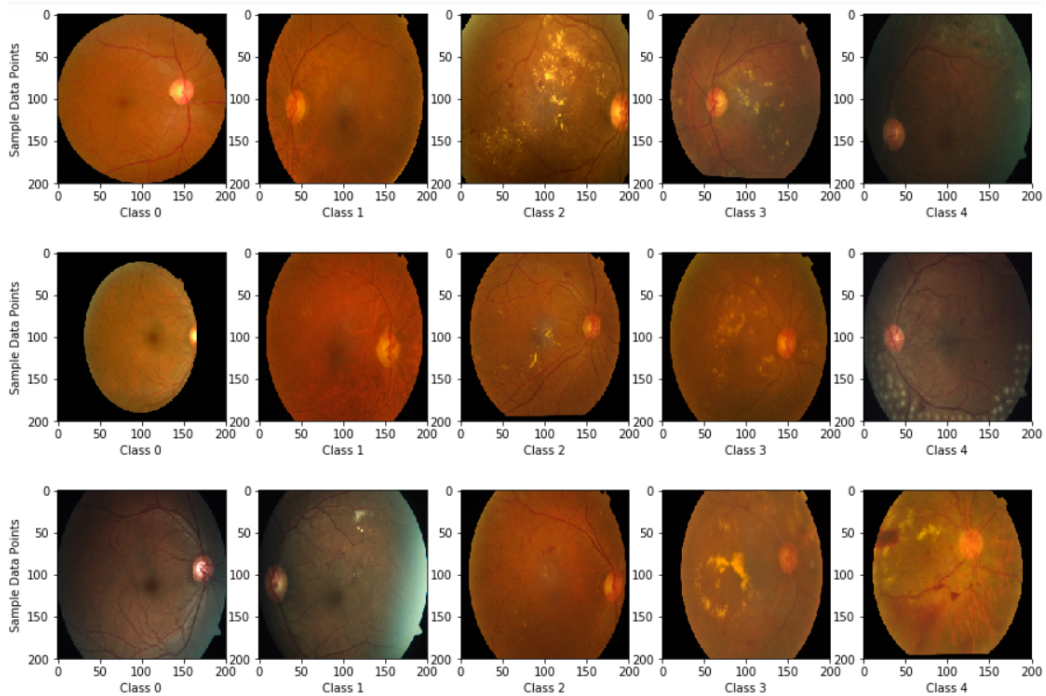


Figure 3.5: Fundus images example of the APTOS 2019 dataset

### 3.6.1.2 EyePACS

Diabetic Retinopathy (DR) is a common complication of diabetes that affects the eyes. The EyePACS Diabetic Retinopathy dataset [46] is a widely used dataset for detecting DR through retinal fundus images. Developed as part of the Kaggle Diabetic Retinopathy Detection challenge, this dataset provides annotated images

for training machine learning models to diagnose DR stages accurately. The main objective is to automate the DR detection, which can help in early diagnosis and better healthcare management for patients with diabetes.

The EyePACS DR dataset includes high-resolution retinal fundus images, each having a label classifying the image according to the severity of DR present. In general, labeling is very important for classification tasks in machine learning. Their quality, resolution, and visual features change significantly from one another, making a realistic dataset for validating robust models in machine learning.

The dataset can be used for several computer vision tasks:

- Classification: Identifying the stage of DR.
- Localization: Emphasizing impacted areas.
- Segmentation: Segmenting lesions or abnormal areas.

The EyePACS dataset has been developed to follow the grading system in the severity of DR, where images are put into classes representing five stages from no DR to the most severe stage. The classification scheme of this dataset is shown in Table 2, while Figure 3.6 illustrates some DR images extracted from different EyePACS classes.

Table 3.2: Class distribution in the EyePACS dataset

| Scale | Severity level   | # Samples |
|-------|------------------|-----------|
| 0     | No DR            | 25810     |
| 1     | Mild DR          | 2443      |
| 2     | Moderate DR      | 5292      |
| 3     | Severe DR        | 873       |
| 4     | Proliferative DR | 708       |
| Total |                  | 35126     |

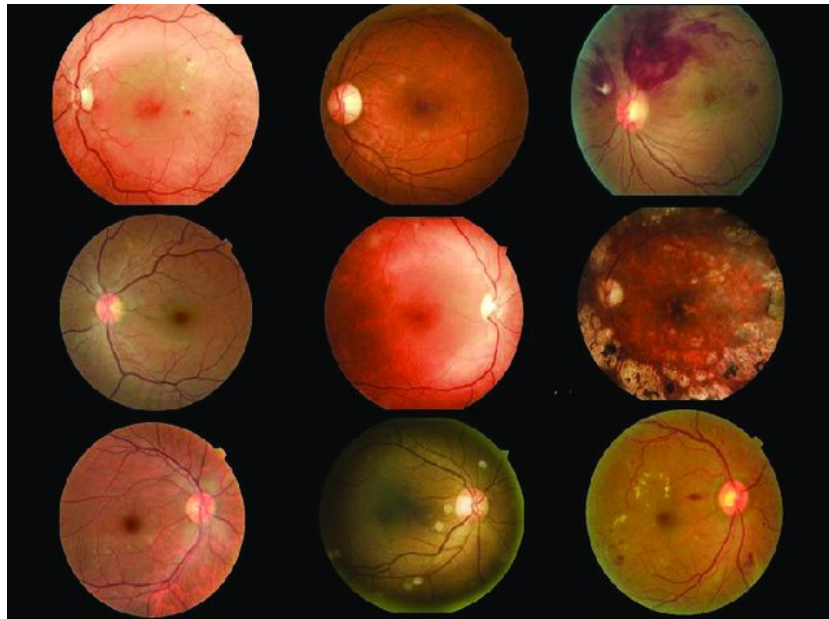


Figure 3.6: Fundus images example of the EyePACS dataset

### 3.6.1.3 MESSIDOR

MESSIDOR [47] is one of the base datasets in diabetic retinopathy research. The aim is to contribute to the development of automated systems for DR detection. This dataset was created by a consortium of French research institutes, and it includes retinal fundus images with annotated severity. This will be helpful for the classification and screening research in DR.

The MESSIDOR dataset contains high-resolution retinal images captured from patients suffering from diabetes. Each image is annotated according to the severity level of DR, apart from an assessment of macular edema risk. The labels for these provide a basis for:

- DR Classification: Assign severity level to the DR.
- DME classification: classify the presence and severity of diabetic macular edema.

This data set can be used for either classification or segmentation tasks. MESSIDOR classifies the data set based on four different levels of DR severity as shown in Table 3, while Figure 3.7 illustrates some DR images extracted from different MESSIDOR classes.

Table 3.3: Class distribution in the MESSIDOR dataset

| Scale | Severity level   | # Samples |
|-------|------------------|-----------|
| 0     | No DR            | 546       |
| 1     | Mild DR          | 153       |
| 2     | Moderate DR      | 247       |
| 3     | Severe DR        | 254       |
| 4     | Proliferative DR | /         |
| Total |                  | 1200      |

#### 3.6.1.4 MESSIDOR-2

MESSIDOR-2 [47] is the extension of the original MESSIDOR dataset, crafted to further support research in the detection and grading of Diabetic Retinopathy using retinal fundus images. In fact, MESSIDOR-2 was proposed with the aim of providing a larger and more varied set than its ancestor, thus supporting various tasks, such as classification, lesion identification, or DR severity grading.

MESSIDOR-2 contains high-resolution retinal fundus images from diabetic patients, all linked with a level of DR severity. The annotations in this dataset come from the original MESSIDOR dataset, which represents a four-level DR grading with respect to the severity of abnormalities in the retina. Moreover, it includes DME risk labels with indications of the presence and severity of edema.

The classification and segmentation tasks for the detection of diabetic retinopathy and macular edema are supported in MESSIDOR-2.

The severity of DR in this dataset is classified as shown in Table 4, while Figure

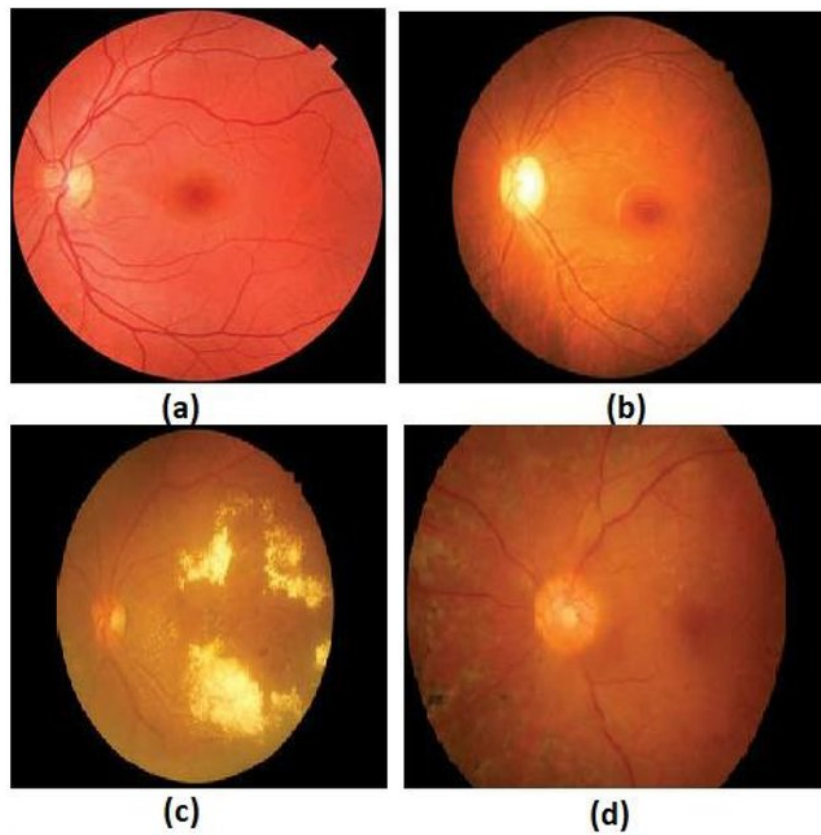


Figure 3.7: Images of MESSIDOR database (a) grade 0 (b) grade 1 (c) grade 2 (d) grade 3

3.8 illustrates some DR images extracted from different MESSIDOR-2 classes:

Table 3.4: Class distribution in the MESSIDOR-2 dataset

| Scale | Severity level   | # Samples |
|-------|------------------|-----------|
| 0     | No DR            | 1017      |
| 1     | Mild DR          | 270       |
| 2     | Moderate DR      | 347       |
| 3     | Severe DR        | 75        |
| 4     | Proliferative DR | 35        |
| Total |                  | 1744      |

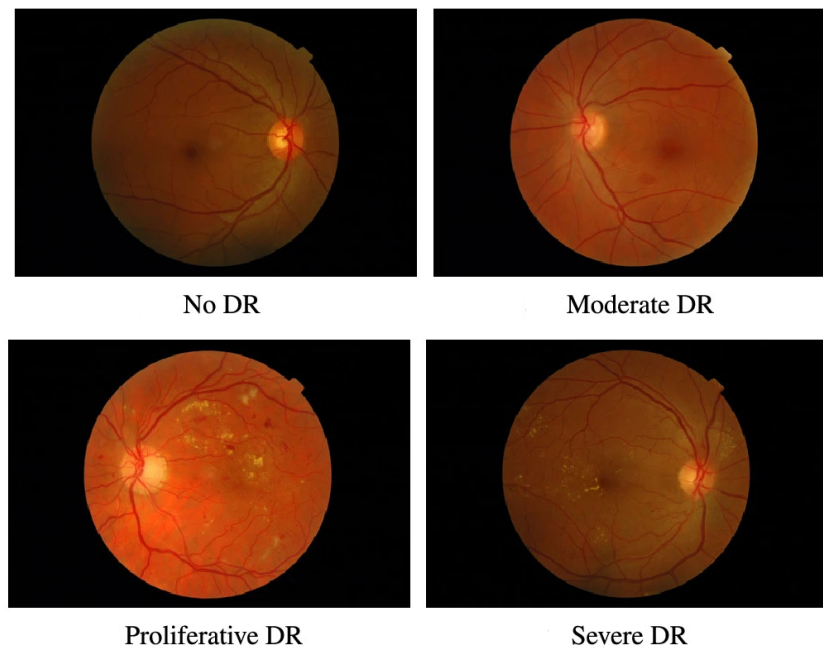


Figure 3.8: Fundus images example of the MESSIDOR-2 dataset

#### 3.6.1.5 IDRiD

IDRiD [48] represents a specialized dataset, developed to help in the diagnosis of Diabetic Retinopathy and diabetic macular edema. It is designed for research purposes, containing both disease severity level-labeled retinal fundus images and lesion segmentations that are appropriate for various computer vision tasks in medical imaging. The IDRiD data set contains high-resolution images of retinas for which annotations have been provided mainly into two categories:

- Disease Grading: Labels indicating DR severity level, starting from no DR to proliferative DR.
- Lesion Segmentation: Pixel level masks for lesions like microaneurysms, hemorrhages, and exudates.

Supported tasks include classification, segmentation of lesions, and abnormal-

ity localization. IDRIID Dataset labels images based on five levels of DR severity as show in Table 4, while Figure 3.9 illustrates some DR images extracted from different IDRIID classes:

Table 3.5: Class distribution in the IDRIID dataset

| Scale | Severity level   | # Samples |
|-------|------------------|-----------|
| 0     | No DR            | 168       |
| 1     | Mild DR          | 25        |
| 2     | Moderate DR      | 168       |
| 3     | Severe DR        | 93        |
| 4     | Proliferative DR | 62        |
| Total |                  | 516       |

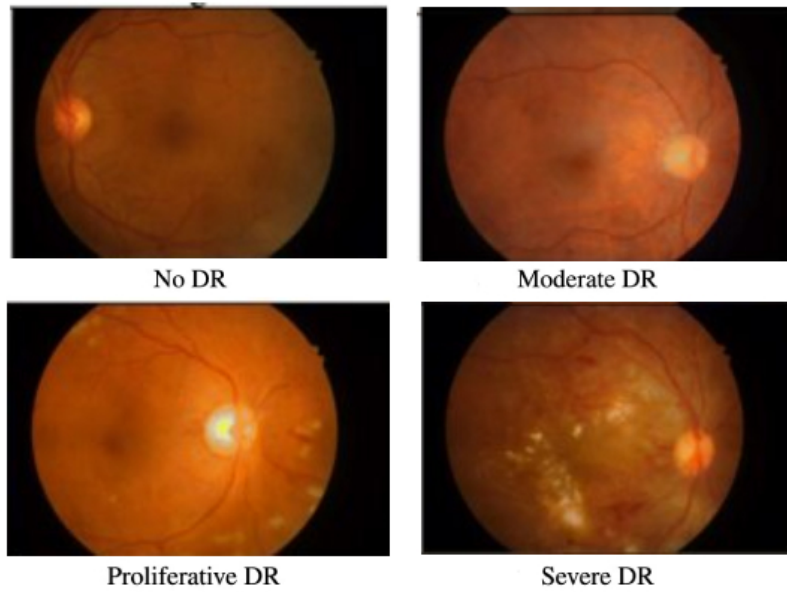


Figure 3.9: Fundus images example of the IDRIID dataset

### 3.6.2 Choice of working database

For the task of diabetic retinopathy (DR) classification, we opted to use the APTOS 2019 dataset because of its high-quality images and consistent labeling,

which are crucial for achieving reliable model performance. The APTOS 2019 dataset provides detailed annotations across five categories of DR severity, ranging from no DR to proliferative DR, allowing for a balanced representation of each level on the severity scale. This well-rounded distribution of DR stages enables the model to learn and recognize patterns that are critical for accurately detecting moderate to severe cases of DR, which are the most impactful in clinical settings.

Another key reason for selecting APTOS 2019 was its alignment with the clinical requirements for DR classification tools. The dataset was specifically curated to support robust, real-world applications in medical diagnosis, making it ideal for building a clinically viable DR screening model. Since our goal was to develop a DR screening tool that could be reliably used in healthcare environments, this clinical compatibility was essential. The real-world focus of the dataset not only supported our objectives but also made APTOS 2019 an industry benchmark due to its foundational role in the APTOS 2019 Blindness Detection competition. This competition brought the dataset to the forefront of machine learning research for DR, allowing us to validate our model against state-of-the-art results from other researchers.

Using this benchmark dataset enabled us to compare our model's performance with established standards and high-quality models within the community, which contributed significantly to refining our own approach. By building on these proven standards, we were able to further tune and optimize our model, ultimately creating a more precise and effective tool for DR classification that meets real-world diagnostic needs.

## 3.7 State of the art of deep learning methods in DR detection

### 3.7.1 Using basic Deep learning models

Recent diabetic retinopathy classification efforts have included several studies that make use of high-quality, annotated retinal fundus images in training and fine-tuning machine learning models. In particular, the APTOS 2019 dataset has been used, alongside other datasets such as EyePACS, MESSIDOR, and IDRiD, as a benchmark for the development and testing of numerous architectures, changes, and transfer learning methods. Here follows a more elaborate review of how some of these studies have made use of the datasets to push forward the state-of-the-art on DR classification.

Khalifa et al. [49] applied deep transfer learning models including AlexNet, SqueezeNet, ResNet18, GoogleNet, VGG16, and VGG19 for the detection of DR. Their work showed that transfer learning using those architectures, which are pre-trained on ImageNet, achieved competitive performances when fine-tuned on fundus images, revealing that by reusing existing deep learning models on medical image datasets, the DR detection accuracy can be increased.

Dekhil et al. [50] utilized a modified VGG16 to develop a transfer learning approach targeting the extraction of DR-related features; the model, by selectively fine-tuning the layers in VGG16, showed higher sensitivity and accuracy, as it became concentrated on DR-specific lesions in retinal images. Equally, Kassani et al. [51] put forward a bespoke Xception architecture for DR detection. This model exploited the depth and feature extraction of Xception, which showed superior results compared to other baseline models with a dataset-augmented architecture.

Tymchenko et al. [52] put forward several new pretrained models, each of which had been evaluated for the task of detecting DR with different severities; their approach has shown the great potential in using pretrained networks to enhance the detection in such challenging medical images, given the comparison of the architectures like DenseNet and EfficientNet, which were promising due to their depth and optimized parameter structures. Qureshi et al. [53] introduced a new multi-layer Active Deep Learning (ADL-CNN) architecture and targeted the selection of informative regions of interest (ROIs) from fundus images. This ROI-focused approach enabled their ADL-CNN to extract disease-specific features better and achieve improved performance on the EyePACS dataset compared to prior models.

Zeng et al. [54] applied a Siamese-like convolutional neural network architecture for the classification of DR and used fundus images of both eyes in order to give a holistic view. In this study, an AUC of 95.10% and sensitivity of 82.2% were achieved, which showed that fusion of images from both eyes may provide a better representation of DR development.

Shanthi et al. [55] have modified AlexNet with the addition of further pooling, softmax, and ReLU layers and achieved an average accuracy of 96.25% on the MESSIDOR dataset. The tailored architecture optimized the classic structure of AlexNet in capturing the subtle changes of the retina associated with DR. Similarly, Jain et al. [56] experimented with various pre-trained networks, including VGG16, VGG19, InceptionV3, to conclude that deeper models would generally do better, especially in binary and five-class classification tasks. They found that VGG19 had the best performance with an accuracy of 80.40%, which underscores the better capability of deeper architectures in the extraction of DR-related features.

Other works, like Quellec et al. [57], used a weakly-supervised model for referable lesion evaluation of DR images by generalizing the backpropagation. This model has achieved an AUC score of 95.5% and 94.9% on the Kaggle and E-Ophtha datasets, respectively. Its use of weak supervision is a promising approach for DR detection that evades the need for pixel-level lesion annotations. Gargeya et al. [58] applied automatic feature learning to MESSIDOR 2 and E-Ophtha datasets for glaucoma assessment and obtained an AUC of 97.00% in 5-fold cross-validation, proving the effectiveness of automatic feature extraction in the detection of ophthalmic diseases.

Hagos et al. [59] proposed a DR classification model based on Inception V3 and transfer learning with a small dataset and trained it using an SGD optimizer along with the cosine loss function and got 90.9% accuracy. This work shows the importance of choosing optimizer and loss function, even in the scenario of a relatively small dataset.

Several works have focused on the APTOS 2019 dataset alone. Shakibania et al. [60] used deep learning models, pre-trained on a big multi-center dataset including APTOS 2019, for the detecting and grading of stages of DR. It had stressed in that study the importance of having large diverse datasets on which models could be generalized for the purpose of improving the accuracy in classifying the severity of DR in clinical scenarios. Both Narayanan et al. [61] and Majumder et al. [62] suggested a two-stage hybrid system based on the APTOS 2019 dataset, which segmented their model into the detection and grading stages. This hybrid system allowed for DR identification in the first stage and a grading stage with high interpretability and accuracy in clinical implementation.

Modified VGG19 and Inception ResNet V2 architectures are used by Shaban et al. [63] and Gangwar et al. [64], respectively, for the classification of DR in the

APTOS 2019 dataset. The modified VGG19 by Shaban retained its superiority in performance with layer-specific adjustment, whereas the Inception ResNet V2 used by Gangwar achieved superior feature extraction capabilities that were very helpful in the high-resolution images of APTOS 2019.

