Ocular Disease Recognition through Deep Learning Architectures

CSE499 report submitted in partial fulfillment of the requirements for the degree

of

Bachelor of Science in Computer Science and Engineering

by

Md Sazid Ahmed Tonmoy **ID: 1911498042**

Hawlader Md Hadiuzzaman Reaz **ID: 1821940042**

Under the guidance of

MR. ABU OBAIDAH

Lecturer

DEPARTMENT OF ELECTRICAL COMPUTER ENGINEERING

NORTH SOUTH UNIVERSITY

Bashundhara, Dhaka-1229, Bangladesh

Summer 2022

Department of Electrical & Computer Engineering
North South University
Bashundhara, Dhaka-1229, Bangladesh

DECLARATION

It is hereby acknowledged that:

a. No illegitimate procedure has been practiced during the preparation of this document.

b. This document does not contain any previously published material without proper citation.

c. This document represents our own accomplishment while being Undergraduate Students in the North South University.

Sincerely,

Md Sazid Ahmed Tonmoy
ID: 1911498042

Hawlader Md Hadiuzzaman Reaz
ID: 1821940042

Department of Electrical & Computer Engineering
North South University
Bashundhara, Dhaka-1229, Bangladesh

Approval

This is to certify that the cse499 report entitled Ocular Disease Recognition through Deep Learning Architectures, submitted by Md Sazid Ahmed Tonmoy (ID: 1911498042) and Hawlader Md Hadiuzzaman Reaz (ID: 1821940042), are undergraduate students of the Department of Electrical Computer Engineering, North South University. This report partially fulfil the requirements for the degree of Bachelor of Science in Computer Science and Engineering on September 11, 2022, and has been accepted as satisfactory.

Date:

MR. ABU OBAIDAH
Lecturer
Department of Electrical Computer Engineering
North South University
Dhaka, Bangladesh

DR. RAJESH PALIT
Professor and Chair
Department of Electrical Computer Engineering
North South University
Dhaka, Bangladesh

Abstract

Early detection and identification of eye disorders using fundus pictures is among ophthalmologists' most difficult responsibilities. However, eye illness diagnosis by hand is challenging, time-consuming, and error-prone. For the purpose of employing fundus pictures for early identification of different ocular disorders, a computer-aided automated ocular disease detection system is necessary. Such a system can now be accomplished because to deep learning algorithms' improved picture categorization skills. Four deep learning-based models for pinpointing ocular diseases are presented in this work. For this work, we used the ODIR dataset, which consists of 5000 fundus images. we took 3404 fundus images divided into 5 distinct groups, to train cutting-edge image classification algorithms including Resnet-50, MobileNetV2, and VGG-19.

Index Terms: Ocular Disease Classification, Color Fundus Photography, Ocular Disease Detection, Convolutional Neural Networks, VGG-19, Resnet-50, MobileNetV2, Deep Transfer Learning

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Various ocular diseases are capable of causing permanent and irreversible damage to the patient's vision, and in extreme cases, they can even lead to blindness [1-3]. Although effective treatments are available for these ocular diseases, these treatment options can only be implemented if the disease is diagnosed as early as possible. Ocular diseases are primarily diagnosed using color fundus photography or CFP [4]. This technique is utilized in order to record the interior surface of the human eye so that various types of possible ocular diseases can be detected [5]. Although this method of diagnosis is effective, it's still quite difficult to detect certain ocular diseases using CFP. Some of the most prevalent ocular diseases, such as cataracts, myopia, and diabetic retinopathy, are difficult to diagnose as they show very few initial symptoms. [6] Moreover, the process of manually inspecting and detecting ocular diseases is a laborious task, and this process is not that accurate [7].

In recent times, deep learning-based neural network models have shown promising results in medical image classification and object detection. Moreover, that is why convolutional neural network-based models have been extensively studied for ocular disease detection. Given this, it's critical to offer these people affordable or free complete eye care treatments. Deep-learning-based algorithms are becoming more common in medical image analysis. Deep-learning-based models have been demonstrated to perform well in numerous tasks such object recognition, sentiment analysis [8], medical picture classification [9], and illness diagnosis [10]. One of the most important steps in minimizing an ophthalmologist's workload is the automated diagnosis of diseases. Without the need for human interaction, illnesses may be detected using deep learning and computer vision technology. Only a small number of these studies have been able to fully diagnose [11] more than one eye illness, despite the fact that many of them have produced encouraging results. To accurately detect diverse eye conditions, more study is required to examine the many elements of fundus imaging [12]. This study suggests a system that uses deep learning to recognize different eye diseases. Multilabel categorization has been used as a different strategy [13]. The ocular disease's datasets [14–15] are quite unbalanced. This imbalance makes it difficult to accurately identify or classify sickness or even a normal picture. This method is not recommended for broad classification problems due to its low accuracy.

As a result, our study initially balanced the dataset by training the classes on the pretrained Resnet50, VGG-19, and MobilenetV2 architecture with the same amount of data for each class. By selecting the equal number of photos for each classes, we first loaded the dataset and the associated image into the dataset. For

the VGG-19, Resnet50, and Mobilenetv2 models in this work, the transfer learning approach was used. The accuracy of each class rose once we correctly balanced the dataset. The remaining portions of the essay are structured as follows: The relevant work for this study is displayed in Section 2. Dataset's description is displayed in Section3. All the tools and techniques are extensively covered in Section 4. After discussing our study's results and performance analysis in Section 5. Section 6 brings our work's deployment and Section 7 brings our work to a close.

Chapter 2

Literature Review

An overview, a summary, and an assessment of the state of knowledge in a particular field of study make up a literature review. It could also highlight methodological concerns and make recommendations for further study.In this chapter, we have reviewed some papers related to our work.

2.1 Ocular Disease Detection Using Advanced Neural Network Based Classification Algorithms

Nadim and his team aimed to train a deep convolutional neural network (CNN) to detect ocular diseases using Resnet- 34,EfficientNet, MobileNetV2, and VGG-16 on the ODIR dataset belong to 8 different classes.The accuracy of the VGG-16 model was 97.23%, that of the Resnet-34 model was 90.85%, that of the MobileNetV2 model was 94.32%, and that of the EfficientNet classification model was

93.82%. A system for diagnosing eye diseases in real time will depend on all of these models.[16]

2.2 Deep learning for ocular disease recognition: An inner-class balance

Shakib and his team aimed to train a deep convolutional neural network (CNN) to detect ocular diseases using VGG-19 on the ODIR dataset.With VGG-19, the binary classifications were taught. The accuracy of the VGG-19 model was 98.13% for the normal (N) vs pathological (M) myopia class, 94.03% for the normal (N) against cataract (C), and 90.94% for the normal (N) versus glaucoma classes (G). When the data is balanced, the accuracy of the other models likewise increases.[17]

2.3 Retinal Eye Disease Detection Using Deep Learning

Lorick Jain and his team aimed to train a deep convolutional neural network (CNN) to detect ocular diseases. The objective of this study is to automatically distinguish between photos with retinal issues and photographs of healthy retinas without conducting any specific segmentation or feature extraction. Instead, any retinal fundus picture is automatically classified as healthy or sick using a deep learning algorithm. The network's architecture is straightforward and quick. Two

datasets, including actual patient retinal fundus pictures collected from a nearby hospital, were used to evaluate the model. This model's accuracy was determined to be between 96.5% and 99.7%.[18]

