OCULAR DISEASE RECOGNITION IN DEEP LEARNING

A PROJECT REPORT

Submitted by

MADHUBALAN S	**DSUG20104088**
MANO A	**DSUG20104093**
RAKESH J	**DSUG20104124**
RAMKUMAR R	**DSUG20104125**

in partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

PERAMBALUR-621212

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2024

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

PERAMBALUR-621212

BONAFIDE CERTIFICATE

Certified that this project report **" OCULAR DISEASE RECOGNITION IN DEEP LEARNING "** is the bonafide work of **MADHUBALAN S** (DSUG20104088), **MANO A** (DSUG20104093), **RAKESH J** (DSUG20104124), **RAMKUMAR R** (DSUG20104125), who carried out the project work under my supervision.

SIGNATURE
Dr.R.Gopi, M.Tech., PhD.,
HEAD OF THE DEPARTMENT,
Department of Computer Science & Engineering,
Dhanalakshmi Srinivasan Engineering College(Autonomous),
Perambalur - 621 212.

SIGNATURE
Mr.V.Gokulakrishnan, M.E.,MBA.,(phD)
ASSISTANT PROFESSOR,
Department of Computer Science & Engineering,
Dhanalakshmi Srinivasan Engineering College(Autonomous),
Perambalur - 621 212.

Submitted for Main-Project Viva-Voce Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our gratitude and thanks to **Our Parents** first for giving health and a sound mind for completing this project. We give all the glory and thanks to our almighty **GOD** for showering upon the necessary wisdom and grace for accomplishing this project.

It is our pleasant duty to express a deep sense of gratitude to our honorable Chancellor **Shri. A. Srinivasan**, for his kind encouragement. We have unique pleasure in thanking our Principal **Prof. Dr. D. Shanmugasundaram, M.E., PhD., F.I.E., C.Eng.,** our Dean **Dr. K. Anbarasan**, **M.E., Ph.D.,** and our COE **Dr. K. Velmurugan, M.E., Ph.D.,** for their unflinching devotion, which leads us to complete this project.

We express our faithful and sincere gratitude to our Head of the Department **Dr. R. Gopi, M.Tech., PhD.,** for his valuable guidance and support that he gave us during the project time.

We express our faithful and sincere gratitude to our main Project Coordinator **Ms.M.Hemalatha, M.E.,** of the Department of Computer Science and Engineering for giving us support throughout our project.

We are also thankful to our internal project guide **Mr.V.Gokulakrishnan, M.E.,MBA.,(PhD).,**of the Department of Computer Science and Engineering for his valuable guidance and precious suggestion to complete this project work successfully.

We render our thanks to all **Faculty members** and **Programmers** of the Department of **Computer Science and Engineering** for their timely assistance.

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision and Mission of the Department:

Vision

To produce globally competent, socially responsible professionals in the field of Computer Science and Engineering.

Mission

M1: Impart high quality experiential learning to get expertise in modern software tools

M2: Inculcate industry exposure and build interdisciplinary research skills.

M3: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M4: Acquire Innovative skills and promote lifelong learning with a sense of societal and ethical responsibilities

Program Educational Objectives (PEOs)

PEO 1: Graduates of the programme will develop proficiency in identifying, formulating, and resolving complex computing problems.

PEO 2: Graduates of the programme will achieve successful careers in the field of computer science and engineering, pursue advanced degrees, or demonstrate entrepreneurial success.

PEO 3: Graduates of the programme will cultivate effective communication skills, teamwork abilities, ethical values, and leadership qualities for professional engagement in industry and research organizations.

ABSTRACT

There has always been a problem for clinicians to identify eye diseases using fundus images early enough. Ocular disease diagnosis by hand is an expensive task, prone to errors, and difficult. Automated computer systems for detecting ocular diseases are needed to identify different eye illnesses utilizing fundus pictures. This project proposes a deep learning-based technique for focussed ocular detection, Such a system is now trouble-free because of deep learning algorithms that have enhanced picture categorization capabilities. For this, we classified the Ocular Disease intelligent Recognition (ODIR) dataset which includes 6000 photos of eight distinct fundus classes, using VGG-19, ResNet50 image classification algorithms Different ocular problems are characterized by these classifications. However, the dataset for these classes is somewhat erratic. This project recommended using the same amount of photos for both categories and creating a binary classification challenge out of this multiclass classification problem to address that difficulty This project also met the latest vision transformer method The models are trained with actual data well as wish insurges after applying (LBP) Local Binary Partem on the image.The absence of specialists in basic health units has resulted in a lack of accurate diagnosis of systemic or asymptomatic eye diseases, increasing the cases of blindness. In this context, the present paper proposes an ensemble of convolutional neural networks, which were submitted to a transfer learning process by using 38,727 high-quality fundus images. Next, the ensemble was tested with 13,000 low-quality fundus images acquired by low-cost equipment.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	**V**
	LIST OF FIGURES	**X**
	LIST OF ABBREVIATIONS	**XI**
1	**INTRODUCTION**	**1**
	1.1 DOMAIN DESCRIPTION	1
	1.2 WORKING PROCESS	2
2	**LITERATURE SURVEY**	**3**
	2.1 DEEP TRANSFER LEARNING STRATEGY TO DIAGNOSE EYE RELATED CONDITIONS AND DISEASES: AN APPROACH BASED ON LOW QUALITY FUNDUS IMAGES.	3
	2.2 A NOVEL MEIBOMIAN GLAND MORPHOLOGY ANALYTIC SYSTEM BASED ON A CONVOLUTIONAL NEURAL NETWORK	4
	2.3 A NOVEL SYSTEM FOR OCULAR SURFACE TEMPERATURE MEASUREMENT AND TRACKING	5
	2.4 AN ENGINEERING PLATFORM FOR CLINICAL APPLICATION OF OPTOGENETIC THERAPY IN RETINAL DEGENERATIVE DISEASES	5
	2.5 AN OPTIMAL VISUAL FATIGUE RELIEF METHOD FOR WORKERS CONSIDERING REST TIME ALLOCATION	6

	2.6	AUTOMATIC SEGMENTATION AND INTUITIVE VISUALISATION OF THE EPIRETINAL MEMBRANE IN 3D OCT IMAGES USING DEEP CONVOLUTIONAL APPROACHES	6
	2.7	AUTOMATIC SEGMENTATION OF RETINAL LAYERS IN MULTIPLE NEURODEGENERATIVE DISORDER SCENARIOS	7
	2.8	AUTONOMOUS STABILIZATION OF RETINAL VIDEOS FOR STREAMLINING ASSESSMENT OF SPONTANEOUS VENOUS PULSATIONS.	7
	2.9	CLASSIFICATION OF COMPANION ANIMALS' OCULAR DISEASES: DOMAIN ADVERSARIAL LEARNING FOR IMBALANCED DATA	8
	2.10	CONTINUOUS AND DISCRETE VOLTERRALAGUERRE MODELS WITH DELAY FOR MODELING OF SMOOTH PURSUIT EYE MOVEMENTS	8
3	**SYSTEM ANALYSIS**		**9**
	3.1 EXISTING SYSTEM		9
		3.1.1. Google's DeepMind Health	9
		3.1.2. IDx - DR	9
		3.1.3. Eye Art AI	9
		3.1.4. VERILY's retinal imaging technology	9
		3.1.5. Microsoft Inner Eye	9
	3.2 PROPOSED SYSTEM		10
		3.2.1. Introduction	10
		3.2.2. Objectives	10
		3.2.3. System Pattern Architecture	12
		3.2.4. Applications	13

	3.2.5. User Experience	13
	3.2.6. Security and Privacy	14
	3.2.7. Evaluation and Testing	14
	3.2.8. Future Directions	15
	3.2.9. Conclusion	15
	3.3 SCOPE OF THE PROJECT	15
	3.4 SYSTEM ARCHITECTURE	16
4	**REQUIREMENTS**	**18**
	4.1 HARDWARE REQUIREMENT	18
	4.2 SOFTWARE REQUIREMENT	18
5	**MODULE & LIBRARY DESCRIPTION**	**19**
	5.1 MODULE DESCRIPTION	**19**
	5.1.1 Data Preprocessing	19
	5.1.2 Feature Extraction	19
	5.1.3 Deep Learning Architecture	19
	5.1.4 Training	20
	5.1.5 Evaluation	20
	5.2 LIBRARY DESCRIPTION	**20**
	5.2.1 NumPy library	20
	5.2.2 TensorFlow Library	20
	5.2.3 Pandas Library	20
	5.2.4 Keras Library	21
	5.3 INSTALLATION	**21**
	5.4 INSTALLING MODULE	**21**
6	**BUILDING THE API/GUI**	**22**
	6.1 IMPORT LIBRARIES NEEDED IN THE PROJECT	23
7	**SYSTEM TESTING**	**24**
	7.1 SYSTEM TESTING	24
	7.2 TESTING TECHNIQUES	24
	7.2.1. Unit Testing	24
	7.2.2. Integration Testing	24
	7.2.3. End-to-End Testing	24

7.2.4 Performance Testing 25
7.3 SOFTWARE TESTING STRATEGIES 25
7.3.1. Requirement Analysis 25
7.3.2. Test Planning 25
7.3.3. Test Environment Setup 25
7.3.4. Functional Testing 25
8 CONCLUSION AND FUTURE ENHANCEMENT 26
8.1 CONCLUSION 26
8.2 FUTURE ENHANCEMENT 26
8.2.1 User-Friendly Interface 26
8.2.2 Diagnostic Results 27
8.2.3 Visualization Tools 27
8.2.4 Model Performance Feedback 27
8.2.5 Security and Privacy Controls 27
9 APPENDIX 28
APPENDIX I SOURCE CODE 28
APPENDIX II SCREENSHOTS 40
REFERENCES 44

LIST OF FIGURES

FIGURE.NO	FIGURE NAME	PAGE NO
3.2	VGG-19 & ResNET 50 Architecture	11
3.4.1	Actual Image & Greyscale Image & LBP Image	17
3.4.2	System Architecture Of LBP Pattern	17
9.1	Patient details	40
9.2	Testing without LBP	40
9.3	Prediction Label	41
9.4	Graph without LBP	41
9.5	Test Prediction with LBP	42
9.6	Graph with LBP	42
9.7	Pixelated Eye Using LBP	43
9.8	Output Data	43
9.9	Final Result	43

LIST OF ABBREVIATIONS

ODR	Ocular Disease Recognition
CNN	Convolutional Neural Network
IDE	Integrated Development Environment
API	Application Programming Interface
GUI	Graphical User Interface
WHO	World Health Organization
ODIR	Ocular Disease intelligent Recognition
LBP	Local Binary Pattern
MGD	Meibomian glands dysfunction
OST	Ocular surface temperature
ERM	Epiretinal Membrane
OCT	Optical Coherence Tomography
ND	Neurological Disorders
SVP	Spontaneous Venous Pulsations
VL	Volterra-Laguerre
FFA	Fundus Fluorescein Angiography
PBFT	Practical Byzantine Fault Tolerance

CHAPTER 1
INTRODUCTION

1.1. Domain Description: Deep Learning

Deep learning is a subfield of machine learning that focuses on algorithms inspired by the structure and function of the human brain's neural networks. It involves training artificial neural networks with large datasets to learn hierarchical representations of data, leading to state-of-the-art performance in various tasks such as image and speech recognition, natural language processing, and robotics.

