**PROJECT REPORT**

**ON**

**Deep learning web app for Diagnosing Multiple Visual Impairment Diseases**

**Carried Out at**



**CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING**

**ELECTRONIC CITY, BANGALORE**

### UNDER THE SUPERVISION OF

**Mrs. Sukeshini Ramadasu**

**C-DAC Bangalore**

**Submitted By**

**Shivam Sharma (220950125086)**

**Utkarsha Khachane (220950125103)**

**Sanket Bhave (220950125075)**

**Tejas Gokhale(220950125101)**

**Yash Ballal (220950125114)**

**Gaurav Rahane (220950125069)**

**PG DIPLOMA IN BIG DATA ANALYTICS C-DAC, BANGALORE**

**Candidate’s Declaration**

We hereby certify that the work being presented in the report entitled **Deep learning web app for Diagnosing Multiple Visual Impairment Diseases**, in partial fulfilment of the requirements for the award of PG Diploma Certificate and submitted in the department of PG-DBDA of the C-DAC Bangalore, is an authentic record of our work carried out during the period, 15th January 2023 to 15th March 2023 under the supervision of **Mrs. Sukeshini Ramadasu** ,C-DAC Bangalore. The matter presented in the report has not been submitted by us for the award of any degree of this or any other

Institute/University.

#### (Name and Signature of Candidate)

Shivam Sharma (220950125086)

Utkarsha Khachane (220950125103)

Sanket Bhave (220950125075)

Tejas Gokhale(220950125101)

Yash Ballal (220950125114)

Gaurav Rahane (220950125069)

#### Counter Signed by

--------------------

##### ACKNOWLEDGMENT

We take this opportunity to express our gratitude to all those people who have been directly and indirectly with us during the competition of this project.

we pay thanks to Mrs. Sukeshini Ramadasu who has given guidance and a light to us during this major project. His versatile knowledge about “title name “has eased us in the critical times during the span of this Final Project.

we acknowledge here out debt to those who contributed significantly to one or more steps. We take full responsibility for any remaining sins of omission and commission.

**Shivam Sharma (220950125086)**

**Utkarsha Khachane (220950125103)**

**Sanket Bhave (220950125075)**

**Tejas Gokhale(220950125101)**

**Yash Ballal (220950125114)**

**Gaurav Rahane (220950125069)**

#### CERTIFICATE

This is to certify that the work titled **Deep learning web app for Diagnosing Multiple Visual Impairment Diseases** is carried out by Shivam Sharma (220950125086),Utkarsha Khachane (220950125103),Sanket Bhave (220950125075),Tejas Gokhale(220950125101), Yash Ballal (220950125114), Gaurav Rahane (220950125069) the bonafide students of Diploma in Big Data Analytics and Diploma IT Infrastructure, Systems and Security of Centre for Development of Advanced Computing, Electronic City, Bangalore from 15th January 2023 – 15th March 2023. The Course End Project work is carried out under my direct supervision and 80% completed.

**Mrs.Sukeshini Ramadasu**

C-DAC #68, Electronic City,

Bangalore - 560100, India

##### ABSTRACT

With the aging of the population, the incidence of eye diseases is expected to rise significantly. Early diagnosis and proper management are crucial for preserving vision and improving the quality of life of affected individuals. Artificial intelligence (AI) integration in ophthalmology holds immense potential to accelerate the diagnostic process and reduce the need for human resources.This project focuses on the classification of visual impairment diseases such as cataract and glaucoma using convolutional neural networks (CNNs). We utilized a publicly available image dataset and tested four different CNN meta-architectures, including InceptionV3,Resnet50,MobileNetV2,CNN-Sequential(From scratch) and EfficientNetB3, using the TensorFlow object detection framework. Our objective was to identify the most accurate and computationally efficient model for classifying visual impairment diseases.

Our results showed that EfficientNetB3 was the most suitable model for classifying visual impairment diseases due to its superior performance in terms of accuracy and computational efficiency. To provide a user-friendly interface for the model, we integrated it with Streamlit.

**TABLE OF CONTENT**

[1. Introduction 1](#_Toc19461)

2. Literature Survey 5

3. Software Requirement Specification 10

3.1Operating System 10

3.2 External Interface Requirements 11

4. Data Pre-processing 11

4.1 Dataset Structure 11

4.2 Data Processing 13

4.3 Data Split 14

5. Methodology 15

5.1 Implementation of CNN Model 15

5.2 Layers in CNN Model 16

5.3 Base Model 18

5.4 Model Evaluation 22

6. Pretrained CNN Models 24

6.1 InceptionV3 24

6.2 ResNet50 25

6.3 MobileNetV2 26

6.4 VGG19 26

6.5 EfficientNetB3 27

7. Result 29

8. Conclusion 31

9. References 32

**CHAPTER 1**

# INTRODUCTION

Diabetic Retinopathy, Cataracts, Glaucoma prevalence are 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. 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 Dia-betic Retinopathy, Cataracts, Glaucoma 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 microa-neurysms 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%

Research Objectives

Diabetic Retinopathy, , Cataracts, Glaucoma, 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 Diabetic Retinopathy

**Research Problems**

Diabetic retinopathy , Cataracts, Glaucoma 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 ashing 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 detection 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 pre-processing, 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 x the images before using the CNN algorithms. There was not a lot of datasets created solely for the purpose of Diabetic Retinopathy, Cataracts, Glaucoma 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.

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 VGG19,ReseNet50,Inception\_V3,MobilNet-V2,CNN-Sequential(from Scratch) and EfficientNetB3. 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 own construct neural network.