Ramchandre et al. [65] used data augmentation by applying the AUGMIX technique to increase the size of the APTOS 2019 dataset, making the model more robust by diversifying the images on which it was trained and counteracting the risk of overfitting, improving model generalizability across different DR severity levels.

For better preprocessing, Md. Nahiduzzaman et al. [66] and Md. Robiul Islam et al. [67] applied Contrast Limited Adaptive Histogram Equalization (CLAHE) on APTOS 2019 images to increase the contrast in order to make DR-related lesions more visible. Nahiduzzaman integrated this preprocessing with a parallel CNN and an Extreme Learning Machine (ELM) classifier, whereas Islam further refined feature extraction by means of supervised contrastive learning, achieving high classification performance. Lastly, Jian et al. [68] proposed "Triple-DRNet," a model designed to subdivide DR classification into five stages of severity. The granular approach of this model provided more exact classification of DR severity, thereby presenting the value of multi-stage architectures in medical classification tasks.

Taken together, these studies present a wide range of innovative techniques in DR classification and underline the great importance of dataset selection, architecture modification, transfer learning, and preprocessing on model performance and reliability in clinical applications.

### 3.7.2 Using attention guided mechanism

The application of attention mechanisms to deep learning models for the classification of diabetic retinopathy has drawn great interest in recent years. Such techniques are meant to enhance model performance by disproportionately focusing on critical regions in the images that are characteristic of DR.

Wang et al. [69] introduced Zoom-in-Net, utilizing the deep mining strategy to explore the lesions for diabetic retinopathy detection. This architecture takes advantage of the attention mechanism in zooming into corresponding regions and is able to perform effective lesion detection and classification with high accuracy. The research thus hints that focusing on specific image regions can indeed improve diagnostic capabilities.

Based on the attention framework, Zhao et al. [70] further developed the BiRA-Net, a bilinear attention network for DR grading. Their work has shown the effectiveness of the attention mechanisms in distinguishing the different grades of diabetic retinopathy and that bilinear attention can improve the extraction of features by weighting the importance of different components within an image.

More recently, Madarapu et al. [71] proposed a deep integrative approach that combines channel-spatial attention with a self-attention mechanism for DR classification. This new model, apart from capturing spatial dependencies, infuses contextual information into the classification task. Their results show that synergistic employment of different attention strategies could significantly enhance the robustness and accuracy of models.

Alahmadi [72] proposed a Texture Attention Network specifically designed for DR classification. The model, by capturing texture features in the attention framework, effectively addresses texture-related discrepancies among images of

diabetic retinopathy and thus shows how texture information is important to improve classification performance.

He et al. [73] presented CABNet, which uses a Category Attention Block for class imbalance in the grading of DR. This model utilizes attention to focus on the underrepresented categories, improving the classification performance on different severity levels of diabetic retinopathy.

Yang et al. [74] further explored attention mechanisms with their GCA Attention Mechanism for classifying DR severity. Their work underscores the potential of attention-guided approaches to improve classification accuracy by focusing on critical image features that correlate with DR severity levels.

Luo et al. [75] presented MVDRNet, which combines DCNNs with attention mechanisms for multi-view diabetic retinopathy detection. In this method, the benefits of multiple views in the attention mechanism can be seen clearly; it enables the exploration of pathology.

Finally, In our work [76, 77], we proposed an Attention-Guided Convolutional Neural Network (AG-CNN) that uses Grad-CAM for the global and local classification of diabetic retinopathy (DR) severity, which helped greatly with the task.

### 3.7.3 Using deep/handcrafted feature fusion

The recent studies have shown growing interests in combining traditional handcrafted features with deep learning approaches for DR classification. This hybrid approach should, therefore, be able to exploit both domain expertise encoded by the handcrafted features and the automatic feature learning capabilities of deep neural networks.

Harangi et al. [78] introduced an automated screening method that combined

CNN architectures with traditional image processing features. The approach included AM-FM Based Image Level and Lesion Specific Feature Extraction, together with the deep features. They used the IDRiD dataset and Kaggle 2015 dataset for training and the ISBI 2018 data for testing. The hybrid model achieved an accuracy of 90.07% for the 5-class DR challenge, and 96.85% for the 3-class DME one, respectively, particularly improving on early-stage DR cases where subtle features become essential.

Bogacsovics et al. [79] attempted to enhance the performance of CNNs at strategic positions by incorporating handcrafted features. Quality metrics, vessel-based features, and microaneurysm detection results provided complementary information to deep features in their work. They used the same handcrafted feature as Harangi et al. [78] They used IDRiD, Kaggle 2015 and the MISSIDOR dataset for training and the IDRiD dataset for testing. This hybrid model achieved an accuracy of 87.4%.

Ishtiaq et al. [80] proposed a hybrid method which utilized the ensemble of optimized CNNs combined with texture features. Their approach was an ensemble of Three methods were used for feature extraction, namely, Local Binary Patterns (LBP) for texture-oriented features, and the novel GraphNet124 and ResNet50 for CNN-based features. Their methodology achieved an accuracy of 98.85% on Kaggle EYEPACS data. The ensemble approach helped in avoiding overfitting and improved generalization.

Al-Shahrani [81] proposed a hybrid approach for DR stage classification by including a variety of feature extraction techniques. The method combined the features of GLCM, LBP and FCH texture with deep features extracted from pre-trained CNN models (GoogleNet and ResNet-18). The combined features obtained an accuracy of 99.7% on the Kaggle 2015 dataset using an ensemble clas-

sifier to fuse the features, hence proving the feature level fusion approach.

Shamsan et al. [82] have developed a broad framework that uses both deep and handcrafted features in the prediction of DR stage. Their approach is based on the extraction of texture features using GLCM, DWT, LBP and FCH, and deep features extracted using the DenseNet-121 as well as the AlexNet models. This feature merged approach indeed obtained 99.1% accuracy for multi-class DR classification on the Kaggle 2015 dataset and proved very effective, thus offering statistical improvement compared to the use of either feature type alone.

The accuracy rates across these studies vary from 87.4% to 99.7%. In all of these studies, the hybrid approaches perform superiorly compared to traditional single-feature methods, and hence, the combination of domain expertise through handcrafted features with deep learning's automatic feature extraction capabilities is a very promising direction for DR classification systems. There is an improving trend over time, with higher metrics achieved for more recent studies and increasingly sophisticated fusion strategies.

Although much progress has been made in the classification of diabetic retinopathy through the fusion of several features, there is still a significant gap in the literature on the APTOS 2019 DR dataset. That will be particularly useful because this dataset contains a rich and complex set of fundus images representing multiple stages of DR, hence creating a very strong base for training a classification model and testing its results. The limited explorations in the previous works on this dataset make it an opportunity to investigate its peculiarities and its possible improvements for classification performance. By choosing to do our work on the APTOS 2019 dataset, we mean to have a look at the combination of handcrafted and deep features effectiveness in a more representative and varied situation. Our objective is to demonstrate, for this dataset, what performance improvement can

be achieved, thus contributing to further refinement in automatic diabetic retinopathy screening methodologies for clinical outcome improvement.

The accuracy rates across these studies vary from 87.4% to 99.7%. In all of these studies, the hybrid approaches perform superiorly compared to traditional single-feature methods, and hence, the combination of domain expertise through handcrafted features with deep learning's automatic feature extraction capabilities is a very promising direction for DR classification systems. There is an improving trend over time, with higher metrics achieved for more recent studies and increasingly sophisticated fusion strategies.

### **3.8 Conclusion**

In summary, deep learning has shown remarkable promise in detecting diabetic retinopathy, offering a scalable solution for early diagnosis and intervention. While challenges such as model robustness and accessibility remain, advancements in this field hold the potential to significantly reduce DR-related vision loss worldwide.

## 4 | Attention guided Convolutional Neural Networks for Diabetic Retinopathy classification

### 4.1 Introduction

Attention-guided convolutional neural networks (AG-CNNs) offer a refined approach to diabetic retinopathy (DR) detection, focusing on relevant retinal regions to improve classification accuracy. This chapter introduces the principles of attention-guided CNNs and examines how they enhance both detection and grading of DR, providing a robust tool for early diagnosis.

### 4.2 Methods

#### 4.2.1 DenseNet-121 model

DenseNet [83] is a convolutional neural network that uses dense connections between layers. These connections are made through Dense Blocks, which connect all layers directly if their feature-map sizes are the same. To keep the feed-forward nature, each layer gets more information from all the layers before it and sends its feature maps to all the layers afterwards; this architecture is displayed in Figure 4.1, while its structure with all the layers is presented in Table 4.1.

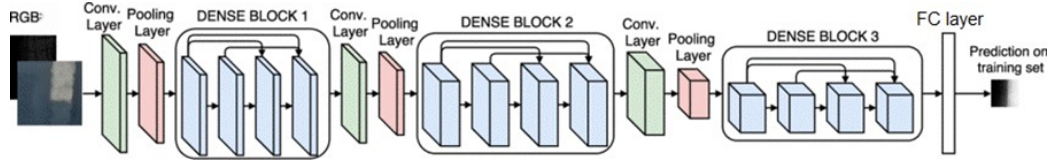


Figure 4.1: Illustration of the DenseNet-121 architecture

Table 4.1: DenseNet-121 summary

| Layer                | DenseNet-121                              |
|----------------------|-------------------------------------------|
| Convolution          | $7 \times 7$ conv, stride 3               |
| Pooling              | $3 \times 3$ max pool, stride 2           |
| Dense Block (1)      | $1 \times 1$ conv $\times 6$              |
|                      | $3 \times 3$ conv                         |
| Transition Layer (1) | $1 \times 1$ conv                         |
|                      | $2 \times 2$ average pool                 |
| Dense Block (2)      | $1 \times 1$ conv $\times 12$             |
|                      | $3 \times 3$ conv                         |
| Transition Layer (2) | $1 \times 1$ conv                         |
|                      | $2 \times 2$ average pool                 |
| Dense Block (3)      | $1 \times 1$ conv $\times 12$             |
|                      | $3 \times 3$ conv                         |
| Transition Layer (3) | $1 \times 1$ conv                         |
|                      | $2 \times 2$ average pool                 |
| Dense Block (4)      | $1 \times 1$ conv $\times 12$             |
|                      | $3 \times 3$ conv                         |
| Classification Layer | $7 \times 7$ global average pool          |
|                      | 14-D fully-connected, elementwise sigmoid |

## 4.2.2 Laplacian of Gaussian (LoG)

The second spatial derivative of an image's Laplacian provides a 2-D isotropic measure, highlighting areas with sudden intensity changes, making it useful for edge detection. To reduce noise sensitivity, the Laplacian is often applied after

Gaussian smoothing [84]. This method detects edges by locating zero crossings in the filtered image, dividing regions of varying intensity. The Laplacian  $L(x, y)$  of an image with pixel intensity values  $I(x, y)$  is given by:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (11)$$

The typical small kernel is displayed in Figure 4.2, while Figure 4.3 shows the input and output of the LoG technique. The 2-D LoG function with a Gaussian standard deviation and a zero center takes the following form:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (12)$$

|     |     |     |
|-----|-----|-----|
| - 1 | 2   | - 1 |
| 2   | - 4 | 2   |
| - 1 | 2   | - 1 |

Figure 4.2: Standard LoG kernel

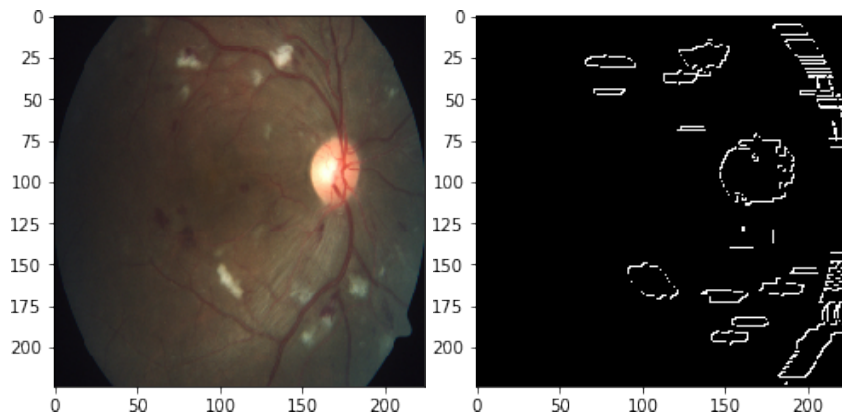


Figure 4.3: Applying LoG filter: Left: original image, Right: filtered image

We note here that we used a sigma equal to 3 in the Laplacian of Gaussian filtering.

### 4.2.3 Contrast Limited Adaptive Histogram Equalization

An image histogram is a representation of the image intensity value. The primary purpose of the histogram is to provide an image's statistical data. This explains why we can adjust the histogram to perform image improvement. Histogram equalization is a widely used technique for image enhancement since it is straightforward and computationally light. We improve the color fundus picture using Contrast Limited Adaptive Histogram Equalization (CLAHE). This preprocessing technique is commonly utilized in ophthalmology to enhance the visibility of the issues present. One essential aspect of a fundus image pertains to the contrast of blood vessels. Image contrast results from combining a range of intensity levels and distinguishing between the full spectrum of pixel values from the highest to the lowest. The application of histogram manipulation for image enhancement aims to achieve a uniform distribution of intensity. In low-contrast images, the effective intensity range is limited. However, histogram equalization expands the intensity distribution, effectively altering the original image's intensity. An example of the input and output of the CLAHE method is shown in Figure 4.4.

Figure 4.5 depicts the flowchart for the fundus image enhancement method suggested in this study. As an input image, we employ an RGB color retinal image. The color picture channel is split in the first step, leaving us with three images: R, G, and B image channels. Only the G channel is used in the picture enhancement procedure utilizing CLAHE because it contains more significant structural information about blood vessels than the other channels. The improved G channel image follows. The subsequent step combines the three image channels (R, enhanced G, and R). After this procedure, we have an improved color retinal

image.

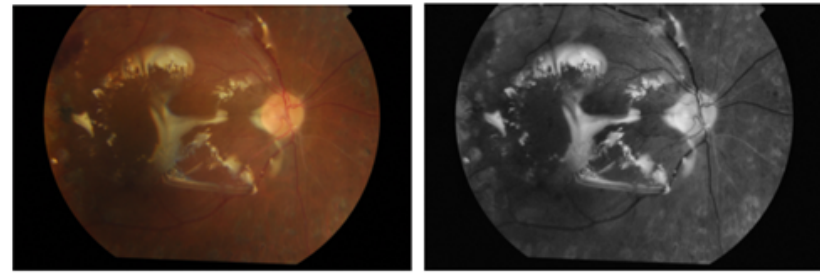


Figure 4.4: The effect of the CLAHE method: left: input, right: output.

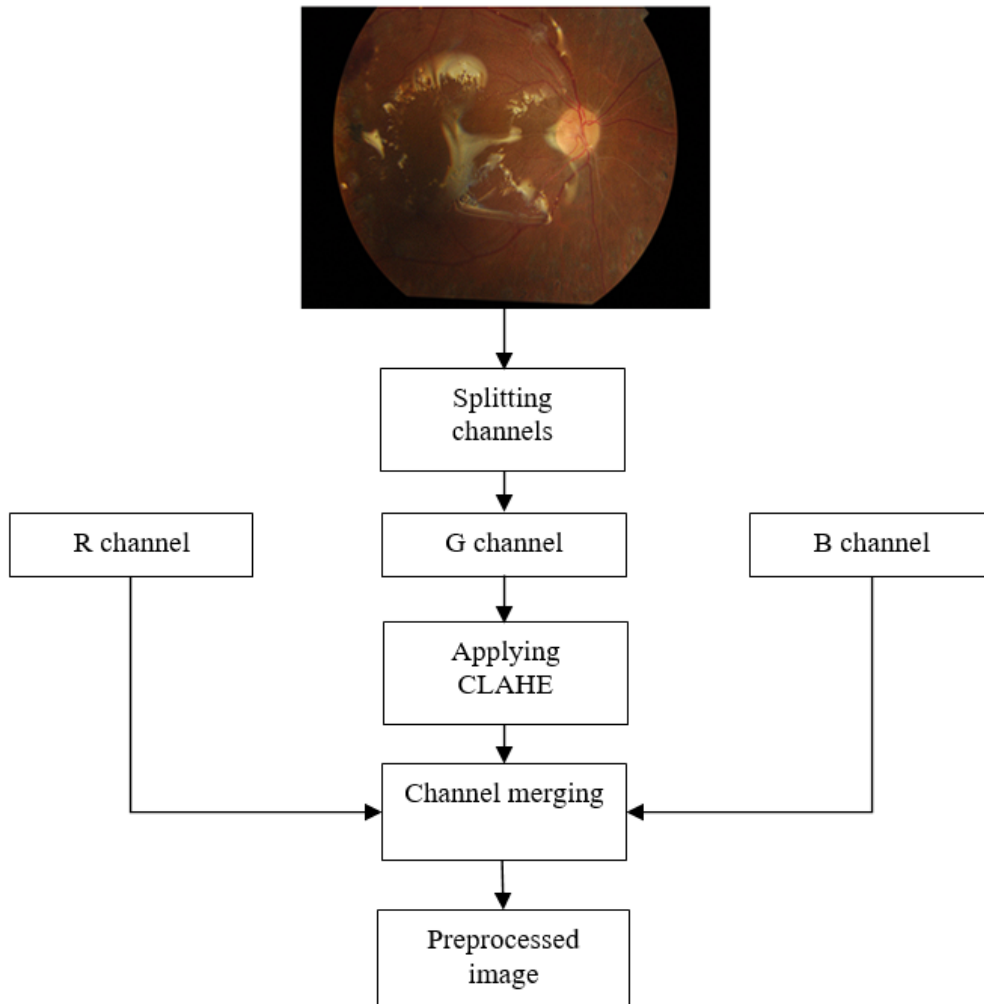


Figure 4.5: Flowchart of the CLAHE technique

#### 4.2.4 Grad-CAM

Selvaraju et al. [85] developed a method called Gradient Weighted Class Activation Mapping (Grad-CAM), which offers a clear understanding of deep learning models. Any highly linked neural network may be graphically described using Grad-CAM, further facilitating model understanding during the prediction phase. Illustrated in Figure 4.6, Grad-CAM utilizes an input image to make predictions

using the proposed model. The model generates the output class based on its forecast, and any one of the Convolution layers is then subjected to the Grad-CAM algorithm. The last Conv layer is often utilized for Grad-CAM. Figure 4.7 show some examples of the results of the Grad-CAM algorithm, The steps of the Grad-CAM algorithm are:

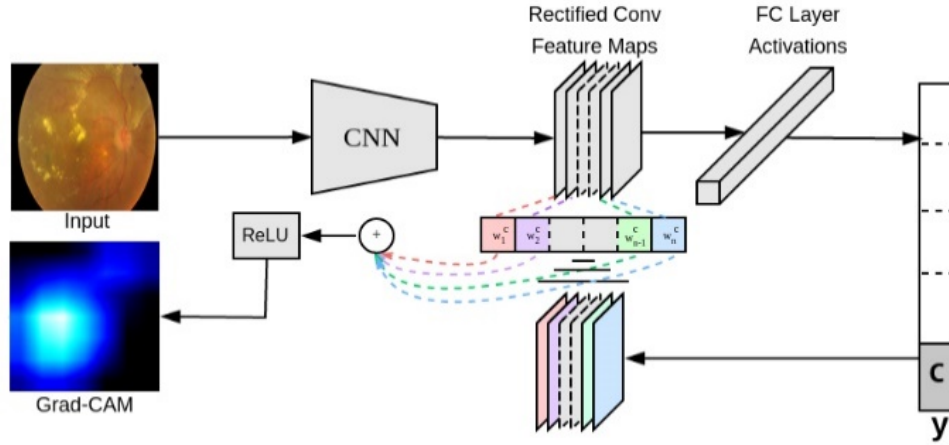


Figure 4.6: Explanatory diagram of the Gradient Weighted Class Activation Mapping (Grad- CAM).

- Acquire the final result of the network after completing the convolution phase.
- Calculate the gradient of the score for class  $c$ , denoted as  $Y_c$ , with respect to  $A_k$ , which represents the activation maps of a conv layer.

$$\frac{\partial Y^c}{\partial A^k} \quad (13)$$

- By utilizing the global average pooling technique, emphasize the importance of each activation map  $k$  for specific classes:

$$\alpha_c^k = 1/Z \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \quad (14)$$

Where  $Z$  is the number of pixels in the activation map in this context, the

“partial differentials” correspond to the gradients computed during the backpropagation process, while the “summation over i and j” represents the global average pooling operation.

- Generate a heatmap by merging activation maps with weights and applying the ReLU function to the resulting values.

$$L_{GRAD-CAM}^c = RELU \sum_k \alpha_c^k A^k \quad (15)$$

ReLU is the preferred activation function because it accentuates characteristics that are beneficial for the specific class being considered. ROIs are identified as pixels whose intensity are dependent on the gradient  $Y_c$ . Without ReLU, localization maps might contain non-pos pixels associated with other image classes, negatively impacting localization accuracy.

- Normalize heatmap (optional but advised step since it enhances outcomes).

