Eye Disease Detection Using Machine Learning

A Seminar Report

Submitted to the National Institute Of Technology Warangal

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electronics and Communication Engineering

by

Meet Popat - 204157

Ravi Shah - 204160

Shahil Khan - 204165

Supervisor:

Dr. A Prakasa Rao

Associate Professor

Department of Electronics and Communication Engineering

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

May 2024

Approval Sheet

This project work entitled **”Eye Disease Detection Using Machine Learning”**

done by

Meet Popat (204157)

Ravi Shah (204160)

shahil Khan (204165)

is approved for the degree of Bachelor of Technology.

Examiners

____

____

____

____

____

____

____

Supervisor

Dr. A Prakasa Rao

(Associate Professor, Department of Electronics and Communication Engineering)

Head of the Department

Prof. D. Vakula

(Professor, Department of Electronics and Communication Engineering)

Date: 14-05-2024 **Place:**Warangal

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY WARNAGAL

2023 - 2024

CERTIFICATE

This is to certify that the report entitled **Eye Disease Detection Using Machine Learning** submitted by **Meet Popat (204157)**, **Ravi Shah (204160)**, **Shahil Khan (204165)**, to the National Institute Of Technology Warangal in partial fulfillment of the B.Tech. in Electronics and Communication Engineering is a bonafide record of the seminar work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. D Vakula
HOD
Professor
Dept. of ECE
National Institute Of Technology
Warangal

Dr. A Prakasa Rao
Guide
Associate Professor
Dept. of ECE
National Institute Of Technology
Warangal

Declaration

I Meet Popat, Ravi Shah and Sahil Khan hereby declare that the seminar report **Eye Disease Detection Using Machine Learning**, submitted for partial fulfillment of the requirements for the award of the degree of Bachelor of Technology of the National Institute Of Technology Warangal is a bonafide work done by me under supervision of Dr. A Prakasa Rao

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the sources.

I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data idea fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute.

Roll no	Name Of the Student	Signature
204157	Meet Popat	
204160	Ravi Shah	
204165	Shahil Khan	

Warangal

14-05-2024

Acknowledgement

I take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. I express my sincere thanks to **Dr. D Vakula**, Head of Department, Electronics and Communication Engineering, National Institue Of Technology Warangal Warangal for providing me with all the necessary facilities and support.

I would like to place on record my sincere gratitude to my guide **Dr. A Prakasa Rao**, Associate Professor, Electronics and Communication Engineering, National Institue Of Technology Warangal for the guidance and mentorship throughout the course.

Finally I thank my family, and friends who contributed to the succesful fulfilment of this project work.

Meet Popat - 204157

Ravi Shah - 204160

Sahil Khan - 204165

Abstract

Diabetic retinopathy, glaucoma, and cataracts are among the most widespread eye-related diseases, posing significant challenges in the realm of global public health. Early diagnosis and intervention are critical to preventing irreversible vision impairment. This study focuses on the development of an efficient deep learning algorithm for the detection of eye diseases depicted in fundus images.

The EfficientNet model, a state-of-the-art CNN-based model recognized for its efficiency and performance in image classification tasks, was employed. An accuracy greater than 95% was achieved by the deep learning model utilized for eye disease detection, surpassing previous works in this field.

Thus, a fast, accurate, and user-friendly solution for diagnosing eye disorders is provided by the system using deep learning algorithms, assisting patients in seeking consultation with an ophthalmologist for screening purposes.

Contents

Chapter 1

Introduction

1.1 Problem Statement

Despite India's high burden of blindness, 85% of cases being treatable, there's a critical need for an efficient diagnostic system. This study aims to create an automatic, deep learning-based solution to rapidly detect Diabetic Retinopathy, Glaucoma, and Cataracts in fundus images with high accuracy. Current categorization methods are limited, and the proposed deep convolutional neural network (DCNN) model offers a promising approach. The study seeks to achieve detection accuracies of 91% for Diabetic Retinopathy, 89% for Cataracts, and 85% for Glaucoma. An intuitive online interface will make accessing this diagnostic tool easy for users.

1.2 Applications

The proposed deep learning-based diagnostic system for eye diseases has various potential applications, including:

Early Disease Detection

The system can help identify Diabetic Retinopathy, Glaucoma, and Cataract at an early stage, enabling timely medical intervention and potentially preventing vision loss.

Telemedicine Support

This technology can be integrated into telemedicine platforms, allowing remote patients to access quick and accurate eye disease diagnoses, particularly in underserved rural areas.

Screening Campaigns

The system can be employed in mass screening campaigns, making it possible to screen a large number of individuals efficiently, thus contributing to public health initiatives.

Resource Optimization

By automating the diagnostic process, healthcare resources, including specialized personnel, can be allocated more efficiently, reducing the burden on eye care professionals.

Patient Education

The online Graphical User Interface (GUI) makes the system accessible to patients, helping them understand their condition and treatment options better, leading to improved patient engagement and compliance.

Research and Data Collection

The system can be used for data collection and analysis, supporting further research into eye diseases and potentially leading to the development of new treatment strategies and public health policies.

1.3 Model Specification

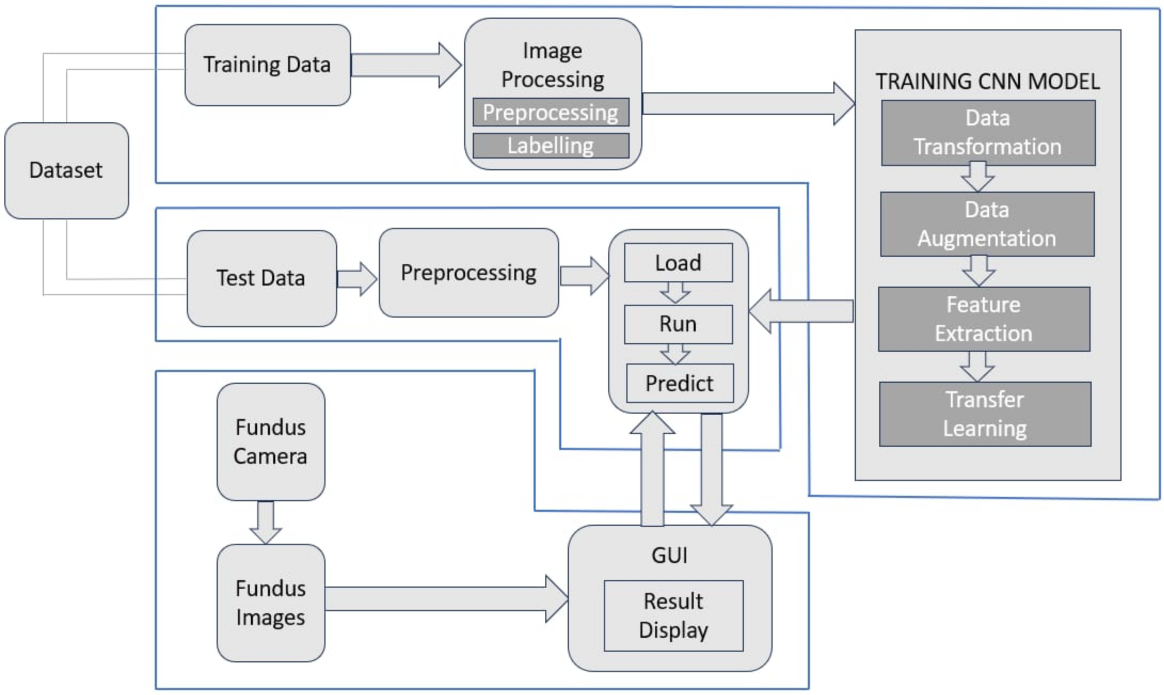

Figure 1.1: Model speification

1.4 Different Models

EfficientNet Architecture

EfficientNet, a convolutional neural network, utilizes "compound scaling" to balance model complexity, accuracy, and computational efficiency. This technique involves adjusting three crucial factors: width (channels), depth (layers), and resolution (input image size) of the network architecture. Width scaling enhances complexity, depth scaling captures intricate representations, and resolution scaling improves detail. The model employs Mobile Inverted Bottleneck (MBConv) layers, featuring depth-wise and point-wise convolutions, and Squeeze-and-Excitation (SE) blocks for optimization. The MBConv layer efficiently maintains representational power, and the SE block focuses on essential features. EfficientNet variants (B0, B1, etc.) offer different trade-offs between size and accuracy to cater to diverse requirements.

