PROJECT REPORT
ON

Deep Learning based Web App for diagnosing multiple visual impairment diseases

Carried Out at

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING
ELECTRONIC CITY, BANGALORE

UNDER THE SUPERVISION OF
Miss. Sukeshani Ramadasu

C-DAC Bangalore

Presented By

Souravkumar Shukla	PRN:220950125095
Shubham Ganesh Kore	PRN:220950125093
Siddhesh Shridhar Chowkekar	PRN:220950125094
Sudarshan Rajendra Kulawade	PRN:220950125097
Aditya Ganesh Survase	PRN:220950125098

PG DIPLOMA IN BIG DATA ANALYTICS
C-DAC, BANGALORE

Candidate's Declaration

We hereby certify that the work being presented in the report titled: "**Deep Learning based Web App for Diagnosing multiple visual impairment diseases"**, in partial fulfilment of the requirements for the award of PG Diploma Certificate and submitted to the department of PG- DBDA of the C-DAC ACTS Bangalore, is an authentic record of our work carried out during the period, 01/01/2023 to 10/03/2023 under the supervision of Mrs. Sukeshani 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:

Souravkumar Shukla	PRN:220950125095
Shubham Ganesh Kore	PRN:220950125093
Siddhesh Shridhar Chowkekar	PRN:220950125094
Sudarshan Rajendra Kulawade	PRN:220950125097
Aditya Ganesh Survase	PRN:220950125098

Counter Signed by:

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Souravkumar Shukla

Shubham Ganesh Kore

Siddhesh Shridhar Chowkekar

Sudarshan Rajendra Kulawade

Aditya Ganesh Survase

Have successfully completed their project on

Deep Learning based Web App for Diagnosing multiple visual impairment diseases

Under the guidance of

Miss. Sukeshani Ramadasu

Miss. Sukeshani Ramadasu	**Mrs. Arun Shankar**
(Project Guide)	**(Course Co-ordinator)**

ACKNOWLEDGMENT

This project "**Deep Learning based Web App for Diagnosing multiple visual impairment diseases**" was a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS).

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

We pay thanks to Miss. Sukeshani Ramadasu who has given guidance and alight to us during this project. His versatile knowledge about "**Deep Learning based Web App for Diagnosing multiple visual impairment diseases**" has eased us in the critical times during the span of this Final Project.

We acknowledge here support of our friends and seniors specially who contributed significantly to one or more steps.

We have tried our best to keep report simple yet technically correct. We hope we succeed in our attempt. We take full responsibility for any remaining sins of omission and commission.

From

The whole team.

Index

List of Figures

List of Tables

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), Cataracts, Glaucoma 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 pre trained models like EfficientnetB0, InceptionV3, Resnet50, VGG16 and Moblienet 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, Efficientnet has reached an accuracy of 90% which is a state-of-the-art result for the used dataset. In addition, our proposed model reached an accuracy of 80%, 78%,87% which worked better on other pre-trained models.

Keywords: Data Pre-processing; Convolutional Neural Network ;pre-trained Models Efficientnet; Inception ;TensorFlow Hub ; Keras

Introduction

Background

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, 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 DR.

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 classification using the models Efficientnet, Inception, ResNet-50, VGG-16, 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 pre-trained models only.

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 classifications and Four different pretrained models and also our proposed model. To summarize, our study offers a potential method to demonstrate that, of all the models tested, the Efficientnet 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 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 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. 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.

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.

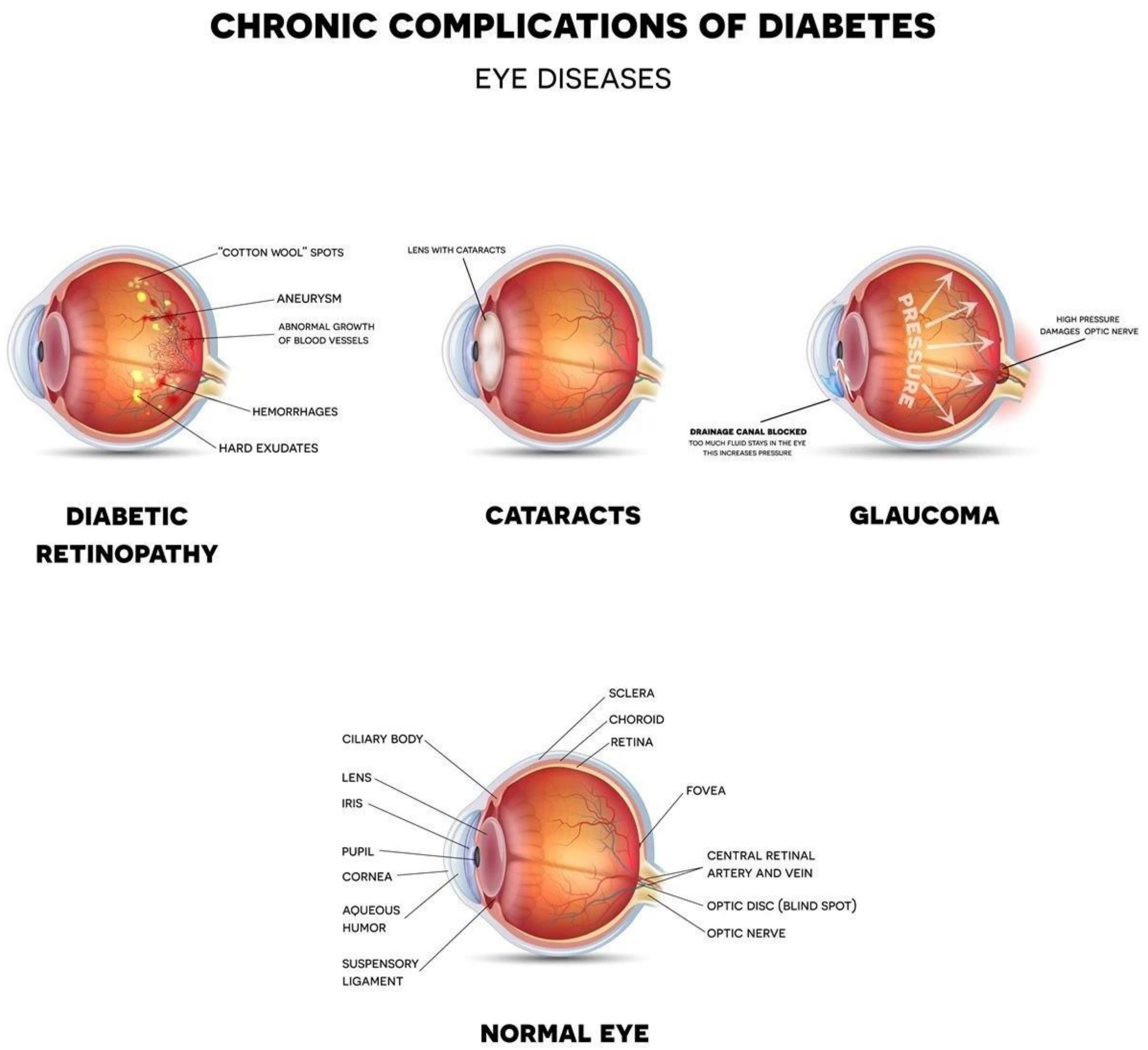
The dataset consists of Normal, Diabetic Retinopathy, Cataract and Glaucoma retinal images where each class have approximately 1000 images. These images are collected from various sources like IDRiD, Ocular recognition, HRF etc.

