DIABETIC RETINOPATHY DETECTION USING ENSEMBLE TRANSFER DEEP LEARNING

2023
MASTER THESIS
COMPUTER ENGINEERING

Shuhad Imad Hadi ALDUJAILI

Thesis Advisor
Assist. Prof. Dr. Nehad T.A. RAMAHA

DIABETIC RETINOPATHY DETECTION USING ENSEMBLE TRANSFER DEEP LEARNING

Shuhad Imad Hadi ALDUJAILI

Thesis Advisor

Assist. Prof. Dr. Nehad T.A. RAMAHA

T.C.

Karabuk University

Institute of Graduate Programs

Department of Computer Engineering

Prepared as

Master Thesis

KARABUK

May 2023

I certify that in my opinion the thesis submitted by Shuhad Imad Hadi ALDUJAILI titled "DIABETIC RETINOPATHY DETECTION USING ENSEMBLE TRANSFER DEEP LEARNING" is fully adequate in scope and quality as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Nehad T.A. RAMAHA .....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. June 15, 2023

Examining Committee Members (Institutions)	Signature
Chairman: Assoc. Prof. Dr. Adib HABBAL (KBU)	.....
Member: Assist. Prof. Dr. Nehad T.A RAMAHA (KBU)	.....
Member: Assist. Prof. Dr. Ali HAMİTOĞLU (İU)	.....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Prof. Dr. Müslüm KUZU .....
Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Shuhad Imad Hadi ALDUJAILI

ABSTRACT

M. Sc. Thesis

DIABETIC RETINOPATHY DETECTION USING ENSEMBLE TRANSFER DEEP LEARNING

Shuhad Imad Hadi ALDUJAILI

Karabuk University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Nehad T.A. RAMAHA
May 2023, 86 pages

Diabetic retinopathy is one of the eye diseases that is a complication of diabetics and can affect vision and even lead to blindness. This disease affects the blood vessels in the retina, which is one of the main marks that help to detect this disease. Diabetic retinopathy detection is a challenging process requiring specialists and too much time to process each image. However, computer science algorithms, including machine learning and deep learning, can help physicians and specialists detect diabetic retinopathy effectively. In this study, a novel diabetic retinopathy approach is introduced. The approach is based on a well-known Kaggle image dataset containing images of four stages (mild, severe, moderate, and proliferate) besides the normal condition.The images are preprocessed (resizing, normalization) and over-sampled (balanced) to get all categories with similar percentages. The balancing is essential so the trained model will treat all categories with similar weights. After that, the data augmentation process is applied to increase the number of training images and supply

the training process with different conditions of the same images. The dataset is split into training and testing subsets. The training process includes two different scenarios; the first is based on the unbalanced version of the dataset, while the second is done using the balanced dataset. In the first and second scenarios, many deep learning models are used as base models for the entire deep model. The classification part of the entire deep models consists of flatten, dropout, and dense layers. The outputs layer uses the softmax function, and the training process is applied using the categorical cross-entropy loss function. All scenarios use the Adam optimizer and 50 epochs with an early stop condition. The DL models used include VGG-16, VGG-19, Inception, Xception, EfficientNet, and NasNetLarge. The main contribution of the current study is using the ensemble learning. The study suggests building an ensemble of the trained models in order to minimize the categories classification errors and improves the performance. Besides those scenarios and for comparative aims, another training scenario is proposed. The stages of diabetic retinopathy are grouped into one category named DR, so the categories became DR and NO_DR. As a result, an ensemble of the VGG, EfficientNet, and Xception is built. All models are evaluated using the performance evaluation metrics (accuracy, precision, recall, and F1-score). Results indicate that the ensemble model achieves the best performance against all individual models with 92% accuracy for the balanced multi-class scenario. The accuracy is enhanced to 99.46% in the case of using the binary class classification scenario (DR and NO_DR). A detailed comparison between the current study and related ones is performed. The comparison proved that the current study either outperformed the previous studies' performance or used more challenging options.

Key Words : Diabetic Retinopathy, Blood Vessels, Machine Learning, Deep Learning, Transfer Learning and Image Classification.

Science Code : 92432

ÖZET

Yüksek Lisans Tezi

TOPLULUK TRANSFERİ DERİN ÖĞRENME YÖNTEMİ KULLANARAK DİYABETİK RETİNOPATİ TESPİTİ

Shuhad Imad Hadi ALDUJAILI

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Nehad T.A. RAMAHA
Mayıs 2023, 86 sayfa

Diyabetik retinopati, şeker hastalarının komplikasyonu olan ve görmeyi etkileyip körlüğe yol açabilen bir göz hastalığıdır. Bu hastalık, bu hastalığın tespit edilmesine yardımcı olan ana işaretlerden biri olan retinadaki kan damarlarını etkiler. Diyabetik retinopati tespiti, her bir görüntü üzerinde çalışmak için bu işin uzmanlarına, çok fazla zaman ve uzman gerektiren zorlu bir süreçtir. Ancak, makine öğrenimi ve derin öğrenme gibi bilgisayar bilimi algoritmaları, doktorların ve uzmanların diyabetik retinopatiyi etkili bir şekilde tespit etmelerine yardımcı olabilir. Bu çalışmada, yeni bir diyabetik retinopati yaklaşımı tanıtılmaktadır. Bu yaklaşım, normal durumun yanı sıra dört evreyi (hafif, şiddetli, orta ve proliferatif) görüntüleri içeren bilinen bir Kaggle görüntü veri kümesine dayanmaktadır. Görüntüler, ön işleme (yeniden boyutlandırma, normalleştirme) ve örnekleme (dengeleme) ile tüm kategorilerin benzer

yüzdeleri ele alınır. Dengeleme, eğitimli modelin tüm kategorilerini benzer ağırlıklarla ele alması için önemlidir. Daha sonra eğitim görüntülerinin sayısını artırmak ve aynı görüntülerin farklı durumlarıyla eğitim sürecini sağlamak için veri artırma işlemi uygulanır. Veri kümesi eğitim ve test alt kümelerine ayrılmıştır.Eğitim süreci iki farklı senaryoyu içermektedir; ilki veri setinin dengesiz versiyonuna dayanırken, ikincisi dengeli veri seti kullanılarak gerçekleştirilir. İlk ve ikinci senaryolarda, birçok derin öğrenme modeli tüm derin model için temel modeller olarak kullanılır. Tüm derin modellerin sınıflandırma kısmı, düzleştirme, bırakma ve yoğun katmanlardan oluşur. Çıktı katmanı, softmax işlevini kullanmaktadır ve eğitim süreci, kategorik çapraz entropi kayıp işlevini kullanarak uygulanır. Tüm senaryolar, Adam optimizasyon algoritmasını ve erken durdurma koşulu ile 50 dönem kullanır. Kullanılan DL modelleri arasında VGG-16, VGG-19, Inception, Xception, EfficientNet ve NasNetLarge bulunmaktadır. Mevcut çalışmanın ana katkısı, topluluk öğrenmenin kullanılmasıdır. Bu Çalışma, kategorilerin sınıflandırma hatalarını en aza indirgemek ve performansı korumak için eğitilmiş modellerin topluluğunu oluşturmayı önermektedir. Bu senaryoların yanı sıra ve karşılaştırmalı amaçlar için, başka bir eğitim senaryosu önerilmiştir. Diyabetik retinopatinin aşamaları, DR olarak adlandırılan tek bir kategoriye gruplandırılmıştır. böylece kategoriler DR ve NO_DR olarak değişmiştir. Sonuç olarak, VGG, EfficientNet ve Xception'ın bir topluluğu oluşturulmuştur. Tüm modeller, performans değerlendirme metrikleri (doğruluk, hassasiyet, hatırlama ve F1 skoru) kullanılarak değerlendirilmiştir. Sonuçlar, toplu modelin, dengelenmiş çok sınıflı senaryo için %92 doğruluk ile tüm bireysel modellere karşı en iyi performansı gösterdiğini göstermektedir. Doğruluk, ikili sınıf sınıflandırma senaryosu (DR ve NO_DR) kullanıldığında %99.46'ya yükselmiştir. Mevcut çalışma ile ilgili karşılaştırmalar arasında ayrıntılı bir karşılaştırma yapılmıştır. Karşılaştırma, mevcut çalışmanın ya bir önceki çalışmaların performansını aştığını ya da daha zorlu seçenekler kullandığını kanıtlamıştır.

Anahtar Kelimeler **:** Anahtar Kelimeler: Diyabetik Retinopati, Kan Damarları, Makine Öğrenmesi, Derin Öğrenme, Transfer Öğrenme ve Görüntü Sınıflandırma.

Bilim Kodu **:** 92432

ACKNOWLEDGMENT

First, thanks to Allah the Almighty for his divine guidance throughout this academic journey. I am also grateful to adviser Asst. Prof. Dr. Nehad T.A. RAMAHA, for his support and guidance in achieving this thesis. My sincere thanks extend to the Karabuk University members, who played a crucial role in my academic courses. I am also grateful to my parents for their support.

Finally, with the utmost respect, I dedicate this thesis to Iraq, my beloved homeland, and Turkey, which has graciously hosted our academic endeavours.

CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

ABBREVIATIONS

1D : 1 Dimensional

2D : 2 Dimensional

AI : Artificial Intelligent

CNN : Convolution Neural Network

VGG : Visual Geometry Group

DL : Deep Learning

EL : Ensemble Learning

ML : Machine Learning

PART 1

INTRODUCTION

1.1. OVERVIEW

Diabetic retinopathy, a widespread condition impacting millions globally, is a complication stemming from diabetes and can impair vision. Medical eye examinations enable doctors to identify this disease. However, numerous images must be analyzed to reach a conclusion. Fortunately, computer-assisted decision support systems can assist physicians in making accurate determinations with minimal effort and time. This study presents a review of existing diabetic retinopathy computer-assisted studies. Diabetes impacts not only the retina but also various other tissues, such as the heart and kidneys [1] [2] [3]. The International Diabetes Federation [4] reports that over 537 million individuals worldwide are affected by diabetes, with 90 million of these patients experiencing diabetic retinopathy. Diabetic retinopathy (DR), a diabetes-related complication, damages the retina through blood vessel swelling and fluid leakage within the eye. This complication impairs vision and can lead to blindness, with approximately 2.6% of blindness cases resulting from retinopathy [5] [6].

Diabetic retinopathy is a condition where diabetes adversely affects the retina, leading to vision issues and potentially blindness [7]. Early detection of this disease helps patients regain their retina's normal function and prevents blindness [8]. Traditional manual methods for identifying diabetic retinopathy, however, are time-consuming due to the vast amount of data involved and often result in misclassifications [9]. In contrast, computer-assisted decision support tools can accurately detect diabetic retinopathy [10]. Around 75% of diabetic retinopathy cases are found in underprivileged countries [11] that lack adequate equipment and detection resources. Consequently, decision-support systems for detecting diabetic retinopathy play a crucial role in early diagnosis. The presence of lesions in retinal images is used to

identify diabetic retinopathy, with lesions including hemorrhages (HM), micro-aneurysms (MA), and soft and hard exudates (EX) [12].

Micro-aneurysm lesions, the earliest indication of diabetic retinopathy, appear as red circular points resulting from weakened blood vessel walls. These points have a size of less than 125 μm and sharp borders. Arrigo et al. [13] identify six primary types of micro-aneurysms: saccular, focal bulge, fusiform, mixed, pedunculated, and irregular, as depicted in Figure 1.1 [14].

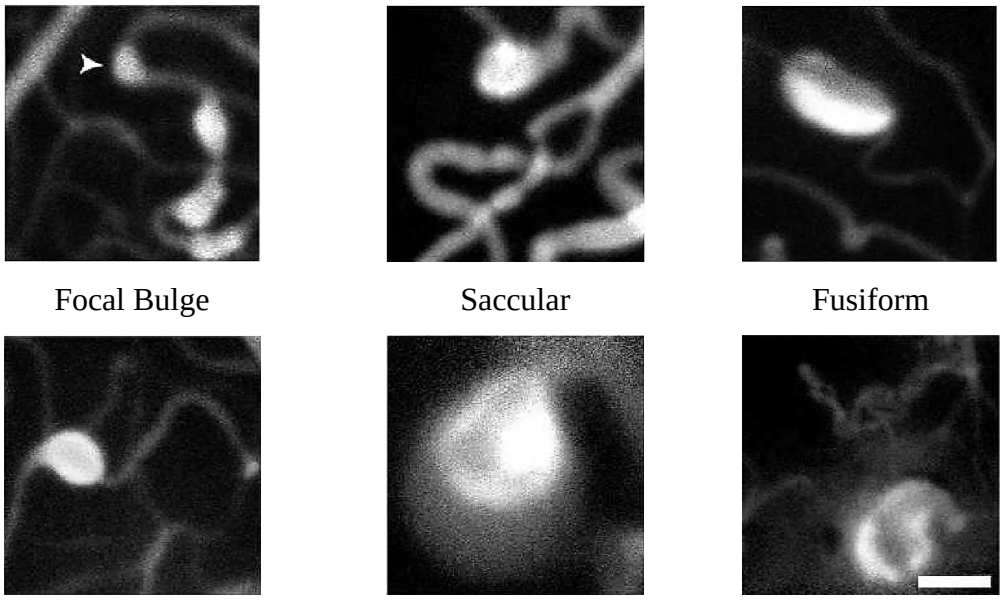


Figure 1.1. Different types of MA diabetic retinopathy.

The second form of diabetic retinopathy, Hemorrhages (HM), manifests as large spots on the retinal tissue, exceeding 125 μm in size and exhibiting irregular edges, as depicted in Figure 1.2 [15].

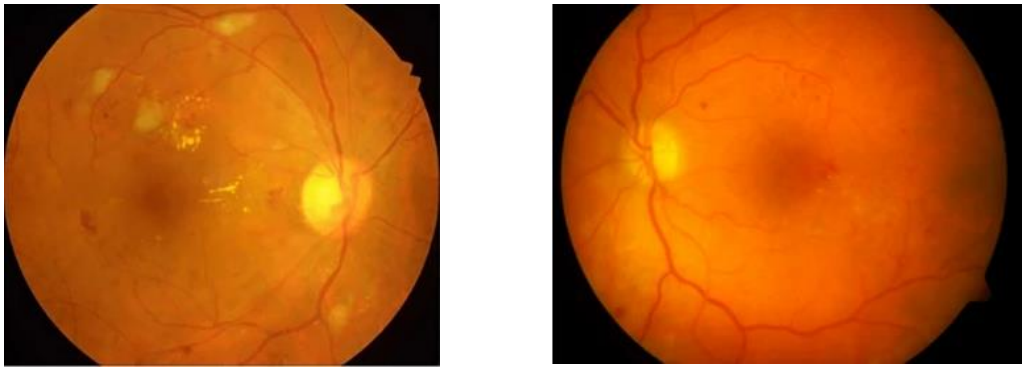


Figure 1.2. Diabetic retinopathy with Hemorrhages (HM)

Hard exudates present as bright-yellow spots with distinct borders on the retinal tissue, resulting from plasma leakage. This diabetic retinopathy variety typically occurs in the retina's outer layers. On the other hand, soft exudates are white spots that arise from nerve fiber swelling. This kind of diabetic retinopathy usually has an oval shape. Figure 1.3 displays Hard and Soft exudates in diabetic retinopathy [16] [17].

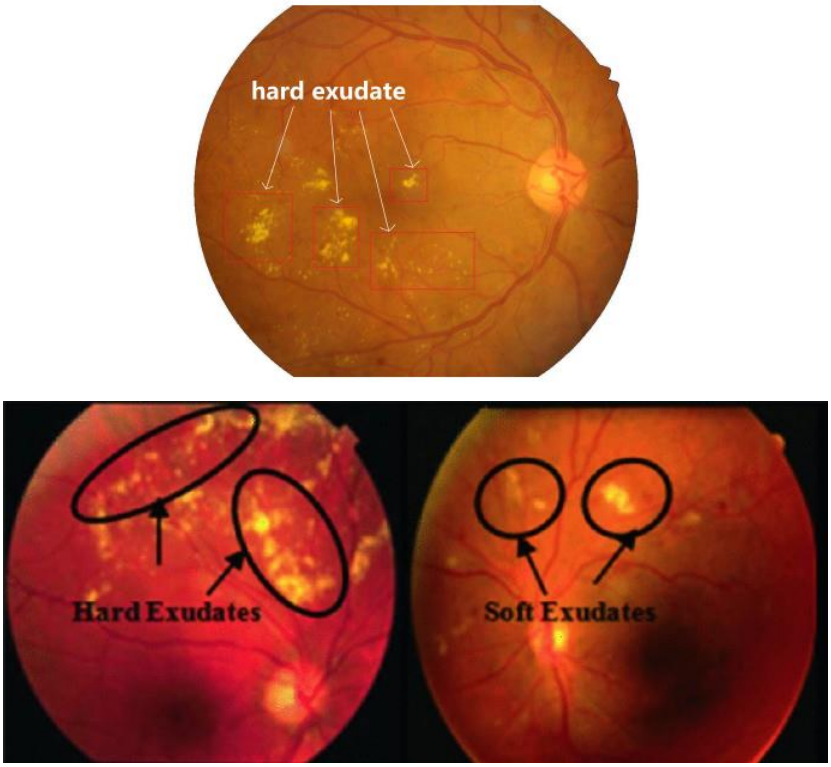


Figure 1.3. Diabetic retinopathy with Soft and Hard exudates

The four primary categories of diabetic retinopathy are demonstrated in Figure 1.4, using an example from the IDRiD dataset [18].

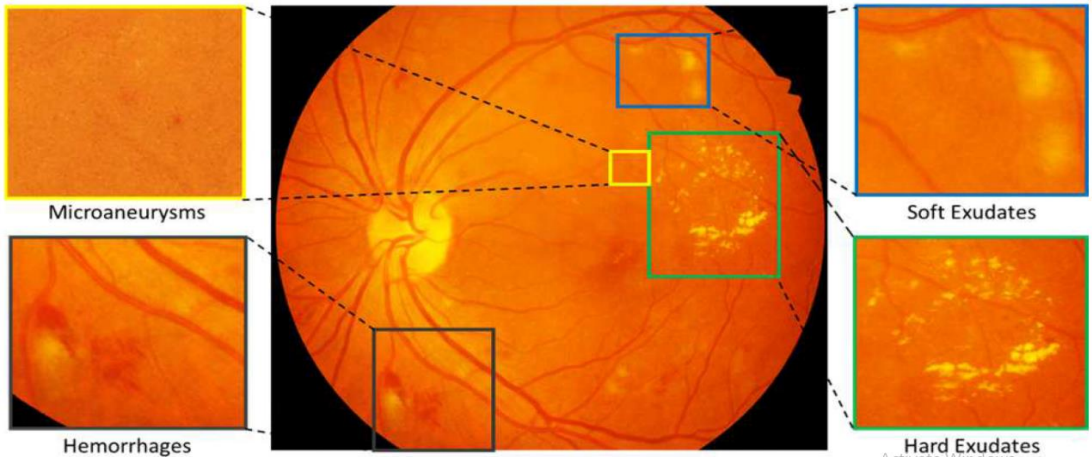


Figure 1.4. Principal classifications of diabetic retinopathy.

1.2. PROBLEM STATEMENT

Detecting diabetic retinopathy is a complex and time-consuming process that requires the expertise of specialists. Although machine learning and deep learning algorithms have demonstrated their effectiveness in detecting diabetic retinopathy, This issue is an important problem in the medical domain since processing a large number of retinal images to make an accurate decision requires time and effort. The current state-of-the-art needs to improve its accuracy and efficiency. Existing approaches deal with diabetic retinopathy as one level of disease, which is not applicable in the medical domain since this disease contains many levels. Therefore, current research problem is to develop a more accurate and efficient system for detecting diabetic retinopathy that can handle the multi-class classification problem and improve the performance of existing deep learning models.

In this study, a novel system for detecting diabetic retinopathy that addresses the limitations of existing approaches is proposed. Our approach uses transfer learning techniques to improve the performance of deep learning models. Moreover, it explores the potential benefits of combining multiple deep learning models into an ensemble system to achieve higher accuracy and robustness in the classification task.

The main contribution of our study is the use of ensemble learning, where we build an ensemble of the trained models to minimize the categories classification errors and improve the performance.

1.3. GOAL AND OBJECTIVES

The main goal of the current study is to create a diabetic retinopathy detection system based on retinal images and deep learning. Thus, the research objectives are:

- To explore the issue of detecting diabetic retinopathy, including identifying the best retinal images and deep learning methods to be used and selecting the most suitable one.
- To enhance diabetic retinopathy detection using the most effective deep models in accuracy and time.
- To improve the performance of retinopathy detection by using the ensemble transfer learning of the best deep-trained models.

1.4. MOTIVATION

Deep learning systems have great potential for detecting and diagnosing DR from retinal images. Here are five important reasons why DR-based deep learning systems are crucial:

- Early detection: Deep learning systems can detect DR before symptoms become apparent. This allows for timely intervention and treatment, which can help prevent blindness and other serious complications.
- Accurate diagnosis: Deep learning systems can provide a more accurate diagnosis of DR than human experts. This is because they can analyze large amounts of data and identify subtle changes in the retinal images that may be missed by the human eye.
- Reduced workload for healthcare professionals: By automating the detection and diagnosis of DR, deep learning systems can reduce the workload of healthcare professionals, allowing them to focus on other essential tasks.

- Cost-effective: Deep learning systems can be cost-effective compared to traditional methods of DR screening. This is particularly important in developing countries where resources are limited.
- Improved patient outcomes: By detecting and treating DR early, deep learning systems can improve patient outcomes and quality of life. This is particularly important for people with diabetes at high risk of developing DR.

1.5. CONTRIBUTION

The contribution of the current state of the art in the field of diabetic retinopathy diagnosis comes from developing and evaluating an ensemble transfer deep learning approach on a multi-class dataset. Specifically, we demonstrate the effectiveness of using an ensemble of pre-trained deep learning models, combined with transfer learning techniques, to achieve high accuracy in classifying retinal images into multiple stages of diabetic retinopathy. Our findings highlight the potential of ensemble transfer deep learning as a powerful tool for improving the accuracy and efficiency of diabetic retinopathy diagnosis, and it can be used to develop more effective screening and treatment strategies for this debilitating condition.

1.6. ORGANIZATION OF THESIS

The rest of the thesis will be organized as follows:

Chapter two will contain the related work and previous studies comparison. Chapter three will introduce the proposed methodologies. Besides that, the used materials, including dataset and software will also be listed. The implementation and experimental results along with the discussion will be included in chapter four. The conclusion and future work will be organized in the final chapter (chapter five).

PART 2

RELATED WORK

2.1. INTRODUCTION

In this chapter, the literature review will be introduced. A retailed comparative study of the most recent studies in the field of diabetic retinopathy will be introduced. The studies will be analyzed and compared in terms of used methodologies, datasets, results and limitations.

2.2. ML AND DL-BASED MODELS

Numerous machine learning methodologies have been proposed for the detection of diabetic retinopathy. Various ML and DL models have been trained and assessed using retinal datasets. However, the primary issue with the ML systems is the limited accuracy resulting from the resemblance between diabetic retinopathy ailments and the shape of the retina image, which exhibits brightness at the center and darkness at the borders. Additionally, the systems are affected by factors such as illumination variations, low contrast, small lesions [19], and insignificant parts within the retinal images that are not actual lesions. The ML-based diabetic retinopathy system's key steps are depicted in Figure 2.1.