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 EffiecientNet-b3 model performed the best in our study, correctly identifying the majority of the data.

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 Diabetic Retinopathy. 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 labelled 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.

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, a detailed overview of data processing is given, including information on the dataset that we used, feature engineering and data transformation. 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.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 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 SURVEY

One of the papers 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 neighbours (KNN) and so on. The data provided by Kaggle Community was used in. And in the experiment, translation, stretching, rotation, and flipping to the labelled 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 colour 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 \_ve-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 haemorrhages 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. Attaining 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 super 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 pre-processing. 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, Diabetic Retinopathy, Cataracts, Glaucoma 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 colour 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 pre-processing, feature extraction and classification. During pre-processing, 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 \_t 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: no proliferative 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 no proliferative diabetic retinopathy, (iii) severe no proliferative 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 artificials 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 Texture 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 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 pre-processing. 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 pre-processing, feature extraction, and classification. Uneven lighting in the photograph was adjusted during pre-processing. Grayscale Conversion, Adaptive Histogram Equalization, Discrete Wavelet Transform, Gaussian Matched Filter Response, and Fuzzy C-Means Clustering are some of the pre-processing approaches used for segmenting blood vessels. Radius, Diameter, Area, Arc Length, Centre Angle, and Half Area are some of the features extracted. To identify the best approach for classifying the image into their respective classifications namely, PDR, NPDR, or Normal, a support vector machine training procedure is used to examine training data. 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**

# SOFTWARE REQUIREMENT SPECIFICATION

### 3.1 Operating Environment

**Server Side:**

Operating System: Windows 7 or above

Processor: Intel i5 3.0 GHz or higher

**Client side**:

Operating System: Windows 7 or above

Processor: Intel i5 3.0 GHz or higher

**3.1.1 Design and Implementation Constraints**

* The time allotted for this project is limited to 2 months.
* The language for this project is Python.

## 3.2 External Interface Requirements

##### 3.2.1 User Interfaces

Front-End (software): HTML,CSS

Language: PYTHON

Streamlit

**Client side:**

Operating System: Windows 7 or above

Processor : Intel i5 3.0 GHz or higher

RAM: 2 GB or more

**CHAPTER 4**

# Data Preprocessing

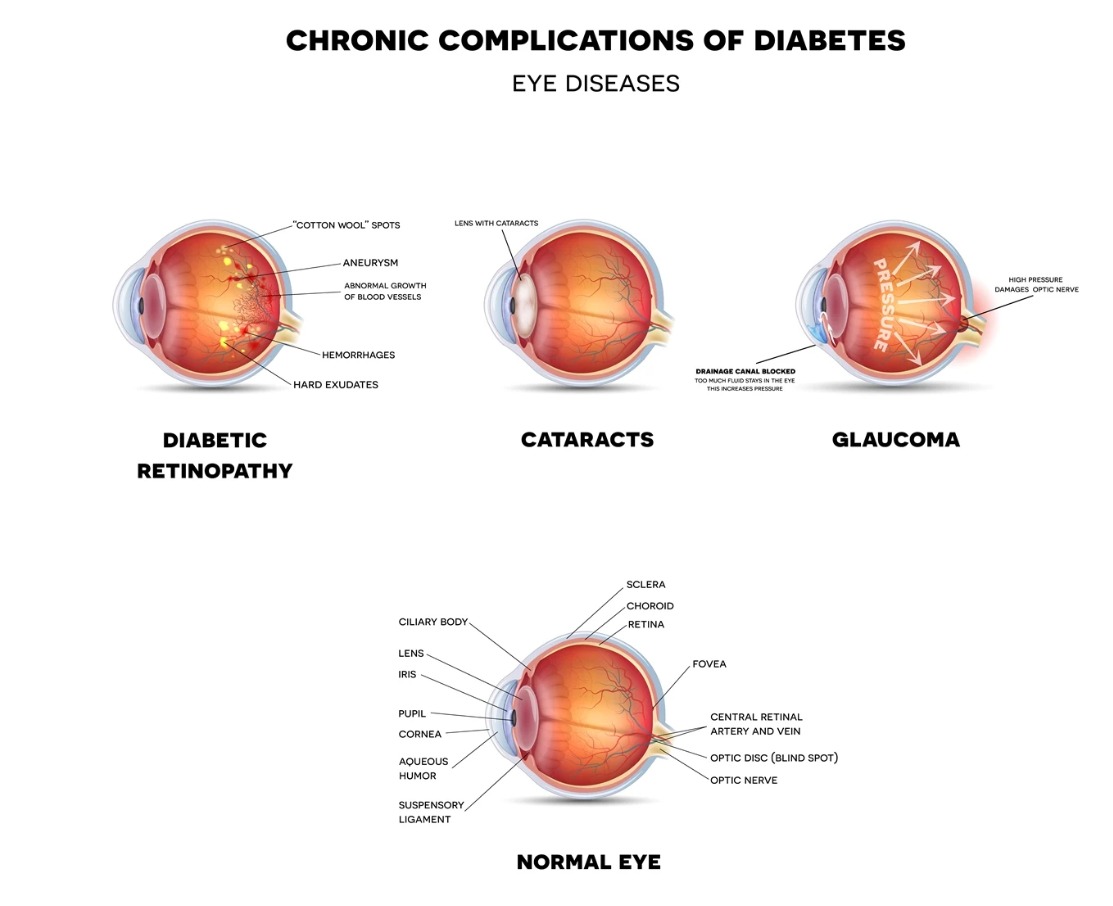
## 4.1 Dataset Structure

Diabetes is a condition that carries an increased risk of developing eye complications. Diabetic eye disease includes complications such as diabetic retinopathy, cataracts and glaucoma. Diabetes is the leading cause of blindness in working-age adults. People with type 1 and type 2 diabetes are at risk. It’s possible to be unaware that you have severe diabetic eye disease and suddenly go blind. Fortunately, most cases of blindness can be prevented with regular eye examinations and proper care. The dataset consists of Normal, Diabetic Retinopathy, Cataract and Glaucoma retinal images where each class have approximately 1000 images. These images are collected from various sources like IDRiD, Oculur recognition, HRF etc.