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 502.

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).

Fig1:-


CHRONIC COMPLICATIONS OF DIABETES
EYE DISEASES
"COTTON WOOL" SPOTS
ANEURYSM
ABNORMAL GROWTH OF BLOOD VESSELS
HEMORRHAGES
HARD EXUDATES
DIABETIC RETINOPATHY
LENS WITH CATARACTS
CATARACTS
PRESSURE
HIGH PRESSURE DAMAGES OPTIC NERVE
DRAINAGE CANAL BLOCKED
TOO MUCH FLUID STAYS IN THE EYE
THIS INCREASES PRESSURE
GLAUCOMA
SCLERA
CHOROID
CILIARY BODY
RETINA
LENS
IRIS
FOVEA
PUPIL
CORNEA
CENTRAL RETINAL ARTERY AND VEIN
AQUEOUS HUMOR
OPTIC DISC (BLIND SPOT)
OPTIC NERVE
SUSPENSORY LIGAMENT
NORMAL EYE

Data preprocessing

Data preprocessing is a crucial step in building any deep learning model, including for detecting eye diseases. Here are some common steps you might take for preprocessing your data:

Data Cleaning: First, it is important to clean the data by removing any missing values, outliers or errors that might affect the performance of the model. You can use tools such as Pandas or NumPy to handle this.

Data Normalization: In deep learning, it is common to normalize the data to make sure that all input features are in the same range. This can be achieved by scaling the values of each feature to a specific range, such as [0,1] or [-1,1].

Data Augmentation: Augmenting the data by adding noise or perturbing the images can help increase the size of the dataset and improve the model's ability to generalize to new images. This can include flipping, rotation, scaling, and shifting of the images.

Data Resizing: Most deep learning models require a fixed size input. Resizing images to the same dimensions can be achieved using tools such as OpenCV or PIL.

Data Splitting: It is essential to split the data into training, validation, and testing sets. The training set is used to train the model, while the validation set is used to tune the hyperparameters and prevent overfitting. Finally, the testing set is used to evaluate the performance of the model on unseen data.

Data Encoding: Categorical variables such as the type of eye disease can be encoded using one-hot encoding or label encoding.

By performing these preprocessing steps, you can ensure that your data is ready for training a deep learning model to detect eye diseases.

PHASE 1 : Data Loading, Cleaning, and Split

Steps followed

The two main libraries used are Pandas and TesorFlow

1. Create a new google colab notebook.
2. Download Dataset from kaggle.
3. Create relationship between google colab and dataset from Kaggle.
4. Import the needed Data, packages, and libraries.
5. Data in different folders so extraction of path for each folder.
6. Loading all dependencies like regular expression, pandas
7. Create Four Classes to depend on our disease.
8. Labeling the four classes.
9. Then Spilt the Data Train, Test and Validation.
10. Given the Image size for Images.
11. Use or apply some TensorFlow functions on Images in our Dataset.
12. Divide the Data into Batches.
13. Creating Test, Train and Validation batches for data.

PHASE 2 : Apply Pre-trained Models On data

The two main libraries used are Keras and TesorFlow

1. Import Keras libraries for pre-trained models.
2. Set the Image size.
3. Import pre-trained Models urls from TensorflowHub.
4. Create Neural Network for the models.
5. Add pre-trained models add layers sequentially.
6. Use Activation functions for layers.
7. Model compile use Loss functions and optimizers.
8. Summarize the model,
9. Run that model. Running the epochs.
10. Get train and validation Accuracy and loss.
11. Save the model.
12. Get test Accuracy and confusion matrix to evaluate Model.
13. For more evolution get precision_score,total_recall,F1score.
14. Plot The graph
15. For checking result deploying model on Flask as well as desktop App.

