Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101

A Project Report submitted in the partial fulfillment of the

Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

K.Santhoshkumar (21471A05G6)

T.Babu (21471A05E4)

M.VenkataThirumala (21471A05H5)

Under the esteemed guidance of

Dr. K.Suresh Babu, M.Tech., Ph.D.,

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier -1

NIRF rank in the band of 201-300 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALLAMANDA VILLAGE,NARASARAOPET-522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "**Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101**" is a bonafide work done by the team K.Santhoshkumar (21471A05G6), T.Babu (21471A05E4), M.Venkata Thirumala (21471A05H5) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE
Dr.K.Suresh Babu, M. Tech.,Ph.D.
Associate Professor

PROJECT CORDINATOR
Dr. Sireesha Moturi, M.Tech.,Ph.D.
Associate Professor

HEAD OF THE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,
Professor & HOD

EXTERNALEXAMINER

DECLARATION

We declare that this project work titled " **Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101** " is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for another degree or professional qualification except as specified.

K.Santhoshkumar (21471A05G6)
T.Babu (21471A05E4)
M.VenkataThirumala (21471A05H5)

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M.V. Koteswara Rao, B.Sc.,** who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr. S. Venkateswarlu, Ph.D.,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,** HOD of CSE department and also to our guide **Dr.K.Suresh Babu, M.Tech., Ph.D.,** Associate professor of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dr. Sireesha Moturi, M.Tech., Ph.D.,** Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying our doubts which had really helped us in successfully completing our project.

By

K.Santhoshkumar (21471A05G6)

T.Babu (21471A05E4)

M.VenkataThirumala (21471A05H5)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertise in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one,,s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyze the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		✓											✓		
C421.2	✓		✓		✓								✓		
C421.3				✓		✓	✓	✓					✓		
C421.4			✓			✓	✓	✓					✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	The study predicts retinal diseases using Resnet-101 deep learning model, achieving high accuracy , achieving high accuracy in the Multi-class classification.	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our Three members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project focuses on retinal disease classification using resnet-101 and future updates can include enhancements for detecting additional retinal conditions .	PO4, PO7
C32SC4.3	The physical design includes web-site for retinal disease classification ,where users can upload retinal images to detect and classify disease	PO5, PO6

ABSTRACT

The classification of retinal diseases is a crucial aspect of medical diagnostics, and advancements in Machine Learning (ML) and Deep Learning (DL) are making significant strides in this domain. This project presents a ResNet-101-based deep learning model trained on the EyeNet dataset, which includes 32 different retinal diseases. The model is optimized using data augmentation techniques (rescaling, zooming, flipping) and employs the Adam optimizer with a learning rate of 0.001, ensuring high classification accuracy. Our approach achieves an accuracy of 98.75%, outperforming traditional diagnostic methods.Performance evaluation metrics such as accuracy, precision, and recall demonstrate the model's effectiveness in recognizing and categorizing different retinal conditions. The results indicate that early and accurate detection of retinal diseases can significantly improve patient outcomes. This study also highlights potential future enhancements in dataset expansion and integration of real-time analysis for clinical applications.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Vision is one of the most critical human senses, and any impairment can severely impact an individual,,s quality of life. Retinal diseases are a major cause of blindness worldwide, affecting millions of people. Early detection and classification of these diseases are essential for timely treatment and effective management. However, traditional diagnostic methods rely heavily on manual inspection by ophthalmologists, which can be time-consuming, costly, and prone to human error[1]. Thus, there is a need for an automated and efficient approach to address this challenge. With advancements in artificial intelligence (AI) and deep learning, automated retinal disease classification systems have emerged as a promising solution. Deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated outstanding performance in medical image classification tasks. These models can learn complex patterns from large datasets, enabling accurate and efficient disease identification[4]. Among various deep learning architectures, ResNet- 101 has proven to be highly effective due to its deep architecture, residual learning capability, and robustness in feature extraction.

This project focuses on developing a retinal disease classification system using the ResNet-101 model, incorporating advanced data preprocessing and augmentation techniques to improve performance.

Need for Automated Retinal Disease Detection

The increasing prevalence of retinal disorders and the limitations of manual diagnosis highlight the need for an automated disease classification system. Some key reasons include:

- **Early Disease Detection:** Timely diagnosis helps prevent vision loss and improves treatment outcomes.
- **High Accuracy and Efficiency:** AI-based models provide more consistent and accurate results compared to traditional methods.
- **Reduction in Clinical Workload:** Automating the classification process minimizes the burden on ophthalmologists and speeds up diagnosis.
- **Scalability and Accessibility:** AI-driven solutions can be deployed in

remote areas, improving access to quality healthcare.

- **Integration with Smart Healthcare Systems:** The use of AI can enhance telemedicine and mobile diagnostic applications, making healthcare more accessible.

Objectives of the Project

The primary goal of this project is to develop a deep learning-based system for the classification of retinal diseases[10]. The key objectives include:

- To develop an automated retinal disease detection system using deep learning techniques.
- To classify multiple retinal diseases with high accuracy, minimizing human intervention.
- To leverage the ResNet-101 model for better feature extraction and classification performance.
- To enhance model generalization using data augmentation and preprocessing techniques.

Significance of the Project

The implementation of an AI-based retinal disease detection system has significant implications for modern healthcare[9]. Some key benefits include:

- **Improved Diagnosis Accuracy:** AI reduces the chances of misdiagnosis and ensures consistent results.
- **Efficient Screening Process:** Automated classification allows for rapid screening of retinal conditions.
- **Cost-Effective Solution:** Reducing dependency on expert ophthalmologists makes the diagnosis process more affordable.
- **Technology-Driven Healthcare:** The adoption of deep learning in ophthalmology enhances the capabilities of medical professionals.

The human retina is a vital component of the visual system, acting as a photosensitive layer that converts light into neural signals, which are transmitted to the brain via the optic nerve. The macula, a central part of the retina, is responsible for processing fine visual details. Any disruption or degeneration of the retinal layers can

lead to serious visual impairments or complete vision loss. Retinal diseases such as Age-Related Macular Degeneration (AMD), Diabetic Macular Edema (DME), Retinal Detachment, and Retinitis Pigmentosa affect millions of individuals globally, posing significant challenges to ophthalmologists in terms of early detection and accurate diagnosis[7]. Timely identification of retinal diseases is critical for effective treatment and prevention of irreversible vision loss. However, traditional diagnostic methods require highly trained ophthalmologists to analyze retinal images manually, making the process time-consuming, prone to human error, and challenging to implement in under-resourced healthcare settings.

In recent years, Artificial Intelligence (AI) and Machine Learning (ML) have revolutionized various aspects of medical diagnostics, particularly in the field of ophthalmology. Deep Learning (DL), a subset of ML, has demonstrated remarkable success in analyzing complex medical images, enabling automated disease detection and classification. Advanced deep learning architectures such as Convolutional Neural Networks (CNNs), ResNet, and AlexNet have been extensively utilized for retinal disease diagnosis, offering high precision and efficiency compared to conventional diagnostic techniques[12]. Computer-aided diagnostic (CAD) systems, powered by deep learning, have the potential to significantly enhance the accuracy and speed of retinal disease detection. These systems can process large datasets of retinal images, extract essential features, and classify diseases with minimal human intervention. By automating the diagnostic process, deep learning models help in early disease detection, reducing the dependency on specialists and improving patient outcomes.

While deep learning-based approaches have shown promise in retinal disease classification, several challenges persist, including the need for extensive labeled datasets, computational complexity, and overfitting issues. Moreover, retinal images often exhibit variations in illumination, contrast, and resolution, making disease classification a complex task[14]. To address these challenges, this study focuses on optimizing deep learning techniques to enhance classification accuracy and generalization capability.ResNet-101, a powerful deep neural network with 101 layers, is chosen for its ability to perform deep feature extraction while mitigating the vanishing gradient problem using residual learning. This architecture enables efficient

learning from retinal images, capturing both low-level and high-level visual patterns crucial for disease identification.

Traditional diagnostic methods for retinal diseases rely heavily on manual examination of fundus and Optical Coherence Tomography (OCT) images by trained ophthalmologists[5]. These methods, while effective, require significant expertise and are time-consuming, making them impractical for large-scale screening programs, particularly in rural or underserved regions where access to specialized healthcare is limited. Furthermore, manual diagnosis is inherently subjective, leading to variability in interpretations across different medical professionals. These challenges highlight the need for automated, highly accurate, and efficient diagnostic tools that can complement traditional ophthalmic screening methods and aid in the early detection of retinal diseases.

Advancements in artificial intelligence and deep learning have brought revolutionary changes to the field of medical imaging. In particular, convolutional neural networks have demonstrated remarkable success in extracting complex patterns from medical images, making them highly suitable for tasks such as disease classification, segmentation, and anomaly detection. Deep learning models can analyze vast amounts of medical data at unprecedented speeds, allowing for rapid and precise disease detection. Among various deep learning architectures, residual networks, or ResNet, have gained prominence due to their ability to facilitate deep learning without encountering the problem of vanishing gradients. This characteristic makes ResNet highly effective in medical imaging applications, where deep feature extraction is essential for differentiating between similar disease patterns. This research focuses on the development of an optimized deep learning model for the classification of multi-class retinal diseases using the ResNet-101 architecture[8]. ResNet-101, an advanced convolutional neural network with 101 layers, is particularly well-suited for medical image classification due to its ability to preserve critical features across multiple layers through residual learning. By implementing this architecture, the study aims to enhance the accuracy and efficiency of retinal disease detection. The model is trained using the EyeNet dataset, which comprises 32 distinct retinal disease categories, making it one of the most comprehensive datasets for retinal disease classification. The large variety of diseases included in the dataset ensures that

the model is capable of distinguishing between different conditions with high precision.

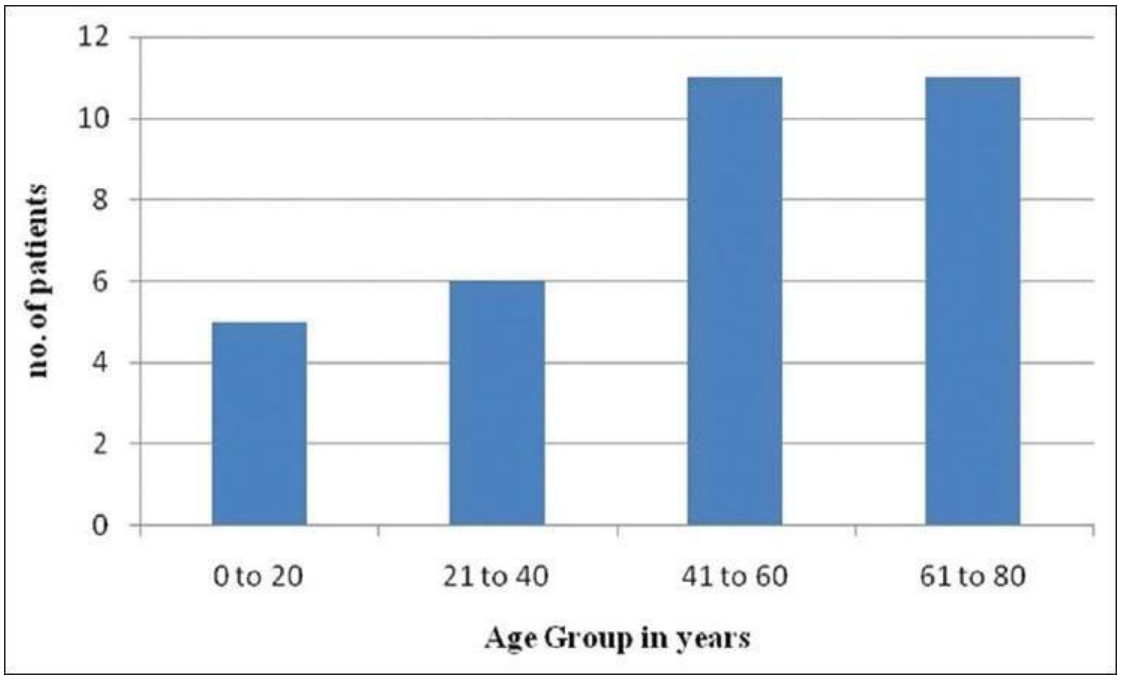


Fig 1.1: STAISTICS OF RETINAL DISEASES BY AGE GROUP

In the Fig 1.1 shows the distribution of patients with retinal diseases across different age groups. The highest number of cases is observed in the 41 to 60 age group, indicating a higher prevalence of this condition in middle-aged individuals.

2. LITERATURE SURVEY

Retinal diseases, such as age-related macular degeneration (AMD), diabetic retinopathy (DR), and diabetic macular edema (DME), are among the leading causes of vision impairment and blindness worldwide. Early and accurate diagnosis of these conditions is critical to prevent irreversible vision loss and improve patient outcomes. Traditional diagnostic methods rely heavily on manual analysis by ophthalmologists, which is not only time-consuming but also subjective and resource-intensive. This is particularly challenging in regions with limited access to healthcare, where the availability of specialized medical professionals is scarce. The growing prevalence of retinal diseases, coupled with the limitations of manual diagnosis, has created a pressing need for automated, scalable, and precise diagnostic solutions [1].

Recent advancements in Machine Learning (ML) and Deep Learning (DL) have revolutionized the field of medical imaging, offering new possibilities for the automated classification of retinal diseases. Deep Learning models, particularly Convolutional Neural Networks (CNNs), have emerged as state-of-the-art tools for analyzing retinal images due to their ability to automatically extract hierarchical features from complex datasets. These models have demonstrated exceptional performance in detecting pathologies from optical coherence tomography (OCT) and fundus images, making them highly suitable for tasks requiring intricate feature extraction, such as distinguishing between subtle retinal abnormalities [3].

The application of deep learning in retinal disease classification has evolved significantly over the past decade. Early studies primarily focused on binary classification tasks, such as distinguishing between healthy retinas and specific diseases like AMD or DR. For instance, Huang et al. developed an AI system using CNNs to screen for diabetic retinopathy, achieving high accuracy on the Messidor-2 dataset. Their work highlighted the potential of deep learning in reducing dependency on specialist evaluations, particularly in underserved regions where access to ophthalmologists is limited [2]. As the field progressed, researchers began to explore more complex multi-class classification tasks, which involve distinguishing between multiple retinal diseases. However, multi-class classification remains challenging due to the high inter-class similarity among retinal diseases. For example, the Eye PACS dataset, which categorizes diabetic retinopathy into five severity stages, posed

significant difficulties for models, with accuracy dropping to 75% in complex multi-class scenarios [12]. To address these challenges, researchers have explored deeper architectures like ResNet-101, which offers enhanced feature extraction through its 101 layers and skip connections. ResNet-101's residual learning framework addresses the vanishing gradient problem, making it highly suitable for tasks requiring detailed feature extraction, such as distinguishing between subtle retinal abnormalities [7].

In addition to standalone deep learning models, researchers have also explored hybrid approaches that combine CNNs with traditional machine learning classifiers like Support Vector Machines (SVMs) and Random Forests. These hybrid models leverage the feature extraction capabilities of CNNs while benefiting from the classification strengths of traditional algorithms. For example, Rahman et al. reviewed hybrid models that integrate CNNs with Random Forests for diabetic retinopathy staging, achieving 96% accuracy on the Messidor-2 dataset [4]. Similarly, ResNet-50 has been integrated with Random Forest classifiers for the detection of diabetic retinopathy, demonstrating the effectiveness of hybrid models in improving classification accuracy [6]. Despite the significant progress in deep learning-based retinal disease classification, several challenges remain. One of the primary limitations is the scarcity of large, annotated datasets. Medical imaging datasets, such as EyeNet and EyePACS, often suffer from class imbalance, limited samples for rare diseases, and variability in image quality. These issues can hinder the performance of deep learning models, particularly in multi-class classification tasks [12]. To address these challenges, data augmentation has become a cornerstone of modern deep learning pipelines. Techniques such as rotation, flipping, zooming, and brightness adjustment artificially expand the training dataset, improving model generalization and robustness to real-world variations in lighting, angles, and imaging equipment. For example, Kalshetty and Parveen emphasized that augmentation not only reduces overfitting but also enhances the model's ability to handle variations in image quality [10]. In the context of retinal imaging, specialized augmentations—such as simulating pathological artifacts or adjusting contrast to mimic OCT noise—have further improved model performance. Lee et al. applied grayscale conversion and saturation adjustments to highlight subtle lesions in OCT scans, enabling ResNet-based models to achieve 98% precision in detecting macular edema [11].