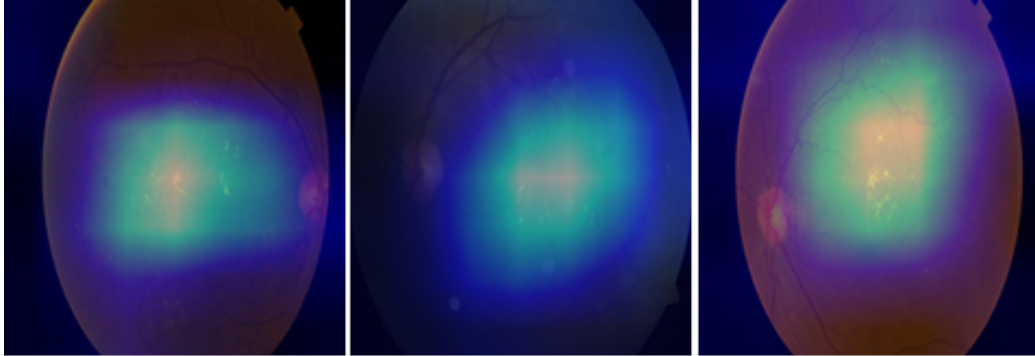


Figure 4.7: Extracted heatmaps superposed on their corresponding input images.

### 4.3 Proposed system

In this section, we describe the proposed Attention-guided system designed for classifying eye fundus images based on the DenseNet-121 model as a backbone.

The overall attention-guided structure is illustrated in Figure 4.8. The mask inference method for discriminative area detection is covered in section 4.3.3. The approach involves two main stages: global and local. The input images are run through a preprocessing step and then are fed to dense convolutional blocks followed by a 1x1 convolution block and a pooling layer; these two are transition blocks. At the end, the blocks are connected to a max-pooling layer, a fully connected (FC) layer, and a sigmoid layer. Contrary to the global stage, the local stage's input is local images cropped by the mask generated from the global stage. The input image is overlaid over the heat map for visualization purposes.

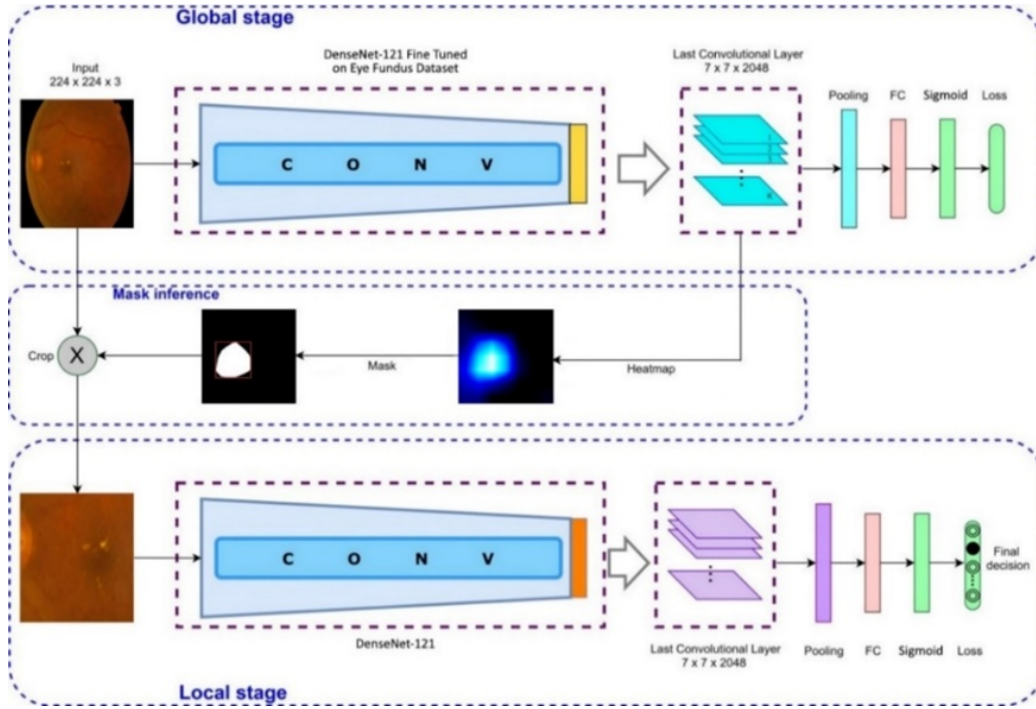


Figure 4.8: Proposed system structure.

### 4.3.1 Multilabel setup

In this classification model, we treat the problem as a multilabel scenario. Each image is assigned a 5-bit class vector representing different severity levels  $L = [l_1,$

12, 13, 14, 15]. Here,  $l \in \{0, 1\}$ , where 1 indicates that the severity level has been reached, and 0 suggests it hasn't. Therefore, if a sample belongs to a specific class (e.g., class 3), it is also considered to belong to all the classes below it (0, 1, and 2). This approach is grounded in the notion that once an eye attains severity levels 1 and 2, it will inevitably advance to level 3.

### 4.3.2 Global and local stages

The global input image gives the global stage access to pertinent information. The DenseNet- 121 served as the primary global model in this study. It comprises three dense blocks and three transition layers, followed by a global average pooling layer and an FC layer for classification. Finally, the output vector  $Z_g(I_c)$  is normalized by applying a sigmoid layer using:

$$\tilde{p}_g(c|I) = \frac{1}{1 + e^{-p(c|I)}} \quad (16)$$

Where  $I$  is the global image,  $pg(c|I)$  indicates the probability score of  $I$  fitting to the  $c$ th class,  $c \in \{1, 2, \dots, C\}$  and  $C$  represents the total number of classes. Conversely, the local branch focuses on the relevant area and is expected to mitigate the disadvantages of only using the global image. More specifically, the local branch shares the same convolutional network structure as the global branch. But do not share weights, as they serve different purposes. The probability score of a local branch is denoted by  $pl(c|I_c)$  where  $I_c$  is the input image of the local branch. We perform the same normalization and optimization as the global branch.

### 4.3.3 Mask inference and ROIs union

In order to isolate the region of interest (ROI) in the global DR images, we build a binary mask by performing thresholding operations on feature maps (ROI

heatmap). These heatmaps are extracted using the Grad-CAM technique.

After that, the heatmap is binarized according to a threshold called  $\tau$ . We created a bounding box around it, then we cropped it out of the original image, this operation is shown in Figure 4.9. For some DR images where more than one region of interest is created, we apply a union of these ROIs and the resulted bounding box will be cropped from the original image, this is shown in Figure 4.10.

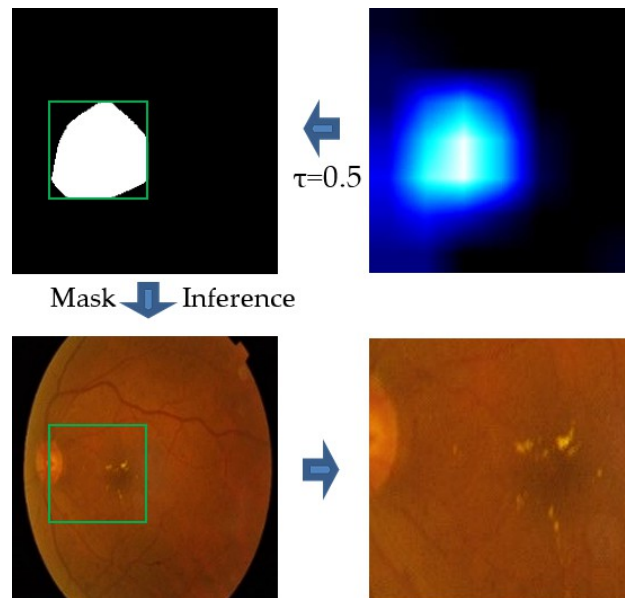


Figure 4.9: ROI cropping process using mask inference.

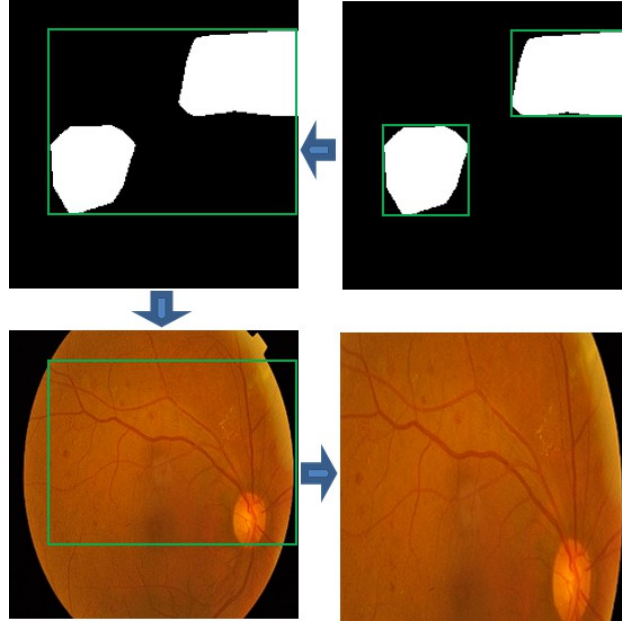


Figure 4.10: Bounding boxes union in case of multiple ROIs.

#### 4.3.4 Training strategy of AG-CNN

This approach adopts a two-stage training scheme for AG-CNN:

**Stage I:** The global branch network pre-trained by ImageNet is finetuned using the global images. After this stage, the weights of the global branch are fixed.

**Stage II:** Local images are obtained using mask inference with threshold  $\tau$ , then we feed it to the local branch for finetuning and normalizing, and we get our final classification results at the output of this branch.

In each stage, we use the model with the highest accuracy/AUC on the validation set for testing. Variants of training strategy may influence the performance of AG-CNN. We discussed it in the next section.

## 4.4 Results and discussion

To investigate the applicability of the proposed approach, we evaluated its performance using the APTOS 2019 diabetic retinopathy dataset. Series of experiments were conducted to evaluate our approach with and without data augmentation. Besides, several preprocessing techniques were considered in order to enhance the DR image quality. For performance assessment, we used a set of measures namely the accuracy rate, the area under ROC curve (AUC), the recall rate, the precision rate and the f1-score (F1).

$$Acc = (TP + TN)/(TP + FP + FN + TN) \quad (17)$$

$$2 * \frac{precision * recall}{precision + recall} \quad (18)$$

$$Recall = (TP)/(TP + FN) \quad (19)$$

$$Precision = (TP)/(TP + FP) \quad (20)$$

Where TP, TN, FP and FN:

- True Positive (TP): The number of positive samples correctly classified as positive.
- True Negative (TN): The number of negative samples correctly classified as negative.
- False Positive (FP): The number of negative samples incorrectly classified as positive.
- False Negative (FN): The number of positive samples incorrectly classified as negative.

These values are extracted from the confusion matrix as follows in figure 4.11:

|        |   | Predicted |    |
|--------|---|-----------|----|
|        |   | 0         | 1  |
| Actual | 0 | TN        | FP |
|        | 1 | FN        | TP |

Figure 4.11: Confusion matrix

#### 4.4.1 Protocols and Setups

During all experiments, the evaluation protocol of the APTOS 2019 database takes 85% of the dataset for training and 15% as the validation set. Experiments with and without data augmentation and different preprocessing strategies are established on the dataset classification. We have merged multiple transformation steps to change the original images and produce other variations in this work. The fundamental transformations applied to each training image in the gallery are:

- Resize the image to fit the model input ( $224 \times 224 \times 3$ ).
- The random image zooms up to 15%.
- Flip the image vertically and horizontally at random.

#### 4.4.2 Experiments

To investigate the applicability of the proposed approach, we evaluated its performance using the APTOS 2019 diabetic retinopathy dataset. We start by justifying the choice of the DenseNet architecture over others, where we conducted an experiment using the most used CNN models without preprocessing. The results

of this experiment are displayed in Table 4.2. Upon analyzing these results, we determine that the DenseNet-121 model outperforms other models on this dataset. As a result, the DenseNet-121 was chosen in this work as the backbone CNN model to explore if Attention-guided CNN can improve the system performance.

Table 4.2: Evaluation of Multiple Models trained using the APTOS 2019 Dataset

| Model               | Accuracy %   | AUC %        | F1 %         | Val loss %   | Precision %  | Recall %     |
|---------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| <b>DenseNet-121</b> | <b>96.18</b> | <b>98.99</b> | <b>95.32</b> | <b>0.112</b> | <b>96.90</b> | <b>93.85</b> |
| MobileNet v2        | 89.31        | 89.86        | 85.22        | 1.424        | 99.64        | 74.57        |
| Resnet50            | 96.07        | 98.67        | 95.53        | 0.131        | 97.69        | 93.53        |
| Inception V3        | 95.71        | 98.39        | 94.81        | 0.154        | 96.53        | 93.22        |
| VGG16               | 95.85        | 99.13        | 95.02        | 0.110        | 94.68        | 95.42        |

After that, we perform a basic data augmentation for training by scaling the original images to 224 x 224, random flipping (horizontal and vertical) and zooming with a range of 15%. The batch size of 64 has been chosen for network optimization. Each branch receives 25 epochs of training. The learning rate is 0.0001 at first. We employ a weight decay of 50%. The same process is used during validation and test. We build AG-CNN with the TensorFlow Keras framework.

By analyzing the potential of each thresholding value, we evaluated the performance of the proposed AG-CNN architecture. We also evaluate the performance of both global and local branches with and without preprocessing.

#### 4.4.2.1 Scenario 1

The first scenario consists of training the system (global and local) with the original dataset without CLAHE or LoG. The only transformations applied are the basic ones mentioned earlier. Table 4.3 shows the resulting metrics of the global and local branches using this first scenario, while Fig. 4.12 represents the ROC curve of the best-performing architecture.

Table 4.3: Evaluation metrics for the scenario 1 experiment

| Branch                  | Threshold             | Score         |               |               |              |               |               |
|-------------------------|-----------------------|---------------|---------------|---------------|--------------|---------------|---------------|
|                         |                       | Accuracy      | AUC           | F1            | Loss         | Precision     | Recall        |
| Global                  | DenseNet-121          | 96.18%        | 98.99%        | 95.32%        | 0.112        | 96.90%        | 93.85%        |
| <b>Local (Cropping)</b> | $\tau = \mathbf{0.1}$ | <b>96.44%</b> | <b>99.38%</b> | <b>95.66%</b> | <b>0.097</b> | <b>96.35%</b> | <b>95.03%</b> |
|                         | $\tau = 0.2$          | 96.25%        | 98.85%        | 95.47%        | 0.127        | 97.26%        | 93.78%        |
|                         | $\tau = 0.3$          | 95.92%        | 98.92%        | 95.03%        | 0.131        | 96.77%        | 93.40%        |
|                         | $\tau = 0.4$          | 96.18%        | 99.08%        | 95.37%        | 0.115        | 96.00%        | 94.80%        |
|                         | $\tau = 0.5$          | 95.96%        | 98.77%        | 95.11%        | 0.133        | 96.79%        | 93.55%        |
|                         | $\tau = 0.6$          | 96.14%        | 95.28%        | 97.29%        | 0.114        | 97.69%        | 93.05%        |
|                         | $\tau = 0.7$          | 96.00%        | 98.51%        | 95.09%        | 0.140        | 97.40%        | 92.95%        |
|                         | $\tau = 0.8$          | 95.67%        | 98.81%        | 94.71%        | 0.130        | 96.49%        | 93.08%        |
|                         | $\tau = 0.9$          | 95.30%        | 98.71%        | 94.26%        | 0.137        | 96.74%        | 91.95%        |

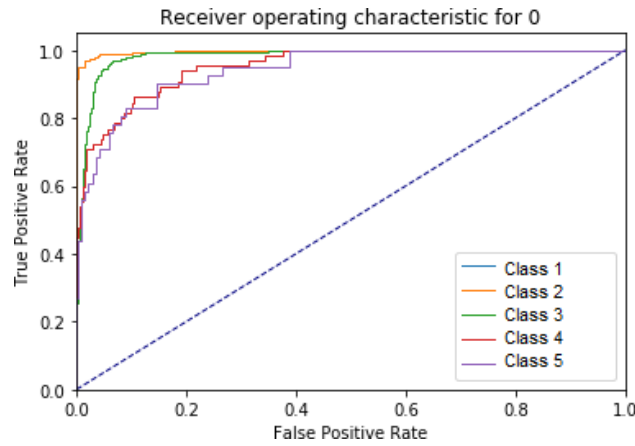


Figure 4.12: ROC curve of the best performing branch/threshold iteration of scenario 1.

#### 4.4.2.2 Scenario 2

For the second scenario, three new datasets are generated using CLAHE and LoG, as shown in figure 4.13. These new datasets are used for the training of this second scenario. Tables 4.4, 4.5 and 4.6 show the resulting validation metrics of the global/local branches corresponding to CLAHE, LoG and CLAHE+LoG preprocessed data, respectively. Figure 4.14 represents the ROC curves of the best-performing iteration for every preprocessing method used in this scenario.

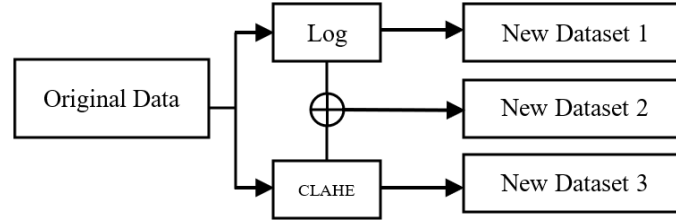


Figure 4.13: Diagram of the new datasets generation.

Table 4.4: Evaluation metrics for the CLAHE preprocessing process

| Branch           | Threshold          | Score         |               |               |              |               |               |
|------------------|--------------------|---------------|---------------|---------------|--------------|---------------|---------------|
|                  |                    | Accuracy      | AUC           | F1            | Loss         | Precision     | Recall        |
| <b>Global</b>    | <b>DenseNet121</b> | <b>96.51%</b> | <b>99.22%</b> | <b>95.77%</b> | <b>0.105</b> | <b>96.73%</b> | <b>94.87%</b> |
| Local (Cropping) | $\tau = 0.1$       | 96.18%        | 98.89%        | 95.33%        | 0.112        | 96.56%        | 94.22%        |
|                  | $\tau = 0.2$       | 95.74%        | 98.55%        | 94.79%        | 0.144        | 96.17%        | 93.55%        |
|                  | $\tau = 0.3$       | 95.89%        | 98.79%        | 95.03%        | 0.123        | 96.07%        | 94.05%        |
|                  | $\tau = 0.4$       | 95.70%        | 98.53%        | 94.79%        | 0.146        | 96.17%        | 93.50%        |
|                  | $\tau = 0.5$       | 95.67%        | 98.87%        | 94.66%        | 0.128        | 96.13%        | 93.33%        |
|                  | $\tau = 0.6$       | 95.38%        | 98.82%        | 94.33%        | 0.134        | 95.79%        | 93.00%        |
|                  | $\tau = 0.7$       | 95.27%        | 98.94%        | 94.24%        | 0.121        | 94.23%        | 94.35%        |
|                  | $\tau = 0.8$       | 95.01%        | 98.74%        | 93.94%        | 0.148        | 93.60%        | 94.32%        |
|                  | $\tau = 0.9$       | 94.50%        | 98.48%        | 93.32%        | 0.157        | 93.59%        | 93.13%        |

Table 4.5: Evaluation metrics for the LoG preprocessing approach

| Branch           | Threshold          | Score         |               |               |       |               |               |
|------------------|--------------------|---------------|---------------|---------------|-------|---------------|---------------|
|                  |                    | Accuracy      | AUC           | F1            | Loss  | Precision     | Recall        |
| <b>Global</b>    | <b>DenseNet121</b> | 92.80%        | 97.20%        | 90.97         | 0.198 | 95.31%        | 87.09%        |
| Local (Cropping) | $\tau = 0.1$       | 92.98%        | 97.29%        | 91.16%        | 0.203 | 96.04%        | 86.81%        |
|                  | $\tau = 0.2$       | <b>93.27%</b> | <b>97.69%</b> | <b>91.60%</b> | 0.181 | <b>93.91%</b> | <b>89.51%</b> |
|                  | $\tau = 0.3$       | 92.47%        | 97.07%        | 90.55%        | 0.208 | 94.52%        | 86.96%        |
|                  | $\tau = 0.4$       | 92.65%        | 97.21%        | 90.63%        | 0.201 | 95.86%        | 86.01%        |
|                  | $\tau = 0.5$       | 92.58%        | 96.94%        | 90.66%        | 0.211 | 94.52%        | 87.19%        |
|                  | $\tau = 0.6$       | 92.54%        | 96.91%        | 90.65%        | 0.209 | 95.21%        | 86.59%        |
|                  | $\tau = 0.7$       | 92.18%        | 96.29%        | 90.23%        | 0.238 | 93.90%        | 86.92%        |
|                  | $\tau = 0.8$       | 91.89%        | 96.17%        | 89.67%        | 0.242 | 93.93%        | 85.83%        |
|                  | $\tau = 0.9$       | 89.56%        | 95.12%        | 87.15%        | 0.275 | 88.43%        | 85.99%        |