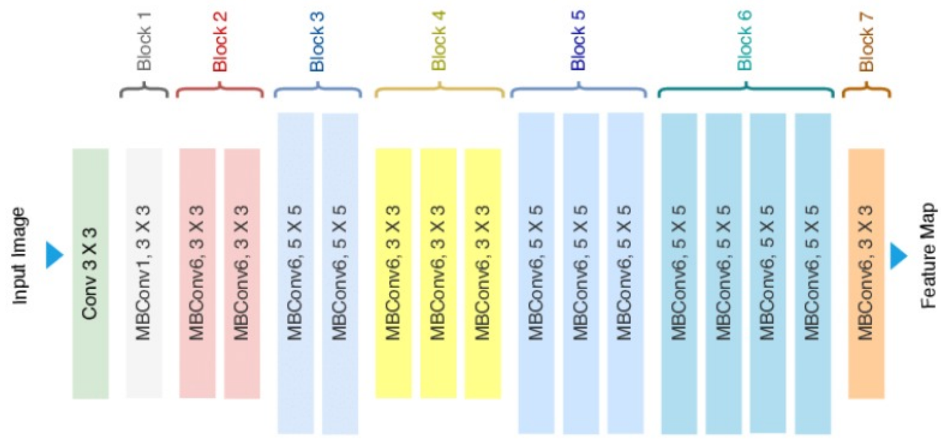

Figure 1.2: EfficientNet Architecture

MobileNetV2

MobileNetV2, tailored for mobile devices, employs inverted residual blocks, linear bottlenecks, and skip connections to strike a balance between model size and accuracy. By leveraging depth-wise separable convolutions, its architecture ensures a lightweight design, ideal for resource-limited environments, while maintaining competitive performance in tasks such as image classification and object detection.

In MobileNetV2, two types of blocks are employed: one with a stride of 1 (residual block) and the other with a stride of 2 for downsizing. Both block types consist of three layers: a 1×1 convolution with ReLU6, a depthwise convolution, and a 1×1 convolution without additional non-linearity. This design, excluding ReLU in the last layer, prevents diminishing the model's expressive power. The expansion factor t is set to 6 for all main experiments, resulting in an internal output of 64×6=384 channels when the input has 64 channels.

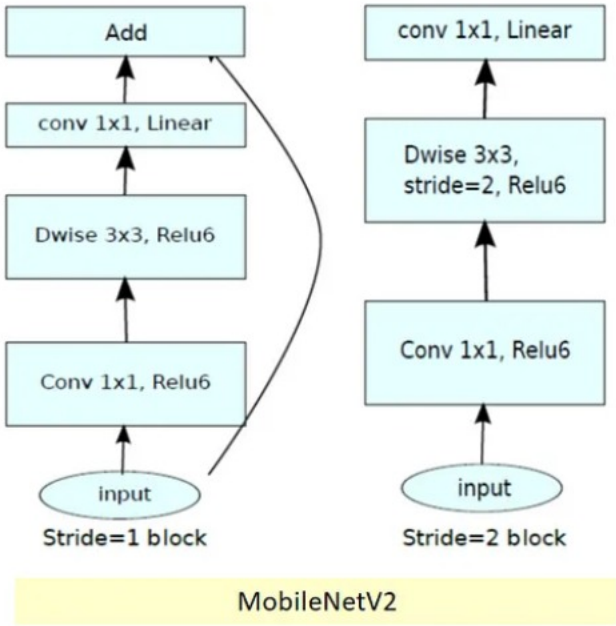

Figure 1.3: MobileNetV2 Architecture

NASNet (Neural Architecture Search Network)

NASNet (Neural Architecture Search Network) is a neural network architecture designed through automated neural architecture search. Developed by Google's DeepMind and Google Brain teams, NASNet employs a repeating cell structure containing normal and reduction cells, each discovered through reinforcement learning during the search process. The architecture is efficient and scalable, adapting well to various tasks and computational resources. NASNet demonstrates competitive performance across various computer vision tasks, including object detection and image classification. Its flexibility lies in automatically discovering architectures suited to different problem domains. The use of neural architecture search allows NASNet to dynamically optimize its structure for specific tasks, demonstrating its adaptability and effectiveness in state-of-the-art performance on benchmark datasets.

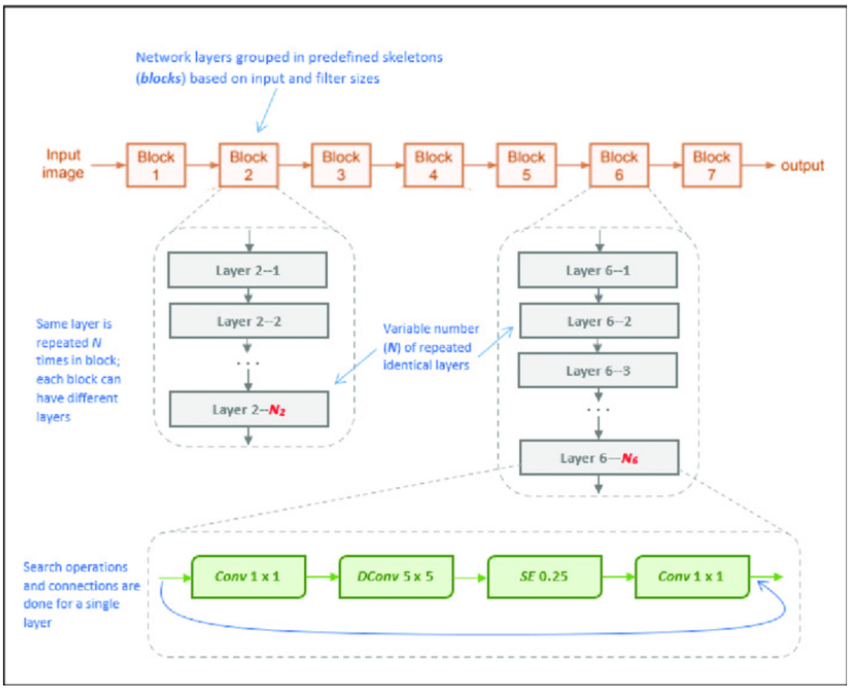

Figure 1.4: NASNet Architecture

1.5 Merits And Demerits

Merits

The merits of doing this projects are:

1. **Improved Healthcare Access**

   The proposed deep learning-based diagnostic system addresses a critical need in India, where access to eye care services can be limited in certain regions. By providing an automated and efficient diagnostic tool, it enhances access to early disease detection and treatment.

2. **Rapid and Accurate Diagnoses**

   The system's ability to diagnose Diabetic Retinopathy, Glaucoma, and Cataract within a minute with high accuracy is a significant merit. Early detection of these conditions can significantly improve patients' prognosis and quality of life.

3. **Telemedicine and Outreach**

   The application of this system in telemedicine and mass screening campaigns extends its reach beyond traditional healthcare settings, making it a valuable tool in reaching underserved populations and increasing the efficiency of healthcare deliver.

Demerits

The demerits are

1. **Limited Diagnostic Scope**

   While the system is designed for the specific diagnosis of Diabetic Retinopathy, Glaucoma, and Cataract, it may not address a broader range of eye diseases or conditions. This limitation could be considered a demerit, as it doesn't provide a comprehensive solution for all eye-related ailments.

2. **Data and Infrastructure Challenges**

   The effectiveness of advanced learning algorithms, such as the suggested DCNN, relies heavily on access to top-tier data and resilient technological frameworks. Collecting fundus images and maintaining the necessary resources may pose challenges, particularly in resource-constrained healthcare settings.

3. **Interpretation and Reliability**

   Despite high accuracy rates, deep learning models can sometimes produce false positives or negatives, and their decision-making processes are often considered ”black boxes.” Ensuring the interpretability and reliability of the system's diagnoses is essential for gaining the trust of both patients and healthcare professionals

Chapter 2

Literature Review

2.1 Eye Diseases Detection Using Machine Learning

It proposed a methodology for cataract detection and classification. While it contains technical details and observations related to the methodology.
Cataract detection and classification are crucial tasks in the field of ophthalmology and computer vision. Various methodologies and techniques have been employed to develop accurate and efficient models for this purpose.

Data Preprocessing and Feature Extraction

The methodology begins with data preprocessing, where required libraries like pandas, scikit-learn (Sklearn) and NumPy are imported. These libraries are used for various tasks, including data loading, processing, and model selection. Data shuffling is applied to ensure randomness in the dataset.

Observation

The text mentions that preprocessing is essential for efficient model training. It highlights the importance of proper data organization.

1. **Image Processing**

   Image processing holds significant importance in extracting features. The discussion encompasses diverse image processing methodologies, including histogram analysis, to extract features such as mean, energy, standard deviations, entropy, and log-kurtosis. Furthermore, the text highlights the utilization of adaptive thresholding to extract valuable information.

Observation

Image processing techniques are essential for extracting meaningful features from eye images, contributing to the accuracy of the classification model.

2. Model Building

The methodology entails developing a multiclass classification model utilizing four distinct types of classifiers: logistic regression, random forest, gradient boosting, and support vector machine (SVM). Each classifier possesses its distinct characteristics and applications.

3. Logistic Regression

This supervised machine learning algorithm is used for classification. The specific type mentioned is multinomial logistic regression.

4. Random Forest

A classification algorithm based on decision trees. It leverages bagging and features randomness to improve prediction accuracy.

5. Gradient Boosting

This technique aims to minimize prediction error by combining models in a boosting fashion.

6. Support Vector Machine (SVM)

A supervised learning algorithm used for both regression and classification, with a focus on categorizing the data into sets based on categories.

Observation

The choice of different classifiers allows for comparative analysis of their performance in classifying various eye diseases.

7. ROC Curve Analysis

Receiver Operating Characteristic (ROC) curves are employed to evaluate the models' performance. ROC analysis is a crucial tool for assessing the ability of the model to predict binary outcomes. The use of ROC analysis indicates a focus on evaluating the models based on their predictive capabilities.