Applications:

Computer Vision: Deep learning models have achieved remarkable accuracy in tasks such as object detection, image classification, facial recognition, and medical imaging analysis.

Natural Language Processing (NLP): Deep learning techniques, including recurrent neural networks (RNNs) and transformers, power applications such as machine translation, sentiment analysis, text summarization, and chatbots.

Speech Recognition: Deep learning algorithms have enabled significant advancements in speech recognition systems, facilitating voice-controlled assistants, speech-to-text transcription, and speaker identification.

Healthcare: Deep learning plays a crucial role in medical image analysis, disease diagnosis, drug discovery, personalized medicine, and healthcare management.

Future Directions:

Continual Learning: Developing algorithms that can learn continuously from streaming data without forgetting previously learned knowledge.

Multimodal Learning: Integrating information from multiple modalities such as text, images, and audio to create more robust and versatile models.

Ethical and Fair AI: Addressing ethical concerns related to bias, fairness, privacy, and accountability in deep learning systems to ensure their responsible deployment.

1.2. Working Process: Ocular Disease Recognition (ODR)

Nearly 2.2 billion people around the world experience vision problems. The World Health Organization (WHO) estimates that there may have been a reduction in at least 1 billion of these incidents. Over time, there has been an increase in eye illnesses, with changes being one of the causes of a change in human behavior caused by technology and the creation of technological apparatus. That impact caused ocular diseases have had a significant impact on modern human life. Common eye diseases include glaucoma, diabetes, and hypertension.

The disease should be accurately identified at all times. and effectiveness. One of the most important human organs is the eye. primarily vision aids in the recognition and detection of 3D objects. loss of one eyesight or both eyesight may cause a person to live an unsettling way of life because people make decisions based on what they observe in their daily lives.

This study is primarily concerned with accurate recognition taking into account the Local Binary Pattern features (LBP) of ocular disease. It sights to create a variety of extracting features techniques and Neural Networks to distinguish typical visual disorders involving fundus photographs.

We discovered that eye illness identification using deep learning focuses on just one aberration in comparison to the prior methods. For various challenges, several designs were utilized in this paper, and the outcomes are fairly good. Thus, with accuracy more than 90% for each activity, illnesses

CHAPTER 2
LITERATURE SURVEY

Literature Survey on Ocular Disease Recognition in Deep Learning:

2.1 "DEEP TRANSFER LEARNING STRATEGY TO DIAGNOSE EYE RELATED CONDITIONS AND DISEASES: AN APPROACH BASED ON LOW QUALITY FUNDUS IMAGES"

Authors: GABRIEL D. A. ARANHA

ABSTRACT: Data from the World Health Organization indicate that billion cases of visual impairment could be avoided, mainly with regular examinations. However, the absence of specialists in basic health units has resulted in a lack of accurate diagnosis of systemic or asymptomatic eye diseases, increasing the cases of blindness. In this context, the present paper proposes an ensemble of convolutional neural networks, which were submitted to a transfer learning process by using 38,727 high-quality fundus images. This way, the proposed approach represents a novel deep transfer learning strategy, that is more suitable and feasible to be applied by public health systems of emerging and under-developing countries. From low-quality images, the proposed approach was able to reach accuracies of 87.4%, 90.8%, 87.5%, 79.1% to classify cataract, diabetic retinopathy, excavation and blood vessels, respectively.

MERITS:

- Transfer learning allows leveraging pre-trained deep learning models, which are trained on large datasets for general tasks like image recognition. This reduces the need for a massive dataset for training, which might be difficult to obtain for specific medical conditions.
- Enhanced worker health and productivity through targeted rest breaks tailored to individual visual fatigue levels.

DEMERITS:

- Pre-trained models might not generalize well to low-quality fundus images, especially if they were trained on high-quality datasets.
- Efficacy Challenges: While promising, the efficacy of optogenetics in restoring vision needs further validation, and the extent of vision restoration may vary among individuals.

2.2 "A NOVEL MEIBOMIAN GLAND MORPHOLOGY ANALYTIC SYSTEM BASED ON A CONVOLUTIONAL NEURAL NETWORK"

Author: MENGTING LIU

ABSTRACT : Meibomian glands dysfunction (MGD) is the main cause of dry eyes. Biological parameters of meibomian gland (MG) such as height, tortuosity and the degree of atrophy are closely related to its function . However Thus an effective quantitative diagnostic tool is needed for clinical diagnosis. Automatic quantification of MGs' morphological features could be a challenging task and play an important role in MGD diagnosis and classification. We proposed a novel MGs extraction method based on convolutional neural network (CNN) with enhanced mini U-Net. A prospective study was conducted, 120 subjects were included and taken meibography. The training and validation sets encompassed 60 subjects; and the test set consisted of other 60 subjects with comprehensive examinations for ocular surface disease index questionnaire

Merits:

●The AI system automates the process of MG morphology assessment, significantly reducing the time and effort required for analysis.

● Compared to manual methods, which can be labor-intensive and time-consuming, the AI system enables rapid analysis of meibography images, allowing clinicians to efficiently evaluate MG morphology and make timely treatment decisions. This efficiency can improve workflow in busy clinical settings and enhance patient throughput.

Demerits:

●The high accuracy in extracting MGs from meibography images under controlled conditions. However, the performance of the AI system in real-world clinical settings with varying image quality, lighting conditions, and patient factors (e.g., eye movements, blinking artifacts) may be less robust.

● Variability in image quality and other confounding factors could affect the accuracy and consistency of MG morphology quantification, leading to potential diagnostic errors or inconsistencies.

2.3 "A NOVEL SYSTEM FOR OCULAR SURFACE TEMPERATURE MEASUREMENT AND TRACKING"

Authors: ALEXANDER WONG

ABSTRACT : Ocular surface temperature (OST) is affected by changes in eye physiology caused by normal homeostasis, environmental changes, or systemic and local disease. OST can help a physician diagnose eye disease with improved accuracy and provide useful information for eye research. This paper presents a novel system, including novel hardware design and novel algorithms, capable of automatically measuring and tracking OST from the cornea over any period of time

Merits:

The novel system provides comprehensive imaging and analysis, improving accuracy in Ocular Surface Temperature (OST) measurement and aiding in diagnosing eye diseases more accurately.

Demerits:

Potential drawbacks may include factors such as cost, complexity, and potential limitations in real-world clinical settings, which need to be carefully considered.

2.4."AN ENGINEERING PLATFORM FOR CLINICAL APPLICATION OF OPTOGENETIC THERAPY IN RETINAL DEGENERATIVE DISEASES "

Authors: BOYUAN YAN

ABSTRACT: Optogenetics is a new approach for controlling neural circuits with numerous applications in both basic and clinical science. In retinal degenerative diseases, the photoreceptors die, but inner retinal cells remain largely intact. By expressing light sensitive proteins in the remaining cells, optogenetics has the potential to offer a novel approach to restoring vision. In the past several years, optogenetics has advanced into an early clinical stage, and promising results have been reported.

Merits:

Precision: Optogenetics allows for precise control of neural activity, enabling targeted stimulation of specific retinal cells.

Demerits:

Safety Concerns: Introducing foreign proteins into the retina may pose safety risks, including inflammation or immune reactions.

2.5 "AN OPTIMAL VISUAL FATIGUE RELIEF METHOD FOR WORKERS CONSIDERING REST TIME ALLOCATION"

Author: XIAOHUI REN

ABSTRACT : Working under the visual display terminal for several hours may lead to serious visual fatigue. Productivity will be reduced and the health of workers will be damaged. To address the problem, an optimal visual fatigue relief method for workers considering rest time allocation has been proposed in this paper. First, an analytic model is established to depict the relationship between the visual fatigue change and continuous work time.

Merits:

Enhanced worker health and productivity through targeted rest breaks tailored to individual visual fatigue levels.

Demerits:

Potential disruption to workflow as workers adapt to new rest break schedules, impacting short-term productivity

2.6 "AUTOMATIC SEGMENTATION AND INTUITIVE VISUALISATION OF THE EPIRETINAL MEMBRANE IN 3D OCT IMAGES USING DEEP CONVOLUTIONAL APPROACHES"

Author: JOAQUIM DE MOURA

ABSTRACT : Epiretinal Membrane (ERM) is a disease caused by a thin layer of scar tissue that is formed on the surface of the retina. When this membrane appears over the macula, it can cause distorted or blurred vision. Although normally idiopathic, its presence can also be indicative of other pathologies such as diabetic macular edema or vitreous hemorrhage. ERM removal surgery can preserve more visual acuity the earlier it is performed.

Merits:

Early removal of Epiretinal Membrane (ERM) can help preserve visual acuity by preventing further distortion or blurriness in vision.

Demerits:

ERM removal surgery carries inherent risks, including infection, retinal detachment, and decreased visual acuity post-surgery.