Table 4.6: Evaluation metrics for the CLAHE+LoG preprocessing approach

| Branch                  | Threshold             | Score         |               |               |              |               |               |
|-------------------------|-----------------------|---------------|---------------|---------------|--------------|---------------|---------------|
|                         |                       | Accuracy      | AUC           | F1            | Loss         | Precision     | Recall        |
| Global                  | DenseNet121           | 93.49%        | 98.01%        | 91.91%        | 0.177        | 95.81%        | 88.42%        |
| <b>Local (Cropping)</b> | $\tau = 0.1$          | 92.65%        | 97.82%        | 91.32%        | 0.194        | 90.11%        | 92.66%        |
|                         | $\tau = \mathbf{0.2}$ | <b>92.95%</b> | <b>97.81%</b> | <b>91.29%</b> | <b>0.182</b> | <b>94.25%</b> | <b>88.57%</b> |
|                         | $\tau = 0.3$          | 92.76%        | 97.61%        | 91.12%        | 0.187        | 93.78%        | 88.70%        |
|                         | $\tau = 0.4$          | 92.62%        | 97.37%        | 90.92%        | 0.198        | 93.18%        | 88.85%        |
|                         | $\tau = 0.5$          | 92.25%        | 97.19%        | 90.19%        | 0.219        | 94.87%        | 86.01%        |
|                         | $\tau = 0.6$          | 92.87%        | 97.24%        | 91.22%        | 0.203        | 92.64%        | 89.96%        |
|                         | $\tau = 0.7$          | 92.61%        | 97.48%        | 90.79%        | 0.205        | 94.24%        | 87.68%        |
|                         | $\tau = 0.8$          | 92.15%        | 97.06%        | 90.42%        | 0.220        | 91.43%        | 89.50%        |
|                         | $\tau = 0.9$          | 90.55%        | 96.02%        | 88.36%        | 0.253        | 90.55%        | 86.45%        |

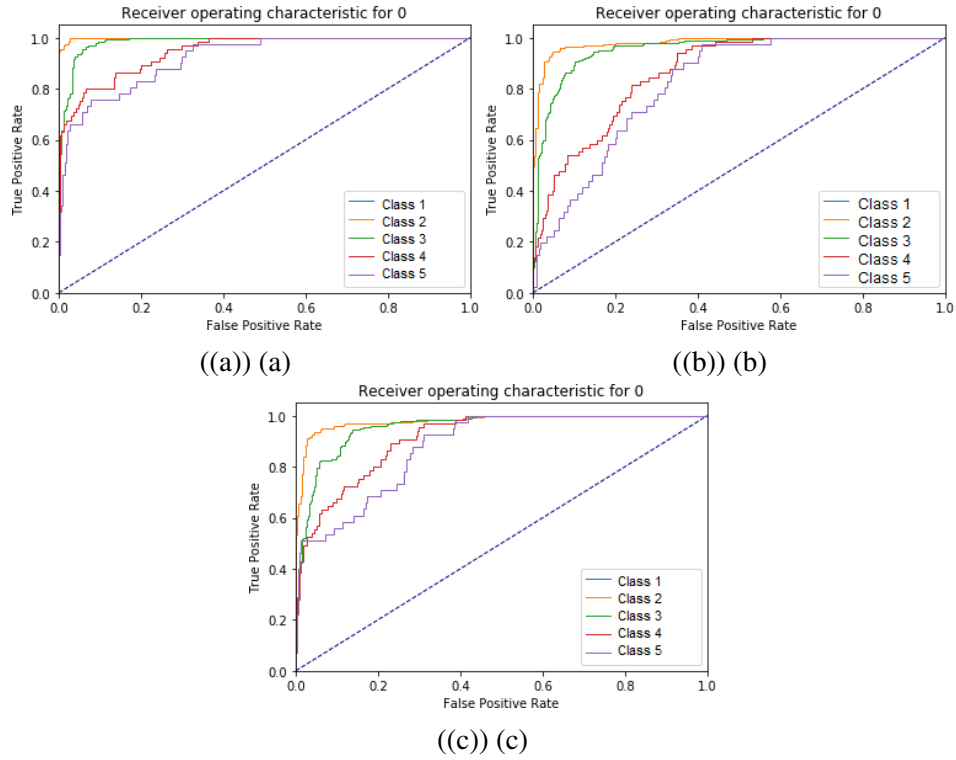


Figure 4.14: ROC curves of the best performing branch/threshold in scenario 2 with different preprocessing techniques: a) CLAHE, b) LoG, and c) CLAHE+LoG.

#### 4.4.2.3 Scenario 3

Data class imbalance can lead to a loss in deep learning performance. For this reason, in this scenario, we implemented a data balance algorithm on the same datasets generated in the previous scenario by applying some transformations (rotation with  $-/+ 45^\circ$ , horizontal flip, and brightness with range from 0.7 to 1.3). Consequently, the sample number for classes 'Mild DR', 'Moderate DR', 'Severe DR' and 'Proliferative DR' become 1730, 1740, 1710 and 1720, respectively. We recall here that we kept the number of samples in class 'No DR' without change (1805 samples).

Tables 4.7, 4.8 and 4.9 show the results of this scenario with the three preprocessing methods used above. Meanwhile, Figs. 4.15 and 4.16 illustrate the ROC curves of each preprocessing technique for the global branch and also for the local branch with best threshold that achieved the highest performance.

After examining the results of this scenario, we noticed that data balancing didn't improve much in the performance of the model. In some cases, it was a bit worse due to the class imbalance that can be considered as a natural distribution of the diabetic retinopathy. As a result of the performance of this scenario, we continue the rest of the experiments without performing the data balancing.

#### 4.4.2.4 Scenario 4

In this scenario, another boost for datasets is implemented to generate two new datasets by merging the original dataset with two modified ones (using CLAHE and LoG). Figure 4.17 shows the process of creating the two new datasets. Tables 4.10 and 4.11 show the resulting validation metrics of the system using this scenario, while Fig.4.18 shows the ROC curves of these results.

Table 4.7: CLAHE preprocessing effect with data balancing

| Branch           | Threshold          | Score         |               |               |            |               |               |
|------------------|--------------------|---------------|---------------|---------------|------------|---------------|---------------|
|                  |                    | Accuracy      | AUC           | F1            | Loss       | Precision     | Recall        |
| <b>Global</b>    | <b>DenseNet121</b> | <b>96.51%</b> | <b>99.38%</b> | <b>96.47%</b> | <b>0.1</b> | <b>96.42%</b> | <b>96.56%</b> |
| Local (Cropping) | $\tau = 0.1$       | 96.12%        | 98.54%        | 95.01%        | 0.102      | 96.33%        | 95.73%        |
|                  | $\tau = 0.2$       | 95.64%        | 98.27%        | 94.84%        | 0.124      | 95.61%        | 94.12%        |
|                  | $\tau = 0.3$       | 95.59%        | 98.42%        | 95.23%        | 0.118      | 95.49%        | 94.63%        |
|                  | $\tau = 0.4$       | 94.67%        | 98.86%        | 94.44%        | 0.14       | 94.89%        | 94.03%        |
|                  | $\tau = 0.5$       | 95.94%        | 98.21%        | 94.83%        | 0.108      | 94.57%        | 94.14%        |
|                  | $\tau = 0.6$       | 95.27%        | 98.85%        | 94.08%        | 0.146      | 94.05%        | 94.16%        |
|                  | $\tau = 0.7$       | 95.25%        | 98.27%        | 94.17%        | 0.105      | 94.06%        | 94.30%        |
|                  | $\tau = 0.8$       | 95.84%        | 98.13%        | 93.64%        | 0.119      | 93.44%        | 93.87%        |
|                  | $\tau = 0.9$       | 93.38%        | 98.68%        | 93.74%        | 0.16       | 93.6%         | 93.07%        |

Table 4.8: LoG preprocessing effect with data balancing

| Branch           | Threshold          | Score         |               |               |            |               |               |
|------------------|--------------------|---------------|---------------|---------------|------------|---------------|---------------|
|                  |                    | Accuracy      | AUC           | F1            | Loss       | Precision     | Recall        |
| <b>Global</b>    | <b>DenseNet121</b> | <b>91.89%</b> | <b>97.49%</b> | <b>91.54%</b> | <b>0.2</b> | <b>92.57%</b> | <b>90.65%</b> |
| Local (Cropping) | $\tau = 0.1$       | 91.21%        | 97.08%        | 90.76%        | 0.219      | 93.79%        | 88.02%        |
|                  | $\tau = 0.2$       | 90.96%        | 97.03%        | 90.4%         | 0.219      | 93.26%        | 87.75%        |
|                  | $\tau = 0.3$       | 90.66%        | 96.92%        | 89.97%        | 0.232      | 93.51%        | 86.73%        |
|                  | $\tau = 0.4$       | 90.96%        | 96.68%        | 90.36%        | 0.233      | 92.73%        | 88.18%        |
|                  | $\tau = 0.5$       | 90.51%        | 96.83%        | 89.83%        | 0.235      | 94.16%        | 85.93%        |
|                  | $\tau = 0.6$       | 89.91%        | 96.31%        | 89.36%        | 0.243      | 91.02%        | 87.81%        |
|                  | $\tau = 0.7$       | 88.69%        | 94.86%        | 87.34%        | 0.307      | 94.45%        | 81.29%        |
|                  | $\tau = 0.8$       | 87.86%        | 94.42%        | 86.64%        | 0.294      | 92.29%        | 81.74%        |
|                  | $\tau = 0.9$       | 85.96%        | 93.05%        | 84.32%        | 0.327      | 90.01%        | 79.46%        |

Table 4.9: CLAHE+LoG preprocessing effect with data balancing

| Branch           | Threshold          | Score         |               |               |              |               |               |
|------------------|--------------------|---------------|---------------|---------------|--------------|---------------|---------------|
|                  |                    | Accuracy      | AUC           | F1            | Loss         | Precision     | Recall        |
| <b>Global</b>    | <b>DenseNet121</b> | <b>91.61%</b> | <b>97.27%</b> | <b>91.00%</b> | <b>0.204</b> | <b>92.32%</b> | <b>90.25%</b> |
| Local (Cropping) | $\tau = 0.1$       | 91.02%        | 97.40%        | 90.62%        | 0.218        | 93.49%        | 87.97%        |
|                  | $\tau = 0.2$       | 91.15%        | 97.11%        | 90.90%        | 0.215        | 93.32%        | 87.68%        |
|                  | $\tau = 0.3$       | 90.71%        | 97.01%        | 89.89%        | 0.224        | 93.63%        | 86.70%        |
|                  | $\tau = 0.4$       | 91.03%        | 96.78%        | 90.23%        | 0.229        | 92.80%        | 88.16%        |
|                  | $\tau = 0.5$       | 90.61%        | 96.91%        | 89.76%        | 0.231        | 94.13%        | 85.92%        |
|                  | $\tau = 0.6$       | 89.88%        | 96.39%        | 89.33%        | 0.241        | 91.10%        | 87.77%        |
|                  | $\tau = 0.7$       | 88.72%        | 94.90%        | 87.32%        | 0.304        | 94.39%        | 81.27%        |
|                  | $\tau = 0.8$       | 87.87%        | 94.41%        | 86.61%        | 0.293        | 92.30%        | 81.76%        |
|                  | $\tau = 0.9$       | 85.98%        | 93.03%        | 84.29%        | 0.325        | 90.02%        | 79.45%        |

## Chapter 4: Attention guided Convolutional Neural Networks for Diabetic Retinopathy classification

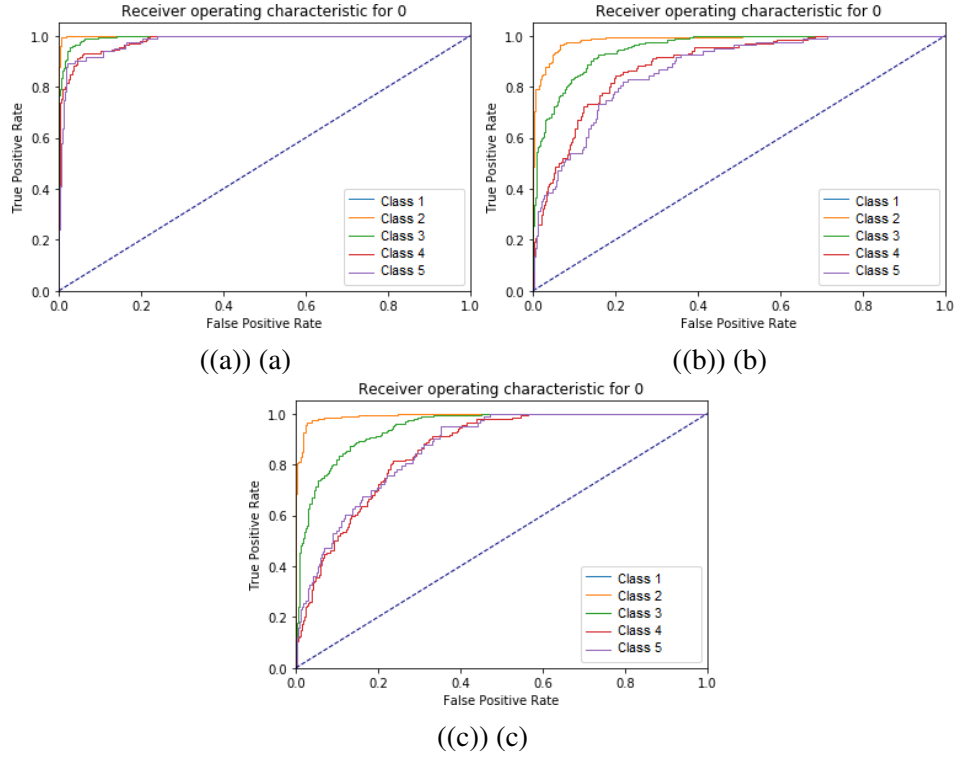


Figure 4.15: ROC curves of the scenario 3 with data balancing and three different prepro- cessing: a) CLAHE, b) LoG, and c) CLAHE+LoG.

Table 4.10: Results using augmented method with CLAHE preprocessing

| Branch           | Threshold    | Score         |               |               |              |               |               |
|------------------|--------------|---------------|---------------|---------------|--------------|---------------|---------------|
|                  |              | Accuracy      | AUC           | F1            | Loss         | Precision     | Recall        |
| Global           | DenseNet121  | 97.29%        | 99.60%        | 96.41%        | 0.082        | 97.10%        | 95.77%        |
| Local (Cropping) | $\tau = 0.1$ | <b>98.42%</b> | <b>99.81%</b> | <b>97.77%</b> | <b>0.052</b> | <b>99.06%</b> | <b>96.78%</b> |
|                  | $\tau = 0.2$ | 98.28%        | 99.76%        | 97.96%        | 0.054        | 98.41%        | 97.26%        |
|                  | $\tau = 0.3$ | 97.92%        | 99.83%        | 97.57%        | 0.059        | 97.41%        | 97.61%        |
|                  | $\tau = 0.4$ | 97.79%        | 99.82%        | 97.30%        | 0.059        | 98.13%        | 96.53%        |
|                  | $\tau = 0.5$ | 97.68%        | 99.69%        | 97.25%        | 0.059        | 96.98%        | 97.64%        |
|                  | $\tau = 0.6$ | 97.82%        | 99.65%        | 97.42%        | 0.068        | 97.59%        | 97.01%        |
|                  | $\tau = 0.7$ | 97.71%        | 99.59%        | 97.19%        | 0.065        | 98.08%        | 96.44%        |
|                  | $\tau = 0.8$ | 97.53%        | 99.63%        | 96.97%        | 0.072        | 97.84%        | 96.20%        |
|                  | $\tau = 0.9$ | 96.33%        | 99.34%        | 95.53%        | 0.096        | 94.86%        | 96.15%        |

## Chapter 4: Attention guided Convolutional Neural Networks for Diabetic Retinopathy classification

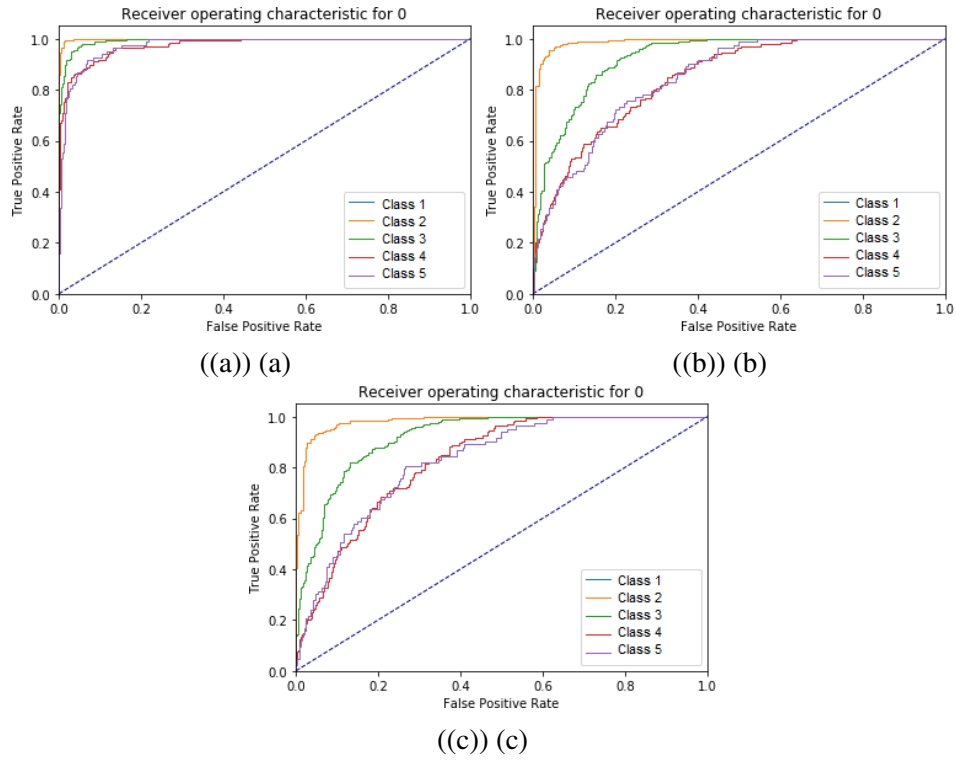


Figure 4.16: ROC curves of the best-performing threshold in scenario 3 for the local branch: a) CLAHE with  $\tau = 0.1$ , b) LoG with  $\tau = 0.1$ , and c) CLAHE+LoG with  $\tau = 0.2$ .

Table 4.11: Results using augmented method with LoG preprocessing

| Branch           | Threshold    | Score         |               |               |              |               |               |
|------------------|--------------|---------------|---------------|---------------|--------------|---------------|---------------|
|                  |              | Accuracy      | AUC           | F1            | Loss         | Precision     | Recall        |
| Global           | DenseNet121  | 95.12%        | 98.70%        | 94.05%        | 0.138        | 96.90%        | 91.41%        |
| Local (Cropping) | $\tau = 0.1$ | <b>95.53%</b> | <b>99.10%</b> | <b>94.57%</b> | <b>0.116</b> | <b>95.64%</b> | <b>93.58%</b> |
|                  | $\tau = 0.2$ | 94.86%        | 98.90%        | 93.78%        | 0.127        | 94.56%        | 93.10%        |
|                  | $\tau = 0.3$ | 94.76%        | 98.69%        | 93.57%        | 0.138        | 95.97%        | 91.35%        |
|                  | $\tau = 0.4$ | 94.34%        | 98.48%        | 93.11%        | 0.147        | 94.82%        | 91.53%        |
|                  | $\tau = 0.5$ | 94.28%        | 98.42%        | 93.01%        | 0.149        | 94.52%        | 91.23%        |
|                  | $\tau = 0.6$ | 94.50%        | 98.36%        | 92.88%        | 0.153        | 95.33%        | 91.17%        |
|                  | $\tau = 0.7$ | 94.10%        | 98.38%        | 92.85%        | 0.155        | 94.15%        | 91.05%        |
|                  | $\tau = 0.8$ | 94.09%        | 98.22%        | 92.73%        | 0.161        | 94.02%        | 91.01%        |
|                  | $\tau = 0.9$ | 94.22%        | 98.11%        | 92.70%        | 0.168        | 94.22%        | 90.94%        |

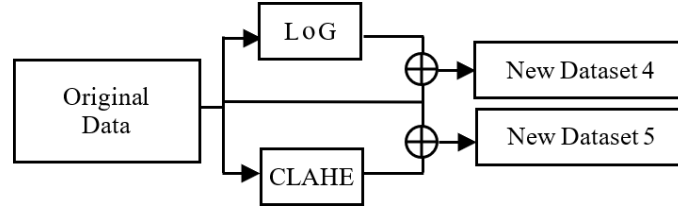


Figure 4.17: Diagram of the two augmented dataset generations.

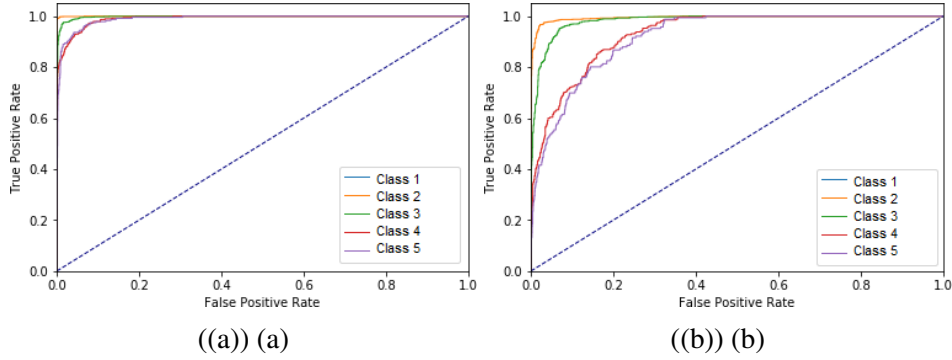


Figure 4.18: ROC curves obtained with augmented preprocessing scenario: a) CLAHE and LoG.