Diabetic retinopathy: The persistently high blood sugar levels that occur with diabetes can damage the retina’s small blood vessels (capillaries), which deliver oxygen and nutrients. Diabetic retinopathy affects up to a third of people with diabetes over the age of 50.

Cataracts: A cataract is a clouding of the lens in the eye. Left untreated, cataracts can eventually lead to blindness. People with diabetes are more likely to develop cataracts at an earlier age and suffer visual impairment faster than those without the condition.1,3

Glaucoma: This is a group of conditions that can damage the optic nerve. The optic nerve transmits signals from the retina to the brain for processing. Glaucoma is often (but not always) a result of increased pressure inside the eye. The risk of glaucoma in people with diabetes is significantly higher than that of the general population.1,4 The two main types are open-angle glaucoma (also called ‘the sneak thief of sight’) and angle-closure glaucoma (this comes on suddenly and is a medical emergency).



Data Processing

Image transformation tasks, 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. To prepare clean image data for model input, preprocessing is necessary. 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 as follows:

1. Image flip- The term " flipped " 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 ip is the same as turning a photograph 180 degrees before horizontally flipped it. When the model comes across an image that is horizontally or vertically flipped and is

a bad eye labelled 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 ip-5 Reversing an image's full row and column pixels vertically (upside down) where the flipping happens on the horizontal axis.

• Random Rotation/ flipped - 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.

## 4.2 Data Split

We have come up with an approach for evaluating the performance of the machine learning algorithm splitting the data into three parts- training, validation and testing. The approach involves dividing the dataset into three separate parts. The training dataset is the one to train the model and prepare it to obtain the optimal result. 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 out-come 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 into the existing data where the inputs and outputs are known and predict accurately the result of the data which is unknown. 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.

**CHAPTER 5**

# Methodology

Implementation of CNN models

CNN

A Convolutional Neural Network, known as CNN, is a specific kind of neural network that is optimized for the processing of input that has an architecture similar to a grid, such as an image. Neural networks are made up of many different parts, one of which is the convolutional neural network (CNN). In order to identify objects, recognize faces, and so on, CNNs employ visual recognition and classification. They are composed of neurons that may be trained to change their weights and biases. The most common usage of CNN is to categorize images, group them into clusters based on similarity, and then identify specific objects. Faces, street signs, animals, and other recognizable objects may all be recognized by algorithms that use CNN architecture. The convolutional, pooling, and fully connected layers of a CNN are the most common. The first layer of a CNN network, the Convolutional Layer,

does the majority of the computing effort. Utilizing filters or kernels to generate convolutional data or images. By adjusting the slider, we may add filters to the data. If the RGB value of the image’s depth is 4, a filter with the same depth would also be applied. For each sliding movement, a particular value is taken from each filter in the picture and added together. A 2D matrix is the result of applying a 3D color filter on a convolution with a 2D output. Down sampling features are the third step in the Pooling Layer. Every layer of the 3D volume is coated with it. Flattening is the last step in the process of creating a fully connected layer. The

neural network is given a single column of the pooled feature map matrix, which is subsequently processed. We were able to develop a model by connecting all of the layers together. We can then use an activation function like SoftMax or Sigmoid to further categorize the data generated by the algorithm. Softplus units increase DNN performance and reduce convergence time compared to sigmoid and ReLU units

Optimizer “Adam”

An optimizer is a procedure or algorithm that alters the properties of a CNN architecture,

such as the parameters and the learning rate. Examples of these properties include: As a result, it contributes to the total reduction of damage and actually increases efficiency. Adam is an extension of stochastic gradient descent, which has gained popularity for deep learning in computer vision and NLP. These include approaches for image processing and voice recognition. Reiterating the optimizer is a deep-learning strategy.

**Optimizer “Adamax”**

The Adamax optimizer is a variant of the Adam optimizer that has been shown to be particularly effective for training deep neural networks with sparse gradients. It is similar to Adam but uses the L-infinity norm to normalize the learning rate, which can help improve stability and convergence. Note that changing the optimizer can affect the performance of the model during training and evaluation, so it's often a good idea to try different optimizers and hyperparameters to find the best combination for your specific problem.

Equation explains the “Adam” optimizer function.

ωt + 1 = ωt − αmt (4.1)

mt = aggregate of gradients at time t, α = learning rate at time t, ωt = weights at

time t, ωt+1 = weights at time t + 1.