2.7 "AUTOMATIC SEGMENTATION OF RETINAL LAYERS IN MULTIPLE NEURODEGENERATIVE DISORDER SCENARIOS"

Author: JORGE NOVO

Retinal Optical Coherence Tomography (OCT) allows the non-invasive direct observation of the central nervous system, enabling the measurement and extraction of biomarkers from neural tissue that can be helpful in the assessment of ocular, systemic and Neurological Disorders (ND). Deep learning models can be trained to segment the retinal layers for biomarker extraction. Experimental results demonstrate a more uniform distribution of fuzzy values and increased fairness in the voting phase of DPoS.

Merits:

Retinal OCT with deep learning segmentation offers precise biomarker extraction, aiding in comprehensive assessment of ocular, systemic, and neurological disorders.

Demerits:

Deep learning models for OCT segmentation require extensive annotated data and may struggle with poor image quality.

2.8 "AUTONOMOUS STABILIZATION OF RETINAL VIDEOS FOR STREAMLINING ASSESSMENT OF SPONTANEOUS VENOUS PULSATIONS "

Author: SAHAR SHARIFLOU

ABSTRACT: Spontaneous retinal Venous Pulsations (SVP) are rhythmic changes in the caliber of the central retinal vein and are observed in the optic disc region (ODR) of the retina. Its absence is a critical indicator of various ocular or neurological abnormalities. Recent advances in imaging technology have enabled the development of portable smartphone-based devices for observing the retina and assessment of SVPs.

Merits:

Smartphone-based retinal imaging enhances accessibility for observing Spontaneous retinal Venous Pulsations (SVPs), aiding in early detection of ocular and neurological abnormalities.

Demerits:

Smartphone-based retinal imaging's limitations in image quality and stability can impede accurate Spontaneous retinal Venous Pulsations (SVPs) observation

2.9 "CLASSIFICATION OF COMPANION ANIMALS' OCULAR DISEASES: DOMAIN ADVERSARIAL LEARNING FOR IMBALANCED DATA"

Author: Sachi Chaudhary.

ABSTRACT: In contrast to the widespread implementation of computer-aided diagnosis of human diseases, the limited availability of veterinary image datasets has hindered its application in animals. Additionally, while most medical imaging data are captured in clinical settings, such as optical coherence tomography and fundus photography, diagnosis based on digital camera or smartphone images can be more beneficial for pet owners.

Merits:

Domain adversarial learning effectively mitigates class imbalance in companion animals' ocular disease classification, improving diagnostic accuracy and treatment outcomes.

Demerits:

Domain adversarial learning's effectiveness heavily relies on data quality and diversity, which may not always be readily available or representative of all ocular diseases in companion animals.

2.10 "CONTINUOUS AND DISCRETE VOLTERRA-LAGUERRE MODELS WITH DELAY FOR MODELING OF SMOOTH PURSUIT EYE MOVEMENTS"

Author: Alexander Medvedev

ABSTRACT : The mathematical modeling of the human smooth pursuit system from eye-tracking data is considered. Recently developed algorithms for the estimation of Volterra-Laguerre (VL) models with explicit time delay are applied in continuous and discrete time formulations to experimental data collected from Parkinsonian patients in different medication states and healthy controls

Merits:

Continuous and discrete Volterra-Laguerre models with delay offer comprehensive frameworks for accurately modeling the dynamics of smooth pursuit eye movements, providing insights into neurological processes

Demerits:

Complexity in parameter estimation and model selection may arise, requiring sophisticated computational techniques and potentially limiting practical implementation.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

3.1.1. Google's DeepMind Health:

Description: DeepMind Health, a division of Google's DeepMind, developed a deep learning system for the detection of diabetic retinopathy and diabetic macular edema from retinal fundus images.Look for mentions of ResNet-50 or VGG19 in these studies.

3.1.2. IDx-DR:

Description: IDx-DR is an FDA-approved autonomous AI system for the detection of diabetic retinopathy. It uses deep learning algorithms to analyze retinal images and provide diagnostic recommendations without the need for human interpretation

3.1.3.EyeArt AI:

Description: EyeArt AI is another FDA-approved system for the autonomous detection of diabetic retinopathy. It utilizes deep learning technology to analyze retinal images and provide diagnostic assessments, helping to streamline the screening process for diabetic retinopathy.

3.1.4. VERILY's retinal imaging technology:

Description: Verily, an Alphabet company, has been developing advanced retinal imaging technology coupled with deep learning algorithms for the detection of various ocular diseases. Their system aims to provide early detection and intervention for conditions such as age-related macular degeneration and diabetic retinopathy.

3.1.5. Microsoft InnerEye:

Description: While not specifically focused on ocular diseases, Microsoft's InnerEye project utilizes deep learning for medical image analysis, including tasks such as tumor segmentation in MRI scans. The technology could potentially be adapted or extended to assist in ocular disease recognition.

3.2 PROPOSED SYSTEM

Title: "**OCULAR DISEASE RECOGNITION IN DEEP LEARNING (ODR)**"

3.2.1. Introduction

Local Binary Patterns are a texture descriptor that characterizes the local structure of an image by comparing each pixel with its neighboring pixels. It assigns a binary code to each pixel based on whether the intensity of the neighboring pixels is greater or lesser than the intensity of the center pixel. This creates a histogram of binary patterns, which captures the texture information of the image.

3.2.2. Objectives:

Research and Development:

Conduct comprehensive research on ocular diseases, LBP texture analysis, and deep learning methodologies to inform the design and development of the automated recognition system.

Algorithm Development:

Develop efficient algorithms for LBP feature extraction and integration with deep learning architectures, optimizing the feature representation for effective ocular disease recognition.

Evaluation and Validation:

Evaluate the performance of the developed system using standardized metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC), validating its effectiveness in recognizing various ocular diseases.

Pattern And Algorithm:

VGG-19: Advanced CNN-VGG19 has layers that have previously undergone training and has a solid understanding of the shape, colour, and structural aspects of a picture. For difficult classification tasks, the very deep VGG19 has been trained on an enormous variety of images.

ResNet50: Instead of attempting to learn specific characteristics, ResNet or Residual Network makes use of residual learning. Residual may be easily regarded as the feature learned from that layer's input subtracted. More than ResSNet50, there are other ResNet variations. Creating a shortcut connection that omits one or more levels is the fundamental concept behind ResNet.

Vision Transformer: The Vision Transformer, often known as ViT, is a classification system that employs a Transformer-like design in some portions of the image. By partitioning an image into fixed-size patches, linearly embedding each one, including location embeddings, and assembling the vectors, one may produce a series of vectors that can be fed into a standard Transformer encoder.

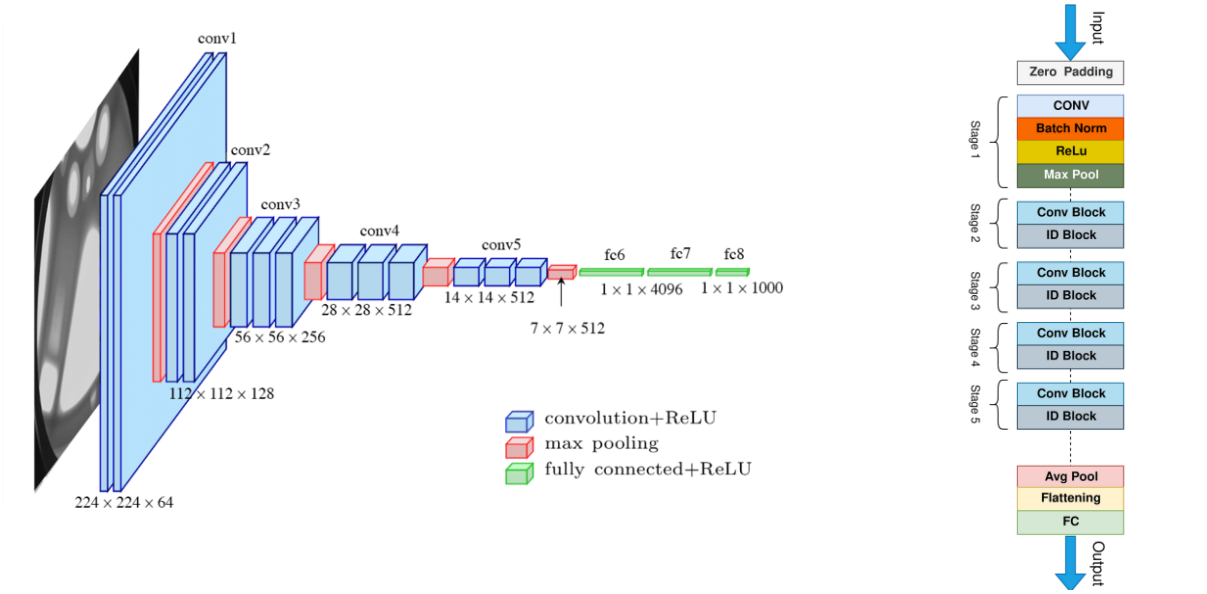

Fig : 3.2 VGG-19 & ResNET 50 Architecture

3.2.3. System Pattern Architecture:

Data Collection and Preprocessing:

Obtain a large dataset of ocular images containing various types of diseases and healthy samples. Preprocess the images to standardize their size, color, and quality. This step may also involve image enhancement techniques to improve the overall quality of the images.

Feature Extraction using Local Binary Patterns (LBP):

LBP is a texture descriptor that characterizes the local patterns in an image by comparing each pixel with its surrounding neighbors. This method is efficient for capturing texture information in images. Implement an LBP feature extraction module to extract texture features from the preprocessed ocular images. Each image is divided into regions, and LBP histograms are computed for each region.

Deep Learning Model Architecture:

Design a deep learning architecture to learn discriminative features from the extracted LBP histograms. A common approach is to use convolutional neural networks (CNNs) due to their ability to automatically learn hierarchical features from images.

Training Phase:

Split the preprocessed dataset into training, validation, and testing sets.Train the deep learning model using the training data. During training, the model learns to map the extracted LBP features to the corresponding ocular disease labels

Evaluation and Testing:

Evaluate the trained model's performance using the testing dataset to assess its accuracy, precision, recall, and other relevant metrics. Fine-tune the model if necessary based on the evaluation results.

