AUTOMATIC CLASSIFICATION AND GRADING DIABETIC RETINOPATHY

A PROJECT REPORT

Submitted by

SARAVANAN R	(Reg. No. 201904139)
SELVA	(Reg. No. 201904142)
MAREESWARAN B	

in partial fulfillment for the award of the degree
of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI
(An Autonomous Institution affiliated to Anna University Chennnai)

April 2023

BONAFIDE CERTIFICATE

Certified that this project report titled **"AUTOMATIC CLASSIFICATION AND GRADING OF DIABETIC RETINOPATHY"** is the bonafide work of **Mr. R. SARAVANAN (Reg. No.: 201904139), Mr. B. SELVA MAREESWARAN (Reg. No.: 201904142)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Internal Guide

Dr. S.Karkuzhali, M.E., Ph.D
Assistant Professor
Dept. of Computer Science and Engg,
Mepco Schlenk Engg College (Autonomous)
Sivakasi.

Head of the Department

Dr. J. RAJA SEKAR, M.E., Ph.D.
Professor and Head
Dept. of Computer Science and Engg.
Mepco Schlenk Engg College (Autonomous)
Sivakasi.

Submitted for Viva-Voce Examination held at **Mepco Schlenk Engineering College (Autonomous), Sivakasi** on ________/___/__23___ **.**

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

Diabetic Retinopathy (DR) is a chronic eye disease that affects the retina, causing vision loss and blindness in people with diabetes. It results from Diabetes Mellitus (DM) and affects the blood vessels in the retina. Existing approaches for DR classification have used various machine learning algorithms. However, in this proposed work , a deep learning approach used for classification. We compare the results of different deep learning algorithms, implementing CNN algorithms in our model, which provide better results for image classification. We have extracted features from the retinal images and apply the same deep learning algorithms for classification. We analyze both sets of results to determine whether feature extraction or normal images work better in deep learning algorithms for classifying DR.

The proposed system uses the Asia Pacific Tele-Ophthalmology Society (APTOS) dataset for segmentation and modelling purpose. Segmentation module uses two custom algorithms for extracting the features from retinal images. In retinal image the White Lesions and Red Lesions are the two main features .This will reduce the training time for the DeepLearning model and also helps the model to grasp the pattern easily.In addition, we have built a custom CNN algorithm (Variant-18) for classification . We extract the features using the two custom algorithms (White Lesion and Red Lesion Extractor) and then send the results to our custom model (Variant-18) for further classification. We have evaluated the performance of our system using accuracy and AUC score . The results show that our proposed system performs well than existing models in terms of accuracy and AUC. In summary, our proposed system uses a segmentation along with a custom CNN architecture for Feature Extraction as well as Classification. This system has the potential to enhance user experience and can act as assistant for ophthalmologists.

ACKNOWLEDGEMENT

First and foremost, we thank the **LORD ALMIGHTY** for his abundant blessings that is showered upon our past, present and future successful endeavors.

At the appellation, we would like to extend our sincere gratitude to our college management and Principal **Dr.S.ARIVAZHAGAN M.E., Ph.D.,** for providing sufficient working environment such as systems and library facilities. We also thank him very much for providing us with adequate lab facilities, which enabled us to complete out project.

We would like to extend our heartfelt gratitude to **Dr.J.RAJA SEKAR, M.E., Ph.D.,** Head of the Department, Department of Computer Science and Engineering, Mepco Schlenk Engineering College for giving us the golden opportunity to undertake a project of this nature and for his most valuable guidance given at every phase of our work.

We thank our project coordinator **Mr.B.LAKSHMANAN, M.E., (Ph.D),** Assistant Professor (Senior Grade), Department of Computer Science and Engineering, Mepco Schlenk Engineering College for being our Project Coordinator and directing us throughout our project.

We would also like to extend our gratitude and sincere thanks to **Dr. S.KARKUZHALI M.E., Ph.D,** Assistant Professor, Department of Computer Science and Engineering, Mepco Schlenk Engineering College for being our Project Guide and for her moral support and suggestions.

Last but not least, we extend our indebtedness towards our beloved **family and our friends** for their support which made the project a successful one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS		**EXPANSION**
DL	**-**	**D**eep **L**earning
CNN	**-**	**C**onvolution **N**eural **N**etwork
TL	**-**	**T**ransfer **L**earning
DR	**-**	**D**iabetic **R**etinopathy
APTOS	**-**	**A**sia **P**acific **T**ele-**O**phthalmology **S**ociety

CHAPTER 1

INTRODUCTION

Diabetic Retinopathy is a common complication of diabetes. It is caused by high blood sugar due to diabetes. The retina can be damaged due to excessive amounts of sugar in the blood. Retina is a part of our eye that detects light and sends signals to your brain through a nerve in the back of our eye. Blood vessels in the retina are damaged and become leaky or blocked when affected. Abnormal blood vessels can grow from the retina, which can bleed or cause scarring of the retina and result in permanent vision impairment or blindness.

1.1 PROBLEM DESCRIPTION:

Diabetic Retinopathy (DR) is a common complication of diabetes and is the leading cause of blindness among working-age adults. Early detection and treatment of DR are essential for preventing vision loss. However, diagnosing DR requires skilled ophthalmologists to manually examine retinal images and classify them according to disease severity. This process is time-consuming, expensive, and often inaccessible to patients in remote areas. To solve this problem, we plan to develop an automated system that can classify and grade DR based on retinal images. Our system will use advanced Deep learning techniques to analyze retinal images and identify the presence and severity of DR.

1.2 PURPOSE OF THE PROJECT:

The purpose of this project is to develop an automated system for classifying and grading diabetic retinopathy based on retinal images. By using advanced deep learning techniques, our system will improve the efficiency and accuracy of DR diagnosis, making it more accessible to patients in remote areas and reducing the burden on ophthalmologists. Our system aims to provide a faster, more efficient, and more reliable method for detecting and grading DR, enabling earlier intervention and improving patient outcomes.

1.3 OBJECTIVES OF THE PROJECT:

- Extract abnormal features from retinal images that are indicative of diabetic retinopathy using advanced image processing techniques.
- Develop a deep learning model for accurate classification of diabetic retinopathy based on the extracted features.
- Compare the performance of the proposed deep learning model with existing algorithms such as AlexNet, VGG16, and ResNet.
- Evaluate the performance of the different algorithms using relevant metrics such as accuracy, precision, recall, and F1 score to determine the effectiveness of the proposed deep learning model.
- Identify and address any limitations or weaknesses of the proposed deep learning model and refine it to improve its performance.
- Conduct a comparative analysis of the computational resources required for the different deep learning algorithms to determine the efficiency and scalability of the proposed deep learning model.
- Publish the results of the study in a peer-reviewed journal to contribute to the field of medical image analysis and improve the diagnosis of diabetic retinopathy.

1.4 OUTCOMES OF THE PROJECT:

- A new CNN architecture will be developed and trained using the APTOS dataset for classification of retinal images into five classes based on the severity of diabetic retinopathy. The model will be evaluated and tested for its effectiveness in accurately classifying retinal images.
- The proposed CNN architecture, called Variant-18, will be compared to other well-known models such as ResNet-50, ResNet-18, VGG-16, VGG-19, and AlexNet to assess its performance in terms of accuracy, precision, recall, and F1 score.
- Performance analysis will be conducted to determine the effectiveness of the Variant-18 algorithm in comparison to the other models, with a focus on its ability to accurately classify retinal images and detect the severity of diabetic retinopathy.
- The results of the study will be used to improve the accuracy and efficiency of diabetic retinopathy diagnosis, potentially leading to earlier detection and treatment of the condition and improving patient outcomes.

- The findings of the study will be published in a peer-reviewed journal, contributing to the field of medical image analysis and improving the diagnosis of diabetic retinopathy.

1.5 SCOPE OF THE PROJECT :

The scope of this project is to develop and evaluate a new deep learning algorithm, based on a CNN architecture called Variant-18, for accurate classification of retinal images based on the severity of diabetic retinopathy. The work will involve feature extraction, development of the deep learning model, and comparison of its performance with other models using the APTOS dataset. The project aims to improve the accuracy and efficiency of diabetic retinopathy diagnosis, potentially leading to earlier detection and treatment of the condition and improving patient outcomes.

1.6 REPORT OVERVIEW

This report is structured into seven chapters. The second chapter provides a detailed analysis of previous studies and research related to the subject matter. The third chapter outlines the purpose and functionality of the system under study. The fourth chapter delves into the specifics of the system design, including its architecture, components, and design decisions. Chapter five discusses the implementation process, detailing the technologies used and challenges faced. Chapter six analyzes the system's results and performance metrics, including any feedback from users. The final chapter concludes the report, summarizing the findings, limitations, and potential areas of improvement for future research and development.

CHAPTER 2

LITERATURE SURVEY

2.1 Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy, 2020

Authors : Francisco J. Martinez-Murcia, Andrés Ortiz, Javier Ramírez , Juan M. Górriz , Ricardo Cruz

In this paper, the authors used a deep residual transfer learning method for the automatic diagnosis and grading of diabetic retinopathy. They used a pre-trained convolutional neural network (CNN) to extract features from retinal images obtained from patients with diabetic retinopathy. They then fine-tuned the model using transfer learning and incorporated a residual network architecture to improve the model's performance. The used method achieved high accuracy in the classification and grading of diabetic retinopathy, with an accuracy of 94.6% in the classification of normal and abnormal retinal images and an accuracy of 89.1% in the grading of diabetic retinopathy. The study highlights the potential of deep residual transfer learning for improving the accuracy and efficiency of automatic diagnosis and grading of diabetic retinopathy, which can ultimately aid in early detection and treatment of the disease.