8. Accuracy Assessment:

Finally, the methodology assesses the accuracy of the classifiers for detecting different eye diseases, including glaucoma, retina disease, cataract, and normal eyes. The accuracy measurements help determine the best-performing classification algorithm.

Observation

The ultimate goal is to achieve accurate and specific predictions for various eye diseases, with the methodology providing a systematic approach for this purpose.

2.2 An Online Platform for Early Eye Disease Detection using Deep Convolutional Neural Networks

It proposed a system for picture segmentation using a Convolutional Neural Network (CNN) for the early detection of eye diseases.

Use of Convolutional Neural Networks (CNNs)

The proposed system employs CNNs, a deep learning technique known for its effectiveness in image analysis and recognition. CNNs have been widely adopted in various medical image analysis tasks, including the detection of eye diseases such as diabetic retinopathy, glaucoma, and cataract.

1. **Data Requirements**

   The text underscores the necessity of ample data for training neural networks, aligning with the widely accepted notion in the realm of deep learning. Access to vast and varied datasets is paramount for training precise models. Researchers frequently resort to publicly accessible medical image datasets, such as the EyePACS dataset for diabetic retinopathy.

2. **Image Pre-processing**

   Image pre-processing is a critical step in medical image analysis. Techniques mentioned in the text, such as contrast enhancement, resizing, and grayscale conversion, are commonly used to improve image quality and prepare the data for CNN-based analysis.

3. **Convolutional Neural Network Architecture**

   CNNs consist of convolutional layers, pooling layers, and fully connected layers.

These layers are fundamental components of CNN architecture, which has shown significant success in various image classification tasks. The text provides a basic overview of these layers, but more specific details on the network architecture would be needed to understand the proposed model fully.

4. Web Interface

The text mentions the development of an online GUI platform named Jarvis Eye Care for users to detect eye defects. User-friendly interfaces for medical diagnosis have become more common with the rise of telemedicine and digital healthcare solutions.

5. Results

The text presents accuracy results for the proposed system in detecting diabetic retinopathy, glaucoma, and cataract. Achieving accuracy levels of over 90% is promising and suggests that the CNN-based model performs well in diagnosing these eye diseases.

6. Performance Comparison

The text compares the proposed deep learning approach with decision tree and random forest algorithms in terms of accuracy. Such comparative studies are essential for assessing the effectiveness of deep learning models in medical image analysis.

2.3 Multiple eye disease detection using Deep Neural Network

In this, we explore a two-phase approach for the detection of eye diseases, focusing on the training and testing of a deep convolutional neural network (CNN) model and the subsequent development of a user-friendly graphical user interface (GUI) for instant disease detection. The datasets for Diabetic Retinopathy (DR) and Glaucoma were obtained from Medimrg and Kaggle, respectively.

Phase One

Training and Testing of Model : This phase includes crucial steps for training and evaluating the CNN model using the obtained datasets. The datasets are partitioned into training and testing sets in an 80:20 ratio. Key activities in this phase involve:

1. **Training the Model**

   The preprocessed training images are fed into a deep convolutional neural network. Features are extracted through convolution, pooling, and ReLU layers, resulting in a high-level abstraction of the input. The proffered deep neural network comprises five latent layers, enabling it to extract diverse features from the input images.

2. **Neural Network Architecture**

   The CNN architecture employs Convolutional Layers, Pooling Layers, and Dense Layers. Convolutional layers perform feature extraction through a convolution operation with filters or kernels, while pooling layers reduce spatial size and complexity. The dense layer is tasked with categorizing the features extracted by the preceding layers.

3. **Testing the Model**

   The testing phase involves preprocessing the test data and running it through the model. The model searches for potential disease-related features and provides an output based on its training. If features indicative of a disease are found, the system alerts the presence of Diabetic Retinopathy (DR) or Glaucoma.

Phase Two

The second phase is dedicated to developing a easy-to-use GUI that facilitates real-time detection of eye diseases. This GUI implementation enhances the accessibility of the model and includes the following components:

1. **User-Friendly Website**

   A website is designed to facilitate real-time submission of fundus images taken with a fundus camera. Users can choose to upload images for DR or Glaucoma testing.

2. **Image Preprocessing**

   The uploaded image undergoes preprocessing to meet the model's requirements, ensuring that it is ready for analysis by the CNN architecture.

3. **Model Execution**

   The preprocessed image is passed through the CNN architecture, and the model leverages the knowledge acquired during training to make predictions. The results are displayed on the GUI with corresponding confidence percentages.

4. **Result Display**

   The outcomes displayed on the GUI indicate whether a disease is detected or not. Users are prompted to seek advice from a healthcare professional if a disease is identified.

Software Stack

The creation of the neural network model and GUI is dependent on a variety of software tools and frameworks. Anaconda, an open-source platform, is employed alongside TensorFlow and Keras for model development and evaluation. The GUI is constructed using Django, a Python-based open-source web framework, while HTML5 is utilized for crafting webpages for the user interface.

Limitations

The text does not discuss the limitations or potential challenges of the proposed system. It's essential to acknowledge potential issues and considerations, such as the need for a large amount of training data, potential biases in the dataset, and generalization to real-world clinical scenarios.

2.4 EYENET: An Eye Disease Detection System using Convolutional Neural Network

The provided methodology presents a detailed process for developing and analyzing the EyeNet model for the classification of eye diseases.

1. Dataset

The dataset used in this research consists of 383 samples of different optical diseases. The dataset split into training and testing sets is essential for machine learning model development. However, the dataset's size might be relatively small compared to other deep learning applications. The use of Kaggle for obtaining datasets is a common practice in machine learning research.

2. Image Processing

Pre-processing is an important process in medical image analysis, as it helps in enhancing image quality and preparing data for the model. The resizing of images to a consistent 100x100 size and RGB channel orientation is a typical practice in deep learning.

3. Model Architecture

The EyeNet model architecture is described in detail, including the number of layers, filter sizes, activation functions, and neuron counts. The model appears to be a Convolutional Neural Network (CNN), which is a standard choice for image classification tasks.

4. Training and Testing

The research presents the results of training the EyeNet model with different training/testing data splits (70:30, 80:20, and 90:10). Such variations help understand the model's performance under different data conditions.

5. Learning Rate

The choice of learning rate is a hyperparameter that significantly impacts the training process and model performance. The analysis shows that a learning rate of 0.001 led to an accuracy of 93%, and other learning rates were less effective. This aligns with the importance of tuning hyperparameters in deep learning models.

6. Performance Metrics

The research uses standard performance metrics, including accuracy, precision, recall, and F1-score, to evaluate the model's classification performance. These metrics are widely accepted in deep and machine learning for assessing model capability.

7. Confusion Matrix

The confusion matrix is used to calculate performance metrics. True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are computed to evaluate the model's accuracy and precision.

8. Class-Wise Classification Report

The classification report offers insights into the model's performance effectiveness for each class (i.e., each type of eye disease). It shows class-specific accuracy, precision, recall, and F1-score. Such analysis is crucial for understanding the model's strengths and weaknesses for different disease types.

The methodology follows standard practices in deep learning research, from data preparation to model architecture, training, and evaluation. However, to assess the completeness and effectiveness of this research, it is essential to compare the proposed EyeNet model with existing methods and datasets, and explore generalizability to real-world clinical scenarios. Additionally, a more extensive dataset and validation against a broader range of eye diseases would further strengthen the research.

2.5 Glaucoma Disease Detection Using Deep Learning

The provided methodology outlines the development of a prediction system for Glaucoma detection, from dataset selection to model development and the creation of a web-based user interface.

Dataset Selection

The methodology emphasizes the significance of dataset selection, highlighting that data is a crucial component of prediction systems. It mentions that the data was gathered from an official government website and was open to the public. Data selection is a pivotal stage in machine learning, and leveraging publicly accessible data sources is a widespread approach. Researchers often choose datasets that align with the specific criteria and factors relevant to their prediction system.

1. Data Proprecessing

Data pre-processing is identified as a critical step in deep learning implementation. The process involves managing missing data, encoding categorical data, splitting the dataset, and feature scaling. Data processing is a standard practice in machine learning, as it ensures that data is in a usable format for model training.

2. Training and Testing

The methodology discusses the importance of training and testing data. It mentions the use of a 70:30 or 80:20 split for training and testing. Splitting data into training set and testing set is a fundamental step to assess a model's performance, allowing for model training on one subset and testing on another to evaluate its accuracy.

3. CNN Model Development

The methodology outlines the creation of a Convolutional Neural Network (CNN) model, which is a standard choice for image classification tasks. The description includes the CNN architecture, specifying count of layers, filter sizes, activation functions, and more. CNNs are widely used in medical image analysis, including the detection of eye diseases like Glaucoma.

4. Web Development

The creation of a web-based user interface using Streamlit for Glaucoma prediction is highlighted. This interface allows users to upload fundus images for analysis. Streamlit is introduced as a tool for integrating web pages with deep learning code. User-friendly web interfaces for medical diagnosis have gained importance with the rise of telemedicine and digital healthcare solutions.