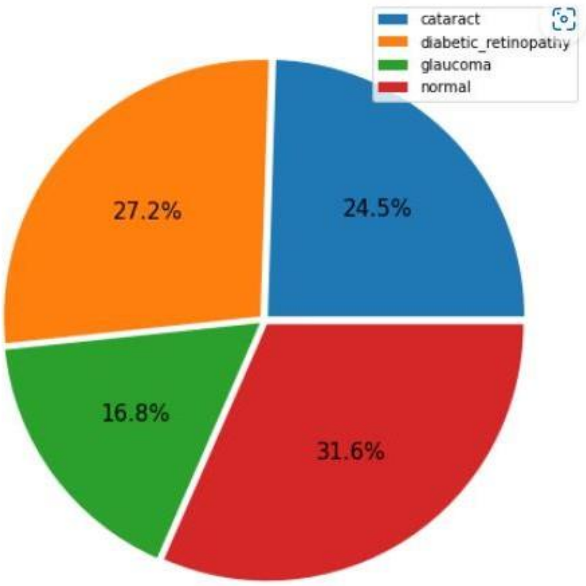


Fig2:- Classes of our Dataset

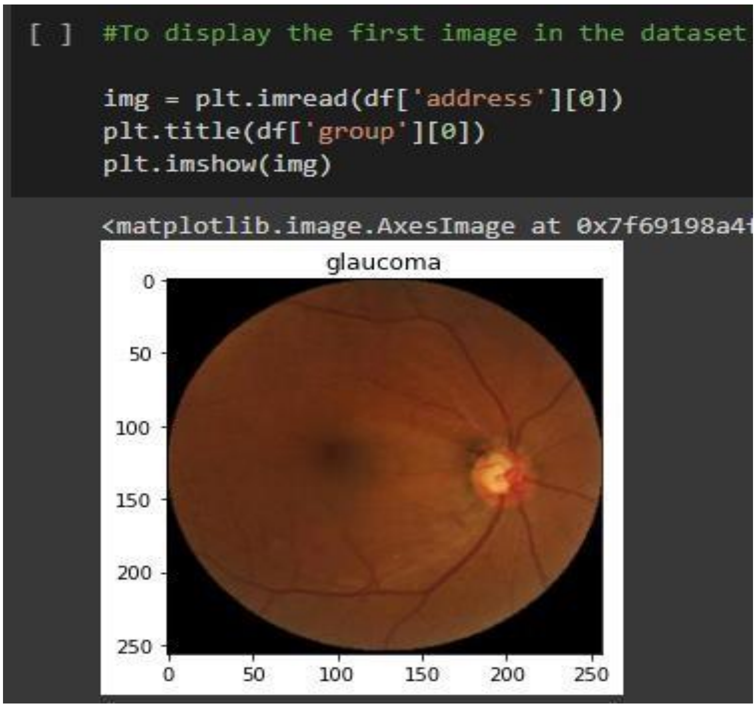


Fig3:- one diseases image of our dataset.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 1000)              5330564

