Early Detection of Diabetic Retinopathy Using Deep Learning Techniques

by

Veronica Jessica Gomes
20241053
Kazi Ahnaf Alavee
17241013
Anirudh Sarda
21341051
Zebel-E-Noor Akhand
17201124

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Veronica Jessica Gomes
Veronica Jessica Gomes
20241053

Kazi Ahnaf Alavee
Kazi Ahnaf Alavee
17241013

Anirudh Sarda
Anirudh Sarda
21341051

Zebel-E-Noor Akhand
Zebel-E-Noor Akhand
17201124

Approval

The thesis/project titled "Early Detection of Diabetic Retinopathy Using Deep Learning Techniques" submitted by

1. Veronica Jessica Gomes (20241053)
2. Kazi Ahanaf Alavee (17241013)
3. Anirudh Sarda (21341051)
4. Zebel-E-Noor Akhand (17201124)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on September 26, 2021.

Examining Committee:

Supervisor:
(Member)

Mr. Moin Mostakim
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Dr. Mohammad Zavid Parvez
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

We, humans, are the bearer of diseases. While most of them have a thoroughly researched and contemplated solution set, some of them do not. Diabetes is one of those common diseases that do not have a clear solution but has ways to minimize its effects. It is a globally prevalent condition that leads to several complications including those that are deadly. One of those intricate complexities includes Diabetic retinopathy (DR), a human eye disease that may affect one or both eyes hampering the functionality and leading to compromised vision and eventually, permanent blindness. Thus, detection of diabetic retinopathy in the primitive stages will help reduce the chances of getting visually impaired, following proper treatment and other necessary precautions. The prime objective of our paper is to take aid from the state-of-the-art models which are pretrained on different images and also to propose a basic CNN model that will have comparative results. To be more precise, we have used transfer learning models like DenseNet121, Xception, Resnet50, VGG16, VGG19, and Inception to classify the data based on single-label and multi-label. In our approach, single-label classification using categorical cross-entropy and softmax function works better as we reached the best accuracy, precision, and recall values using the approach. In our case, Xception has reached an accuracy of 82% which is a state-of-the-art result for the used dataset. In addition, our proposed model reached an accuracy of 71% which worked better than some of the transfer learning models. Finally, most of our approaches classified the data correctly even though the dataset is very unevenly distributed.

Keywords: Data Preprocessing; Transfer Learning; Convolutional Neural Network; Xception; Inception

Acknowledgement

First of all, all praise to the Great Almighty, as we were able to finish our thesis without any serious impediment.

Second of all, to our co-advisor, Mr. Moin Mostakim, and co-supervisor, Dr. Mohammad Zavid Parvez, who consistently supported and guided us throughout this entire project.

Finally, to our parents, whose perennial support and encouragement played a pivotal role in completing this thesis.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

Adam Adaptive Moment Estimation

AUC Area Under the Curve

CNN Convolutional Neural Network

CONV Convolutional

Conv2D Least Recently Used

DCNN Deep convulational neural network

DMO Diabetic macular oedema

DR Diabetic Retinopathy

GBM Gradient Boosting method

ILSVRC ImageNet Large Scale Visual Recognition Challenge

KNN K-nearest neighbours

LDA Linear Discriminant analysis

MLP Multilayer Perceptron

NPDR Non Proliferative Diabetic Retinopathy

PACS Picture Archiving and Communication System

PDR Proliferative Diabetic Retinopathy

PNN Probabilistic neural network

ReLU Rectified Linear Unit

RGB Red Green Blue

SVM Support Vector Machine

VGG16 Visual Geometry Group16

VGG19 Visual Geometry Group19

Chapter 1

Introduction

1.1 Background

Diabetic Retinopathy prevalence is a widespread concern. It is growing at such an alarming rate that researchers presume anyone affected with diabetes has a high chance of being visually impaired by Diabetic Retinopathy. [35] In fact, 4.8% of the world's 37 billion blindness cases are caused due to this condition.

Diagnosis of the retina is based on a complicated domain of features and locations confined in the image. It is especially arduous when it comes to determining Diabetic Retinopathy in patients on the primitive stage, [20] as the microaneurysms, saccular outpouching of capillaries, retinal hemorrhages, and breached blood vessels are mostly obscured in that particular stage. So, to reduce the pressure on Ophthalmologists, researchers introduced a digital method to discern the presence of unwanted substances in the retina and effectively classify them according to their level of severity. The accuracy of the models in terms of correctly identifying microaneurysms juxtaposed with normal patches of the retina was 74%. However, merely detecting microaneurysms did not yield the desired results. Therefore, more ways of distinguishing the aneurysms and grading DR were introduced using KNN, support vector machines, and ensemble-based methods which lead to achieving a sensitivity and specificity of 90%.

In the United Kingdom, this malady is in circulation mostly among the working-age group. [1] Statistics show that 4.1 million people in the United States are victims of Diabetic Retinopathy. While this condition is not as popular in Bangladesh, the number of evident cases continues to escalate every year. Research conducted at a tertiary hospital here, concludes that out of 200 people affected with Type 2 diabetes, 35.5% had Diabetic Retinopathy, 19% had Non-Proliferative Diabetic Retinopathy and 16.5% had Proliferative Diabetic Retinopathy. [14] In general, almost everyone with Type 1 diabetes and two-thirds with Type 2 diabetes possesses some traits of Diabetic Retinopathy. The epidemic of this vision-threatening ailment has been estimated to grow from 37.3 million to 56.3 million globally if necessary precautions and steps are not taken against it. Therefore, a huge percentage of the world population is prone to blindness, no matter how many years it takes to actively threaten their sight.

1.2 Research Objectives

Diabetic Retinopathy, a severe disease, causes permanent blindness in its victims. The quantity of people who are victims of diabetes is dramatically growing due to a rise in life expectancy, an extravagant lifestyle, and many other related factors. Diabetic patients must be treated correctly for DR at the right time for both cost and treatment efficacy. However, because the signs are not seen until the advanced stage, the initiation of treatment is very difficult. Therefore, our main target focuses on the early prediction of different algorithms for Diabetic Retinopathy. Our project's purpose is to create a convolutional neural network model that takes a retina image as input and outputs the right stage of diabetic retinopathy as observed in the image. With a working model correctly predicting the DR in the early stages, an ophthalmologist can be referred for further assessment and treatment. On a global scale, the implementation of such an algorithm could significantly reduce the rate of vision loss attributed to DR.

1.3 Research Problems

Diabetic retinopathy has been the leading cause of blindness for several decades. The prevalence of retinal damage has reached almost the entire population and has proved to be a severe case of Diabetes. However, research shows that 90% of the cases can dwindle to a good extent if there were efficient ways to tackle this. These include detection and vigilant treatment in the primitive stages and proper monitoring of the eyes. The risk of developing Diabetic Retinopathy increases with the duration of having Diabetes. Some of the symptoms include blurred vision, sudden loss of sight in one eye, visions of rings around a light, dark spots, or flashing lights. Moreover, identifiers of DR include micro-aneurysms, swollen retina, leaky blood vessels, development of unusual blood vessels, and nerve tissue impairment. This ailment can be divided into 5 stages: no disease, mild, moderate, severe, and proliferative. With age, the risk of having the disease rises, so middle-aged and elderly diabetics patients are more vulnerable to Diabetic Retinopathy. It must be emphasized that not all patients affected with this will experience serious vision impairment as it only occurs in advanced stages marked by diabetic macular oedema, (DMO) and Proliferative Diabetic Retinopathy (PDR). If DR is identified in time, progression to vision loss can be delayed or prevented, which can be difficult since the condition sometimes displays hardly any signs until it is too late to provide effective care. DR identification is currently toilsome, as a manual procedure is required that involves a qualified ophthalmologist to view and analyze digital retinal fundus images. They can determine whether a person has DR by spotting lesions related to the vascular aberrations induced by the disease. Nevertheless, this approach demands proper expertise and equipment that is usually scarce in the areas where the highest number of people are affected. Furthermore, even highly experienced practitioners were occasionally unable to physically examine and assess the stage of a patient's fundus from diagnostic images. Also, present diagnostic methods are inefficient owing to the length of time required and the number of ophthalmologists involved in the patient's problem resolution. As a result, DR de-

tection techniques began to develop. The earliest algorithms were based on several traditional computer vision techniques and thresholds. Nonetheless, Convolutional neural networks(CNN) in particular have demonstrated their dominance over conventional algorithms in classification and object identification tasks in recent years. In this paper, we propose different transfer learning approaches of CNN and will be doing a comparative study among them, utilizing data preprocessing, to find out which of these models are most effective and helps to detect DR the earliest. But one of the main problems with the majority of CNN techniques used for the DR classification process is that it processes the input data without taking into account that most sections of the retina images are not related to DR but other segments of the input image have more influence on the ultimate label of an image. Thus we will be using different data processing techniques to fix the images before using the CNN algorithms. There was not a lot of datasets created solely for the purpose of Diabetic Retinopathy Detection. The dataset we used was utilized in a kaggle competition and it is unevenly balanced. Therefore, this caused a lot of problems throughout our project.

1.4 Contribution and Impact

The purpose of our research is to verify the fact that our twofold approach of comparing different transfer learning models and the proposed model proved to be effective in finding the optimal model that works best for the disease of diabetic retinopathy. Also comparing the model using different metrics, allowed us to analyze it more precisely on various scales. We have used both single-label and multilabel classification using the models DenseNet-121, Xception, ResNet-50, VGG-16, VGG-19, Inception, and our proposed model. While this model performs well with high accuracy on multi-label classification, restricting the model for single-label classification becomes less ambiguous. To understand the models' performance better, we used both pre-trained and randomly initialized weights in both approaches.

Although many studies on diabetic retinopathy detection have been undertaken, the majority of them have focused on determining the classification accuracy of a few models. However, we have used both classifications and six different transfer learning models and also our proposed model. To summarize, our study offers a potential method to demonstrate that, of all the models tested, the Xception model performed the best in our study, correctly identifying the majority of the data.