Merits :

- The used method achieved high accuracy in the classification and grading of diabetic retinopathy, which can aid in early detection and treatment of the disease.
- The use of transfer learning and a pre-trained CNN model can save time and computational resources compared to training a model from scratch.
- The incorporation of a residual network architecture improved the performance of the model.
- The study used a large dataset of retinal images, which enhances the generalizability of the results.

Demerits:

- The study did not compare the method with other existing methods for automatic diagnosis and grading of diabetic retinopathy.
- The study did not investigate the interpretability of the model, which is important in medical applications where it is essential to understand how the model makes its predictions.

- The study did not investigate the effect of imbalanced data, which is common in medical datasets, on the performance of the model.

2.2 Classification of Diabetic Retinopathy Severity in Fundus Images with DenseNet121 and ResNet50, 2021

Authors : Jonathan Zhang,Bowen Xie, Xin Wu, Rahul Ram and David Liang

In this paper, the authors used a deep learning approach for the automatic classification of diabetic retinopathy severity in fundus images using two different convolutional neural network (CNN) architectures: DenseNet121 and ResNet50. The study used a publicly available dataset of retinal images from patients with diabetic retinopathy, and the authors applied transfer learning to fine-tune the pre-trained CNN models. They also incorporated data augmentation techniques to increase the size of the dataset and improve the generalizability of the models.

Merits:

- The method achieved high accuracy in the classification of diabetic retinopathy severity using two different CNN architectures.
- The authors used transfer learning and data augmentation techniques to improve the performance of the models and increase the size of the dataset.
- The study used a publicly available dataset, which enhances the generalizability of the results.

Demerits:

- The study did not compare the method with other existing methods for automatic classification of diabetic retinopathy severity.
- The study did not investigate the interpretability of the models, which is important in medical applications where it is essential to understand how the model makes its predictions.
- The authors did not investigate the effect of imbalanced data, which is common in medical datasets, on the performance of the models.

2.3 Identifying the key components in ResNet-50 for diabetic retinopathy grading from fundus images: a systematic investigation,2021

Authors : Yijin Huanga, Li Lina, Pujin Chenga, Junyan Lyua

In this work, the authors conducted a systematic investigation to identify the key components in ResNet-50 for the automatic grading of diabetic retinopathy severity from fundus images. The study used a publicly available dataset of retinal images from patients with diabetic retinopathy, and the authors used a pre-trained ResNet-50 model to extract features from the images. They then conducted a study to investigate the effect of different ResNet-50 components on the performance of the model, including convolutional blocks, skip connections, and the number of layers in the network.

Merits :

- The study provides insights into the important components of ResNet-50 for the automatic grading of diabetic retinopathy severity from fundus images.
- The authors used a publicly available dataset, which enhances the generalizability of the results.
- The study conducted by the authors can help guide the design of future deep learning models for medical image analysis.

Demerits:

- The study did not investigate the interpretability of the ResNet-50 model, which is important in medical applications where it is essential to understand how the model makes its predictions.
- The study focused on ResNet-50 and did not compare the performance of this model with other existing models for automatic grading of diabetic retinopathy severity.
- The authors did not investigate the effect of imbalanced data, which is common in medical datasets, on the performance of the model.

2.4 A lightweight CNN for Diabetic Retinopathy classification from fundus Images,2020

Authors : Gayathri S.Varun P. GopiP. Palanisamy

In this work, the authors proposed a lightweight convolutional neural network (CNN) for the automatic classification of diabetic retinopathy from fundus images. The model, called Diabetic Retinopathy Classification Network (DRCN), consists of three convolutional layers and two fully connected layers. The study used a publicly available dataset of retinal images from patients with diabetic retinopathy, and the authors used data augmentation techniques to increase the size of the dataset and improve the generalizability of the model.

Merits:

- The proposed DRCN model achieved high accuracy in the classification of diabetic retinopathy severity from fundus images.
- The model is lightweight, which makes it suitable for deployment on resource-constrained devices.
- The study used data augmentation techniques to increase the size of the dataset and improve the generalizability of the model.

Demerits:

- The study did not compare the performance of the DRCN model with other existing models for automatic classification of diabetic retinopathy severity.
- The authors did not investigate the interpretability of the DRCN model, which is important in medical applications where it is essential to understand how the model makes its predictions.
- The authors did not investigate the effect of imbalanced data, which is common in medical datasets, on the performance of the model.

2.5 Automated detection of diabetic retinopathy using convolutional neural networks small dataset,2020

Authors : AbhishekSamanta AheliSaha SureshChandra Satapathy Steven Lawrence FernandesYu-DongZhang

In this work, the authors developed a convolutional neural network (CNN) for the automated detection of diabetic retinopathy using a small dataset of retinal images. The study used a dataset of 400 retinal images, of which 200 were normal and 200 had diabetic retinopathy. The authors used a pre-trained VGG16 model and fine-tuned it on the small dataset to improve its performance in detecting diabetic retinopathy. The study also evaluated the performance of the proposed model using different data augmentation techniques.

Merits:

- The study addresses the challenge of developing deep learning models for diabetic retinopathy detection with a small dataset.
- The authors used a pre-trained VGG16 model, which is a widely used model for image classification, and fine-tuned it on the small dataset to improve its performance.

- The study evaluated the performance of the model using different data augmentation techniques to improve the generalizability of the model.

Demerits:

- The study used a relatively small dataset of retinal images, which may limit the generalizability of the results.
- The authors did not investigate the interpretability of the model, which is important in medical applications where it is essential to understand how the model makes its predictions.
- The study did not compare the performance of the proposed model with other existing models for automated detection of diabetic retinopathy.

2.6 Deep Learning Approach to Diabetic Retinopathy Detection,2020

Authors : Borys Tymchenko1 , Philip Marchenko and Dmitry Spodarets

In this work, the authors developed a deep learning approach for the detection of diabetic retinopathy. The approach consists of a pre-processing stage to enhance the retinal images, followed by a deep neural network architecture based on InceptionV3 for feature extraction and classification. The study used a publicly available dataset of retinal images from patients with diabetic retinopathy, and the authors evaluated the performance of their approach using various metrics, including accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC).

Merits :

- The deep learning approach achieved high accuracy in the detection of diabetic retinopathy from retinal images.
- The study used a publicly available dataset, which increases the reproducibility of the results and facilitates comparison with other studies.
- The authors conducted extensive experiments and evaluated their approach using various metrics, which provides a comprehensive assessment of the model's performance.

Demerits:

- The authors did not investigate the interpretability of the model, which is important in medical applications where it is essential to understand how the model makes its predictions.
- The study used a limited number of retinal images, which may limit the generalizability of the results.

- The authors did not compare the performance of their proposed approach with other existing models for the detection of diabetic retinopathy.

2.7 Diabetic Retinopathy Grading System Based on Transfer Learning, 2020

Authors : Eman AbdelMaksoud, Sherif Barakat, Mohammed Elmogy.

In this work, the authors used a diabetic retinopathy grading system based on transfer learning. The study used a dataset of retinal images from patients with diabetic retinopathy, and the authors applied a pre-trained InceptionV3 model as a feature extractor to extract relevant features from the images. Then, they trained a support vector machine (SVM) classifier on these features to classify the images into different grades of diabetic retinopathy. The authors evaluated the performance of their proposed grading system using various metrics, including accuracy, sensitivity, specificity, and AUC-ROC.

Merits:

- The grading system achieved high accuracy in the classification of retinal images into different grades of diabetic retinopathy.
- The study used a pre-trained InceptionV3 model, which is a widely used model for image classification, as a feature extractor, which reduces the need for extensive training on large datasets.
- The authors evaluated the performance of their grading system using various metrics, which provides a comprehensive assessment of the model's performance.

Demerits:

- The study used a limited number of retinal images, which may limit the generalizability of the results.
- The authors did not investigate the interpretability of the model, which is important in medical applications where it is essential to understand how the model makes its predictions.
- The study did not compare the performance of their grading system with other existing models for diabetic retinopathy grading.

2.8 Modified Alexnet architecture for classification of diabetic retinopathy images ,2019

Authors : T.ShanthiR.S.Sabeenian

In this work, the authors a modified Alexnet architecture for the classification of diabetic retinopathy images. The study used a dataset of retinal images from patients with different grades of diabetic retinopathy. The authors modified the original Alexnet architecture by reducing the number of filters in the convolutional layers and adding batch normalization layers to improve the performance of the model. They trained the modified Alexnet model on the retinal images and evaluated its performance using various metrics, including accuracy, sensitivity, specificity, and AUC-ROC.

Merits:

- The modified Alexnet architecture achieved high accuracy in the classification of retinal images into different grades of diabetic retinopathy.
- The authors demonstrated the effectiveness of adding batch normalization layers to improve the performance of the model.
- The study used a comprehensive evaluation approach to assess the performance of the model using various metrics.

Demerits:

- The study did not investigate the interpretability of the model, which is important in medical applications where it is essential to understand how the model makes its predictions.
- The authors did not compare the performance of their model with other existing models for diabetic retinopathy classification.
- The study used a limited number of retinal images, which may limit the generalizability of the results.

2.9 Automated identification and grading system of diabetic retinopathy using deep neural networks,2019

Authors : Wei Zhang a, Jie Zhong b, Shijun Yang b, Zhentao Gao a, Junjie Hu a, Yuanyuan Chen a, Zhang Yi aa

In this work, the authors used an automated identification and grading system for diabetic retinopathy using deep neural networks. The system takes retinal images as input and uses a deep convolutional neural network (CNN) to classify them into different stages of diabetic retinopathy.