 flatten (Flatten)           (None, 1000)              0

 dense (Dense)               (None, 256)               256256

 dense_1 (Dense)             (None, 256)               65792

 dense_2 (Dense)             (None, 4)                 1028

=================================================================
Total params: 5,653,640
Trainable params: 323,076
Non-trainable params: 5,330,564
_________________________________________________________________
```

Fig4:- Model Summery of our Model.

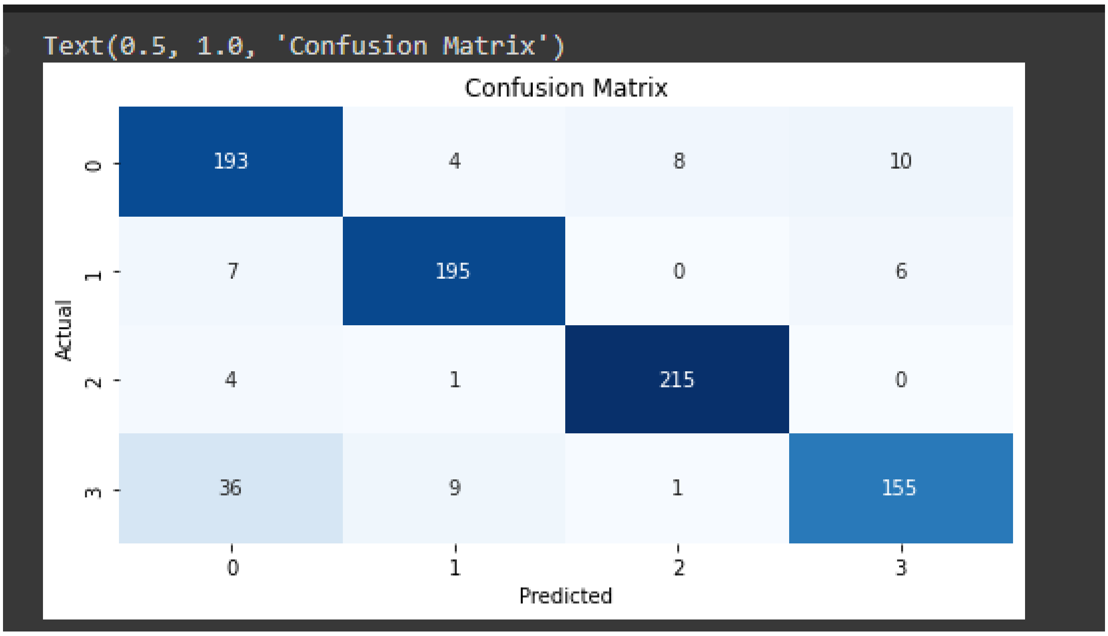


Fig5:-confusion Matrix values for EfficientNetB0

PHASE 3 : DL algorithm and Model Creation Result

Table1:- pre-trained models and its Accuracy

Efficientnet_B0		
Test Accuracy- 0.9087	Train Accuracy:-0.9370 Loss-:0.1360	Val Accuracy--: 0.9037 Loss:-0.2467
Moblienet		
Test Accuracy- 0.8755	Train Accuracy:-0.9540 Loss:-0.1213	Val Accuracy-: 0.8859 Loss:-0.3480
Resnet_50		
Test Accuracy:- 0.8023	Train Accuracy-:0.8617 Loss:-0.3533	Val Accuracy-:0.8030 Loss:-0.4957
Inception_v3		
Test Accuracy:- 0.7819	Train Accuracy-:0.9540 Loss-:0.1215	Val Accuracy-:0.7941 Loss-:0.6347

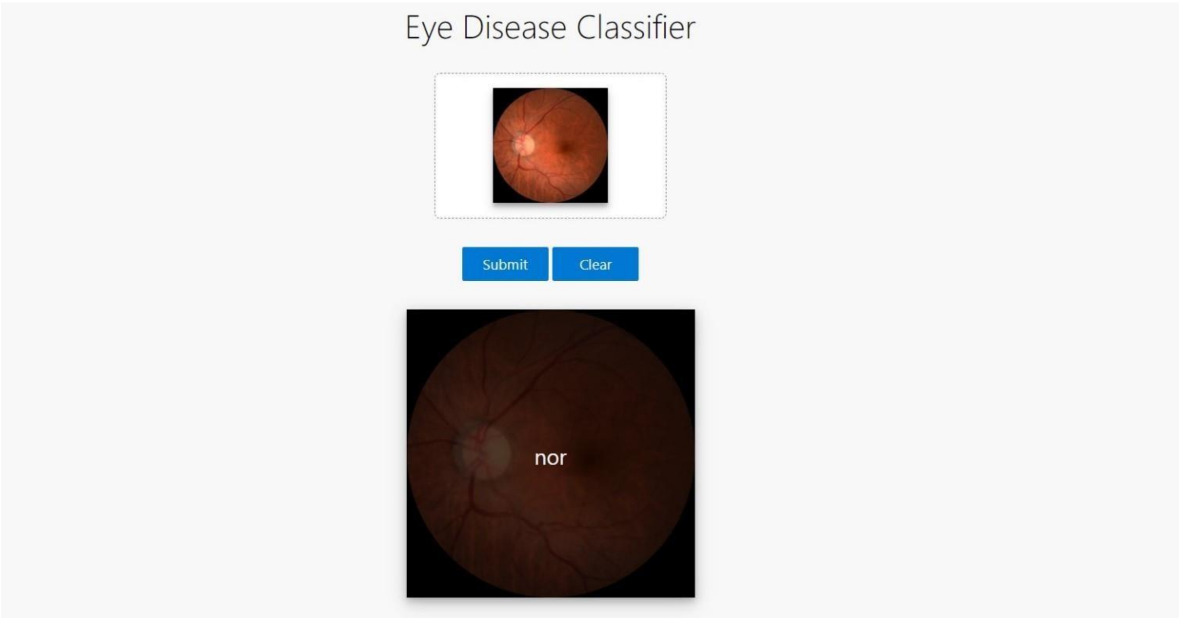
PHASE 4 : Deployment of Model

Detecting eye diseases using a deep learning model deployed on Flask can be a powerful tool to assist medical professionals in diagnosing eye conditions. Here are the general steps to create such a system:

1. Collect and label a dataset of eye images that include various eye diseases such as cataracts, glaucoma, macular degeneration, diabetic retinopathy, etc. You can also include normal, healthy eye images to help the model differentiate between healthy and diseased eyes.
2. Preprocess the data to normalize the size and color of the images. You may also want to perform data augmentation to increase the size of the dataset.
3. Train a deep learning model on the labeled dataset using a convolutional neural network (CNN) architecture. You can use libraries such as Keras or TensorFlow to create and train the model.
4. Save the trained model and create a Flask web application to deploy the model. Flask is a Python web framework that is easy to use and can be deployed on a cloud platform such as Amazon Web Services or Google Cloud Platform.
5. Create an HTML form where users can upload eye images for analysis. The Flask application should take the uploaded image and pass it through the trained deep learning model to make a prediction on the eye disease present in the image.
6. Display the prediction result on the web page and provide additional information about the eye disease for the user to read.
7. Test and evaluate the performance of the model and the Flask application. You may need to fine-tune the model and make improvements to the application to improve accuracy and speed.