Training deep learning models for medical applications requires careful optimization to ensure robust performance. The Adam optimizer, which dynamically adjusts learning rates using gradient moments, has become a standard choice due to its computational efficiency and robustness to sparse gradients. Comparative studies by Hossain and Kabir showed that Adam outperforms traditional Stochastic Gradient Descent (SGD) in terms of convergence speed and final accuracy, particularly when paired with adaptive learning rate scheduling [6]. Evaluation metrics such as precision, recall, and F1-score are critical for assessing model performance in multi-class settings. The confusion matrix and classification report provide granular insights into per-class accuracy and misclassifications. For instance, a model achieving 100% precision but lower recall for a rare disease like Susac's Syndrome might prioritize specificity over sensitivity, necessitating further clinical validation [4]. Recent work by Al-Jabbar et al. also underscores the importance of cross-dataset validation to ensure generalizability beyond the training distribution [12].

Emerging trends in retinal disease diagnosis include the integration of multimodal data, such as combining OCT with fundus photography, to improve diagnostic accuracy. Additionally, vision transformers (ViTs) are gaining traction for their ability to model global context in images, offering a potential alternative to CNNs. However, the computational costs and the need for ultra-large datasets remain significant barriers to the widespread adoption of these advanced models [13]. Lightweight architectures like MobileNet and EfficientNet are being explored for deployment on edge devices, enabling real-time screening in resource-limited settings. These models offer a balance between performance and computational efficiency, making them suitable for deployment in mobile and telemedicine applications [9]. Furthermore, explainable AI (XAI) techniques, such as gradient-weighted class activation mapping (Grad-CAM), are gaining traction to enhance clinician trust by visualizing the model's decision-making processes [14]. Recent advancements in hardware acceleration, such as WinoNN's FPGA-based optimizations, could further enhance the feasibility of deploying complex models like ResNet-101 in clinical environments [9]. Federated learning offers a promising solution to data scarcity by enabling collaborative model training across institutions without sharing sensitive patient data. This approach could facilitate the development of more robust and generalizable models for retinal disease classification[4].

The field of retinal disease classification has seen significant advancements with the adoption of deep learning techniques. Models like ResNet-101, with their deep architectures and residual learning capabilities, have demonstrated exceptional performance in detecting and classifying retinal diseases. However, challenges such as data scarcity, class imbalance, and computational costs remain significant barriers to the widespread adoption of these technologies. Future research should focus on addressing these challenges through innovative approaches such as data augmentation, federated learning, and the development of lightweight architectures for edge deployment. By leveraging these advancements, automated retinal disease classification systems have the potential to revolutionize eye care, improving access to timely accurate diagnoses for patients worldwide.

Looking ahead, the integration of advanced AI techniques with real-world clinical workflows holds immense promise for the future of retinal disease diagnosis. One promising direction is the development of federated learning frameworks, which allow multiple institutions to collaboratively train models without sharing sensitive patient data. This approach not only addresses data privacy concerns but also enables the creation of more robust and generalizable models by leveraging diverse datasets from different geographic and demographic populations [4]. Additionally, the use of explainable AI (XAI) techniques, such as Grad-CAM, can help bridge the gap between AI systems and clinicians by providing transparent insights into how models make decisions. This is particularly important in medical applications, where trust and interpretability are critical for adoption [14].

Another area of future research is the integration of multimodal data, such as combining OCT scans with fundus images or even genetic data, to improve diagnostic accuracy. Multimodal approaches can capture complementary information from different sources, leading to more comprehensive and accurate diagnoses [13]. Furthermore, the development of lightweight models like MobileNet and EfficientNet for deployment on edge devices could enable real-time screening in remote and resource-limited settings, making advanced diagnostic tools more accessible to underserved populations [9].

Finally, the application of vision transformers (ViTs) in retinal disease classification is an emerging trend. ViTs, which leverage self-attention mechanisms to model global context in images, have shown promise in other medical imaging tasks and could offer a viable alternative to CNNs for retinal disease detection [13]. However, the computational costs and the need for large-scale datasets remain challenges that need to be addressed.

In conclusion, the field of retinal disease classification is rapidly evolving, with deep learning models like ResNet-101 leading the way in achieving high accuracy and efficiency. While significant challenges remain, ongoing advancements in data augmentation, federated learning, multimodal integration, and explainable AI are paving the way for more robust, scalable, and accessible diagnostic solutions. By continuing to innovate and address these challenges, researchers and clinicians can work together to improve global eye care standards and reduce the burden of vision loss caused by retinal diseases [1][3][14].

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Traditional approaches to retinal disease classification have predominantly relied on manual diagnosis by ophthalmologists using imaging techniques such as Optical Coherence Tomography (OCT) and fundus photography. These methods involve visual inspection of retinal scans to identify pathologies like age-related macular degeneration (AMD), diabetic retinopathy (DR), and diabetic macular edema (DME). While effective, this process is time-consuming, subjective, and highly dependent on the expertise of specialists, leading to variability in diagnoses.

In recent years, machine learning (ML) models such as Support Vector Machines (SVM), Decision Trees (DT), and Logistic Regression (LR) have been applied to automate retinal disease detection. These models analyze handcrafted features extracted from retinal images, such as texture patterns, blood vessel structures, and lesion characteristics. For example, early studies used SVM with features like Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF) for detecting retinopathy of prematurity (ROP), achieving moderate accuracy (~95.5%). Hybrid approaches combining CNNs with traditional classifiers, such as Random Forest, have also been explored for DR staging, yielding accuracies up to 96% on datasets like Messidor-2.

However, these systems face significant limitations. Traditional ML models require manual feature engineering, which is labor-intensive and may miss subtle, disease-specific patterns in high-resolution retinal images. Most studies use small, imbalanced datasets (e.g., EyePACS), leading to biased models that struggle with rare diseases or underrepresented classes. Models trained on specific datasets (e.g., OCT scans) often fail to generalize across imaging modalities or diverse patient demographics. Techniques like t-SNE for dimensionality reduction, while useful for visualization, do not directly improve predictive performance and add computational overhead. Existing models rarely analyze longitudinal data (e.g., disease progression over time), limiting their utility in chronic condition management.

Deep learning models like AlexNet and VGG have shown promise in automating feature extraction from raw images. However, early implementations suffered from overfitting due to limited training data and lacked the depth required to capture complex retinal abnormalities. For instance, ResNet-50-based models achieved 96% accuracy on DR detection but struggled with multi-class classification across 32 retinal diseases, as seen in the EyeNet dataset.

Overall, while traditional ML and early deep learning methods provided foundational advancements, their reliance on static datasets, manual interventions, and shallow architectures hindered scalability and accuracy in real-world clinical settings.

3.2 DISADVANTAGES OF THE EXISTING SYSTEM

Feature Dependency: Traditional machine learning models rely heavily on manual feature engineering, where features such as texture patterns, blood vessel structures, and lesion characteristics are extracted from retinal images. This process is labor-intensive and may miss subtle, disease-specific patterns in high-resolution images, leading to incomplete or inaccurate feature representations.

Dataset Limitations: Most existing systems are trained on small, imbalanced datasets (e.g., EyePACS), which often lack sufficient representation of rare diseases or underrepresented classes. This imbalance results in biased models that perform poorly on minority classes, reducing their effectiveness in real-world clinical scenarios.

Generalization Issues: Models trained on specific datasets, such as OCT scans, often fail to generalize across different imaging modalities or diverse patient demographics. This lack of adaptability limits their applicability in varied clinical settings, where imaging techniques and patient populations may differ significantly.

Computational Complexity: Techniques like t-SNE (t-distributed Stochastic Neighbor Embedding) are used for dimensionality reduction and visualization of high-dimensional data. While useful for exploratory analysis, these techniques do not directly improve predictive performance and add computational overhead, making the system less efficient.

Neglect of Temporal Data: Existing models rarely analyze longitudinal data, such as disease progression over time. This limits their utility in managing chronic conditions

like AMD or DR, where understanding temporal patterns is crucial for effective diagnosis and treatment planning.

Overfitting in Deep Learning Models: Early deep learning models, such as AlexNet and VGG, suffer from overfitting due to limited training data. This results in poor generalization to unseen data, reducing their reliability in clinical applications. For example, ResNet-50 achieved 96% accuracy on DR detection but struggled with multi- class classification across 32 retinal diseases in the EyeNet dataset.

Shallow Architectures: Traditional machine learning models and early deep learning architectures lack the depth required to capture complex retinal abnormalities. This limitation hinders their ability to achieve high accuracy in multi-class classification tasks, where subtle differences between diseases must be identified.

Manual Interventions: The reliance on manual interventions for feature extraction and model tuning increases the complexity and time required for diagnosis. This dependency on human expertise introduces variability and reduces the scalability of the system in large-scale clinical environments.

Static Datasets: Existing systems are often trained on static datasets that do not account for evolving disease patterns or new imaging technologies. This limits their ability to adapt to changes in clinical practices or advancements in medical imaging.

Limited Multi-Class Classification Performance: While models like ResNet-50 perform well in binary classification tasks (e.g., DR detection), they struggle with multi- class classification across a large number of retinal diseases (e.g., 32 diseases in the EyeNet dataset). This limitation reduces their effectiveness in comprehensive retinal disease diagnosis.

Lack of Explainability : Many existing machine learning and deep learning models, particularly deep neural networks, operate as "black boxes," making it difficult for clinicians to understand how predictions are made. This lack of explainability reduces trust in the system, as healthcare professionals require transparent and interpretable models to make informed decisions. Without clear insights into the decision-making process, the adoption of these systems in clinical practice is hindered.

3.3 PROPOSED SYSTEM

The proposed system introduces a ResNet-101 based deep learning model for multi-class retinal disease classification, automating feature extraction and achieving high accuracy on the EyeNet dataset**.** It employs data augmentation to handle imbalanced data and improve generalization. The system uses adaptive learning rate optimization (Adam) for efficient training and integrates into clinical workflows for real-time predictions. Designed for scalability, it addresses limitations of traditional systems, offering a robust solution for early diagnosis and treatment planning in diverse clinical settings.

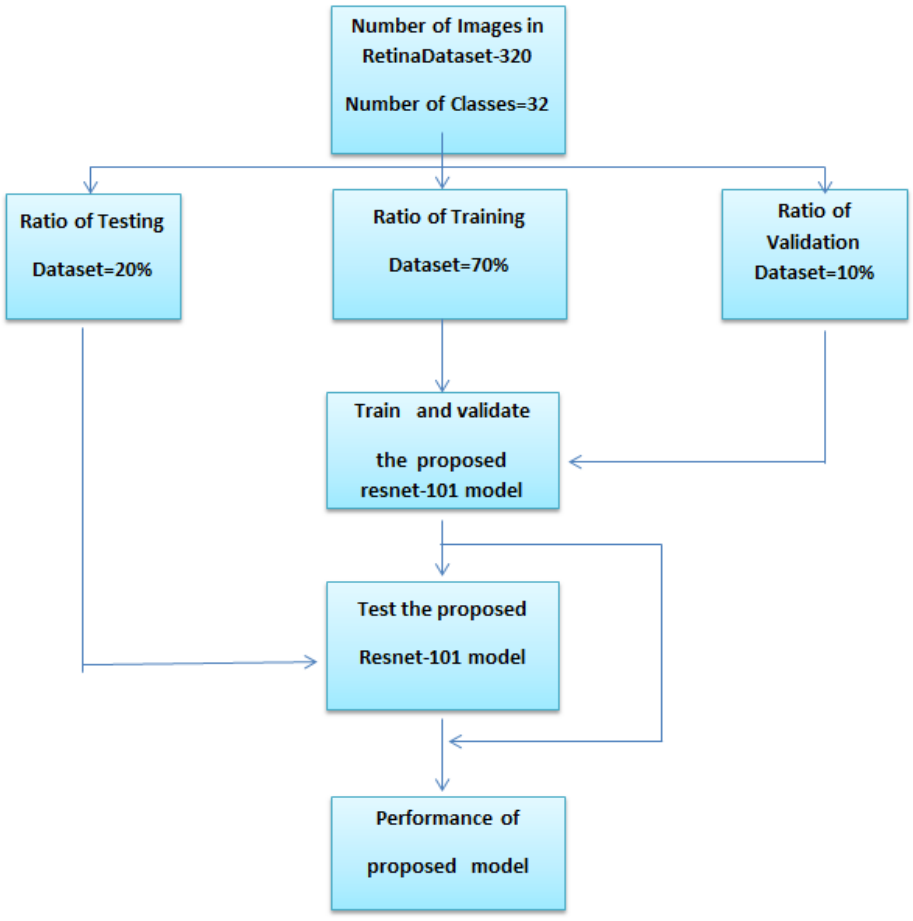


Fig 3.1: FLOWCHART OF PROPOSED MODEL

In Fig 3.1 flowchart begins with the RetinaDataset-320, containing 320 retinal images categorized into 32 distinct classes. The dataset is split into training (70%), validation (10%), and testing (20%) subsets. The ResNet-101 model is trained on the training dataset, learning hierarchical features from low-level patterns (e.g., edges) to high-level, disease-specific patterns. The validation dataset is used to fine-tune the model and prevent overfitting. The model is then tested on the testing dataset, achieving high accuracy. Performance metrics like accuracy, precision, recall, and F1-score are evaluated, and a confusion matrix visualizes classification results. The final output classifies retinal images into 32 disease categories, aiding in early diagnosis and treatment planning in clinical workflows. The methodology aims to accurately classify retinal diseases by analyzing fundus images, applying convolutional feature extraction, and leveraging residual learning for improved accuracy. The process involves the following steps:

1. Data Collection:

The first step involves gathering retinal images from medical sources, including hospital databases and publicly available datasets like Eye Net. The dataset consists of labeled images representing various retinal diseases to ensure comprehensive model training.

2. Data Preprocessing:

The collected data undergoes preprocessing to enhance model performance. This includes resizing images, normalizing pixel values, and applying augmentation techniques such as rotation and flipping to improve robustness. Missing values are handled, and images with poor quality are filtered out.

3. Feature Extraction:

Key features from retinal images are extracted using deep learning techniques. Resnet- 101 is employed to automatically detect patterns and anomalies. These extracted features are crucial in differentiating between normal and diseased retinal images.

4. Model Building :

The model used in this study is ResNet-101, which is chosen for its deep architecture and superior feature extraction capabilities. Resnet-101 employs residual connections

to enhance training efficiency and improve classification accuracy for multi-class retinal disease detection and which help in overcoming the vanishing gradient problem in deep networks. The model is optimized for multi-class retinal disease classification, extracting hierarchical features such as edges, shapes, and disease-specific patterns in deeper layers.

5. Model Training and Evaluation :

The model is trained using the training dataset, with optimization techniques such as Adam optimizer and learning rate scheduling applied to enhance performance. The evaluation metrics include accuracy, precision, recall to measure model effectiveness. A confusion matrix is used to assess classification performance.

6. Classification :

The trained model categorizes retinal images into multiple disease classes, allowing for accurate identification of conditions such as diabetic retinopathy, glaucoma, and macular degeneration. The classification results help in early detection and timely intervention.

7. Results and Visualization:

The classification outcomes are presented using graphical visualizations, including bar charts and confusion matrices, classification report.

8. System Implementation and Integration :

The trained model is integrated into a Flask-based web application, where users can upload retinal images and receive real-time disease classification results. The system is designed for scalability and can be adapted for clinical use in hospitals and diagnostic centers.

3.4 Feasibility Study

A Feasibility Study is a study that evaluates the practicability of a proposed project or system in the field of software engineering. The software project management process includes a feasibility assessment as one of the phases within the critical four stages. The goal of the feasibility study is to determine whether or not the proposed software product will be appropriate in terms of its development, implementation, and contribution to the organization , etc.

This analysis is based on a number of different propositions. Before the beginning of a project‟s actual work, a preliminary investigation called a “Feasibility Study” is carried out to determine how probable it is that the project would be successful. It is an investigation of the many potential options for resolving an issue, followed by a suggestion on which option is superior. The feasibility assessment of the proposed system is going to be carried out during the system analysis phase of the project. After the completion of the feasibility study, the Software Project Management Process provides a conclusion regarding whether or not to continue with the proposed project because it is practically feasible or whether or not to stop the proposed project here because it is not right or feasible to develop or to think about or analyse the proposed project once more. This makes the feasibility study a very important stage of the process.

The study of the possibility takes into account the following three main considerations:

1. Economic Feasibility
2. Technical Feasibility
3. Operational Feasibility

Economic Feasibility:

The economic feasibility of the proposed system evaluates the costs and potential benefits of developing and deploying the retinal disease classification model.

- **Costs**:
  - The primary costs include computational resources for training the ResNet-101 model, such as GPUs or cloud-based services like Google Colab Pro.
  - Additional costs may include data acquisition (if new datasets are required) and maintenance of the system.
- **Benefits**:
  - The system can significantly reduce the workload of ophthalmologists by automating the diagnosis of retinal diseases, leading to faster and more accurate diagnoses.

- Early detection of diseases like AMD, DR and DME can prevent vision loss, reducing long-term healthcare costs.

- The proposed system is economically feasible because it leverages open-source tools (e.g., Tensor Flow, Keras) and affordable cloud-based resources, minimizing development costs while maximizing benefits.