1.5 Scopes and Limitations

Although we achieved excellent results using several models, there are certain limitations to the dataset that have an impact on our results, such as an imbalanced class distribution. The problem is most conspicuous in samples with the labels Severe and Proliferative DR. The reason for ambiguity in classifying the sample labels

mentioned is the uneven distribution of data. In comparison to other samples, the number of samples for severe and proliferative DR cases is extremely low. As a result, the model focuses mostly on samples labeled No DR and Moderate, resulting in relatively high accuracies, precision, and recall of data from these classes thus affecting our overall accuracy. If we can train the model on a dataset with balanced class distribution then we can get even better results.

1.6 Documentation Outline

This section provides an outline of the topics covered in each chapter of our thesis paper. Following our discussion of what we intend to do, why we intend to do it, and what we hope to accomplish in this chapter, we move on to the literature review, which summarizes and discusses information gathered from scholarly articles, books, previous research papers, and other sources of information relevant to our area of research. After that, in section 3, a detailed overview of data processing is given, including information on the dataset that we used, feature engineering and data transformation. The experimental setup is discussed in section 4 of the study. Here, we discuss in detail about the different transfer learning approaches of CNN along with representation of the model summary for each and lastly information about model compilation has been provided. In section 5, we examine and assess the results, showing comparisons between different models as well as how different features affect classification accuracy. Finally, we conclude our work in section 6 where information regarding the challenges we faced and also the contributions and future works that can be done regarding our research are mentioned.

Chapter 2

Literature Review and Related Work

One of the papers [15] stated, their contribution was two-fold: first, a contribution Special neural network architecture was suggested for the image recognition task of diabetic retinopathy, which shows superior performance over traditional extraction-based techniques of function. Furthermore, for the proposed algorithm, a method of data augmentation was implemented, which also enhanced the algorithm's efficiency. The main focus was on the categorization of retinal images into regular images and diabetic retinopathy images. For the classifiers, the characteristics used included rough exudates and red lesions, while the classifiers used for the task contained neural networks, sparse representation classifiers, linear discriminant analysis (LDA), support vector machine (SVM), the algorithm of k-nearest neighbors (KNN) and so on. The data provided by Kaggle Community was used in [15]. And in the experiment, translation, stretching, rotation, and flipping to the labeled dataset were applied. A human expert was added to mark the images as ground truth to juxtapose the results attained by automated classification algorithms with the output of human judgment. Hard exudates, red lesions, micro-aneurysms, and blood vessel detection were all used for feature extraction. For the classification task, two types of classifiers were trained: one that combines the extracted features with a gradient boosting trees-based (GBM) classification method (Hard exudates + GBM, Red lesions + GBM, Micro-aneurysms + GBM, and Blood vessel detection + GBM), and the other that uses CNN-based methods (with or without data augmentation). Therefore, in this study, exploration of the application of deep convolutional neural network methodology had been done for the automatic classification of diabetic retinopathy using color fundus image, and achieved an accuracy of 94.5% on the dataset, surpassing the results obtained by utilizing classical approaches.

For another paper [10], study showed that the five-class problem for national screening of DR can be performed using a CNN method. A network with CNN architecture and data augmentation has been developed which can identify the intricate features involved in the classification task such as micro-aneurysms, exudates and hemorrhages on the retina and render a diagnosis automatically, in the absence of user input. They used the dataset from Kaggle coding website that contains over 80,000 images. On the data set of 80,000 images used, the proposed CNN achieved a sensitivity of 95% and an accuracy of 75% on 5,000 validation images. Attain-

ing advantageous offset in sensitivity and specificity were the prime concerns within automated grading and CNNs. This is even more difficult for national standards, which are divided into five categories: normal, mild DR, moderate DR, severe DR, and proliferative DR.. In training, the learning required to classify the images at the extreme ends of the scale was significantly less. These drawbacks came in making the network distinguish between the mild, moderate and severe cases of DR. The network struggled to learn some characteristics to detect the more complex components of DR, as evidenced by the poor sensitivity, which came primarily from the mild and moderate classes. In [11] a deep learning-based CNN method has been introduced for the problem of classifying DR in fundus imagery. The CNN was initially pre-trained on 10,290 images and after 120 epochs of training on the initial images, the network was then trained on the full 78,000 training images for a further 20 epochs. Neural networks suffer from severe over-fitting, to solve this issue, real-time class weights have been implemented in the network. The class weights were updated using a ratio proportional to how many images in the training batch were categorized as having no evidence of DR for each batch loaded for back-propagation. Stochastic gradient descent with Nesterov momentum was used to train the network. When the network was then trained on the full training set of images with a low learning rate, within a couple of large epochs of the full dataset, the network's accuracy had grown to more than 70%. Every time training loss and accuracy reached a saturation point, the learning rate was reduced by a factor of ten. The network was trained only once using the original pre-processed images. Afterward, real-time data augmentation was used throughout training to improve the localization ability of the network. For this five-class problem, the accuracy has been defined as such- the number of patients with a correct classification. The final trained network achieved 95% specificity, 75% accuracy and 30% sensitivity.

Now, the following paper [46] consists mainly of three approaches of Convolutional Neural Network; Inception, VGG16 and ResNet to classify the DR stages and compare the results that they produce. Various strategies were applied to cleanse and amplify the data, also to optimize the CNN to harbour skewed data sets. They had to run 25 epochs with a short circuit abort if no improvement is shown after 6 epochs, for most of their training. For statistical evaluation, most of the tests determined two key aspects of the model training; Accuracy and AUC. They primarily used categorical accuracy. The main datasets used were the Kaggle DR competition dataset which contains 35000 eye images with 5 stages of DR disease, and the Messidor dataset which consists of 1200 images with 4 stages of DR progression. After carefully experimenting with the three models, some differences were drawn out. Firstly, the weights for Inception V3 are smaller than the other two making it faster to download and train its network. The VGG network is comparatively slower to train and the network architecture weights are also pretty large. However, the VGG model showed the highest level of accuracy(67%) and AUC, with a value of 0.67, making it the most effective model used. The networks were further optimized with features like attention maps and experimentation with various image preprocessing. But the 5-stage DR classification remains a difficult problem, especially when differentiating between the mid phases of DR.

In [19], a deep convolutional neural network was developed to perform early detection by recognizing all microaneurysms and assigns labels correctly to retinal fundus images, which are subsequently graded into five categories. To improve the implementation, they used a 4x4 Kernel-based CNN network and an augmentation method. For dataset, they used Kaggle EyePACS which consisted of 88702 images, from which only a portion was used, so that equal number of images was used for all stages during training. In this paper, they presented, two binary classifications, which were sick (grades: 1, 2, 3, 4) vs healthy (grade: 0) and low (grades: 0, 1) vs high (grades: 2, 3, 4). The low-high DR classification worked better, with a sensitivity of 98% and a specificity of 94% for early-stage detection. Furthermore, they achieved 0.851 quadratic weighted kappa score on the test set of Kaggle dataset in severity grading and 0.844 AUC score. As a result, it was determined that the efficacy of their proposed model was good enough to be used in clinical settings.

According to [26], Diabetic Retinopathy identification has three stages which are image pre-processing, segmentation, feature extraction and classification. This study proposed the use of a neural network classification technique to detect DR. During data pre-processing, the RGB input image was transformed to grayscale format and then further processed for DR detection. In Optical Disk Segmentation, the brighter optic disk was masked and deleted using the region characteristics and area identification in Optical Disk Segmentation. The Canny edge detection algorithm then preserved all the local maxima for enhancing the blurred edges. In Blood Vessel Extraction, the high contrast vessels were removed by applying dilation on the intensity image. Then, with the help of structuring components, the little holes in the picture were filled. The image's highlighted section was segmented using the segmentation technique. Classification, the final stage of detection, was conducted using Neural Networks. The training set was created according to the color properties of the input image, allowing the system to self-train until mistakes were reduced. The diabetic and non-diabetic areas of the eye were depicted in the final classified image. Then on several sets of images, the suggested model's execution was evaluated in terms of sensitivity and specificity. On the basis of specificity and sensitivity, this was compared to an existing SVM classification model. With the help of neural networks, the final outcomes were improved by up to 5%.

In [49], the goal of this study is to use a classifier to detect retinal microaneurysms and exudates for automatic DR screening. As data, they used 94 sets of green Chanel images from the Messidor. Like [26], three stages of the algorithm were used, which are preprocessing, feature extraction and classification. During preprocessing, the image is corrected and enhanced in the first stage utilizing histogram equalization, contrast enhancement, and median computation. Optical disc removal and blood vascular extraction and removal are both conducted during the feature extraction procedure. Using region characteristics and area finding, the brighter optic disc is masked and discarded. The optical disc and blood vessels are then detected using an edge detection method. And then the Canny edge detection algorithm improves fuzzy edges by retaining the gradient, allowing it to recognize feature boundaries more accurately. In blood vessel extraction and removal, the dilatation operation on the intensity image removes vast amounts of contrasted blood vessels.

Furthermore, they applied the structural element in conjunction with the dilation process to fill tiny gaps in photographs. Closing dilation is then used to detect exudate characteristics, followed by the use of an erosion operator. Then, to detect microaneurysms, the morphological opening operation is utilized, making it simple to count their values. In the last stage of the algorithm, which is classification, the SVM classifier and KNN classifier are used to label the retinopathy-free images and abnormal images according to their severity as non-PDR and PDR. Finally, the Graphic User Interface model's performance was assessed, and it received a sensitivity score of 87 percent and a specificity score of 100 percent. The suggested model's accuracy was also tested by the SVM classifier at 87 percent

To address the problem of clinicians misdiagnosing retinopathic diabetes using a manual technique, an artificial neural network model was built in [12]. The data for this study was collected from UCI's machine learning repository, which included attributes from the images of Messidor that were used to predict if an image contains the indication of diabetic retinopathy. The dataset consisted of 20 characteristics and 1151 instant samples of the Messidor pictures' extracted features. Here for data processing, a normalization strategy was utilized to scale the value to fit the input neuron range. Due to its soft-switching capacity and its derivative simplicity, the sigmoid activation mechanism was employed as the activation function for the neurons in the hidden layer and output layer. A training dataset and a research dataset were created from the intelligent machine dataset. The proportions for the training and testing datasets are computed using the 60:40 ratio. The batch algorithm was used to train the intelligent system, in which the training dataset was input into the neural network once with its associated target. Hence a matrix 19*691 sample with its corresponding goal matrix 2*691 was input into the network. Their model surpassed previously constructed models with a recognition rate of 99 percent.