5. Algorithms

The methodology includes algorithm descriptions for different modules, such as dataset training and testing split, CNN model development, and the development of the prediction page. These algorithms detail the steps involved in data processing, model training, and web page development, providing a structured approach to the entire process.

6. Result and Analysis

The methodology presents the results and analysis of each module. This includes the splitting of the dataset, the CNN model's code, and the user interface's functionality. Results include the classification of fundus images as either Glaucoma-affected or healthy, along with appropriate messages for users.

7. Performance Metrices

While the methodology mentions accuracy and classification results for Glaucoma detection, it does not detail the performance metrics used. It's crucial to include metrics like accuracy, precision, recall, and F1-score to assess the model's effectiveness.

Further research should be done to compare the proposed approach with existing Glaucoma detection methods and to assess the generalizability of the system to real-world clinical scenarios.

2.6 Ocular Disease Recognition using Deep Learning

Ocular Disease Recognition utilizes Deep Learning to analyze eye images, detecting conditions like diabetic retinopathy and glaucoma. By leveraging neural networks, this technology enables early diagnosis, paving the way for timely treatment and preventing vision loss. Its integration into ophthalmic diagnostics enhances efficiency and accessibility, improving overall eye care.

1. Data Collection

The methodology begins with data collection, which is a pivotal step in any machine learning project. The dataset in this research comprises images of both right and left eyes from around 4950 patients, including details such as gender and age. These images are color fundus images, which are widely used in ophthalmology for disease diagnosis. Quality control management, which includes annotations by trained human readers, adds credibility to the dataset. Categorizing patients into eight distinct classes, including various eye conditions, enables a comprehensive analysis.

2. Data Preprocessing

The methodology mentions the utilization of information related to cataracts and normal eye conditions for research purposes. Data preprocessing involves tasks like creating a dataset from the collected images and resizing them. Data preprocessing is a crucial step in guaranteeing that the data is appropriately formatted for model training and analysis.

Overview of Proposed Models

The methodology highlights the implementation of three distinct deep learning models for cataract prediction: Convolutional Neural Network (CNN), VGG 19, and Inception V3

1. Convolutional Neural Network (CNN)

The methodology provides an introduction to CNNs, emphasizing their architecture modeled after the human brain. CNNs consist of layers connected through artificial neurons. Convolutional layers in CNNs play a key role in identifying patterns and features in images. The methodology highlights that ConvNets are adept at compressing images into manageable formats while retaining crucial information for accurate predictions. Additionally, the explanation of neurons, activation functions, and the flow of information within CNN layers is informative.

2. VGG 19

The VGG 19 model is presented as an alternative to CNN. It is described as having a specific architecture, including a receptive field of size 3x3 and 19 convolutional layers. The presence of five such levels, each accompanied by a 2x2 Max pooling layer, is explained. The methodology notes the inclusion of fully connected layers and the usage of the softmax classifier. VGG 19's extensive parameters are acknowledged, and the methodology hints at replacing some fully connected layers with an SVM classifier to reduce the parameter count, which can improve efficiency

3. Inception V3

The introduction of Inception V3 is included, emphasizing its association with

GoogleNet and its deep learning nature. Inception V3's architecture is briefly described with specific details such as the number of layers and the inclusion of auxiliary classifiers for label propagation. Additionally, the methodology mentions the usage of RMPS optimizers and label smoothing as regularization techniques. The mention of label smoothing indicates a conscious effort to mitigate overfitting in the model.

Result

The VGG-19 model emerged as the top performer, exhibiting the highest accuracy compared to other models such as CNN and Inception-V3. Its superior accuracy underscores the need for novel model creation aimed at showcasing and extracting characteristics pivotal for the prognosis, diagnosis, and monitoring of ocular disorders. This is crucial for the effective utilization of cutting-edge medical equipment in detecting eye diseases.

Limitations

Deep learning models necessitate extensive datasets to ascertain model accuracy. However, dataset collection often necessitates equipment that might not be readily available in local hospitals, posing a barrier to collecting data on a larger scale.

2.7 RetinalNet-500: A newly developed CNN Model for Eye Disease Detection

The proposed methodology introduces the architecture of RetinalNet-500, which aims to predict eye diseases based on retinal images. The key components and processes are as follows:

The CNN Architecture

The methodology describes the architecture of RetinalNet-500, which begins with an input layer accepting imageswith a resolution of 222 pixels by 222 pixels. This architecture comprises sets of convolutional, max-pooling and activation layers stacked in succession, followed by two sets of fully connected latent layers and the output layer. The valid-padding used in the neural layers is noted.

1. Convolutional Layer

The first two layers consist of 32 filters having a size of 3x3, and the third convolutional layer has 64 filters with the same size. All these filters convolve over the input image with no striding, resulting in feature extraction from the images.

2. Activation Function

The ReLU activation function is applied between the layers to enhance edge detection. ReLU is utilized in the initial dense layer, followed by the application of softmax after the second dense layer.

3. Pooling Layers

Max-pooling is applied using a 2x2 filter in each pooling layer. This operation reduces the spatial dimensions and helps with feature selection

4. Dropout Layer

A dropout layer with a dropout rate of 0.5 is applied after the first dense layer to mitigate overfitting concerns

5. **Hidden Layer**

The hidden layers consist of 500 fully connected units to uncover hidden patterns in the images

6. **Loss Function, optimizer and Output Layer**

The model uses RMSProp as an optimizer and categorical cross-entropy as the loss function for multi-class classification. A softmax activation function with three outputs is applied to produce probability distributions across the three classes.

The methodology highlights the use of specific hyperparameters, such as a batch size of 32, 0.01 learning rate, and 10 epochs for training. The rationale for these choices, including their impact on training, is briefly explained.

Input Data

Input data diverse eye images, enabling neural networks to discern patterns indicative of conditions like diabetic retinopathy and glaucoma.

1. **Data Collection**

The methodology discusses the collection of approximately 6000 retinal images from a local eye hospital. These images are categorized into four classes: Normal, Glaucoma, Cataract, and Diabetic Retinopathy. The dataset is split into training and validation sets in an 80:20 ratio to enable experimentation and evaluation of the model.

2. **Dataset Sample**

The methodology mentions the inclusion of sample images from the dataset, although it is challenging to display the vast dataset comprehensively. A total of 60 images are displayed, featuring samples from each of the four disease categories. This provides an overview of the dataset's diversity

3. **Data Pre-Processing**

Data pre-processing is performed to extract features from the dataset. The methodology describes the use of libraries and resizing images to the specified dimensions. The Prewitt operator, used for edge detection, is also explained as a pre-processing step.

Experiments and Results

The experiments section presents the accuracy of various models, including InceptionV3, MobileNetV2, Xception, and the newly proposed RetinalNet-500. The accuracy for each model, where InceptionV3 and MobileNetV2 achieved the highest accuracy of 97.30%, Xception achieved 96.45%, and RetinalNet-500 achieved an accuracy of 95.15%.

Limitations

This methodology outlines the architecture of RetinalNet-500 and the data processing steps to create a dataset for eye disease prediction. The experiments demonstrate the performance of this model in comparison to other pre-trained models. Further research may delve into additional details about model training, fine-tuning, and potential challenges in eye disease prediction. Moreover, the implications and real-world applications of these models in healthcare can be explored.

Chapter 3

REPORT ON CURRENT INVESTIGATION

3.1 WORKFLOW

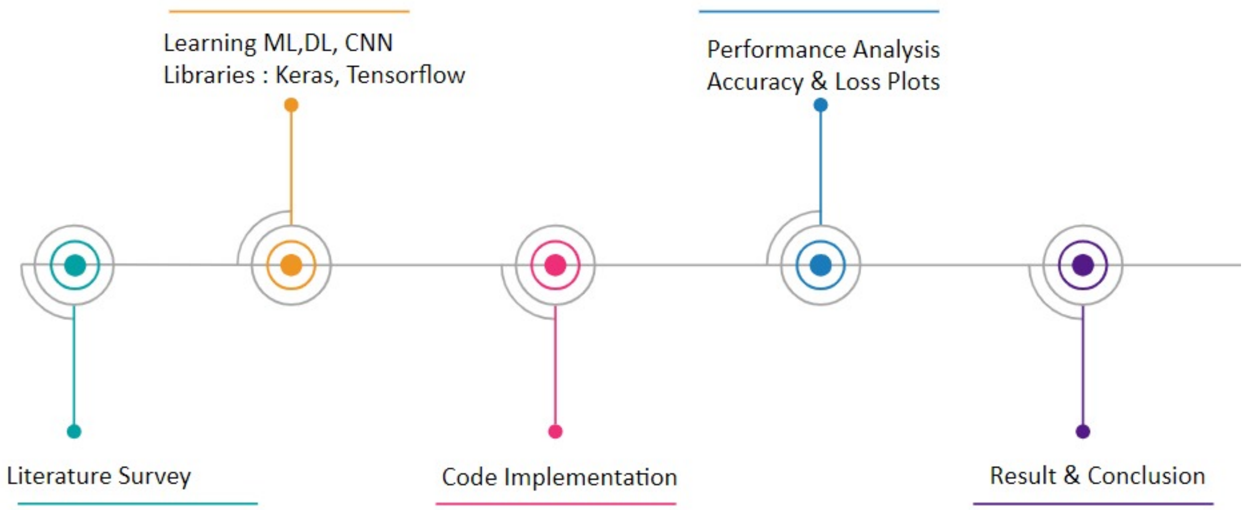

Figure 3.1: Workflow

3.2 WORK DONE

Understanding of earlier used techniques

Prior techniques relied on manual interpretation, lacking the performance and precision of modern Deep Learning methods in recognizing subtle patterns in eye images.