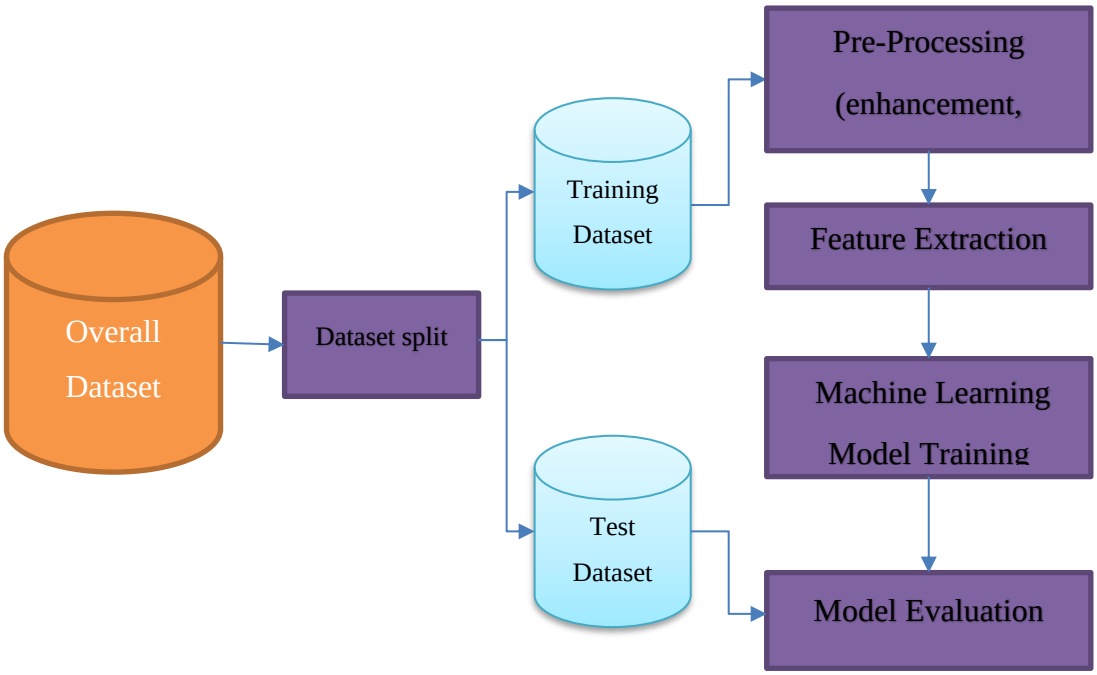


Figure 2.1. ML-based diabetic retinopathy detection system.

While Figure 2.2 shows the general architecture of DL-based diabetic retinopathy systems.

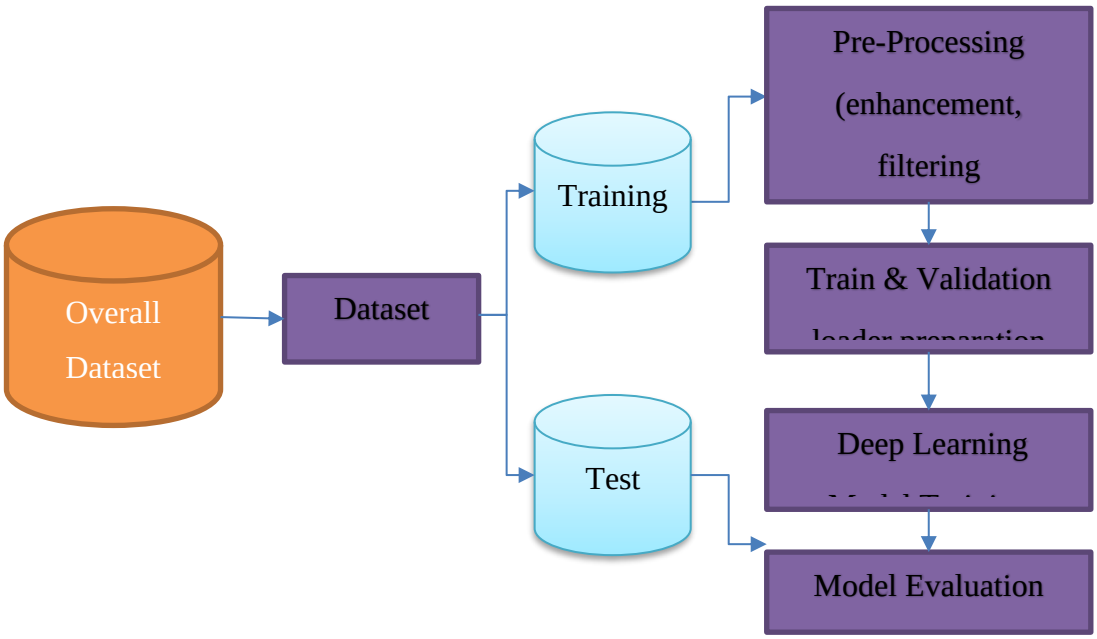


Figure 2.2. DL-based diabetic retinopathy detection system.

The main difference between ML and DL models is that DL models can accept image as input but the ML models need an extra operation (feature extraction) in order to

transform 2D image into 1D feature vector. There are other differences like the data augmentation process that allow models to generate new samples of the same dataset samples but with different shape (resized, cropped, flipped, enhanced, versions of the same sample) allowing network to recognize different shape of the same image.

2.3. RELATED WORK

In this section, the most recent ML and DL diabetic retinopathy systems will be listed and concluded in order to make a good literature review about the most recent systems.

2.3.1. ML Related Work

Various ML methods have been employed in developing diabetic retinopathy systems, such as support vector machines (SVM) [20] [21] [22], Decision Trees (DT), Naïve Bayes (NB), Neural Networks (NN), logistic regression (LR), XGBoost model, K-nearest neighbor (K-NN), and more.

2.3.1.1. SVM-Based DR Models

Bhargavi et al. [20] designed an SVM-based diabetic retinopathy system. They initially segmented retinal images to obtain blood vessels using Bilateral filtering and Hessian matrix transform, followed by extracting foreground bright lesions. Statistical and geometrical features were obtained from the segmented images (20 features in total), and the SVM classifier was trained using these features. The proposed method was applied to the DIARETDB1 dataset (89 images) and MESSIDOR (1200 images), achieving 96.66% accuracy. They only used one type of DR disease in their study. Enrique et al. [21] constructed an SVM-based diabetic retinopathy detection system using their dataset of 400 retinal images. They first isolated blood vessels, hard exudates, and microaneurysms, then extracted features from the original, red, and green components of the segmented images. The SVM classifier was utilized for classification, resulting in 92.4% accuracy. The study detected diabetic retinopathy without classifying main types and used a small dataset. Chetoui et al. [19] developed a diabetic retinopathy detection system utilizing textual features and an SVM model. They extracted Local Ternary Pattern (LTP) and Local Energy-based Shape Histogram

(LESH) from segmented retinal images, which were used to train an SVM classifier. The study found that LESH features were the best, achieving 90.4% accuracy and 0.93 Area Under Curve (AUC). They used 1200 MESSIDOR dataset images and only differentiated between normal and abnormal conditions, without further retinopathy categorization. Hardes et al. [22] presented an SVM-based retinal fundus detection system. They employed the Gaussian mixture model, K-means algorithm, Principle Component Analysis (PCA), Grey-level co-occurrence matrix (GLCM), and SVM, achieving 77.3% accuracy on the DIARETDB1 dataset. Their approach did not modify the proposed ML models, resulting in low accuracy.

2.3.1.2. Decision Trees

Aziza et al. [23] recommended a decision tree classifier for diabetic retinopathy detection, using color fundus DRIVE and Messidor datasets. They first segmented retinal images to obtain blood vessels, then extracted geometric features. Hessian matrix and active contouring algorithms were used for blood vessel segmentation. They classified images into DR or No-DR categories, achieving 93% classification accuracy. Yao et al. [24] proposed detecting early-stage diabetic retinopathy using decision tree models, including two patient categories totaling 241 patients. The model was evaluated using the area under the curve (AUC), sensitivity, and specificity, yielding results of 0.62, 66%, and 76%, respectively. Random Forests Casanova et al. [25] introduced a Random Forests (RF)-based diabetic retinopathy detection system, using 3443 eye-study images. Their approach achieved 90% accuracy without using segmentation or feature extraction methods.

Alzami et al. [26] employed fractal analysis and random forests classifier for diabetic retinopathy detection and classification, using the MESSIDOR dataset. They segmented the green component of retinal images and used morphological Skeltonization to obtain vessels. Connected components and closing morphological operations were also used for the final fundus image. In the feature extraction step, fractal characteristics were utilized. The classification step was performed using the RF algorithm, resulting in an accuracy of 80.37%. Their approach differentiated between healthy individuals and diabetic retinopathy patients but failed to classify the severity of diabetic retinopathy. Zaaboub and Douik [27] proposed a hard exudate

diabetic retinopathy detection system using a dataset of color fundus retinal images. They removed the optic disk and extracted specific parameters from the binary mask of the exudate region. These features were then introduced to the RF classifier, which was trained and evaluated, achieving 94.38% accuracy. Naïve Bayes Kang et al. [28] suggested a Naïve Bayes classifier in their study, using statistical feature extraction such as gray-level co-occurrence matrix, gray-level run-length texture analysis, and statistical texture features. These features were used to train the Naïve Bayes classifier, which classified fundus images of diabetic retinopathy from the China diabetic dataset (568 images) with an accuracy of 93.44%.

Hadistio et al. [29] introduced a diabetic retinopathy detection system using the UCI machine learning diabetic retinopathy dataset (1151 data records and 19 attributes). Stochastic Gradient Descent (SGD) and Naïve Bayes algorithms were employed to classify normal and diabetic retinopathy samples, obtaining an accuracy of 56.74%.

2.3.1.3. Mixed Models and Ensemble Models

In some studies, researchers used multiple ML models and compared their performance.

Roychowdhury et al. [30] presented a computer-aided system for detecting diabetic retinopathy using ML algorithms, including Gaussian mixture model (GMM), AdaBoost, K-NN, and SVM. Their study minimized features using AdaBoost feature ranking to only 30 features. They proposed a two-step hierarchical classification method, where the first step rejected non-lesion parts of retinal images, and the second step classified lesions into four main types: hard exudates, cotton spots, hemorrhages, and micro-aneurysms. The experiments were applied to 1200 MESSIDOR dataset images, achieving 100% sensitivity, 53.16% specificity, and 0.9 AUC. The main issue with their research was the high false positives.

Reddy et al. [31] proposed an ensemble model for diabetic retinopathy detection, combining RF, DT, AdaBoost, K-NN, and Logistic Regression (LR). They first applied normalization to the dataset and then trained the ensemble model. The best model achieved 78% precision, recall, and F1-score, and 77% accuracy. The study

claimed the ensemble model increased performance by nearly 80%. Their research applied to a textual dataset and achieved low accuracy due to a lack of preprocessing operations. The study also employed binary classification (DR or Not DR classes).

Sidker et al. [32] suggested a new ensemble model for diabetic retinopathy based on gray-level intensity, texture feature extraction, and decision trees, using the Asia Pacific Tele-Ophthalmology Society 2019 dataset. Their proposed approach included preprocessing, textual feature extraction, feature selection, and ensemble learner training. The results showed 94.2% accuracy and an F-measure of 93.51%.Another ensemble learning-based diabetic retinopathy detection system was introduced by [33], focusing on microaneurysms eye disease. The ensemble included four classifiers: SVM, K-NN, DT, and Naïve Bayes. First, images were pre-processed, and shape and intensity features were extracted from the pre-processed images. Experiments were applied to the E- ophtha and DIARETDB1 datasets, obtaining AUC scores of 0.928 and 0.873 for the respective datasets.

In summary, various machine learning approaches have been employed to create diabetic retinopathy detection systems. These approaches include support vector machines, decision trees, naïve bayes, neural networks, logistic regression, XGBoost, K-nearest neighbor, and ensemble models. These studies demonstrate the potential of machine learning models for effective diabetic retinopathy detection and classification, with some achieving high accuracy rates. However, certain challenges still need to be addressed, such as reducing false positives and improving the classification of diabetic retinopathy severity. Continued research and development in this area will undoubtedly contribute to the improvement of diabetic retinopathy detection systems, providing better support for clinicians and enhancing patient outcomes.

Table 2.1 includes a detailed comparison of the previous ML-based diabetic retinopathy studies.

Table 2.1. ML-based diabetic retinopathy related work.

Researcher	Year	Methodology	Dataset	Main Results	Limitations
Bhargavi et al. [20]	2016	SVM	DIARETDB1, MESSIDOR	96.66% Accuracy	Only one type of DR disease
Enrique et al. [21]	2017	SVM	400 retinal images	92.4% Accuracy	Small dataset, no DR type classification
Chetoui et al. [19]	2018	SVM (Textual features)	MESSIDOR	90.4% Accuracy, 0.93 AUC	No classification of other retinopathy types
Hardes et al. [22]	2022	SVM	DIARETDB1	77.3% Accuracy	No ML model modifications, low accuracy
Aziza et al. [23]	2019	Decision Trees	DRIVE, Messidor	93% Accuracy	Binary classification (DR or Not DR)
Yao et al. [24]	2022	Decision Trees	241 patients	0.62 AUC, 66% Sensitivity, 76% Specificity	Low dataset size
Casanova et al. [25]	2014	Random Forests	3443 eye-study images	90% Accuracy	No segmentation or feature extraction
Alzami et al. [26]	2019	Random Forests (Fractal analysis)	MESSIDOR	80.37% Accuracy	No severity classification of DR patients
Zaaboub and Douik [27]	2020	Random Forests	Color fundus retinal images	94.38% Accuracy	Detected only one type of diabetic retinopathy
Kang et al. [28]	2020	Naïve Bayes	China diabetic dataset (568)	93.44% Accuracy	Low accuracy, They used three categories for classification
Hadistio et al. [29]	2022	Naïve Bayes, SGD	UCI ML diabetic retinopathy	56.74% Accuracy	Low accuracy
Roychowdhury et al. [30]	2014	GMM, AdaBoost, K-NN, SVM	MESSIDOR	100% Sensitivity, 53.16% Specificity, 0.9 AUC	High false positives
Reddy et al. [31]	2020	Ensemble (RF, DT, AdaBoost, K-NN, LR)	Textual dataset	78% Precision, Recall, F1-score, 77% Accuracy	Low accuracy, no preprocessing, binary classification
Sidker et al. [32]	2021	Ensemble (Gray-level	Asia Pacific Tele-	94.2% Accuracy,	

		intensity, texture, DT)	Ophthalmology Society 2019	93.51% F-measure	
Pendekal et al. [33]	2022	Ensemble (SVM, K-NN, DT, Naïve Bayes)	E-ophtha, DIARETDB1	0.928 and 0.873 AUC	The study detected only one type of diseases (Microaneurysms)
Sopharak et al. [34]	2008	SVM, K-NN	DIARETDB1	89.2% Sensitivity, 75.0% Specificity	Small dataset, binary classification (DR or No-DR)
Ganesan et al. [35]	2014	SVM, Random Forests, Decision Trees, Naïve Bayes	IDRiD	94.0% Accuracy	Limited DR severity classification
Antal et al. [36]	2014	Ensemble of Decision Trees	Messidor, DIARETDB0, DIARETDB1	95.6% Accuracy, 0.93 AUC	Lacks preprocessing optimization, limited dataset variety

2.3.2. DL related Work

Deep learning (DL) is a subfield of machine learning that utilizes deep neural networks. The process for detecting diabetic retinopathy using DL is similar to machine learning, with minor differences. Figure 6 illustrates the DL process for a diabetic retinopathy detection system.

Pratt et al. [37] employed a Convolutional Neural Network (CNN) to extract retinal image features from the Kaggle diabetic retinopathy dataset (80,000 images). They applied color normalization, data augmentation, and L2-regularization during preprocessing. Stochastic Gradient Descent optimization was used in the training step with CNN, resulting in 95% specificity, 75% accuracy, and 30% sensitivity. The results showed a high number of false negatives. Soniya et al. [38] developed single-based and heterogeneous-based CNN systems for diabetic retinopathy detection. They utilized gradient descent and backpropagation algorithms for training. The study classified four primary lesion types: microaneurysms (MAs), hemorrhages (HEs), hard exudates (EXs), and soft EXs. They used 130 images from the DIARETDB0 dataset and achieved varying accuracies depending on the CNN architecture (95%, 65%, 42.5%, 67.5%, and 92.5%). Gargeya and Leng [39] presented an automated diabetic

retinopathy identification system using deep learning models. Their study used MESSIDOR2 and E-Ophtha datasets, and the designed system achieved 0.94 and 0.95 AUC for MESSIDOR2 and E-Ophtha databases, respectively. The sensitivity and specificity values were 93% and 87% for the MESSIDOR2 dataset, while they were 90% and 94% for the E-Ophtha dataset.

Lam et al. [40] proposed an automated diabetic retinopathy detection system using the Kaggle EyePACS dataset, which contained 243 retinal images. They employed several CNN architectures with resized retinal images (128x128x3), including GoogleLeNet-v1, AlexNet, VGG-16, ResNet, and Inception-V3. The InceptionV3 model achieved the highest accuracy of 98%. Khalifa et al. [41] suggested using various DL models, such as AlexNet, ResNet18, SqueezeNet, GoogleNet, VGG16, and VGG19, with the Asia Pacific Tele-Ophthalmology Society (APTOS) 2019 dataset. The best accuracy was obtained by the AlexNet model with 97.9%, and the total average accuracy was 96.3%. They did not use ensemble or fusion approaches.

Nguyen et al. [42] applied transfer learning of VGG16 and VGG19 models for diabetic retinopathy detection, using the Kaggle competition dataset 2015, which included severe, mild, moderate, proliferative DR, and normal cases. The study employed data augmentation and achieved accuracies of 71% and 73% for VGG16 and VGG19, respectively. After modifying with sequential dense layers, the performance improved to 83%. Tymchenko et al. [43] utilized a three-head CNN model to detect diabetic retinopathy stages in retinal images. They proposed a multi-stage approach based on transfer deep learning, enabling the use of similar datasets with different labels. The retinal images were resized, and data augmentation was applied. They achieved a sensitivity of 99% on the APTOS 2019 Blindness Detection Dataset.

Pour et al. [44] proposed the EfficientNet B5 deep model for diabetic retinopathy detection, using three datasets: MESSIDOR, MESSIDOR-2, and IDRiD. Retinal images were enhanced with Contrast Limited Adaptive Histogram Equalization (CLAHE). The efficient model was then trained using these images, achieving an AUC of 0.94 and 0.93 for MESSIDOR and IDRiD, respectively. Thota and Reddy [45] employed the VGG-16 model for diabetic retinopathy detection. They used transfer learning of the VGG-16 pre-trained model to achieve optimal performance. Working

with the Kaggle EyePACS dataset, they attained 74% accuracy, 80% sensitivity, and 65% specificity.

Mushtaq and Siddiqui [46] introduced the Densely CNN (DenseNet-169) model for diabetic retinopathy detection. They classified retinal images into DR, Not-DR, mild, moderate, and proliferative categories. The researchers used two datasets (Diabetic Retinopathy Detection 2015 and Aptos 2019 Blindness datasets) and applied preprocessing steps like cleaning, resizing, and augmentation. The deep learning model was then trained with the processed data, achieving a 90% accuracy rate.

The authors of [47] used an ensemble of five models from the EfficientNet family for DR grading, pre-training on ImageNet. They also tested these models independently for the same task. EfficientNet-B3 performed better than the ensemble model and the other four models. Parthasharathi et al. [48] developed an early diabetic detection system based on convolutional neural networks (CNN). They used a Kaggle dataset of 1000 images (300 diabetics and 700 normal). Images were first converted to HSV format, and yellow exudate extraction was performed from the color components. Median filtering and feature extraction were then applied, and the training process used the "Adam" optimization algorithm. The results showed an accuracy of 91.5%. Shaik and Cherukuri [49] introduced a model called "Hinge Attention Network (HA-Net)," using multiple attention modules for diabetic retinopathy severity grading. They employed the VGG-16 model to extract initial spatial representations and tested their experiments on the IDRid dataset, achieving an accuracy of 66.4%.Oulhadj et al. [50] used four CNN models, including DenseNet-121, Xception, InceptionV3, and ResNet-50. They registered retinal images from the Kaggle APTOS dataset and graded diabetic retinopathy using the CNN models. The results revealed that the highest accuracy achieved was 85.28%.

Lahmar and Idri [51] employed various DL models for feature extraction (VGG16, VGG19, Inception_V3, DenseNet201, MobileNet_V2, Inception_ResNet_V2, and ResNet50). Four different classifiers (SVM, MLP, DT, and KNN) were trained using the extracted features. The performance was evaluated using accuracy, sensitivity, precision, and F1-score. They used three different datasets (APTOS, Kaggle DR, and Messidor-2), achieving accuracies of 88.80%, 84.01%, and 84.05% for the three

datasets, respectively.

Table 2.2. Detailed comparison of the previous DL-based diabetic retinopathy studies.

Researcher	Year	Methodology	Dataset	Main Results	Limitations
Pratt et al. [37]	2016	CNN	Diabetic retinopathy Kaggle (80,000)	Specificity: 95%, Accuracy: 75%, Sensitivity: 30%	High false negatives
Soniya et al. [38]	2016	CNN single-based and CNN heterogeneous-based	DIARETDB 0 (130)	Accuracies ranging from 42.5% to 95%	Limited dataset size
Gargeya and Leng [39]	2017	Deep learning models	MESSIDOR 2, E-Ophtha	AUC: 0.94 (MESSIDOR2), 0.95 (E-Ophtha), Sensitivity: 93%/90%, Specificity: 87%/94%	No accuracy measure was computed.
Lam et al. [40]	2018	GoogleLeNet-v1, AlexNet, VGG-16, ResNet, Inception-V3	Kaggle EyePACS (243)	Best accuracy: 98% (InceptionV3)	Binary classification (DR or Not DR)
Nguyen et al. [42]	2020	Transfer learning of VGG16 and VGG19	Kaggle competition dataset 2015	Accuracies: 71% (VGG16), 73% (VGG19), Improved to 83% after modification	No ensemble or fusion were used
Tymchenko et al. [43]	2020	Three-head CNN	APTOS 2019 Blindness Detection Dataset	Sensitivity: 99%	Compute only one performance metrics
Pour et al. [44]	2020	EfficientNet B5	MESSIDOR , MESSIDOR -2, IDRiD	AUC: 0.94 (MESSIDOR), 0.93 (IDRiD)	Binary classification (DR or Not DR)
Thota and Reddy [45]	2020	VGG-16	Kaggle EyePACS	Accuracy: 74%, Sensitivity: 80%, Specificity: 65%	Low accuracy
Mushtaq and Siddiqui [46]	2021	DenseNet-169	Diabetic Retinopathy Detection 2015, Aptos 2019 Blindness	Accuracy: 90%	Moderate accuracy
Karki and Kulkarni [47]	2021	Ensemble of EfficientNet models	Kaggle APTOS	EfficientNet-B3 performed better than the ensemble and other models	No well-known evaluation metrics
Parthasharathi et al. [48]	2022	CNN	Kaggle (1,000)	Accuracy: 91.5%	Binary classification (DR or Not DR)

Shaik and Cherukuri [49]	2022	Hinge Attention Network (HA-Net)	IDRid	Accuracy: 66.4%	Low accuracy
Oulhadj et al. [50]	2022	DenseNet-121, Xception, InceptionV3, ResNet-50	Kaggle APTOS	Best accuracy: 85.28%	Moderate accuracy
Gulshan et al. [52]	2016	Deep Learning (Inception-v3)	EyePACS, MESSIDOR-2	97.5% Sensitivity, 93.4% Specificity	Retrospective design, results may not generalize to all populations
Ting et al. [53]	2019	Deep Learning (Deep Retinal Image Understanding)	Singapore National Eye Center	90.5% Sensitivity, 91.6% Specificity	Validation on multi-ethnic Asian dataset only; limited DR severity classification
Abramoff et al. [54]	2016	Deep Learning Built-in system (IDx-DR)	MESSIDOR-2, Iowa Detection Program	96.8% Sensitivity, 87.0% Specificity	No direct comparison with human experts
Gulshan et al. [55]	2018	Inception-v3	EyePACS and MESSIDOR-2	AUC: 0.99 (EyePACS), 0.99 (MESSIDOR-2)	Lack of external validation on diverse populations
Raju et al. [56]	2018	Modified U-Net	IDRiD	Sensitivity: 95.6%, Specificity: 98.6%	Limited dataset size
Ramachandran et al. [57]	2018	Transfer learning of ResNet-50	Kaggle EyePACS	Accuracy: 92.3%	Moderate accuracy
Chudzik et al. [58]	2018	VGG-16, VGG-19, Inception-v3, Inception-ResNet-v2, and Xception	EyePACS (243 images)	Best accuracy: 96.8% (Inception-ResNet-v2)	Low dataset size/ Binary classification

2.4. RELATED WORK CONCLUSION

In conclusion, previous work on diabetic retinopathy (DR) detection has employed various machine learning (ML) and deep learning (DL) techniques. ML-based methods, as seen in Table 2.1, have primarily used Support Vector Machines (SVM), Decision Trees, Random Forests, Naïve Bayes, and ensemble models. These models achieved varying degrees of success in terms of accuracy, sensitivity, and specificity. However, common limitations include small dataset sizes, lack of DR type or severity classification, and binary classification (DR or not DR).