**SoftMax**

A nonlinear SoftMax output layer is commonly used when neural networks are used

for pattern classification tasks. This, as we all know, is standard procedure. Because

of its non-linearity, the soft-max output layer of a neural network has the ability to

make significant changes to the frequency at which the network generates outputs

**Layers in CNN Model**

Information processing that has a grid-like structure, such as an image, is the area of expertise of a class of neural networks called Convolutional Neural Network, which is sometimes abbreviated as CNN or ConvNet for short. A binary representation of visual data is what we refer to as a digital picture. It comprises a sequence of pixels. In the solution that has been suggested, a multi-layered deep CNN model has been used in order to differentiate between real and fake images. We have used 11 layers in our

Base model. Convolutional layers, Batch Normalization layer, max pooling layers are the foundation of a CNN model in addition to that, we have used the dropout layer and other fully connected layer such as dense layer and flatten layer in order to prevent overfitting. Below, we will go over the specifics of each layer

**Convolutional Layer**: In a Convolutional Neural Network (CNN), the Convolution layer is used to extract features from the input image. It applies a set of learnable filters to the input image to produce a set of feature maps. The parameters of a Convolution layer include filters, kernel size, activation, and padding.

**Filters**: Filters are the number of learnable kernels in the Convolution layer. Each filter is a small matrix that is applied to the input image to extract a specific feature. For example, a filter can be used to detect edges or corners in an image.

**Kernel size**: The kernel size is the size of the filter used in the Convolution layer. It is typically a square matrix with a specified width and height. The kernel size determines the receptive field of the filter and the size of the output feature map.

**Activation**: The activation function is applied to the output of the Convolution layer to introduce non-linearity into the model. It allows the model to learn more complex patterns in the input data.

**Padding**: In convolutional layer we use same padding .Padding is used to preserve the spatial dimensions of the input image when applying the filters. It adds extra rows and columns of zeros around the edge of the input image, so that the filter can be applied to the edges of the image as well.

**Batch Normalization :** Batch normalization is a technique that helps to improve the training of deep neural networks by normalizing the inputs to each layer, which reduces the internal covariate shift. This means that the distribution of the inputs to each layer is made more consistent and stable, which can help to prevent the network from getting stuck during training. Batch normalization is often used in CNNs because it can help to speed up the convergence of the network and improve its accuracy. By reducing the internal covariate shift, batch normalization can help to ensure that the features learned by the network are more robust and generalizable, which can lead to better performance on new data.

**Pooling Layer**: Following the convolutional layer in convolutional neural networks are layers known as pooling layers. In order to improve the efficiency of the computations being performed, pooling is used to reduce the amount of the features that are extracted, and therefore, the number of trainable parameters. The pooling filter defines the amount of the range that is summarized by the pooling procedure. If a filter’s parameters are 2x2, then the summary section is also 2x2 in size. Here we can detect four layers in total with other layers.

**Fully Connected Layer(FC):** This layer is comprised entirely of feed forward neural networks. Fully Connected Layers are the layers that come after the final few in the network’s architecture. After that, the output of the last pooling or convolutional layer is flattened before it is sent on to the fully connected layer as the input. This model carries the following layers:

**Flatten Layer**: Once the fourth Max Pooling layer has been used, a single flatten layer will be applied. In the end, the Flatten layer is used to convert the output of a convolutional layer into a 1-dimensional vector. This is necessary because fully connected layers in the network require their input to be a 1D vector, whereas the output of a convolutional layer is typically a 3D tensor.

**Dense Layer :** Dense layers are a powerful tool in neural network architectures, allowing the network to learn complex relationships between the input and output data. In this layer we use activation function as ‘softmax’.

**Base Model (CNN – Scratch model):**

We build a deep convolutional neural network with 11 layers .The model architecture is sequential, meaning that each layer follows the previous one in a linear fashion.

The 1st layer is a 2D convolutional layer (Conv2D) with 64 filters and a kernel size of 3x3. This layer takes the input image of size 224x224x3 and applies 64 convolutional filters to it. The output of this layer is a feature map of size 224x224x64.

The 2nd layer is also a 2D convolutional layer with 64 filters and a kernel size of 3x3. This layer takes the feature map from the previous layer as input and applies 64 convolutional filters to it. The output of this layer is another feature map of size 224x224x64.

The 3rd layer is a batch normalization layer (BatchNormalization) which normalizes the output of the previous convolutional layer. This helps in reducing overfitting and improves the training speed of the neural network.

The 4th layer is a max pooling layer (MaxPooling2D) which reduces the size of the feature map by taking the maximum value in each non-overlapping 2x2 region. This layer reduces the size of the feature map from 224x224x64 to 112x112x64.

The 5th layer is a dropout layer (Dropout) which randomly drops out some of the neurons in the layer to prevent overfitting.

The next three layers are similar to the previous layers but with 128 filters each. The convolutional layers are followed by batch normalization, max pooling, and dropout layers.