2.4 Retinal Image Analysis for Diabetes-Based Eye Disease Detection Using Deep Learning

This study presents an automated method for disease localization and segmentation based on the fuzzy k-means (FKM) clustering algorithm and Fast Region-based Convolutional Neural Network (FRCNN) algorithm. Since datasets don't often contain bounding-box annotations, They created them using ground truths. The FRCNN is an object identification method that needs these annotations to function. After segmenting out the annotated pictures using FKM clustering, the annotated images are then used to train the FRCNN for localisation. Through intersection-over-union processes, the segmented areas are then compared to the ground facts. They employed the Diaretdb1, MESSIDOR, ORIGA, DR-HAGIS, and HRF datasets for performance evaluation. The effectiveness of the methodology in terms of both illness identification and segmentation is confirmed by a careful comparison against the most recent techniques. In terms of localisation, the suggested model obtained mAPs for DR, DME, and glaucoma of 0.945, 0.943, and 0.941, respectively. Backpropagation estimation is used by FRCNN, along with the addition of the bbox regression and classification heads, and multi-task loss was used to train the model. Using FRCNN, the suggested technique concur-

rently identifies anomalies in the DR, DME, and glaucoma areas. Finally, FKM clustering correctly removes the regions from localized regions. Our segmentation accuracy for the DR, DME, and glaucoma areas was 0.952, 0.958, and 0.9526, respectively.[19]

2.5 Automated detection of mild and multi-class diabetic eye diseases using deep learning

This study set out to develop an automated classification system that took into account two different diabetic eye disease (DED) scenarios: I mild multi-class DED and (ii) multi-class DED. Our model was evaluated using numerous datasets that an optometrist had annotated. The top two pretrained convolutional neural network (CNN) models from ImageNet were used in the experiment. Various performance-improving approaches, such as fine-tuning, optimization, and contrast enhancement, were also used. The VGG16 model achieved a maximum accuracy of 88.3% for multi-class classification and 85.95% for moderate multi-class classification.[20]

2.6 A Deep Learning Method for Microaneurysm Detection in Fundus Images

In order to detect MA in fundus pictures, a Stacked Sparse Autoencoder (SSAE), an example of a DL technique, is proposed in this study. The first fundus photos are used to create tiny image patches. To find distinctive properties of MA, the SSAE learns high-level features just from pixel intensities. Each image patch is classified as MA or non-MA using the high-level characteristics learned by SSAE. The training/testing data and ground truth are provided via the open benchmark DIARETDB. The 89 photos are divided into 2182 image patches with MA lesions, which serve as positive data, and 6230 image patches without MA lesions, which are produced using a randomly selected sliding window operation, which serve as negative data. SSAE learnt directly from the raw image patches and automatically retrieved the distinctive characteristics to categorize the patches using Softmax Classifier without the need for blood vessel removal or other preprocessing procedures. Using 10-fold cross-validation, the fine-tuning process resulted in an improved F-measure of 91.3% and an average area under the ROC curve (AUC) of 96.2%.[21]

2.7 Data Driven Approach for Eye Disease Classification with Machine Learning

The objective of this project is to create a basic framework for storing diagnostic data in an international standard format to make it easier for machine learning algorithms to anticipate illness diagnosis based on symptoms. A user-friendly interface was created in an effort to assure error-free data entering. A number of machine learning techniques, including as Decision Tree, Random Forest, Naive Bayes, and Neural Network algorithms, were also employed to assess patient data based on a variety of variables, such as age, medical history, and clinical observations. The diagnosis was established using the ICD-10 coding published by the American Academy of Ophthalmology, and the data was organized in accordance with hierarchical hierarchies created by medical specialists. Furthermore, new categories for symptoms as well as diagnoses will be added as part of the system's self-learning evolution.The classification results using tree-based approaches showed that, given enough data, the suggested framework works adequately. In contrast to more sophisticated techniques like neural networks and the naive Bayes algorithm, the random forest and decision tree algorithms' prediction rate is greater than 90% because of a structured data arrangement.[22]

Chapter 3

DATASET

The dataset [23] utilized in this investigation is called ODIR (Ocular Disease Intelligent Recognition). One of the most extensive public resources on Kaggle for identifying eye illnesses is this dataset. Eight classifications of ocular diseases are used to group the fundus photos in this collection. They are normal (N), myopia (M), hypertension (H), diabetes (D), cataract (C), glaucoma (G), age-related macular degeneration (A), and other abnormalities/diseases (O). The 5000 color fundus images in this dataset are split into training and testing groups. All of the photographs for this project were scaled to 224 224. The full ODIR dataset was not used for this research. Normal, cataract, glaucoma, myopia, and hypertension related images are being used. To remove imbalance, normal images are taken according to the disease's image number to train the model. For VGG19 and Resnet50, The dataset was splitted into 90% and 10% ratio for training and testing. For MobilenetV2, the dataset was split into 70%, 20%, and 10% ratios

for training, validation, and testing. For Mobilenetv2, the accuracy was too low. That is why the dataset was augmented for better results. The number of images for each catagory is given below:

Table 3.1: Dataset Size

Class Name	Number of Images	Augmented
Normal	2100	2100
Cataract	401	1899
Glaucoma	396	1896
Myopia	205	1805
Hypertensive	202	1702

Chapter 4

METHODOLOGY

The broad plan and justification for your research effort are referred to as your methodology. It entails researching the theories and ideas that underpin the procedures employed in your industry in order to create a strategy that is in line with your goals.There are various types of images of ocular diseases. We chose cataracts, glaucoma, pathological myopia, and hypertensive retinopathy to work with. For each model (VGG19, Resnet50, Mobilenetv2), we tested Normal vs. Cataract, Normal vs. Glaucoma, Normal vs. Myopia, and Normal vs. Hypertension. We take 100 epoch for every model. We resize every images 224*224. We also did use some function for time reduction. EarlyStopping class stops training when a monitored metric has stopped improving. ReduceLROnPlateau class reduces learning rate when a metric has stopped improving. Here is the working steps.