### 4.4.3 Discussion

Table 4.12 presents a comparative analysis of various preprocessing techniques and their influence on the validation accuracy of the diabetic retinopathy classification system. Several key observations emerge from the results across different scenarios.

Firstly, the baseline setup in Scenario 1, which uses basic preprocessing combined with local attention-based cropping, achieved a validation accuracy of 96.44%, establishing a solid reference point. The use of a binarization threshold ratio ( $\tau$ ) of 0.1 with localized cropping demonstrates the effectiveness of attention mechanisms in focusing on discriminative retinal regions.

In Scenario 2, the impact of individual preprocessing techniques was evaluated. CLAHE (Contrast Limited Adaptive Histogram Equalization) achieved a

slight improvement in accuracy to 96.51%, suggesting that it effectively enhances contrast and improves feature visibility. On the other hand, the use of the Laplacian of Gaussian (LoG), either alone or in combination with CLAHE, resulted in reduced validation accuracies of 93.27% and 93.49% respectively. This indicates that while LoG enhances edges, it may also introduce noise or obscure critical details needed for accurate classification. Furthermore, the performance improvement when LoG was used with localized cropping was minimal (0.47%), reinforcing that its benefit is limited even when applied to attention-focused regions.

Scenario 3 incorporated data balancing techniques in combination with CLAHE and LoG. Contrary to expectations, validation accuracies decreased in this scenario, particularly when LoG was involved. The lowest accuracy of 91.61% was observed for the combination of data balancing with CLAHE and LoG. These results suggest that while data balancing is important, its effectiveness depends heavily on the choice of preprocessing techniques. Inappropriate combinations may degrade the model's ability to generalize.

The best performance was observed in Scenario 4, where original images were fused with CLAHE-enhanced images and processed using local attention cropping. This configuration resulted in the highest validation accuracy of 98.42%, reflecting an improvement of 1.03% over the baseline. This outcome confirms that combining original and contrast-enhanced inputs, along with attention-guided cropping and a finely tuned binarization threshold ( $\tau = 0.1$ ), provides complementary information that significantly enhances model performance.

In summary, the results in Table 4.12 highlight several important findings: CLAHE is a consistently effective preprocessing technique, LoG tends to reduce performance, and data balancing does not inherently guarantee improvement un-

Table 4.12: Scenario results summarized

| Scenario | Preprocessing technique      | Validation accuracy (%) | Branch       | Improvement (%) | Binarization ratio ( $\tau$ )(%) |
|----------|------------------------------|-------------------------|--------------|-----------------|----------------------------------|
| 1        | Basic                        | 96.44                   | Local        | 0.3             | 0.1                              |
| 2        | CLAHE                        | 96.51                   | Global       | /               | /                                |
|          | LoG                          | 93.27                   | Local        | 0.47            | 0.2                              |
|          | CLAHE+LoG                    | 93.49                   | Global       | /               | /                                |
| 3        | Data balancing + CLAHE       | 96.51                   | Global       | /               | /                                |
|          | Data balancing + LoG         | 91.89                   | Global       | /               | /                                |
|          | Data balancing + CLAHE & LoG | 91.61                   | Global       | /               | /                                |
| 4        | <b>Original + CLAHE</b>      | <b>98.42</b>            | <b>Local</b> | <b>1.03</b>     | <b>0.1</b>                       |
|          | Original + LoG               | 95.53                   | Global       | /               | /                                |

less carefully integrated. The most successful approach involves the fusion of original and CLAHE-processed images guided by local attention, demonstrating the importance of both preprocessing quality and attention-driven region selection in diabetic retinopathy classification.

To demonstrate our method's advancements to deep learning-based Diabetic Retinopathy classification, we conducted a comparative analysis with state-of-the-art studies that used the same dataset (APTOS 2019). Table 4.13 reports the authors, used preprocessing operations, results and limitations of each method against our one. Except the method presented by Narayanan (Narayanan et al., 2020) where we achieved the same accuracy rate (98.4%) but our method achieved a superior AUC metric with 99.8%, our method outperforms the state-of-the-art studies in term of accuracy and AUC.

Table 4.13: Comparison with the state-of-the-art methods

| Author, Year              | Method                                              | Accuracy       | AUC          | Limitations                                                                                                                            |
|---------------------------|-----------------------------------------------------|----------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Majumder et al. [62]      | CNN                                                 | 88.5%          | /            | Lack of Generalization with other models                                                                                               |
| Shaban et al., [63]       | DCNN                                                | 88%            | 0.95         | <ul style="list-style-type: none"> <li>• 3 DR classes instead of 5-Feature</li> <li>• Study and selection are not available</li> </ul> |
| Gangwar and Ravi [64]     | Hybrid model (Inception, ResNet v2)                 | 82.18%         | /            | Computationally expensive                                                                                                              |
| Ramchandre et al. [65]    | EfficientNetb3<br>SEResNeXt32x4d                    | 91.4%<br>85.2% | /            | Lack of image enhancement                                                                                                              |
| Nahiduzzaman et al. [66]  | Parallel CNN/ELM                                    | 97.27%         | <b>0.988</b> | Lack of data augmentation                                                                                                              |
| Islam et al. [67]         | SCL/Xception                                        | 84.36%         | /            | Low batch size                                                                                                                         |
| Jian et al. [68]          | Triple-DRNets                                       | 92.08%         | /            | Limitation of feature fusion in the subnets                                                                                            |
| Narayanan et al. [61]     | Hybrid model (AlexNet, ResNet, VGG16, Inception v3) | <b>98.4%</b>   | 0.985        | Lack of data augmentation techniques                                                                                                   |
| Madarapu et al. [71]      | Custom deep CNN (Residual blocs + CSAM + NLB blocs) | 89.38%         | 0.97         | Complex architecture                                                                                                                   |
| Shakibania et al. [60]    | Dual Branch network (ResNet50 and Efficient-NetB0)  | 89.60%         | /            | Misclassifications for severe and proliferative DR                                                                                     |
| Selvakumar and Akila [86] | U-Net KNN model                                     | 80.78%         | /            | Computational complexity                                                                                                               |
| Moustari et al. [77]      | <b>DenseNet-121</b>                                 | <b>98.4%</b>   | <b>0.998</b> | <b>Computationally expensive in the training part</b>                                                                                  |

## 4.5 Conclusion

Deep learning has emerged as a powerful tool across various medical imaging tasks, offering improved accuracy, efficiency, and scalability in disease detection and diagnosis. Its application in retinal analysis particularly for identifying early signs of diabetic retinopathy highlights its potential in supporting large-scale screening efforts. While challenges around model interpretability and integration into clinical workflows remain, the technology continues to show promise as a cornerstone of next-generation medical decision support systems.

## 5 | Deep/handcrafted feature fusion for DR classification

### 5.1 Introduction

Fully automated grading of Diabetic Retinopathy (DR) from fundus images has advanced through deep learning and handcrafted techniques. While deep models capture complex patterns and lesion details, handcrafted features embed clinically validated insights. This chapter introduces a hybrid fusion framework combining both approaches. Deep features are extracted via CNNs, and handcrafted features detect specific DR signs like microaneurysms and exudates. The fused representation improves classification performance while maintaining clinical relevance and interpretability.

### 5.2 Deep feature extraction

The extraction of deep features is based on the very foundation of CNN, which has the ability to learn representations hierarchically in a self-taught fashion from scratch without any prior knowledge about domain expertise. Therefore, in our research study, the DenseNet-121 architecture was adopted for the extraction of deep features. DenseNet-121 is proposed by Huang et al.[\[83\]](#), in which dense connectivity means that each layer takes the feature maps from all previous layers as its input, thus enhancing feature reuse and encouraging the network to learn more diversified feature representations. By capturing low-level and high-level information, it has been regarded as one of the state-of-the-art architectures in medical image analysis.

DenseNet-121 was initialized with pre-trained weights from ImageNet, which allowed performing transfer learning. Transfer learning is effective, in particular, for domains where the target dataset is small and specialized. For instance, APTOS 2019 diabetic retinopathy. Further, the last classification layer of the pre-trained model was truncated so that the dense layers and convolutional features were retained, while adding a Global Average Pooling (GAP) layer for extracting a feature vector  $y = [y_1, y_2, \dots, y_{1024}]$ . This thus gives, for every image, a 1024-dimensional vector that encapsulates rich information on texture, patterns, and structures inherent in the fundus images.

Deep features from DenseNet-121 are advantageous since these abstract complex image patterns, such as blood vessels, lesions, and retinal irregularities, which depict the stages of diabetic retinopathy. These were saved as Numpy array files, where each image has a corresponding feature vector for later use with the handcrafted features.

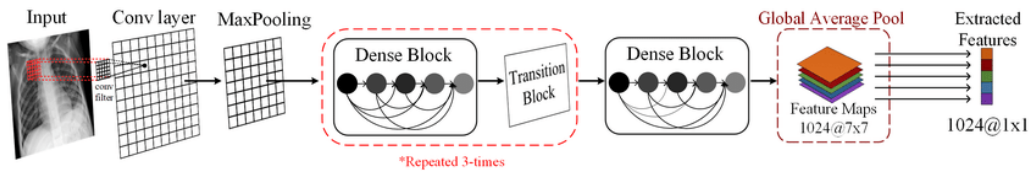


Figure 5.1: DenseNet for feature extraction.

## 5.3 Handcrafted texture feature extraction

### 5.3.1 Local Binary Pattern

LBP. [87] is a very powerful texture descriptor that quantifies the pattern of variation in intensity in the local neighborhoods of an image. Generally speaking, LBP analyzes the intensity difference between a point and its neighbors and then

represents such differences as binary patterns, The mathematical formulation of LBP is given as:

$$LBP(x_c, y_c) = \sum_{n=0}^{P-1} s(I_n - I_c) \bullet 2^n \quad (21)$$

Where  $I_c$  is the intensity of the central pixel at  $(x_c, y_c)$ ,  $I_n$  is the intensity of the neighboring pixel,  $P$  is the number of neighbors, and  $s(x)$  is the threshold function:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (22)$$

The simplicity of LBP is due to the fact that it transforms local texture information into a compact histogram representation while still being invariant under monotonic transformations of intensity. It compares the intensity of each surrounding pixel against the central one, creating a binary code according to whether the neighboring pixel's intensity is higher or lower. These binary codes are then converted into decimal values, which serve as the texture descriptors.

LBP is suited for applications where texture analysis is in focus and has captured small features on edges, corners, and uniform regions. In a concrete example, in medical images like retinal scans, the LBP could be applied in analyzing the texture on the surface of the retina by capturing subtle variations of intensity that might detail lesions or microaneurysms. The LBP operator is very computational-efficient, and it normally operates over small neighborhoods, such as a 3x3 or 5x5 pixel grid. The obtained histograms summarize the distribution of the local texture, and these histograms are usually normalized in order to achieve scale invariance among different images.

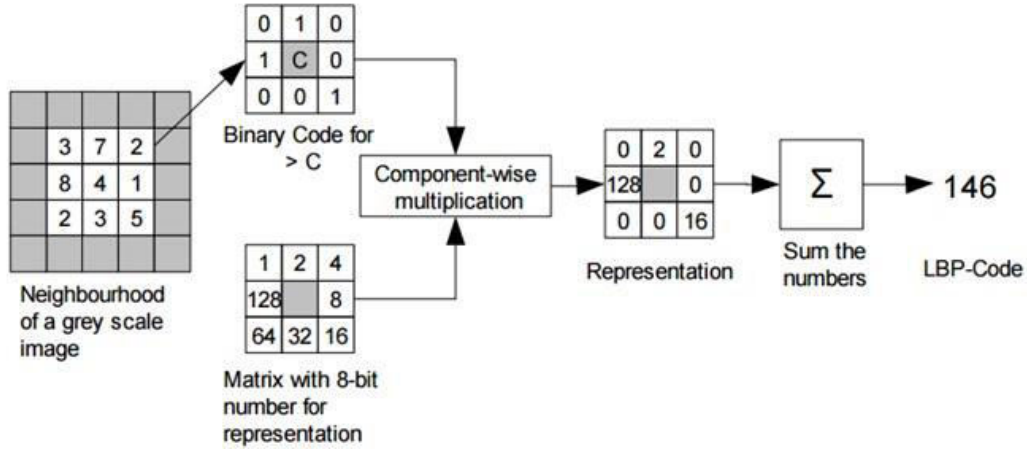


Figure 5.2: The Local Binary Pattern method.

### 5.3.2 Gray-Level Co-occurrence Matrix

The Gray-Level Co-occurrence Matrix (GLCM) [88] refers to a second-order statistical approach used in the description of textures through a study of the spatial relationship that exists between pairs of pixels in an image. Texture, in this case, refers to the regular pattern or structural arrangement of pixel intensities. Whereas first-order methods consider pixel intensities in isolation, GLCM considers the frequency with which pairs of pixels of particular gray levels (intensities) occur at a given spatial distance and orientation. GLCM captures the distribution of co-occurring pixel intensities, making it sensitive to both the local and global structure of an image.

$$P(i, j|d, \theta) = \sum_{p, q} \begin{cases} 1 & \text{if } I(p, q) = i \text{ and } I(p + d_x, q + d_y) = j \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

Where  $P(i, j|d, \theta)$  is the probability of pixel intensity  $i$  occurring with pixel intensity  $j$ , separated by a distance  $d$  in the direction  $\theta$ .

From the GLCM, the following features describing texture can be obtained:

- Contrast: It is a measure of the intensity variation of a pixel and its neighbor throughout the image. A high contrast means high local variation.

$$Contrast = \sum_{i,j} P(i,j) \bullet (i - j)^2 \quad (24)$$

- Correlation: This would describe the correlation between a pixel and its neighbors, which can lead to assessment in smoothness or regularity regarding texture.

$$Correlation = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) P(i,j)}{\sigma_i \sigma_j} \quad (25)$$

Where  $\mu$  and  $\sigma$  are the means and standard deviations of the pixel intensities.

- Energy: It is also known as Angular Second Moment. It is the sum of squares of elements in GLCM. It is a measure of uniformity or smoothness.

$$Energy = \sum_{i,j} P(i,j)^2 \quad (26)$$

- Homogeneity: This represents the closeness of the distribution of elements in GLCM to the diagonal. The higher the homogeneity, the more uniform the texture.

$$Homogeneity = \sum_{i,j} \frac{P(i,j)}{1 + |i - j|} \quad (27)$$

GLCM features have proven very effective for the characterization of large-scale texture patterns of an image, such as the smooth regions of the retina or the irregular textures induced by lesions. The approach is flexible, and different pixel distances and orientations, for example,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ , can be used to capture directional texture patterns that may reveal important diagnostic features.

### 5.3.3 Histogram of Oriented Gradients

HOG [89] assumes that the appearance and shape of an object can be well characterized by the distribution of local intensity gradients or edge directions regardless of the exact image details. Divide the image into small cells, normally 8x8 pixels, then have HOG capture the local patterns of gradients. After that, the histograms are concatenated and normalized over overlapping regions called "blocks" to ensure features can sustain variations related to lighting and contrast conditions.

$$G_x = I(x + 1, y) - I(x - 1, y) \quad (28)$$

$$G_y = I(x, y + 1) - I(x, y - 1) \quad (29)$$

$$|\nabla I(x, y)| = \sqrt{G_x^2 + G_y^2} \quad (30)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (31)$$

HOG seems particularly effective in capturing structural information such as edges and contours. It thus finds useful applications for the analysis of images that have well-defined geometric patterns. In medical images, for example, in retinal scans, it will help in the identification of edges created by blood vessels or lesions, which would be important from a diagnostic point of view. This feature descriptor is useful in cases where fine-scale textures and patterns have to be captured, as it gives a rich description of the local geometry of an image. The gradients are computed using a central difference method normally, and the resulting histograms are normalized by the L2-Hys normalization technique, which makes the feature vectors invariant against the lighting conditions and small shifts in the image.

## 5.4 Proposed method

This section defines the Deep/Handcrafted Feature Fusion system for DR classification, based on DenseNet-121 as a deep feature extraction backbone. It should combine deep learning and handcrafted texture attribute features in an attempt to improve the accuracy of classification.

The idea consists of two major steps: deep feature extraction and handcrafted feature extraction. First, CLAHE is employed to enhance the contrast in the input fundus images, and unsharp masking is utilized to provide a better definition of fine details. Subsequently, the resultant preprocessed images were passed through DenseNet-121, from which the deep features were extracted after being passed through convolutional layers and a global average pooling layer, yielding a 1024-dimensional feature vector.

At the same time, handcrafted texture features, such as LBP, GLCM, and HOG, are extracted from the same preprocessed images. These features represent complex textural information, thus enriching the high-level features obtained using the CNN.

The fusion phase combines the deep features from DenseNet-121 with the handcrafted features into one feature vector. This combined representation now leverages higher-order and low-level visual cues that improve the performance of classification.

These features are finally fed into the different classifiers to classify the stages of DR. This fusion-centric methodology enables the system to leverage both deep learning and manually designed features, thereby improving its capacity to precisely categorize the stages of diabetic retinopathy.

### **5.4.1 Preprocessing**

The efficiency of preprocessing is necessary to ensure the input of high-quality data for both deep learning algorithms and handcrafted feature extraction methods. During this stage, we apply two basic preprocessing methods:

#### **5.4.1.1 Contrast Limited Adaptive Histogram Equalization**

Poor or uneven illumination, a lot of shadows, and low contrast are common in fundus images, which may mask some critical features of the retina. Therefore, local contrast stretching is enhanced using CLAHE with less amplification of noise. It segments the image into small tiles and then applies histogram equalization within each tile to further enhance the contrast for the regions of interest while controlling amplification of noise in regions that are uniform. Thus, this enhances the visibility of structures such as blood vessels and lesions, as shown in Figure 4.4.

#### **5.4.1.2 Unsharp Masking**

An unsharp mask technique is used to emphasize the fine details in retinal images, which enhances sharpness in images by highlighting the edges. This may be achieved by subtracting the blurred form of an image from its original image form, making notable features such as microaneurysms, exudates, and vascular structures more visible. This initial processing stage is particularly useful for locating small abnormalities in the retina and thus gives a boost to deep learning and handcrafted feature extraction techniques.

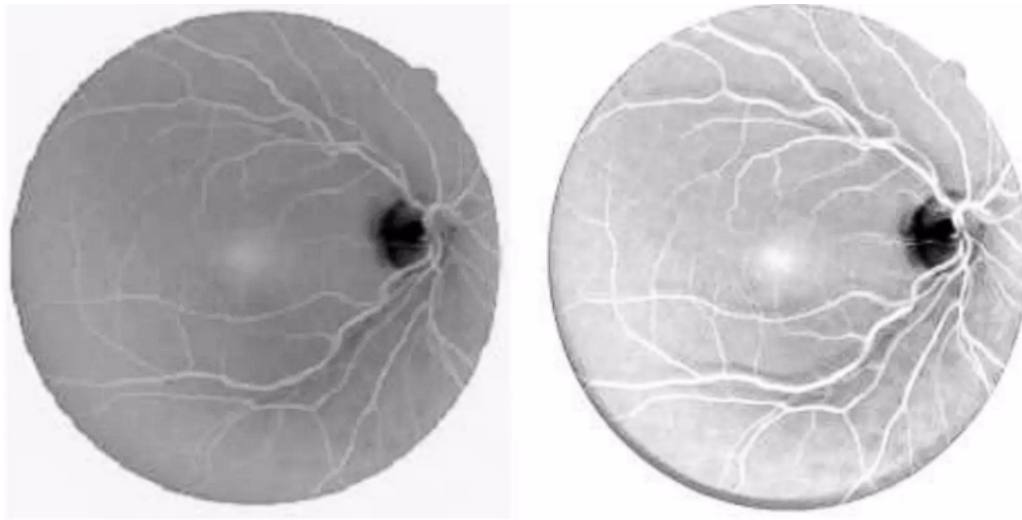


Figure 5.3: Unsharp masking method

## 5.4.2 Deep Feature Extraction

We employ the usage of DenseNet-121, which is the state-of-the-art CNN pre-trained on ImageNet, fine-tuned for our DR classification task. Given its architecture, DenseNet-121 is able to share features across layers with evident complexity in capturing patterns within a medical image.

### 5.4.2.1 DenseNet-121 Architecture

DenseNet-121 is a multilevel dense block structure, in which every layer receives input from all the previous layers. This structure enhances the capturing of more complex representations by the network. First, pre-processing of input images is performed and then further resized to 224 x 224 pixels to feed the network input. DenseNet-121 extracts high-level semantic features through a convolution and pooling layers pattern that identifies the complex patterns in lesions, retinal hemorrhages, and exudates, shown in Figure 4.1.

#### **5.4.2.2 Global Feature Extraction**

Instead of taking the final classification layers of DenseNet-121, we will treat the model as a feature extractor. First, the features pass through the network consisting of dense convolutional blocks and transition layers. These are then passed through a Global Average Pooling layer that compresses the feature maps into a 1024-dimensional feature vector. It captures the global high-level representation of the image, for instance, the presence of lesions, the overall structure of the retina, and any other DR-related abnormality.