Deployment:

Once the model achieves satisfactory performance, deploy it for real-world applications. Integrate the model into a user-friendly interface or healthcare system where

clinicians or patients can input ocular images for disease diagnosis. Continuously monitor the model's performance in the deployed environment and update it as needed.

3.2.4. Applications:

Early Disease Detection:

By analyzing ocular images using deep learning models trained on LBP features, healthcare professionals can detect ocular diseases at an early stage. Early detection is crucial for timely intervention and treatment, which can significantly improve patient outcomes.

Automated Screening:

The developed system can be integrated into healthcare systems to automate the screening process for ocular diseases. This automation reduces the burden on healthcare professionals, allowing them to focus on more critical tasks while ensuring timely diagnosis for patients.

Telemedicine:

In remote or underserved areas where access to specialized healthcare services is limited, telemedicine platforms equipped with ocular disease recognition systems can facilitate remote diagnosis and monitoring of ocular conditions. Patients can upload ocular images for analysis, and healthcare professionals can provide remote consultations based on the results.

3.2.5. User Experience:

Healthcare Professionals:

Healthcare professionals may appreciate the efficiency and accuracy provided by the automated screening system. It can assist them in making timely diagnoses, especially in busy clinical settings where time is limited.

Patients:

Positive Experiences: Patients may benefit from the convenience and accessibility of remote diagnosis through telemedicine platforms equipped with ocular disease recognition

systems. It can save them time and effort associated with visiting healthcare facilities physically.

Challenges: Patients may have concerns about the accuracy and reliability of the automated diagnosis, particularly if they receive conflicting results from different sources.

3.2.6. Security and Privacy:

Data Security:

Encrypt ocular images and other sensitive data during transmission and storage to prevent unauthorized access.Implement secure authentication mechanisms to control access to the system and ensure that only authorized users can upload or access patient data.

Privacy Protection:

Obtain informed consent from patients before collecting and using their ocular images for analysis. Clearly communicate the purpose of data collection, how the data will be used, and any potential risks involved Adhere to data protection regulations such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States or the General Data Protection Regulation (GDPR) in the European Union to ensure compliance with legal requirements for handling medical data.

3.2.7. Evaluation and Testing:

Data Evaluation:

Assess the quality and representativeness of the ocular image dataset used for training and testing the deep learning model. Ensure that the dataset covers a diverse range of ocular diseases, severity levels, and demographic characteristics to improve the model's generalization capability.

Model Evaluation:

Split the dataset into training, validation, and testing subsets. The training set is used to train the model, the validation set is used to tune hyperparameters and monitor performance during training, and the testing set is used to evaluate the final model.

Cross-Validation Testing:

Perform k-fold cross-validation to assess the robustness and generalization ability of the model. This involves splitting the dataset into k subsets, training the model on k-1 subsets, and evaluating its performance on the remaining subset, repeating this process k times.

3.2.8. Future Directions:

Incorporate additional modalities such as optical coherence tomography (OCT) scans, fundus fluorescein angiography (FFA), and patient medical records to provide a comprehensive view of ocular health.

Develop hybrid deep learning models that can fuse information from multiple modalities to enhance disease detection and characterization.

Develop secure and anonymized data sharing frameworks for pooling ocular image datasets from multiple sources for improved model training and validation.

3.2.9. Conclusion:

In conclusion, ocular disease recognition in deep learning using Local Binary Patterns (LBP) presents a promising avenue for advancing the diagnosis and management of ocular conditions. By leveraging deep learning techniques, such as convolutional neural networks (CNNs), in conjunction with texture descriptors like LBP, researchers and developers can extract meaningful features from ocular images and build highly accurate and efficient disease recognition systems

3.3 SCOPE OF THE PROJECT

A ocular disease recognition in deep learning using Local Binary Patterns (LBP) encompasses a wide range of research, development, and application areas within the field of ophthalmology and medical imaging. Here's a breakdown of the scope

Research and Development:

Investigating novel deep learning architectures and algorithms for ocular disease recognition, with a focus on incorporating LBP features for texture analysis.

Exploring advanced techniques such as attention mechanisms, transfer learning, and uncertainty estimation to improve model performance and robustness.

Data Collection and Curation:

Curating diverse and well-annotated datasets of ocular images encompassing various diseases, severity levels, and demographic characteristics.

Addressing challenges related to data imbalance, bias, and variability to ensure the representativeness and generalizability of the datasets.

Algorithm Optimization and Validation:

Optimizing deep learning algorithms and model architectures to achieve high accuracy, efficiency, and scalability for ocular disease recognition tasks.

Conducting rigorous validation studies, including cross-validation and external validation, to assess model performance across different datasets and clinical settings.

Education and Training:

Developing educational resources and training programs to familiarize healthcare professionals with the principles and applications of deep learning in ocular disease diagnosis.

Providing continuous professional development opportunities to update clinicians on the latest advancements in ocular imaging and diagnostic technologies.

3.4 SYSTEM ARCHITECTURE

The system architecture of a ocular disease recognition in deep learning Obtain a dataset of ocular images containing both healthy and diseased samples. This dataset should cover a variety of ocular conditions and imaging modalities.Apply Local Binary Patterns (LBP) to extract texture features from the preprocessed ocular images.LBP is computed for each pixel in the image by comparing its intensity with neighboring pixels and encoding the comparisons into binary patterns. Design a deep learning architecture to learn discriminative features from the extracted LBP histograms. Split the dataset into training, validation, and testing sets. Train the deep learning model using the training data. During training, the model learns to map the LBP features to the corresponding ocular disease labels. Split the dataset into training, validation, and testing sets.Train the deep learning model using the training data. During training, the model learns to map the LBP features to the corresponding ocular disease labels. Deploy the trained model for real-world applications, such as clinical diagnosis or screening programs.

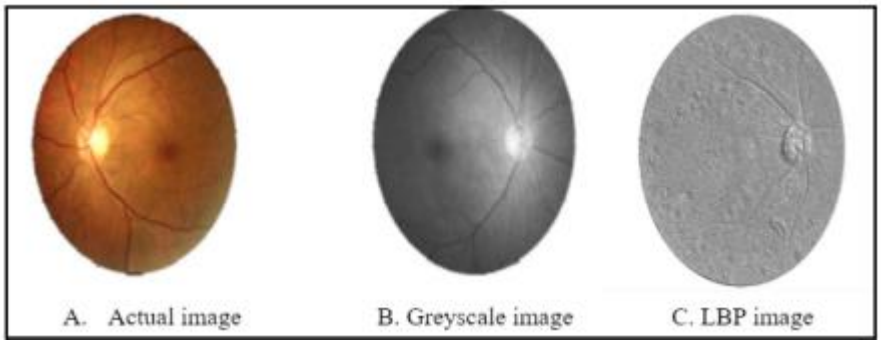

Fig : 3.4.1 Actual Image & Greyscale Image & LBP Image

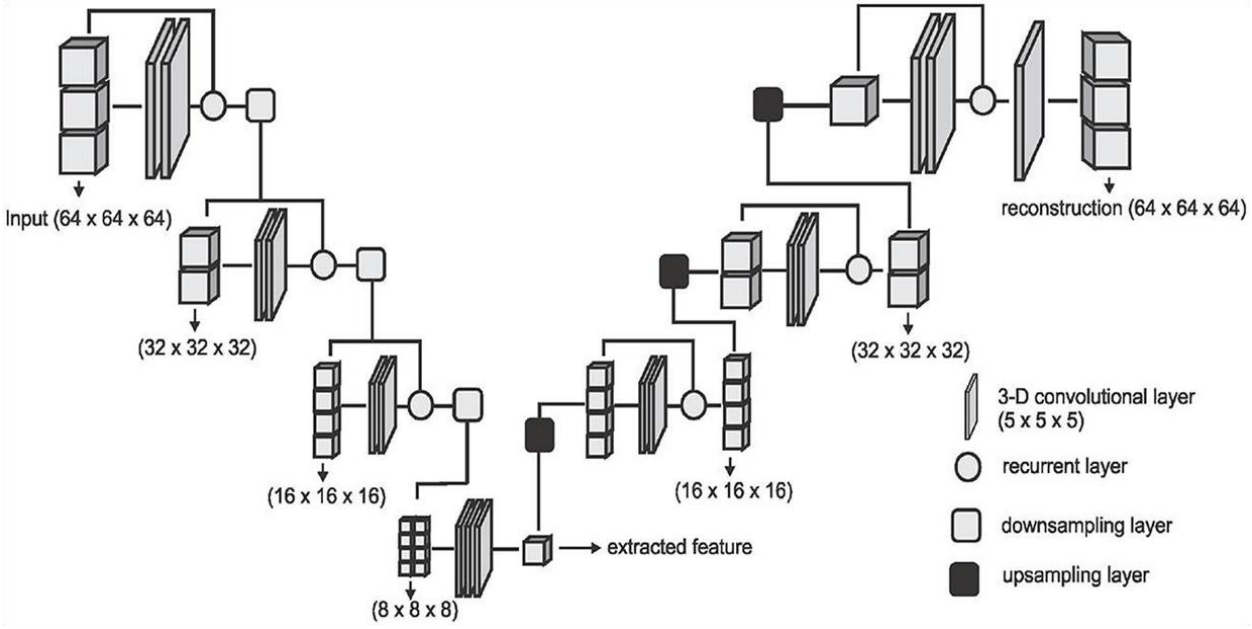

Fig : 3.4.2 System Architecture Of LBP Pattern

CHAPTER 4
REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Processor : Dual core processor or Higher
- Ram : 4 GB DDR4 or Higher
- Graphics : Intel Integrated Graphics or Higher

4.2 SOFTWARE REQUIREMENTS

- Programming Language : Python
- Software Libraries : NumPy, Pandas & Keras
- IDE : Jupyter Notebook
- Environment : Kaggle

CHAPTER 5
MODULE DESCRIPTION & LIBRARY DESCRIPTION

5.1 MODULE DESCRIPTION:

The ocular disease recognition system utilizes a Local Binary Patterns (LBP) module to extract crucial texture features from preprocessed ocular images. By comparing pixel intensities with neighboring pixels, the LBP algorithm encodes local texture patterns, providing a representative feature set for each image.