Technical Feasibility:

The technical feasibility assesses whether the necessary hardware, software, and technologies are available to develop the proposed system.

- **Hardware Requirements**:
  - The system requires GPUs (e.g., NVIDIA T4) for efficient training of the ResNet-101 model.
  - A standard computer with sufficient RAM (e.g., 8GB) and storage (e.g., 229GB) is needed for preprocessing and deployment.
- **Software Requirements**:
  - The system is developed using Python and deep learning frameworks like Tensor Flow and Keras.
  - Data augmentation and preprocessing are implemented using Keras' Image Data Generator.
- The proposed system is technically feasible because all required technologies (e.g., ResNet-101, Python libraries) are readily available and well-documented. The use of cloud-based platforms like Google Colab further reduces the need for high-end local hardware.

Operational Feasibility:

Operational feasibility evaluates whether the proposed system can be integrated into existing clinical workflows and used effectively by healthcare professionals.

- **Ease of Use**:
  - The system is designed to be user-friendly, with a simple interface for uploading retinal images and receiving classification results.

4. SYSTEM REQUIREMENTS

A Deep learning project requires specific hardware and software configurations to ensure efficient data processing, model training, and deployment. Below is a detailed explanation of each requirement: Key Features of Google Colab.

4.1 SOFTWARE REQUIREMENTS

- Operating System : Windows 11, 64-bit Operating System
- Coding Language : Python
- Python distribution : Flask , Visual Studio Code.
- Browser : Any Latest Browser like Google Chrome etc

4.2 HARDWARE REQUIREMENTS

- System Type : intel®core™i3-7500UCPU@2.40gh
- Cache memory : 4MB(Megabyte)
- RAM : 8GB (Gigabyte) or higher
- Hard Disk : 256 GB or Higher

4.3 SOFTWARE DESCRIPTION

1. **Python (3.x):**

   Python is the primary programming language used for developing the retinal disease classification system. It is an interpreted, high-level language known for its simplicity and extensive library support, making it ideal for deep learning and data processing tasks. Python,,s compatibility with deep learning frameworks like TensorFlow and Keras ensures seamless implementation of the ResNet-101 model.A virtual environment (venv) is recommended for dependency management, ensuring that the project,,s libraries do not conflict with other Python installations.

2. **TensorFlow and Keras:**

TensorFlow is an open-source deep learning framework used to build and train the ResNet-101 model**.** It provides a flexible platform for implementing complex neural networks and supports GPU acceleration for faster training. Keras, a high- level API built on TensorFlow, is used to simplify the development of the ResNet-101 architecture. Keras provides pre-built layers and functions for tasks like convolution, pooling, and fully connected layers, reducing the need for low-level coding. Keras,, ImageDataGenerator is used for data augmentation, applying transformations like flipping, zooming, and brightness adjustment to improve the model,,s generalization.

3. **NumPy:**

NumPy is a fundamental library for numerical computing in Python. It is used for handling large datasets, performing mathematical operations, and converting input data into multi-dimensional arrays .In this project, NumPy is essential for preprocessing retinal images, ensuring that the input data is in the correct format (e.g., normalized pixel values) before being fed into the ResNet-101 model.

4. **Google Colab:**

Google Colab is a cloud-based platform used for training the ResNet-101 model. It provides free access to GPUs and TPUs, significantly reducing the computational cost of training deep learning models. Colab,,s integration with Python and TensorFlow makes it an ideal choice for prototyping and deploying the retinal disease classification system.

5. **Matplotlib and Seaborn:**

Matplotlib and Seaborn are visualization libraries used to analyze and interpret the model,,s performance. These libraries are used to plot graphs such as accuracy and loss curves during training, as well as to visualize the confusion matrix for evaluating classification results.

6. HDF5 (.h5) File Format:

The trained ResNet-101 model is saved in the **.**h5 file format, which is the default format for saving Keras models. The .h5 file stores the model,,s architecture, weights, and optimizer state, allowing the model to be loaded and reused without retraining. This enables real-time predictions in clinical workflows.

The model can be saved using model.save(_model.h5,,) and loaded using keras.models.load_model(_model.h5,,).

7. Flask:

Flask is a lightweight web framework used to deploy the retinal disease classification system as a web application. It provides built-in support for routing, request handling, and templates. Flask can be used to create a user-friendly interface for uploading retinal images and displaying classification results, making the system accessible to healthcare professionals.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

Dataset Description

The EyeNet dataset, summarized in Table 1, provides a detailed overview of the retinal disease categories and their distribution. This dataset comprises 32 distinct retinal diseases, including conditions such as diabetic maculopathy, macular dystrophy, optic disc drusen, and retinal detachment.

Table 5.1: DATASET DESCRIPTION

Dataset	Class Label
Eye-Net	Adult Coats„ Disease
	AdultFoveomacularDystrophy
	Age-Related Macular Degeneration
	AMNMacularNeuroretionopathy
	Antiphospholipid Antibody Syndrome
	Behcet„s disease
	Bilteral Macular Dystrophy
	Bulls„s Eye Maculopathy Chloroquine
	Central Serous Chori-oretinopathy
	Choroidal Nevus
	CMV Chorioretinities
	Cone-Rod Dystrophy
	Congential Syphillis
	Diabetic Maculopathy Multiple Myeloma with Retinal Detachment
	Giant Retinal Tear
	North Carolina Dystrophy
	Laber„s Stel-late Maculopathhy
	Multifocal Exudative Detachments
	Macular Dystrophy
	Myelinated Nerve Fibers
	Juxtafoveal Telangiectasis DM Diabetes
	Optic Disc Drusen
	Roth Spot disease
	Pattern Dystrophy Simulating Fundus Flavimaculatus
	Retinal Folds Following Retinal Reattachment Surgery
	Reticular pattern Dystrophy
	Retro hyaloid Hemorrhage
	Solar Retinopathy Familial
	Susac„s Syndrome
	Self-Applied Retinal Detachement
	Terson„s syndrome
	Wyburn-Mason Syndrome

The dataset is structured to ensure a balanced representation of multiple disease classes, allowing for improved deep learning model training. Each class contains 10 images, ensuring uniform representation across all disease categories. The images are collected from authenticated medical sources and undergo preprocessing techniques like normalization and augmentation to enhance classification accuracy.In the Fig 5.2 shows some retinal disease images from the dataset. The dataset is divided into 70% training and 30% validation sets, ensuring robust model evaluation. The data for this research came from reputable sources, and it was combined with other data in order to create the dataset that was required for the project. Our dataset can be found at https://github.com/ huckiyang/EyeNet. This observation was made using Yang et al.„s EyeNet dataset. It offers a sorted variety of thirty-two multiple types of records related to records from hospitals.

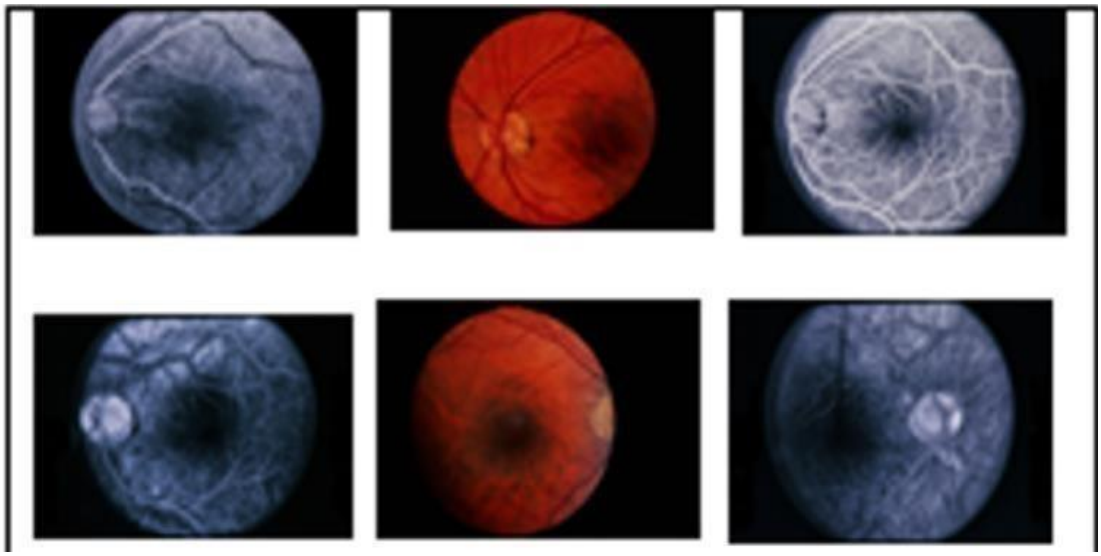


Fig 5.2: RETINAL DISEASE SAMPLE IMAGES

Data Pre-processing

Appropriate data pre-processing is required to ensure that deep learning models are able to interpret the medical images clearly. All the preprocessing steps were uniformly applied to both of the datasets and included the following:

1. Image Resizing and Normalization :

All images were resized to a standard uniform resolution of 224 x 224 pixels.

A conventional used for deep feature extraction, where convolutional layers identify hierarchical 14 features such as edges, textures, and disease-specific patterns. The residual learning framework enhances learning input size of the used convolutional neural networks in this study, this resizing ensured compatibility across the models and decreased the computational load without losing the pertinent detail in the images themselves.

2. Data Augmentation:

To make certain sufficient statistics is available for training and to prevent overfitting, we used information augmentation strategies. Augmentation strategies covered rescaling, zooming, and flipping the images. By applying numerous random differences, we ensured that the model encountered a diverse range of photos, which facilitates in avoiding overfitting and improves generalization. In our proposed version, information augmentation is carried out using Keras,,s Image Data Generator. This procedure includes six essential steps, each of which transforms the pix in distinctive approaches, improving the model,,s capacity to generalize nicely across new facts. Data augmentation is important for improving the overall efficiency and outcomes of system mastering fashions by means of producing extra and sundry education samples. A comprehensive dataset permits the model to perform extra accurately and efficaciously. For example, saturation changes increase the opacity of pics, making crucial capabilities extra visible and helping in higher sickness diagnosis.

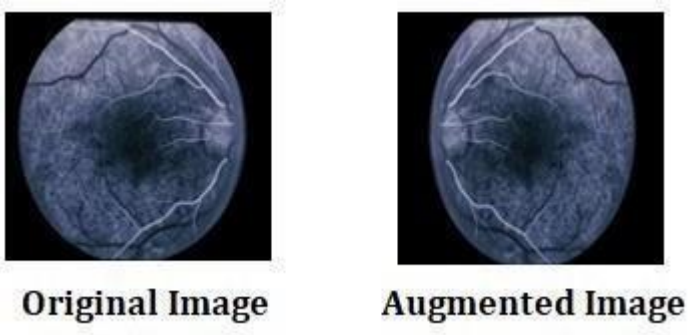


Fig 5.3: AUGMENTED IMAGES

Another augmentation method Fig 5.3 converts the unique image to grayscale, changing its hues to grayscale assessment. Flipping the original photograph changes its direction, at the same time as brightening enhances the picture,,s brightness.

5.2 FEATURE EXTRACTION

Feature extraction in retinal disease classification involves identifying significant patterns and features from fundus images to improve classification accuracy. In this study, ResNet101 is efficiency by preserving critical features across layers, mitigating information loss. Advanced image preprocessing techniques, including contrast enhancement and normalization, further refine extracted features. These deep learning-driven feature extraction methods improve the model,,s ability to differentiate between 32 retinal disease classes in the EyeNet dataset, ensuring robust classification performance.

5.3 MODEL BUILDING

Model building in the context of deep learning refers to the process of designing and constructing neural network architectures to solve specific tasks such as classification, regression, or generation.

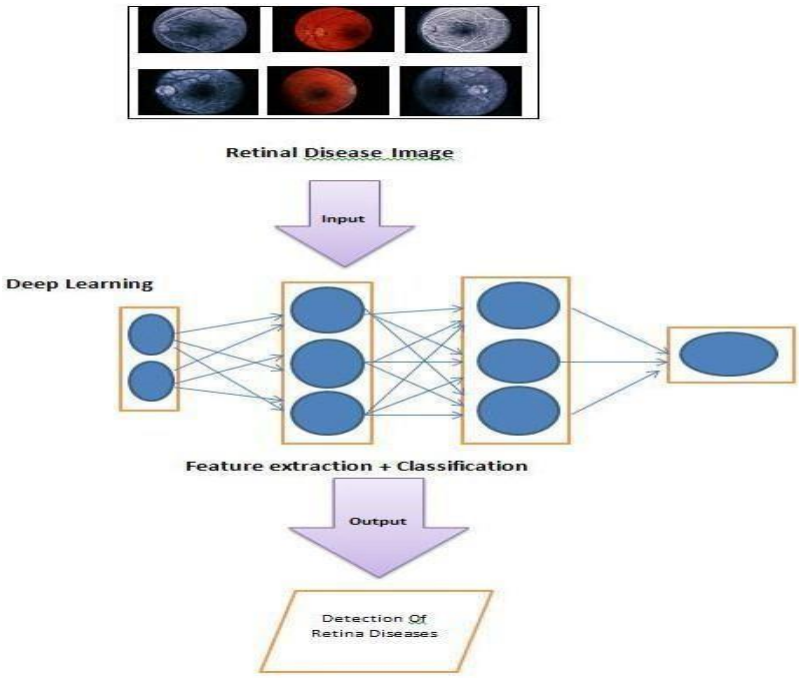


Fig 5.4 : PROPOSED MODEL

Deep learning models typically consist of multiple layers of neurons organized in a hierarchical fashion, enabling the model to learn intricate patterns and representations from the data.

Layers in Convolutional Neural Networks Below are the Layers of convolutional neural networks:

- **Image Input Layer:** The input layer gives inputs (mostly images), and normalization is carried out. Input size has to be mentioned here.
- **Convolutional Layer:** Convolution is performed in this layer. The image is divided into perceptrons (algorithm); local fields are created, leading to the compression of perceptrons to feature maps as a matrix with size m x n.
- **Non-Linearity Layer:** Here feature maps are taken as input, and activation maps are given as output with the help of the activation function. The activation function is generally implemented assigmoid or hyperbolic tangent functions.
- **Rectification Layer:** The crucial component of CNN, this layer does the training faster without reducing accuracy. It performs element-wise absolute value operation on activation maps.
- **Rectified Linear Units (ReLU):** ReLU combines non-linear and rectification layers on CNN. This does the threshold operation where negative values are converted to zero. However, ReLU does not change the size of the input.
- **Pooling Layer:** The pooling layer is also called the down sampling layer, as this is responsible for reducing the size of activation maps. A filter and stride of the same length are applied to the input volume. This layer ignores less significant data; hence image recognition is done in a smaller representation. This layer reduces overfitting. Since the amount of parameters is reduced using the pooling layer, the cost is also reduced. The input is divided into rectangular pooling regions, and either maximum or average is calculated, which returns maximum or average consequently. Max Pooling is a popular one.
- **Dropout Layer:** This layer randomly sets the input layer to zero with a given probability. More results in different elements are dropped after this operation.

This layer also helps to reduce overfitting. It makes the network to be redundant. No learning happens in this layer. This operation is carried out only during training.

- **Fully Connected Layer**: Activation maps, which are the output of previous layers, is turned into a class probability distribution in this layer. FC layer multiplies the input by a weight matrix and adds the bias vector.
- **Output Layer:** FC layer is followed by Soft max and classification layers. The Soft max function is applied to the input. The classification layer computes the cross entropy and loss function for classification problems.
- **Regression Layer**: Half the mean squared error is computed in this layer. This layer should follow the FC layer.

5.4 MODULES

1. Preprocessing Module:

This module is responsible for cleaning and preparing the retinal image data before feeding it into the deep learning model. It includes:

- **Data Cleaning**: Handling missing or corrupted images by either removing them or applying imputation techniques to maintain dataset integrity.
- **Resizing and Normalization**: Resizing retinal images to a uniform resolution and normalizing pixel values to a standard range (e.g., [0, 1]) to ensure consistency and improve model convergence.
- **Data Augmentation**: Applying transformations like flipping, zooming, and brightness adjustment to increase dataset variability and prevent overfitting. This is implemented using Keras' ImageDataGenerator.
- **Dataset Splitting**: Dividing the dataset into training (70%), validation (10%), and testing (20%) subsets to ensure robust model evaluation.