### **5.4.3 Handcrafted Feature Extraction**

Handcrafted feature extraction is done parallel to deep feature extraction. Unlike deep features that catch high-level semantic patterns, handcrafted features capture the local textures and fine structural details that are very useful for understanding the DR.

#### **5.4.3.1 Local Binary Patterns (LBP)**

LBP describes the local texture patterns by examining the neighborhood of each pixel. In other words, for every given pixel, a comparison between itself and the surrounding pixels is drawn and the result encoded into a binary pattern. This histogram of the binary patterns comprises the LBP feature vector. LBP works especially for small-scale textures like retinal vessel structures, which can be an early indicator of DR.

#### **5.4.3.2 Gray-Level Co-occurrence Matrix (GLCM)**

The GLCM calculates a probability of every pair of pixels having a given intensity value concerning each other, based on their spatial relation, and serves to

capture the second order of text information. Accordingly, significant statistical features extracted include contrast, correlation, energy, and homogeneity by determining the frequency of two particular pairs of pixels at a specified distance and orientation. These features thus help in the identification of changes in DR through the detection of texture changes resulting from bleeding or swelling of tissues.

### **5.4.3.3 Histogram of Oriented Gradients (HOG)**

It captures the distribution of edge directions or intensity gradients inside an image. This approach divides the whole image into small regions, calculates the gradient orientation histograms in each region, and then concatenates them to obtain the HOG feature vector. The most helpful application of HOG is detecting larger structural elements such as the boundaries of the retina, optic disc, and lesions.

These three handcrafted features-labeled LBP, GLCM, and HOG-complement the high-level deep features by focusing on fine details of the texture, which are indeed crucial for DR classification.

### **5.4.4 Feature Fusion**

The goal of feature fusion is to combine the merits of deep learning and handcrafted features for the advantage of texture features; it can leverage both high-level representations learned by DenseNet-121 and low-level textures captured through handcrafted means.

#### **5.4.4.1 Deep and handcrafted Feature Fusion**

After extracting the 1024-dimensional deep feature vector from DenseNet-121, it is then concatenated with handcrafted feature vectors, namely, LBP, GLCM, and HOG. The resulting fused feature vector now carries information from both categories. The dimensionality of this fused vector is the sum of the dimensions of the deep and handcrafted features; thus, it forms a highly representative representation of the retinal image.

This stage of fusion ensures that both the global features—for example, overarching structures of retinal lesions and exudates—and the local textures, such as vessel patterns, are considered in classification.

#### **5.4.4.2 Normalization**

Z-score normalization is carried out, which normalizes the attribute values derived from different sources. This ensures that deep features are not, by their higher magnitudes, overshadowing the handcrafted features anywhere in the process of classification.

### **5.4.5 Classification**

After gaining the combined feature vector, it is further fed into different classifiers:

#### **5.4.5.1 Support Vector Machine (SVM)**

introduced in 1995 by Cortes [90], SVM finds an optimal hyperplane that maximizes the margin between different classes. It is effective in high-dimensional spaces and can handle both linear and non-linear classification using kernel functions. Key parameters

- C: Regularization parameter controlling trade-off between maximizing margin and minimizing classification error.
- kernel: Defines the type of decision boundary (e.g., linear, rbf, poly, sigmoid).
- gamma: Controls the influence of a single training example in RBF and polynomial kernels.

#### 5.4.5.2 K-Nearest Neighbors (KNN)

KNN classifies a sample based on the majority vote of its k-nearest neighbors. It does not require training but can be computationally expensive for large datasets [91]. Key parameters

- k: Number of nearest neighbors to consider.
- weights: Determines how neighbors contribute (e.g., uniform or distance).

#### 5.4.5.3 Multilayer Perceptron (MLP)

MLP is a feedforward neural network with one or more hidden layers. It uses backpropagation for training and can learn complex patterns. Key parameters

- hidden layer sizes: Defines the number of neurons in each hidden layer.
- Maximum number of iterations: MLP iterates until convergence or this number of iterations.

#### 5.4.5.4 Random Forest

An ensemble learning method that builds multiple decision trees and combines their outputs to improve accuracy and reduce overfitting [92]. Key parameters

- n of estimators: Number of trees in the forest.

#### **5.4.5.5 1D Convolutional Neural Network (1D CNN)**

1D CNN applies convolutional filters to sequential data, capturing spatial dependencies efficiently. It is widely used in time-series classification and signal processing. its key parameters are the same as the regular CNN network.

#### **5.4.6 Model Evaluation**

This classifier performance is done using the common metrics of accuracy, precision, recall, and F1-score and AUC of ROC curve. This gives insight into strengths and weaknesses of the fused feature-based classification approach. This system will leverage deep features from DenseNet-121 and combine them with handcrafted texture features like LBP, GLCM, and HOG to capture a wide range of visual cues present within the retinal images. This offers further extended capability for enhanced discrimination between subtle and complex stages of diabetic retinopathy, hence improving the classifier's performance.

### **5.5 Results**

For the evaluation of our methodology, we conducted several experiments using binary and multiclass classification tasks on the APTOS 2019 diabetic retinopathy dataset. Initially, we evaluated the performance of different combinations of features, including CNN deep features, Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gray Level Co-occurrence Matrix (GLCM) features. These combinations were tried in various settings: single features along with CNN, two features together along with CNN, and lastly, the combination of all three handcrafted features with CNN.

Afterward, we used a range of classifiers to test their performance in this partic-

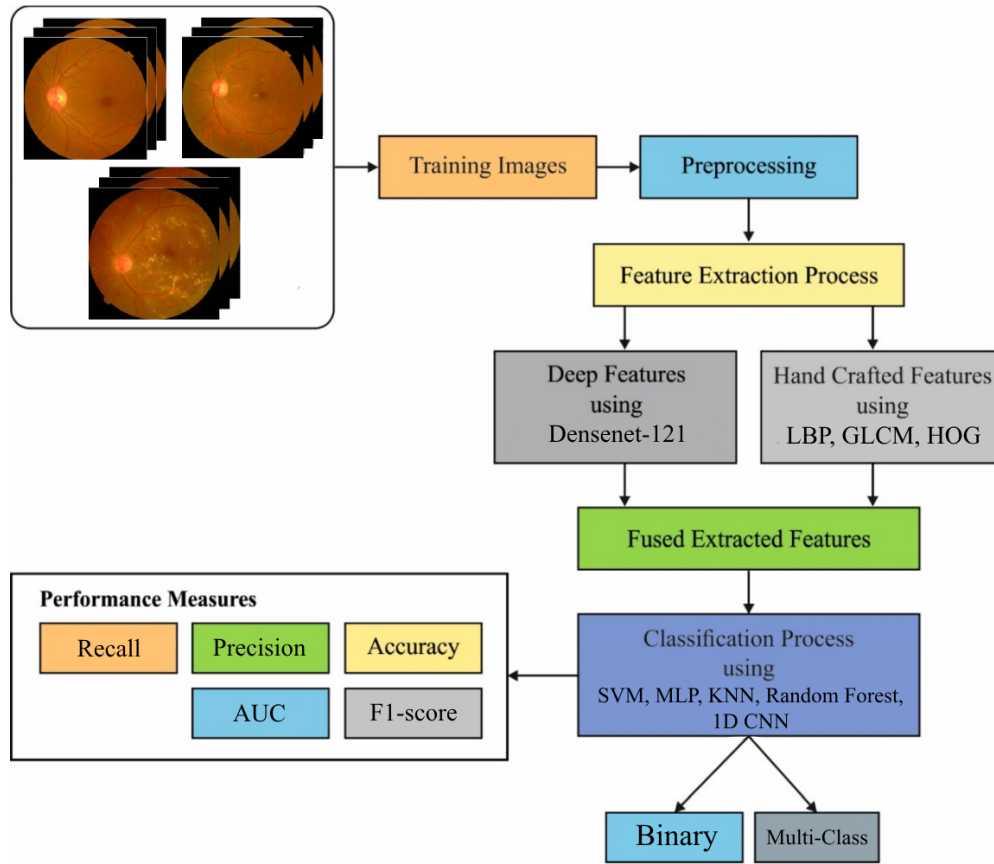


Figure 5.4: Deep/Handcrafted feature extraction, fusion and classification system.

ular setting. Among the classifiers tried were Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and a 1D Convolutional Neural Network (1D CNN). the parameters of the SVM, KNN, MLP and Random Forest are :

- SVM: kernel='rbf', C=100, gamma='scale'
- KNN: n of neighbors=5, weights='distance'
- MLP: hidden layer sizes=(128, 64) , max iterations=300
- Random Forest: n of estimators = 100

Each model's performance was evaluated using a comprehensive set of metrics that included:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC

### **5.5.1 Binary classification**

The tables representing the performance of the various classifiers with the various combinations are presented below, along with it, to visualize performance, we generated ROC curves for classification scenario, illustrating the trade-off between the true positive rate and the false positive rate across varying threshold values. This thereby allowed us to systematically compare the influence of different feature sets combined with diverse classifiers in the tasks of binary and multiclass classification to finally gain insight into the best configurations to perform effective diabetic retinopathy detection.

Table 5.1: performance of the SVM classifier with different feature combination (binary classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 98.27        | 98.37         | 98.27      | 98.27        | 97.99   |
| CNN+LBP             | 98.00        | 98.14         | 98.00      | 98.00        | 98.20   |
| CNN+HOG             | 98.27        | 98.37         | 98.27      | 98.27        | 97.67   |
| CNN+GLCM            | 98.27        | 98.33         | 98.27      | 98.27        | 98.01   |
| CNN+LBP+HOG         | 98.00        | 98.12         | 98.00      | 98.00        | 98.09   |
| CNN+LBP+GLCM        | 98.00        | 98.14         | 98.00      | 98.00        | 98.23   |
| CNN+HOG+GLCM        | 98.27        | 98.37         | 98.27      | 98.27        | 97.68   |
| CNN+ALL             | 98.00        | 98.12         | 98.00      | 98.00        | 98.10   |
| LBP+HOG             | 97.59        | 97.64         | 97.59      | 97.59        | 97.65   |
| LBP+GLCM            | 94.59        | 94.68         | 94.59      | 94.57        | 95.26   |
| HOG+GLCM            | 97.59        | 97.65         | 97.59      | 97.59        | 97.21   |
| LBP+HOG+GLCM        | 98.27        | 98.30         | 98.27      | 98.27        | 97.62   |

Table 5.2: performance of the MLP classifier with different feature combination (binary classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 98.27        | 98.31         | 98.27      | 98.27        | 98.20   |
| CNN+LBP             | 97.59        | 97.65         | 97.59      | 97.59        | 97.89   |
| CNN+HOG             | 98.54        | 98.58         | 98.54      | 98.55        | 98.14   |
| CNN+GLCM            | 97.86        | 97.91         | 97.86      | 97.86        | 98.20   |
| CNN+LBP+HOG         | 98.13        | 98.22         | 98.13      | 98.14        | 98.15   |
| CNN+LBP+GLCM        | 97.72        | 97.82         | 97.72      | 97.73        | 97.77   |
| CNN+HOG+GLCM        | 98.13        | 98.18         | 98.13      | 98.14        | 98.01   |
| CNN+ALL             | 98.27        | 98.34         | 98.27      | 98.27        | 98.14   |
| LBP+HOG             | 97.45        | 97.49         | 97.45      | 97.45        | 97.97   |
| LBP+GLCM            | 86.54        | 86.54         | 86.54      | 86.52        | 90.28   |
| HOG+GLCM            | 96.63        | 96.67         | 96.63      | 96.64        | 97.59   |
| LBP+HOG+GLCM        | 97.86        | 97.89         | 97.86      | 97.86        | 97.83   |

Table 5.3: performance of the KNN classifier with different feature combination (binary classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 96.09        | 96.26         | 96.09      | 96.09        | 95.46   |
| CNN+LBP             | 87.08        | 90.51         | 87.08      | 86.81        | 90.51   |
| CNN+HOG             | 94.86        | 95.32         | 94.86      | 94.86        | 95.54   |
| CNN+GLCM            | 96.22        | 96.38         | 96.22      | 96.23        | 95.49   |
| CNN+LBP+HOG         | 95.13        | 95.54         | 95.13      | 95.13        | 96.21   |
| CNN+LBP+GLCM        | 87.36        | 90.54         | 87.36      | 87.11        | 90.92   |
| CNN+HOG+GLCM        | 94.86        | 95.32         | 94.86      | 94.86        | 95.54   |
| CNN+ALL             | 95.13        | 95.54         | 95.13      | 95.13        | 96.21   |
| LBP+HOG             | 94.31        | 94.47         | 94.31      | 94.32        | 96.09   |
| LBP+GLCM            | 51.59        | 26.93         | 51.89      | 35.01        | 54.31   |
| HOG+GLCM            | 94.45        | 94.88         | 94.45      | 94.45        | 95.25   |
| LBP+HOG+GLCM        | 94.31        | 94.47         | 94.31      | 94.32        | 96.09   |

Table 5.4: performance of the Random Forest classifier with different feature combination (binary classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 96.91        | 97.02         | 96.91      | 96.91        | 97.49   |
| CNN+LBP             | 96.50        | 96.68         | 96.50      | 96.50        | 97.32   |
| CNN+HOG             | 97.32        | 97.40         | 97.32      | 97.32        | 97.36   |
| CNN+GLCM            | 96.50        | 96.65         | 96.50      | 96.50        | 97.29   |
| CNN+LBP+HOG         | 96.91        | 96.98         | 96.91      | 96.91        | 97.50   |
| CNN+LBP+GLCM        | 96.77        | 96.93         | 96.77      | 96.77        | 97.39   |
| CNN+HOG+GLCM        | 96.63        | 96.73         | 96.63      | 96.64        | 97.21   |
| CNN+ALL             | 97.72        | 97.80         | 97.72      | 97.73        | 97.25   |
| LBP+HOG             | 96.50        | 96.51         | 96.50      | 96.49        | 96.85   |
| LBP+GLCM            | 95.13        | 95.25         | 95.13      | 95.11        | 96.75   |
| HOG+GLCM            | 96.50        | 96.50         | 96.50      | 96.50        | 96.79   |
| LBP+HOG+GLCM        | 96.77        | 96.78         | 96.77      | 96.77        | 97.09   |

Table 5.5: performance of the 1D CNN classifier with different feature combination (binary classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 97.32        | 97.65         | 97.32      | 97.32        | 97.71   |
| CNN+LBP             | 97.45        | 97.57         | 97.45      | 97.45        | 97.99   |
| CNN+HOG             | 98.27        | 98.37         | 98.27      | 98.27        | 98.22   |
| CNN+GLCM            | 97.00        | 97.04         | 97.00      | 97.00        | 97.77   |
| CNN+LBP+HOG         | 98.54        | 98.62         | 98.54      | 98.55        | 97.72   |
| CNN+LBP+GLCM        | 96.91        | 97.22         | 96.91      | 96.91        | 97.88   |
| CNN+HOG+GLCM        | 98.68        | 98.73         | 98.68      | 98.68        | 97.97   |
| CNN+ALL             | 98.54        | 98.62         | 98.54      | 98.55        | 98.02   |
| LBP+HOG             | 96.91        | 96.91         | 96.91      | 96.90        | 97.52   |
| LBP+GLCM            | 92.95        | 93.49         | 92.95      | 92.94        | 95.75   |
| HOG+GLCM            | 97.45        | 97.55         | 97.45      | 97.45        | 97.30   |
| LBP+HOG+GLCM        | 97.86        | 97.87         | 97.86      | 97.86        | 97.70   |

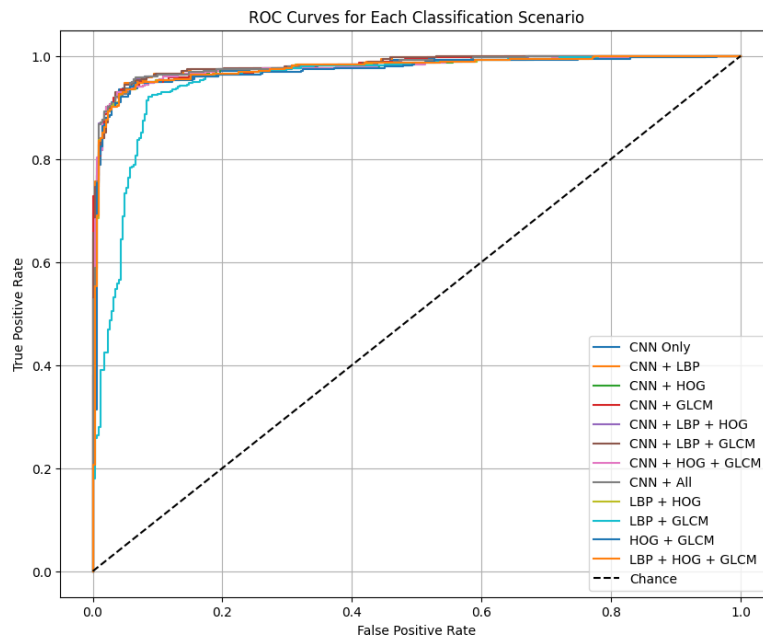


Figure 5.5: ROC curve of SVM classifier (binary classification).

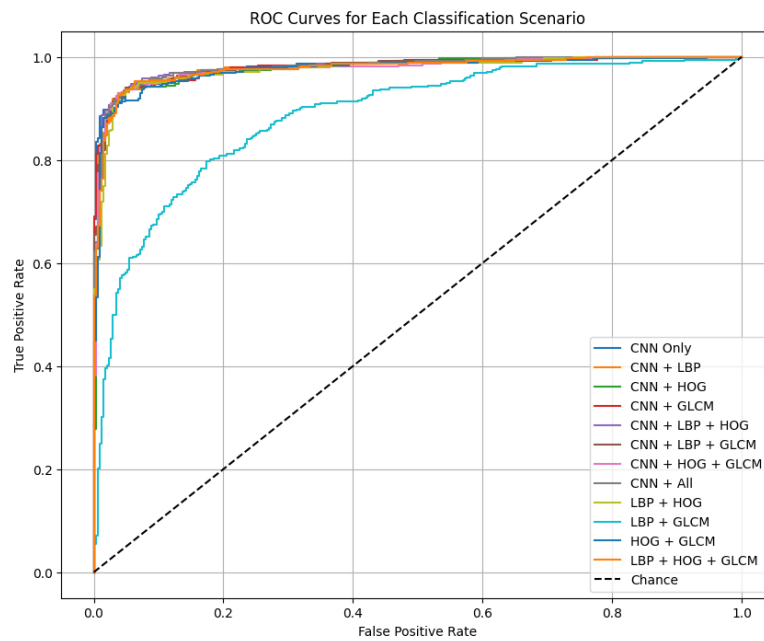


Figure 5.6: ROC curve of MLP classifier (binary classification).

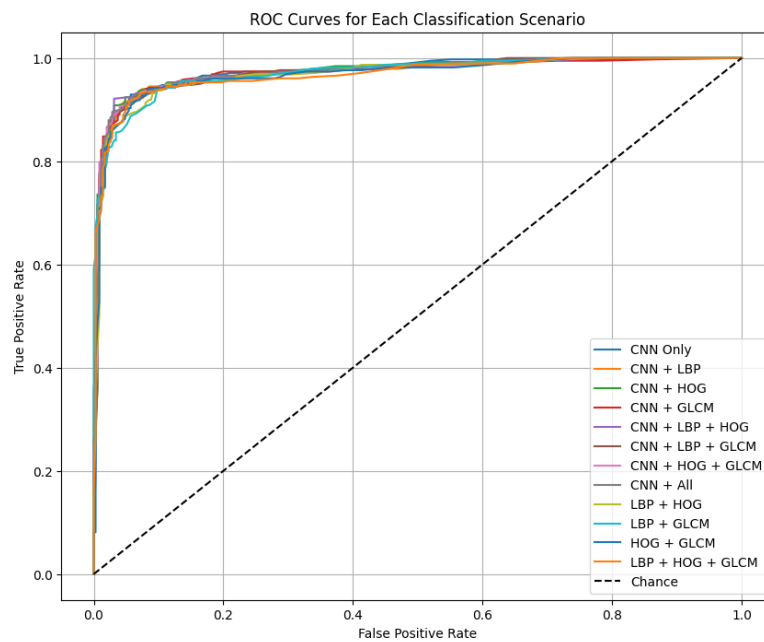


Figure 5.7: ROC curve of KNN classifier (binary classification).

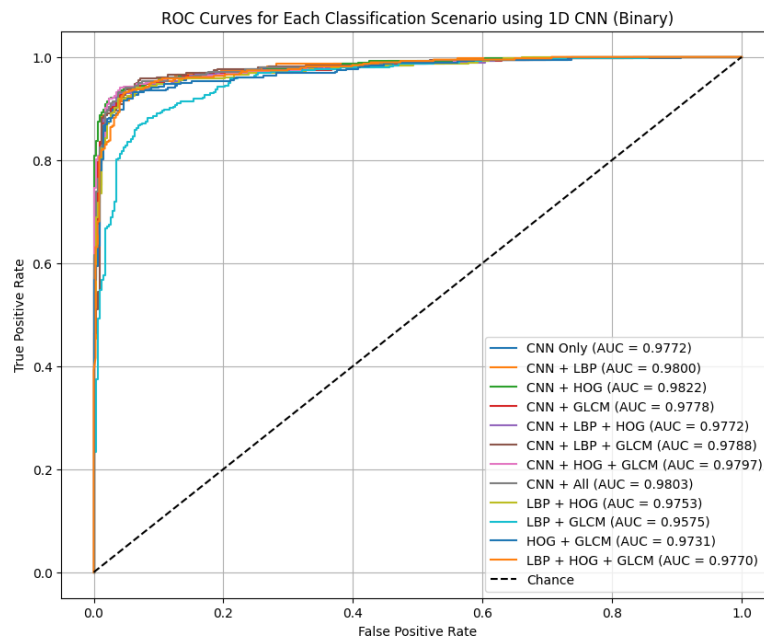


Figure 5.8: ROC curve of Random Forest classifier (binary classification).