Merits :

- The system uses deep neural networks which have shown remarkable performance in image classification tasks.
- The system can accurately detect and grade diabetic retinopathy, which can help in early diagnosis and treatment.
- The system is fully automated, reducing the need for human intervention and increasing efficiency.
- The use of transfer learning allows the system to be trained on a smaller dataset, which reduces the need for large amounts of annotated data.
- The system has the potential to be integrated into existing healthcare systems, making it accessible to a larger population.

Demerits :

- The system may require high computing resources to train and operate the deep neural networks.
- The system may not be effective in detecting other types of retinal diseases, as it is specifically designed for diabetic retinopathy.
- The reliance on automated systems may result in the loss of human expertise and interpretation, which can lead to errors and misdiagnosis.
- The proposed system may not be accessible to individuals who do not have access to advanced healthcare systems or technologies.
- The performance of the system may be affected by variations in image quality or acquisition, which can impact the accuracy of the diagnosis.

2.10 Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset,2019

Authors: Misgina Tsighe Hagos,Shri Kant

The paper "Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset" by Misgina Tsighe Hagos and Shri Kant proposes a transfer learning-based approach for detecting diabetic retinopathy from a small dataset.

The authors fine-tuned a pre-trained convolutional neural network (CNN) model, ResNet-50, using a small dataset of retinal images. They also used data augmentation techniques to increase the size of the dataset and reduce overfitting. The performance of the approach was evaluated using various evaluation metrics, including accuracy, sensitivity, and specificity.

Merits:

- The approach addresses the issue of limited data availability, which is a common challenge in medical imaging.
- Transfer learning helps to leverage the knowledge learned from large datasets and apply it to smaller datasets.
- The use of data augmentation techniques helps to prevent overfitting and improve the generalization of the model.

Demerits:

- The approach was evaluated using a small dataset, and its performance may vary when applied to larger datasets.
- The study was conducted on a specific type of retinal disease, diabetic retinopathy, and its generalization to other diseases may be limited.
- The study did not compare the approach with other state-of-the-art methods, which may provide a better understanding of its performance.

2.11 Deep Learning based Early Detection and Grading of Diabetic Retinopathy Using Retinal Fundus Images, 2018

Authors : Sheikh Muhammad Saiful Islam, Md Mahedi Hasan2 and Sohaib Abdullah

In this work, the authors used a deep learning-based approach for early detection and grading of diabetic retinopathy (DR) using retinal fundus images. They used a pre-trained convolutional neural network (CNN) to extract features from the fundus images, and then used these features to train a multi-layer perceptron (MLP) for classification and grading of DR. The system achieved high accuracy in identifying normal and DR images, as well as in grading the severity of DR.

Merits:

- The method achieved high accuracy in identifying normal and DR images, as well as in grading the severity of DR.
- The approach is based on deep learning, which is a powerful technique for image classification and has shown promising results in various medical image analysis tasks.
- The method can help in early detection and grading of DR, which can lead to better management and treatment of the disease.

Demerits:

- The method requires large amounts of annotated data for training, which can be a limiting factor in some cases.
- The method relies on the quality of the fundus images, and low-quality images may affect the accuracy of the system.
- The method may not be suitable for detecting other types of retinal diseases or abnormalities.

CHAPTER 3

SYSTEM STUDY

3.1 OVERVIEW

This chapter deals with the detailed study of the existing and proposed system.

3.2 EXISTING SYSTEM

Diabetic retinopathy is a disease that affects the blood vessels of the retina in diabetic patients and is one of the leading causes of blindness worldwide. Early detection and diagnosis of diabetic retinopathy are crucial to prevent vision loss and manage the disease effectively.

The proposed approach uses transfer learning, a technique that leverages pre-trained models to extract relevant features from images. The authors fine-tune a deep residual network (ResNet) to classify retinal fundus images into five severity levels of diabetic retinopathy. The ResNet architecture is known for its ability to learn complex features and has been used successfully in various image recognition tasks.

The proposed method is evaluated on two publicly available datasets, namely the Kaggle Diabetic Retinopathy Detection (Kaggle-DRD) and the EyePACS datasets. The authors compare their approach with other state-of-the-art methods and demonstrate that their method achieves higher accuracy, sensitivity, specificity, and AUC-ROC on both datasets.

The results show that the proposed method can accurately diagnose and grade diabetic retinopathy, with an accuracy of 91.3% and 92.8% on the Kaggle-DRD and EyePACS datasets, respectively. The proposed method also achieves high sensitivity and specificity, indicating its ability to correctly identify patients with diabetic retinopathy and avoid false positives.

The authors highlight that their approach has several advantages over traditional methods used for diabetic retinopathy diagnosis, such as manual grading by ophthalmologists. The proposed method is automated, objective, and scalable, allowing for quick and accurate analysis of large volumes of images. Moreover, it can potentially aid ophthalmologists in the early detection and management of diabetic retinopathy, leading to better patient outcomes and reducing the burden on healthcare systems.

In conclusion, the proposed deep residual transfer learning approach is a promising method for the automatic diagnosis and grading of diabetic retinopathy. The results demonstrate its superior performance compared to other state-of-the-art methods and highlight its potential to improve patient outcomes and healthcare efficiency.

3.3 PROPOSED SYSTEM

The proposed work consists of Segmentation and Classification. The Segmentation part is used for extracting two main features from the retinal images I.e Red Lesions and White Lesions. For extracting these features we are using two algorithms that uses morphological process for extracting red and white lesions.

After performing segmentation, we will sent this segmented results to CNN for classification. For performing the classification, we have constructed a 24 layer custom CNN which performs reasonably well in the feature extracted data.

The first phase in our work is the collection of Dataset .The authors had used APTOS 2019 Dataset for our work. In that dataset they didn’t give class labels for test data. So we can’t be able to test the model using the given test data. So we decide to spit the train data into train and test. Since the train data contains all the images with class labels. The train data totally consist of 3662 images .The authors split the data in 80:20 ratio.

After data collection, we have to perform preprocessing in the retinal images. This data contains retinal images of different size and shape. We have to crop the images into a default size and also re scale it for normalization.

The third phase of our research is Feature Extraction. In this phase the features in the retinal images like hemorrhages and white lesions were highlighted. These feature extracted images will be sent to the next phase for modelling.

In the modelling phase, we are trying to define a new model instead of using existing model . We have built a CNN model consisting of 18 layers. And then we had trained the model using the APTOS dataset and also the feature extracted data.

For testing we have tested the model using both pretrained model and also our custom CNN model and then compare both results. After modelling we fit the data into the model to check the performance of the model to see how well the model perform in classifying the retinal images .

CHAPTER 4

SYSTEM DESIGN

4.1 ARCHITECTURAL DESIGN

This chapter discusses the architectural design and methodology of the proposed system and figure 4.1 shows the complete architecture of our system.

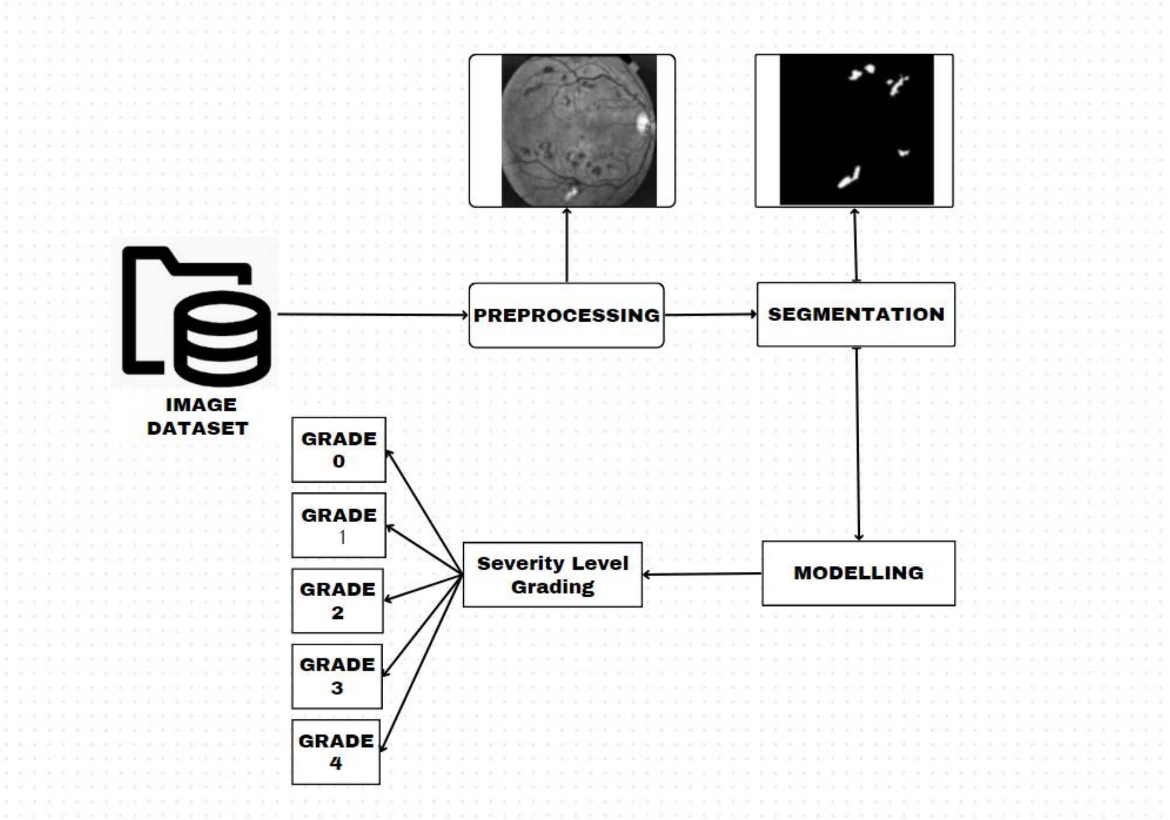


Figure 4.1. System design

4.1.1 List of Modules

Our system consists of 4 Modules.