On the other hand, DL-based methods, as seen in Table 2.2, have primarily utilized convolutional neural networks (CNNs), including well-known architectures such as

Inception, VGG, ResNet, DenseNet, and EfficientNet. These DL models demonstrated improvements in accuracy, sensitivity, and specificity compared to ML-based models. Limitations of these studies include high false negatives or positives, limited dataset size or diversity, binary classification (DR or not DR), lack of ensemble or fusion techniques, and moderate accuracy.

2.5. STUDY CONTRIBUTION

To address these limitations and improve DR detection, this proposal suggests using ensemble transfer learning models in a multi-stage dataset. Ensemble transfer learning leverages the strengths of multiple pre-trained models and combines them to enhance performance. This approach is expected to address the limitations of previous studies by incorporating diverse DR types and severity levels, reducing false negatives and false positives, applying data balancing approaches and improving overall accuracy, sensitivity, and specificity.

PART 3

MATERIALS AND METHODS

3.1. THE PROPOSED METHODS

In this chapter, the proposed methodologies used in the current study will be introduced and discussed. Besides, the utilized materials (datasets and software) will also be listed.

3.2. MATERIALS

3.2.1. Dataset

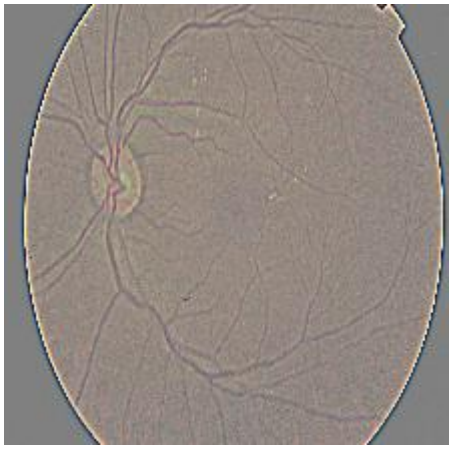
The used dataset is the APTOS 2019 Blindness Detection dataset, which is available for free at Kaggle [59]. This dataset consists of images of retinal scans that have undergone Gaussian filtering for detecting diabetic retinopathy. The original dataset is the APTOS 2019 Blindness Detection. The images of this dataset have been resized to 224x224 pixels to enable their use with various pre-trained deep learning models (using transfer learning).

The images are categorized into five categories, based on the severity of diabetic retinopathy, as specified in the train.csv file provided:

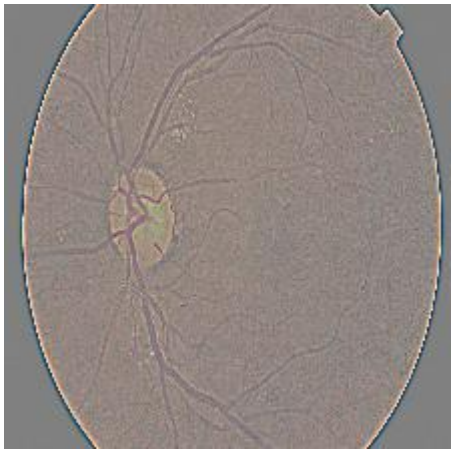
- No DR
- Mild
- Moderate
- Severe
- Proliferate_DR.

Additionally, the dataset contains an export.pkl file that includes a ResNet34 model trained using the FastAI library for 20 epochs on the dataset.

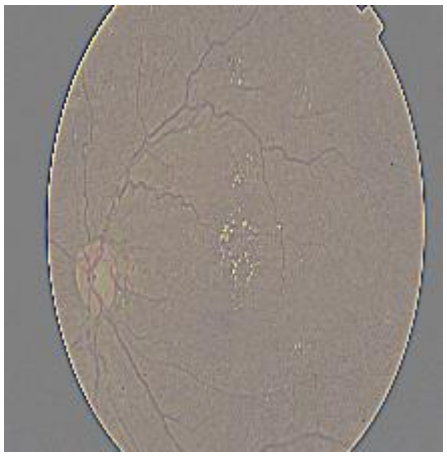
Figure 3.1 shows example of the five classes of this dataset.



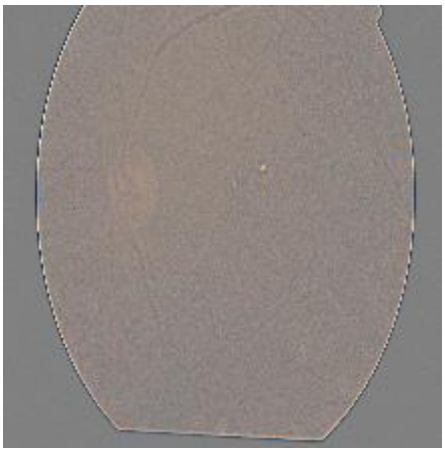
Mild



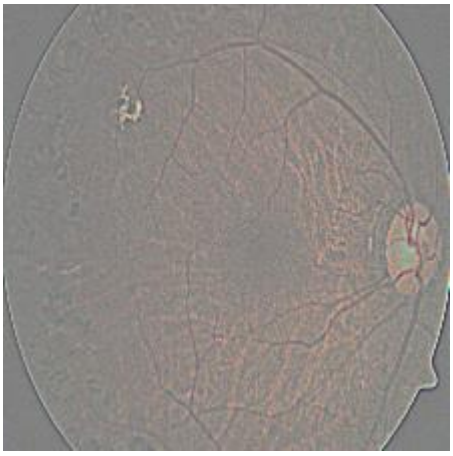
Mild



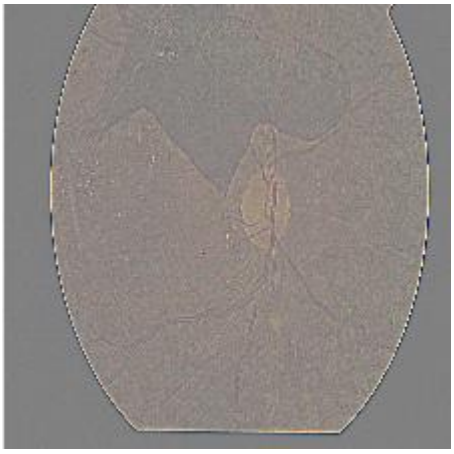
Moderate



Moderate



Proliferate



Proliferate

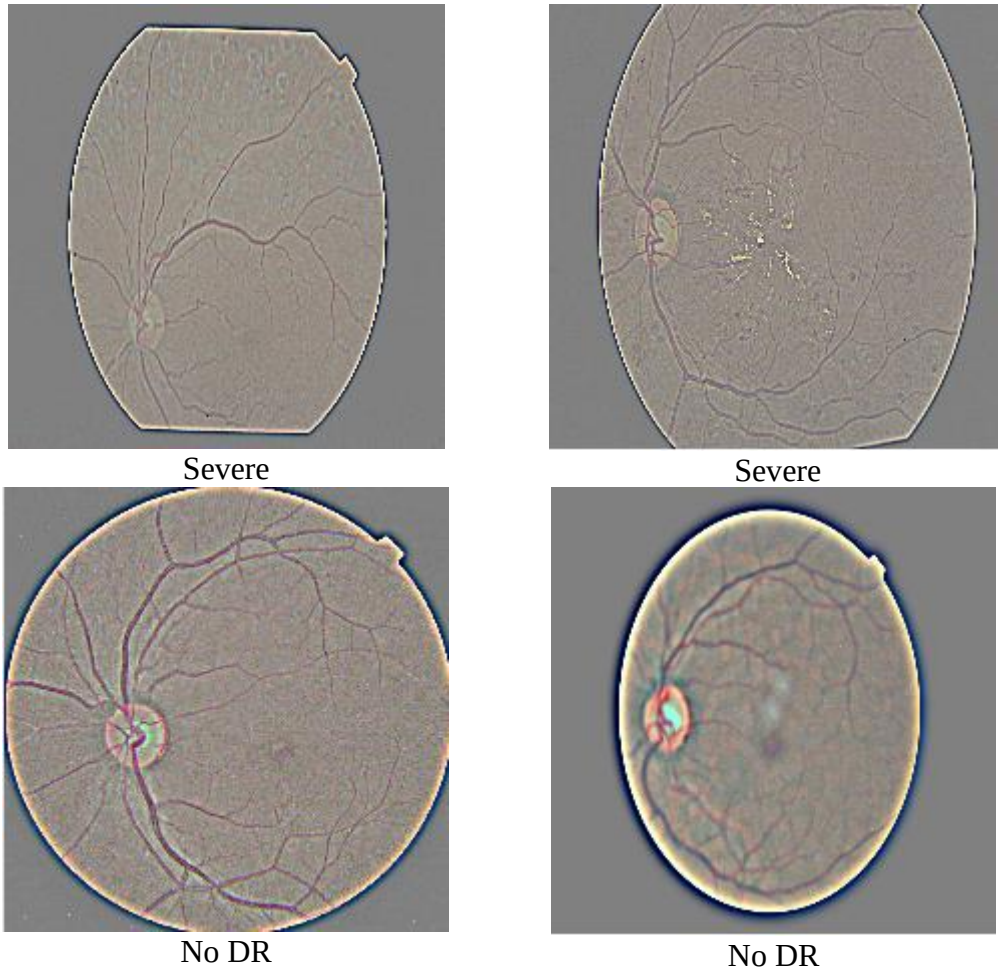


Figure 3.1. Examples of the used dataset samples

3.3. SOFTWARE

In this study, the Python programming language is suggested since it provides with a perfect machine learning and deep learning libraries. For implementing the proposed methodologies, the google Colab is used as a good environment for writing, editing and executing Python language codes.

The following libraries and dependencies are used:

- TensorFlow: TensorFlow is an open-source machine learning framework that is used to build and train deep learning models. It provides a comprehensive set of tools and libraries for building and deploying machine learning applications. TensorFlow is widely used in the fields of image recognition,

natural language processing, and many other applications. For our implementation, the library will be used for deep learning models creation and training.

- Keras: Keras is a high-level API for building and training deep learning models. Keras is built on top of TensorFlow allowing users to easily build and train neural networks with a few lines of code. This library provides a simple interface for building and training deep learning models.
- NumPy: NumPy is a Python library that provides support for large, multi-dimensional arrays and matrices, along with a large collection of mathematical functions to operate on these arrays. It is widely used in scientific computing, data analysis, and machine learning applications.
- Pandas: Pandas is a Python library that provides high-performance data manipulation and analysis tools. It is built on top of NumPy and provides support for working with tabular data, including data reading and writing, data filtering, aggregation, and merging.
- Random: The random library is a built-in Python library that provides tools for generating random numbers, sequences, and selections. It is commonly used in simulations, games, cryptography, and other applications that require randomness.
- OS: The os library is a built-in Python library that provides a way to interact with the operating system, including creating, deleting, and renaming files and directories, manipulating paths, and running system commands.
- CV2: The cv2 library (OpenCV) is an open-source computer vision library that provides tools for image and video processing, including image filtering, feature detection, object detection, and tracking.
- Shutil: The shutil library is a built-in Python library that provides a way to work with high-level file operations, including copying, moving, and deleting files and directories.
- Matplotlib: Matplotlib is a Python library that provides support for creating static, animated, and interactive visualizations in Python. It provides a variety of plot types, including line plots, scatter plots, bar plots, and more. It is widely used in scientific computing, data analysis, and machine learning applications.

3.3. DEEP LEARNING (DL)

For a long time, traditional machine learning methods struggled to address complex problems despite attempts to enhance them. However, deep learning methods have achieved remarkable performance in a variety of applications, including image recognition, big data analysis, natural language processing, and speech recognition. The backpropagation algorithm is the primary training method for deep neural networks, with training taking place in two stages: a forward step to compute errors and a backward step to adjust weights and learn. Convolutional neural networks (CNNs) are the most widely used deep learning networks for image recognition. With the use of CNNs, it is possible to improve the accuracy of many applications, particularly in the field of medical image recognition. Moreover, with the help of transfer learning techniques, pre-trained models can be utilized to reduce the computation required to train deep learning networks [60].

3.3.1. Convolutional Neural Network (CNN)

The ConvNet is a popular deep learning network that comprises two main layers: convolution and pooling. These layers consist of feature maps that store the results of convolutions applied to the input image, and many filters or kernels are applied to the input image in each layer. The Relu non-linear function is then applied to add non-linearity to the output, which is called the activation map. The next layer is the max-pooling layer, which reduces the dimensions of the convolution to save computational time. The output of this combination is then passed to the next combination of layers. The filter values represent the weights of the network that are adjusted during training to achieve the best values for the model. After the convolution-pooling combination, a fully connected layer is added to reshape the feature map into a single vector for output. Sometimes, a dropout layer is used to drop a percentage of neurons in the fully connected layer to avoid overfitting.

Finally, the softmax activation function is applied to produce output as probabilities of all classes, and the class with the highest probability is chosen. The architecture of the ConvNet is depicted in Figure (3-2) [61]. Usually, the convolution and pooling layers

represent the feature extraction part, while the fully-connected layer and the output (softmax) layer is considered the classification layers.

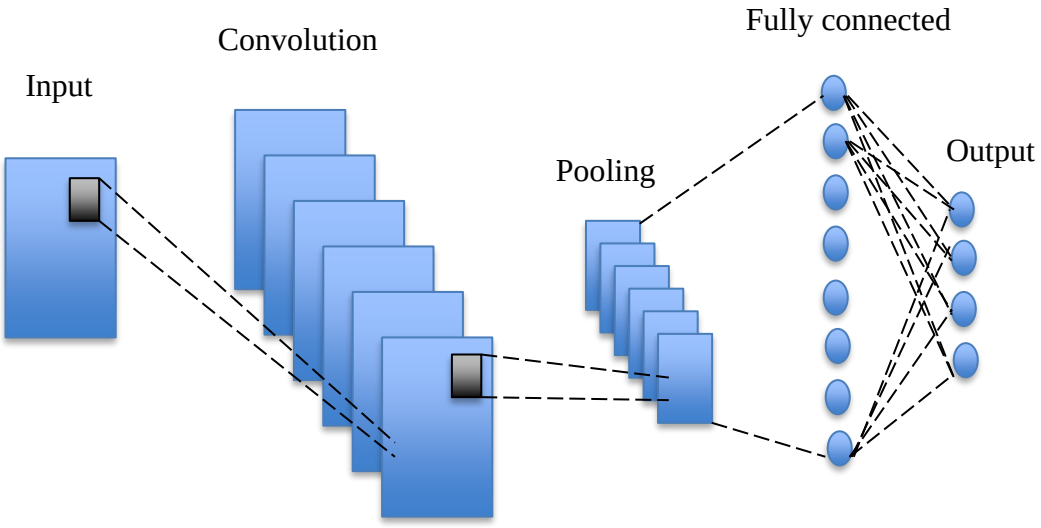


Figure 3.2. CNN Architecture [62].

3.3.2. Some Deep Learning Keywords

To apply the filter kernel on the image inside the convolution layer, two important principles must be defined - padding and stride.

Padding is essential to process the pixels on the border of the image since they don't entirely match the kernel size, and the image must be padded to avoid losing information [63]. If we choose not to pad the image, the resulting activation map of the convolution will be smaller than the original image. To pad the image, we can use zeros, and the value of padding will be as Equation (3.1) illustrates.

$$P = \text{Floor}((F-1)/2) \tag{3.1}$$

Where,

F is the kernel size. This will ensure that the size of the input and output in case of padding will be the same, as shown in Figure (3-3). Proper padding ensures that the

input and output sizes remain consistent throughout the convolutional layers. Moreover, stride refers to the number of pixels the kernel is shifted each time it passes over the input image. By adjusting the stride, we can control the size of the output feature maps.

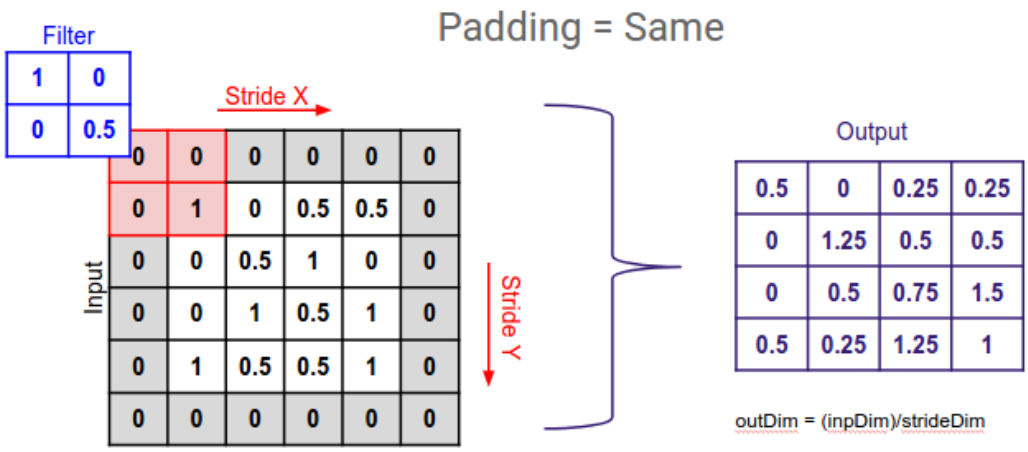


Figure 3.3. Convolution with padding and stride [64]

Stride, is the sliding parameter by which the window of the kernel is moving horizontally and vertically from pixel to the next one. The size of the convolution is defined as equation (3-2) shows.

$$(W - F + 2P) / S + 1 \quad (3.2)$$

Where,

W is the image size, F is the kernel size, P is the padding and S is the stride.
If the used convolutional mask is of size 7x7, padding of 3 and stride of 3, then the convolution result will be of size: (150 – 7 + (2)(3)) / 3 + 1 = 50*50. The activation records after the pooling layer of size 2*2 will be of size 25*25.

3.4. PROPOSED METHODS

The proposed methodology of the current study is illustrated in Figure 3.4.

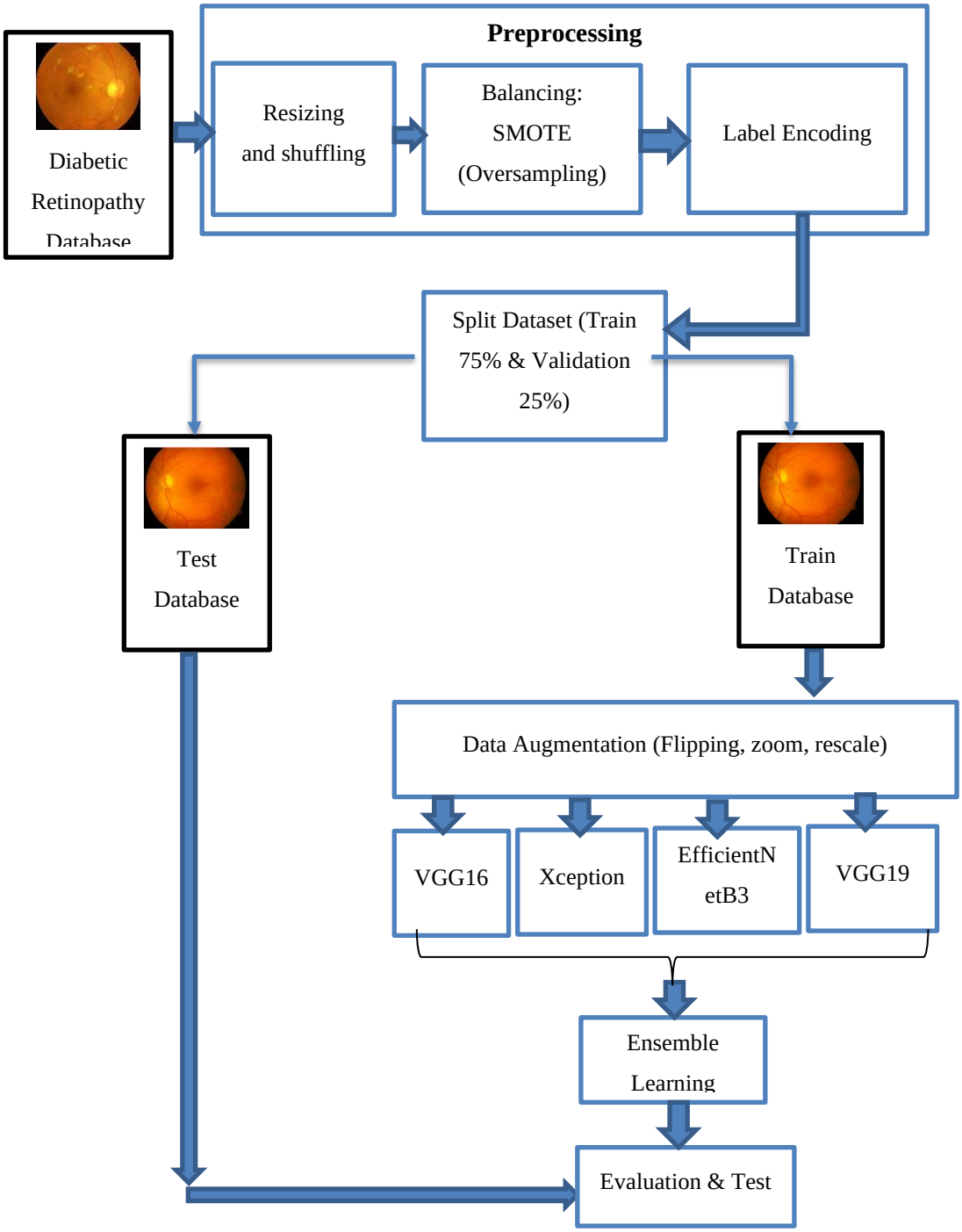


Figure 3.4. Proposed methodology.

3.4.1. Preprocessing

The preprocessing steps are essential to process the image datasets before going to the next steps. For the proposed diabetic retinopathy dataset, many preprocessing steps are suggested. First, the images are resized into a fixed size 150x150 in order to minimize the training time and other next processes.

3.4.2. Balancing (Over sampling)

After preprocessing operations, the balancing technique is applied to make balance to the dataset. This operation is essential since the number of samples of each category in this dataset is very different. Category "No DR" has the highest number of samples which is 1805, while proliferate category has only 295 samples. Severe category contains only 193 samples, while "Moderate" category includes 999 samples. The "Mild" category contains 370 samples.

This difference in the size of each category will affect the training operation by biasing the learning to the dominant category so the models will learn to classify samples to the most frequent class.To avoid this problem, the Synthetic Minority Over-Sampling Technique (SMOTE) algorithm [65] is used to increase the number of minor classes samples and make some balance in the dataset. This operation will balance the learning process and give classes similar weights. The new number of samples of each category will be as follows: No_DR': 1805, 'Mild': 600, 'Moderate':1200, 'Severe':400, 'Proliferate_DR':400.Figure 3.5 shows the dataset label distribution before and after SMOTE. The extreme oversampling (making all categories with the same number of samples) is not suggested here since it can cause fake training and increase the computational training time.