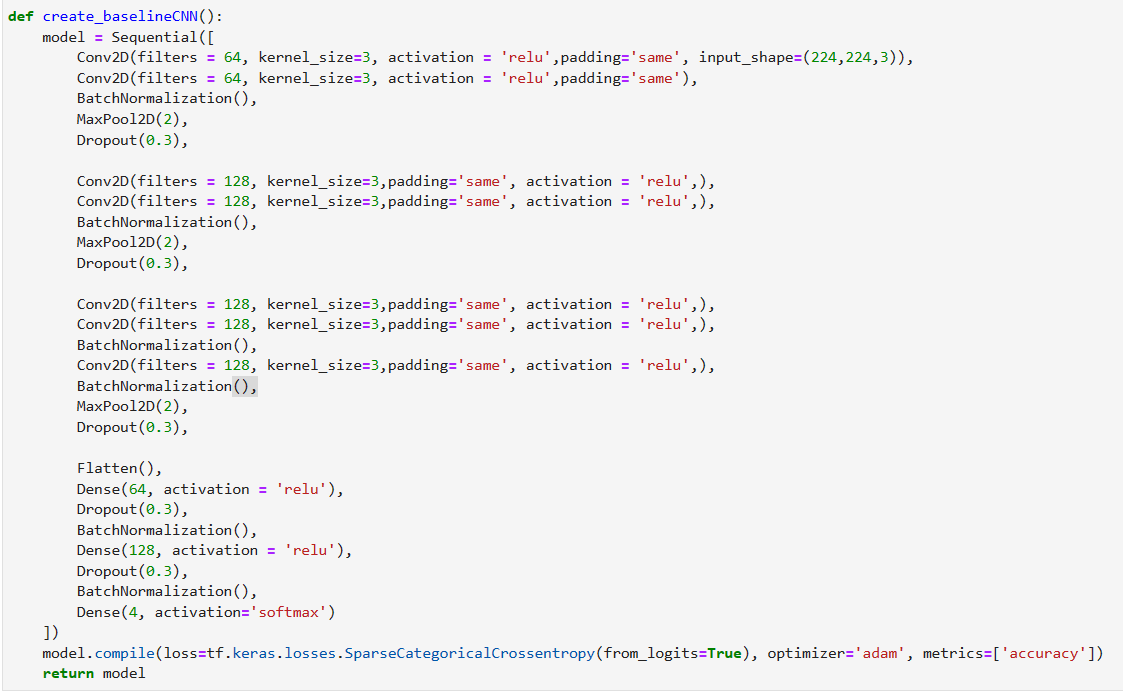
The 9th layer is a flatten layer (Flatten) which flattens the 3D feature map into a 1D vector. This is required to connect the convolutional layers with the fully connected layers.

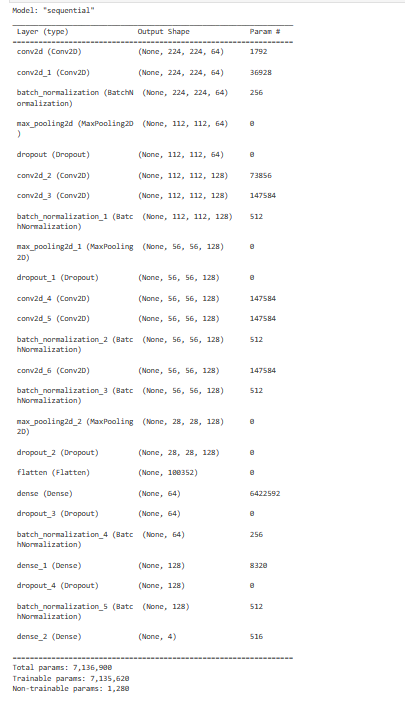
The 10th layer is a fully connected layer (Dense) with 64 neurons and an activation function. This layer takes the flattened feature vector from the previous layer as input and applies 64 neurons to it. The output of this layer is a vector of size 64.

The 11th layer is a dropout layer followed by a batch normalization layer. These layers prevent overfitting and improve the training speed of the neural network.

The 12th and final layer is a fully connected layer (Dense) with 128 neurons and an activation function. This layer takes the output of the previous layer as input and applies 128 neurons to it. The output of this layer is a vector of size 128.

The final layer is another fully connected layer with 4 neurons, which corresponds to the number of classes in the output. This layer applies 4 neurons to the input vector and outputs the class predictions.





## 5.4 Model Evaluation

**Metrics:**

**Accuracy :**

Accuracy 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.

True Positive + True Negative

Accuracy = ------------------------------------------------------------------------------------------------------------------------

True Positive + True Negative + False Positive + False Negative

**Precision :**

Precision 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.

True Positive

Precision = --------------------------------------------

True Positive + False Positive

**Recall**

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.

True Positive

Recall =---------------------------------------------

True Positive + False Negative

**Adam Optimizer**

The Adam optimizer 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 at surface error minima, which is what Adam prefers.

**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.

**Pre-trained CNN Models**

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.



InceptionV3 Architecture

**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.



ResNet-50 Architecture

MobileNet\_V2

In order to reduce the size of the model and the complexity of the network, Mobile Nets are a depth-wise separable convolution design that reduces the number of connections. Embedded and mobile applications benefit from the technology. The author has included two global hyperparameters into this sort of network, which are as follows: A good balance between model latency and accuracy is achieved with this technique. In addition, the hyperparameters give the capability of selecting a suitably scaled model in accordance with the problem restrictions, if necessary.

**VGG19**

VGG19 is a convolutional neural network architecture for image classification that was proposed by researchers at the University of Oxford. It consists of 19 layers, including 16 convolutional layers and 3 fully connected layers. The architecture is known for its simplicity and achieved state-of-the-art performance on the ImageNet dataset. VGG19 uses small 3x3 filters throughout the network, which allows for a more detailed representation of the image features. It also uses max pooling layers to reduce the size of the feature maps and increase computational efficiency. Overall, VGG19 is a powerful and widely-used deep learning model for image classification task



**EfficientNet-b3:**

**\*** EfficientNetB3 is one of the models in the EfficientNet family of convolutional neural networks. It was introduced by Tan et al. in their paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" (2019) and is designed to achieve high accuracy while maintaining computational efficiency.

\* EfficientNetB3 has 25 million parameters and is based on a compound scaling method that uniformly scales the depth, width, and resolution of the network. Specifically, EfficientNetB3 scales up the base architecture of EfficientNetB0 by increasing the depth, width, and resolution of the network.