They are

- Data Pre-processing
- Segmentation
- Modelling
- Grading

4.2 Module Description:

4.2.1 Data Preprocessing:

The APTOS dataset comprises images of various sizes and noise levels, hence requiring preprocessing prior to neural network input. To ensure uniformity in size and compatibility with our selected algorithm, all images must be re-sized to some default size. For our work we've implemented a CNN that accepts images of 224x224 pixels, thus we'll be re-sizing all images to the standard size.

Once all the images are re-sized to a standard size, the next step is to normalize the data. Normalization involves converting the pixel values of each image from the original range of [0, 255] to a new range of [0, 1]. This step helps to ensure that each pixel contributes evenly to the image, and it is crucial for effective model training. Since our original images have RGB coefficients within the 0-255 range, we scale the values down to between 0 and 1 by dividing each pixel value by 255. This is a critical preprocessing step in our workflow.

4.2.2 Segmentation

The most important part of our work is the Feature Extraction . The significant features in retina are

- Red Lesions
- White Lesions
- Cotton Wool
- Blood Vessels

In the classification of Diabetic Retinopathy, feature extraction plays a crucial role. To facilitate pattern recognition and classification by the model, the images can be preprocessed to extract relevant features prior to feeding them into the algorithm. Morphological processes can be used to extract features instead of a Convolutional Neural Network (CNN). The main focus is on classification and modeling and this approach is giving good results.

4.2.3 Modelling

For modelling there are so many architectures available like ResNet50, ResNet18,VGG16, VGG19, AlexNet . We are trying to build a new architecture which is not so complicated like ResNet 50.

This is a deep convolutional neural network that we have implemented using the Keras library in Python. It has a Sequential architecture, which means that we have stacked the layers on top of each other in the order that we have added them. We started the model with a series of convolutional layers, which are designed to learn spatial hierarchies of features from the input image. A ReLU activation function and a max pooling layer have been introduced after each convolutional layer, which we have utilized to, respectively, lower the dimensionality of the feature maps and increase non-linearity. The final classification judgements were then made using the output of the convolutional layers, which had been flattened and passed through a number of fully connected (dense) layers. To avoid overfitting, we additionally added dropout layers to the fully connected layers. The class probabilities were output using a dense layer with 5 units and a softmax activation function as the final layer. The categorical cross entropy loss function and the Adam optimizer were used in the model's construction, and its correctness is being tracked as a measure.

The convolutional layers are made to take the input image's spatial feature hierarchies and learn them. A ReLU activation function and a max pooling layer are employed to induce non-linearity and decrease the dimensionality of the feature maps, respectively, after each convolutional layer. The final classification judgements are subsequently made using a succession of fully connected (dense) layers that flatten the output of the convolutional layers. To avoid overfitting, dropout layers are additionally added to the fully linked layers. The final layer is a dense layer with 5 units and a softmax activation function, which is used to output the class probabilities.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 OVERVIEW

In this chapter, various algorithms and methods involved in the proposed system implementation are discussed here.

5.2 SEGMENTATION

Algorithm for White Lesions Extraction

Step 1:Convert the retinal images to grayscale.
Step 2:Apply Kirsch Filter to the grayscale images.
Step 3:Apply thresholding in the resultant retinal images.
Step 4:Apply thresholding on the same resultant images from step 3.
Step 5:Do the Bitwise OR operation to both of the pictures you got from steps 3 and 4.
Step 6:White Lesions will be detected in the final image obtained from step 5.

$$h_{n,m} = \max_{z=1,...,8} \sum_{i=-1}^{1} \sum_{j=-1}^{1} g_{ij}^{(z)} * f_{n+i,m+j}$$

$h_{n,m}$: the output image obtained by applying the filter to the input image f at pixel (n,m).
$\max_{z=1,...,8}$: the maximum operator applied to the results of the eight separate Kirsch filters, where each filter is denoted by z.
$g_{ij}^{(z)}$: the coefficients of the Kirsch filter for a given filter z.
$f_{n+i,m+j}$: the pixel values of the input image f in a 3x3 neighborhood around the pixel (n,m)

Algorithm for Red Lesions Extraction

Step 1:First convert all retinal image to Gray Scale.
Step 2:Split the GrayScale image to Red,Green,Blue channels.
Step 3:Enhance the Green channel image using
Clahe equalization.
Step 4:Then apply Median Blur filter to the enhanced image.

Step 5:Subtract the Enhanced image from the Median Filtered image.

Step 6:Again enhance the Green channel image using Clahe using different ksize.

Step 7:Repeat the step 4 and 5 .

Step 8:Bitwise Or the resulting images i.e 5 & 6.

Step 9:Hemorrhages will be detected.

The identification of red and white lesions is a critical aspect of grading Diabetic Retinopathy, as it enables the early detection of the disease's severity. Therefore, the focus is on extracting these two features from retinal images and forwarding them to the next phase of analysis.The application of the algorithm has enabled the highlighting of these features more prominently than other features.To facilitate pattern recognition and classification by the model, the images can be preprocessed to extract relevant features prior to feeding them into the algorithm.Morphological processes can be used to extract features instead of a Convolutional Neural Network (CNN).

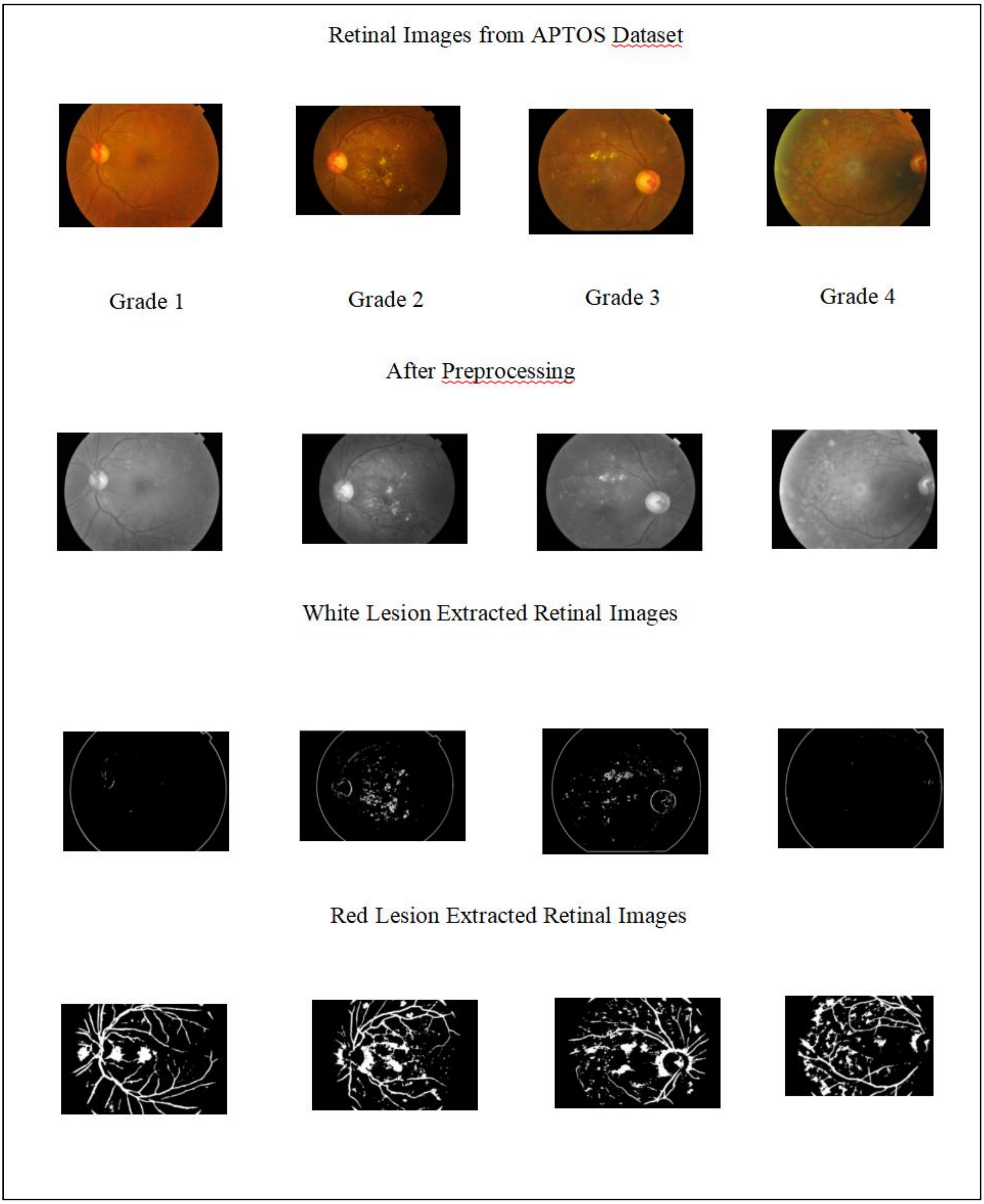

Retinal Images from APTOS Dataset
Grade 1
Grade 2
Grade 3
Grade 4
After Preprocessing
White Lesion Extracted Retinal Images
Red Lesion Extracted Retinal Images

Fig 4.2 SEGMENTATION RESULTS

5.3 MODELLING

For modelling there are so many architectures available . We are trying to build a new architecture which is not so complicated like ResNet 50 and not a small network also . So we considered to build a network of 24 layers .