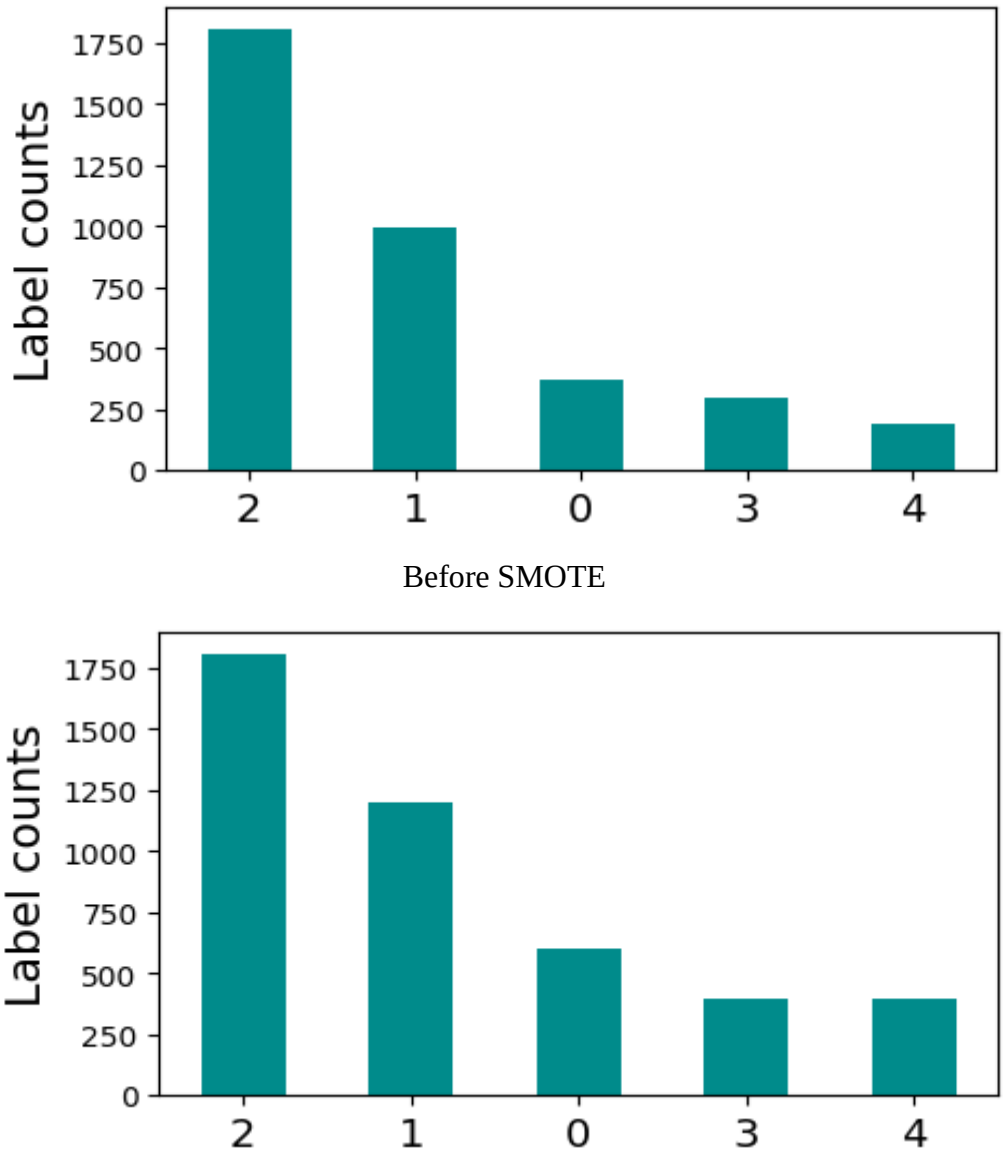


Figure 3.5. Dataset balancing using SMOTE

3.4.3. Label Encoding

The labels (Targets) of the dataset are sometimes written in textual formula, so they need to be encoded before the training process. Label Encoding is an essential step in preprocessing data for machine learning algorithms, especially when dealing with categorical variables that are represented as text. Label Encoding refers to converting categorical data, which is in the form of text or labels, into numerical shape. This is necessary because most machine learning algorithms can only work with numerical data.

The sklearn.preprocessing module includes a LabelEncoder class that can be used to perform Label Encoding. It class takes a categorical variable (textual form) and encodes its values into numbers (numerical form). Each unique value in the categorical variable is assigned a unique number. For example, if a categorical variable has four unique values "hard", "medium", "easy" and "None", the LabelEncoder class would convert "hard" to 0, "medium" to 1, "easy" to 2 and "None" to 3. The aim of label encoding is that it enables the algorithms to process the data accurately and efficiently. Without applying the label encoding, the algorithm may deal with the categorical variables as ordinal variables. It may try to apply mathematical operations to them, leading to inappropriate results.

Label encoding is also helpful with large datasets that include many categorical variables, as it can significantly reduce the memory required to store the data.Overall, Label Encoding is an essential process to preprocess data for machine learning and deep algorithms and the LabelEncoder class from the sklearn.preprocessing module makes it easy to perform this task efficiently and accurately.For the diabetic retinopathy dataset, the Label encoder will produce the following: 0 for "Mild", 1 for "Moderate", 2 for "No DR", 3 for "Proliferate", and 4 for "Severe".

3.4.4. Dataset Split

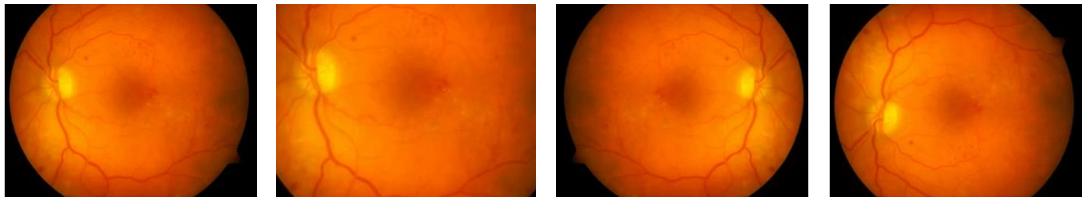
This step aims to split the dataset into two different parts: the training dataset that will be used for training the DL models, and the test dataset for evaluation process. In the current study, the dataset is split into 75% for training and 25% for test.

3.4.5. Data Augmentation

Data augmentation is a process used to increase the size of an image dataset by generating new samples with specific operations on the input images. This process is done by creating new images from the existing ones, usually through the application of various geometrical operations and transformations. These augmented images can be used to train machine learning and deep learning models more effectively, especially in cases where the original dataset has small size or is insufficient.

The proposed data augmentation techniques used for the current diabetic retinopathy datasets include rescaling, zooming, and flipping. Rescaling involves dividing the image values by 255 in order to normalize all gray levels to be in the range [0-1], while zooming involves selecting a specific region of the image and increasing its size. Flipping, on the other hand, involves flipping the image horizontally or vertically. These techniques can be used alone or in combination to create a large number of new images, which can help improve the accuracy and robustness of machine learning models trained on image datasets.

These operations are done randomly in each epoch of the training process.



Original Zoom in Flip Horizontally Flip Vertically

Figure 3.6. Examples of data augmentation operations on a sample of the diabetic retinopathy dataset

3.4.6. VGG Models

VGG16 and VGG19: The Visual Geometry Group (VGG) at the University of Oxford proposed VGG16 and VGG19 in their 2014 publication [66]. Both networks use a thick stack of convolutional layers and small 3x3 convolutional filters.

In contrast to VGG19, which contains 19 weight layers and 16 convolutional layers, VGG16 consists of 16 weight layers, comprising 13 convolutional levels and 3 fully linked layers. Whereas VGG19 has over 144 million parameters, VGG16 has about 138 million [66].

Figure 3.7 shows the difference between VGG16 and VGG19 models [66] [67].

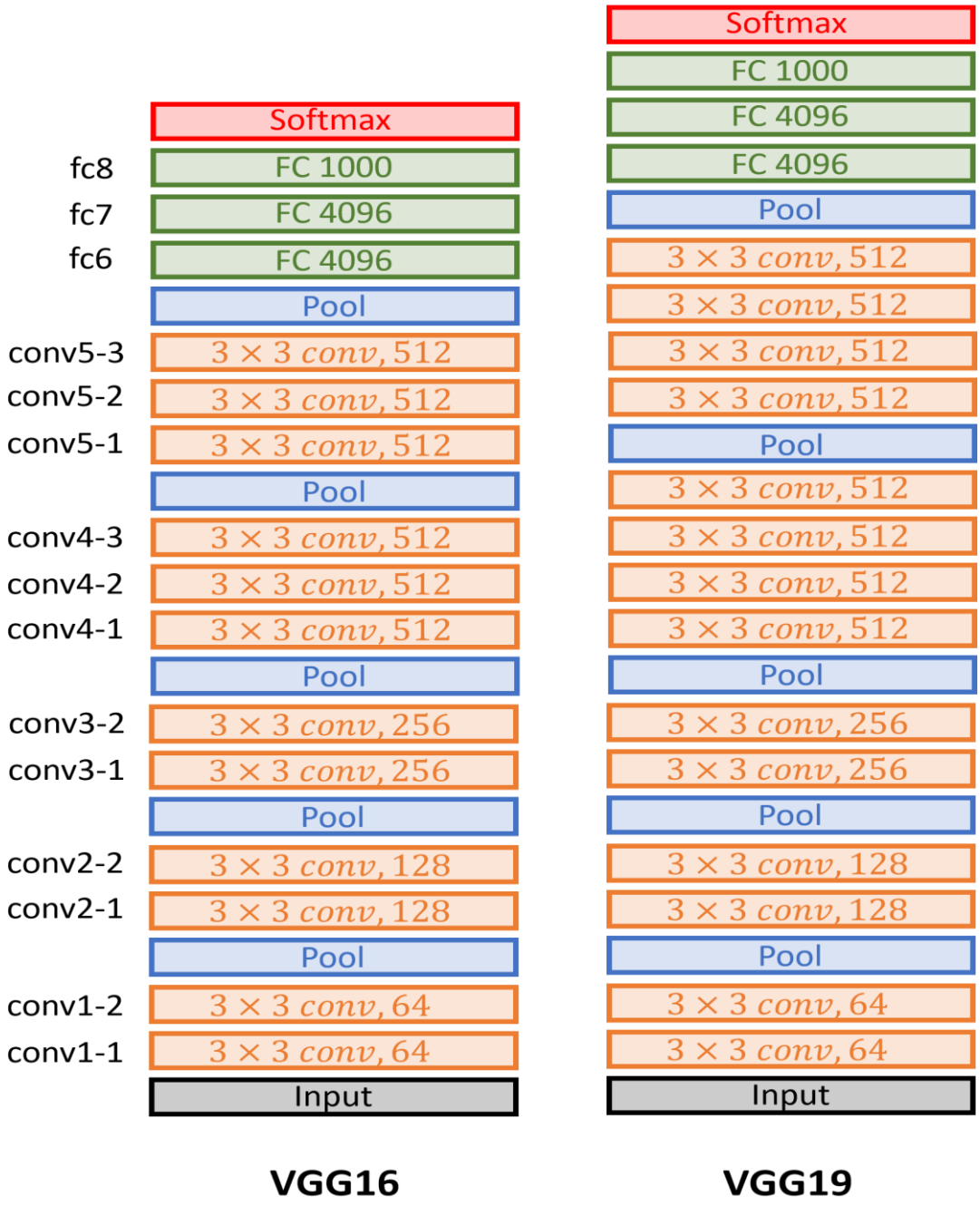


Figure 3.7. VGG16 VS. VGG19 models.

3.4.7. Xception Model

In the 2016 [68], François Chollet, the developer of the Keras library, developed the deep learning model known as Xception (short for "Extreme Inception"). Depthwise separable convolutions, which are more performant than conventional convolutions and are an extension of the Inception architecture, are used in this model. The 36 levels

of the Xception model include an entering flow, middle flow, and exit flow. Compared to VGG16 and VGG19, it has much fewer parameters (22.9 million) while performing better [68]. Figure 3.8 shows the architecture of Xception model.

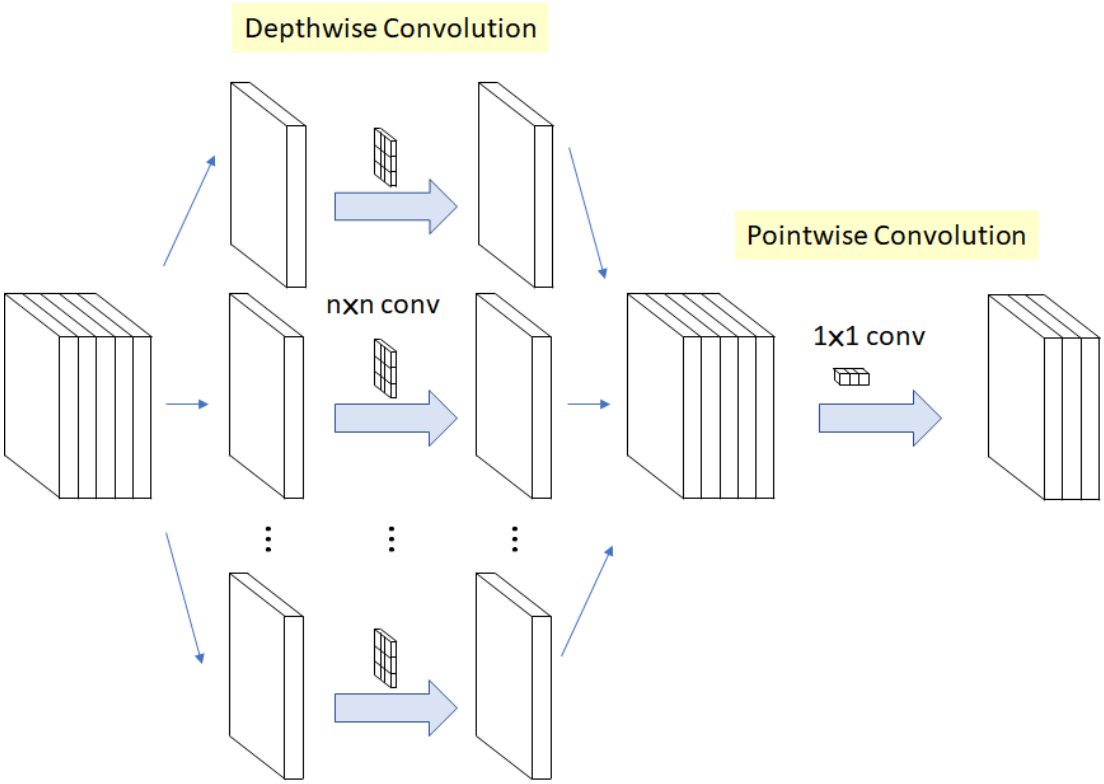


Figure 3.8. Xception model.

3.4.8. EfficientNetB3 model

EfficientNet is a family of deep networks that was invented by Tan and Le [69]. These models use a compound coefficient that is designed to scale breadth, depth, and resolution all at once, increasing accuracy and efficiency. Many versions of this model were created. For the current study, the EfficientNetB3 model is utilized. It comprises 154 layers with about 12 million parameters total, including layers for squeeze-and-excitation, batch normalization, and convolution [69]. The main benefit of EfficientNetB3's is its capacity to retain high accuracy while maintaining a more compact model size comparing to other previous architectures. Figure 3.9. shows the architecture of compound scaling approach used in EfficientNet model [69].

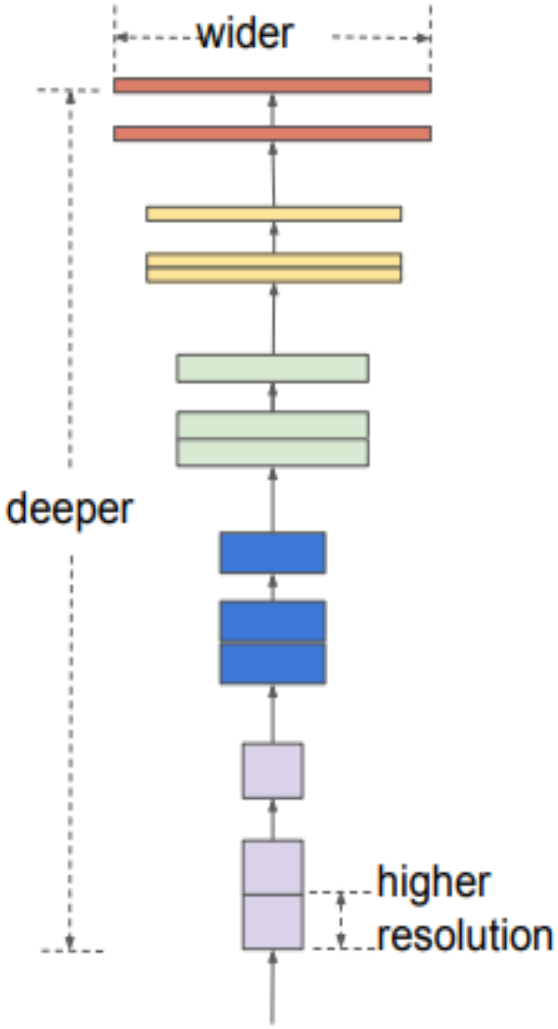


Figure 3.9. Compound Scaling.

3.4.9. Ensemble Learning

The ensemble learning is a well-known approach that construct a unified model of many individual ones. The main benefit of this technique is to get the efficiency and powerful of each individual model. There are many methods to generate the ensemble, including the bagging and boosting ways.

There are three main ensemble learning approaches [70]:

- Boosting: The main problem of some trained models is the low accuracy which can be enhanced by using an ensemble of weak classifiers. Each subsequent

model works to minimize the errors made by the previous models. The final classification/prediction result is a weighted fusion of the predictions from all the base models, where the weights depend on the accuracy of each individual model.

- Bagging (Bootstrap Aggregating): Bagging includes creating multiple copies of the original training dataset. This approach is done through random sampling with replacement. Each of these datasets is used to train a separate base model. The final prediction decision is calculated by averaging the predictions of all base models (for regression) or by taking a majority vote (for classification).
- Stacking (Stacked Generalization): Stacking trains many base models on the original dataset. After that, the stacked model uses the predictions as input features to train a second-level model called the meta-model. The meta-model learns to fuse the predictions of the base models to produce the final prediction.

Ensemble learning techniques are widely used in various machine learning tasks such as classification, regression, and even unsupervised learning tasks. They have shown to be highly effective in reducing model bias and variance, thus leading to improved generalization and better performance on unseen data.

Figure 3.10 shows the architecture of the proposed ensemble model which uses the stacking approach.

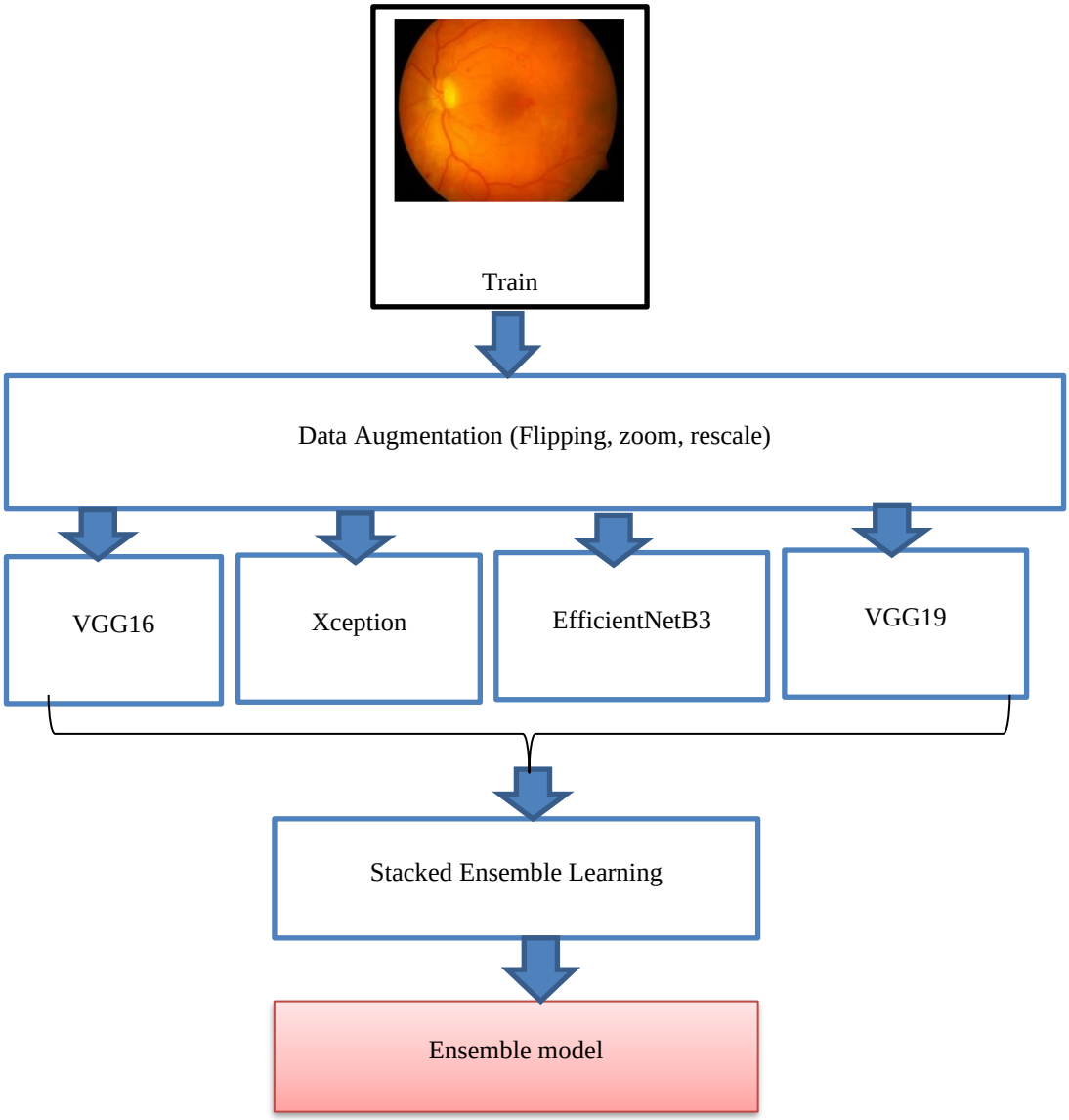


Figure 3.10. The proposed ensemble learning method.

Figure 3.11 shows the ensemble model, including training and evaluation steps.

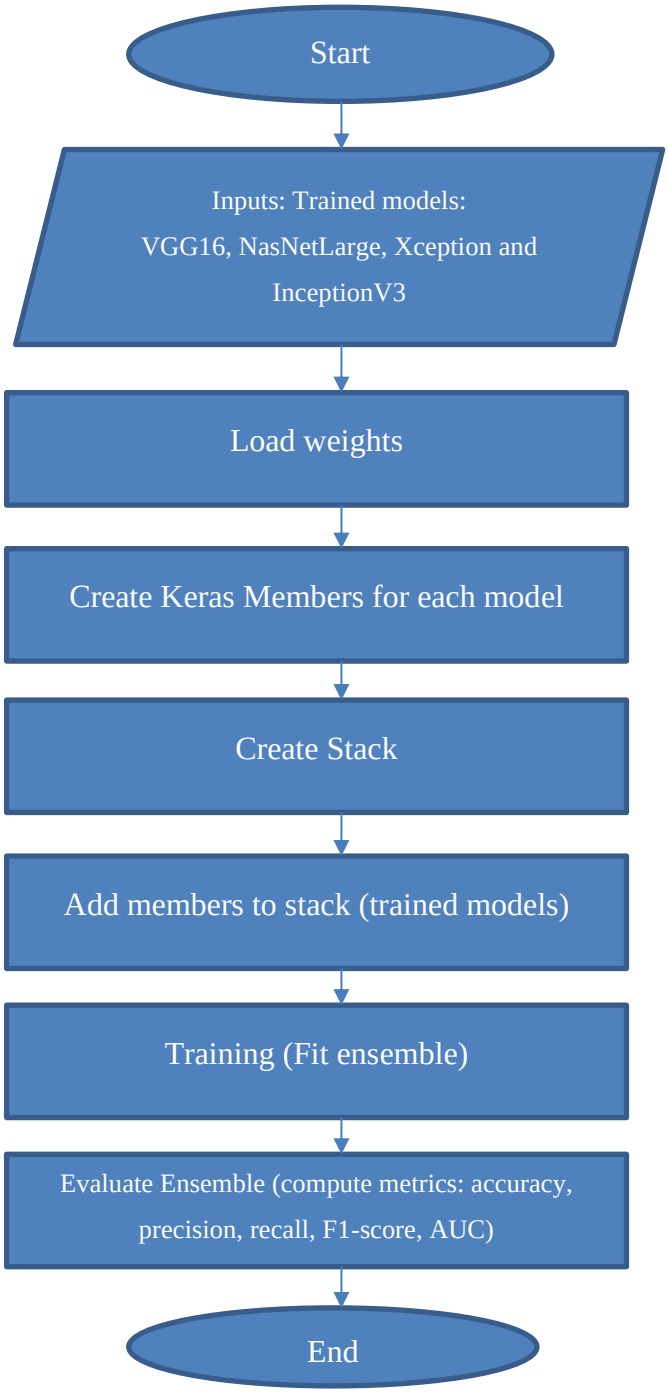


Figure 3.11 Proposed Ensemble model.