2. System Module:

This module handles the training, evaluation, and deployment of the ResNet-101 model for retinal disease classification. It includes:

- **Model Training & Development**: The ResNet-101 model is trained on the preprocessed retinal images to automatically extract hierarchical features, starting from low-level patterns (e.g., edges) to high-level, disease-specific patterns. The model is trained using the Adam optimizer with an adaptive learning rate to ensure efficient convergence.
- **Feature Extraction**: The ResNet-101 model automatically extracts features from the retinal images, eliminating the need for manual feature engineering. Early layers capture low-level features, while deeper layers identify disease-specific patterns.
- **Prediction & Classification**: The trained ResNet-101 model classifies retinal images into one of the 32 disease categories using a softmax activation function in the final layer. The model outputs probability scores for each disease class, and the class with the highest probability is selected as the final prediction.
- **Performance Evaluation**: The model,,s performance is evaluated using metrics like accuracy, precision, recall, and F1-score**.**A confusion matrix is used to visualize the model,,s classification results, providing insights into its strengths and weaknesses across different disease classes.
- **Optimization & Fine-Tuning**: Hyperparameter tuning (e.g., adjusting learning rate, batch size) is applied to improve model accuracy. The final optimized ResNet-101 model is saved in the **.**h5 file format for deployment.

3. User Interface (UI) Module:

This module provides an interactive interface for users to input retinal images, view predictions, and analyze results. It includes:

- **Data Input Forms**: Allows users to upload retinal images for classification.
- **Prediction Display**: Shows the predicted disease class along with confidence scores.
- **Analytics Dashboard**: Provides insights into the model,,s performance using visualizations like accuracy curves, loss curves, and confusion matrices.
- **Model Comparison & Reports**: Displays evaluation results of different models (e.g., ResNet-101 vs. traditional models) for better decision-making.

6. IMPLEMENTATION

6.1 MODEL IMPLEMENTATION

The Python code is a graphical user interface (GUI) application using the Tkinter library for intrusion detection in Software-Defined Networking (SDN) environments. The application allows users to perform the following tasks**:**

App.py

#app.py is a Python script, commonly used as the main entry point for web applications built with Flask or other frameworks. It typically handles routes, API endpoints, and application logic, making it the core of the project.

```
import tensorflow as tf
import numpy as np
import cv2
import os
from flask import Flask, request, render_template, jsonify

# Suppress TensorFlow logging messages
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

# Initialize Flask application
app = Flask(__name__)

# ========== MODEL CONFIGURATION ==========
INPUT_SIZE = (100, 100)        # Must match training dimensions
MODEL_PATH = 'retinal_disease_resnet101_32class.h5' # Trained model file
CLASS_COUNT = 32               # Number of disease classes
MIN_CONFIDENCE = 0.03          # 3% minimum confidence threshold

# ========== LOAD AND VERIFY MODEL ==========
try:
    model = tf.keras.models.load_model(MODEL_PATH)
    print(f"Model loaded successfully. Input shape: {model.input_shape}")
except Exception as e:
    print(f"Error loading model: {str(e)}")
```

```
    exit(1)

# Compile model (required for some TensorFlow versions)
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# ========== DISEASE LABEL MAPPING ==========
LABEL_MAP = {
    0: 'AMN Macular Neuroretinopathy',
    1: "Adult Coats' Disease",
    2: 'Adult Foveomacular Dystrophy Pattern',
    3: 'Age-Related Macular Degeneration With Pattern Dystrophy Appearance',
    4: 'Antiphospholipid Antibody Syndrome',
    5: "Behcet's",
    6: 'Bilateral Macular Dystrophy',
    7: "Bull's Eye Maculopathy Chloroquine",
    8: 'CMV Chorioretinitis',
    9: 'Central Serous Chorioretinopathy',
    10: 'Choroidal Nevus',
    11: 'Cone - Rod Dystrophy',
    12: 'Congenital Syphillis',
    13: 'Diabetic Maculopathy Multiple Myeloma with Retinal Detachment',
    14: 'Giant Retinal Tear',
    15: 'Juxtafoveal Telangiectasis DM Diabetes',
    16: "Leber's Stellate Maculopathy",
    17: 'Macular Dystrophy',
    18: 'Multifocal Exudative Detachments Due to VKH',
    19: 'Myelinated Nerve Fibers',
    20: 'North Carolina Dystrophy',
    21: 'Optic Disc Drusen',
    22: 'Pattern Dystrophy Simulating Fundus Flavimaculatus',
    23: 'Reticular Pattern Dystrophy',
    24: 'Retinal Folds Following Retinal Reattachment Surgery',
```

```
    25: 'Retrohyaloid Hemorrhage',
    26: 'Roth Spot',
    27: 'Self-Applied Retinal Detachment',
    28: 'Solar Retinopathy Familial',
    29: "Susac's Syndrome",
    30: "Terson's Syndrome",
    31: 'Wyburn-Mason Syndrome'
}

# ========== IMAGE PROCESSING ==========
def preprocess_image(image_stream):
    """Convert uploaded file to model-ready input"""
    try:
        img = cv2.imdecode(np.frombuffer(image_stream.read(), np.uint8),
                           cv2.IMREAD_COLOR)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, INPUT_SIZE)
        img = img.astype(np.float32) / 255.0
        return np.expand_dims(img, axis=0)
    except Exception as e:
        app.logger.error(f"Image processing failed: {str(e)}")
        return None

# ========== FLASK ROUTES ==========
@app.route('/')
def home():
    return render_template('home.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/evaluation')
def evaluation():
```

```
    return render_template('evaluation.html')

@app.route('/flowchart')
def flowchart():
    return render_template('flowchart.html')

@app.route('/prediction')
def prediction():
    return render_template('prediction.html')

@app.route('/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        return jsonify({'error': 'No file uploaded'}), 400

    try:
        file = request.files['file']
        if file.filename == '':
            return jsonify({'error': 'Empty file submitted'}), 400

        processed_img = preprocess_image(file)
        if processed_img is None:
            return jsonify({'error': 'Invalid image file'}), 400

        predictions = model.predict(processed_img)[0]

        if len(predictions) != CLASS_COUNT:
            return jsonify({'error': 'Model output mismatch'}), 500

        max_confidence = np.max(predictions)
        if max_confidence < MIN_CONFIDENCE:
            return jsonify({
                'error': 'Invalid retinal image',
                'message': 'Low confidence prediction (minimum 3% required)',
```

```
                'max_confidence': f"{max_confidence*100:.2f}%",
                'confidence_scale': round(float(max_confidence)*10, 1)
            }), 400

        top_indices = np.argsort(predictions)[-3:][::-1]

        results = []
        for idx in top_indices:
            confidence = float(predictions[idx])
            results.append({
                'class': LABEL_MAP[int(idx)],
                'confidence': confidence,
                'confidence_percent': f"{confidence*100:.2f}%",
                'confidence_scale_1_10': round(confidence * 10, 1)
            })

        return jsonify({
            'predictions': results,
            'status': 'Valid retinal image',
            'confidence_check': f"Minimum confidence threshold
({MIN_CONFIDENCE*100}%) passed"
        })

    except Exception as e:
        app.logger.error(f"Prediction failed: {str(e)}")
        return jsonify({'error': 'Processing error'}), 500

if__name__== '__main__':
    app.run(host='0.0.0.0', port=5000, debug=False)
```

6.2 SOURCE CODE

```
#import necessary libraries

import os
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split


# Data Augmentation and Preprocessing Function
def augment_images(input_path, img_size=(100, 100)):
    # Define an ImageDataGenerator for augmentation
    datagen = ImageDataGenerator(
        rescale=1./255,           # Normalize pixel values (0-255 → 0-1)
        zoom_range=0.2,           # Randomly zoom images (up to 20%)
        horizontal_flip=True,     # Randomly flip images horizontally
        vertical_flip=True,       # Randomly flip images vertically
        rotation_range=20,        # Randomly rotate images (up to ±20 degrees)
        width_shift_range=0.2,    # Random horizontal shift (20% of width)
        height_shift_range=0.2,   # Random vertical shift (20% of height)
        shear_range=0.2,          # Shear transformations (20%)
        fill_mode='nearest'       # Fill missing pixels using nearest neighbors
    )

    images = []  # List to store image data
    labels = []  # List to store corresponding labels
    label_map = {} # Dictionary to map class names to numerical labels

    # Loop through each disease category folder
    for idx, folder_name in enumerate(sorted(os.listdir(input_path))):
        folder_path = os.path.join(input_path, folder_name)
```

```
        if not os.path.isdir(folder_path):  # Skip if it's not a directory
            continue

        label_map[folder_name] = idx # Assign numerical labels to categories

        # Loop through images in each folder
        for file_name in sorted(os.listdir(folder_path)):
            file_path = os.path.join(folder_path, file_name)
            file_extension = os.path.splitext(file_name)[1].lower()

            # Ensure the file is an image
            if file_extension in {'.jpg', '.jpeg', '.png', '.bmp', '.tiff'}:
                image = cv2.imread(file_path)  # Read image using OpenCV
                image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
# Convert BGR to RGB
                image = cv2.resize(image, img_size)  # Resize image to target size
                images.append(image) # Add processed image to the list
                labels.append(idx) # Assign numerical label to the image

    # Convert lists to NumPy arrays
    images = np.array(images, dtype='float32')
    labels = np.array(labels, dtype='int')

    # One-hot encode the labels (needed for multi-class classification)
    labels = tf.keras.utils.to_categorical(labels, num_classes=len(label_map))

    return images, labels, label_map
# Return processed images, labels, and category mapping


# Load  Dataset
input_dataset_path = '/content/drive/MyDrive/project41/output_augmentation'
# Path to the dataset (update as needed)
train_images, train_labels, label_map = augment_images(input_dataset_path)
```

 # Apply data augmentation and load images

 # Split the data into training and testing sets

train_images, test_images, train_labels, test_labels = train_test_split(train_images, train_labels, test_size=0.2, random_state=42)

#Model Definition

 # Import necessary modules from TensorFlow

from tensorflow.keras.applications import ResNet101 **# Pre-trained ResNet-101 model**

from tensorflow.keras.models import Model **# To define a custom model**

from tensorflow.keras.layers import Dense, Flatten, Dropout, BatchNormalization, GlobalAveragePooling2D

 # Load the ResNet101 model with pre-trained weights from ImageNet

base_model = ResNet101(weights='imagenet', include_top=False, input_shape=(100, 100, 3))

 # Get the output from the base model

x = base_model.output

 # Apply Global Average Pooling to reduce feature dimensions

x = GlobalAveragePooling2D()(x) **# Converts feature maps into a vector**

 # Fully connected (Dense) layers with ReLU activation

x = Dense(256, activation='relu')(x) **# 256 neurons for feature extraction**

x = BatchNormalization()(x) **# Normalizes activations to stabilize training**

x = Dropout(0.5)(x) **# Drops 50% of neurons to prevent overfitting**

x = Dense(128, activation='relu')(x) **# Another fully connected layer**

x = BatchNormalization()(x)

x = Dropout(0.5)(x)

predictions = Dense(len(label_map), activation='softmax')(x) **# Assuming 32 classes for the EyeNet dataset**

```
# Create the full model
model = Model(inputs=base_model.input, outputs=predictions)

#Compile and Summary
from tensorflow.keras.optimizers import Adam

# Freeze the layers of the base model
for layer in base_model.layers:
    layer.trainable = False

# Compile the model with Adam optimizer
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Summary of the model
model.summary()

#Model Training
# Fit the model
history = model.fit(train_images, train_labels, epochs=32, batch_size=32,
validation_split=0.2)

# Access the history dictionary
print(history.history.keys())

# Plot training & validation accuracy values
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
```

```
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

# Compile the model with Adam optimizer
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
# Assuming your model is already trained and test_images, test_labels are available
# Evaluate the model on test data
loss, accuracy = model.evaluate(test_images, test_labels)


from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import numpy as np

# Generate predictions for the test set
y_pred = model.predict(test_images)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true_classes = np.argmax(test_labels, axis=1)
```

```
# Compute the confusion matrix
cm = confusion_matrix(y_true_classes, y_pred_classes)
# Ensure label_map keys are in list format
label_names = list(label_map.keys())
# Check if the number of label names matches the number of classes
if len(label_names) != len(np.unique(y_true_classes)):
  print("Warning: Number of label names does not match number of classes.
Rectifying...")
  # Adjust label_names to match the number of classes
label_names = label_names[:len(np.unique(y_true_classes))]
# Plot the confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=label_names)
disp.plot(cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.show()

# Generate the classification report
report = classification_report(y_true_classes, y_pred_classes,
target_names=label_names)
# Print the classification report
print("Classification Report:")
print(report)
```

Flask code for Frontend

home.html

‘‘‘This is the Home screen that which have the home, about, predictions, model evaluation and the flowchart‘‘‘‘‘‘

```
<!DOCTYPE html>
<html>
<head>
   <style>
      body {
         margin: 0;
```

```
    padding: 0;
    background-size: cover;
    background-position: center;
    font-family: Arial, sans-serif;
}

.navbar {
    display: flex;
    justify-content: space-around;
    align-items: center;
    background-color: #575757;
    color: white;
    padding: 10px 20px;
}

.navbar a {
    text-decoration: none;
    color: white;
    padding: 8px 15px;
    border-radius: 4px;
}

.navbar a:hover {
    background-color: #575757;
}

.header {
    background-color: #B1F0F7;
    display: flex;
    align-items: center;
    justify-content: space-between;
    padding: 10px 20px;
}
```

```
        .logo {
            height: 100px;
            width: auto;
        }

        .title {
            text-align: center;
            font-size: 30px;
            color: #333333;
            font-weight: bold;
            flex-grow: 1;
            font-family: 'Times New Roman', Times, serif;

        }

        h1 {
            text-align: center;
            margin-top: 0;
        }
        .content img{
            height: 550px;
            width: 1519px;
        }
        p{
            font-family: 'Times New Roman', Times, serif;
            font-size: 20px;
            text-align: center;
            margin: 0px;
            font-weight: 500;
            background-color: #B1F0F7;

        }
    </style>
</head>
```

```
<body>
    <div class="header">
        <img src="{{ url_for('static', filename='logo.jpg') }}" alt="Project Logo"
class="logo">
        <div class="title"> OPTIMIZED DEEP LEARNING <br>FOR MULTI-CLASS
RETINAL DISEASE CLASSIFICATION USING RESNET-101

        </div>
    </div>
    <p>Team Members:  1.Kattiri Santhoshkumar   2.Talari Babu  
3.Madana Venkata Thirumala</p>

    <div class="navbar">
        <a href="/">Home</a>
        <a href="/about">About Project</a>
        <a href="/prediction">Predictions</a>
        <a href="/evaluation">Model Evaluation</a>
        <a href="/flowchart">Flowchart</a>
    </div>

    <div class="content">
        <img src="{{ url_for('static', filename='s-1.webp') }}" alt="Project Logo"
class="logo">
    </div>
</body>
</html>
```

about.html

‘‘‘The about screen will contain the project description that have complete project and the team members details’’’

```
<!DOCTYPE html>
<html>
<head>
    <title>About Project</title>
```

```
<style>
    body {
        font-family: 'Times New Roman', Times, serif;
        margin: 0;
        padding: 0;
        background-color: #f4f4f4;
    }

    .navbar {
        display: flex;
        justify-content: space-around;
        align-items: center;
        background-color: #333;
        color: white;
        padding: 10px 20px;
    }

    .navbar a {
        text-decoration: none;
        color: white;
        padding: 8px 15px;
        border-radius: 4px;
    }

    .navbar a:hover {
        background-color: #575757;
    }

    .content {
        text-align: center;
        padding: 50px;
    }

    h1 {
```

```
            color: #333;
        }
        .about-section h2, .team-section h2, .methods-section h2 {
            margin-bottom: 10px;
            color: #004d99;
        }

        ul {
            list-style: square;
            margin-left: 20px;
        }

        ul li {
            margin-bottom: 5px;
        }

        /* Footer Styling */
        .main-footer {
            background-color: #004d99;
            color: #fff;
            text-align: center;
            padding: 15px 0;
            margin-top: 20px;
            font-size: 0.9em;
        }
    </style>
</head>
<body>
    <div class="navbar">
        <a href="/">Home</a>
        <a href="/about">About Project</a>
        <a href="/prediction">Predictions</a>
        <a href="/evaluation">Model Evaluation</a>
        <a href="/flowchart">Flowchart</a>
```

```
</div>

<section class="about-section">
    <h2>About the Project</h2>
    <p>
        The project <strong>"Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101"</strong>
        is an initiative aimed at leveraging advanced deep learning methodologies to enhance the classification of retinal diseases.
        Our goal is to provide accurate and efficient diagnostic insights to ophthalmologists, healthcare professionals, and researchers by utilizing deep learning techniques.
    </p>
    <p>
        Through the application of medical image analysis, this project identifies patterns in retinal scans to assist in the detection and classification of retinal diseases.
        By early detection of challenges, it ensures timely intervention to improve outcomes.
    </p>
    <p>
        The study incorporates advanced deep learning algorithms, specifically ResNet-101, to enhance the accuracy of multi-class retinal disease classification.
        The model is evaluated using precision, recall and accuracy metrics, ensuring robust and reliable performance in diagnosing 32 distinct retinal diseases from the EyeNet dataset.
    </p>
    <p>
        Beyond classification, this project contributes to advancing AI-driven ophthalmic diagnostics by improving the early detection and analysis of retinal diseases.
        The results aim to enhance automated disease diagnosis, reduce dependence on manual assessments, and support ophthalmologists in making accurate and timely clinical decisions, ultimately leading to better patient care and improved accessibility to retinal healthcare solutions.
```

```
    </p>
  </section>
  <section class="team-section">
    <h2>Meet the Team</h2>
    <ul>
      <li><strong>Kattiri Santhoshkumar</strong> - Team Lead & Feature Engineering & Frontend Development</li>
      <li><strong>Talari Babu</strong> - Data Preprocessing</li>
      <li><strong>Madana Venkata Thirumala</strong> - Evaluation & Performance Metrics</li>
    </ul>
  </section>
  <section class="methods-section">
    <h2>Key Methods</h2>
    <p>
      This project employs a range of advanced methodologies, including:
      <ul>
        <li><strong>Deep Learning Architecture:</strong> Implementing ResNet-101, a powerful convolutional neural network for multi-class retinal disease classification.</li>
        <li><strong>Metrics Evaluation:</strong> Assessing model performance using precision, recall and accuracy to ensure robust and reliable predictions.</li>
        <li><strong>Adaptive Optimization:</strong> Training the model with the Adam optimizer (learning rate = 0.001) for improved convergence and efficiency.</li>
      </ul>
    </p>
  </section>
  <footer class="main-footer">
    <p>© 2024 Narasaraopeta Engin.eering College. All Rights Reserved.</p>
  </footer>
</body>
</html>
```