In [50], Diabetic Retinopathy is divided into two categories: nonproliferative diabetic retinopathy (NPDR) and proliferative diabetic retinopathy (PDR) (PDR). In order to identify Diabetic Retinopathy in the fundus area, computer vision and deep learning approaches employing artificial neural networks were used. Images from DR databases were used to train neural networks. Depending on the training datasets, it was used to determine if the person had no diabetic retinopathy, (ii) mild nonproliferative diabetic retinopathy, (iii) severe nonproliferative diabetic retinopathy, and (iv) proliferative diabetic retinopathy. They created a system employing Convolutional Neural Networks in this case (Deep Learning in H2O) H2O's "Deep-Learning" is still based on a "multi-layer feedforward artificial neural network," which is capable of "stochastic gradient descent" via "back-propagation." In H2O, Deep Learning was built into a Multi-Layer Perceptron (MLP). H2O, on the other hand, is permitted to create autoencoders (a neural net that receives a collection of inputs, compresses and encodes them, and then attempts to reconstruct the input as precisely as possible). H2O's Deep Water Project can be used to build recurrent and convolutional neural networks utilizing "third-party integrations" of other "Deep-Learning frameworks" like Caffe and TensorFlow. Image Processing using SVM had an 88 percent recognition rate, Neural Network using Local Tex-

ture Classifier Based on Multilayer Perception had a 91 percent recognition rate, and Deep Learning using DCNN had a 93 percent recognition rate. A deep convolutional neural network is a healthy way toward the level of diabetic retinopathy phases, according to this predicted explication.

The next research paper [41] proposed a CNN model called DenseNet-169 classifier that yielded an accuracy of 90%, outperforming all the other models used for comparison. They used two datasets: "Diabetic Retinopathy Detection 2015" and "APTOS 2019 Blindness Detection" from Kaggle. The images had various problems like a black background, black corners and different sizes which were all solved during data preprocessing. Subsequently, the data turned out to be particularly imbalanced as the majority of them belonged to the "No DR" class. Therefore, data augmentation was used to solve this issue. Data was finally fed into the proposed model which produced an accuracy level of 95% and a validation accuracy of 90%. This was higher than all the models used for comparison; namely SVM with an accuracy of 85.6%, Decision Tree with an accuracy of 85.1%, Regression with an accuracy of 78% and KNN with an accuracy of 55.17%.

Two models, the Probabilistic Neural Network (PNN) and the Support Vector Machine (SVM), are defined and their performances are compared in [3] to diagnose Diabetic Retinopathy. Here an automated method for classifying Diabetic Retinopathy using fundus pictures is introduced. Like [41] and [3] , three phases of the algorithm were used, they were preprocessing, feature extraction, and classification. Uneven lighting in the photograph was adjusted during preprocessing. Grayscale Conversion, Adaptive Histogram Equalization, Discrete Wavelet Transform, Gaussian Matched Filter Response, and Fuzzy C-Means Clustering are some of the preprocessing approaches used for segmenting blood vessels. Radius, Diameter, Area, Arc Length, Center Angle, and Half Area are some of the features extracted. To identify the best approach for classifying the image into their respective classification namely, PDR, NPDR, or Normal, a support vector machine training procedure is used to examine training data. The The Probabilistic Neural Network design consists of layers of interconnected processing units or neurons. The input layer unit makes no calculations and simply transfers the data to the pattern layer neurons. Although all of the classification algorithms tested performed well, the results demonstrate that SVM is more efficient than PNN, with SVM having an accuracy of 97.608 percent and PNN having an accuracy of 89.60 percent.

Chapter 3

Data Preprocessing

3.1 Dataset Structure

The dataset used in this paper comes from Kaggle named - "APTOS 2019 Blindness detection", containing 3662 labeled high-resolution colour fundus retinal images. Researchers have implemented their work on the dataset previously and this dataset contains pictures that were collected over a long period from various clinics using a variety of cameras, resulting in further variance and inconsistencies in image quality, aspect ratio, [28] and other characteristics. We have used 3012 images for training and 512 for validation where the images in this dataset are categorized into 5 different classes based on the severity [41], on a scale of 0-4, each depicting a different stage of the disease. The label of the image will represent the degree of DR determined by its characteristics, such as :

0-No DR
1-Mild
2-Moderate
3-Severe
4-Proliferative DR

There are around 1805 images of eyes that show no signs of Diabetic Retinopathy, 999 images that show signs of Mild Diabetic Retinopathy, 370 images that show signs of Moderate Diabetic Retinopathy, 295 images that show signs of Severe Diabetic Retinopathy, and 193 images that show signs of Proliferative Diabetic Retinopathy.

- No Diabetic Retinopathy- The patient does not have any DR. A perfectly healthy eye with no damages to the retina.
- Mild Diabetic Retinopathy- The very initial stage of Diabetic Retinopathy is identified by the expansion of the retina's blood vessels in the eye which is called Micro aneurysms. A small volume of fluid might flow into the retina causing the macula to expand more which is a region of the retina pointing to the center.
- Moderate Diabetic Retinopathy- The second stage of Diabetic Retinopathy is where inflammation of the blood vessels starts to block the blood supply to

the retina causing nutrients and other important supplies to get trapped. This causes the macula to get occluded with blood and other fluids.

- Severe Diabetic Retinopathy- The more intense stage where a greater part of the retina's blood vessels becomes occluded, causing the reduction in blood flow to the parts of the retina. This gives signals to the body to build up new blood vessels.

- Proliferative Diabetic Retinopathy- The most severe stage of Diabetic Retinopathy causes the blood vessels to build up in the retina. At this stage, the weaker blood arteries cause fluid leakage. As a result several complications related to the vision like- blurriness, narrowing of vision, and some even leading to blindness.

Figure 3.1 shows the labels with their corresponding number of samples in the dataset. The samples are unevenly distributed across the dataset but they are of very good quality. Therefore, our approach is to propose a state-of-the-art methodology without any extra augmentation of the data.

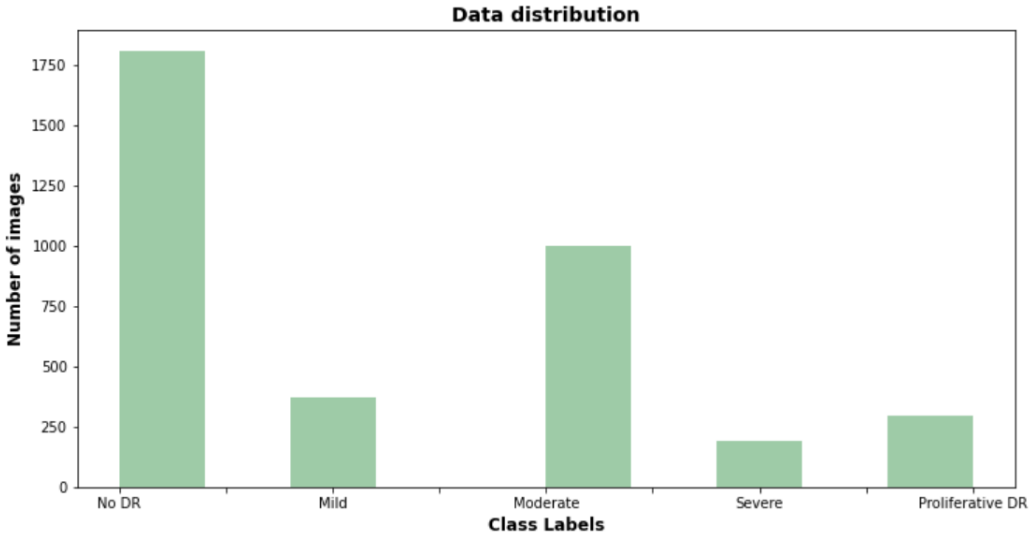


Figure 3.1: Data distribution across the dataset

3.2 Data Processing

Image transformation tasks, [46] in which the input and output are both images, are critical in a variety of applications, including edge recognition, semantic segmentation, and black-and-white image colorization. Convolution-based operators have historically been employed for low-level image processing, but with the introduction of new deep learning approaches, they are now applicable to a wider range of issues. The fully convolutional network, which is made up of multiple convolution operators stacked along with non-linear activation functions, allows for the representation of any image modification functions and the processing of pictures of any resolution. [36] To prepare clean image data for model input, preprocessing is necessary. [13] Image preprocessing can help speed up model inference and reduce model training

time. If the input photos are exceptionally large, shrinking them will greatly reduce model training time without compromising model performance. Identifying the most effective preprocessing for improving model performance necessitates a thorough knowledge of the problem, the data gathered, and the production environment. Some of the image transformation techniques that we have used in our paper are as follows:

1. Image flip- The term "flipping" refers to the rotation of a picture along a horizontal or vertical axis. Flipping pictures is one of the most essential aspects of CNN's image work. Several picture characteristics remain within the image and must be eliminated to get the correct focus and the highest accuracy. The photos are flipped randomly so that the model can learn how to identify flipped images and label them correctly when it sees them. A vertical flip is the same as turning a photograph 180 degrees before horizontally flipping it. When the model comes across an image that is horizontally or vertically flipped and is a bad eye labeled 3- Severe DR, it will identify and label the image correctly and there will not be any anomalies or discrepancies. The flipping methods used here are as follows:

    - Horizontal flip- Reversing an image's full row and column pixels horizontally (left to right) where the flipping happens on the vertical axis.

    - Vertical flip- Reversing an image's full row and column pixels vertically (upside down) where the flipping happens on the horizontal axis.

    - Random Rotation/flipping- Random rotation augmentation rotates the pictures in a clockwise manner from 0 to 360 degrees at random.