\* The architecture of EfficientNetB3 consists of 30 layers, including a stem, multiple blocks, and a top classifier. The stem includes a sequence of convolutional, pooling, and activation layers that extract features from the input image. The blocks are composed of a sequence of depth wise and pointwise convolutions that further extract features and reduce the spatial dimension of the input. The top classifier is a fully connected layer that maps the extracted features to the output classes.

\* EfficientNetB3 achieves state-of-the-art performance on a range of computer vision tasks, including image classification, object detection, and segmentation. It has been pre-trained on large datasets such as ImageNet and can be fine-tuned on specific tasks with smaller datasets. EfficientNetB3 is a larger and more complex variant of EfficientNetB0, with more layers and more parameters. It was designed to provide better accuracy and performance on large-scale image classification tasks compared to EfficientNetB0**.**

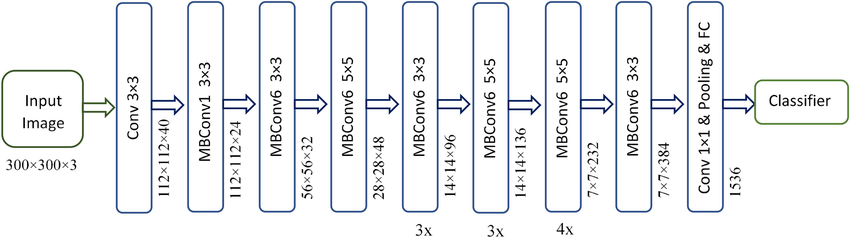
**Why EfficientNetB3 is better than other models**

1. Higher Accuracy: EfficientNetB3 is a larger model and has more parameters, which allows it to capture more complex features in the images. This can lead to better accuracy on image classification tasks.

2. Better Performance on Large-Scale Datasets: EfficientNetB3 was designed to perform well on large-scale image classification tasks such as the ImageNet dataset.

3. More Powerful Features: The additional layers and parameters in EfficientNetB3 allow it to capture more powerful and complex features in the images. This can lead to better performance on tasks that require more complex visual understanding, such as object detection and segmentation.

4. Transfer Learning: EfficientNetB3 can be used as a pre-trained model for transfer learning, which allows us to use the pre-trained weights and features to train a model on a new dataset. The larger size and complexity of EfficientNetB3 can make it a more effective pre-trained model for transfer learning tasks**.**