Prediction.html

‘‘‘This is the prediction screen and this contains the all the input fields and we have to enter the input in the input fields’’’

```
<!DOCTYPE html>
<html>
<head>
   <title>Retinal Disease Detection</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         max-width: 800px;
         margin: 0 auto;
         padding: 0;
         background-color: #f4f4f4;
         margin:0;
      }

      /* Navbar Styling */
      .navbar {
         display: flex;
         justify-content: space-around;
         align-items: center;
         background-color: rgba(0, 0, 0, 0.8);
         color: white;
         padding: 10px 20px;
         position: fixed;
         top: 0;
         width: 100%;
         z-index: 1000;
         box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3);
      }

      .navbar a {
         text-decoration: none;
```

```
        color: white;
        padding: 8px 15px;
        border-radius: 4px;
    }

    .navbar a:hover {
        background-color: #575757;
    }

    /* Centered Form */
    .form-container {
        display: flex;
        flex-direction: column;
        align-items: center;
        margin-top: 80px;
    }

    #uploadForm {
        text-align: center;
        background: white;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
        width: 100%;
        max-width: 600px; /* Increased width to accommodate image and file input
side by side */
    }

    .upload-section {
        display: flex;
        align-items: center;
        gap: 20px; /* Space between file input and image preview */
        margin-bottom: 20px; /* Space between upload section and button */
    }
```

```
input[type="file"] {
    margin-bottom: 0; /* Remove default margin */
}

button {
    background-color: #4CAF50;
    color: white;
    border: none;
    padding: 10px 20px;
    cursor: pointer;
    border-radius: 5px;
    width: 100%; /* Make button full width */
}

button:hover {
    background-color: #45a049;
}

/* Image Preview Styling */
#imagePreview {
    max-width: 200px; /* Limit image preview size */
    height: auto;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
    display: none; /* Hidden by default */
}

/* Prediction Styling */
.prediction {
    margin: 10px 0;
    padding: 2px;
    border: 1px solid #ddd;
    border-radius: 5px;
```

```
            background-color: white;
            align-content: center;
        }

        .confidence-bar {
            height: 10px;
            background: #eee;
            margin: 5px 0;
            border-radius: 5px;
            overflow: hidden;
        }

        .confidence-fill {
            height: 100%;
            background: #4CAF50;
        }

        #results {
            margin-top: 10px;
        }
        h1{
            text-align: center;
            margin-top: 100px;
        }
    </style>
</head>
<body>

    <div class="navbar">
        <a href="/">Home</a>
        <a href="/about">About Project</a>
        <a href="/prediction">Predictions</a>
        <a href="/evaluation">Model Evaluation</a>
        <a href="/flowchart">Flowchart</a>
```

```
</div>
<div>
<h1 style="text-align: center; margin-top: 100px;">Multi Class Retinal Disease Classifier</h1>
<div class="form-container">

<form id="uploadForm">
<div class="upload-section">
<input type="file" name="file" id="fileInput" accept="image/*" required>
<img id="imagePreview" src="#" alt="Image Preview">
</div>
<button type="submit">Predict</button>
</form>
</div>
</div>
<div id="results"></div>

<script>
// Display the uploaded image preview
document.getElementById('fileInput').onchange = function (event) {
const file = event.target.files[0];
if (file) {
const reader = new FileReader();
reader.onload = function (e) {
const imagePreview = document.getElementById('imagePreview');
imagePreview.src = e.target.result;
imagePreview.style.display = 'block';
};
reader.readAsDataURL(file);
}
};

// Handle form submission
document.getElementById('uploadForm').onsubmit = async (e) => {
```

```
e.preventDefault();
const fileInput = document.getElementById('fileInput');
const file = fileInput.files[0];

// Validate if file is an image
if (!file) {
    alert("Please select an image file.");
    return;
}

const validImageTypes = ["image/jpeg", "image/png", "image/gif"];
if (!validImageTypes.includes(file.type)) {
    alert("Invalid file type. Please upload an image (JPG, PNG, or GIF).");
    return;
}

const formData = new FormData();
formData.append('file', file);

try {
    const response = await fetch('/predict', {
        method: 'POST',
        body: formData
    });

    const data = await response.json();

    if (data.error) {
        alert(`Error: ${data.error}`);
        return;
    }

    let html = '<h2 style="text-align:center;">Analysis Results:</h2>';
    data.predictions.forEach((pred, index) => {
```

```
                html += `
                <div class="prediction">
                    <h3>#${index + 1}: ${pred.class}</h3>
                    <div class="confidence-bar">
                        <div class="confidence-fill"
                            style="width: ${parseFloat(pred.confidence_percent)}%"></div>
                    </div>
                    <p>Confidence: ${pred.confidence_percent}</p>
                </div>`;
            });

            document.getElementById('results').innerHTML = html;

        } catch (error) {
            alert('Error communicating with server');
        }
    };
  </script>
</body>
</html>
```

7. TESTING

In software development, effective testing is crucial to ensure code correctness, reliability, and performance. The primary types of testing include unit testing, integration testing, and system testing, each serving distinct purposes and aligning with different stages of the development cycle.

1. Unit Testing

- **Purpose**: To test individual components or functions of the system, such as data preprocessing functions and the ResNet-101 model.
- **Testing Areas**:
  - Data preprocessing functions (image resizing, normalization, augmentation).
  - ResNet-101 model architecture.
  - Model training and validation accuracy.
  - Checking for overfitting and underfitting.
- **Tools Used**: Python's unittest, PyTest, or TensorFlow/Keras testing utilities.

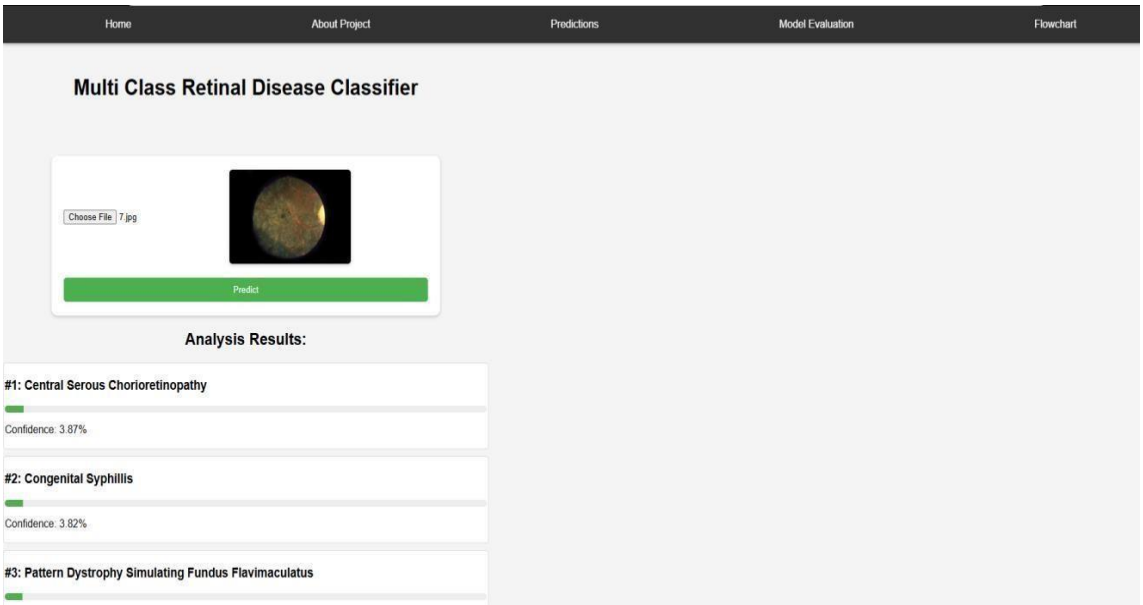


Fig 7.1: TEST CASE - 1

- In the Fig 7.1 inputs are calculated with confidence scores that is Central Serous Chorioretinopathy and Congenital Syphilis , according to the model the result is predicted.

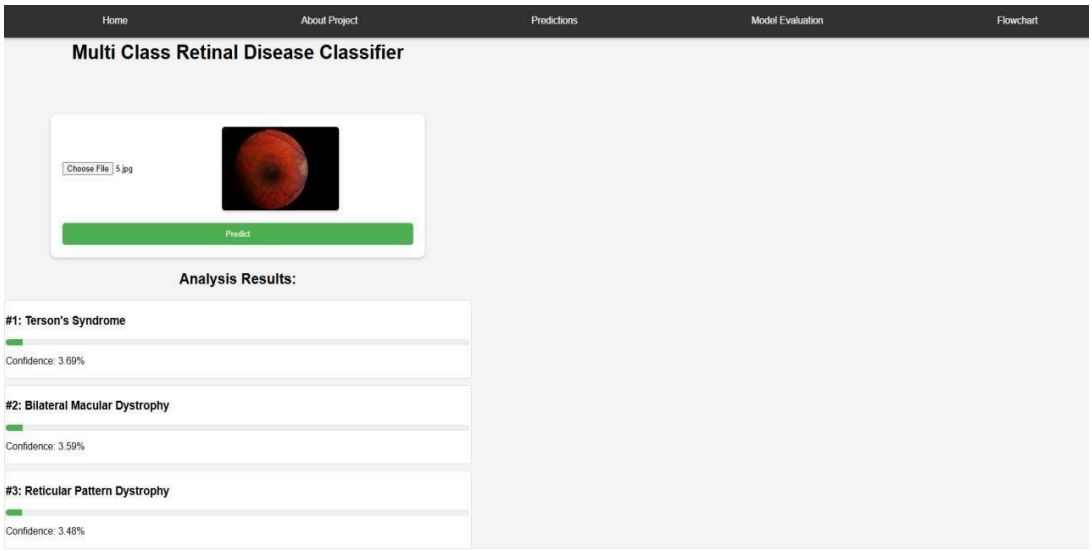


Fig 7.2: TEST CASE-2

- In Fig 7.2 inputs are calculated with confidence scores that is Terson,,s Syndrome and Bilateral Macular Dystrophy, according to the model the result is predicted.

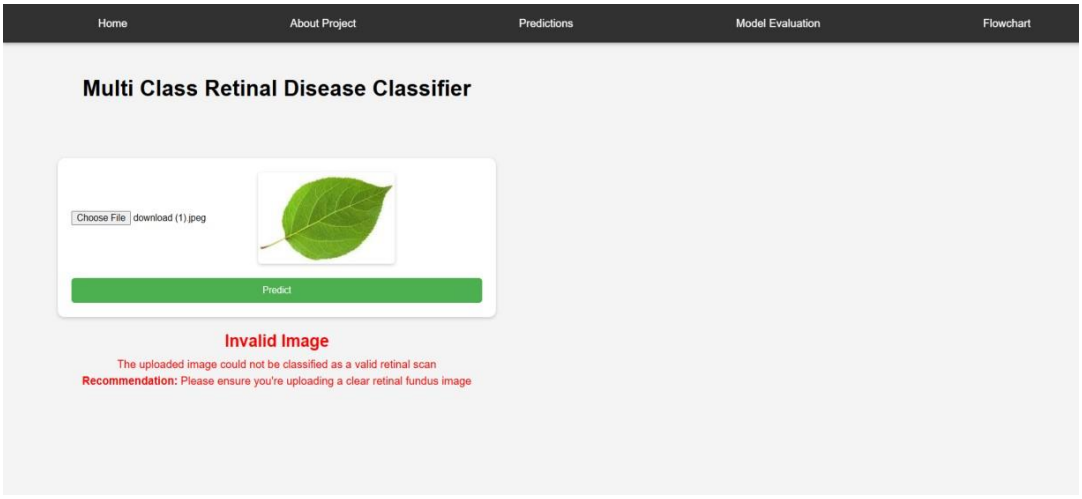


Fig 7.3: TEST CASE – 3

- In Fig 7.3 the uploaded image is evaluated by the model, which determines that it is not a valid retinal image. Consequently, the system returns an "Invalid Image" error message, instructing the user to upload a proper retinal fundus image for classification.

2. System Testing

- **Purpose:** To test the complete Retinal Disease Classification System as a whole to ensure smooth functionality.
- **Testing Areas**:
  - Integration of the Preprocessing Module**,** ResNet-101 Model**,** and User Interface Module**.**
  - Accurate prediction of retinal diseases from input images.
  - Checking performance metrics (Accuracy, Precision, Recall, F1-score).
  - User interaction and result visualization in the GUI.
- **Tools Used**: Flask Testing Environment, Postman API Testing, Selenium for UI testing.

3. Integration Testing

- **Purpose**: To check the data flow between modules and the interaction between the ResNet-101 model and the database.
- **Testing Areas**: Data flow from the Preprocessing Module to the ResNet-101 Model.Handling of real-time user input through the User Interface Module. Accuracy of final predictions and feedback loop.
- **Tools Used**: PyTest, Integration Testing Framework.

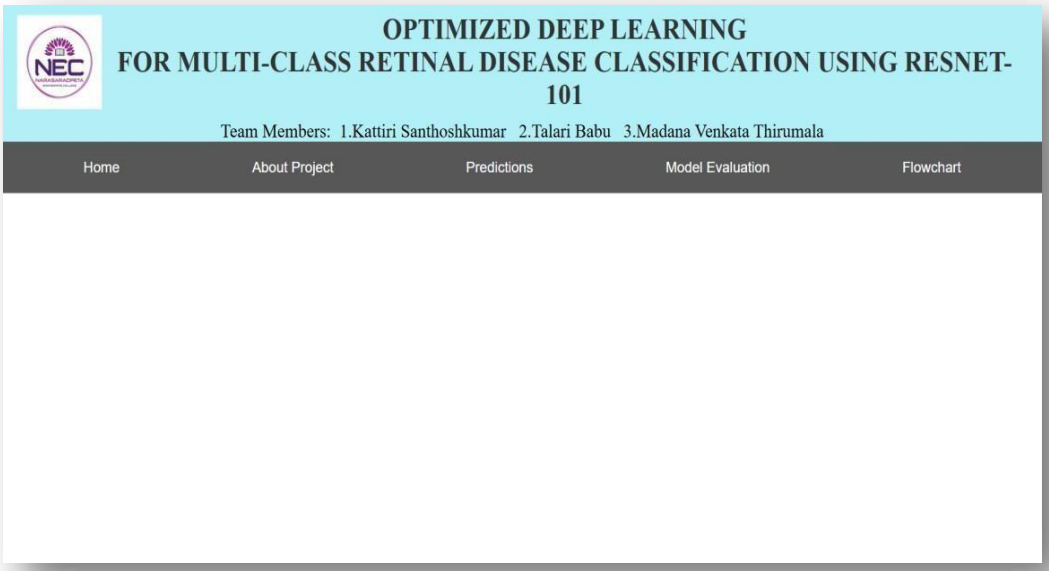


Fig 7.4: Home Screen

- This is the Home screen that which have the home, about, predictions, model evaluation and the flowchart.