2. Resample=Image.LANCZOS- The Lanczos technique can provide a clearer, sharper image than a fuzzy one. When it comes to maintaining detail and reducing aliasing and artifacts, the Lanczos interpolation approach is the best. An original image's input samples are then filtered via the Lanczos kernel (or process) to rebuild it. Along any sharper edge of the produced image, the effect may have a dark or bright halo. It improves the image's apparent sharpness.

3. Image Zoom- Zooming is just expanding an image such that its features become more visible and distinct. Zooming an image will allow the empty areas to be excluded and allow the model to emphasize more on the areas which are truly needed. There is a point at which the model must focus, to properly label the severity of the damaged eye; here is where zooming the image comes in handy. Each pixel value has an influence on the model training during prediction; the smaller the effect of pixels beyond the border, the better the classification.

4. Constant Fill- The unnecessary (black) regions surrounding the retina in the pictures must be adjusted to null to get a neat and clean classification of the images. Because the dark areas may influence the prediction, so they should not be paid any attention.

3.3 Data Split

We have come up with an approach for evaluating the performance of the machine learning algorithm splitting the data into two parts- training and testing. The approach involves dividing the dataset into two separate parts [37]. The training dataset is the one to train the model and prepare it to obtain the optimal result [2]. The validation dataset is prepared so that the machine can make correct predictions regarding the model's performance and provide the best accuracy.

Training the data is essential to make the model learn and expect the correct outcome from the model built. Testing the data is the evaluation of how well the model is trained and performing on the new data which was not used initially. The goal is to use this model in the real world, where it is expected that the model will fit into the existing data where the inputs and outputs are known and predict accurately the result of the data which is unknown [27]. Training the models is not always cost-friendly, there are a few which need to go through some expensive processes starting from random assessments over and over again to maintaining several scopes and limitations making the process troublesome and lengthy.

Models and features are developed from the training set where it acts as the basis for estimating parameters, comparing different models, and performing required tasks to reach the goal for the final model [16]. Once these operations are done only then the test set comes into the role where an endpoint is reached after evaluating the model's performance.

We used 3662 pictures for training and 1928 images for validation in this research. With a larger number of photos to train, the model will learn to correctly recognize and categorize images and will be able to offer promising findings after the testing is completed. To improve our model's performance even further, we ran the images through a model that had previously been pre-trained, making it simpler for the model to distinguish between the infected eye and the healthy eye when the images were sent through. Furthermore, we were able to obtain excellent training and validation accuracy by employing this method.

Chapter 4

Methodology

For this experiment we used 6 pre-trained models from Keras and conducted transfer learning to enhance their performance, thereby increasing their accuracies. The models we used are InceptionV3, VGG16, VGG19, ResNet50, Xception, and DenseNet-121. The images from the Kaggle dataset named Aptos, are fed into each of the models. As they are categorized into 5 different classes of Diabetic Retinopathy; 0-No DR, 1-Mild DR, 2-Moderate DR, 3-Severe DR, and 4-Proliferative DR, the pre-trained models determine and allocate the images into their respective categories. Using the results obtained from the experiment, we did a comparison between the models to determine which one yielded the highest accuracy, precision, and recall.

4.1 Transfer Learning

4.1.1 DenseNet-121

DenseNet architecture [33] intends to improve the depth of deep learning networks while simultaneously making training more cost-effective by using a shorter connection between the layer. There are multiple layers in DenseNet, and each layer is connected to all the others below it, that is, the first layer is connected to the second, third, fourth, and so on, while the second layer is connected to the third, fourth, fifth and so on. This enables the highest possible flow of information across the different layers of the network. Input from all preceding layers are sent to each layer and its feature mappings will be transmitted to all subsequent levels, to maintain feed-forward nature.

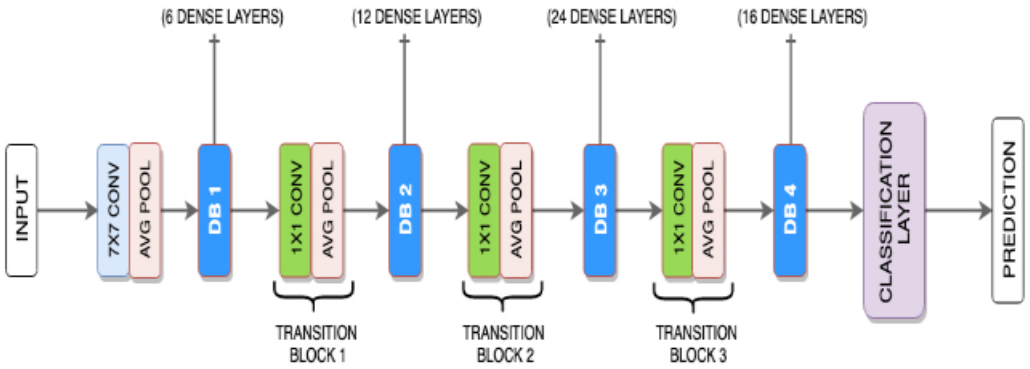


Figure 4.1: DenseNet-121 Architecture (Source: [42])

DenseNet-121 [42] is divided into Dense Blocks, with the size of the features remaining the same inside the blocks while the number of filters changes between the blocks. According to the fig.4.1, [42], the measurements of each volume correspond to the distance and the depth, whereas the numbers at the top contain the feature dimension. The model has a 32 percent growth rate. The volume of each dense block gets larger when growth rates are multiplied by the number of dense layers inside each block. The new feature is then added to the previous rate of growth of this 32.

4.1.2 Xception

Xception [25] comprised of a linear stack of depthwise separable convolution layers that are linked together via residual connections. The goal of depthwise separable convolution is to minimize memory consumption and computing power. These layers are arranged into 14 modules with 36 convolutional layers, which have linear residual connections except the first and last. In Xception, separable convolution categorizes the learning of channel-wise and space-wise features. The architecture of the Xception model [40] is categorized into three key segments: entry flow, middle flow, and exit flow. It is demonstrated in [8] fig.1. Image data is initially sent through the entry flow, then the middle flows eight times, and finally the exit flow. Moreover, the residual link proposed by He et al. [9] created a shortcut in the sequential network, which solves the problem of disappearing gradient representation bottlenecks.

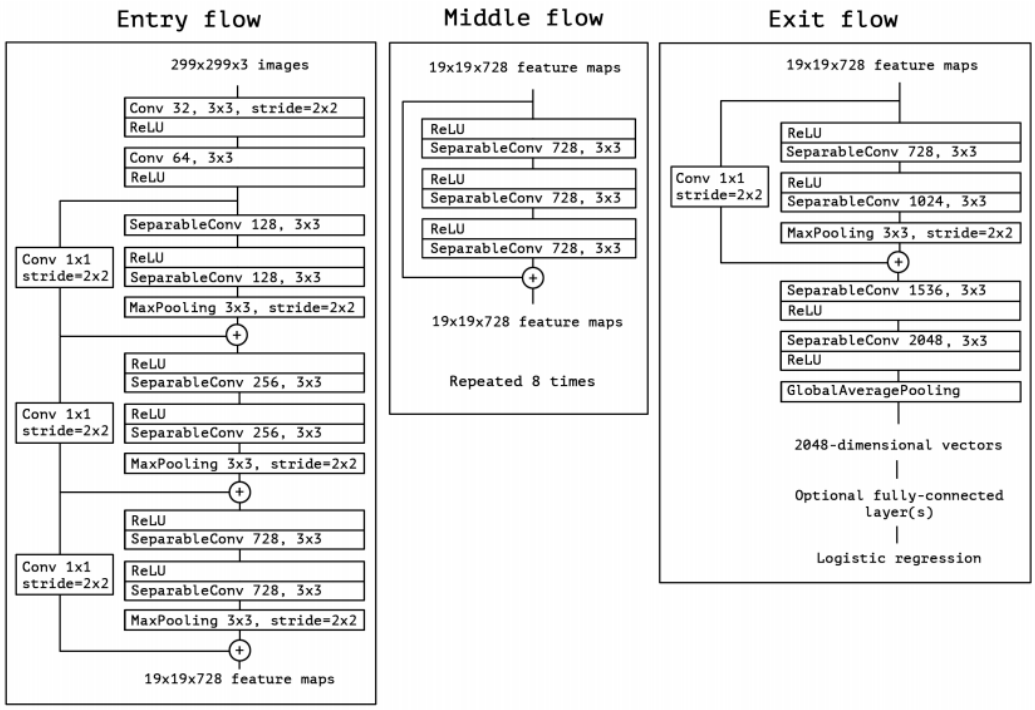


Figure 4.2: Xception Architecture (Source: [8])

4.1.3 InceptionV3

A convolutional neural network model, Inception V3, [51] is an extensively used image recognition and object detection model that has shown its credibility by achieving an accuracy of greater than 78.1% on the ImageNet dataset. This was primarily launched during the ImageNet Recognition Challenge. Multiple researchers are responsible for the production and synthesis of the Inception V3, over many years. This CNN model is 27 layers deep, consisting of an inception layer that is a coalescence of the 1*1 convolutional layer, 3*3 convolutional layer, and 5*5 convolutional layers. Each of their output filter banks merges into a distinct output vector that acts as the input of the next phase. Another 1*1 convolutional layer is added to reduce dimensionality, along with the max pooling layer that is left as a second option for the inception layer. This model works in a hierarchical order where the intrinsic details are considered the first stage, leading to the overall outline of the subject. For this, the layers demand precise filter sizes to correctly detect objects. The Inception layer, therefore, facilitates the internal layers to adopt the filter size that is ideal for their respective functions.