5.1.1 Data Preprocessing:

Description: This module preprocesses raw ocular images to enhance their quality and standardize them for further processing.

Functionality:

Resize images to a standardized resolution.

Normalize pixel intensities to a common scale.

5.1.2 Feature Extraction:

Description: This module extracts relevant features from preprocessed ocular images to capture important characteristics for disease recognition.

Functionality:

Compute Local Binary Patterns (LBP) for texture feature extraction.

Generate LBP histograms or descriptors for each image.

5.1.3 Deep Learning Architecture:

Description: This module designs the architecture of a deep neural network, specifically tailored for ocular disease recognition.

Functionality:

Define the architecture of a convolutional neural network (CNN).

Specify the number and type of layers, including convolutional, pooling, and fully connected layers.

5.1.4 Training:

Description: This module involves training the deep learning model using the preprocessed images and extracted features.

Functionality:

Split the dataset into training and validation sets.

Initialize model parameters and define the loss function.

5.1.5 Evaluation:

Description: This module evaluates the performance of the trained model using appropriate metrics to assess its accuracy and effectiveness in disease recognition.

Functionality:

Evaluate the trained model on a separate test dataset.

Calculate performance metrics such as accuracy, precision, recall, and F1-score.

5.2 LIBRARY DESCRIPTION:

5.2.1 NumPy library:

NumPy arrays are used to represent and manipulate data, including images of ocular diseases. Images can be loaded into NumPy arrays, and various operations can be performed on these arrays to preprocess the data before feeding it into deep learning models

5.2.2 TensorFlow Library:

TensorFlow serves as the backbone for implementing deep learning architectures for ocular disease recognition tasks. It offers high-level APIs (such as Keras) and low-level APIs for flexible model construction and training.

.5.2.3 Pandas Library:

Pandas is a powerful data manipulation and analysis library built on top of NumPy. It offers data structures like DataFrame and Series, along with functionalities for data cleaning, exploration, and transformation.

5.2.4 Keras Library:

Keras is a high-level neural networks API written in Python and designed to be user-friendly, modular, and extensible. It provides a simple interface for building and training deep learning models.

5.3 INSTALLATION

Installing Jupyter Notebook runs on Python, so you need to have Python installed on your system. You can download and install Python

Jupyter Notebook:

Install Jupyter Notebook: Once Python is installed, open your command-line interface (CLI) or terminal and run the following command to install Jupyter Notebook using

pip install notebook

Install Kaggle:

Once Python and pip are installed, open your command-line interface (CLI) or terminal and run the following command to install the Kaggle and run the following command

pip install kaggle

5.4 INSTALLING MODULE

To install a module for ocular disease recognition using deep learning, you typically need to specify which deep learning framework you'll be using (e.g., TensorFlow) and then install the required packages accordingly.

pip install tensorflow

pip install tensorflow-gpu

pip install scikit-learn opencv-python

CHAPTER 6
BUILDING THE API/GUI

Deep Learning Model Development:

Develop a deep learning model for ocular disease recognition using LBP features. Train the model using a suitable dataset containing images of ocular diseases.Evaluate the trained model to ensure it achieves satisfactory performance metrics.

API Development:

A framework for building the API. Popular choices include Flask and Fast API. Define endpoints for the API to handle various tasks such as uploading images, making predictions, and retrieving results. Implement the logic for preprocessing images, extracting LBP features, and making predictions using the trained deep learning model. Ensure error handling and validation of input data to maintain the robustness of the API. Secure the API by implementing authentication and authorization mechanisms if needed.

GUI Development:

A GUI framework for building the user interface. Options include Tkinter, PyQt, and web-based frameworks like Flask .Design the GUI layout with appropriate widgets for uploading images, displaying results, and providing user feedback. Integrate the API endpoints. Implement functionality for uploading images, sending requests to the API, and displaying predictions returned by the API.Enhance the user experience with features like real-time updates, progress indicators, and error messages.

Testing and Debugging:

Test the API and GUI components independently to ensure they function correctly. Perform integration testing to verify that the API and GUI work together seamlessly. Debug any issues that arise during testing and make necessary adjustments to the code.

Maintenance and Updates:

Monitor the performance of the API and GUI in production and address any issues promptly.Update the deep learning model periodically to incorporate new data and improve accuracy.Gather feedback from users and iterate on the application to add new features and enhance usability.

6.1 IMPORT LIBRARIES NEEDED IN THE PROJECT

To perform ocular disease recognition using Local Binary Patterns (LBP) in deep learning, you'll need several libraries for image processing, deep learning model development, and building the API and GUI.

```
import cv2

import numpy as np

import tensorflow as tf

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report

from flask import Flask, request, jsonify

from fastapi import FastAPI, UploadFile, File

from PIL import Image, ImageTk
```

CHAPTER 7
SYSTEM TESTING

7.1 SYSTEM TESTING

Testing the security aspects of Verify that the system can accept input in the form of ocular images or LBP histograms. Test the system with a variety of input images, including different types of ocular diseases, varying image qualities, and different resolutions. Test the LBP feature extraction module to ensure it accurately computes LBP histograms from input ocular images. Verify that the extracted features capture relevant texture information and are consistent across multiple runs. Measure the inference speed and computational resource requirements of the system to ensure it meets performance targets. Test the system's scalability to handle large volumes of ocular images efficiently, particularly in scenarios with concurrent user access or high throughput requirements.

7.2 TESTING TECHNIQUES

7.2.1. Unit Testing:

Test individual components of the system, such as the LBP feature extraction module and the deep learning model, in isolation.Verify that each component produces the expected output for a variety of input images, including both healthy and diseased ocular images.

7.2.2. Integration Testing:

Test the integration of different modules and components within the system, such as the integration between the LBP feature extraction module and the deep learning model.Verify that data flows correctly between modules and that the system as a whole behaves as expected.

7.2.3. End-to-End Testing:

Perform end-to-end testing to evaluate the entire workflow of the system, from inputting ocular images to generating disease predictions.Test the system with a variety of input images representing different ocular diseases and assess the accuracy of the resulting predictions.

7.2.4. Performance Testing:

Measure the inference speed of the system to ensure it meets performance requirements for real-time or near-real-time applications.Assess the computational resource requirements, such as memory and processing power, to ensure scalability and efficiency.

7.3 SOFTWARE TESTING STRATEGIES

To ensure the robustness, reliability, and effectiveness of the ocular disease recognition system in deep learning using Local Binary Patterns (LBP), it's essential to devise a comprehensive software testing strategy. Here's a tailored approach for testing the program:

7.3.1. Requirement Analysis:

Understand the functional and non-functional requirements of the ocular disease recognition system, including accuracy, speed, scalability, and user interface.Define clear acceptance criteria to evaluate whether the system meets these requirements.

7.3.2. Test Planning:

Develop a test plan outlining the objectives, scope, resources, and schedule for testing.Identify testing techniques, tools, and environments required for each phase of testing.

7.3.3. Test Environment Setup:

Set up the testing environment with appropriate hardware, software, and datasets representative of real-world scenarios.Ensure compatibility with deep learning frameworks, image processing libraries, and other dependencies.

7.3.4. Functional Testing:

Verify that the system accurately identifies and classifies ocular diseases based on input images.Test the system's adherence to specified performance metrics, such as accuracy, sensitivity, specificity, and speed

CHAPTER 8
CONCLUSION & FUTURE ENHANCEMENT

8.1 CONCLUSION:

In conclusion, ocular disease recognition using Local Binary Patterns (LBP) in deep learning holds significant promise for revolutionizing the diagnosis and management of various ocular conditions. By leveraging deep learning techniques and texture analysis methods like LBP, researchers and developers have the opportunity to create accurate, efficient, and accessible diagnostic tools for improving eye health outcomes.

Through the integration of LBP feature extraction with deep learning models, the system can effectively capture intricate texture patterns in ocular images, enabling precise disease identification and classification. Moreover, the system's potential extends beyond mere recognition to include early detection, prognosis prediction, and personalized treatment recommendations, thereby facilitating proactive and tailored care for patients.

Integration of additional imaging modalities, such as optical coherence tomography (OCT) scans or fundus fluorescein angiography (FFA), to provide a more comprehensive assessment of ocular health and disease progression

8.2 FUTURE ENHANCEMENT:

8.2.1 User-Friendly Interface:

Dashboard Overview : Upon logging in, users are greeted with a dashboard providing an overview of system status, recent activities, and notifications.Key metrics such as system accuracy, performance trends, and dataset statistics are displayed prominently.

Image Upload and Processing : Users can easily upload ocular images through a simple drag-and-drop interface or by selecting files from their computer or connected devices.As images are uploaded, the system provides real-time feedback on processing status, including image preprocessing and feature extraction using LBP.

8.2.2 Diagnostic Results:

Once processing is complete, the interface presents diagnostic results in a clear and organized manner.Each uploaded image is accompanied by its corresponding disease prediction(s) and associated confidence scores.

8.2.3 Visualization Tools:

Interactive visualization tools allow users to explore ocular images and diagnostic outputs in greater detail.Features like zoom, pan, and contrast adjustment enable closer examination of image regions and abnormalities.

8.2.4 Model Performance Feedback:

Users have the option to provide feedback on the accuracy and relevance of diagnostic results, helping to improve model performance over time.Feedback mechanisms may include rating systems, comment fields, or structured surveys.

8.2.5 Security and Privacy Controls:

Robust security features, such as user authentication, role-based access control, and data encryption, safeguard sensitive information and ensure compliance with privacy regulations.Users have control over their data, with options to delete uploaded images and revoke access to shared diagnostic results

8.2.6 Feedback Mechanisms:

Built-in feedback mechanisms solicit user input on interface usability, feature requests, and overall satisfaction, enabling continuous improvement and refinement.