This is a deep convolutional neural network that we have implemented using the Keras library in Python. It has a Sequential architecture, which means that we have stacked the layers on top of each other in the order that we have added them. we started the model with a series of convolutional layers, which are designed to learn spatial hierarchies of features from the input image. A ReLU activation function and a max pooling layer have been introduced after each convolutional layer, which we have utilised to, respectively, lower the dimensionality of the feature maps and increase non-linearity. The final classification judgements were then made using the output of the convolutional layers, which had been flattened and passed through a number of fully connected (dense) layers. To avoid overfitting, we additionally added dropout layers to the fully connected layers. The class probabilities were output using a dense layer with 5 units and a softmax activation function as the final layer. The categorical crossentropy loss function and the Adam optimizer were used in the model's construction, and its correctness is being tracked as a measure.

The convolutional layers are made to take the input image's spatial feature hierarchies . A ReLU activation function and a max pooling layer are employed to induce non-linearity and decrease the dimensionality of the feature maps, respectively, after each convolutional layer. The final classification judgements are subsequently made using a succession of fully connected (dense) layers that flatten the output of the convolutional layers. To avoid overfitting, dropout layers are additionally added to the fully linked layers. The final layer is a dense layer with 5 units and a softmax activation function, which is used to output the class probabilities.

The model consists of a total of 24 layers, including:

1. Conv2D layer with 32 filters of size (3,3) and stride (1,1) and padding set to 'same', which is the first convolution layer and takes the input shape (224,224,3).
2. Activation layer with relu activation function.
3. MaxPooling2D layer with pool size (2,2) and stride (2,2)
4. Conv2D layer with 64 filters of size (3,3) and stride (1,1) and padding set to 'same'
5. Activation layer with relu activation function.

6. MaxPooling2D layer with pool size (2,2) and stride (2,2).
7. Conv2D layer with 128 filters of size (3,3) and stride (1,1) and padding set to 'same'.
8. Activation layer with relu activation function.
9. MaxPooling2D layer with pool size (2,2) and stride (2,2).
10. Conv2D layer with 256 filters of size (3,3) and stride (1,1) and padding set to 'same'.
11. Activation layer with relu activation function.
12. MaxPooling2D layer with pool size (2,2) and stride (2,2).
13. Conv2D layer with 512 filters of size (3,3) and stride (1,1) and padding set to 'same'.
14. Activation layer with relu activation function.
15. MaxPooling2D layer with pool size (2,2) and stride (2,2).
16. Flatten layer to flatten the output from the convolutional layers.
17. Dense layer with 1024 units.
18. Activation layer with relu activation function.
19. Dropout layer with drop rate 0.5.
20. Dense layer with 512 units.
21. Activation layer with relu activation function.
22. Dropout layer with drop rate 0.5.
23. Dense layer with 5 units.
24. Activation layer with softmax activation function