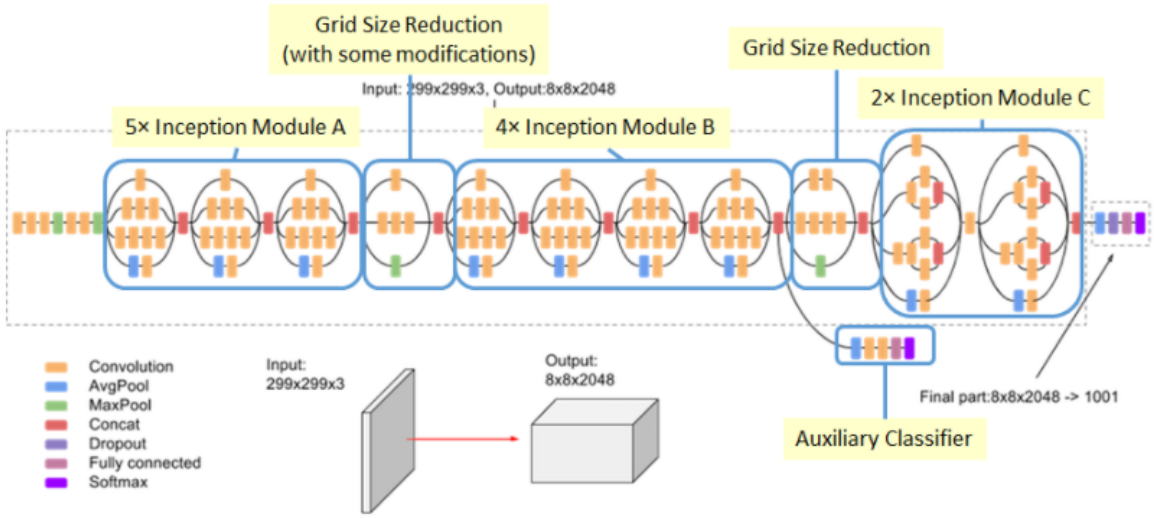


Figure 4.3: InceptionV3 Architecture (Source: [6])

4.1.4 VGG-16

The VGG-16 model is named after the group that manufactured it; Visual Geometry Group from Oxford. It is a convolutional neural network that is lauded as one of the best vision model architectures. [29] This model was built in 2014 to win the ILSVRC(Imagenet) competition and it achieved an [18] accuracy of 92.7% using the ImageNet dataset that comprises 14 million images. VGG16 stands out from other models as it emphasized more on having a 3*3 filter with stride 1 convolutional layer, consistent padding, and 2*2 filter of stride 2 max pool layer throughout the entire architecture, instead of having numerous hyper-parameters. Having a total of 16 layers with weights, it is a colossal network consisting of about 138 million parameters. The input images are fed into a pile of convolutional layers, followed by a spatial pooling. Finally, max pooling is carried out. The end part has 2 fully linked layers before a soft-max layer, for the output. All the covert layers have rectification non-linearity (ReLu) property, with no Local Response Normalization (LRU). Therefore, memory is economically used and computation time is faster.

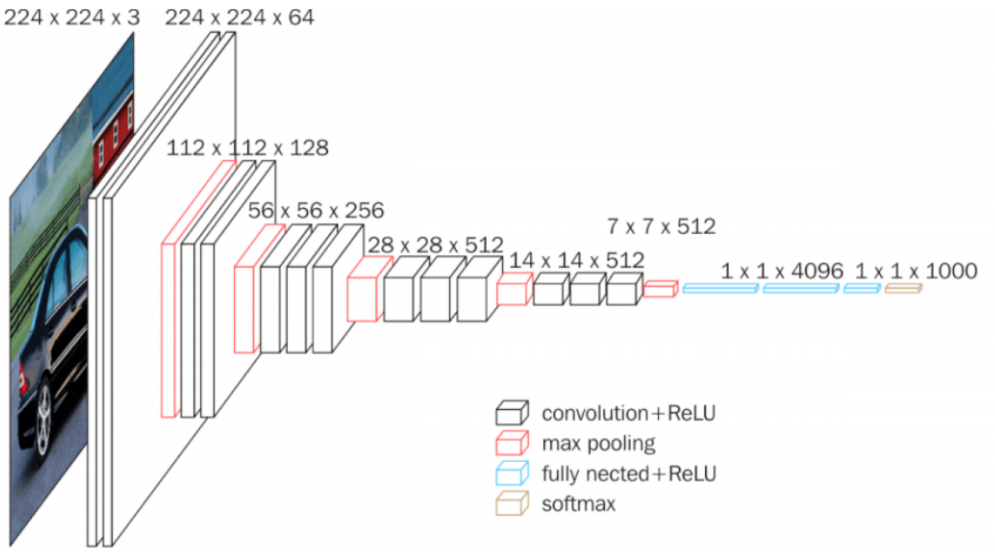


Figure 4.4: VGG-16 Architecture (Source: [7])

4.1.5 ResNet-50

ResNet-50, a convolutional neural network has 50 layers. It comprises 48 convolutional layers with 64 different kernels, [47] 1 max pool layer with a stride of size 2. These layers are replicated 3 times to give a total of 9 layers. The next layer has different kernels and is repeated 4 times to give a total of 12 layers. Following layers consist of other variants of kernels which are repeated many times to form a total of 49 layers. Consequently, an average pool is done with a thoroughly networked layer consisting of 1000 nodes and a softmax function, giving us the last layer of this architecture. A pre-trained version of the network, trained with the images from the ImageNet database, can be loaded in this model. Thus, giving the network an enriched knowledge of feature representation for a large assortment of images.

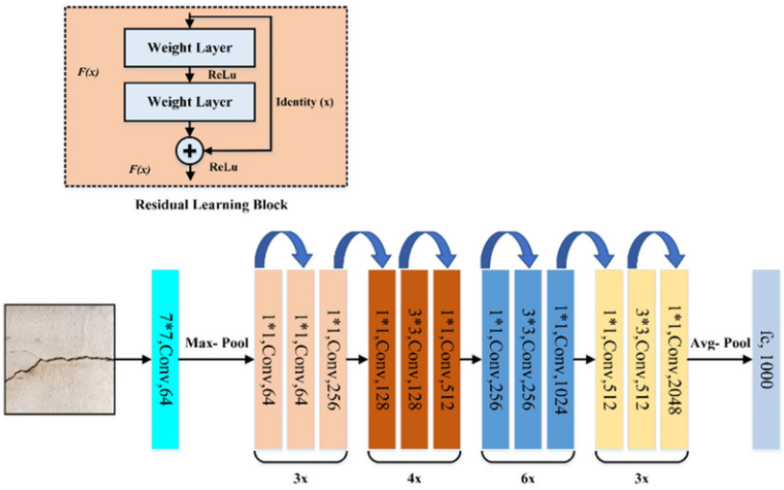


Figure 4.5: ResNet-50 Architecture (Source: [39])

4.1.6 VGG-19

VGG-19 is a convolutional neural network model that is utilized as a good classifier for numerous datasets. [52] It can also be used for transfer learning, content, and style loss. Being a variant of the VGG model, it has various layers including 16 convolutional, 5 max pool, 3 fully connected, and 1 softmax layer. At first, a fixed size image is fed into the model followed by preprocessing. To conserve the spatial resolution, spatial padding is used. After this the model carries out max-pooling over 2*2-pixel windows with stride 2, subsequently, the Rectified Linear Unit (ReLu) is utilized to bring about non-linearity, allowing VGG-19 to enhance its ability to classify images and boost computational time. This model, therefore, performs much better than the previous models as they used tanh and sigmoid functions. Being akin to VGG-16, this architecture is also fed images as input, into a pile of convolutional layers. However, VGG-19 is a better model even though it demands more memory usage.

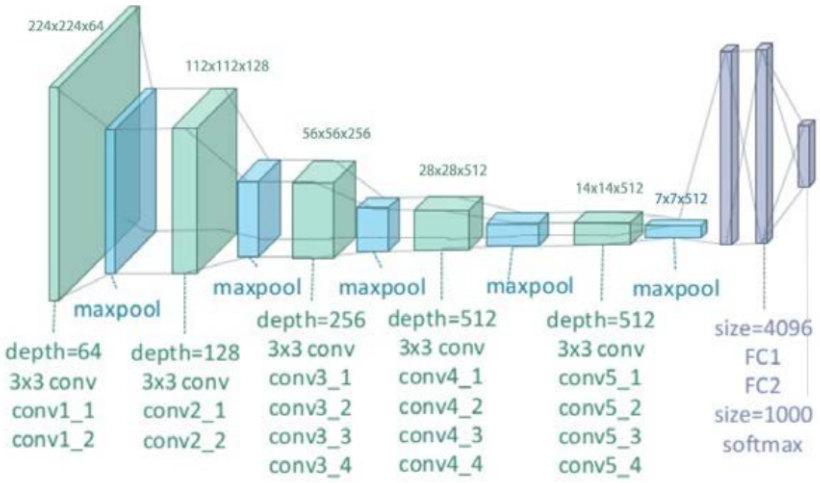


Figure 4.6: VGG-19 Architecture (Source: [21])

4.2 Proposed Model

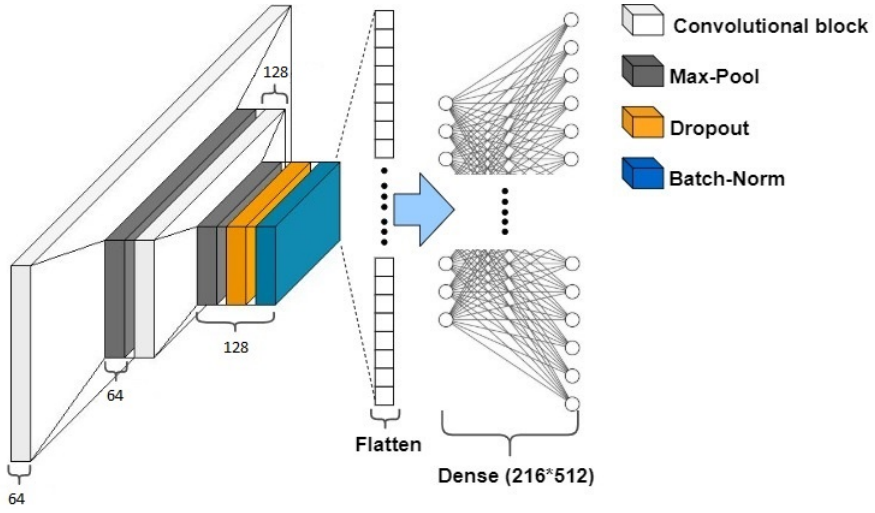


Figure 4.7: Proposed Model architecture

4.2.1 Convolution layer

The simplest process of applying a filter to an input to generate activation is a convolution [23]. A feature map showing the positions and intensity of a recognized feature in input, such as an image, is generated if the same filter is applied to a single input several times. In our model, we used a 2D convolutional layer, abbreviated as conv2D. The kernel slides via 2D input data on a conv2D layer by performing multiplication element-wise. All findings are therefore summarized in a single output pixel. Whenever the kernel passes through a location, the same procedure is performed and the 2D matrix of features is changed into a different 2D matrix of features.