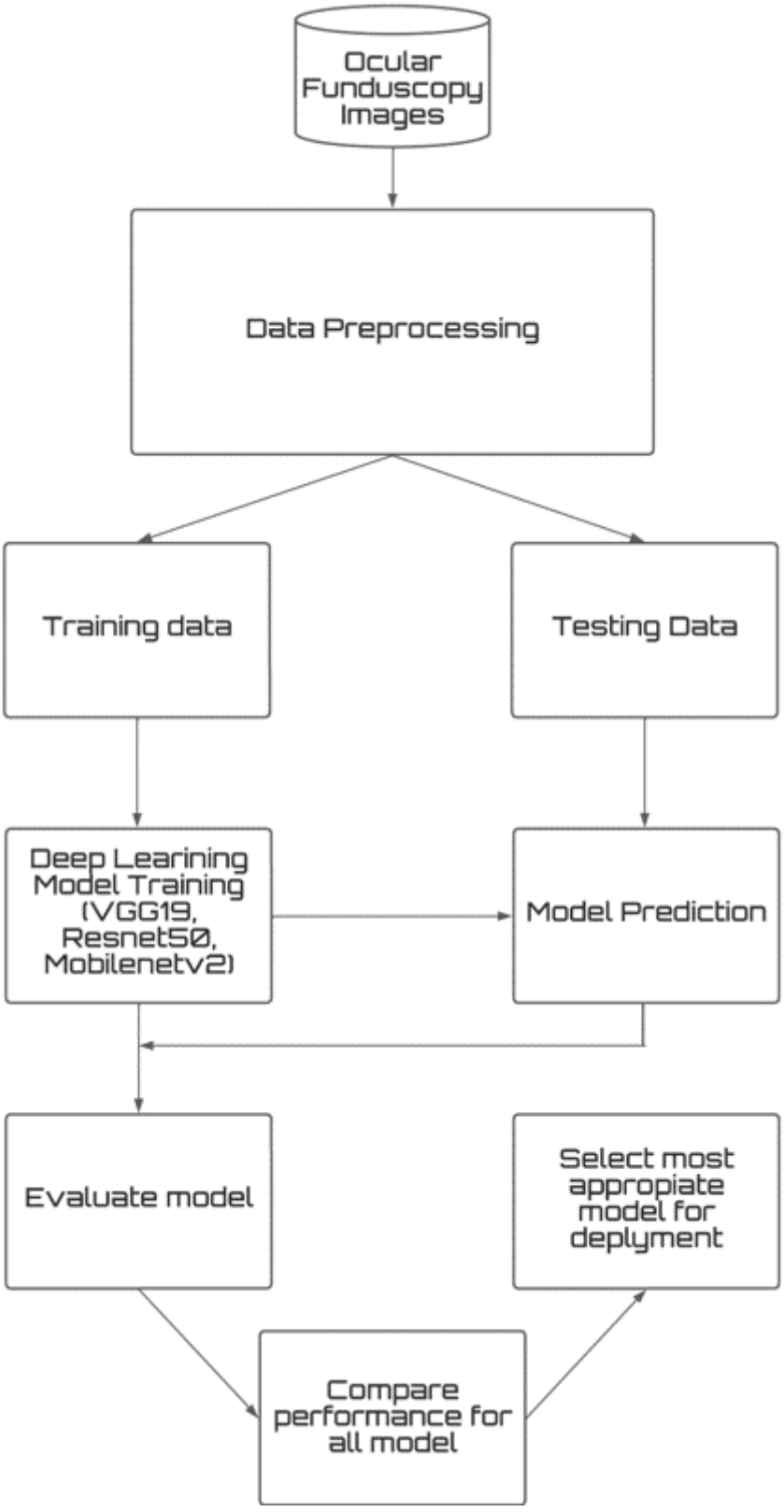

Figure 4.1: Working steps

4.1 Classification Using Resnet50

ResNet50 is a variant of the Resnet model,, which has 48 convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 109 floating point operations. It is a widely used ResNet model, and we have explored the ResNet50 architecture in depth. So as we can see in the resnet 50 architecture contains the following element:

1. A convolution with a kernel size of 7 * 7 and 64 different kernels all with a stride of size 2 giving us 1 layer.

2. Next we see max pooling with also a stride size of 2.

3. In the next convolution there is a 1 * 1,64 kernel following this a 3 * 3,64 kernel and at last a 1 * 1,256 kernel, These three layers are repeated in total 3 time so giving us 9 layers in this step.

4. Next we see kernel of 1 * 1,128 after that a kernel of 3 * 3,128 and at last a kernel of 1 * 1,512 this step was repeated 4 time so giving us 12 layers in this step. After that there is a kernal of 1 * 1,256 and two more kernels with 3 * 3,256 and 1 * 1,1024 and this is repeated 6 time giving us a total of 18 layers.

5. And then again a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1,2048 and this was repeated 3 times giving us a total of 9 layers.

6. We do a average pool and end it with a fully connected layer containing 1000 nodes and at the end a softmax function so this gives us 1 layer.

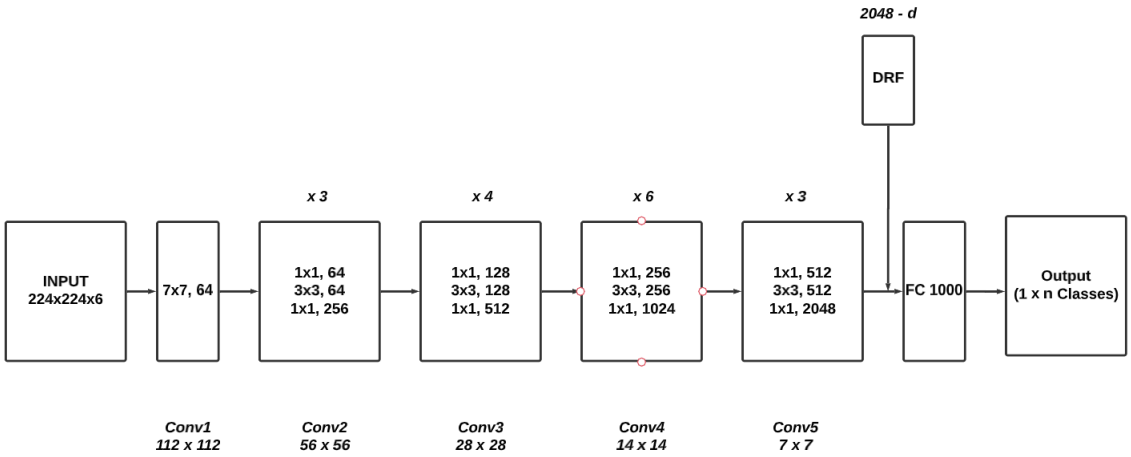

Figure 4.2: Resnet50 Architecture

4.2 Classification Using VGG19

A variation of the VGG model called VGG19 has 19 layers in total (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer). There are further VGG variations, including VGG11, VGG16, and others. 19.6 billion FLOPs make up VGG19. So as we can see in the VGG19architecture contains the following element:

1. A fixed size of (224 * 224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3).

2. The only preprocessing that was done is that they subtracted the mean RGB value from each pixel, computed over the whole training set.

3. Used kernels of (3 * 3) size with a stride size of 1 pixel, this enabled them to cover the whole notion of the image.

4. spatial padding was used to preserve the spatial resolution of the image.

5. max pooling was performed over a 2 * 2 pixel windows with sride 2.

6. this was followed by Rectified linear unit(ReLu) to introduce non-linearity to make the model classify better and to improve computational time as the previous models used tanh or sigmoid functions this proved much better than those.

7. implemented three fully connected layers from which first two were of size 4096 and after that a layer with 1000 channels for 1000-way ILSVRC classification and the final layer is a softmax function.