Overall, building a deep learning model to detect eye diseases and deploying it on Flask can be a powerful tool to assist medical professionals in diagnosing eye conditions. It can also be a valuable resource for individuals to get a preliminary assessment of their eye health before seeing a doctor.

Fig6:-Web App Development



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.

21-Layered 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 21 layers in our model. Convolutional layers, 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: An essential component of a CNN is a convolutional layer. All the settings for these filters (or kernels) must be learned during the training process. It's common for filters to be smaller in size than the image they are intended to enhance. This layer uses kernel filters to extract essential information from the input images that are convolutionally processed. The kernel filters are similar to the input images, but they have lower constant parameters. Edge detection, blurring, and sharpening can be accomplished through the convolution of an image with several filters. In the convolutional layer, we used the Conv2D layer to construct this CNN model. The model was built with a total of ten Conv 2D layers.

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(FNN) 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 MaxPooling layer has been used, a single flatten layer will be applied. In the end, this is beneficial for the network as a whole in general.

Dense Layer: In addition to the flattening layer, this model has two dense layers. The outputs of previous levels are sent to all neurons in this layer.

Dropout Layer: During the training process, this layer will periodically reset all of the inputs to zero as a means of reducing the possibility of the model being too accurate.

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.

Activation function:-

<u>ReLU:-</u>

The Rectified Linear Unit (ReLU) activation function is a popular non-linear activation function used in artificial neural networks. It is defined as:

f(x) = max(0, x)

where x is the input to the function

The ReLU activation function is a piecewise linear function that maps negative inputs to zero and positive inputs to their original values. It is simple and computationally efficient, which makes it a popular choice in deep learning.

One advantage of using ReLU is that it helps address the vanishing gradient problem, which is a common issue in deep neural networks. The vanishing gradient problem occurs when the gradient of the loss function with respect to the parameters of the network becomes very small, which makes it difficult for the network to learn. ReLU's linear behavior for positive inputs ensures that the gradient doesn't vanish for positive inputs, which can help the network to learn more effectively.