We utilized two convolutional layers in our model, followed by a layer of max pool. The first conv2D layer contains 64 layers input, a size 5 kernel which results in a 5x5 filter matrix and a stride of 1. The filter converges by moving one unit at a time around the input volume, as the stride is set to 1. The activation function used here is ReLU. It offers various advantages over conventional units, including effective gradient propagation and faster computing power than sigmoid units. Following the initial Conv2D layer comes the MaxPooling2D layer, and the output from the max-pool block further goes to another convolution block, where the number of channels are increased to 128.

4.2.2 Max-Pooling layer

The Max-pooling layer decreases the dimensionality of images by lowering the number of pixels in the preceding convolutional layer's output. This is done because using a large number of pixels contributes to more parameters, which might entail enormous amounts of data. This operation chooses the most significant element from the filter's feature map region. As a result, the output of the max-pooling layer's output would be a feature map with the most significant features from the preceding feature map. In our model, we employ two max-pooling layers. In the first max-pooling layer, it reduces the height and width of the image, that it obtains from the first conv2D layer from 224 to 112. The output is then transmitted to the second conv2D layer, which feeds it to the next max-pooling layer, further lowering the height and width to 56. The number of filters in max-pooling remains unchanged during this process.

4.2.3 Dropout layer

Large networks [5] prone to overfitting, as a lot layers stack together to learn a complex function. In Overfitting [38], the model tries to learn too many features from the training dataset, as a result the model performs poorly and is often unable to generalize the pattern of the training data. The dropout method is used to address this problem. Dropout [17] is a regularization technique for simulating simultaneous training of a large number of neural networks of different architectures. By dropping a percentage of units [5], we momentarily remove it from the network, together with all of its incoming and outgoing connections, in turn, drastically decreasing overfitting. This provides significant improvement over other regularization methods. In our model, the output from the second max-pooling layer is sent to the dropout layer, where the rate of the dropout is 0.3. This means that in every epoch, 30% of the neurons in this layer will be dropped at random.

4.2.4 Batch-Normalization layer

When we feed data into a deep learning system [44], we usually modify the values to a balanced scale. The purpose of normalization is to ensure that our model can generalize correctly. By adding extra layers to deep neural networks, batch normalization makes them faster and more stable. Batch normalization [31] is a network layer that enables each layer to learn more independently. It is used to normalize the previous layers' output. In our model, the output from the dropout layer is sent to the batch-norm block as input, where the activation scales it in normalization. Without a neuron, Batch Norm would be computed as follows [4]:

$$z = g(w, x) + b; \quad a = f(z) \tag{4.1}$$

where g() is the neuron's linear transformation, w is its weight, b is its bias, and f()

is its activation function. Now adding Batch Norm

$$z = g(w, x); z^N = \left(\frac{z - m_z}{s_z}\right) \cdot \gamma + \beta; \quad a = f\left(z^N\right) \tag{4.2}$$

Here, z^N is the output of the Batch Norm, m_z is the neuron's output, s_z is the standard deviation of the output of the neurons, and γ and β is the learning parameters of Batch Norm.

4.2.5 Flatten layer

Flattening [24] is the process of turning data into a one-dimensional array for use in the subsequent layer. The flattened layer takes the output of the batch-norm layer and turns it into a one-dimensional array that can then be fed into the dense layers.

4.2.6 Dense layer

The dense layer [34] is a neural network layer that is deep, which means that in this layer each neuron gets input in the preceding layer from all neurons. The first two dense layers in our model are followed by a dropout layer, and the activation function used here is ReLU, for which all the negative activation is converted to zero and drastically lowering the time needed for gradient descent. The last dense layer, which is also the classification layer, and the activation function used is softmax. Softmax [32] is a mathematical function that transforms a number vector into a probability vector, with the probability of each value proportional to the relative scale of the vector. The major benefit of using Softmax is the output probability range. It normalizes the output, which converts weighted sum values into probabilities that add up to one.

4.3 Model Compilation

4.3.1 Metrics

Accuracy [48] is a metric that measures a model's performance throughout all classes. When all the classes are equally important, they can be useful. The ratio between the number of correct predictions and the total number of predictions is used to compute it. For it to operate properly, there must be an equal number of samples in each class.

$$\text{Accuracy} = \frac{\text{True}_{\text{positive}} + \text{True}_{\text{negative}}}{\text{True}_{\text{positive}} + \text{True}_{\text{negative}} + \text{False}_{\text{positive}} + \text{False}_{\text{negative}}} \quad (4.3)$$

Precision [48] is determined by dividing the total number of Positive samples by the number of samples correctly classified as Positive (either correctly or incorrectly). It is a measure for evaluating how well a model classifies a sample as positive. The denominator rises and the accuracy lowers when the model generates a huge number of erroneous Positive classifications or a small number of accurate Positive classifications. When, on the other hand, the model makes a substantial percentage of correct Positive classifications and fewer incorrect Positive classifications, the precision is high (maximize True Positive). Thus, according to high precision, false positives are costly.

$$\text{Precision} = \frac{\text{True}_{\text{positive}}}{\text{True}_{\text{positive}} + \text{False}_{\text{positive}}} \quad (4.4)$$

The recall [48] is determined by dividing the total number of Positive samples by the number of Positive samples correctly classified as Positive. The model's ability to identify Positive samples is determined by the recall. The higher the recall, the greater the number of positive samples detected. Only the classification of positive samples is subject to recall. It makes no difference how the negative samples are categorized.

$$\text{Recall} = \frac{\text{True}_{\text{positive}}}{\text{True}_{\text{positive}} + \text{False}_{\text{negative}}} \quad (4.5)$$

4.3.2 Adam Optimizer

The Adam optimizer [30] is excellent for large data and/or parameter problems because of its great computational power, low memory needs, and invariance to gradient diagonal rescaling. Each parameter is assigned a unique learning level using adaptive models of learning rates. The first and second gradient computations are used to modify the learning rate of growing neural weight. There are numerous situations where it is shown to be a good optimizer for flat surface error minima, which is what Adam prefers.

4.3.3 Loss Function

The loss function indicates how accurate your model is in predicting. If the model predictions are closest to the real data, the Loss will be the least; if the predictions are entirely different from the original values, the Loss will be the largest. We used

two loss functions in our model, Binary Cross-Entropy for multi-label and Categorical Cross-Entropy for a single label. Each of the predicted probabilities is compared to the actual class result in Binary Cross-Entropy, which might be 0 or 1. The score is then determined, with probabilities being penalized based on their deviation from the predicted value This refers to how close or far the value is to the actual value.

To calculate Binary Cross Entropy multi-label classification problem, the following formula can be used [43]:

$$\text{logloss } = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log\left(p_{ij}\right) \tag{4.6}$$

- N is the number of rows.
- M is the number of classes

Because our model has a larger number of classes, Categorical Cross Entropy is utilized to determine the classification of a single label. The Categorical Cross entropy will be used to compare the distribution between predictions (activations of output layers) and the real distribution, where the probability of the true class is 1 or 0 for the other classes. A lower score indicates that the model is performing better.

The formula for Categorical Cross-Entropy for single label is given below [45]:

$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}\left(y_{ij} * \log\left(\hat{y}_{ij}\right)\right) \tag{4.7}$$

Chapter 5

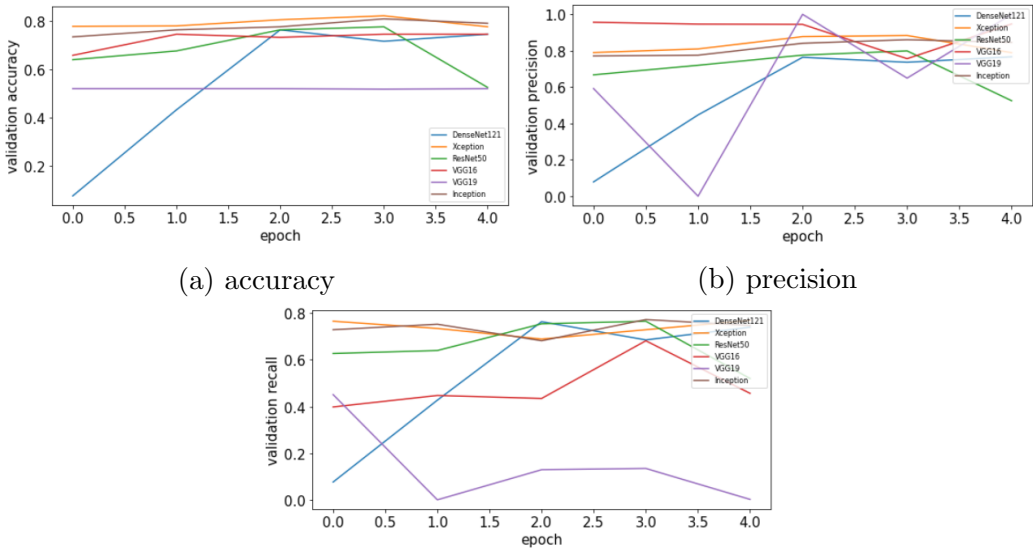
Results

5.1 Overview

Our twofold approach of comparing the transfer learning models and the proposed model proved to be effective in finding the optimal model that works best for the disease of diabetic retinopathy. Since the disease has been prevalent for a long period and is one of the major causes of blindness, early detection of such a disease can save a diabetic person from losing their vision. On the other hand, comparing the model using different metrics also helped us evaluate a model more precisely on different scales. In the first approach, we have used multi-label classification utilizing all the models mentioned in the paper. For instance, the models are glued together with a dense layer with *sigmoid* activation function to score each label between a range of $0 - 1$. This is futile in detecting diabetic retinopathy at an early stage. To be more precise, if the actual label of the sample is 2, then the model predicts values more than 0.5 for labels 0,1 and 2 respectively. As a result, the model can predict different values for each label from where we can take the maximum value for a certain label, which is the predicted label by the model. While this model works well on multi-label classification with very high accuracy, narrowing the model down for single-label classification becomes ambiguous and the accuracy drastically decreases.