First, the trained models will be used as input for the ensemble model. Then, model's weights will be loaded, and the stack members will be created and added to the stack. The next step is to train the ensemble model. The final step is the evaluation of the

ensemble model using the test set. The metrics that are computed are the accuracy, precision, recall and F1-score.

3.4.10. Performance Evaluation

To evaluate our segmentation and recognition methodologies, we suggest using the following metrics [71] [72]:

- Training/validation/test Accuracy, which expresses the accuracy of diabetic retinopathy system. The accuracy will be computed for both training and test sets.
- Precision: TP/(TP + FP)
- Recall: TP/ (TP+FN)
- F1-score: 2*Precision*Recall / (Precision + Recall)

These calculations need the following information:

True Positive (TP): Represents the number of correctly classified samples against all samples in the dataset.

False Positive (TN): the number of correctly rejected samples against all samples in the dataset.

Figure 3.12 shows these calculations.

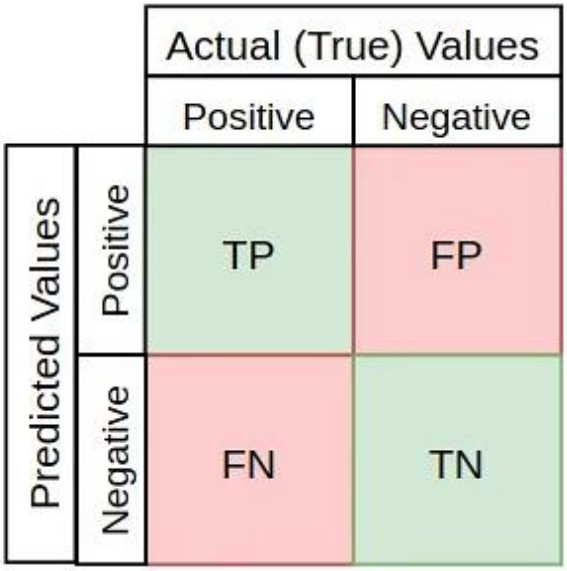


Figure 3.12. TP, TN, FP and FN calculations

3.4.11. Binary Classification VS. Multi-Class Classification

In the current study, the multi-class classification process is applied to detect all stages of diabetic retinopathy, including (mild, severe, moderate and proliferate) besides the normal condition.

In the binary classification, the diabetic categories are merged in one class (disease class). Besides this class, the normal class is left. Due to this modification, two categories are obtained (0: Normal condition, 1: for diabetic retinopathy disease).

The idea of using this scenario is that most previous studies applied the binary classification so in order to compare the current study to the previous state-of-art, the binary classification scenario is applied. The Table 3.1 shows the categories distribution over the dataset in the binary classification scenario.

Table 3.1. Class distribution of the binary classification scenario.

Class	**Number of samples**	**Percentage**	**Fused categories of the original dataset**
Normal	1805	49.29%	Normal
Disease	1857	50.71%	Mild, severe, moderate and proliferate

PART 4

RESULTS

4.1. INTRODUCTION

In this chapter, the training and evaluation experiments, including all applied scenarios, will be listed, compared, and discussed in a detailed way. The results will also be compared with the previous related studies in the same field of diabetic retinopathy detection.

4.2. THE PROPOSED TRAINING SCENARIOS

In this study, two main training scenarios are proposed. The first one is the training without balancing the dataset, while the second one is the balancing scenario in which the dataset is balanced, and then the training is performed.

Under those two different scenarios, many training scenarios are also performed.

In these sub-scenarios, many DL architectures were proposed and the ensemble learning approach is also applied to improve the performance of those selected DL models.

Another third scenario is proposed to compare the performance of multi-class diabetic retinopathy with the binary-class classification case. In the third scenario, the dataset labels are grouped into only DR and NO_DR. The same DL models and the ensemble model are also used and evaluated.

4.3. UNBALANCED TRAINING RESULTS

First, the dataset is used in its original state. Figure 4.1 shows the distribution over the original dataset. The dominant class is the 'NO DR' class, while the least frequent class is the 'Severe' class.

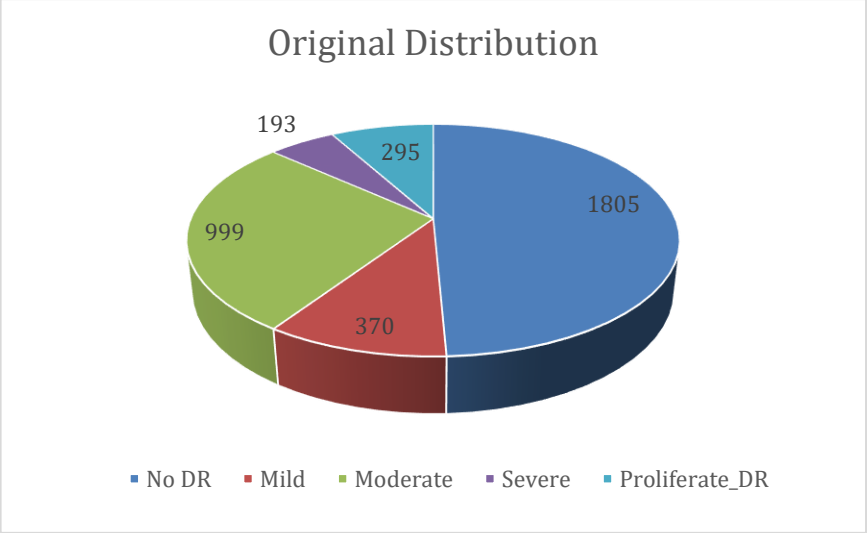


Figure 4.1. Original dataset distribution.

The dataset is split using 80% for training and 20% for validation and test. The following transformations are applied:

- Convert all images into RGB
- Resize all images into 224*224 in order to reduce the training time.
- Convert the targets to categorical form in order to compute probabilities for each one in the training step (the class with the highest probability will be the predicted class).
- Shuffle training images in order to prevent the training process from memorizing the order of the training samples.

4.3.1. Training Parameters (Training Options)

In order to accomplish the training process, the following training options are used:

- The batch size is 64. Batch size is used to send images to the training processes as patches which reduces the training time since the training step will include 64 images at the same time instead of using only one image. This option requires using the GPU instead of CPU in order to apply this parallel computing.
- Output size: the output size is the number of classes which is 5.
- The hidden layers' activation function: Relu is the common activation function and is used in the current study.
- Output layer activation function: SoftMax.
- Optimizer: Adam algorithm.
- Loss function: categorical cross entropy (since our problem is a multi-class classification problem).
- The training metrics: accuracy.
- Learning rate: 0.001.
- Dropout rate= 25% (This rate represents the proportion of neurons that are dropped from the preceding layer before the dropout layer is applied).
- Early Stop condition: The early stop condition is set based on the validation loss with a patience factor of 10 so if the training process caused no enhancement in the validation loss for 10 epochs, the training will be stopped even though the maximum epochs haven't reached).
- Reduce learning rate option: this option is used to reduce the learning rate at a specific condition. In the current study, the learning rate will be decreased by a factor of 0.3 if the validation loss is not enhanced for two epochs.

4.3.2. Training Scenarios

In all training scenarios, the used model consists of two main parts; the feature extraction part and the classification part.

In the feature extraction part, many DL models are suggested, while the classification part is fixed for all models.
Many training scenarios are proposed including the following:

- Training the VGG-16-based model using the unbalanced version of the dataset.
- Training the NasNetLS2-based model using the unbalanced version of the dataset.
- Training the Xception-based model using the unbalanced version of the dataset.
- Training the InceptionV3-based model using the unbalanced version of the dataset.
- Make an ensemble of the models of the scenarios (1-4) using the unbalanced version of the dataset.

4.3.3. Results of Training VGG-16 As A Base Model Using the Unbalanced Version of the Dataset

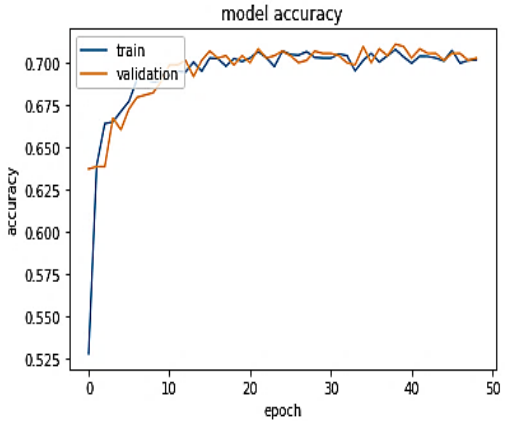
In this scenario, the VGG16 model is used as the base model in order to extract image features. The output of VGG16 model is a feature vector of size 512. Figure 4.2 shows the architecture, the output size and number of trainable parameters of the VGG16-based model.

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 512)	14714688
dropout (Dropout)	(None, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 64)	32832
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
dense_2 (Dense)	(None, 5)	165

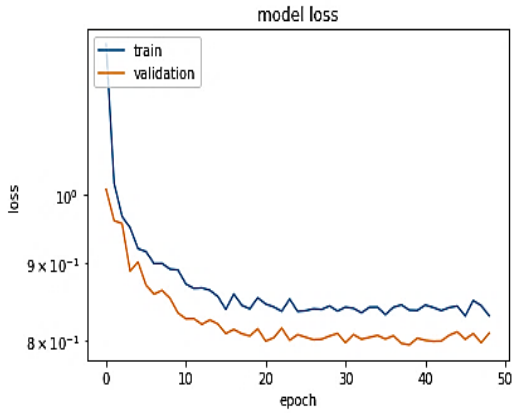
Total params: 14,749,765
Trainable params: 35,077
Non-trainable params: 14,714,688

Figure 4.2. VGG16-based DL model for diabetic retinopathy detection

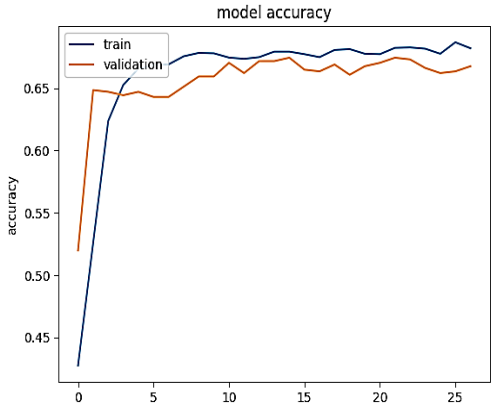
The VGG base model weights are frozen so that the number of trainable parameters is computed for the classification part only (35077 parameters). For the first model, three different optimizers will be used and compared in order to define the best optimizer and continue the other scenarios with the best optimizer. The first optimizer is Adam, the second one is Stochastic Gradient Descent (SGD), while the third one is Root Mean Square Propagation (RMSprop)). Figure 4.3 shows the training and validation accuracy and loss using three different optimizers (Adam, SGD (Stochastic Gradient Descent) and RMSprop).



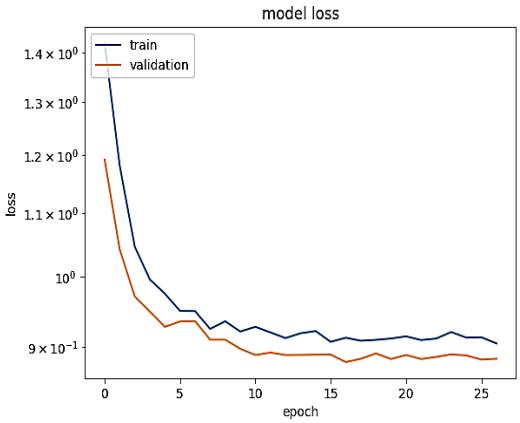
Adam - Accuracy



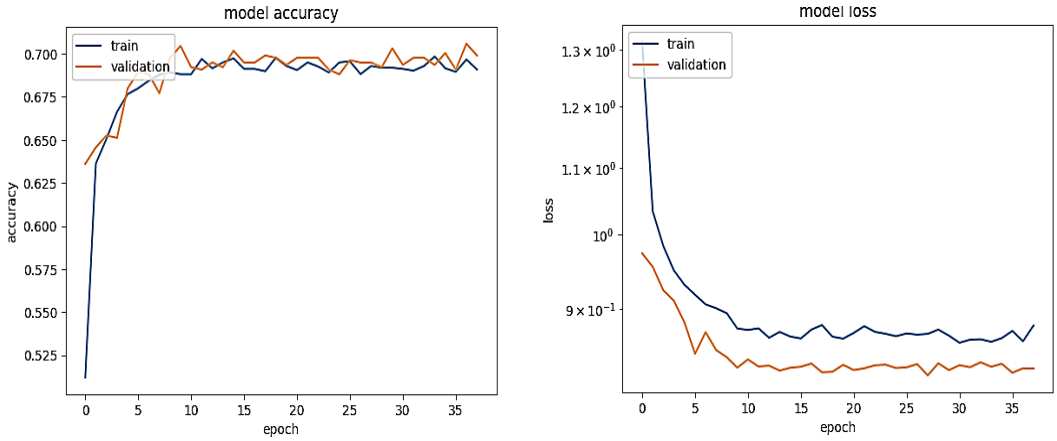
Adam - Loss



SGD - Accuracy



SGD – Loss



RMSprop - Accuracy RMSprop - Loss

Figure 4.3. Training and validation accuracy and loss curves of the VGG16 based DL model of diabetic retinopathy detection using three different optimizers (Adam, SGD, RMS)

Results of this scenario observed by Figure 4.3 show that the validation accuracy is almost 71% and the validation loss is 0.804 using the Adam optimizer. While the training accuracy is 70.69% and the training loss is 0.8425. The average training time is 49s/step. For SGD optimizer, the validation accuracy is 67% and the training accuracy is 66.76%. For RMSprop, the training and validation accuracy are 69.5% and 69%, respectively. The training time for SGD and RMS are 50 s/step and 54s 5s/step, respectively. The best optimizer is the Adam optimizer so the next scenarios will be continued with Adam optimizer.

The detailed results, including the precision, recall and F1-score for all optimizers are illustrated in Table 4.1.

Table 4.1. Precision, Recall, and F1-score of the VGG based DL model of diabetic retinopathy detection.

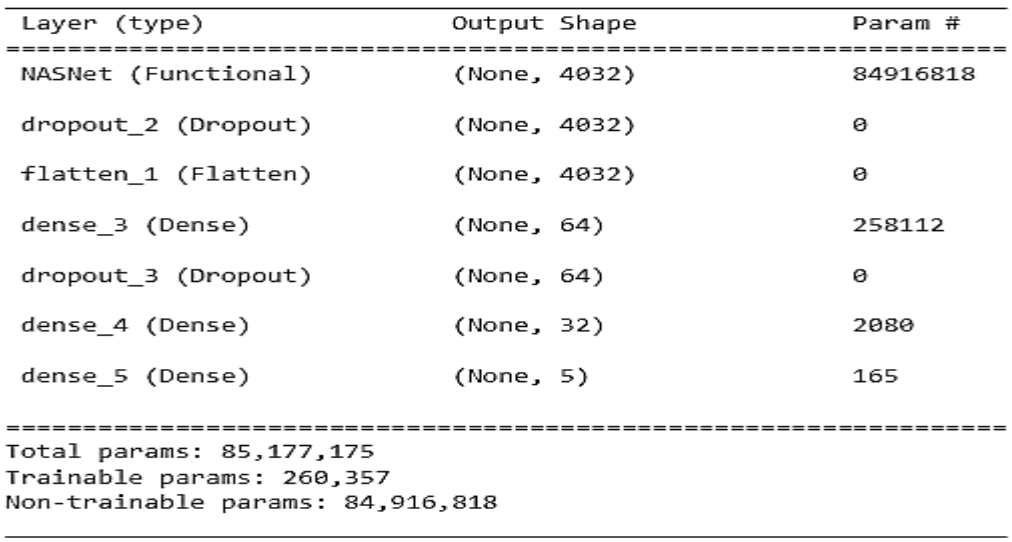
VGG16	Precision %	Recall %	F1-score %	Num. of test samples
		Adam		
No_DR	90	94	92	361
Mild	0	0	0	74
Moderate	50	88	64	199
Severe	0	0	0	38
Proliferate_DR	0	0	0	59
Macro avg	28	37	31	731
Weighted avg	58	71	63	731
		SGD		
No_DR	92	86	89	361
Mild	0	0	0	74
Moderate	45	88	60	199
Severe	0	0	0	38
Proliferate_DR	0	0	0	59
Macro avg	27	35	30	731
Weighted avg	58	67	60	731
		RMSprop		
No_DR	88	94	91	361
Mild	0	0	0	74
Moderate	49	85	62	199
Severe	0	0	0	38
Proliferate_DR	0	0	0	59
Macro avg	27	36	31	731
Weighted avg	57	69	62	731

Table 4.1 shows that the classes with small number of samples have 0% result for all calculations which is due to the unbalance issue between classes. The same result of the best optimizer is shown again in Table 4.1, where the best optimizer is Adam.

4.3.4. Results of training NasNetLS2 as a Base Model Using the Unbalanced Version of the Dataset

The NasNetLarge model is a large model with more number of trainable parameters. NasNetLarge is used as a base model to extract image features. The output of NasNetLarge model is a feature vector of 4032 samples. Figure 4.4 shows the

architecture, the output size and number of trainable parameters of the NasNetLarge-based model.



Layer (type)	Output Shape	Param #
NASNet (Functional)	(None, 4032)	84916818
dropout_2 (Dropout)	(None, 4032)	0
flatten_1 (Flatten)	(None, 4032)	0
dense_3 (Dense)	(None, 64)	258112
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 32)	2080
dense_5 (Dense)	(None, 5)	165

Total params: 85,177,175
Trainable params: 260,357
Non-trainable params: 84,916,818

Figure 4.4. NasNetLarge-based DL model for diabetic retinopathy detection.

The NasNetLarge base model weights are frozen so that the number of trainable parameters is computed for the classification part only (260357 parameters). The number of trainable parameters is increased here (although the architecture of the classification part is the same as in previous scenario) due to the fact that the output of the base model is bigger (4032 instead of 512) which is exactly the output size of the flatten layer of the classification model.

Figure 4.5 shows the training and validation accuracy and loss.

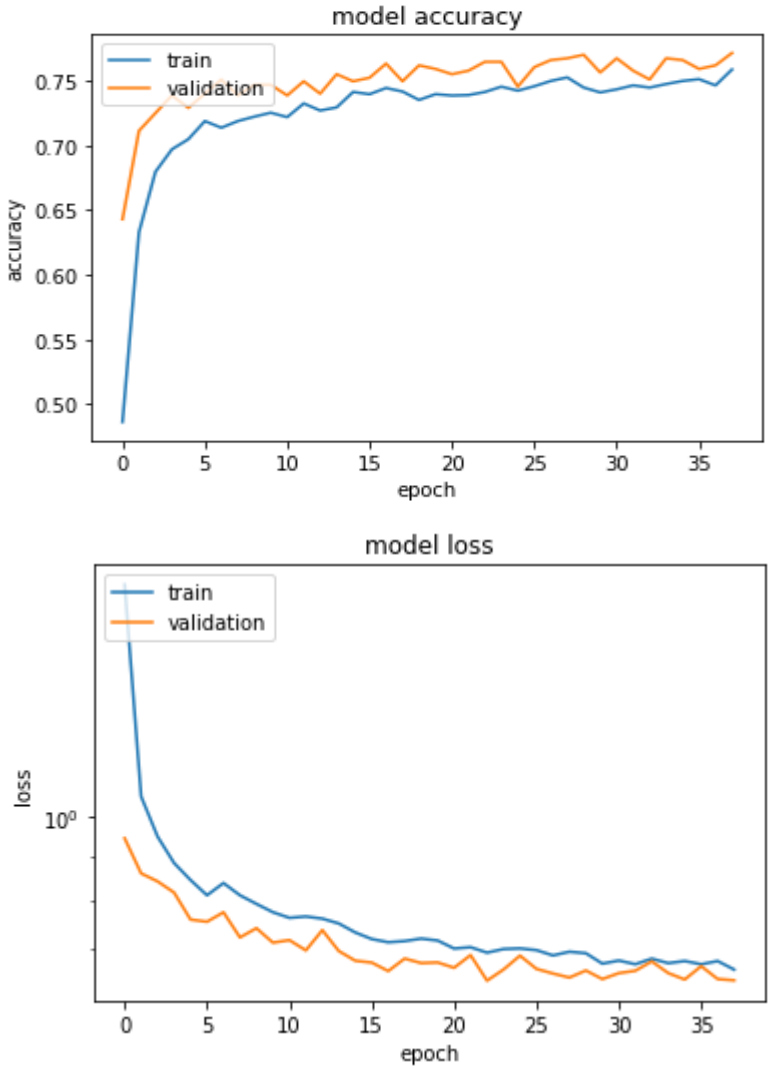


Figure 4.5. Training and validation accuracy and loss curves of the NasNetLarge based DL model of diabetic retinopathy detection

Results of Figure 4.5 show that the validation accuracy is almost 77% and the validation loss is 0.643. While the training accuracy is 75.88% and the training loss is 0.662. The average training time is 66s/step (the NasNetLarge model requires a higher computational time comparing to the VGG16 model). The detailed results, including the precision, recall, and F1-score, are illustrated in Table 4.2.

Table 4.2. Precision, Recall, and F1-score of the NasNetLarge-based DL model of di diabetic retinopathy detection.

NasNetLarge	Precision %	Recall %	F1-score %	Num. of test samples
No_DR	95	98	97	361
Mild	68	35	46	74
Moderate	58	93	72	199
Severe	33	3	5	38
Proliferate_DR	0	0	0	59
Macro avg	51	46	44	731
Weighted avg	72	77	72	731

Table 4.2 shows that some minor classes' performance enhanced slightly compared to the NasNetLarge model. However, some other classes' performance is still 0% (Proliferate) due to the unbalance issue between classes.

4.3.5. Results of training Xception as a Base Model Using the Unbalanced Version of the Dataset

The Xception model is a moderate-size model with a number of trainable parameters which is bigger than VGG16 but less than NasNetLarge. In this scenario, Xception is used as a base model to extract image features. The output of the Xception model is a feature vector of 2048 samples. Figure 4.6 shows the architecture, the output size, and the number of trainable parameters of the Xception-based model.