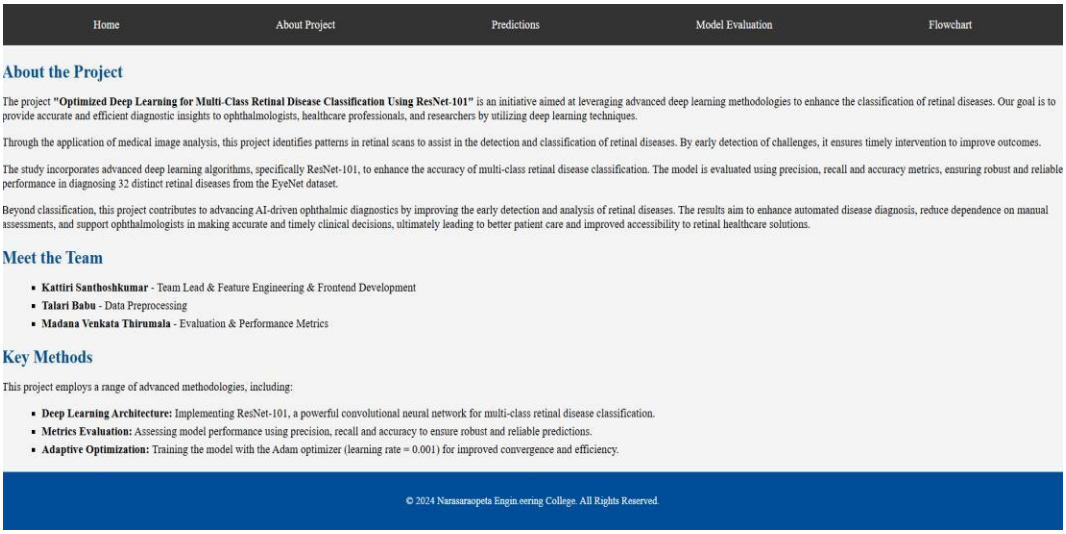


Fig 7.5: About Screen

- The about screen will contain the project description that have complete project and the team members details

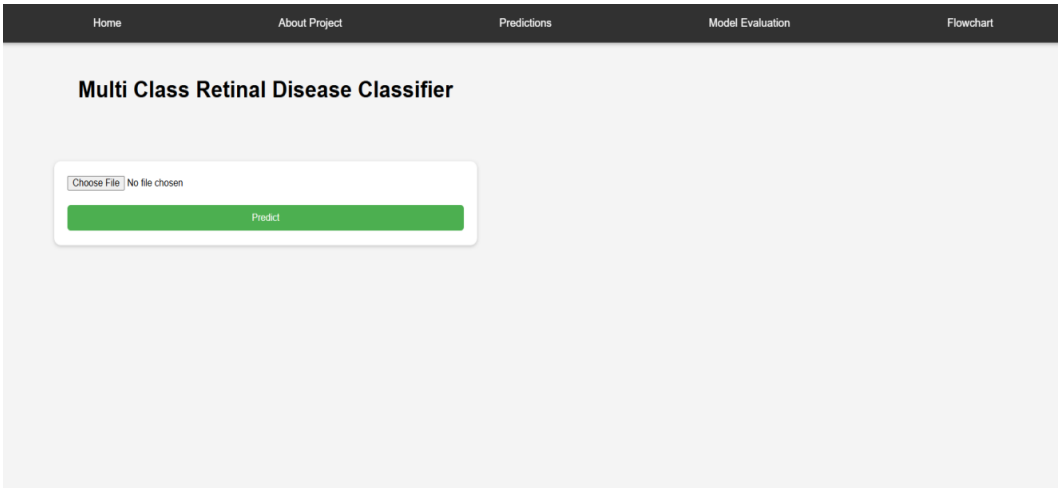


Fig 7.6: Prediction screen

- This is the prediction screen and this contains the input field and we have to upload the input i.e images in the input fields.

8. RESULT ANALYSIS

The study evaluates the performance of ResNet-101 in multi-class retinal disease classification, measuring its accuracy using key evaluation metrics such as precision, recall, F1-score, and accuracy. The findings demonstrate that the model achieves 98.75% validation accuracy, showcasing its effectiveness in distinguishing between 32 retinal disease classes.

1. ResNet-101 Performance

- Precision: 98%
- Recall: 96%
- F1-Score: High performance across all classes

lassification Report:

	precision	recall	f1-score	support
.ipynb_checkpoints	1.00	1.00	1.00	8
AMN Macular Neuroretinopathy	1.00	1.00	1.00	12
Adult Coats' Disease	1.00	1.00	1.00	10
Adult Foveomacular Dystrophy Pattern	1.00	1.00	1.00	11
ge-Related Macular Degeneration With Pattern Dystrophy Appearance	0.90	1.00	0.95	9
Antiphospholipid Antibody Syndrome	1.00	1.00	1.00	12
Behcet's	1.00	1.00	1.00	11
Bilateral Macular Dystrophy	1.00	0.94	0.97	16
Bull's Eye Maculopathy Chloroquine	1.00	1.00	1.00	9
CMV Chorioretinitis	1.00	1.00	1.00	10
Central Serous Chorioretinopathy	1.00	1.00	1.00	14
Choroidal Nevus	1.00	1.00	1.00	14
Cone - Rod Dystrophy	1.00	1.00	1.00	11
Congenital Syphillis	1.00	1.00	1.00	7
Diabetic Maculopathy Multiple Myeloma with Retinal Detachment	1.00	1.00	1.00	9
Giant Retinal Tear	1.00	1.00	1.00	8
Juxtafoveal Telangiectasis DM Diabetes	0.91	1.00	0.95	10
Leber's Stellate Maculopathy	1.00	1.00	1.00	8
Macular Dystrophy	1.00	1.00	1.00	10
Multifocal Exudative Detachments Due to VKH	0.88	1.00	0.93	7
Myelinated Nerve Fibers	1.00	1.00	1.00	10
North Carolina Dystrophy	1.00	1.00	1.00	11
Optic Disc Drusen	1.00	1.00	1.00	12
Pattern Dystrophy Simulating Fundus Flavimaculatus	1.00	1.00	1.00	11
Reticular Pattern Dystrophy	1.00	1.00	1.00	8
Retinal Folds Following Retinal Reattachment Surgery	1.00	1.00	1.00	9
Retrohyaloid Hemorrhage	1.00	1.00	1.00	8
Roth Spot	1.00	1.00	1.00	11
Self-Applied Retinal Detachment	1.00	0.86	0.92	7
Solar Retinopathy Familial	1.00	1.00	1.00	10
Susac's Syndrome	1.00	0.88	0.93	8
Terson's Syndrome	1.00	1.00	1.00	9
accuracy			0.99	320
macro avg	0.99	0.99	0.99	320
weighted avg	0.99	0.99	0.99	320

Fig 8.1: CLASSIFICATION REPORT

- In Fig 8.1 indicates that the model performs exceptionally well on the training data, achieving perfect accuracy for most classes.

2. **Training and Validation Metrics**

- Training accuracy stabilizes above 90% after 10 epochs. In Fig 8.2Validation loss decreases progressively, ensuring model generalization. The model demonstrates a low validation loss of 0.0498, indicating minimal prediction errors.

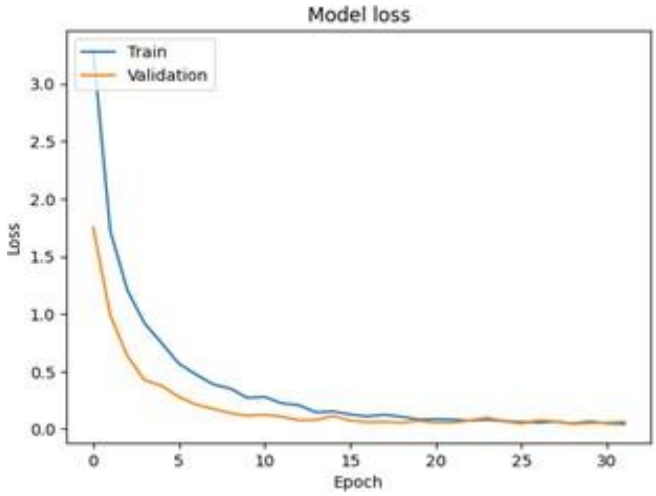


Fig 8.2: VALIDATION & TRAINING LOSS COMPARSION

3. **Confusion Matrix Analysis**

- The model correctly classifies most disease categories, with high precision and recall for complex retinal conditions.

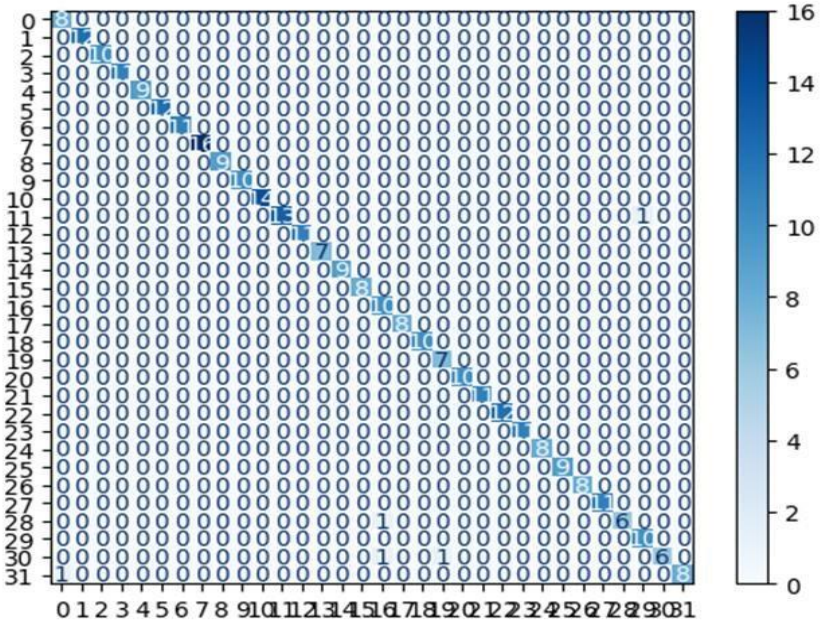


Fig 8.3: CONFUSION MATRIX

The Fig 8.3 helps to evaluate your model's performance by showing correct predictions (diagonal elements) and misclassifications (off-diagonal elements) across the 32 retinal disease classes. It provides insights into which diseases are frequently confused, helping identify areas for improvement. By analyzing the matrix, you can calculate key metrics like accuracy, precision, and recall, aligning your results with the paper's high performance (98.75% accuracy, 100% precision, 99% recall). This ensures your model effectively classifies retinal diseases and supports early diagnosis.

4. Graphical Representation

The Fig 8.4 shows accuracy curves for training and validation over 30 epochs. The curves indicate rapid improvement in early epochs, with accuracy stabilizing near optimal performance (close to 1.0 or 100%) within 15 epochs. This aligns with the paper's findings, where the ResNet-101 model achieved 98.75% validation accuracy and demonstrated effective convergence. The stabilization of both training and validation accuracy at high values confirms the model's robustness and suitability for real-world clinical applications in retinal disease classification. If the image reflects these trends, it effectively supports the paper's conclusions.

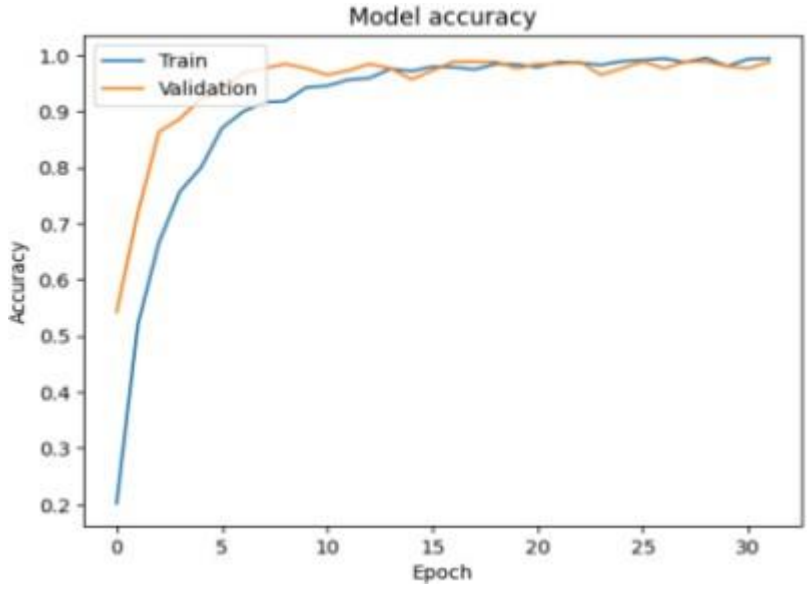


Fig 8.4: MODEL ACCURACY BY IMPLEMENTING 30 EPOCHS

9. CONCLUSION

This study presents a comprehensive approach to multi-class retinal disease classification using deep learning techniques, particularly ResNet-101. By leveraging the EyeNet dataset, which includes 32 retinal disease categories, the model achieves 98.75% accuracy, demonstrating its efficacy in automated retinal disease detection. The use of data augmentation techniques such as rescaling, flipping, and zooming has enhanced model generalization, ensuring robust classification across diverse retinal conditions. The findings highlight the importance of deep feature extraction using residual learning, which significantly improves diagnostic accuracy. The results emphasize the potential clinical impact of deep learning in ophthalmology. Early detection of retinal diseases can lead to timely intervention and better patient outcomes, reducing the risk of vision impairment. The research also underscores the advantages of ResNet-101 over traditional diagnostic methods, proving its ability to detect complex retinal abnormalities with high precision and recall.

10. FUTURE SCOPE

Future advancements can focus on expanding the dataset by incorporating higher- resolution retinal images and multi-modal imaging techniques, such as Optical Coherence Tomography (OCT) and fluorescein angiography, which provide detailed structural insights into the retina. By integrating these imaging modalities, the model's ability to detect subtle pathological changes can be significantly improved. Enhancing the model with self-supervised learning and attention mechanisms can further refine classification performance. Self-supervised learning allows the model to learn patterns from unlabelled data, reducing reliance on manually annotated datasets. Attention mechanisms, such as Vision Transformers (ViTs), can help focus on critical regions of the retinal image, improving disease localization and classification accuracy.

Additionally, integrating the system into real-time clinical workflows via a decision support system will enhance its usability for ophthalmologists and telemedicine applications. Deploying the model on edge devices like Raspberry Pi or mobile platforms will allow for point-of-care diagnostics, making retinal screening accessible in remote areas. Implementing explainable AI (XAI) techniques will improve the interpretability of model decisions, enabling clinicians to trust AI-based diagnoses. Addressing challenges such as data imbalance, generalizability, and ethical concerns will be crucial for widespread adoption in medical practice. Future research can explore federated learning, enabling hospitals to collaborate on training models while preserving patient privacy. The development of personalized AI-driven treatment recommendations based on disease severity and progression trends will further revolutionize retinal disease management.

11. REFERENCES

[1] Hasan, D. A., Zeebaree, S. R., Sadeeq, M. A., Shukur, H. M., Zebari, R. R., & Alkhayyat, A. H. (2021, April). Machine learning-based diabetic retinopathy early detection and classification systems-a survey. In 2021 1st Babylon International Conference on Information Technology and Science (BICITS) (pp. 16-21). IEEE.

[2] Flores, R., Carneiro, A., Vieira, M., Tenreiro, S., & Seabra, M. C. (2021). ˆ Age-related macular degeneration: pathophysiology, management, and future perspectives. Ophthalmologica, 244(6), 495-511.

[3] Samdal, K. (2007). Cost-Utility Analysis of replacing photodynamic therapy with verteprofin by anti-VEGF treatment with ranibizumab on patients with predominantly classic neovascular agerelated macular degeneration (Master,,s thesis).

[4] El-Baz, A., Beache, G. M., Gimel farb, G., Suzuki, K., Okada, K., Elnakib, A., ... Abdollahi, B. (2013). Computer-aided diagnosis systems for lung cancer: challenges and methodologies. International journal of biomedical imaging, 2013(1), 942353.

[5] Ganasegeran, K. (2021). Deep learning for disease prediction in public health. In Deep Learning for Biomedical Applications (pp. 157-180). CRC Press.

[6] Hussain, S., Mubeen, I., Ullah, N., Shah, S. S. U. D., Khan, B. A., Zahoor, M., ... & Sultan, M. A. (2022). Modern diagnostic imaging technique applications and risk factors in the medical field: a review. BioMed research international, 2022(1), 5164970.

[7] Anwar, S. M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., & Khan, M. K. (2018). Medical image analysis using convolutional neural networks: a review. Journal of medical systems, 42, 1-13.

[8] Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., ... & Fujimoto, J. G. (1991). Optical coherence tomography. science, 254(5035), 1178-1181.

[9] Maheshwari, S., Gupta, P. K., Sinha, R., & Rawat, P. (2020). Knowledge, attitude, and practice towards coronavirus disease 2019 (COVID-19) among medical students: A cross-sectional study. Journal of Acute Disease, 9(3), 100-104.

[10] Lakshminarayanan, V., Kheradfallah, H., Sarkar, A., & Jothi Balaji, J. (2021). Automated detection and diagnosis of diabetic retinopathy: A comprehensive survey. Journal of imaging, 7(9), 165. 47.