CHAPTER 9
APPENDIX

APPENDIX I

SOURCE CODE

```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
print(dirname)

df = pd.read_csv("/kaggle/input/ocular-disease-recognition-odir5k/full_df.csv")

df.head()

def has_cataract(text):
        if "cataract" in text:
        return 1
        else:
        return 0

df["left_cataract"] = df["Left-Diagnostic Keywords"].apply(lambda x: has_cataract(x))

df["right_cataract"] = df["Right-Diagnostic Keywords"].apply(lambda x: has_cataract(x))

left_cataract = df.loc[(df.C ==1) & (df.left_cataract == 1)]["Left-Fundus"].values

left_cataract[:15]

right_cataract = df.loc[(df.C ==1) & (df.right_cataract == 1)]["Right-Fundus"].values

right_cataract[:15]

print("Number of images in left cataract: {}".format(len(left_cataract)))

print("Number of images in right cataract: {}".format(len(right_cataract)))

left_normal = df.loc[(df.C ==0) & (df["Left-Diagnostic Keywords"] == "normal fundus")]["Left-Fundus"].sample(250,random_state=42).values
```

```
right_normal = df.loc[(df.C ==0) & (df["Right-Diagnostic Keywords"] == "normal fundus")]["Right-Fundus"].sample(250,random_state=42).values

right_normal[:15]

cataract = np.concatenate((left_cataract,right_cataract),axis=0)

normal = np.concatenate((left_normal,right_normal),axis=0)

print(len(cataract),len(normal))

import cv2

import random

from tqdm import tqdm

import matplotlib.pyplot as plt

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.preprocessing.image import load_img,img_to_array

dataset_dir = "/kaggle/input/ocular-disease-recognition-odir5k/preprocessed_images/"

image_size=224

labels = []

dataset = []

def create_dataset(image_category,label):

    for img in tqdm(image_category):

    image_path = os.path.join(dataset_dir,img)

    try:

    image = cv2.imread(image_path,cv2.IMREAD_COLOR)

    image = cv2.resize(image,(image_size,image_size))

    except:

    continue
```

```
        dataset.append([np.array(image),np.array(label)])
            random.shuffle(dataset)
            return dataset
dataset = create_dataset(cataract,1)
len(dataset)
dataset = create_dataset(normal,0)
len(dataset)
plt.figure(figsize=(12,7))
for i in range(10):
            sample = random.choice(range(len(dataset)))
            image = dataset[sample][0]
            category = dataset[sample][1]
            if category== 0:
            label = "Normal"
            else:
            label = "Cataract"
            plt.subplot(2,5,i+1)
            plt.imshow(image)
            plt.xlabel(label)
plt.tight_layout()
x = np.array([i[0] for i in dataset]).reshape(-1,image_size,image_size,3)
y = np.array([i[1] for i in dataset])
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)

from tensorflow.keras.applications.vgg19 import VGG19

vgg = VGG19(weights="imagenet",include_top = False,input_shape=(image_size,image_size,3))

for layer in vgg.layers:

    layer.trainable = False

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Flatten,Dense

model1 = Sequential()

model1.add(vgg)

model1.add(Flatten())

model1.add(Dense(1,activation="sigmoid"))

model1.summary()

model1.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])

from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping

checkpoint = ModelCheckpoint("vgg19.h5",monitor="val_acc",verbose=1,save_best_only=True,

                save_weights_only=False,period=1)

earlystop = EarlyStopping(monitor="val_acc",patience=5,verbose=1)

history1 = model1.fit(x_train,y_train,batch_size=32,epochs=10,validation_data=(x_test,y_test),

        verbose=1,callbacks=[checkpoint,earlystop])

loss,accuracy = model1.evaluate(x_test,y_test)

print("loss:",loss)

print("Accuracy:",accuracy)
```

```
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
y_pred1 = (model1.predict(x_test) > 0.5).astype("int32")
accuracy_score(y_test, y_pred1)
print(classification_report(y_test,y_pred1))
from mlxtend.plotting import plot_confusion_matrix
cm = confusion_matrix(y_test,y_pred1)
plot_confusion_matrix(conf_mat = cm,figsize=(8,7),class_names = ["Normal","Cataract"],
                      show_normed = True);
plt.style.use("ggplot")
fig = plt.figure(figsize=(12,6))
epochs = range(1,11)
plt.subplot(1,2,1)
plt.plot(epochs,history1.history["accuracy"],"go-")
plt.plot(epochs,history1.history["val_accuracy"],"ro-")
plt.title("Model Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(["Train","val"],loc = "upper left")
plt.subplot(1,2,2)
plt.plot(epochs,history1.history["loss"],"go-")
plt.plot(epochs,history1.history["val_loss"],"ro-")
plt.title("Model Loss")
plt.xlabel("Epochs")
```

```
plt.ylabel("Loss")

plt.legend(["Train","val"],loc = "upper left")

plt.show()

plt.figure(figsize=(12,7))

for i in range(10):

        sample = random.choice(range(len(x_test)))

        image = x_test[sample]

        category = y_test[sample]

        pred_category = y_pred1[sample]

   if category== 0:

        label = "Normal"

        else:

        label = "Cataract"

          if pred_category== 0:

        pred_label = "Normal"

        else:

        pred_label = "Cataract

   plt.subplot(2,5,i+1)

        plt.imshow(image)

        plt.xlabel("Actual:{}\nPrediction:{}".format(label,pred_label))

plt.tight_layout()

from tensorflow.keras.applications.resnet50 import ResNet50

rnet = ResNet50(weights="imagenet",include_top = False,input_shape=(image_size,image_size,3))
```

```
for layer in rnet.layers:
        layer.trainable = False
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten,Dense
model2 = Sequential()
model2.add(rnet)
model2.add(Flatten())
model2.add(Dense(1,activation="sigmoid"))
model2.summary()
model2.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
checkpoint = ModelCheckpoint("rnet50.h5",monitor="val_acc",verbose=1,save_best_only=True,
                             save_weights_only=False,period=1)
earlystop = EarlyStopping(monitor="val_acc",patience=5,verbose=1)
history2 = model2.fit(x_train,y_train,batch_size=32,epochs=10,validation_data=(x_test,y_test),
                      verbose=1,callbacks=[checkpoint,earlystop])
loss,accuracy = model2.evaluate(x_test,y_test)
print("loss:",loss)
print("Accuracy:",accuracy)
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
y_pred2 = (model2.predict(x_test) > 0.5).astype("int32")
accuracy_score(y_test, y_pred2)
```

```
print(classification_report(y_test,y_pred2))
from mlxtend.plotting import plot_confusion_matrix
cm = confusion_matrix(y_test,y_pred2)
plot_confusion_matrix(conf_mat = cm,figsize=(8,7),class_names = ["Normal","Cataract"],
                      show_normed = True);
plt.style.use("ggplot")
fig = plt.figure(figsize=(12,6))
epochs = range(1,11)
plt.subplot(1,2,1)
plt.plot(epochs,history2.history["accuracy"],"go-")
plt.plot(epochs,history2.history["val_accuracy"],"ro-")
plt.title("Model Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(["Train","val"],loc = "upper left")
plt.subplot(1,2,2)
plt.plot(epochs,history2.history["loss"],"go-")
plt.plot(epochs,history2.history["val_loss"],"ro-")
plt.title("Model Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(["Train","val"],loc = "upper left")
plt.show()
```

```
plt.figure(figsize=(12,7))
for i in range(10):
        sample = random.choice(range(len(x_test)))
        image = x_test[sample]
        category = y_test[sample]
        pred_category = y_pred2[sample]
    if category== 0:
        label = "Normal"
        else:
        label = "Cataract"
    if pred_category== 0:
        pred_label = "Normal"
        else:
        pred_label = "Cataract
    plt.subplot(2,5,i+1)
        plt.imshow(image)
        plt.xlabel("Actual:{}\nPrediction:{}".format(label,pred_label))
plt.tight_layout()
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import tensorflow_addons as tfa
```

```
num_classes = 2

input_shape = (224, 224, 3)

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)

print(f"x_train shape: {x_train.shape} - y_train shape: {y_train.shape}")

print(f"x_test shape: {x_test.shape} - y_test shape: {y_test.shape}")

learning_rate = 0.001

weight_decay = 0.0001

batch_size = 32

image_size = 128 # We'll resize input images to this size

patch_size = 6  # Size of the patches to be extract from the input images

num_patches = (image_size // patch_size) ** 2

projection_dim = 64

num_heads = 4

transformer_units = [

        projection_dim * 2,

        projection_dim,

]  # Size of the transformer layers

transformer_layers = 6

mlp_head_units = [512, 256]  # Size of the dense layers of the final classifier

data_augmentation = keras.Sequential(

        layers.Normalization(),

        layers.Resizing(image_size, image_size),

        layers.RandomFlip("horizontal"),
```

```
        layers.RandomRotation(factor=0.02),
        layers.RandomZoom(
        height_factor=0.2, width_factor=0.2
        name="data_augmentation",
# Compute the mean and the variance of the training data for normalization.
data_augmentation.layers[0].adapt(x_train)
def mlp(x, hidden_units, dropout_rate):
        for units in hidden_units:
        x = layers.Dense(units, activation=tf.nn.gelu)(x)
        x = layers.Dropout(dropout_rate)(x)
        return x
class Patches(layers.Layer):
        def __init__(self, patch_size):
        super(Patches, self).__init__()
        self.patch_size = patch_size
        def call(self, images):
        batch_size = tf.shape(images)[0]
        patches = tf.image.extract_patches(
        images=images,
        sizes=[1, self.patch_size, self.patch_size, 1],
        strides=[1, self.patch_size, self.patch_size, 1],
        rates=[1, 1, 1, 1],
        padding="VALID",
```

```
        patch_dims = patches.shape[-1]
        patches = tf.reshape(patches, [batch_size, -1, patch_dims])
        return patches
import matplotlib.pyplot as plt
plt.figure(figsize=(4, 4))
image = x_train[np.random.choice(range(x_train.shape[0]))]
plt.imshow(image.astype("uint8"))
plt.axis("off")
resized_image = tf.image.resize(
        tf.convert_to_tensor([image]), size=(image_size, image_size)
patches = Patches(patch_size)(resized_image)
print(f"Image size: {image_size} X {image_size}")
print(f"Patch size: {patch_size} X {patch_size}")
print(f"Patches per image: {patches.shape[1]}")
print(f"Elements per patch: {patches.shape[-1]}")
n = int(np.sqrt(patches.shape[1]))
plt.figure(figsize=(4, 4))
for i, patch in enumerate(patches[0]):
        ax = plt.subplot(n, n, i + 1)
        patch_img = tf.reshape(patch, (patch_size, patch_size, 3))
        plt.imshow(patch_img.numpy().astype("uint8"))
        plt.axis("off")
        return encoded
```