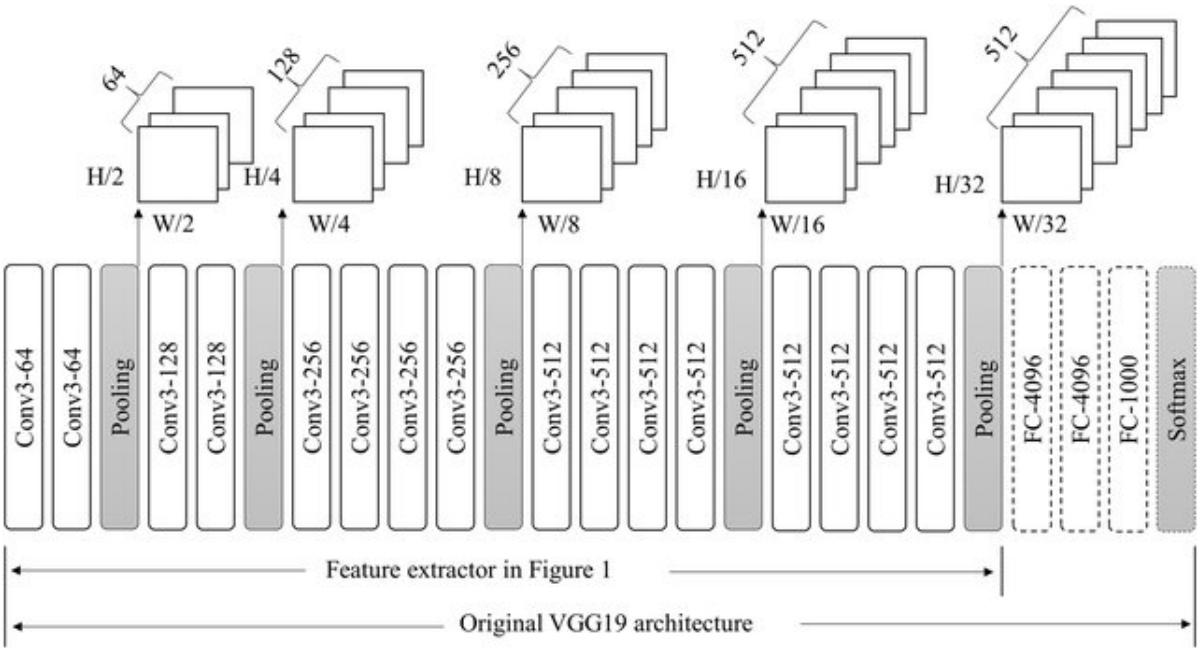

Figure 4.3: VGG19 Architecture

4.3 Classification Using MobilenetV2

We have explored MobileNet V2 architecture in depth. MobileNet V2 model has 53 convolution layers and 1 AvgPool with nearly 350 GFLOP. It has two main components:

1. Bottleneck Residual Block
2. Inverted Residual Block

There are two types of Convolution layers in MobileNet V2 architecture:

1. 1x1 Convolution
2. 3x3 Depth wise Convolution

There are three streams and the input shape is 224x224. Our design involves a filter size of 32 for padding, a kernel size of 3, and activation function based on ReLU for the two first layers. The first max pooling layer has a pool size of 2 and strides of 2. The further plain layer combines all of the pooled characteristics into a separate cell. In the end, two thick layers were produced. The activation function for the first layer is ReLU, while the activation function for the least thick layer is softmax. The features are added to the network once they have been pre-processed. A bird's-eye perspective of the structure is shown below.

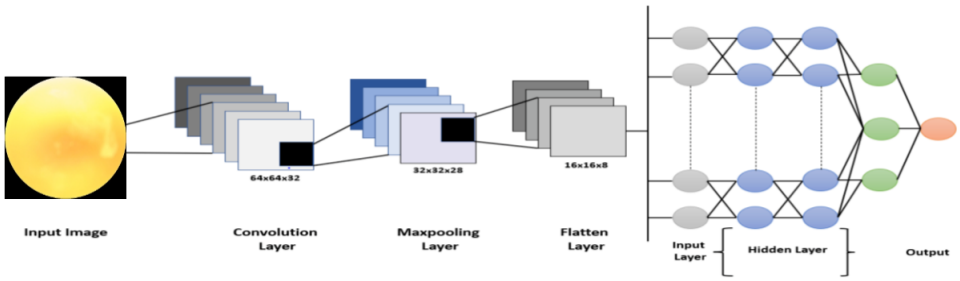

Figure 4.4: MobileNetV2 Architecture

4.4 Evaluation

We should be prepared with several assessment metrics to examine the classification algorithm in the event of a classification problem. As follows:

1. **Confusion Matrix:** The classification model's accuracy in classifying instances into distinct groups is summarized in a table called the confusion matrix. The model's anticipated label is on one axis of the confusion matrix, while the actual label is on the other. When comparing several models, we may use the confusion matrix to assess how well each one predicted true positives (TP) and true negatives (TN). We chose a model as our basic model if it accurately predicted TP and TN compared to other models.

	Positive	negative
Positive	TP	FP
Negative	FN	TN

Figure 4.5: Confusion Matrix

TP = True Positive (The total number of images that are correctly detected to be positive)

FP = False Positive (The total number of images that are predicted to be positive but actually are negative)

TN = True Negative (The number of images that are accurately predicted to be negative)

FN = False Negative (The number of images that are incorrectly predicted to be negative)

2. **Precision and Recall:** Precision and recall are two metrics used to evaluate classification and retrieval systems' performance. Precision is the percentage of relevant occurrences among all retrieved examples. Recall, also known as sensitivity, is the percentage of recovered instances among all appropriate models. In a perfect classifier, precision and recall are both one.

$$Recall = \frac{TP}{TP + FP}$$
$$Precision = \frac{TP}{TP + FP}$$

3. **Accuracy:** It is calculated by dividing the total number of correctly categorized instances by the overall number of classified examples. When the importance of each class's prediction error is equal, this measure is crucial. Here, false positives should be addressed more than false negatives.

$$Accuracy = \frac{TP + tn}{TP + TN + FP + FN}$$

4. **F1 score:** A weighted average of recall and accuracy is the F1 score. False positive and false negative results can occur in accuracy and recall, as is well known, thus both are taken into account. In most cases, the F1 score is more helpful than accuracy, particularly if your class is distributed unevenly. When false positives and false negatives cost about the same, accuracy performs best. It is preferable to include both Precision and Recall if the costs of false positives and false negatives are significantly different.

$$F1Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

5. **Learning Curve:** For algorithms that learn (optimize their internal parameters) gradually over time, like deep learning neural networks, learning curves are frequently employed in machine learning. If maximization is the metric used to measure learning, then higher scores (bigger numbers) signify more learning. Accuracy in categorisation might serve as an example. It is more typical to employ a score that minimizes, like loss or error, where better scores (lower numbers) imply greater learning and a value of 0.0 indicates that the training dataset was learnt properly with no errors. Additionally, a hold-out validation dataset that is separate from the training dataset can be used to test it. An assessment of the validation dataset provides insight into the model's "generalizability."

Chapter 5

EXPERIMENTAL RESULTS

We will be analyze our model by there accuracy. Based on the input, or training, data, machine learning model accuracy is the statistic used to discover which model is best at recognizing correlations and patterns between variables in a dataset. The better predictions and insights a model can generate, which in turn provide more commercial value, depend on how well it can generalize to "unseen" data.

1. **VGG19:** Using VGG-19, the accuracy of cataract vs. normal original data is 97.24%. Glaucoma vs. normal has an accuracy of 90.94%. In addition, myopia vs. normal has an accuracy of 98.13%. The accuracy for hypertensive vs. normal is 94.73%. It performed very well. We removed bias by balancing images for data classes.

2. **Resnet50:** Resnet50 is the best performed model according to accuracy. Using Resnet50, the accuracy of cataract vs. normal original data is 99.45%.

Glaucoma vs. normal has an accuracy of 90.98%. In addition, myopia vs. normal has an accuracy of 99.26%. The accuracy for hypertensive vs. normal is 94.73%. It performed very well. We removed bias by balancing images for data classes.

3. **MobileNetV2:** It is the worst performed model out of three. The result without augmented data is too bad. But after taking augmented data, we did get satisfactory results form this model.

   (a) Cataract V Normal original data Using MobilenetV2 has 77% accuracy.

   (b) Cataract V Normal augmented data Using MobilenetV2 has 90% accuracy

   (c) Glaucoma V Normal original data Using MobilenetV2 has 59% accuracy

   (d) Glaucoma V Normal augmented data Using MobilenetV2 has 84% accuracy

   (e) Myopia V Normal original data Using MobilenetV2 has 77% accuracy

   (f) Myopia V Normal augmented data Using MobilenetV2 has 89% accuracy

   (g) Hypertensive V Normal original data Using MobilenetV2 has 52% accuracy

   (h) Hypertensive V Normal augmented data Using MobilenetV2 has 88% accuracy

we can see in the training performance of MobileNetV2, its accuracy is getting improved and it can be inferred that the accuracy will certainly be improved if we run the training for more number of epochs. Also, There is hundred percentage training accuracy issue on VGG19 and Resnet50. We had tried to train those model in various dataset split ratio. There were various issue, but we selected the most optimized one. When we got hundred percentage training accuracy, we looked the accuracy of the model and the curve. If training accuracy curve is above validation curve and the accuracy was the highest, we recorded necessary information. But There is no hundred percent training accuracy issue on MobilenetV2. So, there is no overfitting issue also.