Overall, ReLU is a powerful activation function that can help improve the performance of deep neural networks. However, it is not suitable for all types of problems and architectures, and other activation functions may be more appropriate depending on the task at hand.

Adam Optimizer:-

Adam optimizer is a popular optimization algorithm used in machine learning and deep learning. It is an extension of stochastic gradient descent (SGD) and is designed to be more efficient and effective in training deep neural networks.

The name "Adam" stands for Adaptive Moment Estimation, which refers to the way the algorithm adapts the learning rate during training. Adam computes individual adaptive learning rates for different parameters from estimates of the first and second moments of the gradients. This approach helps the optimizer to converge faster and more reliably than traditional gradient descent methods.

In addition to the adaptive learning rate, Adam optimizer also includes momentum, which smooths out the optimization process and helps the optimizer to avoid getting stuck in local minima.

Overall, Adam optimizer has become a popular choice for optimizing deep learning models due to its ability to converge quickly and efficiently on a wide range of optimization problems.

Loss Function:

CategoricalCrossentropy:-

CategoricalCrossentropy is a common loss function used in deep learning models for multi-class classification problems. It measures the difference between the predicted probability distribution and the true probability distribution.

In practice, the CategoricalCrossentropy loss is often used in combination with the softmax activation function in the output layer of a neural network. The softmax function normalizes the predicted probabilities so that they sum to 1, and the CategoricalCrossentropy loss measures the difference between these normalized probabilities and the true class labels.

Pre-Trained Models:-

1. EfficientNet B0:-

EfficientNet B0 is a convolutional neural network architecture designed to achieve high accuracy while maintaining efficiency in terms of computational resources and model size. It was introduced in the paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" by Tan et al. in 2019.

The architecture of EfficientNet B0 is based on a compound scaling method that uses a combination of depth, width, and resolution scaling to balance the trade-off between model size and accuracy. It starts with a base convolutional neural network (CNN) architecture and scales it up or down by adjusting the number of layers, width of the layers, and resolution of the input image.