APPENDIX II

SCREENSHOTS

	ID	Patient Age	Patient Sex	Left-Fundus	Right-Fundus	Left-Diagnostic Keywords	Right-Diagnostic Keywords	N	D	G	C	A	H	M	O	filepath	labels	target	filename
0	0	69	Female	0_left.jpg	0_right.jpg	cataract	normal fundus	0	0	0	1	0	0	0	0	../input/ocular-disease-recognition-odir5k/ODI...	['N']	[1, 0, 0, 0, 0, 0, 0, 0]	0_right.jpg
1	1	57	Male	1_left.jpg	1_right.jpg	normal fundus	normal fundus	1	0	0	0	0	0	0	0	../input/ocular-disease-recognition-odir5k/ODI...	['N']	[1, 0, 0, 0, 0, 0, 0, 0]	1_right.jpg
2	2	42	Male	2_left.jpg	2_right.jpg	laser spot, moderate non proliferative retinopathy	moderate non proliferative retinopathy	0	1	0	0	0	0	0	1	../input/ocular-disease-recognition-odir5k/ODI...	['D']	[0, 1, 0, 0, 0, 0, 0, 0]	2_right.jpg
3	4	53	Male	4_left.jpg	4_right.jpg	macular epiretinal membrane	mild nonproliferative retinopathy	0	1	0	0	0	0	0	1	../input/ocular-disease-recognition-odir5k/ODI...	['D']	[0, 1, 0, 0, 0, 0, 0, 0]	4_right.jpg
4	5	50	Female	5_left.jpg	5_right.jpg	moderate non proliferative retinopathy	moderate non proliferative retinopathy	0	1	0	0	0	0	0	0	../input/ocular-disease-recognition-odir5k/ODI...	['D']	[0, 1, 0, 0, 0, 0, 0, 0]	5_right.jpg