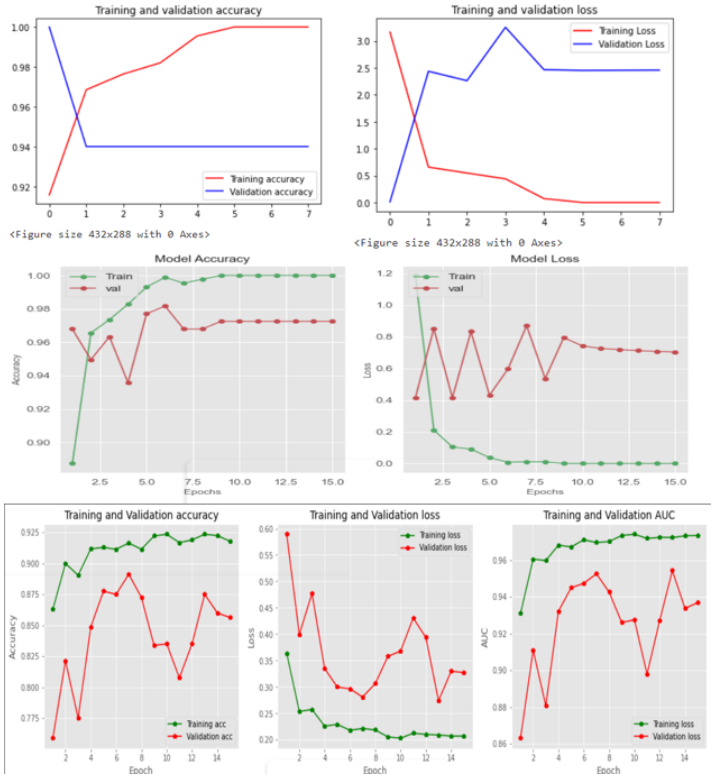

Figure 5.1: Plotting for Cataract v Normal.(Upper: Resnet50, Middle: VGG19, Lower: Mobilenetv2 on Augmented data)

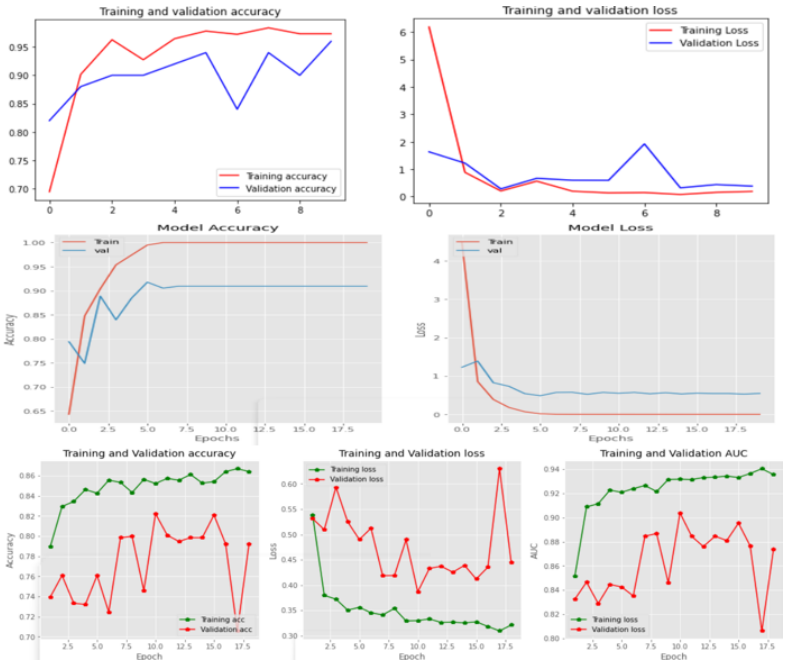

Figure 5.2: Plotting for Glaucoma v Normal.(Upper: Resnet50, Middle: VGG19, Lower: Mobilenetv2 on Augmented data)

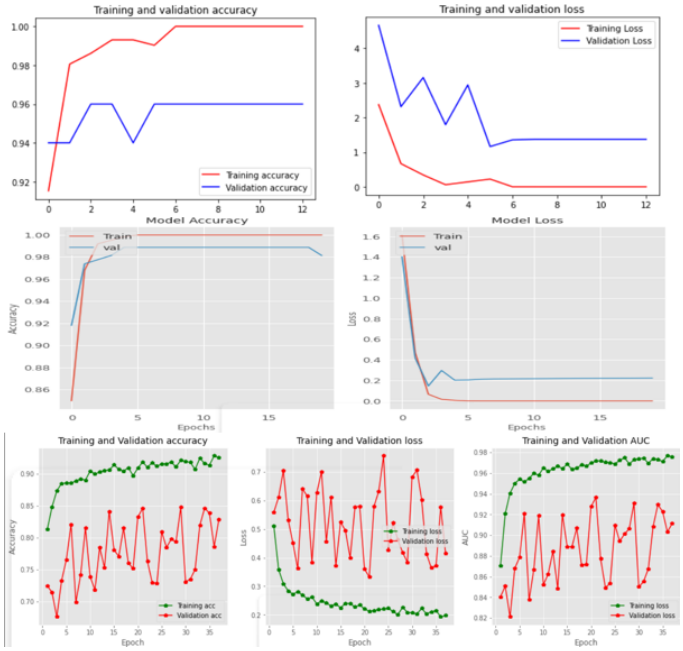

Figure 5.3: Plotting for Myopia v Normal.(Upper: Resnet50, Middle: VGG19, Lower: Mobilenetv2 on Augmented data)

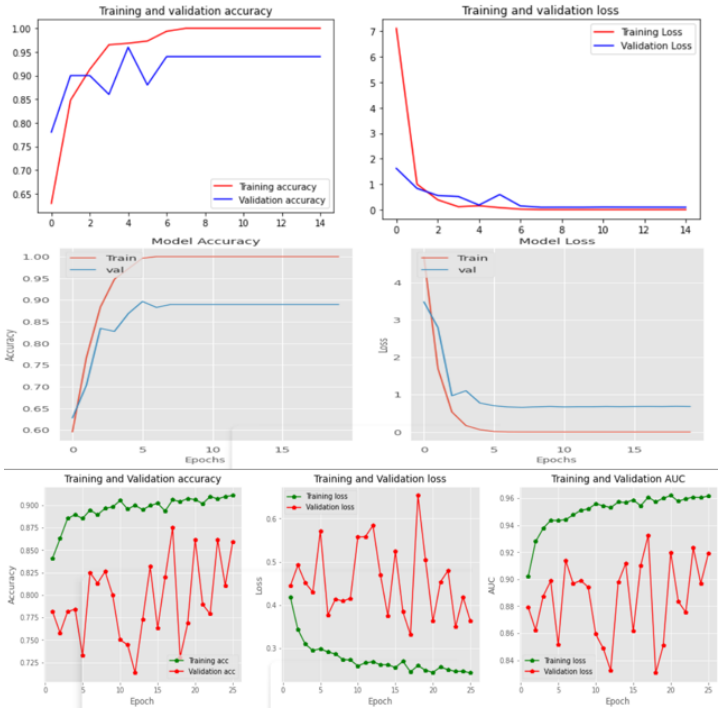

Figure 5.4: Plotting for Hypertensive v Normal.(Upper: Resnet50, Middle: VGG19, Lower: Mobilenetv2 on Augmented data)