In contrast, the single-label classification using *softmax* activation function and *categorical crossentropy* brought fruitful results for many models such as Xception and Inception. On the other hand, our proposed model has competitive results in comparison to Xception which proved to be the most ideal model for our research. The accuracies of Xception and Inception are 82.18% and 80.91% respectively. Whereas, the accuracy of our proposed model for single-label classification is 70.18% which is better than other transfer learning models gathered for this manuscript.

For both the approaches, we have used both pre-trained and randomly initialized weights to gather more intuition about the performance of the models. The compilation of the model is done with several metrics like accuracy, precision, and recall. Additionally, *Adam* optimizer is used to train the models more quickly [22]. The number of epochs used to train the model varied for both approaches. On the single-label classification, we used 5 epochs to train the model previously trained on the ImageNet dataset. On contrary, 10 epochs are preferred for the model that is



(a) accuracy

(b) precision

(c) recall

Figure 5.1: Validation of different pre-trained transfer learning models on different metrics

initialized with random weights. In the same way, 2 epochs are used for multi-label classification (models with pre-trained weights). Whereas, we used 5 epochs to train the models initialized with randomized weights.

5.2 Comparison of different model used

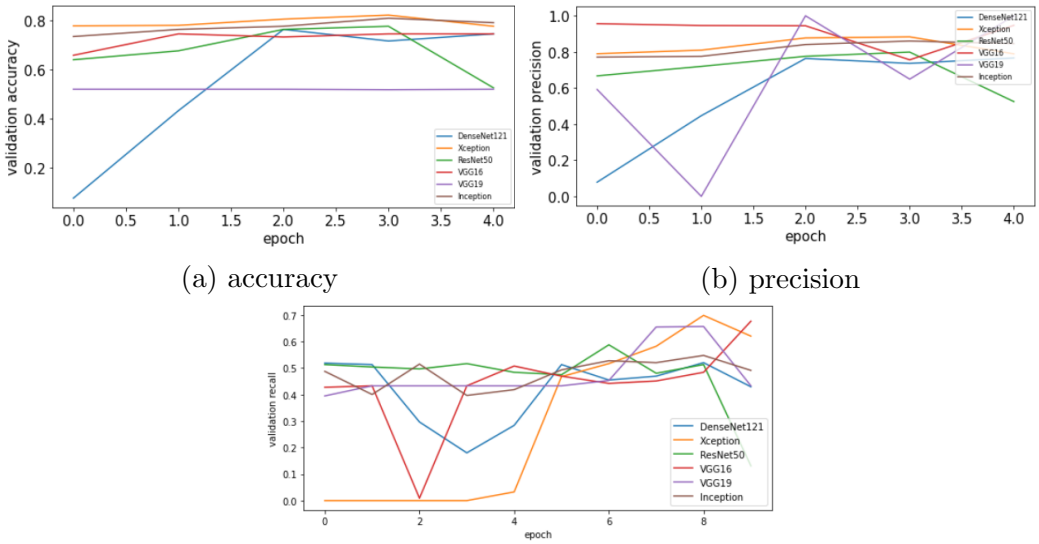
In single-label classification, DenseNet-121, Xception, ResNet-50, VGG-16, VGG-19 and Inception have accuracies of 76.4%, 82.18%, 77.64%, 74.55%, 52%, 80.91% respectively which is shown in Table 5.1 along with the other evaluation metrics like precision and recall. Amongst all the transfer learning models, Xception works best with a considerable amount of parameters compared to other models. However, our proposed model also has an accuracy of 69.64% which is a competitive accuracy compared to the state-of-the-art models that are used. In comparison, most of the models worked well in multi-label classification whereas VGG-19 did not provide a satisfactory result for our domain of research. Table 5.2 demonstrates the accuracy, precision, and recall when multiple labels need to be classified.

Model	Trainable Parameters	Accuracy	Precision	Recall	Epoch Used
DenseNet121(random weights)	7,085,701	0.5873	0.6596	0.4545	10
DenseNet121(imagenet weights)	7,085,701	0.0764	0.0785	0.0764	5
Xception(random weights)	21,069,869	0.7491	0.8525	0.6200	10
Xception(imagenet weights)	21,069,869	0.8218	0.8833	0.7291	5
ResNet50(random weights)	23,797,509	0.7345	0.8730	0.5873	10
ResNet50(imagenet weights)	23,797,509	0.7764	0.7989	0.7655	5
VGG16(random weights)	14,780,997	0.7491	0.9348	0.4691	10
VGG16(imagenet weights)	14,780,997	0.7455	0.9472	0.4564	5
VGG19(random weights)	20,090,693	0.7491	0.9540	0.4527	10
VGG19(imagenet weights)	20,090,693	0.5200	1	0.0018	5
Inception(random weights)	22,031,269	0.7382	0.9125	0.4927	10
Inception(imagenet weights)	22,031,269	0.8091	0.8603	0.7727	5
Retino-CNN(Proposed)	103,104,901	0.6964	0.9421	0.4436	10

Table 5.1: Comparison of model trained for single-label classification

Model	Trainable Parameters	Accuracy	Precision	Recall	Epoch Used
DenseNet121(random weights)	6,958,981	1	0.9314	0.8377	5
DenseNet121(imagenet weights)	6,958,981	0.7236	0.9819	0.8139	2
Xception(random weights)	20,817,197	1	0.7458	0.8668	5
Xception(imagenet weights)	20,817,197	0.7873	0.9618	0.9092	2
ResNet50(random weights)	23,544,837	1	0.9871	0.5406	5
ResNet50(imagenet weights)	23,544,837	0.9200	0.9394	0.8748	2
VGG16(random weights)	14,717,253	1	0.9005	0.8862	5
VGG16(imagenet weights)	14,717,253	1	0.9865	0.5802	2
VGG19(random weights)	20,026,949	1	1	0.4850	5
VGG19(imagenet weights)	20,026,949	1	0.7904	0.7152	2
Inception(random weights)	21,778,597	1	0.9394	0.8607	5
Inception(imagenet weights)	21,778,597	0.9964	0.7922	0.8942	2
Retino-CNN(Proposed)	103,104,901	1	0.8755	0.8871	5

Table 5.2: Comparison of model trained for multi-label classification



(a) accuracy

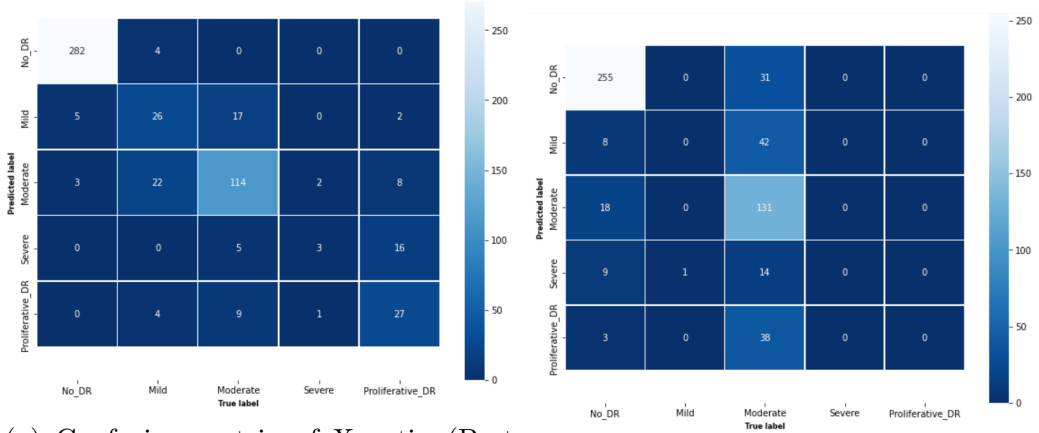
(b) precision

(c) recall

Figure 5.2: Validation of different transfer learning models initialized with random weights on different metrics

5.3 Analysis of the best approach

The Xception model worked best in our research, classifying most of the samples accurately. The problem mainly occurs for the samples with labels- Severe and Proliferative DR. The reason for ambiguity in classifying the sample labels mentioned is the uneven distribution of data. Figure 3.1 shows that the number of samples for the severe and proliferative DR cases is very low compared to the other samples. As a result, the model emphasizes mainly the samples with a label of No_DR and Moderate, therefore, the accuracies, precision, and recall of data from these classes are relatively high. Figure 5.3a shows the confusion matrix of Xception, where we can see that most of the samples are correctly predicted out of 550 validation samples. The model makes most of the mistakes when the labels are Severe and Proliferative DR as the amount of data is relatively less.



(a) Confusion matrix of Xception(Best Model)

(b) Confusion matrix of Proposed Model

Figure 5.3: Our Best and Proposed Approach

Nonetheless, our proposed model has an accuracy of 70% because the model only emphasizes the two classes (No_DR and Moderate) whereas, it completely fails to learn the features of the three classes (Severe, Mild, and Proliferative_DR) as shown in Figure 5.3b. Thus, the accuracy of our proposed model is drastically brought down. But the accuracy reached by Xception is a state-of-the-art accuracy for this particular dataset.

Chapter 6

Conclusion

As the traditional method for detecting DR is time-consuming, difficult, and expensive, several studies have been conducted to automatize the detection process utilizing machine learning and deep learning methodologies. We conducted a complete examination of numerous approaches for automatically detecting this ailment in this paper, and we sought to offer our deep learning methodology for the early detection of diabetic retinopathy.