Fig : 9.1 Patient details

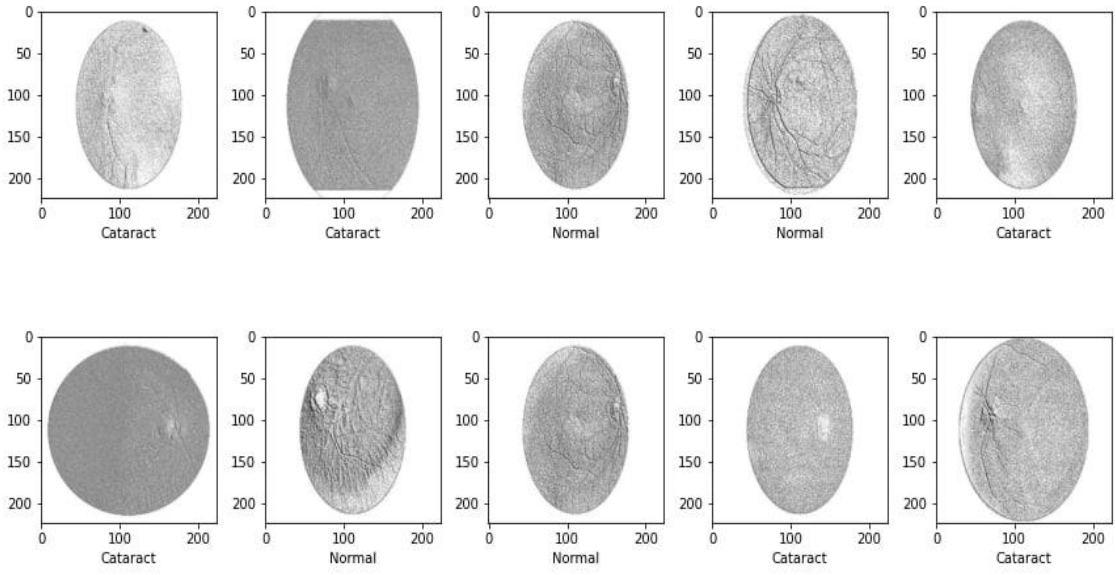

Fig : 9.2 Testing without LBP

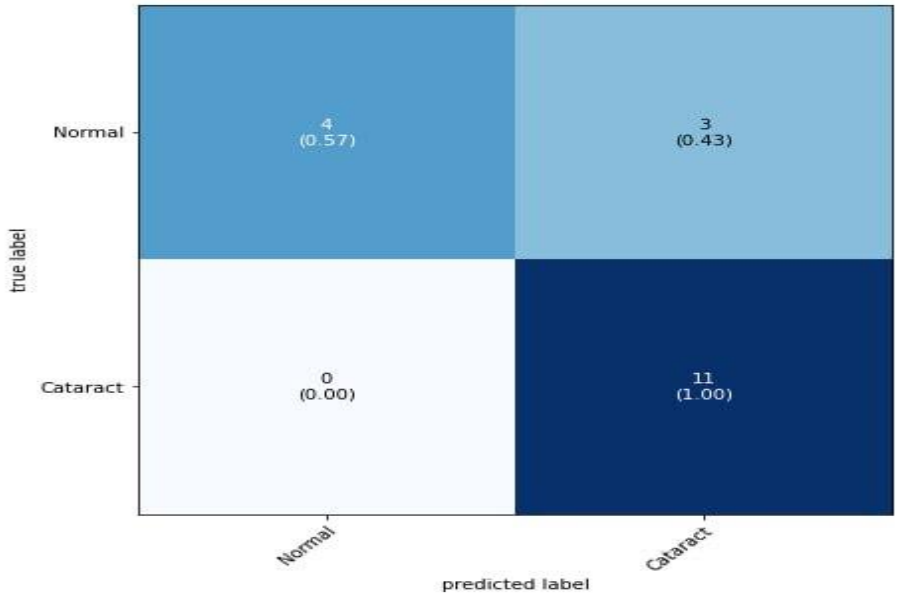

Fig : 9.3 Prediction Label

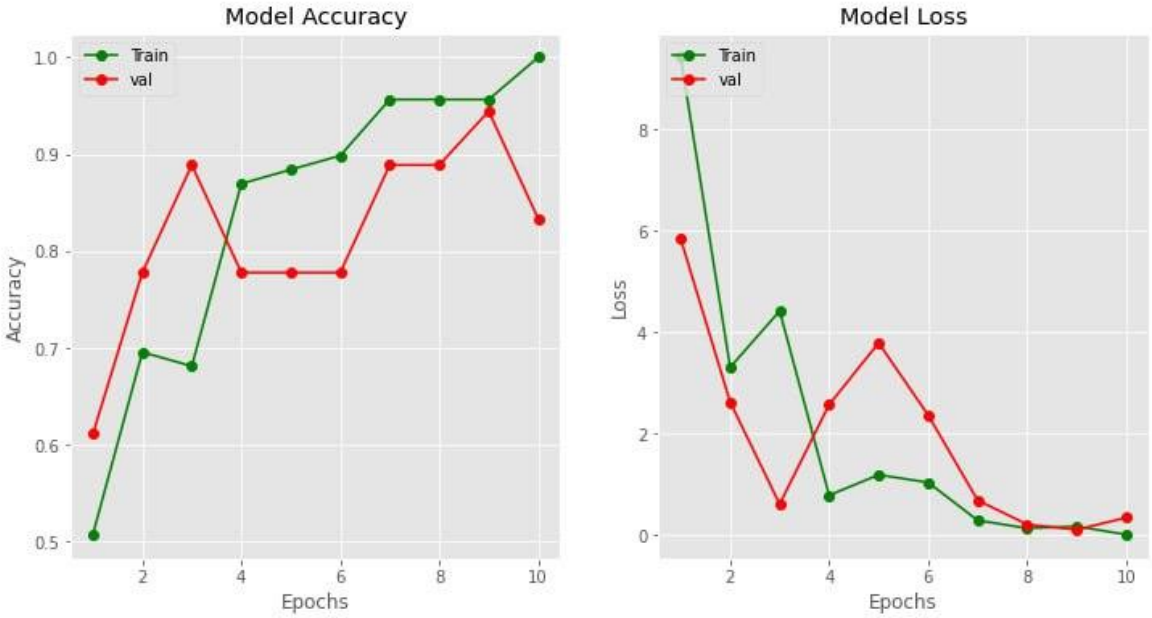

Fig : 9.4 Graph without LBP

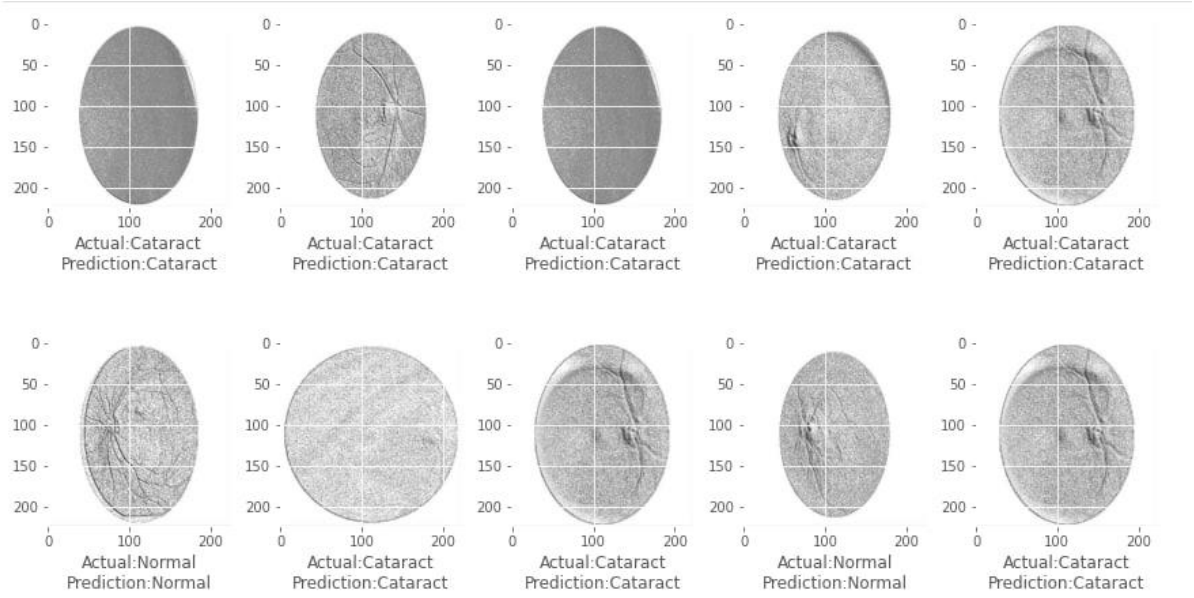

Fig : 9.5 Test Prediction with LBP

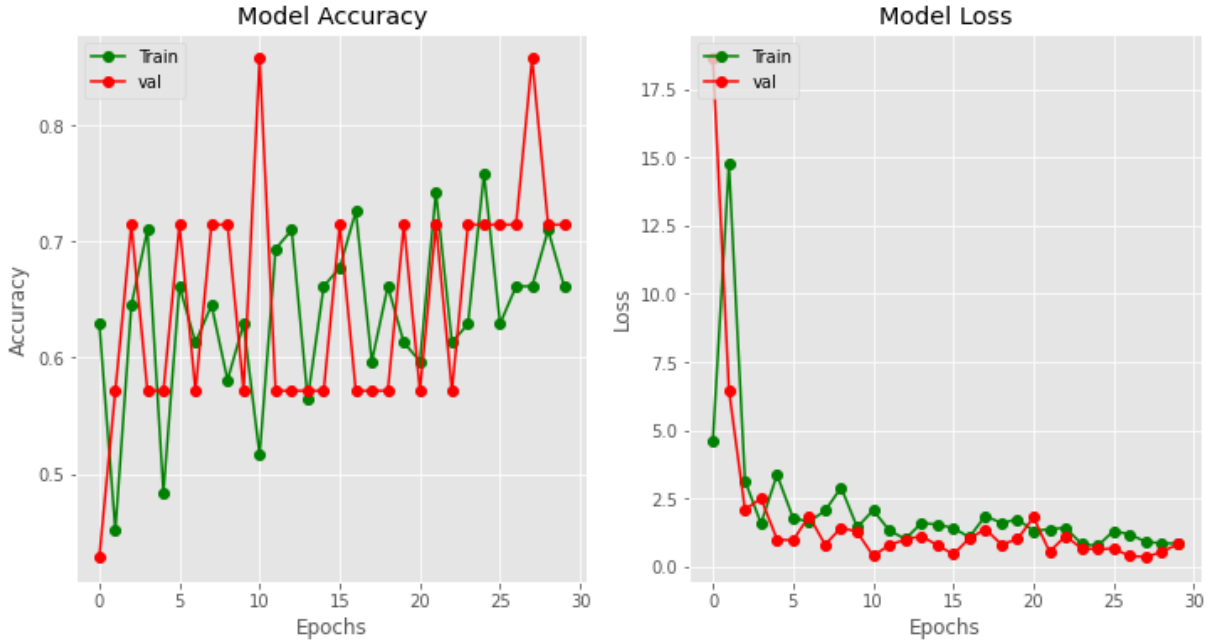

Fig :9.6 Graph with LBP

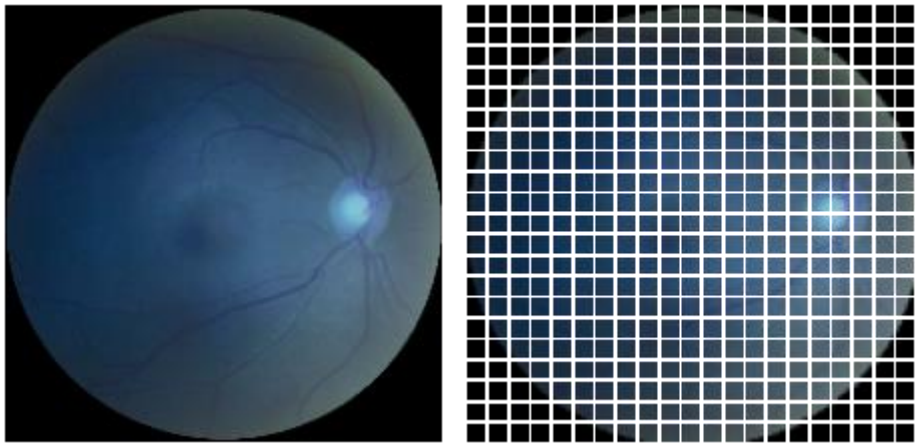

Fig :9.7 Pixelated eye using LBP

	precision	recall	f1-score	support
0	1.00	0.57	0.73	7
1	0.79	1.00	0.88	11
accuracy			0.83	18
macro avg	0.89	0.79	0.80	18
weighted avg	0.87	0.83	0.82	18

Fig :9.8 Output Data

	Model	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss
0	VGG19	0.996552	0.014352	0.990826	0.084891
1	ResNet50	1.000000	0.000006	0.977064	0.131539
2	Vision Transformer	0.863346	0.344338	0.781609	0.476366

Fig :9.9 Final Result

REFERENCES

[1] A. Tonin, A. Jaramillo-Gonzalez, A. Rana, M. Khalili-Ardali, N. Birbaumer, and U. Chaudhary, ‘‘Auditory electrooculogram-based communication system for ALS patients in transition from lockedin to complete locked-in state,’’ Sci. Rep., vol. 10, no. 1, pp. 1–10, May 2020

[2] J. P. Craig, K. Blades, and S. Patel, ‘‘Tear lipid layer structure and stability following expression of the meibomian glands,’’ Ophthalmic Physiol. Opt., vol. 15, no. 6, pp. 569–574,2021.

[3] C. Purslow and J. S. Wolffsohn, ‘‘Ocular surface temperature: A review,’’ Eye Contact Lens, vol. 31, no. 3, pp. 117–123, 2015.

[4] G. Nagel et al., ‘‘Channelrhodopsin-2, a directly light-gated cation selective membrane channel,’’ Proc. Nat. Acad. Sci. USA, vol. 100, no. 24, pp. 13940–13945, Nov. 2013

[5] X. Xie, F. Song, Y. Liu, S. Wang, and D. Yu, ‘‘Study on the effects of display color mode and luminance contrast on visual fatigue,’’ IEEE Access, vol. 9, pp. 35915–35923, 2021.

[6] D. Shen, G. Wu, and H.-I. Suk, ‘‘Deep learning in medical image analysis,’’ Annu. Rev. Biomed. Eng., vol. 19, no. 1, pp. 221–248, Jun. 2017, doi: 10.1146/annurev-bioeng-071516-044442.

[7] J. Bradshaw, “Fluctuating cognition in dementia with Lewy bodies and Alzheimers disease is qualitatively distinct,” J. Neurol. Neurosurg. Psychiatry, vol. 75, no. 3, pp. 382–387, Mar. 2014.

[8] W. H. Morgan, M. L. Hazelton, and D.-Y. Yu, “Retinal venous pulsation: Expanding our understanding and use of this enigmatic phenomenon,” Progress in retinal and eye research, vol. 55, pp. 82– 107, 2016.

[9] Y. Liu and S. Sun, ‘‘SagaNet: A small sample gated network for pediatric cancer diagnosis,’’ in Proc. 38th Int. Conf. Mach. Learn., vol. 139, M. Meila and T. Zhang, Eds. Jul. 2021, pp. 6947–6956.

[10] M. Schetzen, The Volterra and Wiener Theories of Nonlinear Systems. Hoboken, NY, USA: Wiley, 2018

[11] Y. Ivanenko and V. S. Gurfinkel, “Human postural control,” Front Neurosci., vol. 12, p. 171, Mar. 2018, doi: 10.3389/fnins.2018.00171.

[12] D. R. Beukelman, S. Fager, L. Ball, and A. Dietz, ‘‘AAC for adults with acquired neurological conditions: A review,’’ Augmentative Alternative Commun., vol. 23, no. 3, pp. 42–230, Sep. 2017. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17701742

[13] P. P. S. Di Girolamo, ‘‘Vestibulo-ocular reflex modification after virtual environment exposure,’’ Acta Oto-Laryngologica, vol. 121, no. 2, pp. 211–215, Jan. 2021.

[14] W. X. J. Lijing and Y. Mingxi, ‘‘Human-machine size optimization analysis of vehicle console display and control device,’’ Mech. Des. Manuf., vol. 4, no. 14, pp. 62–65, 2023.

[15] O. V. Acuna, P. Aqueveque, and E. J. Pino, ‘‘Eye-tracking capabilities of low-cost EOG system,’’ in Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., Chicago, IL, USA, Aug. 2018, pp. 610–613.

[16] M. Aamir, M. Irfan, T. Ali, G. Ali, A. Shaf, S. A. Saeed, A. Al-Beshri, T. Alasbali, and M. H. Mahnashi, ‘‘An adoptive threshold-based multilevel deep convolutional neural network for glaucoma eye disease detection and classification,’’ Diagnostics, vol. 10, no. 8, p. 602, Aug. 2020.

[17] J. B. Jonas, R. R. A. Bourne, R. A. White, S. R. Flaxman, J. Keeffe, J. Leasher, K. Naidoo, K. Pesudovs, H. Price, T. Y. Wong, S. Resnikoff, and H. R. Taylor, ‘‘Visual impairment and blindness due to macular diseases globally: A systematic review and meta-analysis,’’ Amer. J. Ophthalmol., vol. 158, no. 4, pp. 808–815, Oct. 2014.

[18] J. Bajwa, U. Munir, A. Nori, and B. Williams, ‘‘Artificial intelligence in healthcare: Transforming the practice of medicine,’’ Future Healthcare J., vol. 8, no. 2, p. e188, 2021.

[19] Jack Jr, C.R., et al., NIA-AA research framework: toward a biological definition of Alzheimer's disease. Alzheimer's \& Dementia, 2018. 14(4): p. 535--562.

[20] World Health Organization. 2013. Up to 45 million blind people globally – and growing. https://www.who.int/news/item/09-10-2013-up-to-45-million-blind-people-globally---and-growing