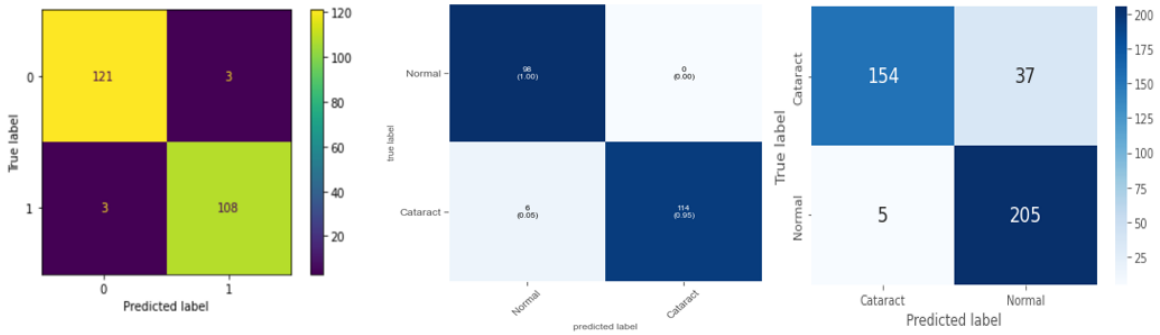

Figure 5.5: Confusion Matrix for Cataract v Normal.(Left: Resnet50, Middle: VGG19, Right: Mobilenetv2 on Augmented data)

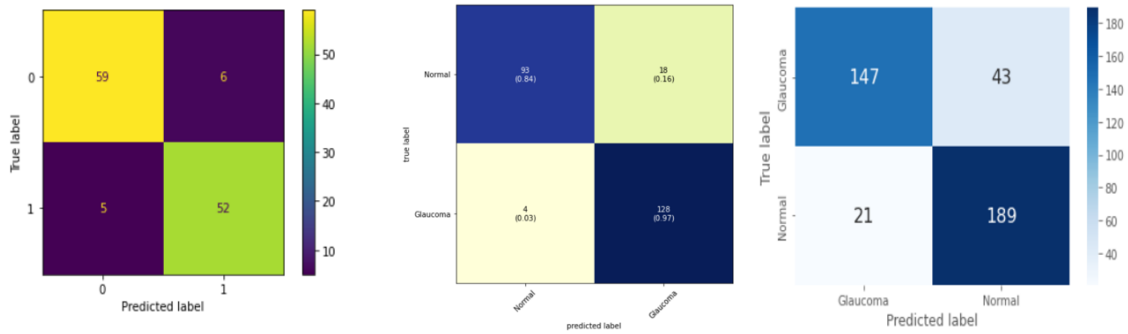

Figure 5.6: Confusion Matrix for Glaucoma v Normal.(Left: Resnet50, Middle: VGG19, Right: Mobilenetv2 on Augmented data)

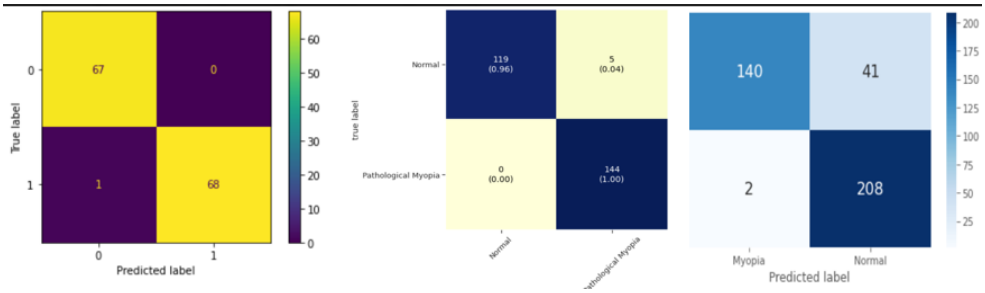

Figure 5.7: Confusion Matrix for Myopia v Normal.(Left: Resnet50, Middle: VGG19, Right: Mobilenetv2 on Augmented data)

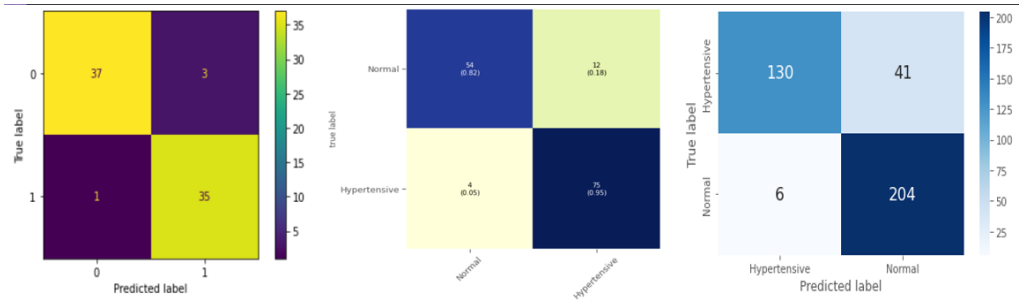

Figure 5.8: Confusion Matrix for Hypertensive v Normal.(Left: Resnet50, Middle: VGG19, Right: Mobilenetv2 on Augmented data)

It is crucial to compare machine learning algorithms. Better performance of the machine learning software or solution is unquestionably the main goal of model comparison and selection. The goal is to select the fewest algorithms possible that meet the needs of the company and the data. If the chosen model fails to comprehend unknown input and is strongly associated with the training data, high performance may be short-lived. Finding a model that comprehends underlying data patterns is crucial in order to ensure that predictions are reliable and that little retraining is required. Minute details and information are captured when models are reviewed and prepared for comparisons, and they are useful during retraining.With the model information at hand, it is simple to focus on models that can provide fast processing and make efficient use of memory resources. Ad-

ditionally, a number of parameters must be configured for the machine learning solutions throughout production.

Table 5.1: MobileNetV2's Results Report without Augmented Data

Model	Accuracy
Normal VS Cataract	77%
Normal VS Glaucoma	59%
Normal VS Myopia	77%
Normal VS Hypertensive	52%

Table 5.2: Resnet50's Classification Report

Model	Accuracy	Disease	Precision	F1-score	Recall
Normal VS Cataract	99.45%	Normal	0.98	0.98	0.98
		Cataract	0.97	0.97	0.97
Normal VS Glaucoma	90.98%	Normal	0.92	0.91	0.91
		Glaucoma	0.9	0.9	0.91
Normal VS Myopia	99.26%	Normal	0.99	0.99	1
		Myopia	1	0.99	0.99
Normal VS Hypertensive	94.73%	Normal	0.97	0.95	0.93
		Hypertensive	0.92	0.95	0.97

Table 5.3: VGG19's Classification Report

Model	Accuracy	Disease	Precision	F1-score	Recall
Normal VS Cataract	97.24%	Normal	0.94	0.97	1
		Cataract	1	0.97	0.95
Normal VS Glaucoma	90.94%	Normal	0.96	0.89	0.84
		Glaucoma	0.88	0.92	0.997
Normal VS Myopia	98.13%	Normal	1	0.98	0.96
		Myopia	0.97	0.98	1
Normal VS Hypertensive	88.96%	Normal	0.97	0.95	0.93
		Hypertensive	0.92	0.95	0.97

Table 5.4: MobileNetV2's Classification Report with Augmented Data

Model	Accuracy	Disease	Precision	F1-score	Recall
Normal VS Cataract	89.53%	Normal	0.85	0.91	0.98
		Cataract	0.97	0.88	0.81
Normal VS Glaucoma	84.00%	Normal	0.81	0.86	0.9
		Glaucoma	0.88	0.82	0.77
Normal VS Myopia	89.00%	Normal	0.84	0.91	0.99
		Myopia	0.99	0.87	0.77
Normal VS Hypertensive	87.66%	Normal	0.83	0.9	0.97
		Hypertensive	0.96	0.85	0.76

Chapter 6

DEPLOYMENT

The process of integrating a machine learning model into an already-existing production environment is known as deployment, and it allows you to use data to make useful business choices. It can be one of the most challenging stages of the machine learning life cycle and is one of the final ones. Frequently, traditional model-building languages are incompatible with an organization's IT systems, requiring data scientists and programmers to spend considerable time and brainpower rebuilding them. A model must be successfully put into production before it can be used for making useful decisions. The effect of your model will be much diminished if you are unable to consistently derive useful insights from it.Machine learning models must be smoothly deployed into production in order for businesses to use them to start making useful judgments. This will maximize their worth.