EfficientNet B0 has 7.8 million parameters and achieves state-of-the-art accuracy on several image classification tasks, while being up to 8.4x smaller and up to 6.1x faster than previous state-of-the-art models like ResNet-50 and ResNeXt-101. The model has been pre-trained on the ImageNet dataset and can be fine-tuned for various computer vision tasks such as object detection, segmentation, and image captioning.

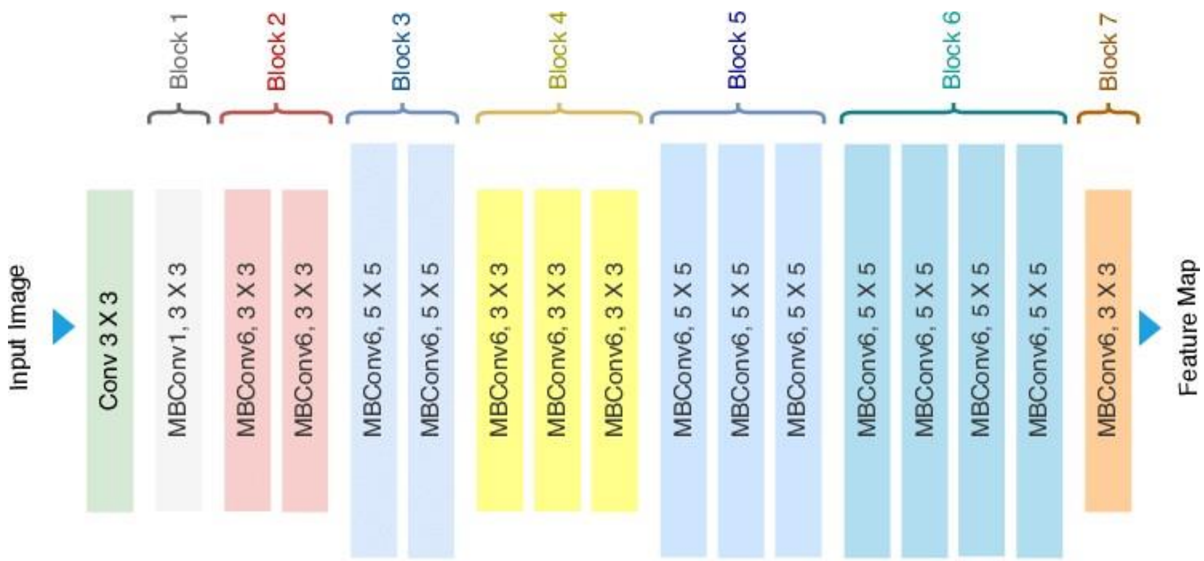


Fig7:- EfficientNet B0 model

2. Inception v3:-

Inception v3 is a convolutional neural network (CNN) architecture designed for image classification tasks. It is part of the Inception family of neural networks, which were developed by Google researchers.

Inception v3 has a deep architecture with 48 convolutional layers, and it incorporates several innovative features that improve its performance over previous versions. One of the most important features of Inception v3 is the use of "inception modules," which are blocks of layers that allow the network to capture features at different spatial scales.

Another key feature of Inception v3 is the use of "factorization," which involves breaking down large convolutions into smaller ones, making the network more computationally efficient. In addition, Inception v3 uses batch normalization and auxiliary classifiers, which help improve the accuracy of the network.

Inception v3 has achieved state-of-the-art performance on a number of image classification benchmarks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and the Common Objects in Context (COCO) dataset. It is a powerful tool for image classification and has applications in areas such as computer vision, autonomous vehicles, and medical imaging.

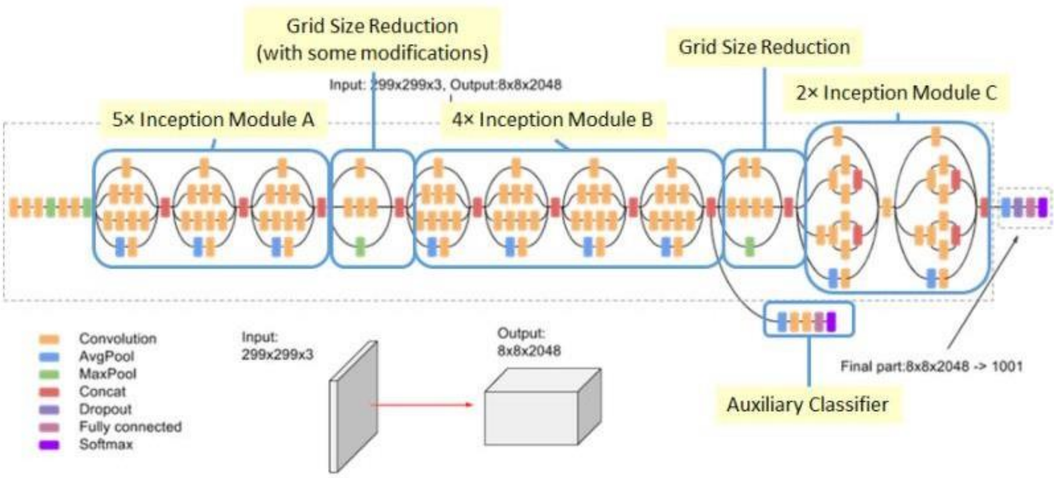


Fig8:- Inception V3 architecture

3. ResNet50:-

ResNet50 is a type of deep neural network architecture that was developed by Microsoft Research Asia in 2015. It is part of the larger family of ResNet (short for "Residual Networks") architectures and is one of the most popular and widely-used deep learning models in computer vision applications.

The ResNet50 model is made up of 50 convolutional layers and is capable of processing high-resolution images with millions of pixels. It uses skip connections, or residual connections, that allow information to flow directly from one layer to another without being processed by the intermediate layers. This approach helps to address the problem of vanishing gradients and makes it easier to train very deep neural networks.

ResNet50 has achieved state-of-the-art performance on a wide range of image classification tasks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It has also been used in a variety of applications, such as object detection, semantic segmentation, and image captioning.

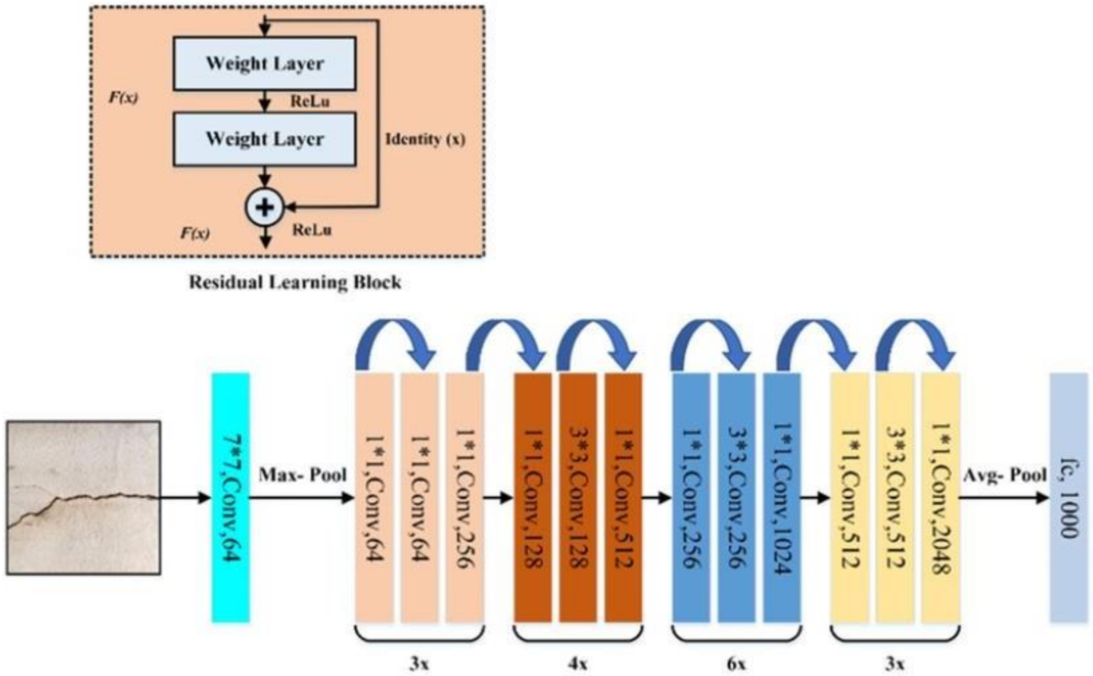


Fig9:- ResNet50 architecture

4. MobileNet:-