We present a Diabetic Retinopathy Detection Model that can classify images based on disease pathology of varying severity levels. An input image is convolved with a defined weight matrix in a CNN transfer learning model to extract specific image attributes without losing spatial arrangement information. We first evaluate various models to determine and identify the best performing CNN for single-label and multi-label classification, as well as use various methods of data preprocessing to improve test accuracy. To gain a better understanding of the models' performance, we used both pre-trained and randomly initialized weights in both approaches. Xception performs best with less number of parameters in single-label classification when compared to other models, however, our proposed model also had a competitive accuracy when compared to the state-of-the-art models used. In comparison, most of the models performed well in multi-label classification.

The purpose of our study was to find a more efficient technique to diagnose early-stage diabetic retinopathy, and we discovered that the Xception model performs the best in accurately classifying the majority of the samples, and accuracy reached by Xception is 82%, the current state-of-the-art accuracy for this particular dataset. But the unequal distribution of data is the cause of the ambiguity in identifying the sample labels presented. If the distribution of data was even, then the results would have been far more precise. Nonetheless, through our research, we hope to help healthcare organizations detect this malady at its primitive stage so that patients do not have to become victims of permanent blindness. In our future work, we aim to produce a dataset that has even distribution of samples and will be made for the sole purpose of research in this field.

Bibliography

[1] J. H. Kempen, B. J. O'Colmain, M. C. Leske, S. M. Haffner, R. Klein, S. E. Moss, H. R. Taylor, and R. F. Hamman, "The prevalence of diabetic retinopathy among adults in the united states.," *Archives of ophthalmology (Chicago, Ill.: 1960)*, vol. 122, no. 4, pp. 552–563, 2004.

[2] Z. Reitermanova, "Data splitting," in *WDS*, vol. 10, 2010, pp. 31–36.

[3] R. Priya and P. Aruna, "Svm and neural network based diagnosis of diabetic retinopathy," *International Journal of Computer Applications*, vol. 41, no. 1, 2012.

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[5] ——, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. arXiv: 1512.00567. [Online]. Available: http://arxiv.org/abs/1512.00567.

[7] X. Zhang, J. Zou, K. He, and J. Sun, "Accelerating very deep convolutional networks for classification and detection," *CoRR*, vol. abs/1505.06798, 2015. arXiv: 1505.06798. [Online]. Available: http://arxiv.org/abs/1505.06798.

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016. arXiv: 1610.02357. [Online]. Available: http://arxiv.org/abs/1610.02357.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] 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.

[11] ——, "Convolutional neural networks for diabetic retinopathy," *Procedia computer science*, vol. 90, pp. 200–205, 2016.

[12] A. A. Adekunle, A. Khashman, E. O. Olaniyi, and O. K. Oyedotun, "Diabetic retinopathy diagnosis using neural network arbitration," *Bulletin of the Transilvania University of Brasov. Mathematics, Informatics, Physics. Series III*, vol. 10, no. 1, p. 179, 2017.

[13] C. Affonso, A. L. D. Rossi, F. H. A. Vieira, A. C. P. de Leon Ferreira, *et al.*, "Deep learning for biological image classification," *Expert Systems with Applications*, vol. 85, pp. 114–122, 2017.

[14] R. Mathur, I. Douglas, K. Bhaskaran, and L. Smeeth, "Diabetic eye disease: A uk incidence and prevalence study," 2017.

[15] K. Xu, D. Feng, and H. Mi, "Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image," *Molecules*, vol. 22, no. 12, p. 2054, 2017.

[16] Aug. 2018. [Online]. Available: https://docs.microsoft.com/en-us/analysis-services/data-mining/training-and-testing-data-sets?view=asallproducts-allversions.

[17] *A gentle introduction to dropout for regularizing deep neural networks,* Dec. 2018. [Online]. Available: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

[18] M. ul Hassan, "Vgg16-convolutional network for classification and detection," *en línea].[consulta: 10 abril 2019]. Disponible en: https://neurohive. io/en/popular-networks/vgg16*, 2018.

[19] S. M. S. Islam, M. M. Hasan, and S. Abdullah, "Deep learning based early detection and grading of diabetic retinopathy using retinal fundus images," *arXiv preprint arXiv:1812.10595*, 2018.

[20] C. Lam, D. Yi, M. Guo, and T. Lindsey, "Automated detection of diabetic retinopathy using deep learning," *AMIA summits on translational science proceedings*, vol. 2018, p. 147, 2018.

[21] Y. Zheng, C. Yang, and A. Merkulov, "Breast cancer screening using convolutional neural network and follow-up digital mammography," May 2018, p. 4. DOI: 10.1117/12.2304564.

[22] K. Bae, H. Ryu, and H. Shin, "Does adam optimizer keep close to the optimal point?" *arXiv preprint arXiv:1911.00289*, 2019.

[23] *How do convolutional layers work in deep learning neural networks?* Apr. 2019. [Online]. Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/.

[24] Jiwon, *The most intuitive and easiest guide for convolutional neural network*, Jan. 2019. [Online]. Available: https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480.

[25] S. H. Kassani, P. H. Kassani, R. Khazaeinezhad, M. J. Wesolowski, K. A. Schneider, and R. Deters, "Diabetic retinopathy classification using a modified xception architecture," in *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, 2019, pp. 1–6.

[26] P. Kaur, S. Chatterjee, and D. Singh, "Neural network technique for diabetic retinopathy detection," *Int J Eng Adv Technol (IJEAT)*, vol. 8, no. 6, pp. 440–445, 2019.

[27] ——, "Neural network technique for diabetic retinopathy detection," *Int J Eng Adv Technol (IJEAT)*, vol. 8, no. 6, pp. 440–445, 2019.

[28] S. Qummar, F. G. Khan, S. Shah, A. Khan, S. Shamshirband, Z. U. Rehman, I. A. Khan, and W. Jadoon, "A deep learning ensemble approach for diabetic retinopathy detection," *IEEE Access*, vol. 7, pp. 150 530–150 539, 2019.

[29] R. Thakur, "Step by step vgg16 implementation in keras for beginners," *Medium*, 2019.

[30] J. Yuan and Y. Tian, "An intelligent fault diagnosis method using gru neural network towards sequential data in dynamic processes," *Processes*, vol. 7, no. 3, p. 152, 2019.

[31] Sep. 2020. [Online]. Available: https://analyticsindiamag.com/everything-you-should-know-about-dropouts-and-batchnormalization-in-cnn/.

[32] Oct. 2020. [Online]. Available: https://machinelearningmastery.com/softmax-activation-function-with-python/.

[33] D. Ezzat, H. A. Ella, *et al.*, "Gsa-densenet121-covid-19: A hybrid deep learning architecture for the diagnosis of covid-19 disease based on gravitational search optimization algorithm," *arXiv preprint arXiv:2004.05084*, 2020.

[34] Palash, *Keras dense layer explained for beginners*, Oct. 2020. [Online]. Available: https://machinelearningknowledge.ai/keras-dense-layer-explained-for-beginners/.

[35] M. Rahman, A. Kamrul-Hasan, M. Islam, A. Hasan, F. Chowdhury, O. Miah, M. Islam, S. Wadud, and A. Akhanda, "Frequency and risk factors of diabetic retinopathy among patients with type 2 diabetes mellitus: A single-center study from bangladesh.," *Mymensingh Medical Journal: MMJ*, vol. 29, no. 4, pp. 807–814, 2020.

[36] M. Shaban, Z. Ogur, A. Mahmoud, A. Switala, A. Shalaby, H. Abu Khalifeh, M. Ghazal, L. Fraiwan, G. Giridharan, H. Sandhu, *et al.*, "A convolutional neural network for the screening and staging of diabetic retinopathy," *Plos one*, vol. 15, no. 6, e0233514, 2020.

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

[38] Jun. 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-prevent-overfitting-in-neural-networks-part-1/?__cf_chl_captcha_tk__=pmd_cVfjRZeeU91wDf0GUju8wp0SrRN_oh450rdUe.kwdQY-1632659543-0-gqNtZGzNAyWjcnBszQl9.

[39] L. Ali, F. Alnajjar, H. Jassmi, M. Gochoo, W. Khan, and M. Serhani, "Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures," *Sensors*, vol. 21, p. 1688, Mar. 2021. DOI: 10.3390/s21051688.

[40] S. Majumder and N. Kehtarnavaz, "Multitasking deep learning model for detection of five stages of diabetic retinopathy," *arXiv preprint arXiv:2103.04207*, 2021.

[41] G. Mushtaq and F. Siddiqui, "Detection of diabetic retinopathy using deep learning methodology," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1070, 2021, p. 012 049.

[42] B. Pansare, N. Deorukhakar, T. Hajare, P. Nalawade, and P. Nawghare, "Deep learning for diabetic retinopathy," *International Journal of Recent Advances in Multidisciplinary Topics*, vol. 2, no. 6, pp. 27–31, 2021.

[43] Shipra, *Binary cross entropy/log loss for binary classification*, Mar. 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/.

[44] ——, *Introduction to batch normalization*, Mar. 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-batch-normalization/.

[45] [Online]. Available: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy.

[46] R. Cai, "Study of convolutional neural networks for early detection of diabetic retinopathy,"

[47] P. Dwivedi, *Understanding and coding a resnet in keras.* [Online]. Available: https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33.

[48] A. Fawzy, *Evaluating deep learning models: The confusion matrix, accuracy, precision, and recall,* [Online]. Available: https://www.kdnuggets.com/2021/02/evaluating-deep-learning-models-confusion-matrix-accuracy-precision-recall.html#.YDAlN7B6GNo.blogger.

[49] S. Gupta, "And karandikar am,"" *Diagnosis of Diabetic Retinopathy using Machine Learning*, vol. 3, no. 2, p. 1 000 127,

[50] V. K. Gurani, A. Ranjan, and C. L. Chowdhary, "Diabetic retinopathy detection using neural network,"

[51] F. Shaikh, *Deep learning in the trenches: Understanding inception network from scratch.*

[52] *Understanding the vgg19 architecture.* [Online]. Available: https://iq.opengenus.org/vgg19-architecture/.