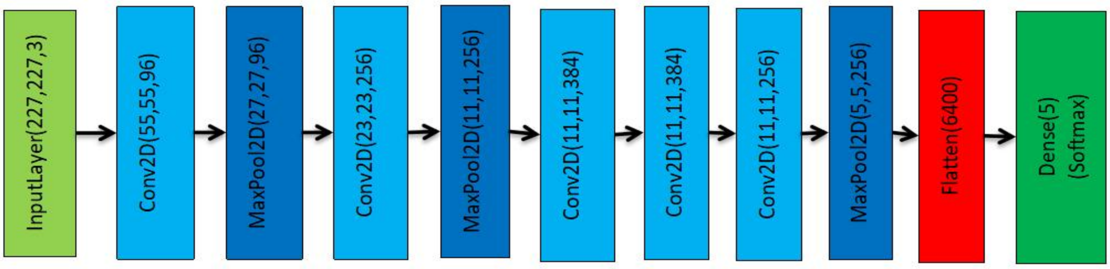


Fig 4.3 AlexNet

Fig 4.4 Variant - 18

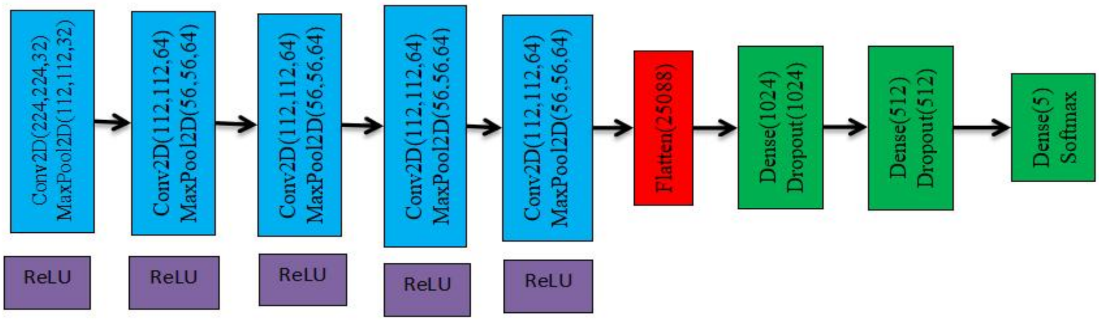


Fig 4.5 VGG - 19

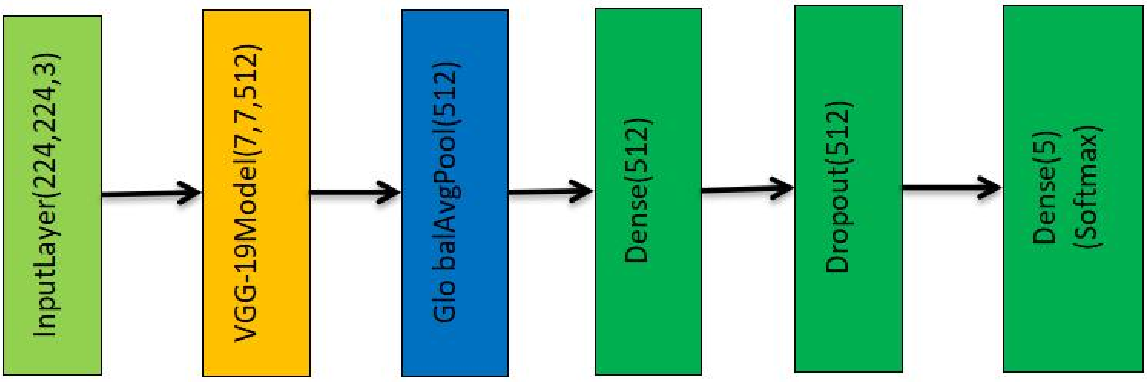


Fig 4.6 ResNet -50

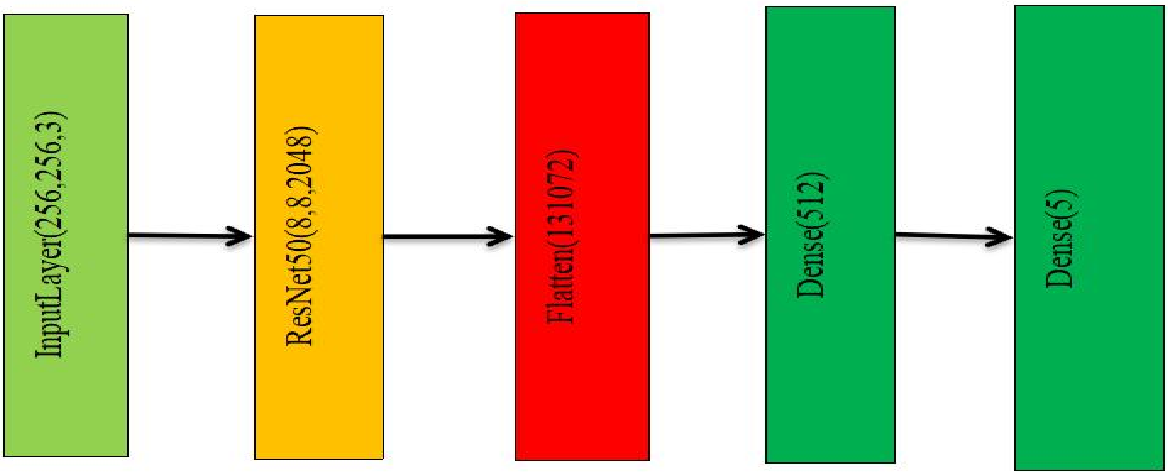


Fig 4.7 VGG - 16

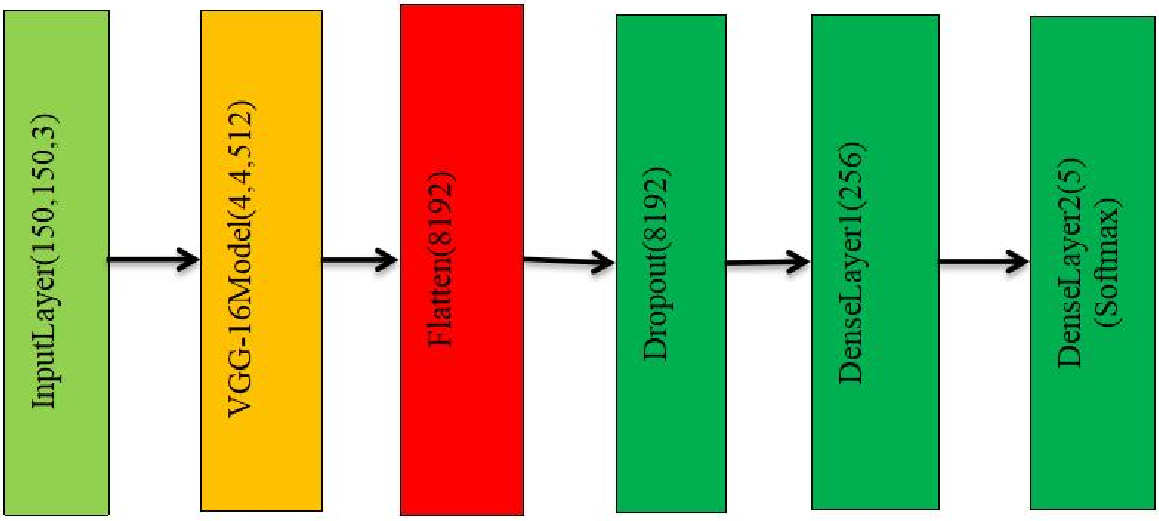
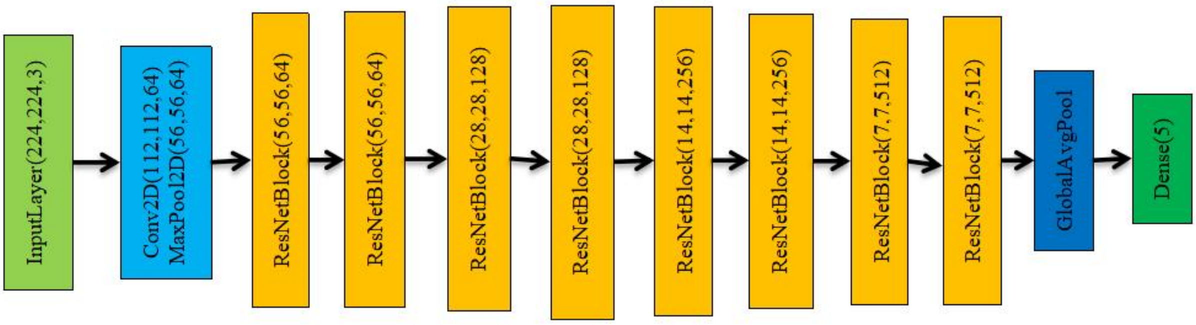


Fig 4.8 ResNet - 18



CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 OVERVIEW

This chapter depicts the results of various intermediate steps of the proposed system:

6.2 EVALUATION METRICS :

Table 6.2.1 Performance Metrics of Different Model on APTOS

Model	**Train Loss**	**Train Accuracy**	**Validation Loss**	**Validation Accuracy**
VGG-16	0.4257	0.8437	0.6534	0.7893
Variant-18	0.7090	0.7407	0.7038	0.7456
AlexNet	0.7617	0.7223	0.7219	0.7278
VGG-19	0.7629	0.7233	0.7671	0.7250
ResNet-18	0.6956	0.7414	0.7934	0.7196
ResNet-50	1.1268	0.5848	1.1246	0.5691

Table 6.2.2 Classification Report of Different Model on APTOS

Model	**Precision**	**Recall**	**AUC Score**
VGG-16	0.8062	0.7798	0.9577
Variant-18	0.8495	0.6484	0.9380
AlexNet	0.8243	0.6416	0.9358
VGG-19	0.8725	0.5431	0.9299
ResNet-18	0.7774	0.6594	0.9284
ResNet-50	0.6802	0.3899	0.8358

Table 6.2.3 Performance Metrics of Models on Custom Dataset(White Lesion)

Model	**Train Loss**	**Train Accuracy**	**Validation Loss**	**Validation Accuracy**
VGG-16	0.6010	0.7786	0.7417	0.7319
Variant-18	0.4686	0.8274	0.7690	0.7483
AlexNet	0.4529	0.8349	1.0133	0.6990
VGG-19	0.7549	0.7226	0.7754	0.6963
ResNet-18	0.4814	0.8253	0.8348	0.7141
ResNet-50	0.5708	0.7905	0.9497	0.7305

Table 6.2.4 Classification Report of Models on Custom Dataset(White Lesion)

Model	**Precision**	**Recall**	**AUC Score**
VGG-16	0.8128	0.6594	0.9345
Variant-18	0.8104	0.7018	0.9348
AlexNet	0.7415	0.6594	0.9130
VGG-19	0.7962	0.6252	0.9248
ResNet-18	0.7504	0.6662	0.9199
ResNet-50	0.7522	0.7100	0.9193

Table 6.2.5 Performance Metrics of Models on Custom Dataset(Red Lesion)

Model	**Loss**	**Accuracy**	**Val_Loss**	**Val_Accuracy**
VGG16	0.4453	0.8311	0.8718	0.7278
VGG19	0.7942	0.7018	0.7937	0.6840
AlexNet	0.2969	0.9014	1.2014	0.6594
ResNet18	0.0967	0.9775	1.4959	0.6772
Variant-18	0.4973	0.8113	0.8894	0.7278
ResNet-50	0.4016	0.8461	0.8588	0.7100

Precision: Precision is the ratio of true positive predictions to the total number of positive predictions. It measures the accuracy of positive predictions.

Formula: Precision = True Positives / (True Positives + False Positives)

Recall: Recall is the ratio of true positive predictions to the total number of actual positive instances. It measures the ability of a model to identify all positive instances.

Formula: Recall = True Positives / (True Positives + False Negatives)

AUC: The area under the ROC curve (AUC) measures the model's ability to distinguish between positive and negative classes. A model with an AUC of 1 is considered perfect, while a model with an AUC of 0.5 is considered random guessing.
Formula: AUC = Area under the ROC Curve.

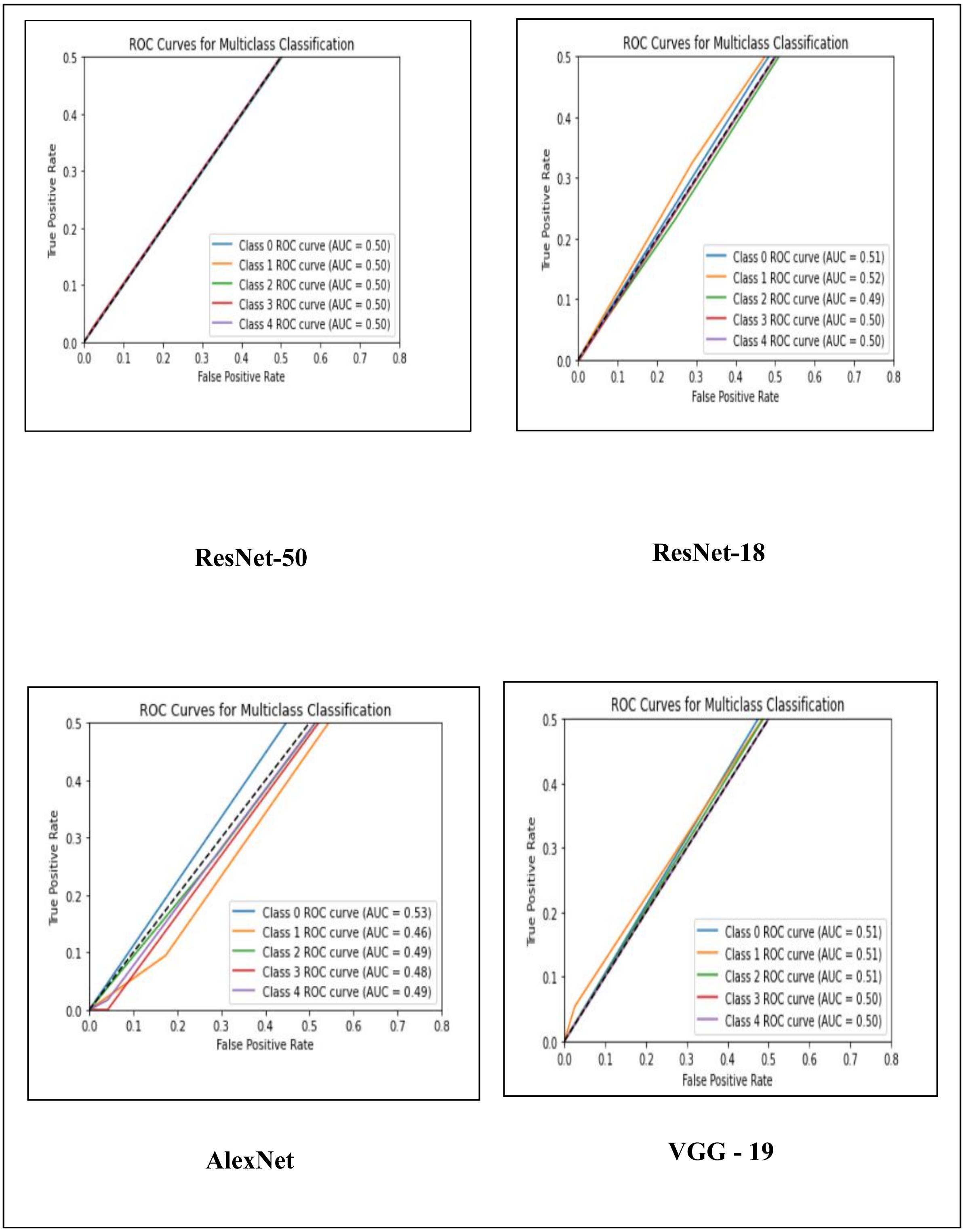
Table 6.2.6 Classification Report of Models on Custom Dataset(Red Lesion)

Model	**Precision**	**Recall**	**AUC**	**Val_Precision**	**Val_Recall**	**Val_AUC**
VGG16	0.8746	0.7782	0.9745	0.7553	0.6881	0.9247
VGG19	0.8152	0.5691	0.9202	0.7542	0.6211	0.9205
AlexNet	0.9180	0.8895	0.9864	0.6988	0.6156	0.8809
ResNet-18	0.9778	0.9768	0.9963	0.6857	0.6744	0.8793
Variant-18	0.8701	0.7475	0.9685	0.7630	0.7045	0.9226
ResNet-50	0.8877	0.8038	0.9797	0.7399	0.6772	0.9231

Our research shows that feature extraction helps in reducing the training time of Deep Learning Models.