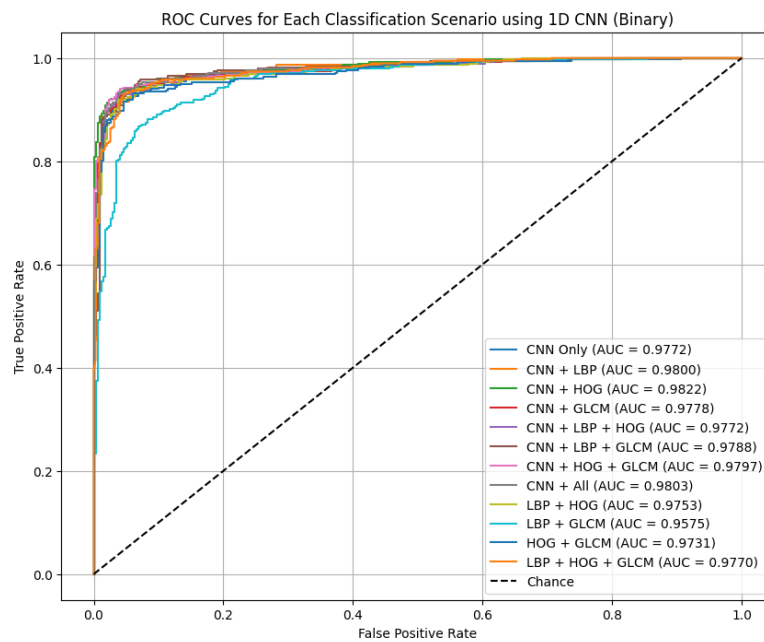


Figure 5.9: ROC curve of 1D CNN classifier (binary classification).

#### **5.5.1.1 Discussion**

The results of the binary classification using different models and feature combinations, in terms of accuracy, precision, recall, and F1 score, are very promising. All the models showed extremely strong performance characteristics, meaning that the combination of CNNs with various feature extraction techniques significantly improves the classification performance.

#### **5.5.1.2 Convolutional Neural Networks (CNNs):**

The CNN alone gave an accuracy of 98.27% and high precision and recall values of 98.37% and 98.27%, respectively. This has a very balanced F1 score at 98.27%. The CNN is, therefore, well adapted for binary classification tasks and can differentiate the two classes very well.

#### **5.5.1.3 Feature Integration**

Results have shown that, even combined with handcrafted features like LBP, HOG, and GLCM, the CNN still maintains strong performance. More specifically, in the case of CNN combined with HOG, it achieves 98.27% for both accuracy and F1 score, meaning the features will further improve the model in capturing texture and spatial patterns.

The results suggest that combining multiple feature sets can yield marginal improvements, as seen with the CNN+HOG+GLCM model, which maintained high performance metrics. However, adding LBP to this combination (CNN + LBP + HOG + GLCM) did not significantly boost performance.

#### **5.5.1.4 Traditional Machine Learning Models results:**

More traditional classifiers, such as Random Forest, SVM, MLP, and KNN, achieved good results but slightly worse than the CNN-based models. The Random Forest model achieved 97.91% accuracy with an F1-score of 97.91%, showing itself to be a very good baseline model for binary classification tasks.

The KNN model was a bit unstable, especially when using a combination of CNN and LBP, for which its accuracy fell to 87.09%, which would imply that with some combinations of features, it performs poorly.

The consistently high AUC values across all models (mostly above 97%) indicate that the models are well-calibrated on the binary classification task. This thus shows that, apart from being accurate, the models are reliable in distinguishing between the positive and negative classes. Conclusion The results obtained give evidence that convolutional neural networks, especially combined with additional feature extraction processes, can show a substantial improvement over the standard binary classification task. While traditional machine learning models remain useful, the increased effectiveness of the CNN-based approach makes them better suited to more complex tasks, particularly those involving image data. The future research shall focus on the optimization of these models in a further direction and explore other feature sets in order to continue improving the classification results.

### **5.5.2 Multi-class classification**

The tables representing the performance of the various classifiers with the various combinations are presented below, along with it, a generated ROC curve for best multi class classification scenario (1D CNN).

Table 5.6: performance of the SVM classifier with different feature combination (multiclass classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 89.49        | 87.50         | 89.49      | 88.22        | 98.72   |
| CNN + LBP           | 90.03        | 87.28         | 90.03      | 85.95        | 98.86   |
| CNN + HOG           | 90.58        | 88.60         | 90.58      | 87.83        | 98.64   |
| CNN + GLCM          | 89.62        | 87.56         | 89.62      | 88.30        | 98.80   |
| CNN + LBP + HOG     | 90.72        | 88.11         | 90.72      | 87.40        | 99.19   |
| CNN + LBP + GLCM    | 89.90        | 87.33         | 89.90      | 85.88        | 98.90   |
| CNN + HOG + GLCM    | 90.58        | 88.60         | 90.58      | 87.83        | 98.65   |
| CNN + ALL           | 90.72        | 88.11         | 90.72      | 87.40        | 99.20   |
| LBP + HOG           | 90.58        | 89.06         | 90.58      | 86.84        | 97.89   |
| LBP + GLCM          | 84.44        | 83.57         | 84.44      | 78.76        | 92.11   |
| HOG + GLCM          | 90.03        | 87.41         | 90.03      | 87.09        | 97.09   |
| LBP + HOG + GLCM    | 90.58        | 89.06         | 90.58      | 86.84        | 97.88   |

Table 5.7: performance of the KNN classifier with different feature combination (multiclass classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 86.21        | 82.52         | 86.12      | 83.82        | 91.48   |
| CNN + LBP           | 83.35        | 79.95         | 83.35      | 78.85        | 86.39   |
| CNN + HOG           | 87.71        | 84.56         | 87.71      | 85.59        | 93.77   |
| CNN + GLCM          | 85.94        | 82.28         | 85.94      | 83.65        | 92.02   |
| CNN + LBP + HOG     | 89.76        | 86.93         | 89.76      | 87.47        | 92.24   |
| CNN + LBP + GLCM    | 82.94        | 79.26         | 82.94      | 78.52        | 86.80   |
| CNN + HOG + GLCM    | 87.71        | 84.56         | 87.71      | 85.59        | 93.45   |
| CNN + ALL           | 89.76        | 86.98         | 89.76      | 87.46        | 92.22   |
| LBP + HOG           | 85.26        | 82.23         | 85.26      | 83.25        | 91.17   |
| LBP + GLCM          | 66.02        | 78.54         | 66.02      | 52.77        | 67.87   |
| HOG + GLCM          | 85.80        | 81.92         | 85.80      | 83.17        | 92.55   |
| LBP + HOG + GLCM    | 85.40        | 82.33         | 85.40      | 83.36        | 91.04   |

Table 5.8: performance of the MLP classifier with different feature combination (multiclass classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 87.85        | 86.02         | 87.85      | 86.67        | 97.62   |
| CNN + LBP           | 89.22        | 86.72         | 89.22      | 87.49        | 97.11   |
| CNN + HOG           | 88.26        | 85.72         | 88.26      | 86.45        | 97.48   |
| CNN + GLCM          | 88.40        | 86.68         | 88.40      | 87.18        | 97.99   |
| CNN + LBP + HOG     | 90.44        | 88.00         | 90.44      | 88.60        | 98.46   |
| CNN + LBP + GLCM    | 87.99        | 85.23         | 87.99      | 86.11        | 96.80   |
| CNN + HOG + GLCM    | 90.58        | 87.87         | 90.58      | 88.26        | 97.32   |
| CNN + ALL           | 89.35        | 86.72         | 89.35      | 87.53        | 98.00   |
| LBP + HOG           | 89.08        | 86.58         | 89.08      | 86.99        | 95.96   |
| LBP + GLCM          | 74.75        | 70.85         | 74.75      | 72.36        | 89.50   |
| HOG + GLCM          | 87.71        | 84.96         | 87.71      | 85.97        | 95.78   |
| LBP + HOG + GLCM    | 88.94        | 86.02         | 88.94      | 86.88        | 97.19   |

Table 5.9: performance of the Random Forest classifier with different feature combination (multiclass classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 88.26        | 85.44         | 88.26      | 83.50        | 96.48   |
| CNN + LBP           | 87.85        | 84.91         | 87.85      | 82.54        | 96.89   |
| CNN + HOG           | 89.62        | 87.44         | 89.62      | 85.24        | 96.85   |
| CNN + GLCM          | 88.81        | 85.78         | 88.81      | 83.86        | 96.99   |
| CNN + LBP + HOG     | 89.22        | 86.87         | 89.22      | 84.31        | 97.42   |
| CNN + LBP + GLCM    | 87.99        | 84.72         | 87.99      | 82.58        | 96.45   |
| CNN + HOG + GLCM    | 89.76        | 87.81         | 89.76      | 84.98        | 97.60   |
| CNN + ALL           | 88.67        | 86.65         | 88.67      | 84.08        | 96.59   |
| LBP + HOG           | 87.58        | 86.62         | 87.58      | 82.75        | 95.00   |
| LBP + GLCM          | 84.99        | 82.78         | 84.99      | 78.32        | 92.16   |
| HOG + GLCM          | 88.94        | 87.57         | 88.94      | 84.00        | 95.50   |
| LBP + HOG + GLCM    | 88.12        | 86.12         | 88.12      | 83.11        | 95.80   |

Table 5.10: performance of the 1D CNN classifier with different feature combination (multiclass classification)

| Feature combination | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | AUC (%) |
|---------------------|--------------|---------------|------------|--------------|---------|
| CNN                 | 90.17        | 87.89         | 90.17      | 87.09        | 97.18   |
| CNN + LBP           | 89.08        | 85.57         | 89.08      | 86.74        | 96.70   |
| CNN + HOG           | 90.03        | 88.49         | 90.03      | 88.28        | 97.61   |
| CNN + GLCM          | 90.44        | 86.77         | 90.44      | 86.48        | 96.97   |
| CNN + LBP + HOG     | 91.81        | 89.25         | 91.81      | 89.99        | 98.47   |
| CNN + LBP + GLCM    | 88.53        | 85.67         | 88.53      | 85.10        | 96.62   |
| CNN + HOG + GLCM    | 90.31        | 87.53         | 90.31      | 87.79        | 97.17   |
| CNN + ALL           | 91.13        | 88.56         | 91.13      | 88.21        | 97.29   |
| LBP + HOG           | 89.49        | 86.48         | 89.49      | 86.47        | 96.90   |
| LBP + GLCM          | 82.94        | 79.13         | 82.94      | 80.42        | 89.70   |
| HOG + GLCM          | 89.90        | 86.30         | 89.90      | 86.59        | 96.46   |
| LBP + HOG + GLCM    | 88.94        | 86.02         | 88.94      | 84.78        | 97.25   |

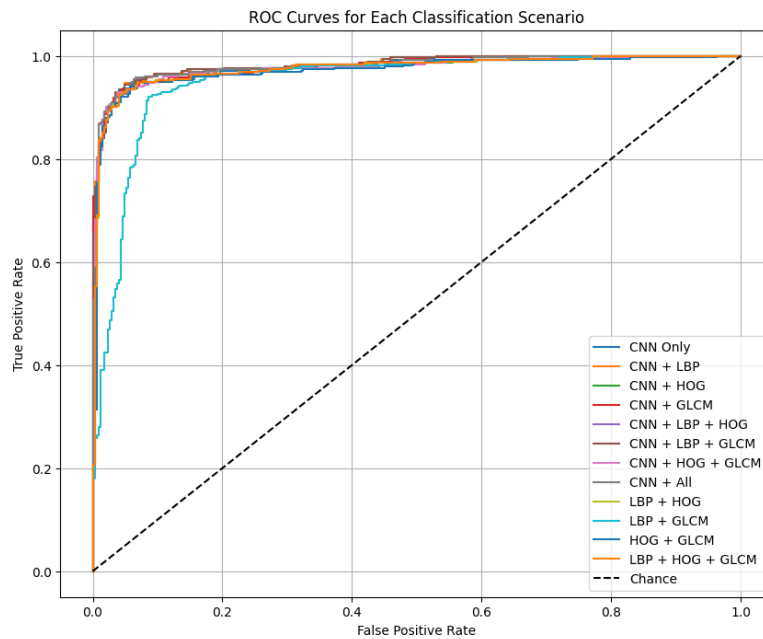


Figure 5.10: ROC curve of 1D CNN classifier (multiclass classification).

### 5.5.2.1 Discussion

The model evaluation of the classification models—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), Random Forest,

and 1D Convolutional Neural Network (CNN)—presented notable outcomes concerning the performance in multiclass and binary classification scenarios. The multiclass classification results, taken from combinations of CNN features and handcrafted features such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gray Level Co-occurrence Matrix (GLCM), are very meaningful for deducing the capabilities of these models.

### **5.5.2.2 All-Inclusive Performance**

In multiclass classification, the highest accuracy achieved was 91.81% using a combination of CNN and LBP + HOG features together with the 1D CNN model. This performance indicates the integration of the hand-crafted features further enhances the model's ability to capture diverse information, especially helpful in the distinction among multiple classes.

### **5.5.2.3 Comparison with Binary Classification**

In contrast, binary classification results showed a very strong maximum accuracy of 98%. This difference displays the inherent hardness in multiclass classification, where the model needs to distinguish between a lot more classes and hence be prone to misclassification. The binary classification task allows for simpler decision boundaries and thus can be achieved with a higher accuracy rate by models.

### **5.5.2.4 KNN Analysis**

In the multiclass setting, KNN achieved its best accuracy of 89.76% with the combination of CNN and LBP + HOG. This result suggests that while KNN remains competitive in multiclass tasks, its performance is lower than that observed in binary classification, where higher accuracy rates are more easily attainable due

to fewer classes. The reliance on local information in KNN can lead to increased noise and errors when distinguishing among several classes.

### **5.5.2.5 SVM Analysis**

The SVM model also performed exceedingly well in the multiclass classification, producing an accuracy of 90.72% with the combination of CNN, LBP, and HOG. This shows that SVM has the ability to create complicated decision boundaries, especially when the number of classes is large. However, the gap between its performance in the multiclass and binary classifications points out that SVM would perform even better in the tasks that have fewer classes.

### **5.5.2.6 MLP Analysis**

In the case of multiclass classification, the MLP model recorded 90.44% accuracy using CNN + LBP + HOG features. The architecture of MLP enables it to learn the intricate relationships in the data. Still, it suffers from problems inherent in all models as they scale up to multiclass issues. The performance metrics indicate that though MLP can effectively capture complex patterns, tuning and optimization are paramount in enhancing accuracy.

### **5.5.2.7 Random Forest Analysis**

The accuracy of Random Forest at 89.76% in the multiclass domain shows its strength, even when the accuracy is not at the best among all models. The ensemble nature of Random Forest lends stability to many combinations of features; however, its performance could be less sensitive to the subtleties of the multiclass relationships compared to more tailored approaches like SVM or MLP.

#### 5.5.2.8 1D CNN Analysis

The 1D CNN model, with the highest accuracy at 91.81%, very much exhibits the strong performance of deep learning architectures in capturing both CNN features and hand-crafted information. This model, by being able to harness such diverse feature representations, gives it an upper hand in multiclass scenarios and shows the effectiveness of advanced architectures on more involved classification tasks. Conclusion The results show a clear distinction in the capabilities of the models with regard to binary versus multiclass classification tasks. While the highest accuracy in binary classification reached an impressive 98%, the multiclass classification results, with a maximum accuracy of 91.81%, showed the added complexity and challenges associated with the differentiation between classes. The present analysis underscores the importance of feature selection and model architecture to achieve the best possible performance, especially for multiclass settings. Future research may focus on refining feature combinations or exploring novel deep learning techniques to further enhance accuracy in challenging multiclass classification scenarios.

#### 5.5.3 Comparaison with other works

The proposed method demonstrates a balanced performance compared to other works as you can see in the table below. In terms of accuracy, the proposed method achieves 91.81%, which, while slightly lower than Shamsan [82] at 99.1% and Alshahrani [81] at 99.7%, is still higher than Bogacsovics [79] at 80.5% and Harangi [78] at 90.07%. This highlights that while the accuracy of the proposed method does not lead, it remains competitive.

A key strength of the proposed method lies in its AUC performance, where it achieves 98.47%. This result is second only to Shamsan [82] at 99.56% and

Alshahrani [81] at 99.86%, indicating the proposed method's strong capability to distinguish between classes. The high AUC demonstrates the reliability of the proposed method in scenarios where class separation is critical, such as medical diagnosis.

The proposed method also stands out for its effective use of handcrafted features. It combines LBP, GLCM, and HOG, features that are known for their strong descriptive power, especially in capturing texture and structural information. While other works like Shamsan [82] and Alshahrani [81] utilize features such as FCH, LBP, and GLCM, the inclusion of HOG in the proposed method provides a unique advantage.

Another distinguishing aspect is the use of a multi-classifier approach, including SVM, KNN, MLP, Random Forest, and 1D CNN. This flexibility contrasts with Shamsan [82] and Alshahrani [81], which rely on single classifiers such as ANN or SVM. The diverse classifier set employed by the proposed method enhances its adaptability to varying dataset characteristics, making it versatile for different application scenarios.

Overall, the proposed method excels in its AUC performance and classifier adaptability, showcasing its robustness and reliability. Although the accuracy metric leaves room for improvement, the balanced integration of handcrafted and deep features, combined with its multi-classifier framework, positions the proposed method as a solid and versatile approach. With further optimization, particularly in accuracy, the proposed method could set a new benchmark for performance in this domain.

Table 5.11: Comparison with the other works

| Author, Year           | Deep features                       | Handcrafted features      | Classifier                                   | Accuracy (%) | AUC (%) |
|------------------------|-------------------------------------|---------------------------|----------------------------------------------|--------------|---------|
| Harangi, 2019 [78]     | AlexNet                             | AM-FM                     | Softmax                                      | 90.07        | /       |
| Bogacsovics, 2022 [79] | AlexNet<br>MobileNetV3<br>ResNet-50 | AM-FM                     | AlexNet<br>MobileNetV3<br>ResNet-50          | 80.5         | /       |
| Ishtiaq, 2023 [80]     | GraphNet124<br>ResNet-50            | LBP                       | SVM<br>KNN                                   | 98.85        | /       |
| Alshahrani, 2023 [81]  | ResNet-18<br>GoogLeNet              | FCH<br>GLCM<br>LBP        | SVM                                          | 99.7         | 99.86   |
| Shamsan, 2023 [82]     | DenseNet-121<br>AlexNet             | DWT<br>LPB<br>FCH<br>GLCM | ANN                                          | 99.1         | 99.56   |
| Proposed method        | DenseNet-121                        | LBP<br>GLCM<br>HOG        | SVM<br>KNN<br>MLP<br>Random forest<br>1D CNN | 91.81        | 98.47   |

## 5.6 Conclusion

Combining deep and handcrafted features proves to be a compelling strategy for diabetic retinopathy classification. This hybrid approach capitalizes on the strengths of both modern deep networks and traditional domain-specific descriptors, enabling a richer, more balanced representation of retinal pathology. The result is improved classification accuracy and greater reliability, especially in complex or borderline cases.

## 6 | Conclusion

### 6.1 Thesis summary

This thesis in summary is about the classification of diabetic retinopathy based on two state-of-the-art methodologies: the Grad-CAM attention-guided mechanism and deep/handcrafted feature fusion. One of the major complications of diabetes is diabetic retinopathy, which may lead to blindness if left untreated. Early detection and accurate classification of DR severity are crucial to the care of patients and the possibility of effective treatment; hence, this study strived to contribute to the development of more accurate and reliable diagnostic tools.

### 6.2 Extracted contributions

The first to be implemented was Grad-CAM, which is the gradient-weighted Class Activation Mapping that helps to provide an attention-guided mechanism in visualizing and interpreting what areas of retinal images contribute most significantly to the classification decisions. Grad-CAM enhances interpretability of CNNs by highlighting spatial regions within retinal images that the model identifies as most indicative of DR severity. This approach not only provided high-performance classification but also insight into the decision-making process of the neural network, thereby supporting a more transparent AI model. The Grad-CAM highlights the areas of focus, which thus allows clinical validation and ensures that the model is identifying pathologically relevant regions of the retina. Results: This attention mechanism significantly improved the sensitivity and specificity of DR classification, which shows the potential of this method in helping clinicians diagnose and plan treatment.

The second method fused the deep and handcrafted features for more extensive feature representation in retinal images. Deep features were extracted utilizing DenseNet-121 and further complemented with hand-crafted texture features, such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gray Level Co-occurrence Matrix (GLCM) in this fusion approach. In this way, this fusion approach could offer a multi-dimensional view of DR by capturing high-level semantic features and low-level texture characteristics in retinal images. These handcrafted features, which are known to capture fine-grained patterns and textures in images, complemented the CNN-extracted features, which primarily capture broader, spatial hierarchies. This fusion approach leveraged the unique strengths of each feature type, contributing to a more robust classification model that achieved substantial improvements in classification accuracy and precision across different DR severity levels.