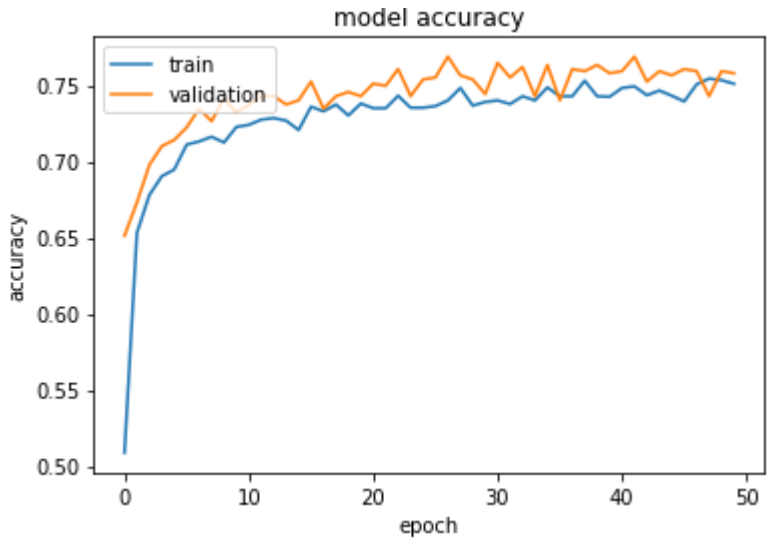
Layer (type)	Output Shape	Param #
xception (Functional)	(None, 2048)	20861480
dropout_4 (Dropout)	(None, 2048)	0
flatten_2 (Flatten)	(None, 2048)	0
dense_6 (Dense)	(None, 64)	131136
dropout_5 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 32)	2080
dense_8 (Dense)	(None, 5)	165

Total params: 20,994,861
Trainable params: 133,381
Non-trainable params: 20,861,480

Figure 4.6. Xception-based DL model for diabetic retinopathy detection.

The Xception base model weights are frozen, so the number of trainable parameters is computed for the classification part only (133381 parameters).

Figure 4.7 shows the training and validation accuracy and loss.



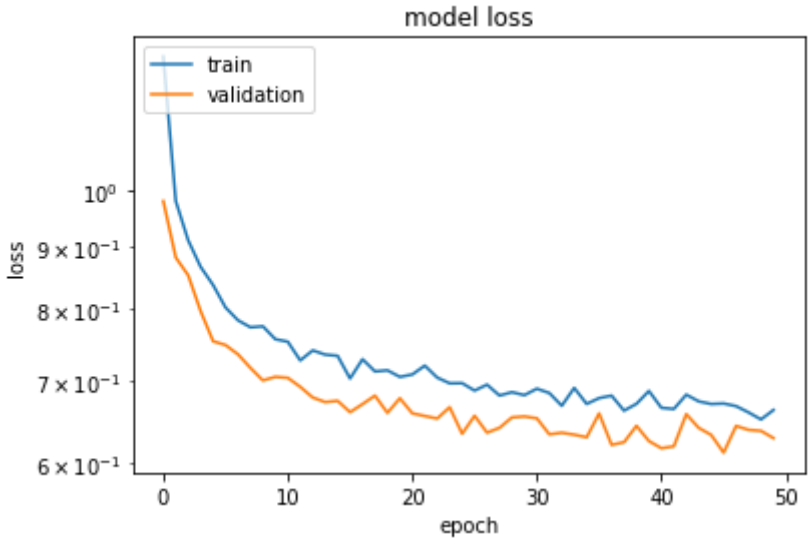


Figure 4.7. Training and validation accuracy and loss curves of the Xception based DL model of diabetic retinopathy detection

Results of Figure 4.7 shows that the validation accuracy is almost 76.88% and the validation loss is 0.6356. While the training accuracy is 74% and the training loss is 0.6952. The average training time is 49s/step (The Xception model requires a similar computational time comparing to the VGG16 model).

The detailed results, including the precision, recall, and F1-score, are illustrated in Table 4.3.

Table 4.3. Precision, Recall, and F1-score of the Xception-based DL model of diabetic retinopathy detection.

Xception	**Precision %**	**Recall %**	**F1-score %**	**Num. of test samples**
No_DR	94	98	96	361
Mild	56	39	46	74
Moderate	56	84	68	199
Severe	0	0	0	38
Proliferate_DR	60	5	9	59
Macro avg	53	45	44	731
Weighted avg	72	76	71	731

Table 4.3 conclude the same observation of VGG and NasNetLarge models.

4.3.6. Results of training InceptionV3 as a Base Model Using the Unbalanced Version of the Dataset

The InceptionV3 model is a moderate-size model with a number of trainable parameters which is bigger than VGG16, less than NasNetLarge and similar to Xception model. In this scenario, InceptionV3 is used as a base model to extract image features. The output of InceptionV3 model is a feature vector of 2048 samples. Figure 4.8 shows the architecture, the output size and number of trainable parameters of the InceptionV3-based model.

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 2048)	20861480
dropout_4 (Dropout)	(None, 2048)	0
flatten_2 (Flatten)	(None, 2048)	0
dense_6 (Dense)	(None, 64)	131136
dropout_5 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 32)	2080
dense_8 (Dense)	(None, 5)	165

Total params: 20,994,861
Trainable params: 133,381
Non-trainable params: 20,861,480

Figure 4.8. InceptionV3-based DL model for diabetic retinopathy detection

The InceptionV3 base model weights are frozen so that the number of trainable parameters is computed for the classification part only (133381 parameters) which is the same as for the Xception model.

Figure 4.9 shows the training and validation accuracy and loss.

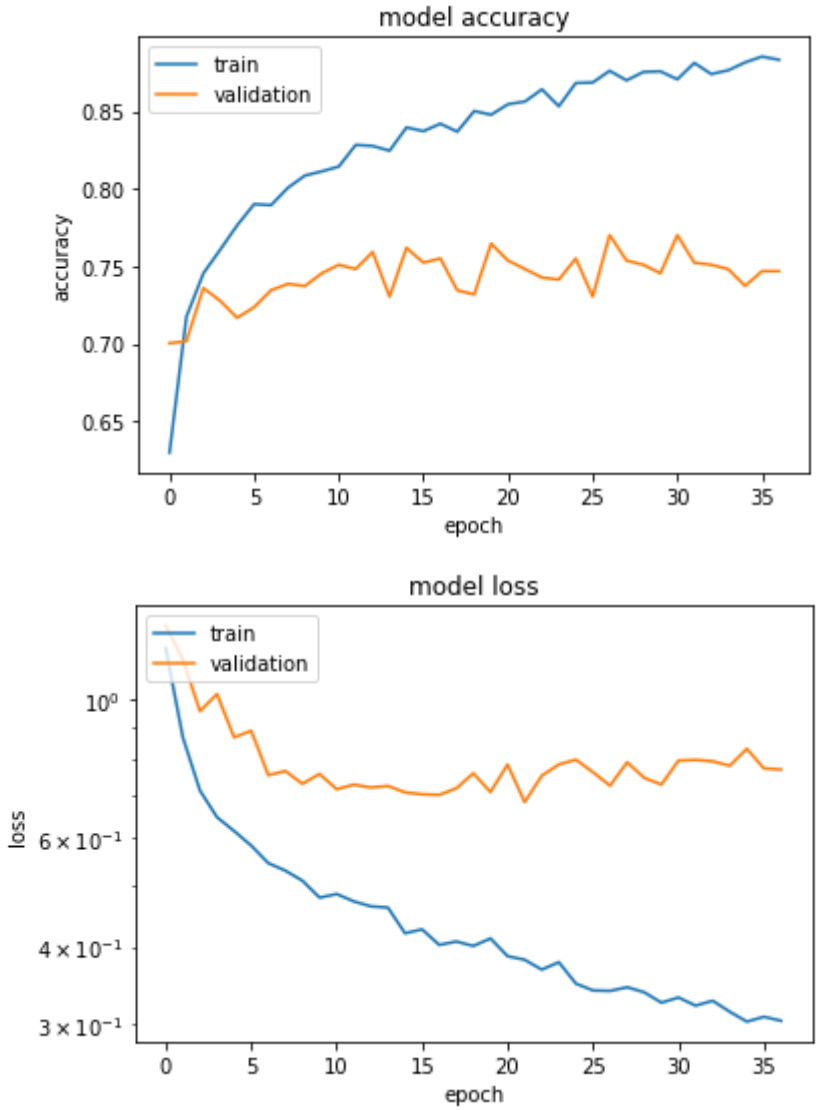


Figure 4.9. Training and validation accuracy and loss curves of the InceptionV3 based DL model of diabetic retinopathy detection.

Results of Figure 4.9 show that the validation accuracy is almost 74% and the validation loss is 0.78. While the training accuracy is 70% and the training loss is 0.89. The average training time is 48s/step (InceptionV3 model requires a similar computational time comparing to VGG16 model).

The detailed results, including the precision, recall and F1-score are illustrated in Table 4.4.

Table 4.4. Precision, Recall, and F1-score of the InceptionV3 based DL model of diabetic retinopathy detection.

InceptionV3	Precision %	Recall %	F1-score %	Num. of test samples
No_DR	95	97	96	361
Mild	45	41	43	74
Moderate	60	72	65	199
Severe	21	13	16	38
Proliferate_DR	40	20	27	59
Macro avg	52	49	49	731
Weighted avg	72	74	72	731

Table 4.4 conclude the same observation of VGG, Xception and NasNetLarge models. However, the result of this model is the worst.

4.3.7. Results of Training an Ensemble of Deep Models Using the Unbalanced Version of the Dataset

To improve the performance of diabetic retinopathy system, the ensemble learning model is proposed. Table 4.5 includes the detailed results of training an ensemble learning model including the previous trained models (VGG16, NasNetLarge, Xception and InceptionV3).

Table 4.5. Precision, Recall, and F1-score of the ensemble DL model of diabetic retinopathy detection.

Model	Accuracy %	Precision %	Recall %	F1-score %
Ensemble	89.93	90.64	90.65	90.65
Best Individual model (NasNetLarge)	77	72	77	72

Table 4.5 proved that the ensemble model achieved the best performance. Comparing to the best results, the ensemble model enhanced the accuracy, the precision, the recall, and the F1-score by 12.93%, 18.64%, 13.65% and 18.65%, respectively.

As a result, we can conclude that the ensemble model reduced the minor classes errors and improved the performance. However, the performance can also be enhanced using the balancing technology by which the minor classes samples will be oversampled and the balanced will be partially retrieved.

4.4. BALANCED TRAINING RESULTS

Figure 4.10 illustrates the distribution over the balanced dataset. The original dominant class (which is the 'NO DR' class) has a lower percentage comparing to the original distribution. Besides, the minor classes samples are oversampled.

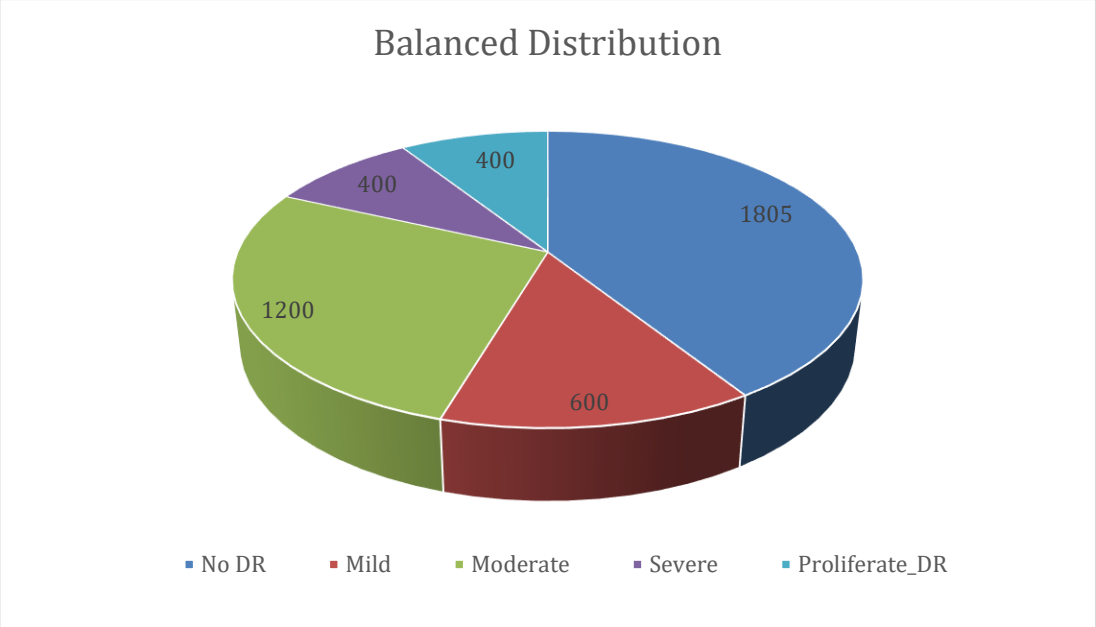


Figure 4.10. Balanced dataset distribution

The dataset is split using 75% for training and 25% for validation and test. In this scenario, the dataset split is different since the number of images are increased in terms of balancing. The following transformations are applied:

- Convert all images into RGB
- Resize all images into 150*150 in order to reduce the training time. The images are resized into a smaller size 150*150 instead of 224*224 since the balancing

operation will increase the number of images and this will increase the training time.

- Convert the targets to categorical form in order to compute probabilities for each one in the training step (the class with the highest probability will be the predicted class).
- Shuffle training images in order to prevent the training process from memorizing the order of the training samples.

4.4.1. Training Parameters (Training Options)

For this scenario, the same training options of the unbalanced scenario is used.

4.4.2. Training Scenarios

In all training scenarios, the used model consists of two main parts; the feature extraction part and the classification part.

In the feature extraction part, many DL models are suggested, while the classification part is fixed for all models.

Many training scenarios are proposed including the following:

- Training the VGG-16-based model using the balanced version of the dataset.
- Training the VGG-19-based model using the balanced version of the dataset.
- Training the Xception-based model using the balanced version of the dataset.
- Training the EfficientNet-based model using the balanced version of the dataset.
- Make an ensemble of the models (1-4) using the balanced version of the dataset.

For these scenarios, we added another enhancement rather than the balancing process. The base models are re-trained using our training dataset to get a better performance (In previous scenario of unbalanced dataset, the base models weights are frozen).

4.4.3. Results of Training VGG-16 as a Base Model Using the Unbalanced Version of the Dataset

In this scenario, the VGG16 model is used as the base model in order to extract image features. The output of VGG16 model is a feature vector of size 512. Figure 4.11 shows the architecture, the output size and number of trainable parameters of the VGG16-based model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 512)               14714688

 dropout_2 (Dropout)         (None, 512)               0

 flatten_1 (Flatten)         (None, 512)               0

 dense_3 (Dense)             (None, 64)                32832

 dropout_3 (Dropout)         (None, 64)                0

 dense_4 (Dense)             (None, 32)                2080

 dense_5 (Dense)             (None, 5)                 165

=================================================================
Total params: 14,749,765
Trainable params: 14,749,765
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.11. VGG16-based DL model for diabetic retinopathy detection.

The VGG base model weights are not frozen so that the number of trainable parameters is computed for both the base and the classification parts (14749765 parameters).

Figure 4.12 shows the training and validation accuracy and loss.

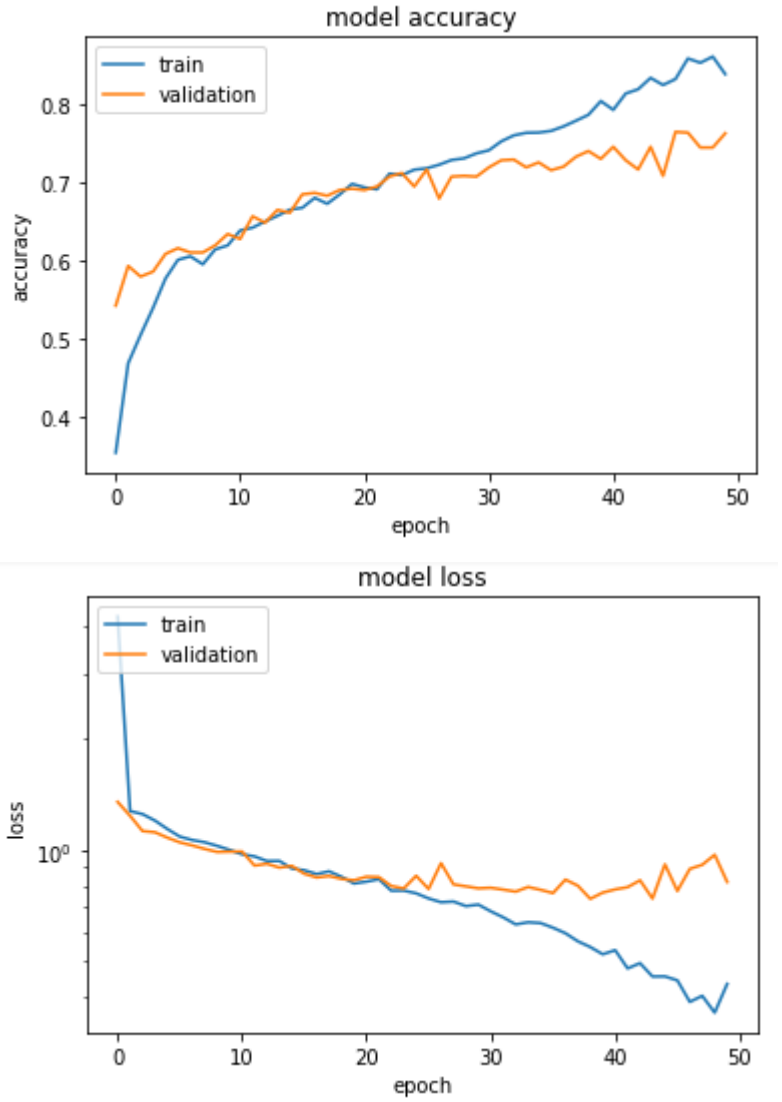


Figure 4.12. Training and validation accuracy and loss curves of the VGG16 based DL model of diabetic retinopathy detection (balanced version)

Results of this scenario observed by Figure 4.12 show that the validation accuracy is 76.41% and the validation loss is 0.7757. While the training accuracy is 83.17% and the training loss is 0.4452. The average training time is 22s/step.

The detailed results, including the precision, recall and F1-score are illustrated in Table 4.6.

Table 4.6. Precision, Recall, and F1-score of the VGG based DL model of diabetic retinopathy detection.

VGG16	Precision %	Recall %	F1-score %	Num. of test samples
No_DR	93	95	94	443
Mild	66	79	72	311
Moderate	63	74	68	152
Severe	74	49	59	100
Proliferate_DR	45	16	23	96
Macro avg	68	62	63	1102
Weighted avg	75	76	75	1102

Table 4.6 shows that there is no 0% result since the classes are balanced. However, the accuracy is not good and needs improvement.

4.4.4. Results of Training VGG-19 as a Base Model Using the Unbalanced Version of the Dataset

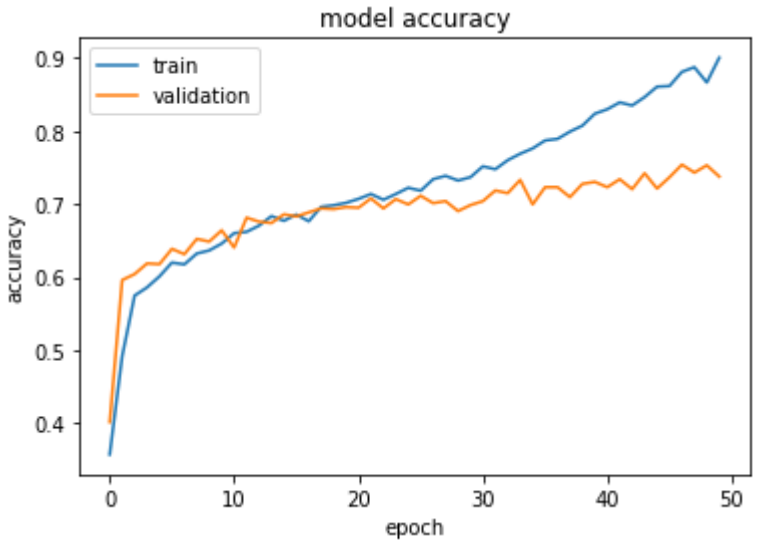
In this scenario, the VGG19 model is used as the base model in order to extract image features. The output of VGG19 model is a feature vector of size 512. The VGG19 model is deeper than VGG16 so the number of trainable parameters is higher (since VGG19 has three more Conv layers than VGG16). Figure 4.13 shows the architecture, the output size and number of trainable parameters of the VGG19-based model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 512)               20024384

 dropout_6 (Dropout)         (None, 512)               0

 flatten_3 (Flatten)         (None, 512)               0

 dense_9 (Dense)             (None, 64)                32832

 dropout_7 (Dropout)         (None, 64)                0

 dense_10 (Dense)            (None, 32)                2080

 dense_11 (Dense)            (None, 5)                 165

=================================================================
Total params: 20,059,461
Trainable params: 20,059,461
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.13. VGG19-based DL model for diabetic retinopathy detection.

The VGG19 base model weights are not frozen so that the number of trainable parameters is computed for both the base and the classification parts (20059461 parameters).

Figure 4.14 shows the training and validation accuracy and loss.



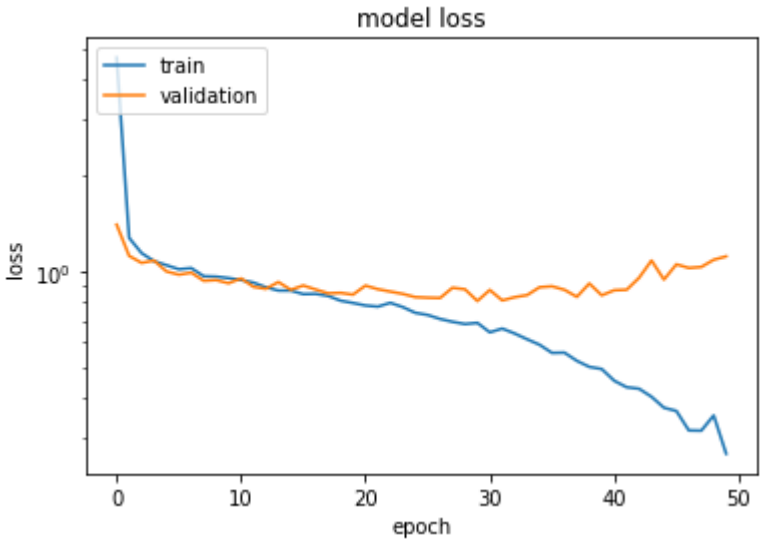


Figure 4.14. Training and validation accuracy and loss curves of the VGG16 based DL model of diabetic retinopathy detection (balanced version).

Results of this scenario observed by Figure 4.14 show that the validation accuracy is 75.32% and the validation loss is 1.08. While the training accuracy is 86.65% and the training loss is 0.352. The average training time is 25.5s/step.

The detailed results, including the precision, recall and F1-score are illustrated in Table 4.7.

Table 4.7. Precision, Recall, and F1-score of the VGG19 based DL model of diabetic retinopathy detection.

VGG19	**Precision %**	**Recall %**	**F1-score %**	**Num. of test samples**
No_DR	92	95	94	443
Mild	69	69	69	311
Moderate	59	69	63	152
Severe	72	58	64	100
Proliferate_DR	43	33	37	96
Macro avg	67	65	66	1102
Weighted avg	75	75	75	1102

Table 4.7 shows that there is no 0% result since the classes are balanced. However, the accuracy is not good and needs improvement. VGG19 model demonstrates a similar performance to the VGG16 model. However, VGG16 results are better than VGG19.

4.4.5. Results of Training Xception as a Base Model Using the Unbalanced Version of the Dataset

In this scenario, the Xception model is used as the base model in order to extract image features. The output of Xception model is a feature vector of size 2048 which is higher than VGG and VGG19 output size. The Xception model is deeper than VGG so the number of trainable parameters is higher (since Xception has 71 layers deep). Figure 4.15 shows the architecture, the output size and number of trainable parameters of the Xception-based model.

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 2048)	20861480
dropout (Dropout)	(None, 2048)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 64)	131136
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
dense_2 (Dense)	(None, 5)	165

Total params: 20,994,861
Trainable params: 20,940,333
Non-trainable params: 54,528

Figure 4.15. Xception-based DL model for diabetic retinopathy detection.