MobileNet is a convolutional neural network architecture designed for mobile devices with limited computational resources, such as smartphones and embedded devices. It was developed by Google researchers in 2017, and it has become one of the most popular mobile-friendly architectures for image classification and object detection tasks.

MobileNet is designed to be both lightweight and efficient, with a small number of parameters and computations required for inference. The network uses depth wise separable convolutions, which split the standard convolutional operation into two parts: a depth wise convolution that applies a single filter to each input channel separately, followed by a pointwise convolution that applies 1x1 convolutions to combine the outputs of the depth wise convolution. This approach reduces the computational cost of convolutional operations while maintaining performance.

MobileNet has several variants with different levels of depth and computational complexity, including MobileNet V1, V2, and V3. MobileNet V2 and V3 also introduced several optimizations, such as linear bottlenecks and squeeze-and-excitation blocks, to further improve accuracy and reduce computation.

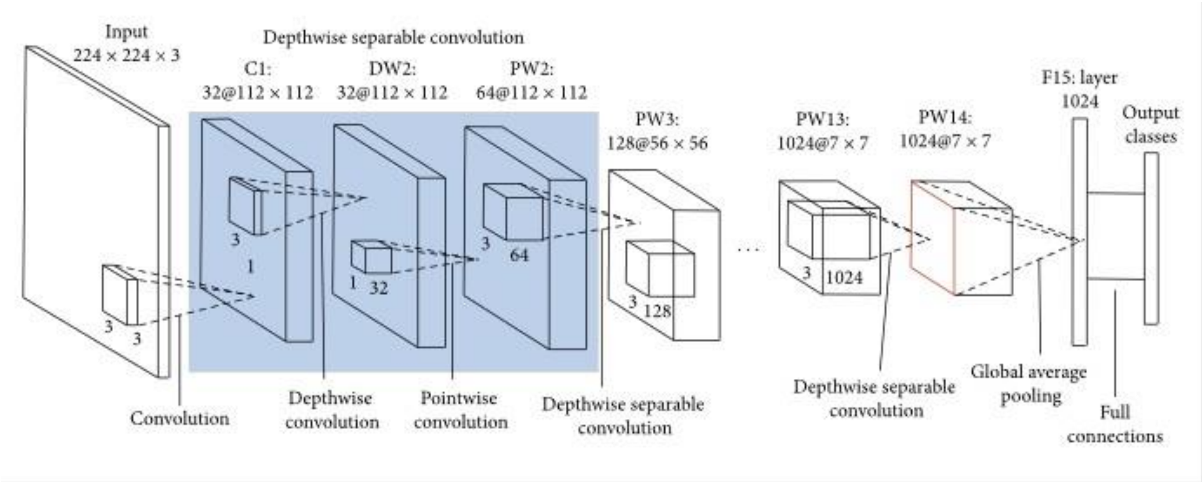


Fig10:- MoblieNet architecture

Result By Graph

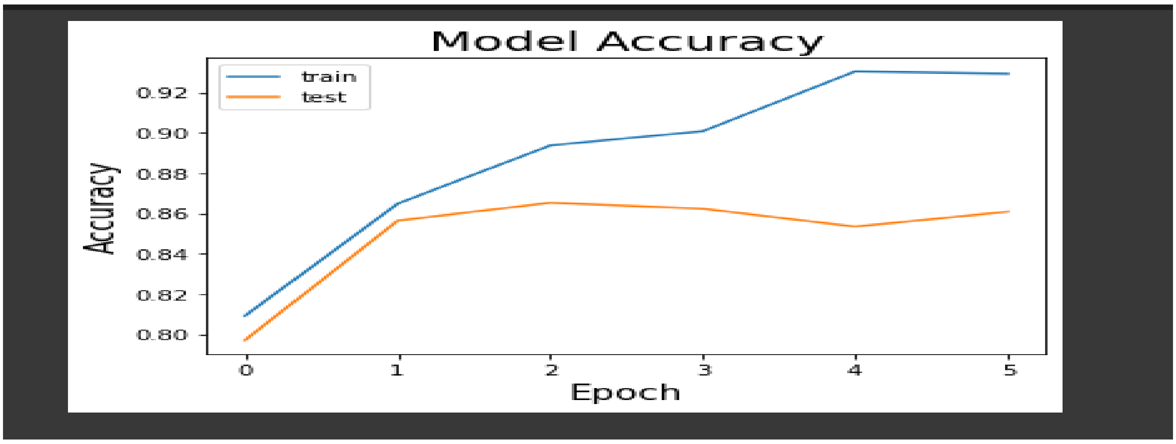


Fig11:- Model Accuracy of EfficientNetB0

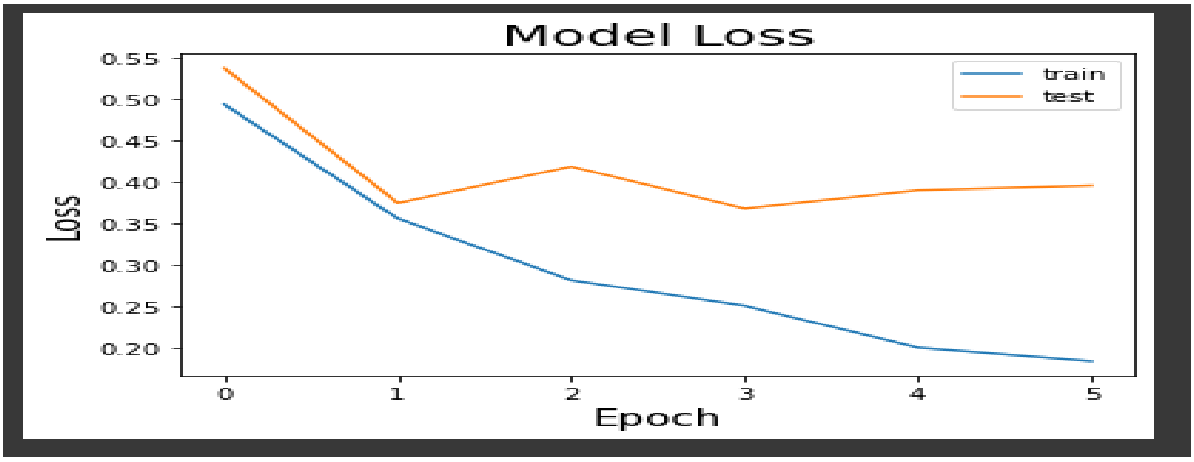


Fig12:- Model Loss of EfficientNetB0

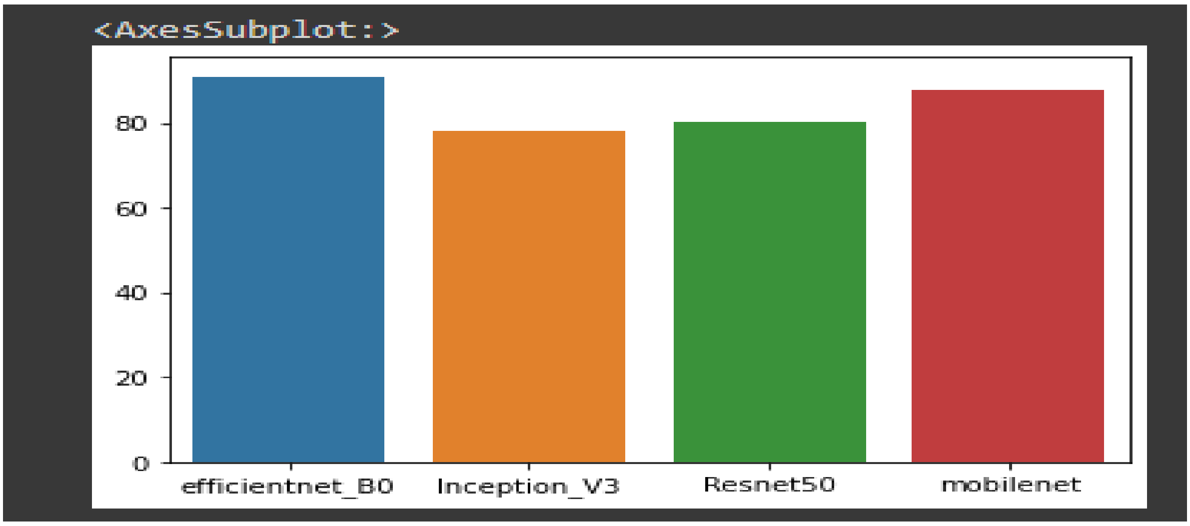


Fig13:- Comparisons of All pretrained Models

Conclusion

Deep learning models can be used to analyze eye images and detect signs of eye diseases with high accuracy. By training these models on large datasets of annotated eye images, they can learn to identify specific patterns and features that indicate the presence of various eye diseases, including glaucoma, diabetic retinopathy, age-related macular degeneration, and more.

The use of deep learning models for eye disease diagnosis has several potential benefits, including faster and more accurate diagnosis, reduced cost and complexity, and improved access to care in remote or underserved areas.

However, it's important to note that deep learning models are not perfect and can still make errors or miss subtle signs of disease. Therefore, it's important to use these models in conjunction with human expertise and follow-up examinations to confirm any diagnoses.

Here, we have done this model by using four pretrained model and we got the 90% accuracy from EfficientNetB0 and we train and deploy our model with that accuracy.

Overall, deep learning models have shown great promise in the field of eye disease diagnosis and have the potential to significantly improve the quality and accessibility of eye care.

Bibliography

[1] J. H. Kempen, B. J. O'Colmain, M. C. Leske, S. M. Hafiner, 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 depth wise 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.