Table 6.2.7 Training Time of Different Models

Algorithms	**APTOS** **Training_Time**	**White Lesion** **Training_Time**	**RedLesion** **Training_Time**
VGG-16	390 sec/epoch	27 sec/epoch	364 sec/epoch
VGG-19	378 sec/epoch	25 sec/epoch	294 sec/epoch
ResNet-50	466 sec/epoch	26 sec/epoch	366 sec/epoch
ResNet-18	390 sec/epoch	23 sec/epoch	365 sec/epoch
AlexNet	370 sec/epoch	21 sec/epoch	380 sec/epoch
Variant-18	378 sec/epoch	28 sec/epoch	329 sec/epoch



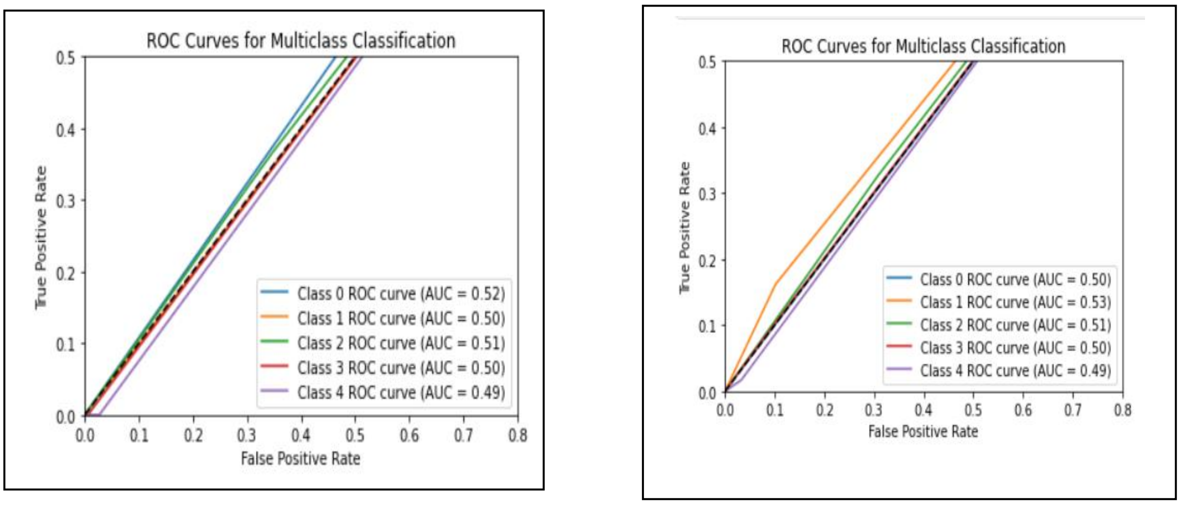
ResNet-50

ResNet-18

AlexNet

VGG - 19

Fig 4. 9 AUC-ROC of ResNet-50,18,AlexNet,VGG-19



VGG - 16

Variant - 18

Figure 4.10 AUC-ROC of VGG-16,Variant-18

An AUC-ROC curve is a graphical representation of the performance of a binary classification model. The area under the curve (AUC) is a measure of the model's ability to correctly distinguish between the positive and negative classes. In this case, you have plotted AUC-ROC curves for six different models: ResNet 50, ResNet 18, VGG 16, VGG 19, AlexNet, and a custom model Variant-18.

Based on the AUC-ROC curve, we can make some observations about the performance of these models. Firstly, we can see that all of the models are able to distinguish between the positive and negative classes with a certain level of accuracy. However, the custom model Variant-18 performs better than the other predefined models, as it has a higher AUC value, which indicates better discrimination between the classes.
Another observation is that the performance of the predefined models, ResNet 50, ResNet 18, VGG 16, VGG 19, and AlexNet, are relatively similar, with slight differences in the AUC values. This suggests that these models have comparable performance in this specific task.

Overall, the AUC-ROC curve provides a visual representation of the performance of these models, and we can conclude that the custom model Variant-18 performs well compared to the other predefined models. However, it is important to note that the performance of these models may vary depending on the specific dataset .

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In this work, we have used for Classification of Diabetic Retinopathy using Deep Learning algorithms . Although the use of Deep Learning and Transfer Learning approach doesn't require feature extraction technique . We have used Feature Extraction on our dataset and pass it on to the Deep Learning Algorithms to see how it performs on the data which shows the features in a highlighted manner .

For further testing ,we even defined our own architecture and tested our model on given dataset and the feature extracted dataset. And then will compare the results of both .

Our custom model Variant - 18 performs well on White Lesion Extracted dataset and give good results compared to other Deep Learning Models.

7.1 Future Work

In future we will test our algorithm on different dataset to see how it performs on the data.

Use multiple datasets: While using a single dataset can give you good results, it is always advisable to use multiple datasets to test the generalizability of your model. Collecting and using different datasets that have different characteristics, such as different image resolutions, different types of diabetic retinopathy images, and different levels of severity, can help you evaluate the performance of your model on unseen data.

Apply data augmentation: Data augmentation techniques, such as rotation, scaling, and flipping, can be used to artificially increase the size of your dataset, thereby reducing overfitting and improving the robustness of your model.

Incorporate more advanced deep learning models: There are various deep learning architectures available that you can explore and incorporate into your work. For instance, you can try using a Generative Adversarial Network (GAN) to generate synthetic images that can be used to augment your dataset. Alternatively, you can also try using a more advanced architecture like a Mask R-CNN or U-Net to improve the segmentation accuracy .

APPENDIX I

WORKING ENVIRONMENT

HARDWARE SPECIFICATION

System: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz

System Type : 64-bit operating system, x64-based processor

RAM : 8 GB

SOFTWARE SPECIFICATION

➢	Operating System	: Windows 11
➢	Tool	: Kaggle Notebook, Colab
➢	Language used	: Python 3.7