The Xception base model weights are not frozen so that the number of trainable parameters is computed for both the base and the classification parts (20994861 parameters). This number of parameters is higher than VGG16 and VGG19 parameters. Figure 4.16 shows the training and validation accuracy and loss of Xception model on the balanced dataset.

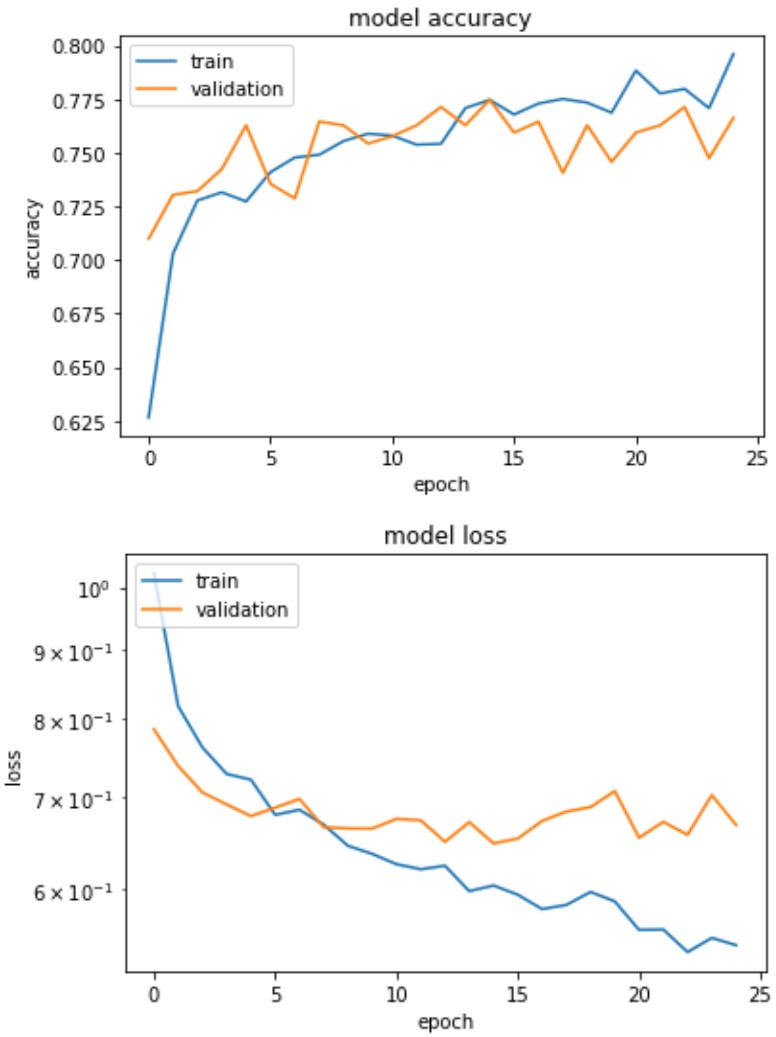


Figure 4.16. Training and validation accuracy and loss curves of the Xception based DL model of diabetic retinopathy detection (balanced version).

Figure 4.16 illustrates that the validation accuracy is 82.49% and the validation loss is 1.03 which is better than all other previous models. Similarly, the training accuracy is 97.97% and the training loss is 0.069. The average training time is 24s/step.

The detailed results, including the precision, recall and F1-score are illustrated in Table 4.8.

Table 4.8. Precision, Recall, and F1-score of the Xception based DL model of diabetic retinopathy detection.

Xception	**Precision %**	**Recall %**	**F1-score %**	**Num. of test samples**
No_DR	96	97	97	443
Mild	67	91	77	311
Moderate	86	66	75	152
Severe	81	65	72	100
Proliferate_DR	83	31	45	96
Macro avg	83	70	73	1102
Weighted avg	84	82	81	1102

Table 4.8 shows that the performance of this model is better than all previous models since the validation accuracy is 82%, the precision 84%, the recall 82% and F1-score is 81% which are all better than results of previous models. However, some classes like "Proliferate_DR" need enhancement.

4.4.6. Results of Training EfficientNetB3 as a Base Model Using the Unbalanced Version of the Dataset

In this scenario, the EfficientNet model is used as the base model in order to extract image features. The output of EfficientNet model is a feature vector of size 1536 which is higher than VGG and VGG19 output size but less than Xception output size. The EfficientNet model is deeper than VGG but less than Xception.

Figure 4.17 shows the architecture, the output size and number of trainable parameters of the EfficientNet-based model.

Layer (type)	Output Shape	Param #
efficientnetb3 (Functional)	(None, 1536)	10783535
dropout_4 (Dropout)	(None, 1536)	0
flatten_2 (Flatten)	(None, 1536)	0
dense_6 (Dense)	(None, 64)	98368
dropout_5 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 32)	2080
dense_8 (Dense)	(None, 5)	165

Total params: 10,884,148
Trainable params: 10,796,845
Non-trainable params: 87,303

Figure 4.17. Xception-based DL model for diabetic retinopathy detection

The EfficientNet base model weights are not frozen so that the number of trainable parameters is computed for both the base and the classification parts (10796845 parameters). This number of parameters is less than all previous models and this is why this model called an efficient model.

Figure 4.18 shows the training and validation accuracy and loss of EfficientNet model on the balanced dataset.

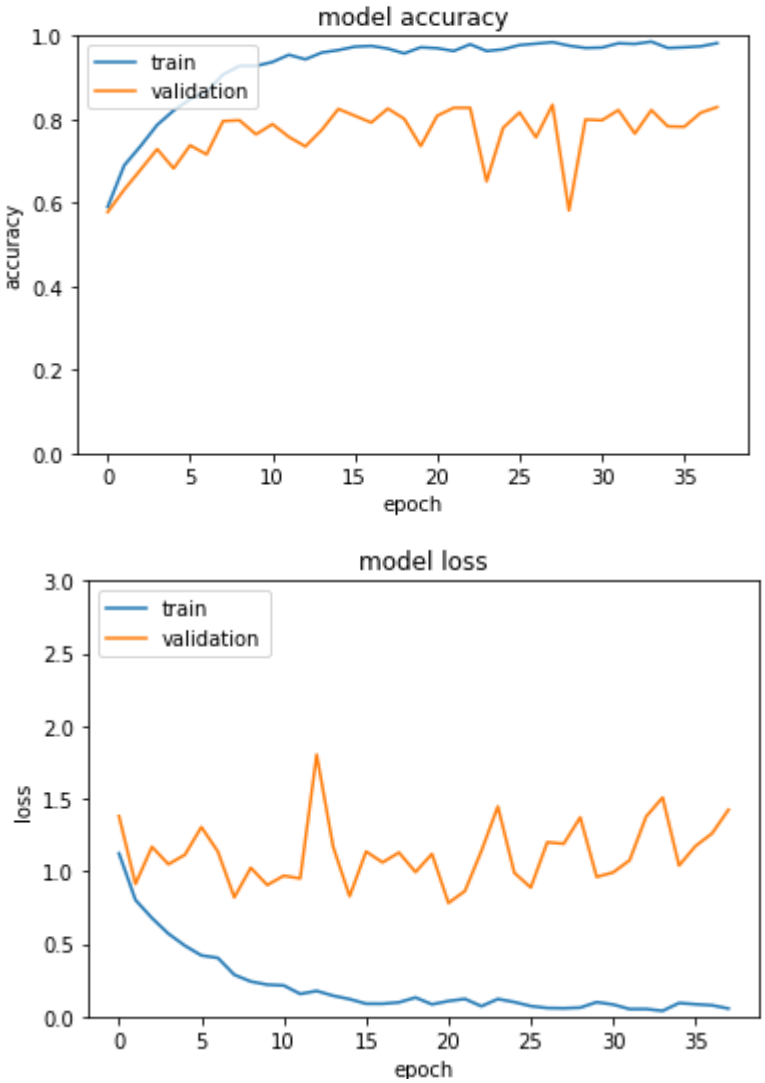


Figure 4.18. Training and validation accuracy and loss curves of the EfficientNet based DL model of diabetic retinopathy detection (balanced version)

Figure 4.18 shows that the validation accuracy is 83.48% and the validation loss is 1.18 which is better than all other previous models. Similarly, the training accuracy is 98.49% and the training loss is 0.056. The average training time is 26s/step. The detailed results, including the precision, recall and F1-score are illustrated in Table 4.9.

Table 4.9. Precision, Recall, and F1-score of the EfficientNet based DL model of diabetic retinopathy detection.

EfficientNet	Precision %	Recall %	F1-score %	Num. of test samples
No_DR	97	97	97	443
Mild	72	87	79	311
Moderate	74	82	78	152
Severe	82	61	70	100
Proliferate_DR	78	38	51	96
Macro avg	81	73	75	1102
Weighted avg	84	83	83	1102

Table 4.8 shows that the performance of this model is better than all previous models since the validation accuracy is 82%, the precision 84%, the recall 83% and F1-score is 83% which are all better than results of previous models. However, some classes like " Proliferate_DR " need enhancement.

4.4.7 Results of the Ensemble Model (Balanced Version of the Dataset)

An ensemble of all previous models is created and the weighted average method is used to get the final score of the ensemble. Table 4.10 includes the detailed performance calculations of the proposed ensemble model. The validation accuracy is 92% which is better than the validation accuracy of all previous models.

Table 4.10. Precision, Recall, and F1-score of the ensemble-based DL model of diabetic retinopathy detection.

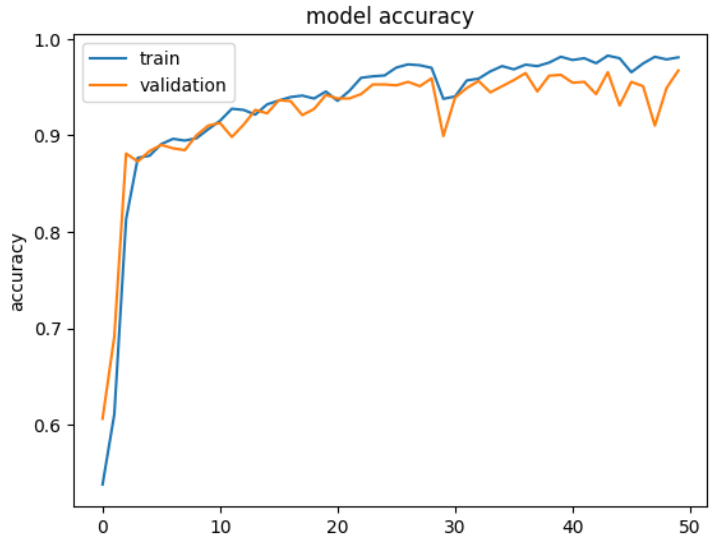
Model	Accuracy %	Precision %	Recall %	F1-Score %
Ensemble	92	92	92	91
Best Model	82	84	83	83

Ensemble model has the best performance against all previous individual models. Comparing to the best model (EfficientNet), the accuracy is increased by 10%, the precision is increased by 8%, the recall is increased by 9% and F1-score is increased by 8%, respectively.

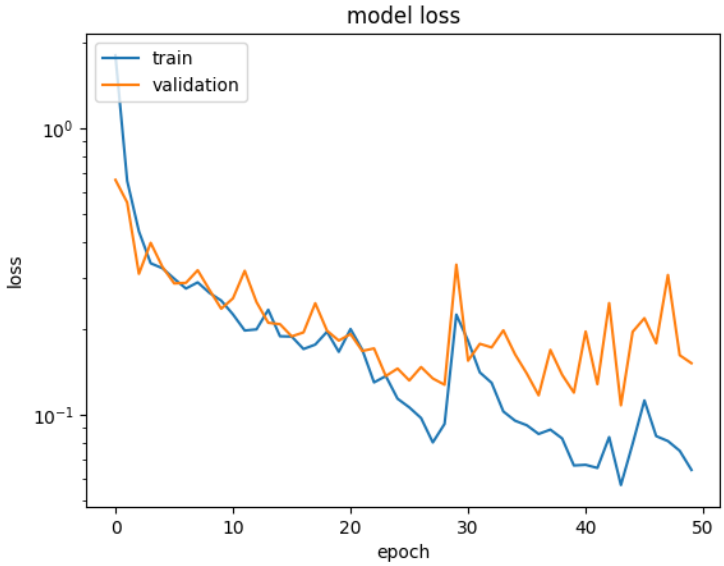
The performance of the worst class "Proliferate DR" is also improved comparing to all previous individual models.

4.5. BINARY-CLASS DIABETIC RETINOPATHY DETECTION SCENARIO

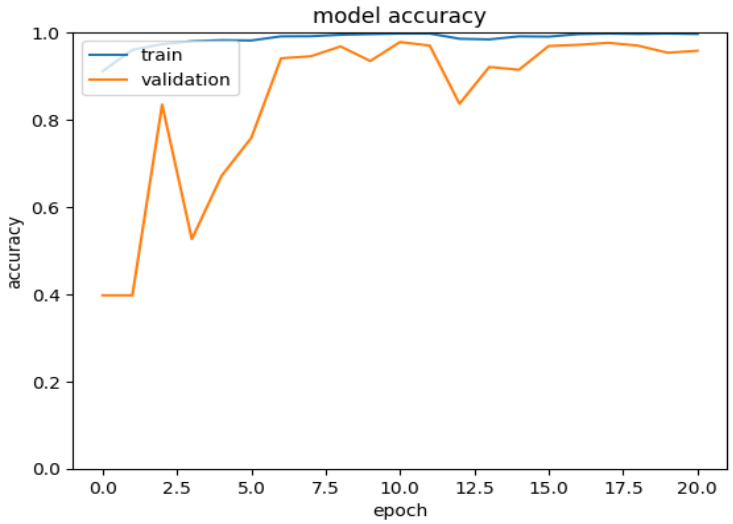
In this scenario, the labels are grouped into only two categories (DR or NO_DR). Based on this modification, the VGG-16, Xception, and EfficientNet models are re-trained in terms of this modification. Figure 4.19 includes the training and validation curves of the trained models. Besides, the ensemble model is built again, and the results are concluded in Table 4.12.



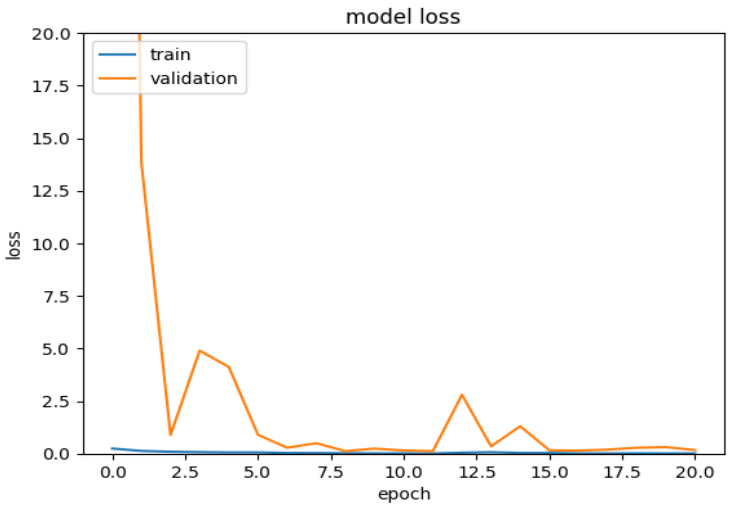
A.



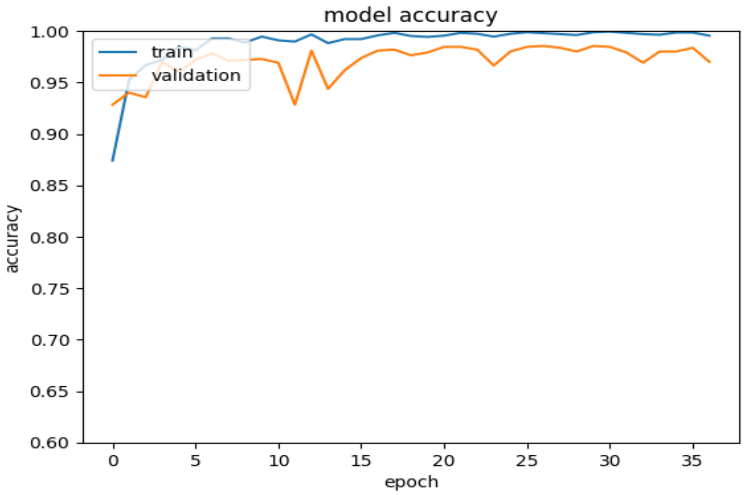
B.


model accuracy
train
validation
accuracy
epoch
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

C.


model loss
train
validation
loss
epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

D.


model accuracy
train
validation
accuracy
epoch
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
5
10
15
20
25
30
35

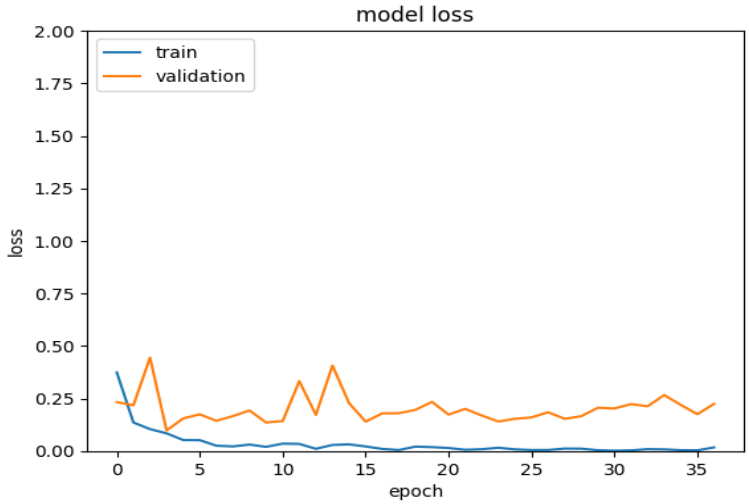


Figure 4.19. Training and validation accuracy of the trained models (A,B: VGG-16, C,D: Xception, E,F: EfficientNet) for the binary-class classification problem.

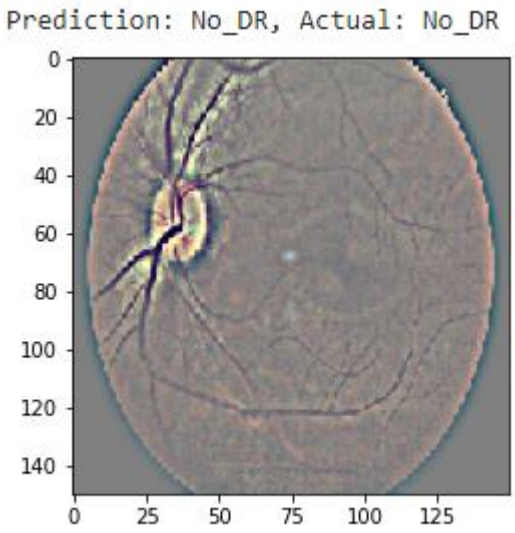
Table 4.11. Precision, Recall, and F1-score of the best individual and ensemble-based DL models of diabetic retinopathy detection system based on both balanced and unbalanced version of the dataset.

Model	Accuracy %	Precision %	Recall %	F1-score %
VGG-16	98.46	98.46	98.46	98.4
Xception	97.01	97	97	97
EfficientNet	99.27	99.28	99.27	99.27
Ensemble	**99.46**	**99.46**	**99.46**	**99.46**

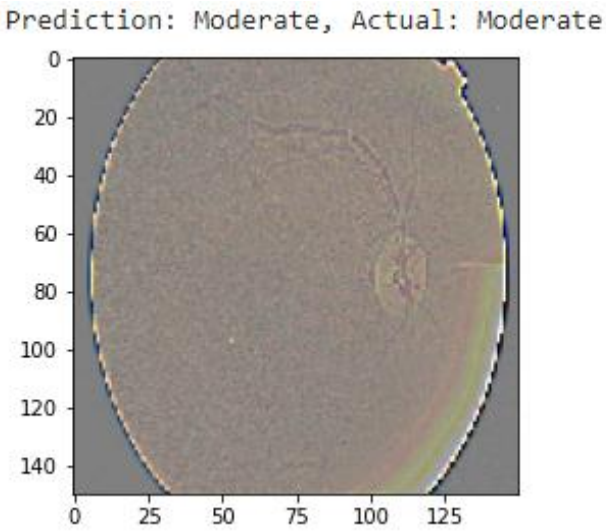
Results of Table 4.11 shows that the best accuracy corresponds to the ensemble model with 99.46%. The accuracy is enhanced by 0.19% compared to the best model (EfficientNet).

4.6. TEST SOME SAMPLES

In order to show the output of evaluating the ensemble model after using some test samples, Figure 4.20 shows some evaluation results of multi-class case while Figure 4.21 includes examples of testing the binary class classification ensemble model.

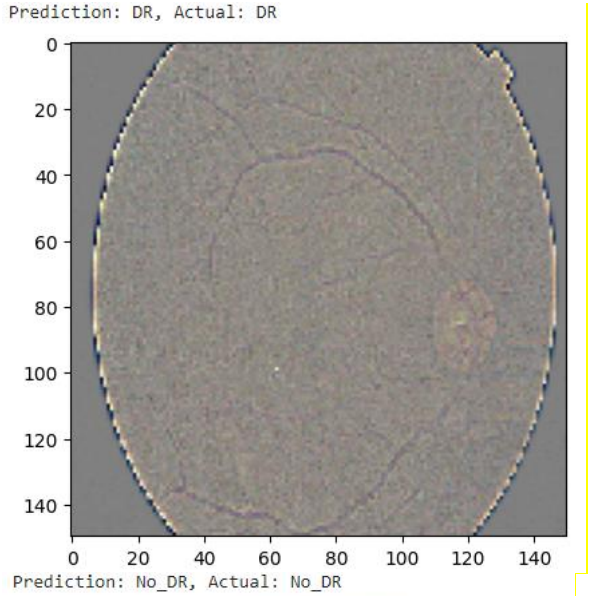


The Actual Class is No_DR and the ensemble model predicted it as No_DR

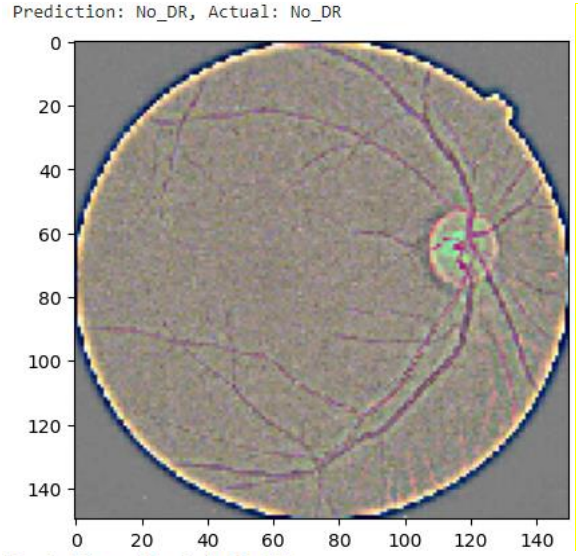


The Actual Class is moderate and the ensemble model predicted it as moderate.

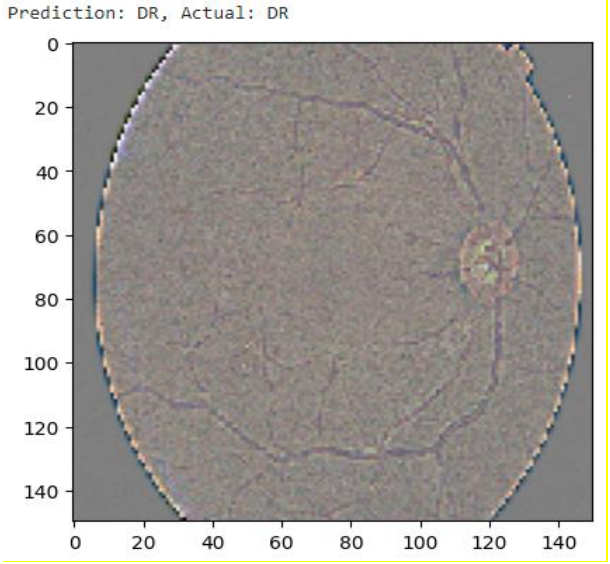
Figure 4.20. Some evaluation results of multi-class scenario.