1. **Logistic Regression**

   Logistic regression is a type of supervised machine learning algorithm commonly employed for classification purposes.It analyzes how independent variables (features) are associated with a dependent variable (the target class or category). In this case, it is used for multiclass classification, where there are more than two classes.
   Multinomial logistic regression is a specific variant of logistic regression tailored for multiclass classification, allowing the model to predict which of the multiple classes a data point belongs to base on the input features.

2. **Random Forest**

   Random forest is a type of classification algorithm that utilizes an ensemble of decision trees.It employs a technique called bagging (Bootstrap Aggregating) and introduces randomness in feature selection during the creation of individual decision trees. This randomness aims to create a diverse set of trees, which collectively provide more accurate predictions.
   The ”committee” of decision trees in a random forest model combines their predictions to improve accuracy. The key idea is that the ensemble's predictions are often more reliable than those of individual trees.

3. **Gradient Boosting**

   Gradient boosting is a boosting algorithm in ML.It works on the idea that the most effective model to add to the ensemble is the one that, when combined with the previous models, decreases the overall prediction error the most.
   Gradient boosting builds models sequentially, with each new model addressing

the errors or residual errors of the previous models. This process continues until the prediction errors are minimized to the greatest extent possible.

4. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a flexible supervised machine learning algorithm capable of performing both classification and regression tasks.
In this context, SVM is applied as a classifier to categorize data into the provided classes.
SVM works by creating boundaries or ”hyperplanes” that separate different classes in the data. It is particularly effective for binary and multiclass classification tasks, as it separates the input data into groups according to the specified number of categories.

3.3 Data Pre-Processing

In the ”Image Preprocessing” section, several essential steps are outlined to improve the quality of fundus images and address data-related issues

A. Image Enhancement

i. Extract the green channel from RGB images, providing additional information.

ii. Enhance image contrast using Contrast Limited Adaptive Histogram Equalization (CLAHE).

iii. Correct lighting to improve image brightness and luminance

iv. Reduce noise through Gaussian filtering for a smoother image

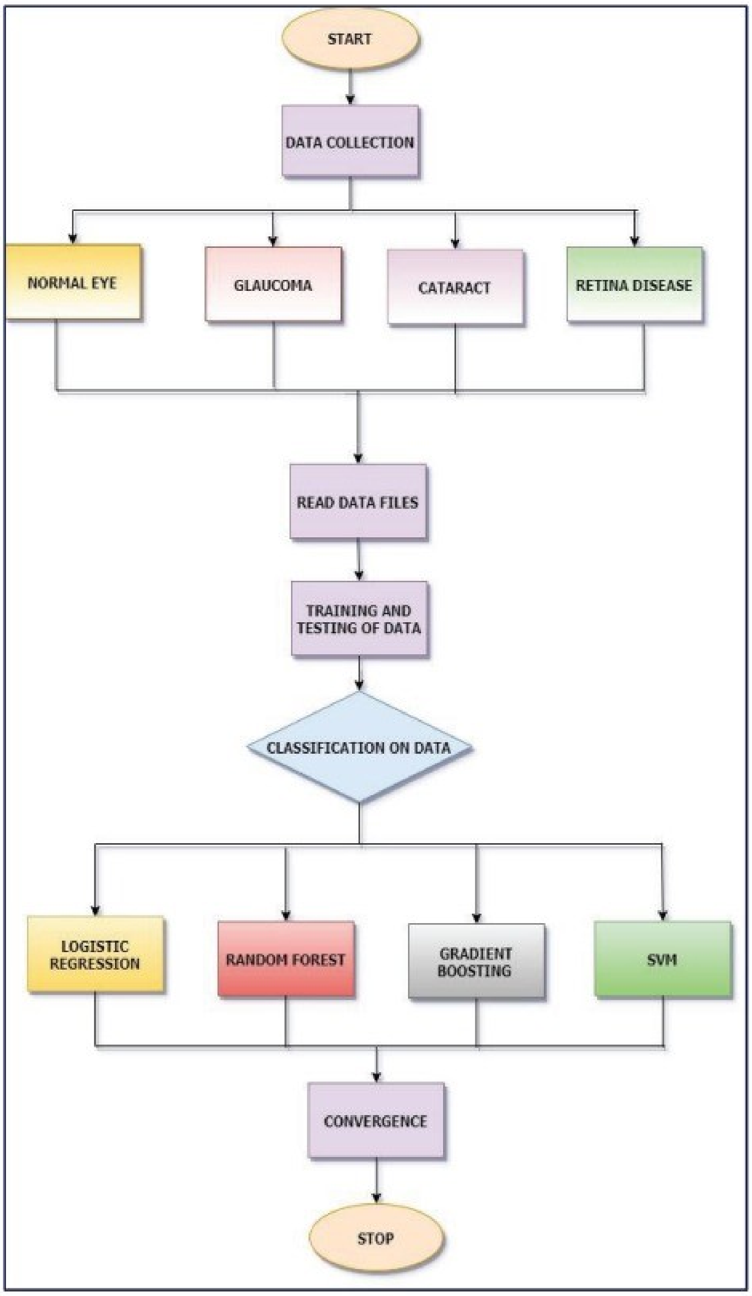

Figure 3.2: Flowchart of Eye Disease Detection

B. Resizing

i. Standardize the size of images for network training by resizing them to a uniform dimension, such as 1000x1000 pixels.

ii. Reduce RGB images to grayscale with a single channel to save time and resources.

C. Grayscale

i. Represent grayscale images using 8-bit brightness values ranging from 0 to 255.

ii. Convert color images (24-bit) to grayscale images (8-bit) for further processing.Convert color images (24-bit) to grayscale images (8-bit) for further processing.

These preprocessing steps are crucial for improving image quality and consistency, which is essential for effective feature extraction and the training of deep learning systems, particularly in cases where image data may have low fidelity and variations in quality.

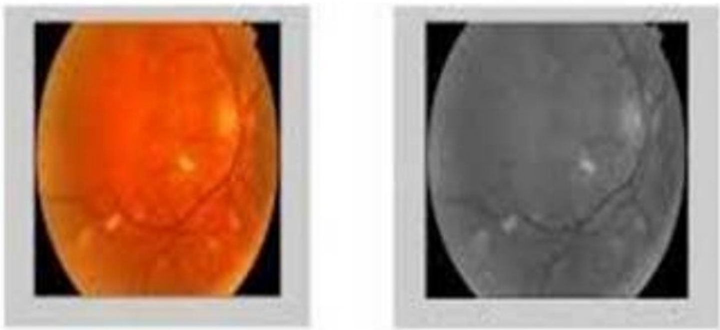

Figure 3.3: Color Space Conversion

3.4 Model Development

CNNs for Image Recognition

i. CNNs are popular for image recognition in deep learning.

ii. Keras, a Python module, simplifies CNN creation.

iii. Convolutions are applied to identify patterns and edges in images.

iv. Convolutions leverage the arrangement of pixels to aid in image recognition

Convolution Layer

i. Convolutions are mathematical operations that transform one function into another, used to blur, sharpen, emboss, or enhance images.

ii. They extract information from the original function, aiding in image analysis.

Pooling Layer

i. Pooling layers reduce data dimensionality by consolidating outputs from one layer into a single neuron in the next layer

ii. Max pooling selects the highest value from neuron clusters, while average pooling uses the average value

Fully Connected Layer

i. In fully connected layers, each neuron in one layer is connected to every neuron in the next layer, similar to a traditional multilayer perceptron (MLP) neural network.

ii. A flattened matrix passes through this layer to classify images.

iii. CNNs leverage these layers to process and analyze images effectively

3.5 Materials and Methodology

Eye disease detection is done in 2 stages. First stage includes training and testing of the model. Second stage is the development of a graphical user interface for real-time disease detection.

3.5.1 Testing and Training of Model

The organization and saving of image datasets on disk for training and testing deep learning models must be done in conformity with the accepted protocols, which can also help to achieve better loading times. This may be achieved using tools such as a Data Generator of image in the Keras package used in deep learning that helps to upload images automatically into separate train, test, and validation sets after formatting. In this way, it ensures efficient dataset photos are incrementally loaded onto memory so that even large-scale image datasets running into hundreds of millions could be processed by any system whose size would otherwise hinder its loading at once.