APPENDIX II

CODING

```
target_size = (224,224)
batch_size = 32
train_val_split = 0.2
num_classes = 5
nb_epochs =100
#init_epoch = 29
wandb_resume_state = True
exp_name = 'test11'
import shutil
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
from keras.models import save_model
import cv2
from tensorflow.keras.metrics import AUC, Precision, Recall
import wandb
from wandb.keras import WandbCallback

wandb.login(key='017a7b4516d494ef8f402dfcbc3204b00f6cedbc')
train = pd.read_csv('/kaggle/input/aptos2019-blindness-detection/train.csv')
train.head()
if wandb_resume_state:
   #wandb.init(project="Fashion-Classification", resume=True, group=exp_name)
    wandb.init(project="DR_Classification", entity="super-saiyan",resume = True,group = exp_name)
```

```
else:
   exp_name = wandb.util.generate_id()
   myrun = wandb.init(
         project='DR_Classification',
         group=exp_name,
         config={
            'Image Size':224,
            'Num Channels':3,
            'Epoch': nb_epochs,
            'Batch_size':batch_size,
            'Loss':"categorical_crossentropy",
            'Optimizer':'Adam',
         }
   )
config = wandb.config
print(exp_name)
train_val_split = 0.2
#train_data_gen = ImageDataGenerator(rescale=1./255, validation_split=train_val_split)
train_data_gen = ImageDataGenerator(rescale = 1./255,
                           validation_split=train_val_split)
train_generator =
train_data_gen.flow_from_directory( directory='/kaggle/input/customdataset/Feature_Extraction', target_size = (224,224), batch_size = 32, class_mode = 'categorical', subset='training')
validation_generator =
train_data_gen.flow_from_directory( directory='/kaggle/input/customdataset/Feature_Extraction', target_size = (224,224), batch_size = 32,  class_mode = 'categorical', subset='validation')
labels = list(train_generator.class_indices.keys())
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, Activation
if wandb.run.resumed: #if run is to be resumed
   model = keras.models.load_model(wandb.restore("model-best.h5").name)
```

```
else:#else new run
    model = Sequential()
    # Add the first convolutional laye
          model.add(Conv2D(32, kernel_size=(3, 3),strides=(1, 1), padding='same',
        input_shape=(224,224,3)))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

    # Add the second convolutional layer
    model.add(Conv2D(64, kernel_size=(3, 3), strides=(1, 1), padding='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

    # Add the third convolutional layer
    model.add(Conv2D(128, kernel_size=(3, 3), strides=(1, 1), padding='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

    # Add the forth convolutional layer
    model.add(Conv2D(256, kernel_size=(3, 3), strides=(1, 1), padding='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

    # Add the fifth convolutional layer
    model.add(Conv2D(512, kernel_size=(3, 3), strides=(1, 1), padding='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

    # Flatten the output from the convolutional layers
    model.add(Flatten())

    # Add a fully connected layer
    model.add(Dense(1024))
```

```
   model.add(Activation('relu'))
   model.add(Dropout(0.5))

   # Add a final fully connected layer
   model.add(Dense(512))
   model.add(Activation('relu'))
   model.add(Dropout(0.5))

   # Add the output layer with 5 class labels
   model.add(Dense(5))
   model.add(Activation('softmax'))

   # Compile the model
     model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy',
Precision(),  Recall(), AUC()])

#wandb keras compatibility
wandb_call = WandbCallback(save_model=True,
                    save_graph=True,
                    save_weights_only=True,
                    log_weights=True,
                    log_gradients=True,
                    training_data=train_generator,
                    validation_data=validation_generator,
                    validation_steps = validation_generator.samples // batch_size,
                    labels=labels,
                    predictions = 180,
                    input_type='images')
from keras.callbacks import EarlyStopping
early_stopping = EarlyStopping( patience=3, verbose=1)
history = model.fit(train_generator,steps_per_epoch = len(train_generator),
                    validation_data = validation_generator,
                                          validation_steps  =len(  validation_generator),epochs  =
nb_epochs ,callbacks=[early_stopping,wandb_call])
from sklearn.metrics import roc_auc_score, roc_curve
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, mean_squared_error,mean_absolute_error
import numpy as np
# Make predictions for the test set
y_pred = model.predict(validation_generator)
y_pred = np.argmax(y_pred,axis=1)
y_true = validation_generator.classes

# Binarize the true labels and predictions
y_true_bin = label_binarize(y_true, classes=np.unique(y_true))
y_pred_bin = label_binarize(y_pred, classes=np.unique(y_true))

# Compute the AUC for each class
aucs = []
for i in range(y_true_bin.shape[1]):
    fpr, tpr, thresholds = roc_curve(y_true_bin[:, i], y_pred_bin[:, i])
    auc = roc_auc_score(y_true_bin[:, i], y_pred_bin[:, i])
    aucs.append(auc)
    plt.plot(fpr, tpr, label='Class {} ROC curve (AUC = {:.2f})'.format(i, auc))

plt.plot([0, 1], [0, 1], 'k--')  # Random classifier line
plt.xlim([0.0, 0.8])
plt.ylim([0.0, 0.5])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curves for Multiclass Classification')
plt.legend(loc="lower right")
plt.show()

y_true = np.argmax(y_true_bin, axis=1)
```

```
y_pred = np.argmax(y_pred_bin, axis=1)
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import itertools

# Generate predictions for the test set
#y_pred = model.predict(validation_generator)
#y_pred = np.argmax(y_pred,axis=1)
#y_true = validation_generator.classes

# Calculate the confusion matrix
cm = confusion_matrix(y_true, y_pred)

# Define the class labels
class_names = validation_generator.class_indices.keys()

# Plot the confusion matrix
plt.figure(figsize=(8, 8))
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.colorbar()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names, rotation=90)
plt.yticks(tick_marks, class_names)

fmt = 'd'
thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, format(cm[i, j], fmt),
             horizontalalignment="center",
             color="white" if cm[i, j] > thresh else "black")

plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.tight_layout()
plt.show()

#save model
model_json = model.to_json()
with open('model_variant18.json', 'w') as json_file:
   json_file.write(model_json)

model_saved = save_model(model, './weights.hdf5')
#load model
with open('model_variant18.json', 'r') as json_file:
   json_saved_model = json_file.read()

model_loaded = tf.keras.models.model_from_json(json_saved_model)
model_loaded.load_weights('weights.hdf5')
model_loaded.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

#test single image
path = '/kaggle/input/traindata/train/3/0104b032c141.png'
image = cv2.imread(path)
image = cv2.resize(image, (224,224))
image = image/ 255    #normalise
#print(image.shape)
image = image.reshape(-1, 224,224,3)      #reshape in format to send more than one image to predict
#print(image.shape)

result = model_loaded(image)
#print(result)  #probabilities that the image belong to each class
result = np.argmax(result, axis=1)
```

```
if(result==0):
   print('Grade 0')
elif(result == 1):
   print('Grade 1')
elif(result == 2):
   print('Grade 2')
elif(result == 3):
   print('Grade 3')
else:
   print('Grade 4')
#train_data_gen = ImageDataGenerator(rescale=1./255, validation_split=train_val_split)
train_data_gen = ImageDataGenerator(rescale = 1./255,
                                    validation_split=train_val_split)
r_train_generator =
train_data_gen.flow_from_directory( directory='/kaggle/input/redlesion-
customdataset/RedLesion_CustomDataset', target_size = (224,224), batch_size = 32,
class_mode = 'categorical', subset='training')
r_validation_generator =
train_data_gen.flow_from_directory( directory='/kaggle/input/redlesion-
customdataset/RedLesion_CustomDataset', target_size = (224,224), batch_size =
32,  class_mode = 'categorical', subset='validation')
history = model.fit(r_train_generator,steps_per_epoch = len(r_train_generator),
                          validation_data = r_validation_generator,
                                                       validation_steps =len( r_validation_generator),epochs =
nb_epochs ,callbacks=[early_stopping,wandb_call])
from sklearn.metrics import roc_auc_score, roc_curve
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, mean_squared_error,mean_absolute_error
import numpy as np
# Make predictions for the test set
y_pred = model.predict(r_validation_generator)
y_pred = np.argmax(y_pred,axis=1)
```

```
y_true = r_validation_generator.classes

# Binarize the true labels and predictions
y_true_bin = label_binarize(y_true, classes=np.unique(y_true))
y_pred_bin = label_binarize(y_pred, classes=np.unique(y_true))

# Compute the AUC for each class
aucs = []
for i in range(y_true_bin.shape[1]):
    fpr, tpr, thresholds = roc_curve(y_true_bin[:, i], y_pred_bin[:, i])
    auc = roc_auc_score(y_true_bin[:, i], y_pred_bin[:, i])
    aucs.append(auc)
    plt.plot(fpr, tpr, label='Class {} ROC curve (AUC = {:.2f})'.format(i, auc))

plt.plot([0, 1], [0, 1], 'k--')  # Random classifier line
plt.xlim([0.0, 0.8])
plt.ylim([0.0, 0.5])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curves for Multiclass Classification')
plt.legend(loc="lower right")
plt.show()
```

REFERENCES

[1]Francisco J. Martinez-Murcia, Andrés Ortiz, Javier Ramírez , Juan M. Górriz , Ricardo Cruz,“Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy”,Neurocomputing , April 2020.

[2] Tien Yin Wong ,Charumathi Sabanayagam, “Strategies to Tackle the Global Burden of Diabetic Retinopathy: From Epidemiology to Artificial Intelligence”, Ophthalmologica , July 2019.

[3] A. Bilal , G. Sun , S. Mazhar , “Survey on recent developments in automatic detection of diabetic retinopathy”, Volume 44, Issue 3, Pages 420-440, March 2021.

[4] EmanAbdelMaksoud,SherifBarakat,Mohammed Elmogy, “Diabetic Retinopathy Grading System Based on Transfer Learning”,Dec 2020.

[5] Borys Tymchenko , Philip Marchenko and
Dmitry Spodarets, “Deep Learning Approach to Diabetic Retinopathy Detection” ,March 2020.

[6] Misgina Tsighe Hagos , Shri Kant , “Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset”

[7] Yijin Huanga , Li Lina , Pujin Chenga , Junyan Lyua , Xiaoying Tanga, “Identifying the key components in ResNet-50 for diabetic retinopathy
grading from fundus images: a systematic investigation”,Oct 2021.

[8] “Deep Learning based Early Detection and Grading of Diabetic Retinopathy Using Retinal Fundus Images”,Sheikh Muhammad Saiful Islam, Md Mahedi Hasan2 and Sohaib Abdullah -arXiv - 2018.

[9] “Modified Alexnet architecture for classification ofdiabeticretinopathyimages”T.ShanthiR.S.Sabeenian,Science Direct , Computer and Electrical Engineering.-2019.

[10] “Diabetic Retinopathy Grading System Based on Transfer Learning” ,Eman AbdelMaksoud, Sherif Barakat, Mohammed Elmogy- arXiv- 2020

Document Information

Analyzed document	SelvaReport-V3.doc (D163825675)
Submitted	2023-04-13 11:58:00
Submitted by	karkuzhali
Submitter email	karkuzhali@mepcoeng.ac.in
Similarity	6%
Analysis address	karkuzhali.mepco@analysis.urkund.com

Sources included in the report

SA	**Mepco Schlenk Engineering College / Full manuscript_v5.docx** Document Full manuscript_v5.docx (D130978729) Submitted by: priyanka@mepcoeng.ac.in Receiver: priyanka.mepco@analysis.urkund.com	⊞	3
SA	**1636060004_R.pdf** Document 1636060004_R.pdf (D154580959)	⊞	1
SA	**18134691541-TS.pdf** Document 18134691541-TS.pdf (D130735864)	⊞	2
W	URL: https://arxiv.org/abs/2108.08473 Fetched: 2021-11-30 11:53:59	⊞	2
SA	**20 - edit_Research Article_Bivek.docx** Document 20 - edit_Research Article_Bivek.docx (D140036672)	⊞	3
W	URL: https://www.mdpi.com/1424-8220/21/11/3704/pdf Fetched: 2021-06-30 15:49:35	⊞	2
W	URL: https://www.researchgate.net/publication/339991922_Transfer_Learning_with_Convolutional_Neural... Fetched: 2023-01-07 11:55:41	⊞	1
SA	**Manuscript.pdf** Document Manuscript.pdf (D79931818)	⊞	4
W	URL: https://www.researchgate.net/publication/336284134_Automated_Identification_and_Grading_System... Fetched: 2020-01-31 10:23:14	⊞	2
W	URL: https://www.sciencedirect.com/science/article/abs/pii/S1746809420302676 Fetched: 2021-08-02 04:58:58	⊞	1
SA	**DR FULL THESIS.pdf** Document DR FULL THESIS.pdf (D125931614)	⊞	1
W	URL: https://www.researchgate.net/publication/322978904_Classification_of_Diabetic_Retinopathy_Imag... Fetched: 2019-09-26 08:11:24	⊞	2
SA	**SmartTech-IC_2021_paper_94.pdf** Document SmartTech-IC_2021_paper_94.pdf (D117049722)	⊞	1