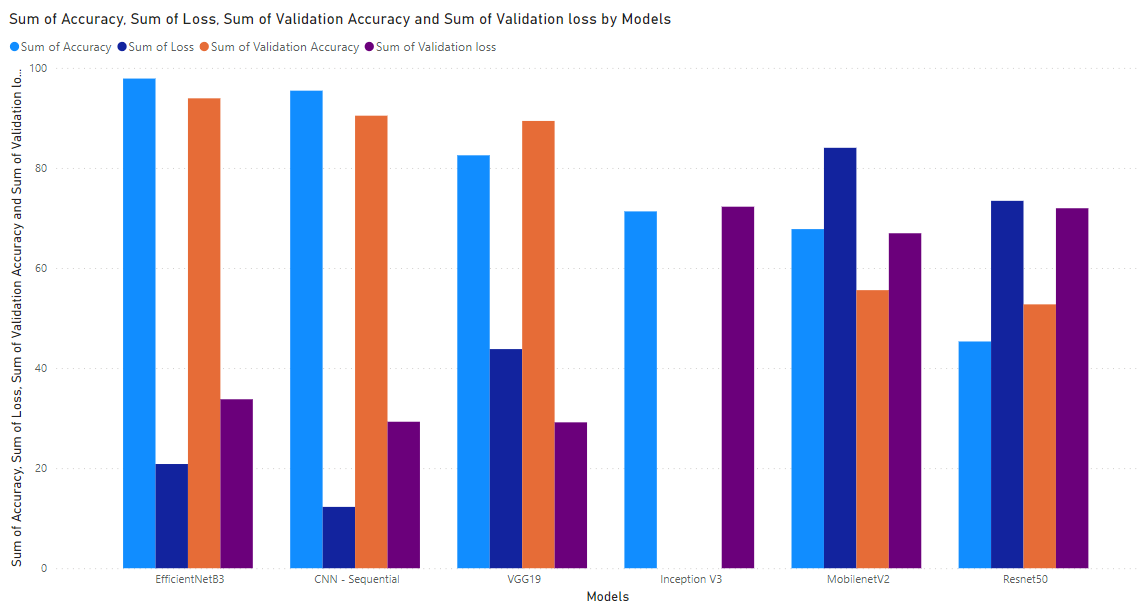
**CHAPTER 6**

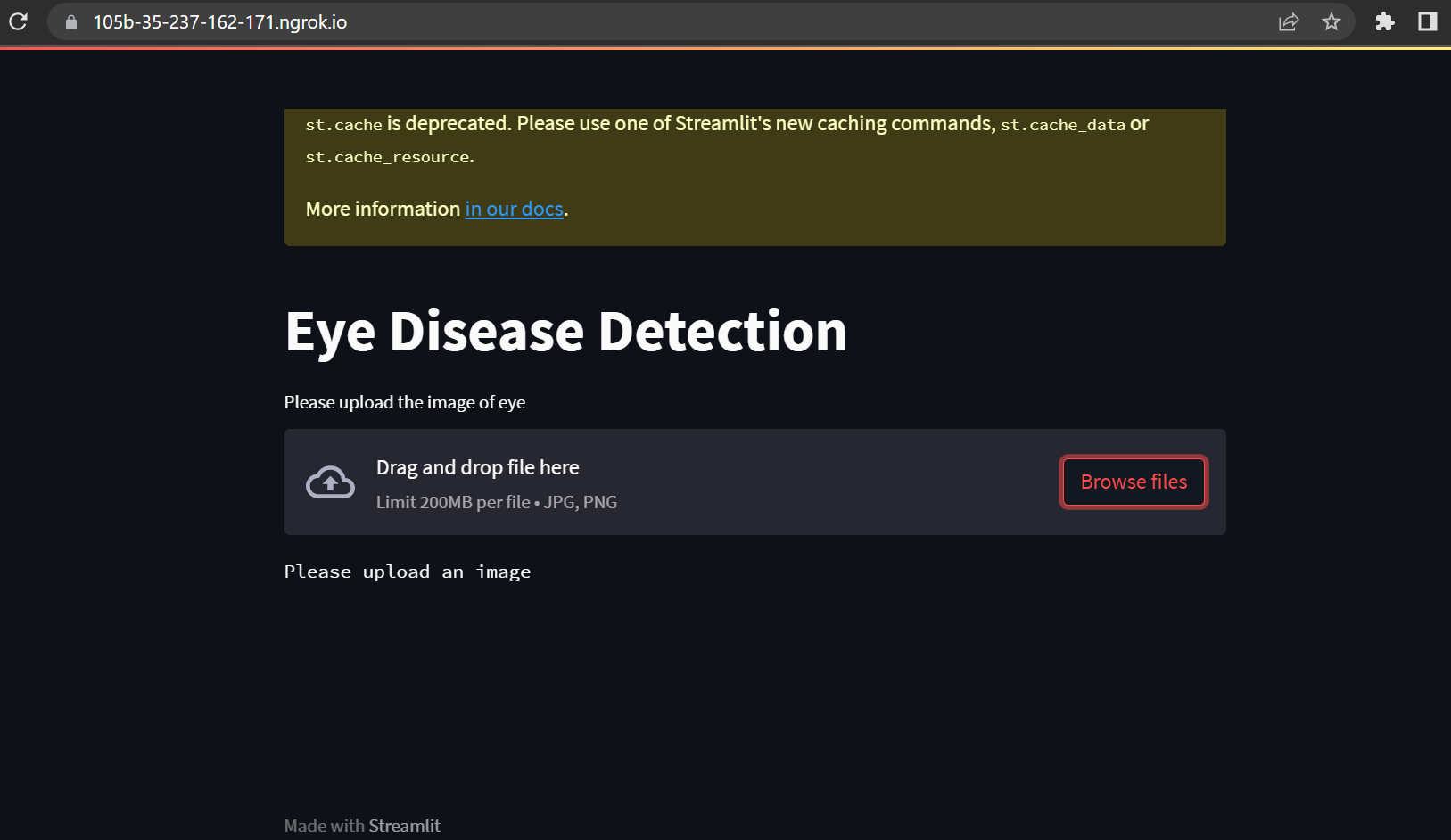
# RESULT

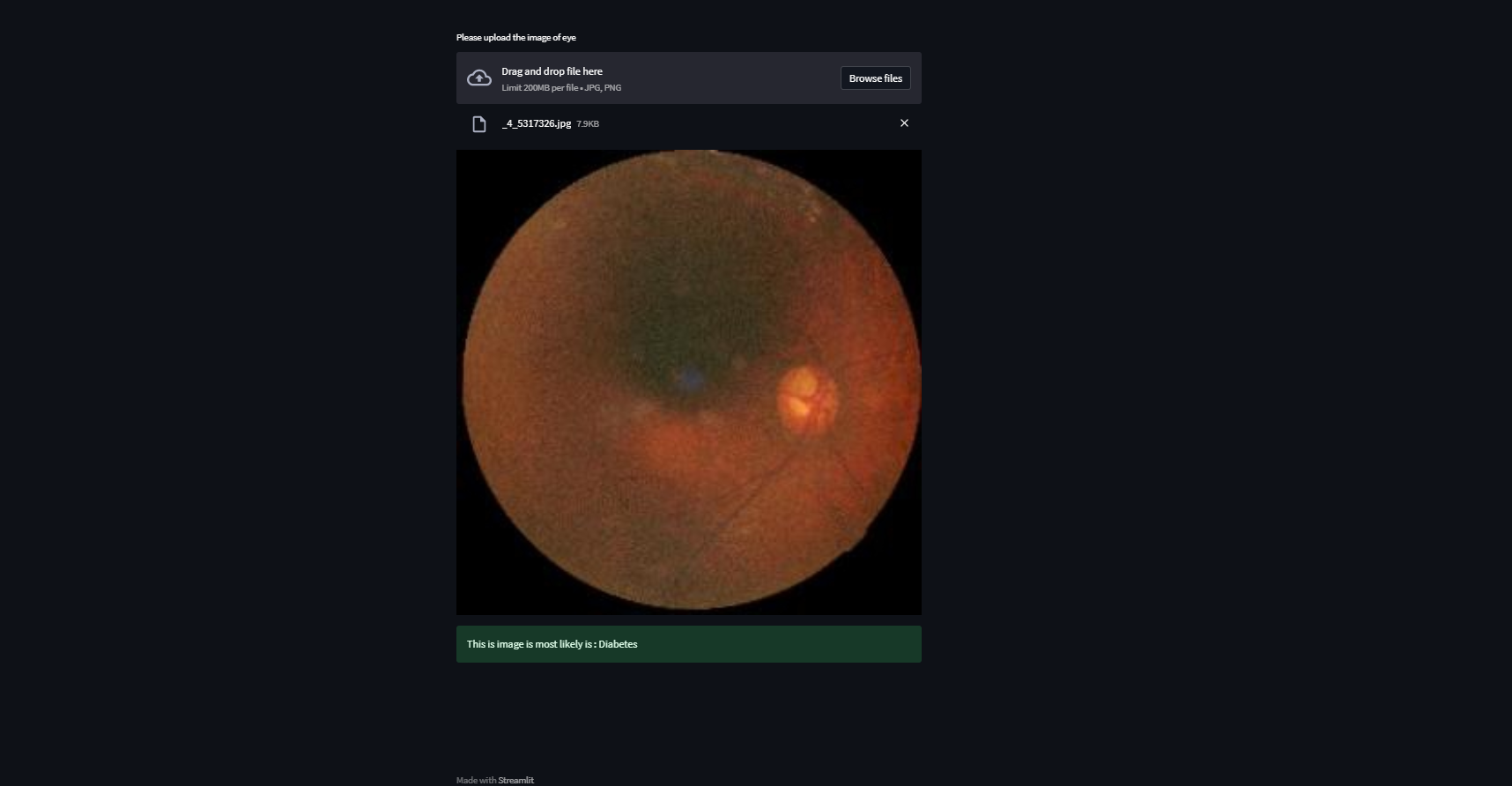
Our approach of comparing the transfer learning models and the CNN-Sequential proved to be effective in finding the optimal model that works best for the disease of diabetic retinopathy. Since the disease has been prevalent for a long period and is one of the major causes of blindness, early detection 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 CNN-Sequential deep neural model with ‘relu’ activation function in convolutional layer and ‘softmax’ activation function in dense layer ,we got an accuracy of 90% , but we got lot of fluctuations in the loss and accuracy graph. As the graph was not consistent we used EfficientNetB3 pretrained model with “adamax” activation function. Through EfficientNetB3 we got accuracy of 93% with minimum fluctuations in loss graph.