The Actual Class is DR, and the ensemble model predicted it as DR



The Actual Class is No_DR and the ensemble model predicted it as No_DR



The Actual Class is DR and the ensemble model predicted it as DR.

Figure 4.21. Some evaluation results of binary-class scenario.

4.7. DISCUSSION OF THE RESULTS

An analysis and detailed comparison of the results of the used deep learning models of diabetic retinopathy detection will be discussed in this section. The proposed scenarios of using both unbalanced and balanced versions of the dataset will also be compared.

The results of the ensemble DL model on the unbalanced dataset show an accuracy of 89.93% and F1-score of 90.65%. These performance metrics are higher than the best individual model (NasNetLarge with accuracy: 77%, F1-score: 72%). Tables 4.8, 4.9, and 4.10 include the results of the proposed DL models trained on the balanced version of the dataset. The main observation here is that the ensemble model consistently achieves the best performance exceeding the performance of the Xception and EfficientNet individual models in terms of accuracy, precision, recall, and F1-score.

The ensemble model achieves 98% precision, recall, and F1-score in the case of the category "No-DR." The results of class "Proliferate_DR" are the worst ones in all scenarios. However, the results of the individual models of this category is completely less than the corresponding results of the ensemble model (since the ensemble model achieves a precision of 98%, a recall of 62%, and an F1-socre of 76% for this class). Compared to the Xception-based model, the "Proliferate" class has lower performance metrics (precision: 83%, recall: 31%, and F1-score: 45%). Similarly, the EfficientNet-based model achieves a precision of 78%, a recall of 38% and an F1-score of 38% % for the "Proliferate" class.

The rest of classes, including "Mild", "Moderate", and "Severe", achieve better performance in case of using ensemble model. A significant improvement in performance is observed. For the case of "Mild" class, the ensemble model achieves an F1-score of 88%, compared to 77% in the Xception model and 79% in the EfficientNet model. While for the case of "Moderate" class, the ensemble model scores an F1-score of 88%, compared to 75% in the Xception model and 78% in the EfficientNet model.

An F1-score of 91% of the "Severe" class is registered compared to only 72% and 70% of the same class in the Xception and EfficientNet models, respectively.To conclude, the ensemble-based deep model of diabetic retinopathy system achieves the best performance exceeding all other individual models.This result is observed in case of all classes and in term of all performance metrics. For the case of the best and worst classes (No_DR and Proliferate_DR), the ensemble model also outperformed all individual models. The final observed result is that the balanced dataset improves the performance of the diabetic retinopathy system. The concluded results are illustrated in Table 4.11.

Table 4.11. Precision, Recall, and F1-score of the best individual and ensemble-based DL models of diabetic retinopathy detection system based on both balanced and unbalanced version of the dataset.

Model	**Precision %**	**Recall %**	**F1-score %**	**Num. of test samples**
NasNetLarge	72	77	72	731
Xception (Balanced)	84	82	81	1102
EfficientNet (Balanced)	97	97	97	443
Ensemble (Unbalanced)	77	72	77	731
Ensemble (Balanced)	92	92	91	1102

A comparison between the current study and previous ones in the field of diabetic retinopathy is illustrated in Table 5.1.

Table 4.12. Comparison between the current study and related works.

Researcher	Methodology	Dataset	Main Results		Limitations
Soniya et al. [38]	CNN single-based and CNN heterogeneous-based	DIARETDB0 (130)	Accuracies ranging from 42.5% to 95%		Limited dataset size
Lam et al. [40]	GoogleLeNet-v1, AlexNet, VGG-16, ResNet, Inception-V3	Kaggle EyePACS (243 images)	Best accuracy: 98% (InceptionV3)		Binary classification (DR or Not DR)
Pour et al. [44]	EfficientNet B5	MESSIDOR, MESSIDOR-2, IDRiD	AUC: 0.94 (MESSIDOR), 0.93 (IDRiD)		Binary classification (DR or Not DR)
Thota and Reddy [45]	VGG-16	Kaggle EyePACS	Accuracy: 74%, Sensitivity: 80%, Specificity: 65%		Low accuracy
Parthasharathi et al. [48]	CNN	Kaggle (1,000)	Accuracy: 91.5%		Binary classification (DR or Not DR)
Raju et al. [56]	Modified U-Net	IDRiD	Sensitivity: 95.6%, Specificity: 98.6%		Limited dataset size
Chudzik et al. [58]	VGG-16, VGG-19, Inception-v3, Inception-ResNet-v2, and Xception	EyePACS (243 images)	Best accuracy: 96.8% (Inception-ResNet-v2)		Low dataset size/ Binary classification
Current Study	VGG-16 VGG-19 NasNetLarge, Inception, Xception, EfficientNet, Ensemble learning	Kaggle Dataset	Multi-class	Best accuracy 92%, Preiciosn 92%, recall 92%, and F1-score 91%	Limited dataset size
			Binary -Class	Accuracy, precision, recall, and F1-score: 99.46%	

To conclude, the current study applied ensemble learning with the balancing (over-sampling) approach to achieving the best performance. The current study outperformed all previous studies. However, some studies have good accuracy, like Chudzik et al. [58], Raju et al. [56], and Lam et al. [40], but all of these studies didn't take into account the different stages of diabetic retinopathy disease (they used binary classification). Some studies used lower data sizes [56] ,[58], [38]. Although Pratt et al. [37] study demonstrated a high specificity but the sensitivity and accuracy were low. In terms of binary classification, the current study outperformed all similar studies.

PART 5

CONCLUSION, FUTURE WORK

5.1. CONCLUSION

In the current study, a new diabetic retinopathy detection system is introduced. The study consisted of many steps. In the first one, an image dataset was acquired. The images included different cases of the disease (severe, mild, moderate, and proliferate) besides the benign (normal) cases. The main problem of this dataset is that the normal case contains most of the dataset images. To solve this problem, an over-sampling technique (SMOTE algorithm) was used. By using the over-sampling approach, the dataset became balanced, and the minor class's samples were increased. Besides the balancing step, the data preprocessing was used to resize images to a lower size in order to minimize the training time. Data augmentation was also used to increase the number of images and generate different images with specific variations (rescaling and flipping). The images values are also normalized by dividing the pixels' values by 255. For the training step, two main scenarios were proposed. The first is based on the unbalanced version of the dataset, while the second is the balanced scenario.

In the first scenario, the VGG-16, NasNet, Xception, and Inception models were selected as base models for the proposed detection systems. Besides, a classification part (flatten, dropout, and dense layers) was used. All models were trained using a training split of 75% of the dataset and evaluated using a validation split of 25% of the dataset. For the second scenario, VGG-16, VGG-19, EfficientNet, and Xception were used as base models, and the same previous classification part was also used. The experiments were obtained from all those scenarios. The results showed that the balanced version of the dataset achieved a better performance in all scenarios.

The ensemble model of the DL models of the balanced and unbalanced scenarios were also used. The results illustrated that the best individual model was the EfficientNet, while the best model was the ensemble (balanced) model with an accuracy of 92%.

Moreover, the binary-class prediction showed a better performance of 99.46% of the ensemble model 5.2. LIMITATIONS

Since the current study dealt with most of previous studies limitations (like lower dataset size, low accuracy, dealing with only two condition of disease), the limitations of this study is the use of only one specific dataset which will include specific categorizations and conditions.

5.3. FUTURE WORK

Based on the previous limitations, the future work and recommendation can be concluded as follows:

- Increase the number of dataset samples.
- Try other deep learning models.
- Apply some other different fusion techniques like data fusion or feature-level fusion in which many base models can be fused to produce a fusion feature vector then apply the classification part.
- Try other different classifiers in the classification part like using an SVM/k-NN classifier after the based model.

REFERENCES

[1] S. C. Sen, F. B. S., M. Iglarz and S. Chakrabarti, "Renal, re`tinal and cardiac changes in type 2 diabetes are attenuated by macitentan, a dual endothelin receptor antagonist," ***Life sciences,*** vol. 91, no. 13, pp. 658-668, 2012.

[2] R. Taylor and D. Batey, ***Handbook of retinal screening in diabetes: diagnosis and management., John Wiley & Sons***, 2012.

[3] International diabetes federation, "What is diabetes," 16 1 2023. [**Online**]. Available: https://www.idf.org/aboutdiabetes/what-is-diabetes.html. [Accessed 20 1 2023].

[4] diabetesatlas, "IDF Diabetes Atlas 2022 Reports," [**Online**]. Available: https://diabetesatlas.org/. [Accessed 20 1 2023].

[5] B. Mounirou, N. Adam, A. Yakoura, M. Aminou, Y. Liu and L. Tan, "Diabetic Retinopathy: An Overview of Treatments," ***Indian J Endocr Metab***, vol. 26, no. 2, pp. 111-118, 2022.

[6] R. Bourne, G. A. Stevens, R. A. White, J. L. Smith, S. R. Flaxman, H. Price and J. B. Jonas, "Causes of vision loss worldwide, 1990–2010: a systematic analysis," ***The lancet global health*** , vol. 1, no. 6, pp. 339-349, 2013.

[7] C. JM and S. AW, "Racial disparities in the screening and treatment of diabetic retinopathy," ***Journal of the National Medical Association***, vol. 114, no. 2, pp. 171-181, 2022.

[8] M. D. Saleh and C. Eswaran, "An automated decision-support system for non-proliferative diabetic retinopathy disease based on MAs and HAs detection," ***Computer methods and programs in biomedicine***, vol. 108, no. 1, pp. 186-196, 2012.

[9] W. L. Alyoubi, W. M. Shalash and M. F. Abulkhair, "Diabetic retinopathy detection through deep learning techniques: A review," ***Informatics in Medicine Unlocked***, vol. 20, 2020.

[10] A. Masood, B. Sheng, P. Li, X. Hou, X. Wei, J. Qin and D. Feng, "Computer-assisted decision support system in pulmonary cancer detection and stage classification on CT images," ***Journal of biomedical informatics***, vol. 79, pp. 117-128, 2018.

[11] L. Guariguata, D. R. Whiting, I. Hambleton, J. Beagley, U. Linnenkamp and J. E. Shaw, "Global estimates of diabetes prevalence for 2013 and projections for 2035," ***Diabetes research and clinical practice*** , vol. 103, no. 2, pp. 137-149, 2014.

[12] P. H. Scanlon, A. Sallam and P. V. Wijngaarden, A practical manual of diabetic retinopathy management, John Wiley & Sons, 2017.

[13] A. Arrigo, M. Teussink, E. Aragona, F. Bandello and M. B. Parodi, "MultiColor imaging to detect different subtypes of retinal microaneurysms in diabetic retinopathy," ***Eye*** , vol. 1, pp. 277-281, 2021.

[14] M. Dubow, A. Pinhas, N. Shah, R. Cooper, A. Gan, R. Gentile and V. Hendrix, "Classification of human retinal microaneurysms using adaptive optics scanning light ophthalmoscope fluorescein angiography," ***Investigative ophthalmology & visual science***, vol. 55, no. 3, pp. 1299-1309, 2014.

[15] A. Skouta, A. Elmoufidi, S. Jai-Andaloussi and O. Ouchetto, "Hemorrhage semantic segmentation in fundus images for the diagnosis of diabetic retinopathy by using a convolutional neural network," ***Journal of Big Data*** *volume*, vol. 9, no. 1, pp. 1-24, 2022.

[16] S. Guo, "LightEyes: A Lightweight Fundus Segmentation Network forMobile Edge Computing," ***Sensors***, vol. 22, pp. 1-21, 2022.

[17] D. Das, S. Biswas, S. Bandyopadhyay and S. Sarkar, "Early Detection of Diabetic Retinopathy Using Machine Learning Techniques: A Survey on Recent Trends and Techniques," **in** ***Lecture Notes in Electrical Engineering book series***, 2020.

[18] P. Porwal, S. Pachade, R. Kamble, M. Kokare, G. Deshmukh, V. Sahasrabuddhe and F. Meriaudeau, "Indian Diabetic Retinopathy Image Dataset (IDRiD): A Database for Diabetic Retinopathy Screening Research," *data*, vol. 3, no. 3, 2018.

[19] M. Chetoui, M. Akhloufi and M. Kardouchi, "Diabetic Retinopathy Detection Using Machine Learning and Texture Features," in ***IEEE Canadian Conference on Electrical & Computer Engineering***, 2018.

[20] R. Senapati, "Bright lesion detection in color fundus images based on texture features," ***Bulletin of Electrical Engineering and Informatics*** , vol. 5, no. 1, pp. 92-100, 2016.

[21] E. Carrera, A. González and R. Carrera, "Automated detection of diabetic retinopathy using SVM," in ***IEEE XXIV international conference on electronics, electrical engineering and computing***, Cusco, Peru, 2017.

[22] M. Hardas, S. Mathur, A. Bhaskar and M. Kalla, "Retinal fundus image classification for diabetic retinopathy using SVM predictions," ***Physical and Engineering Sciences in Medicine***, vol. 45, p. 781–791, 2022.

[23] E. Z. Aziza, L. M. E. Amine, M. Mohamed and B. Abdelhafid, "Decision tree CART algorithm for diabetic retinopathy classification," in ***International Conference on Image and Signal Processing and their Applications (ISPA),*** **Mostaganem**, Algeria, 2019.

[24] H. Yao, S. Wu, Z. Zhan and Z. Li, "A Classification Tree Model with Optical Coherence Tomography Angiography Variables to Screen Early-Stage Diabetic Retinopathy in Diabetic Patients," ***Journal of Ophthalmology***, no. Special Issue, 2022.

[25] R. Casanova, S. Saldana, E. Y. Chew, R. P. Danis, C. M. Greven and W. T. Ambrosius, "**Application of Random Forests Methods to Diabetic Retinopathy Classification Analyses**," *PLOS one*, vol. 9, no. 6, 2014.

[26] F. Alzami, R. Abdussalam, A. Megantara, A. Zainul and F. Purwanto, "Diabetic Retinopathy Grade Classification based on Fractal Analysis and Random Forests," in ***International Seminar on Application for Technology of Information and Communication (iSemantic)***, 2019.

[27] N. ZAABOUB and A. DOUIK, "Early Diagnosis of Diabetic Retinopathy using Random Forest Algorithm," in ***International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)***, Sousse, Tunisia, 2020.

[28] Y. Kang, Y. Fang and X. Lai, "Automatic Detection of Diabetic Retinopathy with Statistical Method and Bayesian Classifier," ***Journal of Medical Imaging and Health Informatics***, vol. 10, no. 5, pp. 1225-1233, 2020.

[29] R. Hadistio, H. Mawengkang and M. Zarlis, "Perbandingan Algoritma Stochastic Gradient Descent dan Naïve Bayes Pada Klasifikasi Diabetic Retinopathy," ***JURNAL MEDIA INFORMATIKA BUDIDARMA***, vol. 6, no. 1, 2022.

[30] S. Roychowdhury, D. D. Koozekanani and K. K. Parhi, "DREAM: Diabetic Retinopathy Analysis Using Machine Learning," ***IEEE Journal of Biomedical and Health Informatics*** , vol. 18, no. 5, pp. 1717-1728, 2014.

[31] G. T. Reddy, S. Bhattacharya, S. S. Ramakrishnan, C. L. Chowdhary and S. Hakak, "An Ensemble based Machine Learning model for Diabetic Retinopathy Classification," in ***International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)***, Vellore, India, 2020.

[32] N. Sikder, M. Masud, A. K. Bairagi, A. S. M. Arif, A.-A. Nahid and H. A. Alhumyani, "Severity classification of diabetic retinopathy using an ensemble learning algorithm through analyzing retinal images," ***Symmetry*** , vol. 13, no. 4, 2021.

[33] M. J. Pendekal and S. Gupta, "An Ensemble Classifier Based on Individual Features for Detecting Microaneurysms in Diabetic Retinopathy," ***Indonesian Journal of Electrical Engineering and Informatics (IJEEI)***, vol. 10, no. 1, pp. 60-71, 2022.

[34] A. Sopharak, B. Uyyanonvara, S. Barman and T. H. Williamson., "Automatic detection of diabetic retinopathy exudates from non-dilated retinal images using mathematical morphology methods," ***Computerized medical imaging and graphics*** , vol. 32, no. 8, pp. 720-727, 2008.

[35] K. Ganesan, R. J. Martis, U. R. Acharya, C. K. Chua, L. C. Min and A. Laude, "Computer-aided diabetic retinopathy detection using trace transforms on digital fundus images," ***Medical & biological engineering & computing***, vol. 52, pp. 663-672, 2014.

[36] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," ***Knowledge-based systems*** , vol. 60, pp. 20-27, 2014.

[37] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding and Y. Zheng, "Convolutional neural networks for diabetic retinopathy," ***Procedia computer science*** , vol. 90, pp. 200-205, 2016.

[38] S. Paul and L. Singh, "Heterogeneous modular deep neural network for diabetic retinopathy detection," in ***IEEE Region 10 Humanitarian Technology Conference***, 2016.

[39] R. Gargeya and T. Leng, "Automated Identification of Diabetic Retinopathy Using Deep Learning," ***Ophthalmology***, pp. 1-8, 2017.

[40] C. Lam, C. Yu, L. Huang and D. Rubin, "Retinal lesion detection with deep learning using image patches," ***Investigative ophthalmology & visual science***, vol. 59, no. 1, pp. 590-596, 2018.

[41] N. M. Khalifa, M. H. Taha and H. N. Mohamed, "Deep transfer learning models for medical diabetic retinopathy detection," ***Acta Informatica Medica***, vol. 27, no. 5, 2019.

[42] Q. Nguyen, R. Muthuraman and L. Singh, "Diabetic Retinopathy Detection using Deep Learning," in *4th* ***international conference on machine learning and soft computing***, 2020.

[43] B. Tymchenko, P. Marchenko and D. Spodarets, "Deep learning approach to diabetic retinopathy detection," ***arXiv preprint arXiv****:2003.02261* , 2020.

[44] A. M. Pour, H. Seyedarabi, S. Hassan, A. Jahromi and A. Javadzadeh, "Automatic detection and monitoring of diabetic retinopathy using efficient convolutional neural networks and contrast limited adaptive histogram equalization," ***IEEE Access***, vol. 8, pp. 136668-136673, 2020.

[45] N. Thota and D. Reddy, "Improving the accuracy of diabetic retinopathy severity classification with transfer learning," in *Proceedings of the* ***IEEE*** *63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Springfield, 2020.

[46] G. Mushtaq and F. Siddiqui, "Detection of diabetic retinopathy using deep learning methodology," in ***IOP Conference Series****: Materials Science and Engineering*, 2021.

[47] S. Karki and P. Kulkarni, "Diabetic Retinopathy Classification using a Combination of EfficientNets," in ***International Conference on Emerging Smart Computing and Informatics*** *(ESCI)*, Pune, India, 2021.

[48] G. U. Parthasharathi, K. V. kumar, R. Premnivas and K. Jasmine, "Diabetic Retinopathy Detection Using Machine Learning," ***Journal of Innovative Image Processing***, vol. 4, no. 1, pp. 26-33, 2022.

[49] N. Shaik and T. Cherukuri, "Hinge attention network: A joint model for diabetic retinopathy severity grading," ***Applied Intelligence,*** vol. 52, p. 15105–15121, 2022.

[50] M. Oulhadj, J. Riffi, K. Chaimae, A. M. Mahraz, B. Ahmed, A. Yahyaouy, C. Fouad, A. Meriem, B. A. Idriss and H. Tairi, "Diabetic retinopathy prediction based on deep learning and deformable registration," ***Multimedia Tools and Applications volume*** , vol. 81, p. 28709–28727 , 2022.

[51] C. Lahmar and A. Idri, "Deep hybrid architectures for diabetic retinopathy classification," ***Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization***, pp. 1-19, 2022.

[52] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy and S. Venugopalan, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," ***Jama*** , vol. 316, no. 22, pp. 2402-2410, 2016.

[53] D. Ting, S. Wei, L. R. Pasquale, L. Peng, J. P. Campbell, A. Y. Lee, R. Raman, G. S. W. Tan, L. Schmetterer, P. A. Keane and T. Y. Wong, "Artificial intelligence and deep learning in ophthalmology," ***British Journal of Ophthalmology***, vol. 103, no. 2, pp. 167-175, 2019.

[54] M. D. Abràmoff, Y. Lou, A. Erginay, W. Clarida, R. Amelon, J. C. Folk and M. Niemeijer, "Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning," ***Investigative ophthalmology & visual science***, vol. 57, no. 13, pp. 5200-5206, 2016.

[55] M. Guan, V. Gulshan, A. Dai and G. Hinton, "Who said what: Modeling individual labelers improves classification," in ***Proceedings of the AAAI conference on artificial intelligence***, 2018.

[56] M. Treder, J. L. Lauermann and N. Eter, "Deep learning-based detection and classification of geographic atrophy using a deep convolutional neural network classifier," ***Graefe's Archive for Clinical and Experimental Ophthalmology*** , vol. 256, pp. 2053-2060, 2018.

[57] N. Ramachandran, S. C. Hong, M. J. Sime and G. A. Wilson, "Diabetic retinopathy screening using deep neural network," ***Clinical & experimental ophthalmology***, vol. 46, no. 4, pp. 412-416, 2018.

[58] P. Chudzik, S. Majumdar, F. Caliva, B. Al-Diri and A. Hunter, "Microaneurysm detection using deep learning and interleaved freezing," ***Medical imaging 2018: image processing***, vol. 10574, pp. 379-387, 2018.

[59] S. R. RATH, "diabetic retinopathy 224x224 gaussian-filtered," 2020.

[60] Y. LeCun, Y. Bengio and G. Hinton, ""Deep learning," ***Nature***, vol. 521, no. 7553, pp. 436-444, 2015.

[61] J. Wu, "Introduction to convolutional neural networks," ***National Key Lab for Novel Software Technology. Nanjing University. China***, vol. 5, no. 23, 2017.

[62] M. Gurucharan, "basic cnn architecture," 7 12 2020. [**Online**]. Available: https://www.upgrad.com/blog/basic-cnn-architecture/. [Accessed 1 10 2021].

[63] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Dive into Deep Learning," **arXiv preprint arXiv**:2106.11342, 2021.

[64] J. D. McCaffrey, "Convolution Image Size, Filter Size, Padding and Stride," wordpress, 5 30 2018. [**Online**]. Available:

https://jamesmccaffrey.wordpress.com/2018/05/30/convolution-image-size-filter-size-padding-and-stride/. [Accessed 1 10 2021].

[65] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," ***Journal of artificial intelligence research***, vol. 16, pp. 321-357, 2002.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in ***arXiv preprint arXiv****:1409.1556* , 2014.

[67] datahacker, "CNN VGG 16 and VGG 19," 10 11 2018. [**Online**]. Available: https://datahacker.rs/deep-learning-vgg-16-vs-vgg-19/. [Accessed 1 2 2023].

[68] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in ***IEEE conference on computer vision and pattern recognition***, 2017.

[69] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in ***International conference on machine learning***, 2019.

[70] Z.-H. Zhou, Ensemble methods: foundations and algorithms, **CRC press**, 2012.

[71] S. C.Weller, "Cultural Consensus Model," in ***Encyclopedia of Social Measurement,*** **Galveston, Texas, USA, Elsevier**, 2005, pp. 579-585.

[72] Mathwork, "Assess classifier performance," Mathwork, 2020. [**Online**]. Available: https://www.mathworks.com/help/stats/assess-classifier-performance.html. [Accessed 1 8 2021].

RESUME

Shuhad Imad Hadi ALDUJAILI began her academic journey in Baghdad, Iraq, completed her high school at Al Hariri High School (2009-2010). Post-high school, she pursued undergraduate studies at Al-Turath University (2013-2014). In 2021, she moved to Karabuk, Turkey, to continue her academic progression, enrolling in a Master of Science in computer engineering program at Karabuk University.