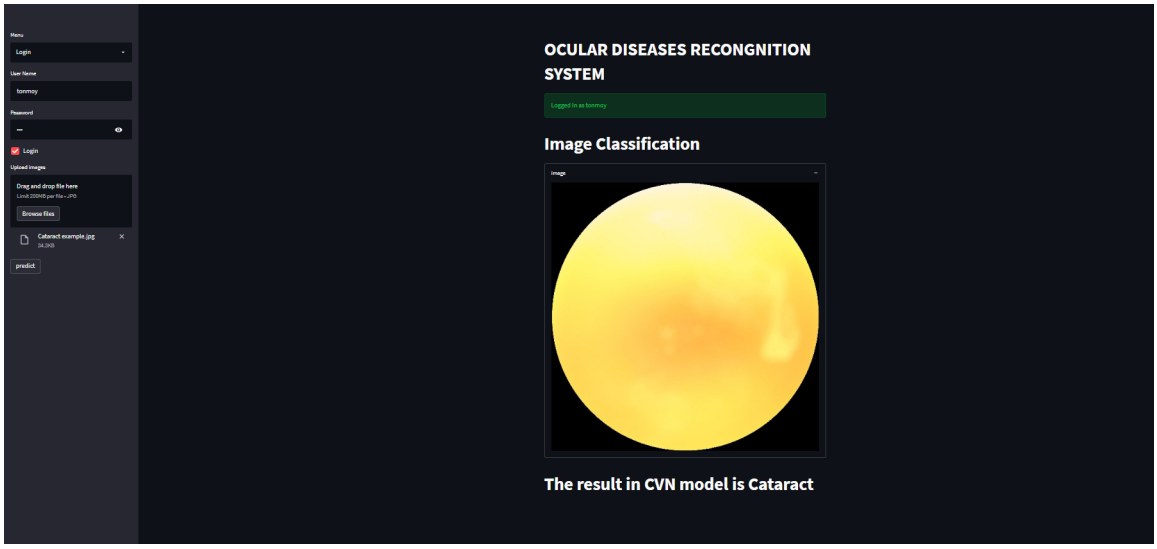

Figure 6.1: Deployment

A Python-based open source app framework is called Streamlit. It enables us to quickly develop web applications for data science and machine learning. Major Python libraries like scikit-learn, Keras, PyTorch, SymPy (latex), NumPy, pandas, and Matplotlib are all compatible with it. We used Streamlit to deploy our project. We use Sqlite3 for temporary Data Management. Python SQLite3 module is used to integrate the SQLite database with Python. It is a standardized Python DBI API 2.0 and provides a straightforward and simple-to-use interface for interacting with SQLite databases. There is no need to install this module separately as it comes along with Python after the 2.5x version.

Chapter 7

CONCLUSION

In this study, we created four neural network-based models for classifying eye diseases. These models are VGG-19, Resnet-50, and MobileNetV2. When it comes to categorizing ocular disorders from fundus pictures, the Resnet50 offered the best accuracy for all models. The other models' performance was similarly acceptable. MobilenetV2 has no issue with hundred percent training accuracy. To verify the efficacy of our suggested strategy, we conducted extensive tests using the ODIR-2019 dataset, which is open to the general public. In comparison to the current CNN-based ocular illness classification models, our suggested technique can produce results that are more spectacular while using less computing power. The nicest thing about our suggested approach is how easy it may be applied to various kinds of illness categorization based on medical images. Such a method will change the area of visual illness diagnostics and be of enormous use to medical experts.Additionally, this work could benefit from the use of ocular image segmen-

tation. To address the imbalance issue, research can create comparable pictures of eye illness using generative adversarial networks (GANs). Additionally, a system like this would revolutionize the field of diagnosing eye diseases and be very helpful to medical professionals. Our opinion is that it can still be a valuable model, and there will likely be possibilities to improve it with further research and study in the near future.

7.1 Data Availability

The data used to support the findings of this study are freely available at https://www.kaggle.com/andrewmvd/ocular-disease-recognition-odir5k.

Appendix A

CODE

A.1 Deployment Code

Listing A.1: app.py

```
import streamlit as st
import pandas as pd


# Security
#passlib , hashlib , bcrypt , scrypt
import hashlib
def make_hashes(password):
    return hashlib.sha256(str.encode(password)).hexdigest()


def check_hashes(password,hashed_text):
```

```
    if make_hashes(password) == hashed_text:
        return hashed_text
    return False
# DB Management
import sqlite3
conn = sqlite3.connect('data.db')
c = conn.cursor()
# DB  Functions
def create_usertable():
    c.execute('CREATE TABLE IF NOT EXISTS
    userstable(username TEXT, password TEXT)')


def add_userdata(username,password):
    c.execute('INSERT INTO userstable(username,password)
    VALUES (?,?)',(username,password))
    conn.commit()


def login_user(username,password):
    c.execute('SELECT * FROM userstable
    WHERE username =? AND password = ?',(username,password))
    data = c.fetchall()
    return data
```

```
def view_all_users():
    c.execute('SELECT * FROM userstable')
    data = c.fetchall()
    return data


def main():
    """OCULAR DISEASES RECONGNITION SYSTEM"""
    st.title("OCULAR DISEASES RECONGNITION SYSTEM")
    menu = ["Login","SignUp"]
    choice = st.sidebar.selectbox("Menu",menu)
    if choice == "Login":
        username = st.sidebar.text_input("User Name")
        password = st.sidebar.text_input("Password",type='password')
        if st.sidebar.checkbox("Login"):
            # if password == '12345':
            create_usertable()
            hashed_pswd = make_hashes(password)
            result=
            login_user(username,check_hashes(password,hashed_pswd))
            if result:
                st.success("Logged In as {}".format(username))
                # Custom imports
                import tensorflow as tf
```

```
import numpy as np
from PIL import Image, ImageOps
st.title("Image Classification")
upload_file = st.sidebar.file_uploader(
"Upload images", type = 'jpg')
generate_pred = st.sidebar.button("predict")
model = tf.keras.models.load_model('CVN.h5')
def import_n_pred(image_data, model):
    size = (224,224)
    image = ImageOps.fit
    (image_data, size, Image.ANTIALIAS)
    img = np.asarray(image)
    reshape = img[np.newaxis,...]
    pred = model.predict(reshape)
    return pred
if generate_pred:
    image = Image.open(upload_file)
    with st.expander('image', expanded=True):
        st.image(image, use_column_width=True)
    pred = import_n_pred(image, model)
    labels = ['Cataract', 'Normal']
    st.title("The result in CVN model is {}"
    .format(labels[np.argmax(pred)]))
```

```
            if (np.argmax(pred)==1):
                model = tf.keras.models.load_model
                ('GVN.h5')
                pred = import_n_pred(image, model)
                labels = ['Glaucoma', 'Normal']
                st.title("The result in GVN model is {}"
                .format(labels[np.argmax(pred)]))
                if (np.argmax(pred)==1):
                    model = tf.keras.models.load_model
                    ('MVN.h5')
                    pred = import_n_pred(image, model)
                    labels = ['Myopia', 'Normal']
                    st.title("The result in MVN model is {}"
                    .format(labels[np.argmax(pred)]))
                    if (np.argmax(pred)==1):
                        model = tf.keras.models.load_model
                        ('HVN.h5')
                        pred = import_n_pred(image, model)
                        labels = ['Hypertensive','Normal']
                        st.title("The result in HVN model is {}"
                        .format(labels[np.argmax(pred)]))
    else:
        st.warning("Incorrect Username/Password")
```

```
    elif choice == "SignUp":
        st.subheader("Create New Account")
        new_user = st.text_input("Username")
        new_password = st.text_input("Password",type='password')

        if st.button("Signup"):
            create_usertable()
            add_userdata(new_user,make_hashes(new_password))
            st.success("You have successfully created a valid Account")
            st.info("Go to Login Menu to login")

if __name__ == '__main__':
    main()
```