[11] Lin, C. L., & Wu, K. C. (2023). Development of revised ResNet-50 for diabetic retinopathy detection. BMC bioinformatics, 24(1), 157.

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[13] Chollet, F. (2021). Deep learning with Python. Simon and Schuster.

[14] Jais, I. K. M., Ismail, A. R., & Nisa, S. Q. (2019). Adam optimization algorithm for wide and deep neural network. Knowl. Eng. Data Sci., 2(1), 41-46.

[15] Li, B., Chen, H., Yu, W., Zhang, M., Lu, F., Ma, J., ... & Chen, Y. (2024). The performance of a deep learning system in assisting junior ophthalmologists in diagnosing 13 major fundus diseases: a prospective multi-center clinical trial. npj Digital Medicine, 7(1), 8.

Certificate-1

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE Women in Engineering We
IEEE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 3091

Kattiri Santhoshkumar

Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India

for presenting the research paper entitled "Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101" in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF

IEEE SICSR PUNE STUDENT BRANCH

IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

Certificate-2

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE Women in Engineering WIE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 3091

Talari Babu

Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India

for presenting the research paper entitled “Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101” in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF IEEE SICSR PUNE STUDENT BRANCH IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

Certificate-3

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE Women in Engineering WIE IEEE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 3091

Madana VenkataThirumala

Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India

for presenting the research paper entitled "Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101" in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF IEEE SICSR PUNE STUDENT BRANCH IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

Certificate-4

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE Women in Engineering We
IEEE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 3091

Dr.K.Suresh Babu

Associate Professor, Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India

for presenting the research paper entitled "Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101" in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF

IEEE SICSR PUNE STUDENT BRANCH

IEEE Education Society PUNE SECTION

Certificate-5

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE Women in Engineering
IEEE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 3091

Dodda Venkatareddy

Assistant Professor,Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India

for presenting the research paper entitled “Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101” in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF

IEEE SICSR PUNE STUDENT BRANCH

IEEE Education Society PUNE SECTION

Certificate-6

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE Women in Engineering
IEEE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 3091

G.Saranya

Assistant Professor , Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India

for presenting the research paper entitled “Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101” in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale
General Chair

Prof. Dr. Rajashree Jain
General Chair

Dr. Surekha Deshmukh
Honorary Chair

Prof. Dr. Jatinderkumar R. Saini
Honorary Chair

Our Partners

SSBF

IEEE
SICSR PUNE
STUDENT BRANCH

IEEE
Education Society
PUNE SECTION

Optimized Deep Learning for Multi-Class Retinal Disease Classification Using ResNet-101

1st Dr.K.Suresh Babu
Associate Professor, Department of CSE ,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India
sureshkunda546@gmail.com

2nd G Sarayana
Assistant Professor , Department od CSE ,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India
gaddamsaranya4@gmail.com

3rd Kattiri SanthoshKumar
Student, Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India
kattirisanthoshkumar7@gmail.com

4th Talari Babu
Student, Department of CSE,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India
babutalarissr@gmail.com

5th Madana Venkata Thirumala
Student, Department of CSE ,
Narasaraopeta Engineering College ,
Narasaraopeta-522601, India
venkatathirumala5@gmail.com

***Abstract*—Machine Learning (ML) approaches, such as Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Deep Learning and advanced architectures like AlexNet and ResNet, are at the leading edge of studies in the identification and type of crucial sicknesses. These techniques leverage the strength of records-driven models to research complex scientific data, main to more correct and efficient diagnostic processes. This work suggests a ResNet-101 model that is meant to handle multi-class classification problems , offering potentially higher accuracy and deeper feature extraction at the cost of increased memory consumption and computational requirements. The ResNet-101 model was tested using the EyeNet dataset, which included 32 distinct types of diseases of the retina. The method achieved accuracy of 98.75% when evaluated on the EyeNet dataset.**

***Index Terms*—ResNet-101, Categorization, Advanced machine learning , Retinal disease dataset , Eye's photosensitive layer.**

I. INTRODUCTION

The retinal disorders have become a major concern for people across all age groups.The retina, which comprises a key photosensitive layer connected to the optic nerve within the human eye, plays an important role in converting low information into visible signals[1]. This photosensitive layer, after focussing light via the lens, sends those signals to the brain, allowing the device to have a visual repute. The macula is a crucial area of the retina, primarily responsible for processing detailed visual information. The macula processes data, which is subsequently passed from retina to brain via optic nerve, allowing for an understanding of images.Several retinal illnesses can disrupt this visual pathway, resulting in diseases include age-related macular degeneration (AMD), retinal tumors, and DME[2].In many industrialized countries, people elderly between 50 and 60 are increasingly prone to imaginative and prescient loss because of AMD. Based on recent studies in the US, this disease affects approximately 35% of those over the lifespan of 80[3]. Accurately detecting illnesses of the retina is especially difficult due to its variety, necessitating the expertise of skilled ophthalmologists. However, the emergence of laptop-aided diagnostic (CAD) systems has greatly enhanced the early detection and treatment of numerous ailments.Advances in technology have brought tremendous advantages to a variety of fields, particularly science[4]. Numerous methods and trends have been created to enhance the speed and standard of healthcare services. Advancement of Automatic Disease Detection (ADD) has significantly improved public fitness frameworks[5]. Among these advancements, retinal symptom analysis as part of ADD packages has a great potential to increase global eye care standards. Recently advances in machine learning (ML) and deep learning (DL) have resulted in creation of new techniques for classifying, segmenting, and detecting retinal problems.However, issues in gathering information and labeling continue, as researchers have emphasized in prior studies. Development of several ML and DL frameworks, includes as Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), AlexNet, ResNet, and VGG, has enabled researchers and healthcare professionals to more easily identify and categorize critical eye disorders. Furthermore, a hybrid machine learning-based method has been presented for the automated detection of retinal disorders.This have a look at introduces a deep learning technique making use of the ResNet-101 model for the classification of a couple of retinal sicknesses. The effectiveness of the proposed version has been assessed the use of the EyeNet Dataset, which accommodates 32 wonderful folders, every containing pix related to specific retinal situations. In order to ensure robust analysis, 70% of the dataset has been used for teaching, with the remaining 30% being utilized for validation. The

experimental results demonstrate that the ResNet-101 version achieved an exceptional accuracy rate of 98.75%. The research provides a significant improvement over conventional diagnostic techniques by applying this deep learning-based ResNet-101 model to the diagnosis of serious retinal diseases[6]. This could lead to earlier and more accurate analysis in clinical situations. The rest of the paper based on the following. Section II discusses previous studies relevant to this study. In Section III, we have detail proposed architecture and provide a comprehensive overview of dataset used .Section IV covers the experimental evaluation results, focusing on ResNet-101 model's performance. Section V includes a discussion and analysis of results. Finally, Section VI closes the report with recommendations for future research.

II. Related Work

Deep learning and pattern analysis have resulted in significant advancements in image processing and typography, particularly in clinical domain names. These technologies are frequently used to process complicated medical images, including retinal scans, which may be useful for identifying a variety of eye conditions. The adoption of these models has showed excellent efficacy due to their ability to robotically extract functions and identify diseases with great precision[7]. This is especially useful because it decreases reliance on guidance diagnosis from professionals. Many developing countries have limited access to healthcare specialists, making such technology even more important for early disease identification and treatment.

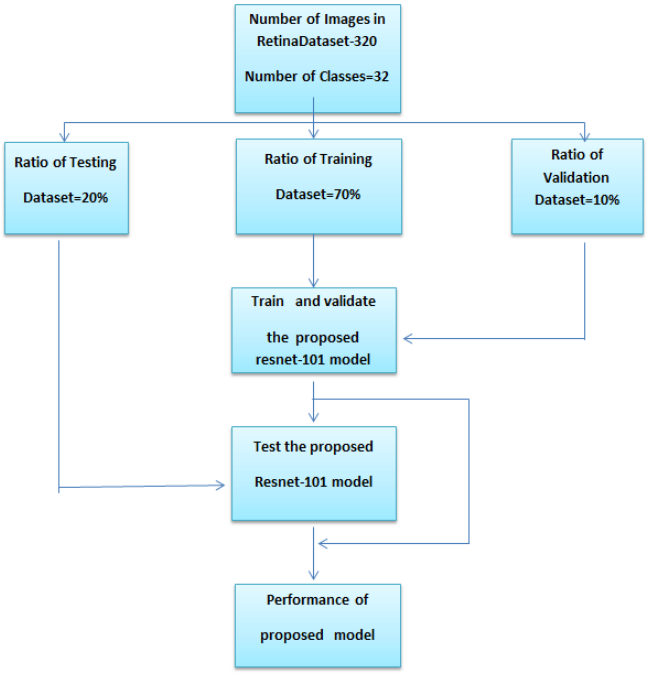


Fig. 1. Work flow diagram of proposed model

One of the critical applications of these technologies is optical coherence tomography, which is a non-invasive imaging technique that gives high-decision OCT move-sectional views of the retina. Now it is one of the most important techniques for early detection and control of retinal diseases. OCT allows healthcare businesses to visualize the intricate form of the retinal layers, which aids in the detection of diffused anomalies that may represent the early stages of a condition[8]. However, understanding and examining those snapshots in an automated manner remains difficult, and device mastery models are constantly improving to address these limitations. The increasing ability to gather comprehensive knowledge of systems positions them as promising solutions for widespread scientific usage in retinal disease prognosis. In the past decades, many models have been proposed for using OCT images in the classification of retinal diseases automatically. Amongst the most profound approaches involves combining CNNs with SVMs, which have been implemented with success. For example, a study with ROP, this is a condition associated with premature infants, proposed a new device that uses SegNet for vessel segmentation. This method is useful because ROP is associated with abnormal blood vessel development in the retina[9]. The study used a mixture of SIFT, Scale Invariant Feature Transform, and SURF, Speeded-Up Robust Features for extracting the feature from images. These techniques are well-reviewed for their ability to extract distinct and stable features from images.Once such characteristics were abstracted they get classified using Quantum SVM which is one of the upgraded traditional SVM. Their accent on the principles of quantum computing also helps improve its speed and accuracy of learning. In this context, the results achieved by the proposed system turn out to be incredibly accurate, like 95.5%, as compared to many other machine and deep learning systems. Such high accuracy makes this system very promising for early prognosis of ROP, especially in areas in which it is not easy to have access to specialized health care providers. Therefore, such a system will be an effective solution to early intervention in the disease.Another important area of research in retinal diseases is the detection of diabetic retinopathy, a frequent complication in patients with diabetes, which if not detected early, may lead to severe impairment of vision. The detection of the different grades of DR-from mild to proliferative-would prevent permanent loss of vision[10]. To this effect, some researchers have integrated ResNet-50, a powerful deep CNN, with Random Forest classifiers for diagnosis. ResNet-50 would capture crucial details by the deep feature extraction of images due to its capability to mimic the powerful deep CNN. Therefore, ResNet-50 would be very effective in the detection of early signs of DR from retinal images. Another area of important research in the detection of retinal disease is diabetic retinopathy (DR), that could, if untreated, lead to complete impairment of vision. Detection of the progression from mild to proliferative DR is critical for preventing irreversible loss of vision. In this regard, hybrid models have been developed that incorporate ResNet-50 as a deep CNN, classifying features through the Random Forest classifier in making a diagnosis. The deep capability for feature extraction by ResNet-50 enables it to be extremely

effective when capturing subtle details from retinal images. It's thus very effective in identifying early DR[11]. Application of Random Forest classifier After feature extraction, the categorization of the severity of the disease is performed through the Random Forest classifier. By using the advantage of CNNs in feature extraction, in conjunction with the high accuracy of Random Forest algorithms in classification tasks, this hybrid model becomes very promising. It has been tested on two publicly available significant datasets, Messidor-2 and EyePACS, containing different challenges of classification for DR stages. The data set Messidor-2 is classified into two: No Referable Diabetic Macular Edema and Referable DME, meaning whether a patient requires referral to a specialist or not. This model did very well with an accuracy rate of 96% for the given dataset, indicating its fitness in distinguishing between the two classes. The developing of such models shall, therefore, be of great promise for early detection and management of retinal diseases in the underserved population. It can empower healthcare providers to make decisions in a timely manner that may prove fruitful in relation to treatment by utilizing accurate and automated diagnosis tools. This is very beneficial for people in rural or economically disadvantageous regions where access to ophthalmologists and specialty diagnostic equipment will be relatively limited. Thus, the diagnostic workflows can significantly lighten the healthcare professional's workload to focus more on complicated cases that require a human mind. The EyePACS dataset is images classified into five categories representing different stages of diabetic retinopathy, from no signs of the disease up to proliferative DR. Similarly, on the same dataset, it sets up a respectable accuracy of 75.09% to its ability in working with a more complex task of multi-class classification. As a result, although worse than Messidor-2 two-category accuracy of classification, a certain certain value is saved which demonstrates good prospective applicability of proposed models in dealing with detailed DR classification.

III. METHODOLGY

The proposed resnet-101architecture and the dataset that was used are also covered in this section. The suggested approach has been step-by-step demonstrated in Fig 1.

A. Dataset

This observation was made using Yang et al.'s EyeNet dataset. It offers a sorted variety of thirty-two multiple types of records related to records from hospitals. The Table 1,32 retinal disease types found in the Eye-Net dataset are used. This dataset is used for the entirety of the model's implementation. The accompanying dataset contains pictures that have been labeled appropriately.

Dataset	Class Label
Eye-net	Adult Coats' Disease
	AdultFoveomacularDystrophy
	Age- Related Macular Degeneration
	AMNMacularNeuroretinopathy
	Antiphospholipid Antibody Syndrome
	Behcet's disease
	Bilateral Macular Dystrophy
	Bull's Eye Maculopathy Chloroquine
	Central Serous Chori- oretinopathy
	Choroidal Nevus
	CMV Chorioretinitis
	Cone -Rod Dystrophy
	Congenital Syphillis
	Diabetic Maculopathy Multiple Myeloma with Retinal Detachment
	Giant Retinal Tear
	North Carolina Dystrophy
	Leber's Stel- late Maculopathy
	Multifocal Exudative Detachments
	Macular Dystrophy
	Myelinated Nerve Fibers
	Juxtafoveal Telangiectasis DM Diabetes
	Optic Disc Drusen
	Roth Spot disease
	Pattern Dystrophy Simulating Fundus Flavimaculatus
	Retinal Folds Following Retinal Reattachment Surgery
	Reticular Pattern Dystro- phy
	Retro hyaloid Hemorrhage
	Solar Retinopathy Familial
	Susac's Syn- drome
	Self-Applied Retinal Detachment
	Terson's Syndrome
	Wyburn-Mason Syndrome

TABLE I
DATASET DESCRIPTION

B. Proposed Model

Deep learning knowledge of is one of the maximum widely used technology these days. It makes use of more than one processing layers within its framework, permitting computational fashions to research records styles at various levels of abstraction. These fashions excel at recognizing speech, figuring out visual items, and tackling various other detection duties. Deep learning concepts are inspired by the complicated form of the human mind. Resnet-101 is the most effective and powerful deep learning version. Although researchers have accelerated its application in other areas, Resnet-101 is widely used in the scientific enterprise. One method for constructing a deep learning model for the retinal disease category is to process retinal pictures sequentially. Initially, low-degree characteristics are obtained, followed by the gathering of the more delicate center and intermediate-degree features. These final functions are then provided to a trainable classifier to determine the proper class.

1) Resnet-101 model: ResNet-101: It is a deep neural network architecture with 101 layers. The idea behind this network is useful specifically for the problem of vanishing gradients in very deep networks, which happens at the time when gradients become too small to propagate backwards. In contrast, fig 2 residual connections, also known as skip connections, are used to achieve this. The model bypasses a particular layer, allowing information to flow directly between earlier and later layers. This kind of architecture enables deeper networks to be trained more efficiently without loss of gradient strength[12]. The residual network was designed using residual blocks, which consist of convolutional layers followed by a batch normalization and a ReLU activation function. Batch normalization stabilizes and accelerates the training with normalized inputs across each mini-batch; in addition, the ReLU activation provides the non-linearity for the learning of complex patterns..It thus basically simplified the training process of deep networks by focusing on learning

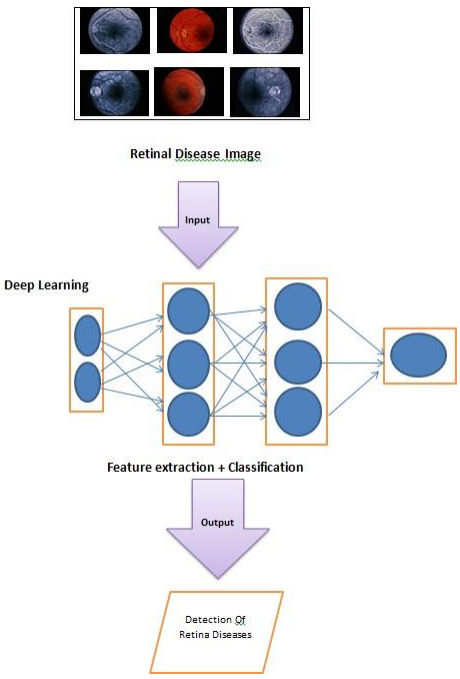


Fig. 2. Proposed model architecture

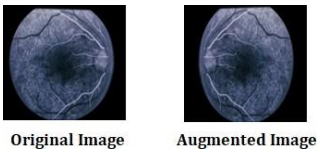


Fig. 3. Flip

the residuals of training-output difference rather than trying to learn the full mapping.Shortcut connections can also be one of the innovations on ResNet-101: The gradients directly pass through the network, bypassing one or more layers. Thus, the vanishing gradient problem is avoided, while the updates of weights in deeper layers are eased. With this factor in mind - accuracy, which is now no longer influenced by the depth of the network, there has emerged a basic architecture in deep learning for image classification tasks, object detection, segmentation, etc and further for classification of images, object detection, and segmentation.