In particular, both methodologies provided great performance, while each of them excelled in different aspects of classification. Not only did the Grad-CAM attention mechanism yield a high classification accuracy, but it also brought the critical interpretability features to address the "black-box" problem usually associated with CNN-based models. Interpretability is very important, especially in the medical field, since it greatly impacts clinical adoption and trust in AI systems. On the other hand, the deep/handcrafted feature fusion model provided a very effective way for handling feature diversity by enriching the model generalization capabilities over different presentations of DR. Incorporating different types of features allowed the creation of a richer feature space, which generally improved the classification performance over several classes of DR.

These two methodologies have been successful and, hence, these key contributions can be presented. First, the use of Grad-CAM has shown the potential of attention-guided mechanisms in creating more interpretable AI models for med-

ical image analysis. By pinpointing regions of the retinal images that are highly indicative of DR, this method demonstrates a pathway toward clinically reliable AI tools. Second, deep and handcrafted feature fusion shows that it is very important to fuse different types of features in capturing the complexity present in medical images, especially in diseases like DR, where variations may be subtle but diagnostically important.

Both approaches, being unique, also underline the potential of fusing advanced deep learning with domain-specific knowledge in feature engineering. The results from this study show that likely neither deep learning nor traditional feature extraction alone can fully grasp the intricacies of DR; it is only by integration—hybrid approach delving into the attention mechanisms and melding the same with deep feature extraction and engineered feature crafting—that truly superior classification performance can be achieved, nudging at the boundaries of current DR detection systems.

### **6.3 Future prospects**

Finally, this thesis opens up pathways for future research in several directions: deeper combination of deep learning and hand-crafted features possibly with other feature extraction techniques or additional neural network architectures to capture even finer aspects of DR, as well as further refinements of attention-guided methods aiming at improving the precision of spatial focus within images. Advanced techniques, such as the combination of improved attention layers with Grad-CAM, might yield better results in interpretability and diagnostic performance. Moreover, future studies can leverage these methods for other retinal diseases or even other areas of medical imaging.

In a nutshell, the promising results of both the Grad-CAM attention-guided

mechanism and the deep/handcrafted feature fusion approach give clear evidence of the effectiveness of the proposed methods in DR classification. With this research, there is a contribution of valuable methodologies in the field of medical image analysis for bettering early detection and classification of diabetic retinopathy by addressing both interpretability and feature diversity. With more validation and refinement, these methods could define a new standard in automated DR screening and yield real improvements in patient care and outcomes.

# Bibliography

- [1] International Diabetes Federation. *IDF Diabetes Atlas*. 10th edition, 2021.
- [2] D. A. Antonetti, R. Klein, and T. W. Gardner. Diabetic retinopathy. *New England Journal of Medicine*, 366:1227–1239, 2012.
- [3] N. Cheung, P. Mitchell, and T. Y. Wong. Diabetic retinopathy. *The Lancet*, 376(9735):124–136, 2010.
- [4] American Academy of Ophthalmology. Diabetic retinopathy preferred practice pattern. *Ophthalmology*, 125(10):P1–P85, 2018.
- [5] D. S. Fong, L. P. Aiello, F. L. Ferris, and R. Klein. Diabetic retinopathy. *Diabetes Care*, 27(10):2540–2553, 2004.
- [6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007.
- [7] C. A. Pickover. La fabuleuse histoire de l’intelligence artificielle des automates aux robots humanoïdes. *Soins*, 66(855):9, 2021.
- [8] P. Kim. *Matlab deep learning: With machine learning, neural networks and artificial intelligence*. Apress Berkeley, CA, 2017.
- [9] G. Saint-Cirgue. *Apprendre le Machine Learning en une semaine*. Machine Learnia, 2019.
- [10] G. Keifer and F. Effenberger. Big data et machine learning. 6(11), 1967.

## BIBLIOGRAPHY

---

- [11] A. Chaudhary, K. S. Chouhan, J. Gajrani, and B. Sharma. *Deep Learning With PyTorch in Machine Learning and Deep Learning in Real-Time Applications*. 2020.
- [12] H. Kinsley and D. KukiÅa. *Neural Networks from Scratch in Python*. 2020.
- [13] Rakotomalala. Ricco rakotomalala tutoriels tanagra-[http://tutoriels-datamining.blogspot.fr/perceptrons simples et multicouches](http://tutoriels-datamining.blogspot.fr/perceptrons-simples-et-multicouches).
- [14] A. Romero. *Assisting the training of deep neural networks with applications to computer vision*. PhD thesis, University of Barcelona, 10 2015.
- [15] C.-A. Azencott. *Introduction au Machine Learning*. Dunod, France, 2018.
- [16] T. Okatani. Deep learning. *Machine Learning Professional Series, Kodansha Ltd*, 33(2), 2015.
- [17] M. Bertalmío, editor. *Denoising of Photographic Images and Video Fundamentals, Open Challenges and New Trends*. Springer, UK, 2018.
- [18] M. S. Fukushima Kunihiro. Neocognitron: a self-organizing neural network model for mechanism of visual pattern recognition. *NHK Broadcasting Science Research Laboratories*, 1981.
- [19] NetApp. Les autos encodeurs et les réseaux de neurones dits convolutif.
- [20] H. Habibi Aghdam, E. Jahani Heravi, and S. I. P. Ag. Guide to convolutional neural networks a practical application to traffic-sign detection and classification. 2018.
- [21] Harry Pratt, Frans Coenen, Deborah M. Broadbent, Simon P. Harding, and Yalin Zheng. Convolutional neural networks for diabetic retinopathy. *Pro-*

## BIBLIOGRAPHY

---

- cedia Computer Science*, 90:200–205, 2016. 20th Conference on Medical Image Understanding and Analysis (MIUA 2016).
- [22] E. Nasr-Esfahani, S. Samavi, N. Karimi, S.M.R. Soroushmehr, M.H. Jafari, K. Ward, and K. Najarian. Melanoma detection by analysis of clinical images using convolutional neural network. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1373–1376, 2016.
- [23] Halgurd S. Maghdid, Aras T. Asaad, Kayhan Zrar Ghafoor, Ali Safaa Sadiq, Seyedali Mirjalili, and Muhammad Khurram Khan. Diagnosing COVID-19 pneumonia from x-ray and CT images using deep learning and transfer learning algorithms. In Sos S. Agaian, Vijayan K. Asari, Stephen P. DelMarco, and Sabah A. Jassim, editors, *Multimodal Image Exploitation and Learning 2021*, volume 11734, page 117340E. International Society for Optics and Photonics, SPIE, 2021.
- [24] Md. Kamrul Hasan, Md. Ashraful Alam, Md. Toufick E. Elahi, Shidhartho Roy, and Sifat Redwan Wahid. Cvr-net: A deep convolutional neural network for coronavirus recognition from chest radiography images. *ArXiv*, abs/2007.11993, 2020.
- [25] Jun Xu, Xiaofei Luo, Guanhao Wang, Hannah Gilmore, and Anant Madabhushi. A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. *Neurocomputing*, 191:214–223, 2016.
- [26] Zhicheng Jiao, Xinbo Gao, Ying Wang, and Jie Li. A deep feature based framework for breast masses classification. *Neurocomputing*, 197:221–231, 2016.

## BIBLIOGRAPHY

---

- [27] Grégory Apou, Nadine S Schaadt, Benoît Naegel, Germain Forestier, Ralf Schönmeier, Friedrich Feuerhake, Cédric Wemmert, and Anne Grote. Detection of lobular structures in normal breast tissue. *Computers in biology and medicine*, 74:91–102, 2016.
- [28] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H. Beck. Deep learning for identifying metastatic breast cancer, 2016.
- [29] Rajkumar Sadagopan, Saravanan Ravi, Sairam Vuppala Adithya, and Sapthagirivasan Vivekanandhan. Polyeffnetv1: A cnn based colorectal polyp detection in colonoscopy images. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 237(3):406–418, 2023. PMID: 36683465.
- [30] A. Tahir, A. Saadia, K. Khan, A. Gul, A. Qahmash, and R.N. Akram. Enhancing diagnosis: ensemble deep-learning model for fracture detection using x-ray images. *Clinical Radiology*, 79(11):e1394–e1402, 2024.
- [31] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.
- [32] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.
- [33] Majd Zreik, Tim Leiner, Bob D. de Vos, Robbert W. van Hamersvelt, Max A. Viergever, and Ivana Isgum. Automatic segmentation of the left ventricle in cardiac ct angiography using convolutional neural network, 2017.

## BIBLIOGRAPHY

---

- [34] Zhenwei Li, Mengli Jia, Xiaoli Yang, and Mengying Xu. Blood vessel segmentation of retinal image based on dense-u-net network. *Micromachines*, 12:1478, 11 2021.
- [35] Zhongxing Zhou, Akitoshi Inoue, Cynthia H. McCollough, and Lifeng Yu. Self-trained deep convolutional neural network for noise reduction in ct. In Joseph Webster Stayman, editor, *7th International Conference on Image Formation in X-Ray Computed Tomography*, Proceedings of SPIE - The International Society for Optical Engineering. SPIE, 2022. Publisher Copyright: © 2022 SPIE.; 7th International Conference on Image Formation in X-Ray Computed Tomography ; Conference date: 12-06-2022 Through 16-06-2022.
- [36] Sabina Umirzakova, Shabir Ahmad, Latif U. Khan, and Taegkeun Whangbo. Medical image super-resolution for smart healthcare applications: A comprehensive survey. *Information Fusion*, 103:102075, 2024.
- [37] Sarah Wild, Gojka Roglic, Anders Green, Richard Sicree, and Hilary King. Global prevalence of diabetes: Estimates for the year 2000 and projections for 2030. *Diabetes Care*, 27(5):1047–1053, 05 2004.
- [38] G. Zago. Dépistage de la rétinopathie diabétique. *la Haute Autorité de Santé, France*, 2010.
- [39] Margaux Ricard. Prise en charge de larétinopathie diabétique par l’orthoptiste. *MÃ©decine humaine et pathologie.*, 2016.
- [40] M. Khazaeni. Examens ophtalmologiques. *Loma Linda University School of Medicine*, 2019.
- [41] H. khattak, “identification of diabetic retinopathy through machine learning,” hindawi, npage 115 116. *p*, 8:16, October 2021. obre.

## BIBLIOGRAPHY

---

- [42] Dolly Das, Saroj Kr. Biswas, and Sivaji Bandyopadhyay. A critical review on diagnosis of diabetic retinopathy using machine learning and deep learning. *Multimedia Tools Appl.*, 81(18):25613â25655, July 2022.
- [43] Fond d’oeil : quand le faire, pourquoi, normal ou pas ? — passe-portsante.net. <https://www.passeportsante.net/fr/Maux/examens-medicaux-operations/Fiche.aspx?doc=fond-d-oeil>. [Accessed 16-05-2025].
- [44] Fond d’œilÂ : quand le faire, pourquoi, normal ou pasÂ ? — sante.journaldesfemmes.fr. <https://sante.journaldesfemmes.fr/fiches-anatomie-et-examens/2624417-fond-oeil-examen-quand-pourquoi-normal-glaucome-dmla/>. [Accessed 16-05-2025].
- [45] Maggie Karthik and Sohier Dane. Aptos 2019 blindness detection, 2019.
- [46] Foundation Consumer Healthcare EyePACS. Diabetic retinopathy detection, May 2023. [(accessed on 16/05/2025)].
- [47] E. Decenci‘ere, X. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, and P. Or Gain. donez, r., massin, p., erginay, a., et al. *Feedback on a publicly distributed image database: the messidor database. Image Analysis Stereology*, 33(3):231–234, 2014.
- [48] P. Porwal, S. Pachade, R. Kamble, M. Kokare, G. Deshmukh, V. Sahasrabud-dhe, and F. Meriaudeau. Indian diabetic retinopathy image dataset (idrid): A database for diabetic retinopathy screening research. *Data*, 3:25, 2018.
- [49] N. E. M. Khalifa, M. Loey, M. H. N. Taha, and H. N. E. T. Mohamed. Deep transfer learning models for medical diabetic retinopathy detection. *Acta Informatica Medica*, 27(5):327, 2019.

## BIBLIOGRAPHY

---

- [50] O. Dekhil, A. Naglah, M. Shaban, M. Ghazal, F. Taher, and A. Elbaz. Deep learning based method for computer aided diagnosis of diabetic retinopathy. In *2019 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–4, 2019.
- [51] S. H. Kassani, P. H. Kassani, R. Khazaeinezhad, M. J. Wesolowski, K. A. Schneider, and R. Deters. Diabetic retinopathy classification using a modified xception architecture. In *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 1–6, 2019.
- [52] B. Tymchenko, P. Marchenko, and D. Spodarets. Deep learning approach to diabetic retinopathy detection. arXiv preprint, 2020.
- [53] I. Qureshi, J. Ma, and Q. Abbas. Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning. *Multimedia Tools and Applications*, 80(8):11691–11721, 2021.
- [54] X. Zeng, H. Chen, Y. Luo, and W. Ye. Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. *IEEE Access*, 7:30744–30753, 2019.
- [55] T. Shanthi and R. Sabeenian. Modified alexnet architecture for classification of diabetic retinopathy images. *Computers Electrical Engineering*, 76:56–64, 2019.
- [56] 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 (ICIICT)*, Ieee, pages 1–6, 2019.
- [57] G. Quellec, K. Charriere, Y. Boudi, B. Cochener, and M. Lamard. Deep

## BIBLIOGRAPHY

---

- image mining for diabetic retinopathy screening. *Medical image analysis*, 39:178–193, 2017.
- [58] R. Gargeya and T. Leng. Automated identification of diabetic retinopathy using deep learning. *Ophthalmology*, 124(7):962–969, 2017.
- [59] M. T. Hagos and S. Kant. Transfer learning-based detection of diabetic retinopathy from small dataset. arXiv preprint, 2019.
- [60] H. Shakibania, S. Raoufi, B. Pourafkham, H. Khotanlou, and M. Mansoorizadeh. Dual branch deep learning network for detection and stage grading of diabetic retinopathy. *Biomedical Signal Processing and Control*, 93:106168, 2024.
- [61] B. N. Narayanan, R. C. Hardie, M. S. De Silva, and N. K. Kueterman. Hybrid machine learning architecture for automated detection and grading of retinal images for diabetic retinopathy. *Journal of Medical Imaging*, 7(3):034501–034501, 2020.
- [62] S. Majumder, Y. Elloumi, M. Akil, R. Kachouri, and N. Kehtarnavaz. A deep learning-based smartphone app for real-time detection of five stages of diabetic retinopathy. *Real-Time Image Processing and Deep Learning*, 1140:13–22, 2020. 2020, 1, . SPIE.
- [63] M. Shaban, Z. Ogur, A. Mahmoud, A. Switala, A. Shalaby, H. Abu Khalifeh, M. Ghazal, L. Fraiwan, G. Giridharan, H. Sandhu, et al. A convolutional neural network for the screening and staging of diabetic retinopathy. *Plos one*, 15(6):e0233514, 2020.
- [64] A. K. Gangwar and V. Ravi. Diabetic retinopathy detection using transfer learning and deep learning. *Evolution in Computational Intelligence: Fron-*

## BIBLIOGRAPHY

---

- tiers in *Intelligent Computing: Theory and Applications (FICTA, 1:679–689, 2021. 2020)*, , . Springer.
- [65] S. Ramchandre, B. Patil, S. Pharande, K. Javali, and H. Pande. A deep learn- ing approach for diabetic retinopathy detection using transfer learn- ing. In *2020 IEEE International Conference for Innovation in Technology (INOCON), IEEE*, pages 1–5, 2020.
- [66] M. Nahiduzzaman, M. R. Islam, M. O. F. Goni, M. S. Anower, M. Ahsan, J. Haider, and M. Kowalski. Diabetic retinopathy identification using parallel convo- lutional neural network based feature extractor and elm classifier. *Expert Systems with Applications*, 217:119557, 2023.
- [67] M. R. Islam, L. F. Abdulrazak, M. Nahiduzzaman, M. O. F. Goni, M. S. Anower, M. Ahsan, J. Haider, and M. Kowalski. Applying supervised con- trastive learning for the detection of diabetic retinopathy and its severity lev- els from fundus images. *Computers in Biology and Medicine*, 146:105602, 2022.
- [68] M. Jian, H. Chen, C. Tao, X. Li, and G. Wang. Triple-drnet: A triple-cascade convolution neural network for diabetic retinopathy grading using fundus images. *Computers in Biology and Medicine*, 155:106631, 2023.
- [69] Z. Wang, Y. Yin, J. Shi, W. Fang, H. Li, and X. Wang. Zoom-in-net: Deep min- ing lesions for diabetic retinopathy detection. In *Medical Image Com- puting and Computer Assisted Intervention- MICCAI 2017: 20th Interna- tional Conference, Quebec City, QC, Canada, September 11-13, 2017, Pro- ceedings, Part III 20*, pages 267–275. Springer, 2017.
- [70] Z. Zhao, K. Zhang, X. Hao, J. Tian, M. C. H. Chua, L. Chen, and X. Xu. Bira- net: Bilinear attention net for diabetic retinopathy grading. In *2019*

## BIBLIOGRAPHY

---

- IEEE International Conference on Image Processing (ICIP)*, IEEE, pages 1385–1389, 2019.
- [71] S. Madarapu, S. Ari, and K. Mahapatra. A deep integrative approach for diabetic retinopathy classification with synergistic channel-spatial and self-attention mechanism. *Expert Systems with Applications*, 249:123523, 2024.
- [72] M. D. Alahmadi. Texture attention network for diabetic retinopathy classification. *IEEE Access*, 10:55522–55532, 2022.
- [73] A. He, T. Li, N. Li, K. Wang, and H. Fu. Cabnet: Category attention block for imbalanced diabetic retinopathy grading. *IEEE Transactions on Medical Imaging*, 40(1):143–153, January 2021.
- [74] B. Yang, T. Li, H. Xie, Y. Liao, and Y.-P. P. Chen. Classification of diabetic retinopathy severity based on gca attention mechanism. *IEEE Access*, 10:2729–2739, 2022.
- [75] Xiaoling Luo, Zuhui Pu, Yong Xu, Wai Keung Wong, Jingyong Su, Xiaoyan Dou, Baikang Ye, Jiying Hu, and Lisha Mou. Mvdrnet : Multi-view diabetic retinopathy detection by combining dcnn and attention mechanisms. *Pattern Recognition*, 120:108104, 2021.
- [76] Mohamed Abderaouf Moustari, Youcef Brik, Bilal Attallah, Rafik Bouaouina, and Mohamed Djerioui. Enhancement of diabetic retinopathy classification using attention guided convolutional neural network. In *2022 International Conference of Advanced Technology in Electronic and Electrical Engineering (ICATEEE)*, pages 1–6, 2022.
- [77] Abderaouf M. Moustari, Youcef Brik, Bilal Attallah, and Rafik Bouaouina and. Two-stage deep learning classification for diabetic retinopathy using

## BIBLIOGRAPHY

---

- gradient weighted class activation mapping. *Automatika*, 65(3):1284–1299, 2024.
- [78] B. Harangi, J. Toth, A. Baran, and A. Hajdu. Automatic screening of fundus images using a combination of convolutional neural network and hand-crafted features. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany*, pages 2699–2702, 2019.
- [79] G. Bogacsovics, J. Toth, A. Hajdu, and B. Harangi. Enhancing cnns through the use of hand-crafted features in automated fundus image classification. *Biomedical Signal Processing and Control*, 76:103685, July 2022.
- [80] Uzair Ishtiaq, E. Rahayu, and Zubair Ishtiaque. A hybrid technique for diabetic retinopathy detection based on ensemble-optimized cnn and texture features. *Diagnostics*, 13(10):1816–1816, May 2023.
- [81] M. Alshahrani and M. Al-Jabbar. Ebrahim mohammed senan, ibrahim abdulrab ahmed, and jamil, “hybrid methods for fundus image analysis for diagnosis of diabetic retinopathy development stages based on fusion features. *Diagnostics*, 13(17):2783–2783, August 2023.
- [82] Ahlam Shamsan, Ebrahim Mohammed Senan, and A. Shatnawi. Predicting of diabetic retinopathy development stages of fundus images using deep learning based on combined features. *PLOS ONE*, 18(10):e0289555–e0289555, October 2023.
- [83] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

## BIBLIOGRAPHY

---

- [84] B. Garg and G. Sharma. A quality-aware energy-scalable gaussian smoothing filter for image processing applications. *Microprocessors and Microsystems*, 45:1–9, 2016.
- [85] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.
- [86] V. Selvakumar and C. Akila. Efficient diabetic retinopathy diagnosis through u-net knn integration in retinal fundus images. *Automatika*, 64(4):1148–1157, 2023.
- [87] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51, 1996.
- [88] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973. vol. SMC.
- [89] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, vol*, pages 886–893, 2005.
- [90] Corinna Cortes and Vladimir Vapnik. Support vector network. *Machine Learning*, 20:273–297, 09 1995.
- [91] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

## BIBLIOGRAPHY

---

- [92] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.