1. **Dataset**: The dataset is classified into 4 classes: diabetic retinopathy, glaucoma, cataract, and normal eye. The datasets are taken from various contributors across the kaggle website. Source [17] dataset is used in this research. This dataset contain the above four classes, but there are a few technical issues in the dataset for the particular disease- Glaucoma. Hence the dataset for this disease is referred from various different available datasets . Similar findings are evident in other research papers, as observed in reference [13]. Thus, the final total number of images are 3721, containing eye diseases and the normal eye images with following distribution:

   Data is partitioned in ratio of 80-10-10 for the following three processes:

   Training Dataset It is the fundamental data which is used for training the model. Maximum percentage of the overall dataset is used for training purposes. The content of supervised learning is an output variable for one or multiple input variables.

   Validation Dataset A small percentage of dataset is used for validation purposes in DNN models. Once the model is trained and the model is ready for prediction, validation data is used to evaluate the model performance on unfamiliar data. Thus, cross-validation is performed to check the accuracy of the model on unseen data.

   Test Dataset : It has almost the same distribution as the validation dataset and is used for final testing of the model. The model suffers the problem of overfitting if training accuracy is much greater compared to testing accuracy, as the model is inaccurate for new data. Thus, we train our model in a way to get better test accuracy.

2. **Methodology**:

The technique used in this neural network has a precise approach for image processing and the model development. The images preprocessed from the training dataset are given to the proposed deep convolutional network model. A sequence of multiple convolution, pooling, and Rectified Linear Unit (ReLU) layers are present in this model, through which features are extracted from the images. These layers form the hidden layers, which helps in abstracting high-level features from input data. The depth of the neural network affects the training process and thus influences the feature extraction. We have used transfer learning techniques to effectively increase the accuracy of the model keeping the computational resources minimal.

The main objective of this paper is to automate multi-class eye disease detection and evaluate the outcome based on various outcomes. We used transfer learning with EfficientNetB3 as the base model to improve the accuracy. Also, we have compared the accuracy of the model with various other predefined models including mobileNet V3 and resNet. MobileNetV3 is an evolution from MobileNetV1 & MobileNetV2, and is particularly useful for small mobile computing devices having compact parameters and minimal latency. MobileNetV3 further used Squeeze-and-Excitation module to further improve feature extraction and classification accuracy [15]. ResNet, a convolutional neural network (CNN) architecture, tackles the issue of vanishing gradient by employing skip connections [11]. It incorporates residual blocks to enable the learning of residual functions, thereby facilitating the training of the network.

.

Our proposed transfer learning model has an input image of 300X300 with 3 channels. EfficientNetB3 is the base model on which further layers are augmented. The base model comprises a combination of convolutional, fully connected and pooling layers. Convolutional maps are generated by convolutional layers by convolving with input pixels and ReLU activation function is utilized to generate feature maps. Diverse pooling layers, each having distinct filters, are used to identify specific features and image components. The output from tha base model is passed through various dense and dropout layers. Finally, the input image is categorized among 4 classes for disease classification.

Various regularization techniques, such as L2 kernel regularization and L1 activity and bias regularization, are applied within the dense layer to combat overfitting and enhance the model's ability to generalize. We have also used dropout layers with a rate of 0.45 to further prevent overfitting by randomly dropping out some layers during the training process. Adam optimizer is used with a standard learning rate of 0.001 for model training. Model is further optimized using categorical crossentropy loss function to compute difference between predicted and actual class distributions.

In that phase, several critical parameters were determined to fine-tune the training process of a deep-learning model. The number of samples processed per iteration was regulated by using batch size of 40 during training. Such a value affects the speed and accuracy of the training operation. By conducting the training for 30 epochs, which is the complete number of passes over the whole training dataset, the model parameters were optimized iteratively. Furthermore, a patience value of 1 was allotted, implying the number of epochs to be patient for the improvement in monitored outputs before trying to revise the learning rate. For cases in which monitored statistics do not record progress for 3 epochs, training is over to keep the model from overfitting and increasing the stop_patience value. In addition, a threshold of 0.9 was defined in order to dynamically change training by monitoring metrics dependent on the performance of the model. If the training accuracy falls below this point, the monitoring is conducted based on accuracy; otherwise, it is based on validation loss. The learning rate is reduced by 0.5 factor after reaching the patience threshold. Furthermore, training is managed through an inquiry function every 5 epochs which is designated by ask_epoch value.

However, together they do improve the training process and the model's performance. The training process of the model was properly maintained and learning rate was regularly updated at different stages by developing a custom callback function called MyCallback, which was integrated into the pipeline. This callback function keeps track of various model parameters including patience, stop_patience, threshold, factor, batches, epochs, and ask_epoch, thus making the training process much more accurate as well as time and cost efficient.

3.6 Model Evaluation Parameters

Performance Metrics

Evaluation metrics rely on the information provided by the confusion matrix. In assessing method performance, three key parameters are utilized: Accuracy, Precision, and Recall..

1. **F1 Source**

   Four outcomes of classification

   a. **True Positive (TP)**

      The term "TP" represents the count of classifiers correctly identifying instances belonging to the positive class.

   b. **True Negative (TN)**

      The term "TN" denotes the count of classifiers accurately identifying instances belonging to the negative class.

   c. **False Positive (FP)**

      The term "FP" indicates the count of classifiers incorrectly labeling instances as positive when they actually belong to the negative class.

d. **False Negative (FN)**

The term ”FN” represents the count of classifiers erroneously labeling instances as negative when they actually belong to the positive class.

2. **Accuracy**

Accuracy is calculated by summing the number of true positives (TP) and true negatives (TN), then dividing this sum by the total number of instances. The formula for accuracy is as follows:

$$accuracy = (TP + TN)/(TP + FP + TN + FN) \tag{3.1}$$

3. **Precision Value**

Precision is determined by dividing the number of true positive predictions by the total number of positive predictions made by the classifier. The formula for precision is:

$$precision = (TP)/(TP + FP) \tag{3.2}$$

4. **Recall**

Recall measures the ability of a model to correctly identify positive samples. It is calculated by dividing the number of true positive predictions by the total number of actual positive instances.

$$Recall = (TP)/(TP + FN) \tag{3.3}$$

5. **F1 Score**

The F1 score incorporates both the recall and precision scores of a model, providing a balanced assessment of its performance. It is calculated by considering the harmonic mean of precision and recall.

$$F1\ score = (2 * recall * Precision)/(Recall + precision) \tag{3.4}$$

3.7 Code and Software Development

GUI Implementation

A easy to use website was build to present the model in an appealing way. The website allows for instant screening of fundus images taken with a fundus camera.
The uploaded image undergoes preprocessing and is then processed by the **Convolutional Neural Network (CNN) architecture**, built using **Keras** and **Tensorflow**.
The system, based on its training, provides results, which are displayed on the GUI along with corresponding confidence percentages.These percentages indicate the system's confidence in determining the presence or absence of a disease.
Development Tools:
Software tools used for development include **Jupyter** (an open-source platform) with **Keras** and **TensorFlow** for building, training, and testing the network.
The GUI was developed with the help of **FastAPI** for backend deployment, **React** for GUI/frontend
HTML5 was employed to design the webpage, defining the structure and behaviour of web page content.

3.8 Code Implementation

1. Import System Libraries.

2. Import Data Handling Tools.

3. Import Deep Learning Libraries.

4. Create functions to have three DataFrames: **train_df** for training data, **valid_df** for validation data, and **test_df** for testing data. These DataFrames contain file

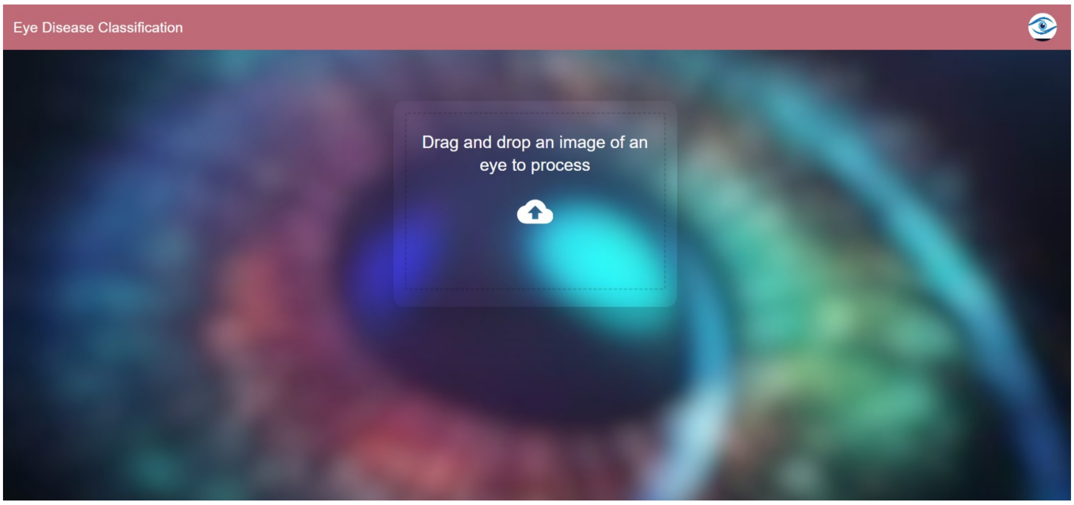

Figure 3.4: GUI First page

paths and corresponding labels, which can be used to load and preprocess images for training and evaluating a machine learning or deep learning model.