C. Data Augmentation

To make certain sufficient statistics is available for training and to prevent overfitting, we used information augmentation strategies. Augmentation strategies covered rescaling, zooming, and flipping the images. By applying numerous random differences, we ensured that the model encountered a diverse range of photos, which facilitates in avoiding overfitting and improves generalization. In our proposed version, information augmentation is carried out using Keras's ImageDataGenerator. This procedure includes six essential steps, each of which transforms the pix in distinctive approaches, improving the model's capacity to generalize nicely across new facts[13]. Data augmentation is important for improving the overall efficiency and outcomes of system mastering fashions by means of producing extra and sundry education samples. A comprehensive dataset permits the model to perform extra accurately and efficaciously. For example, saturation changes increase the opacity of pics, making crucial capabilities extra visible and helping in higher sickness diagnosis. Another augmentation method converts the unique image to grayscale, changing its hues to grayscale assessment. Flipping **fig3** the original photograph changes its direction, at the same time as brightening enhances the picture's brightness.

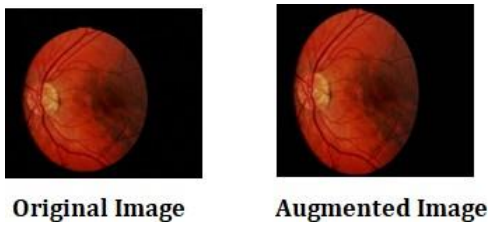


Fig. 4. Zoom

Zooming **fig4** is every other step inside the augmentation manner. The specific augmentation strategies used can range relying at the studies and its requirements.

D. Adaptive Learning Rate Estimation

Optimization techniques based on stochastic gradient methods necessitate improvement in performance for machine learning models. Among these, a very effective and popular algorithm is Adaptive Moment Estimation, or Adam. Adaptation in Adam is beneficial over the traditional method of gradient descent in the sense that it utilizes the first and second moments of the gradient to dynamically adapt its learning rate. It allows for very efficient updates of the parameters of a model, making Adam remarkably effective for dealing with massive amounts of data and deep neural networks[14]. Besides this, its ability to robustly handle noisy and sparse gradients guarantees stability and smoothness of the optimization procedure, even in complex scenarios.One reason Adam achieves great popularity lies in its computation efficiency and low memory usage; this means it is suitable for big models with many parameters. In contrast, traditional stochastic gradient descent needs learning rates to be manually tuned laboriously. Learning rates of Adam are automatically scaled by using individual adaptive learning rates based on recent momentum, thus preventing slow convergence and vanishing gradients during the training process. In addition, the hyperparameters of Adam are typically easy to tune. In most cases, default settings prove effective, and a heavy fine-tuning is not required.We have chosen Adam since it is quite appropriate for optimization in the context of deep models, especially the ones that are complexly arranged. Its seamless integration by the Keras library built-in functionalities will ensure us centering ourselves on the improvement of the

accuracy of the model as well as efficiency. Adam's adaptive learning rates enabled the network to train faster. Its high ability to process large-scale data enabled us to remain efficient and accurate during optimization. Thus, it was Adam-the most ideal optimizer for this project-that made this a great success contributor for our model's performance.

IV. PERFORMACE EVALUATION

To assess performance of the proposed model, the model's predictions are compared with the actual labels. These labels are furnished by an ophthalmologist who annotates the snap shots. In **Figure 5**, the confusion matrix is displayed to evaluate the performance of the Resnet-101. This matrix provides a comprehensive overview of the model's classification results, allowing for a clear assessment of its strengths and weaknesses in predicting different classes. **Figure 6**illustrates the outcomes of the classification report produced by applying the proposed model. This report summarizes the model's performance metrics, offering insights into its effectiveness in classifying the different categories . We evaluate the model based on these labels[15]. Precision is sometimes called as positive predictive value, and recall is the real positive rate or actual values (as shown in equations 1 and 2).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Dataset consists of 32 kinds of retina diseases, so the overall performance measures are calculated for each magnificence in my opinion. The version completed a 100% precision and 99% recollect across all classes.

V. EXPERIMENTAL RESULTS AND OBSERVATIONS

The collection of data is separated into two subsets: training and validation. The program's development is done in Python using the Keras package and operates on an Intel Core i5-7200 CPU at 2.70GHz.For training, a Google GPU is used. We tested diverse optimizers, and Adam furnished the excellent effects. Learning rate used for this optimizer is 0.001. All other hyperparameters are chosen after thorough testing and evaluation. Trained data on different epochs and achieved 98.75% validation accuracy.Validation accuracy is 0.9822, and the validity loss is 0.0498. The x-axis shows a large number of epochs (from 0 to 30), whereas the y-axis denotes accuracy (from 0 to 1).The blue line denotes training accuracy, whereas the orange line represents validation accuracy. **Fig 7** Initially, training accuracy increases frequently, whereas the validation accuracy increases rapidly, nearly reaching 1.0 (100%) in the first 10 epochs.Both training and validation accuracies stabilize after about 10-15 epochs and remain consistent, varying slightly between 0.90 and at least a 1.0, showing proper version generalization.Based on the x-axis of the graphic, it is shown that 30 epochs were used to train the version. It is usual for education to use a deep learning model to evaluate trends in data while avoiding overfitting or underfitting.

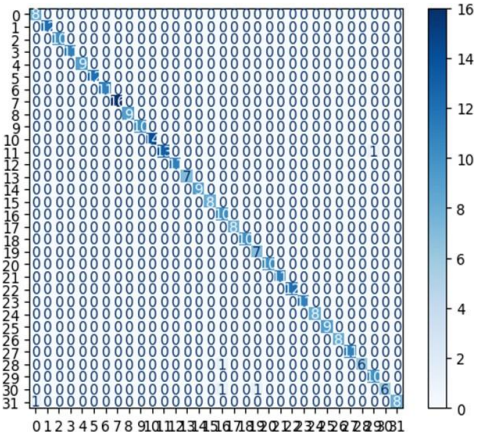
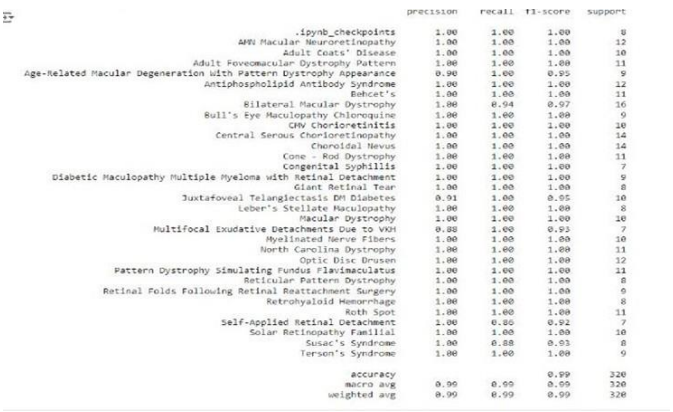


Fig. 5. Confusion matrix



	precision	recall	f1-score	support
.ipynb_checkpoints	1.00	1.00	1.00	8
AMN Macular Neuroretinopathy	1.00	1.00	1.00	12
Adult Coats' Disease	1.00	1.00	1.00	10
Adult Foveomacular Dystrophy Pattern	1.00	1.00	1.00	11
Age-Related Macular Degeneration With Pattern Dystrophy Appearance	0.90	1.00	0.95	9
Antiphospholipid Antibody Syndrome	1.00	1.00	1.00	12
Behcet's	1.00	1.00	1.00	11
Bilateral Macular Dystrophy	1.00	0.94	0.97	16
Bull's Eye Maculopathy Chloroquine	1.00	1.00	1.00	9
CMV Chorioretinitis	1.00	1.00	1.00	10
Central Serous Chorioretinopathy	1.00	1.00	1.00	14
Choroidal Nevus	1.00	1.00	1.00	14
Cone - Rod Dystrophy	1.00	1.00	1.00	11
Congenital Syphillis	1.00	1.00	1.00	7
Diabetic Maculopathy Multiple Myeloma with Retinal Detachment	1.00	1.00	1.00	9
Giant Retinal Tear	1.00	1.00	1.00	8
Juxtafoveal Telangiectasis DM Diabetes	0.91	1.00	0.95	10
Leber's Stellate Maculopathy	1.00	1.00	1.00	8
Macular Dystrophy	1.00	1.00	1.00	10
Multifocal Exudative Detachments Due to VKH	0.88	1.00	0.93	7
Myelinated Nerve Fibers	1.00	1.00	1.00	10
North Carolina Dystrophy	1.00	1.00	1.00	11
Optic Disc Drusen	1.00	1.00	1.00	12
Pattern Dystrophy Simulating Fundus Flavimaculatus	1.00	1.00	1.00	11
Reticular Pattern Dystrophy	1.00	1.00	1.00	8
Retinal Folds Following Retinal Reattachment Surgery	1.00	1.00	1.00	9
Retrohyaloid Hemorrhage	1.00	1.00	1.00	8
Roth Spot	1.00	1.00	1.00	11
Self-Applied Retinal Detachment	1.00	0.86	0.92	7
Solar Retinopathy Familial	1.00	1.00	1.00	10
Susac's Syndrome	1.00	0.88	0.93	8
Terson's Syndrome	1.00	1.00	1.00	9
accuracy			0.99	320
macro avg	0.99	0.99	0.99	320
weighted avg	0.99	0.99	0.99	320

Fig. 6. Classification repor

At the start (epoch zero), both training and validation losses are fairly high **fig 8**. This is expected because the version starts with random weights and hence generates incorrect predictions, resulting in a large loss.The training loss begins at roughly 3.0, whereas the validation loss begins just below 2.0.During the first few epochs (up to epoch 5), the loss for training and validation drops significantly. This implies that the model is learning quickly and modifying its parameters effectively in the early stages. By epoch five, the training loss is less than 0.5, and the validation loss has also decreased, showing that the model is fitting both the training and validation data more accurately over time. After epoch 10, both training and validation losses will continue to decline, though at a slower rate. This is expected since the version fine-tunes its parameters and receives feedback in the direction of convergence. Both losses begin to stabilize around epoch 20, with the training loss approaching zero and the validation loss also reaching very low levels, indicating little errors in predictions based on both education and unseen validation statistics.

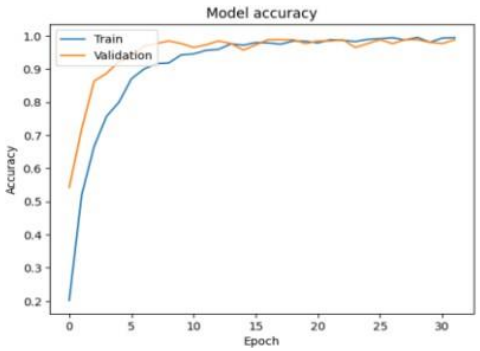


Fig. 7. Model accuracy by implementing 30 epochs

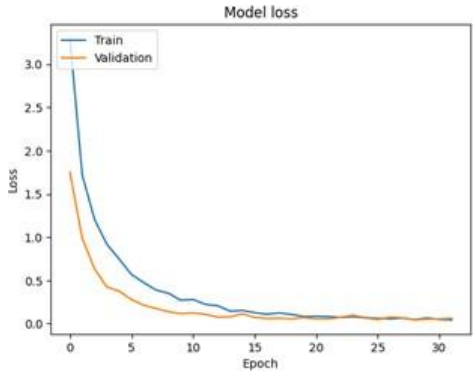


Fig. 8. Validation and training loss comparsion at 30 epochs

VI. CONCLUSION

We will use the architecture modified ResNet-101 to classify types of multi-class retinal disease. The architecture has been specifically designed to raise the ability to differentiate between different retinal conditions and make more proficient use of operations of convolution with the addition of several residual layers. We will also use statistical augmentation techniques for improvement in the generalization ability of the model. For instance, these augmentation techniques that were used comprised random rotations, zoom, flip, and other brightness manipulations. Essentially, they added variations to the training dataset through creating slightly changed versions of the existing images in order to assist the model in learning a larger, more diverse set of features that improved performance on unseen data.

REFERENCES

[1] Hasan, D. A., Zeebaree, S. R., Sadeeq, M. A., Shukur, H. M., Zebari, R. R., & Alkhayyat, A. H. (2021, April). Machine learning-based diabetic retinopathy early detection and classification systems-a survey. In *2021 1st Babylon International Conference on Information Technology and Science (BICITS)* (pp. 16-21). IEEE.

[2] Flores, R., Carneiro, Â., Vieira, M., Tenreiro, S., & Seabra, M. C. (2021). Age-related macular degeneration: pathophysiology, management, and future perspectives. *Ophthalmologica*, *244*(6), 495-511.

[3] Samdal, K. (2007). *Cost-Utility Analysis of replacing photodynamic therapy with verteprofin by anti-VEGF treatment with ranibizumab on patients with predominantly classic neovascular age-related macular degeneration* (Master's thesis).

[4] El-Baz, A., Beache, G. M., Gimel farb, G., Suzuki, K., Okada, K., Elnakib, A., ... Abdollahi, B. (2013). Computer-aided diagnosis systems for lung cancer: challenges and methodologies. *International journal of biomedical imaging*, *2013*(1), 942353.

[5] Ganasegeran, K. (2021). Deep learning for disease prediction in public health. In *Deep Learning for Biomedical Applications* (pp. 157-180). CRC Press.

[6] Hussain, S., Mubeen, I., Ullah, N., Shah, S. S. U. D., Khan, B. A., Zahoor, M., ... & Sultan, M. A. (2022). Modern diagnostic imaging technique applications and risk factors in the medical field: a review. *BioMed research international*, *2022*(1), 5164970.

[7] Anwar, S. M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., & Khan, M. K. (2018). Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, *42*, 1-13.

[8] Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., ... & Fujimoto, J. G. (1991). Optical coherence tomography. *science*, *254*(5035), 1178-1181.

[9] Maheshwari, S., Gupta, P. K., Sinha, R., & Rawat, P. (2020). Knowledge, attitude, and practice towards coronavirus disease 2019 (COVID-19) among medical students: A cross-sectional study. *Journal of Acute Disease*, *9*(3), 100-104.

[10] Lakshminarayanan, V., Kheradfallah, H., Sarkar, A., & Jothi Balaji, J. (2021). Automated detection and diagnosis of diabetic retinopathy: A comprehensive survey. *Journal of imaging*, *7*(9), 165.

[11] Lin, C. L., & Wu, K. C. (2023). Development of revised ResNet-50 for diabetic retinopathy detection. *BMC bioinformatics*, *24*(1), 157.

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[13] Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.

[14] Jais, I. K. M., Ismail, A. R., & Nisa, S. Q. (2019). Adam optimization algorithm for wide and deep neural network. *Knowl. Eng. Data Sci.*, *2*(1), 41-46.

[15] Li, B., Chen, H., Yu, W., Zhang, M., Lu, F., Ma, J., ... & Chen, Y. (2024). The performance of a deep learning system in assisting junior ophthalmologists in diagnosing 13 major fundus diseases: a prospective multi-center clinical trial. *npj Digital Medicine*, *7*(1), 8.

cb3.pdf

ORIGINALITY REPORT

5%	3%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Asif Nawaz, Tariq Ali, Ghulam Mustafa, Muhammad Babar, Basit Qureshi. "Multi-Class Retinal Diseases Detection Using Deep CNN With Minimal Memory Consumption", IEEE Access, 2023 Publication	2%
2	pt.scribd.com Internet Source	1%
3	Submitted to Izmir Katip Ãelebi Ãniversitesi Student Paper	1%
4	www.mdpi.com Internet Source	<1%
5	Submitted to University of Edinburgh Student Paper	<1%
6	www.scpe.org Internet Source	<1%
7	www.csc.jfn.ac.lk Internet Source	<1%
8	www.ijraset.com Internet Source	