Bibliography

[1] Bourne, R. R., Stevens, G. A., White, R. A., Smith, J. L., Flaxman, S. R., Price, H., Jonas, J. B., Keeffe, J., Leasher, J., Naidoo, K., Pesudovs, K., Resnikoff, S., Taylor, H. R. (2013). Causes of vision loss worldwide, 1990–2010: a systematic analysis. The Lancet Global Health, 1(6). https://-doi.org/10.1016/s2214-109x(13)70113-x

[2] Sommer, A., Tielsch, J. M., Katz, J., Quigley, H. A., Gottsch, J. D., Javitt, J. C., Martone, J. F., Royall, R. M., Witt, K. A., Ezrine, S. (1991). Racial Differences in the Cause-Specific Prevalence of Blindness in East Baltimore. New England Journal of Medicine, 325(20), 1412–1417. https://-doi.org/10.1056/nejm199111143252004

[3] Congdon, N., O'Colmain, B., Klaver, C. C., Klein, R., Muñoz, B., Friedman, D. S., Kempen, J., Taylor, H. R., Mitchell, P., Eye Diseases Prevalence Research Group (2004). Causes and prevalence of visual impairment among adults in the United States. Archives of ophthalmology (Chicago, Ill. : 1960), 122(4), 477–485.

[4] Application of Ocular Fundus Photography and Angiography. (2014). Ophthalmological Imaging and Applications, 154–175. https://-doi.org/10.1201/b17026-12

[5] Rowe, S., MacLean, C. H., Shekelle, P. G. (2004). Preventing Visual Loss From Chronic Eye Disease in Primary Care. JAMA, 291(12), 1487. https://-doi.org/10.1001/jama.291.12.1487

[6] Kessel, L., Erngaard, D., Flesner, P., Andresen, J., Tendal, B., Hjortdal, J. (2015). Cataract surgery and age-related macular degeneration. An evidence-based update. Acta Ophthalmologica, 93(7), 593–600. https://-doi.org/10.1111/aos.12665Li, N., Li, T., Hu, C., Wang, K.

[7] Li, N., Li, T., Hu, C., Wang, K., Kang, H. (2021). A Benchmark of Ocular Disease Intelligent Recognition: One Shot for Multi-disease Detection. Benchmarking, Measuring, and Optimizing, 177–193. https://doi.org/10.1007/978-3-030-71058-3-11

[8] K. N. Alam, M. S. Khan, A. R. Dhruba et al., "Deep learning-based sentiment analysis of COVID-19 vaccination responses from Twitter data," Computational and Mathematical Methods in Medicine, vol. 2021, pp. 1–15, 2021.

[9] J. He, C. Li, J. Ye, Y. Qiao, and L. Gu, “Self-speculation of clinical features based on knowledge distillation for accurate ocular disease classification,” Biomedical Signal Processing and Control, vol. 67, Article ID 102491, 2021.

[10] K. N. Alam and M. M. Khan, “CNN based COVID-19 prediction from chest X-ray images,” in Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), pp. 0486–0492, IEEE, New York, USA, 1 December 2021.

[11] A. G. Roy, S. Conjeti, S. P. K. Karri et al., “ReLayNet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks,” Biomedical Optics Express, vol. 8, no. 8, pp. 3627–3642, 2017.

[12] C. S. Lee, A. J. Tyring, N. P. Deruyter, Y. Wu, A. Rokem, and A. Y. Lee, “Deep-learning based, automated segmentation of macular edema in optical coherence tomography,” Biomedical Optics Express, vol. 8, no. 7, pp. 3440–3448, 2017.

[13] S. P. K. Karri, D. Chakraborty, and J. Chatterjee, “Transfer learning based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration,” Biomedical Optics Express, vol. 8, no. 2, pp. 579–592, 2017.

[14] M. Oda, T. Yamaguchi, H. Fukuoka, Y. Ueno, and K. Mori, “Automated Eye Disease Classification Method from Anterior Eye Image Using Anatomical Structure Focused Image Classification Technique,” 2020, https://arxiv.org/abs/2005.01433.

[15] F. Eperjesi, C. W. Fowler, and A. J. Kempster, “Luminance and chromatic contrast effects on reading and object recognition in low vision: a review of the literature,” Ophthalmic and Physiological Optics, vol. 15, no. 6, pp. 561–568, 1995.

[16] N. M. Dipu, S. Alam Shohan, and K. M. A. Salam, “Ocular disease detection using advanced neural network based classification algorithms,” ASIAN JOURNAL OF CONVERGENCE IN TECHNOLOGY, vol. 7, no. 2, pp. 91–99, 2021.

[17] M. S. Khan et al., “Deep learning for ocular disease recognition: An inner-class balance,” Comput. Intell. Neurosci., vol. 2022, p. 5007111, 2022.

[18] L. Jain, H. V. S. Murthy, C. Patel and D. Bansal, ”Retinal Eye Disease Detection Using Deep Learning,” 2018 Fourteenth International Conference on Information Processing (ICINPRO), 2018, pp. 1-6, doi: 10.1109/ICINPRO43533.2018.9096838.

[19] Nazir, T.; Irtaza, A.; Javed, A.; Malik, H.; Hussain, D.; Naqvi, R.A. Retinal Image Analysis for Diabetes-Based Eye Disease Detection Using Deep Learning. Appl. Sci. 2020, 10, 6185. https://doi.org/10.3390/app10186185

[20] Sarki, R., Ahmed, K., Wang, H. et al. Automated detection of mild and multi-class diabetic eye diseases using deep learning. Health Inf Sci Syst 8, 32 (2020). https://doi.org/10.1007/s13755-020-00125-5

[21] J. Shan and L. Li, "A Deep Learning Method for Microaneurysm Detection in Fundus Images," 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016, pp. 357-358, doi: 10.1109/CHASE.2016.12.

[22] Malik, S.; Kanwal, N.; Asghar, M.N.; Sadiq, M.A.A.; Karamat, I.; Fleury, M. Data Driven Approach for Eye Disease Classification with Machine Learning. Appl. Sci. 2019, 9, 2789. https://doi.org/10.3390/app9142789

[23] "Ocular disease recognition," https://www.kaggle.com/andrewmvd/ocular-disease-recognition-odir5k.