5. Define a function **create_gens** that prepares data generators for training, validation, and testing. This function is responsible for setting up data generators that can load and preprocess images from dataframes for training, validation, and testing. These generators are important when working with deep learning models, as they allow you to efficiently handle large datasets and apply data augmentation techniques during training.

6. Define a function called **show_images**. This function allows us to quickly visualize a set of images and their associated class labels from a data generator. It can be helpful for inspecting the data, checking if the images and labels are loaded correctly, and understanding the input that your model will receive during training and testing.

7. Define a custom Keras callback class named **MyCallback**. This custom callback provides a high degree of control and monitoring over the training process, allowing for learning rate adjustments, early stopping, and user interaction during training. This technique can optimize the training process of deep learning models, allowing for effective adaptation to different training situations.

8. Define a function **plot_training**. This function is useful for visually assessing the training progress of a deep learning model. You can see how training and

validation metrics change over epochs and easily identify the epochs with the best performance. This information can be crucial for optimizing the model's architecture and training process.

9. Define a function called **plot_confusion_matrix** that is used to create a visual representation of a confusion matrix. A confusion matrix is a table that is often used to evaluate the performance of a classification model.

10. After implementing all the functions, we prepare data for a machine learning or deep learning model by splitting it into training, validation, and test sets and setting up data generators for these sets. It's important for training and evaluating a model for the classification of eye diseases.

11. We create a deep neural network model that combines a pretrained feature extractor (EfficientNetB3) with custom layers for the specific task of classifying eye diseases. The model is ready for training using the prepared data generators **(train_gen, valid_gen, and test_gen)** and can be used to make predictions on new eye images.

12. We set up the training parameters, calculate the number of batches per epoch, and create a custom callback to manage the training process, including learning rate adjustments and user interaction. This callback can be very useful for controlling and monitoring the training process effectively.

13. The **model.fit** method trains the deep learning model by iteratively processing batches of training data, updating model weights, and evaluating the model's performance on the validation data. The training process continues for the specified number of epochs, with the custom callback **(MyCallback)** monitoring and potentially adjusting the learning rate and asking for user input during training.

14. We assess the model's performance on the training, validation, and test datasets after training. It provides insights into how well the model has learned to classify eye diseases and can help in evaluating its generalization performance.

15. We make predictions using the trained model on the test dataset and printing the predicted class labels for each test image. These predicted labels can be used to assess the model's performance and compare it to the ground truth labels for the test dataset.

16. We evaluate the model's performance, understanding how well it classifies different eye diseases, and identifying areas for improvement. The confusion matrix and classification report are valuable tools for assessing the model's classification accuracy and its ability to distinguish between different classes.

17. We save the model and its weights for future use or further analysis. This is especially important when you want to deploy the trained model for making predictions on new data or to continue training from a specific checkpoint. The saved model can be loaded later using TensorFlow to make predictions or fine-tune the model as needed.

18. We create a record of class information, which can be handy for reference and analysis, especially when dealing with image classification tasks and deep learning models. The saved CSV file can be used for documentation and further analysis of the dataset.

Chapter 4

RESULTS AND OBSERVATIONS

The product underwent evaluation by testing it with both images from the test dataset and those uploaded in real-time for the applicable diseases. In all cases, the output disease is predicted along with the confidence percentage. The training of the model automatically halts when there is no significant improvement in the accuracy and loss curves, based on a callback algorithm. Thus the training process stopped after 15 epochs and achieved test accuracy of 95.3%. Performance analysis parameters are shown in Table 4.1. Training and test accuracies are shown in Figure 4.3 , Training and validation loss and accuracy curves are shown in Figure 4.4. Figure 4.5 represents the confusion matrix.A sample output is shown for all the four outcomes in Figure 4.6

.

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb3_notop.h5
43941136/43941136 [==============================] - 0s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb3 (Functional)  (None, 1536)             10783535

 batch_normalization (BatchN  (None, 1536)             6144
 ormalization)

 dense (Dense)               (None, 256)               393472

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 4)                 1028

=================================================================
Total params: 11,184,179
Trainable params: 11,093,804
Non-trainable params: 90,375
_________________________________________________________________
```

Figure 4.1: EfficientNetB3 Model Summary

```
Do you want model asks you to halt the training [y/n] ?
 n
 Epoch     Loss   Accuracy  V_loss    V_acc     LR      Next LR  Monitor  % Improv  Duration
 1 /30     5.891   82.787   3.73299  87.822   0.00100  0.00100  accuracy     0.00   1312.75
 2 /30     2.040   92.155   1.71805  89.930   0.00100  0.00100  val_loss    53.98   1262.72
 3 /30     1.013   93.326   1.12924  85.012   0.00100  0.00100  val_loss    34.27   1279.15
 4 /30     0.815   90.486   0.71615  88.993   0.00100  0.00100  val_loss    36.58   1335.61
 5 /30     0.511   94.116   0.52596  89.696   0.00100  0.00100  val_loss    26.56   1320.57
 6 /30     0.407   95.521   0.44242  92.272   0.00100  0.00100  val_loss    15.88   1322.47
 7 /30     0.358   95.814   0.48965  88.993   0.00100  0.00050  val_loss   -10.68   1303.24
 8 /30     0.303   96.897   0.38206  91.101   0.00050  0.00050  val_loss    13.64   1404.25
 9 /30     0.246   97.804   0.39001  93.208   0.00050  0.00025  val_loss    -2.08   1389.84
10 /30     0.227   98.390   0.33583  93.208   0.00025  0.00025  val_loss    12.10   1362.90
11 /30     0.196   98.858   0.35290  92.974   0.00025  0.00013  val_loss    -5.08   1285.77
12 /30     0.172   99.444   0.32637  93.911   0.00013  0.00013  val_loss     2.82   1352.01
13 /30     0.164   99.590   0.33194  94.614   0.00013  0.00006  val_loss    -1.71   1344.87
14 /30     0.164   99.297   0.33237  93.443   0.00006  0.00003  val_loss    -1.84   1340.08
15 /30     0.160   99.707   0.33032  93.911   0.00003  0.00002  val_loss    -1.21   1371.81
 training has been halted at epoch 15 after 3 adjustments of learning rate with no improvement
training elapsed time was 5.0 hours, 39.0 minutes, 52.56 seconds)
```

Figure 4.2: EfficientNetB3 Model Training

```
107/107 [==============================] - 312s 3s/step - loss: 0.1385 - accuracy: 0.9988
107/107 [==============================] - 39s 321ms/step - loss: 0.3264 - accuracy: 0.9391
107/107 [==============================] - 38s 354ms/step - loss: 0.2947 - accuracy: 0.9533
Train Loss:  0.13851530849933624
Train Accuracy:  0.9988290667533875
--------------------
Validation Loss:  0.326372891664505
Validation Accuracy:  0.9391100406646729
--------------------
Test Loss:  0.29473981261253357
Test Accuracy:  0.9532710313796997
```

Figure 4.3: Training and Test Accuracies