For the approaches, we have used 5 pre-trained and 1 CNN-Sequential(from scratch) 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. The number of epochs used to train the model varied for the approaches we used in VGG19,ReseNet50,Inception\_V3,MobilNet-V2,CNN-Sequential(from Scratch) and EfficientNetB3.

****

****





**Chapter 7**

# CONCLUSION

This study outlines a procedure that, when it comes to the treatment and prevention of eye diseases, can be of great use to ophthalmologists. According to the findings of the research article, this system has a detection rate of 93.92% and may detect these diseases in their earliest stages, hence assisting individuals in avoiding the potential consequences. We use our recommended CNN model to analyze the data, which consists of 3000 OCT images that have been divided into three portions and pre-processed. This allows the model to distinguish between images that are clean and those that include faults. In conclusion, we are able to establish that the CNN model that was suggested is an improved method for diagnosing eye disorders based on OCT data. The accuracy of the model that was proposed is noticeably greater than that of the other methodologies that were considered. It is possible that ophthalmologists who use this technology will be able to carry out effective retinal image analysis. This will allow them to provide improved treatment at an earlier stage in the progression of the disease, thereby preventing blindness. Because of this, we intend to increase the effectiveness of our model in the coming years.

**Chapter 8**

**REFRENCES**

[1] Z. Reitermanova, \Data splitting," in WDS, vol. 10, 2010, pp. 31{36.

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

[3] 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>.

[4] ||, \Dropout: A simple way to prevent neural networks from overfitting, "The journal of machine learning research, vol. 15, no. 1, pp. 1929{1958, 2014.

[5] 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>.

[6] 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>.

[7] F. Chollet, \Xception: Deep learning with depth wise separable convolutions, "CoRR, vol. abs/1610.02357, 2016. arXiv: 1610.02357. [Online].

Available: http: //arxiv.org/abs/1610.02357.

[8] 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.

[9] 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.

[10] ||, \Convolutional neural networks for diabetic retinopathy," Procedia computer science, vol. 90, pp. 200{205, 2016.

[11] 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, 20

[12] 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.

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

[14] 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.

[15] Aug. 2018. [Online]. Available: <https://docs.microsoft.com/en-us/analysis>

services/data- mining/training-and- testing- data- sets?view=asallproducts-allversions.

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

[17] M. ul Hassan, \Vgg16-convolutional network for classification and detection," en linea].[consulta: 10 april 2019].Disponible

en: https://neurohive.io/en/popular-networks/vgg16, 2018.

[18] 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.

[19] 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.

[20] 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.

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

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

[23] Jiwon, The most intuitive and easiest guide for convolutional neural network, Jan. 019. [Online]. Available: https://towardsdatascience.com/the- most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480.

[24] 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)

[25] 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.

[26] ||, \Neural network technique for diabetic retinopathy detection," Int J Eng Adv Technol (IJEAT), vol. 8, no. 6, pp. 440{445, 2019.

[27] S. Qummar, F. G. Khan, S. Shah, A. Khan, S. Shamshir band, 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.

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

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

[31] F. Shaikh, Deep learning in the trenches: Understanding inception network

from scratch.

[32] Understanding the vgg19 architecture. [Online]. Available: https://iq.opengenus.

org/vgg19-architecture/.