```
plot_training(history)
```

Figure 4.4: EfficientNetB3 Model Performance

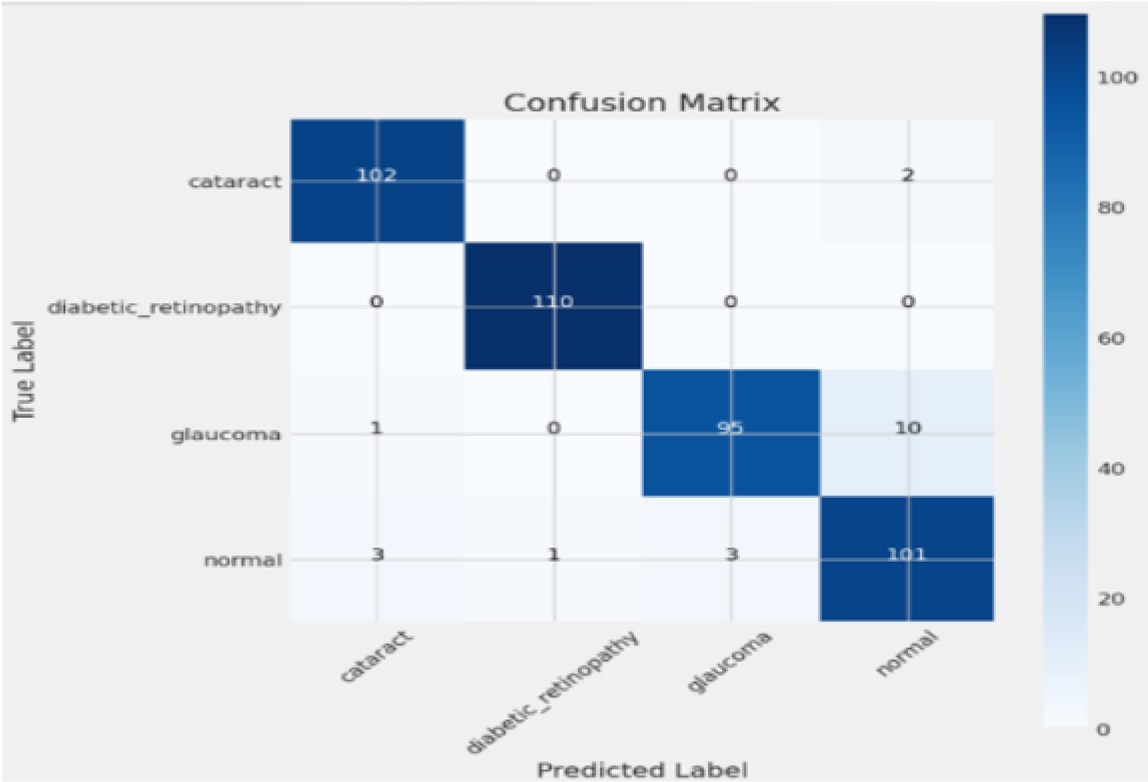

Figure 4.5: Confusion Matrix

	Precision	Recall	F1_score	Support
Cataract	0.96	0.98	0.97	104
Diabetic Retinopathy	0.99	1	1	110
Glaucoma	0.97	0.9	0.93	106
Normal Eye	0.89	0.94	0.91	108

Table 4.1: Performance metrics

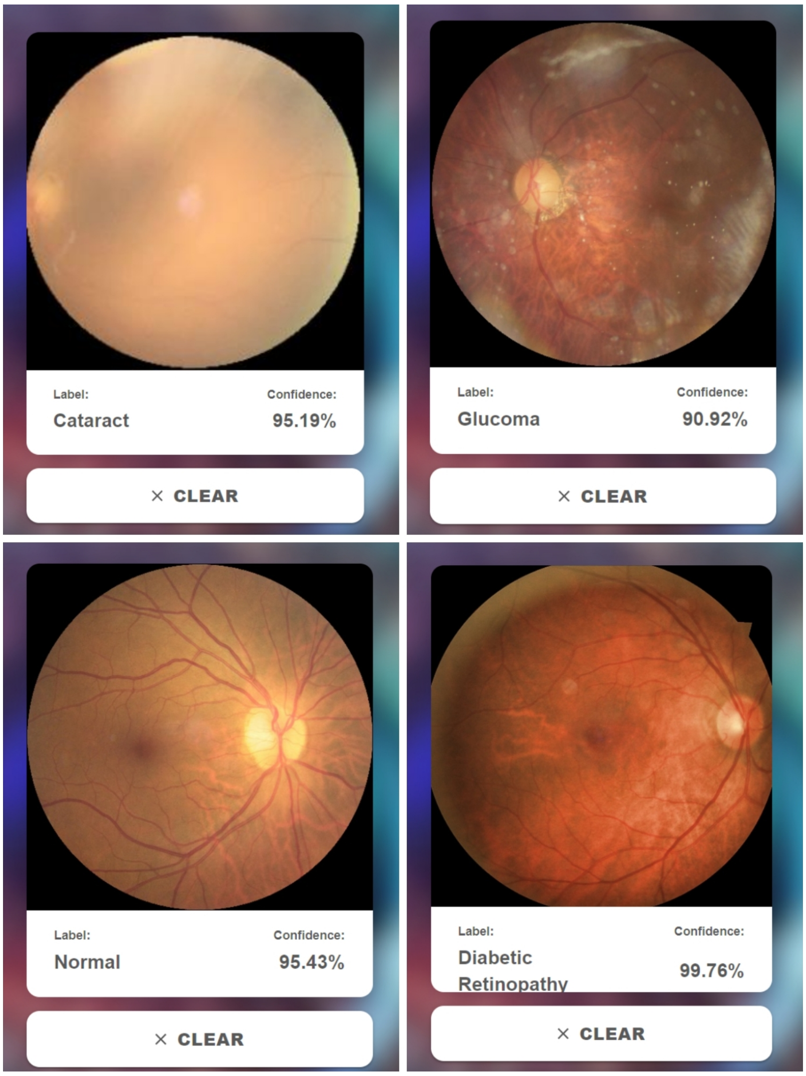

Figure 4.6: GUI showing detected eye diseases

Chapter 5

CONCLUSION

The project has achieved significant success by successfully deploying a multifaceted optical disease detection system catering to four distinct classes: cataract, glaucoma, diabetic retinopathy, and normal ocular conditions. Furthermore, an intuitive graphical user interface (GUI) has been developed to expedite and optimize the preliminary identification process of these eye ailments. Its utility extends to enabling expedited verification by healthcare practitioners and empowering patients with the means for early monitoring, thereby fortifying proactive healthcare management practices. This innovative approach enhances detection speed and ensures cost efficiency, rendering it applicable across a spectrum of scenarios where swift diagnosis is paramount. A research paper has been authored based on our investigation, titled **Automated Multiple Eye Disease Detection via EfficientNet Model using Transfer Learning**. This paper has been submitted for publication at the forthcoming conference, **First International Conference on Distributed Systems, Computer Networks and Cybersecurity (ICDSCNC 2024)**.

Chapter 6

REFERENCES

[1] D. Shamia, S. Prince and D. Bini, ”An Online Platform for Early Eye Disease Detection using Deep Convolutional Neural Networks,” 2022 6th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2022, pp. 388-392, doi: 10.1109/ICDCS54290.2022.9780765.

[2] G. Ramanathan, D. Chakrabarti, A. Patil, S. Rishipathak and S. Kharche, ”Eye Disease Detection Using Machine Learning,” 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-5, doi: 10.1109/GCAT52182.2021.9587740.

[3] D. Helen and S. Gokila, ”EYENET: An Eye Disease Detection System using Convolutional Neural Network,” 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 839-842, doi: 10.1109/ICECAA58104.2023.10212139.

[4] G. V. Datta, S. R. Kishan, A. Kartik, G. B. Sai and S. Gowtham, ”Glaucoma Disease Detection Using Deep Learning,” 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2023, pp. 1-6, doi: 10.1109/ICECCT56650.2023.10179802.

[5] K. Prasad, P. S. Sajith, M. Neema, L. Madhu and P. N. Priya, ”Multiple eye disease detection using Deep Neural Network,” TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 2148-2153, doi: 10.1109/TENCON.2019.8929666.

[6] K. Vayadande, V. Ingale, V. Verma, A. Yeole, S. Zawar and Z. Jamadar, ”Ocular Disease Recognition using Deep Learning,” 2022 International Conference on Signal and Information Processing (IConSIP), Pune, India, 2022, pp. 1-7, doi: 10.1109/ICoNSIP49665.2022.10007470.

[7] S. A. Toki, S. Rahman, S. M. Billah Fahim, A. Al Mostakim and M. K. Rhaman, ”RetinalNet-500: A newly developed CNN Model for Eye Disease Detection,” 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2022, pp. 459-463, doi: 10.1109/MIUCC55081.2022.9781785.

[8] Atul and S. Dhingra, ”Classification of Diabetic Retinopathy Disease with Improved Transfer Learning Techniques using EfficientNets,” 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-5, doi: 10.1109/ICRITO56286.2022.9965087

[9] Amin, Javeria & Sharif, Muhammad & Yasmin, Mussarat. (2016). A Review on Recent Developments for Detection of Diabetic Retinopathy. Scientifica. 2016. 1-20. 10.1155/2016/6838976.

[10] R. S. Salvi, S. R. Labhsetwar, P. A. Kolte, V. S. Venkatesh and A. M. Baretto, ”Predictive Analysis of Diabetic Retinopathy with Transfer Learning,” 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), NaviMumbai, India, 2021, pp. 1-6, doi: 10.1109/ICNTE51185.2021.9487789

[11] Roy, Amit & Abdullah, Riasat & Ahmed, Fahim & Mashfi, Md.Shahriar & Khan, Sazid & Karim, Dewan. (2023). RetNet: Retinal Disease Detection using Convolutional Neural Network. 1-6. 10.1109/ECCE57851.2023.10101661

[12] W. Liu, Z. Zhao and M. Levkiv, ”Automatic Diagnosis of Diabetic Retinopathy Based on EfficientNet,” 2023 17th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Jaroslaw, Poland, 2023, pp. 64-67, doi: 10.1109/CADSM58174.2023.10076511

[13] Lazuardi, Rachmadio & Abiwinanda, Nyoman & Suryawan, Tafwida & Hanif, Muhammad & Handayani, Astri. (2020). Automatic Diabetic Retinopathy Classification with EfficientNet. 756-760. 10.1109/TENCON50793.2020.9293941.

[14] D. R. Raman, S. Nishanthi and P. Babysha, "Diagnosis of Diabetic Retinopathy by using EfficientNet-B7 CNN Architecture in Deep Learning," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 430-435, doi:10.1109/ICSCSS57650.2023.10169453.

[15] Prasher, Shikha & Nelson, Leema & Sun, Gomathi. (2023). Automated Eye Disease Classification using MobileNetV3 and EfficientNetB0 Models using Transfer Learning. 1-5. 10.1109/WCONF58270.2023.10235193.

[16] Babaqi, Tareq & Jaradat, Manar & Yildirim, Ayse & Al-Nimer, Saif & Won, Daehan. (2023). Eye Disease Classification Using Deep Learning Techniques. 10.48550/arXiv.2307.10501.

[17] G. V. Doddi, "eye diseases classification," 2022. [Online]. Available: `https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification`.