Chapter 1

INTRODUCTION

1.1 Overview

Pediatric glaucoma is a rare but severe group of eye diseases characterized by progressive damage to the optic nerve, often associated with elevated intraocular pressure. According to the World Health Organization (WHO), glaucoma affects over 76 million individuals globally, and this number is expected to rise as the population ages. Early detection is critical, as it significantly reduces the risk of irreversible vision loss through timely intervention. However, traditional diagnostic methods, such as tonometry and visual field testing, are often subjective, time-consuming, and heavily reliant on specialized healthcare facilities. This leads to many cases remaining undiagnosed until substantial damage has occurred.

The urgent need for innovative, accurate, and efficient screening solutions has driven the exploration of advanced technologies. Recent advancements in machine learning and computer vision have opened new avenues for automating medical diagnostics. In particular, deep learning models, especially Convolutional Neural Networks (CNNs), have demonstrated outstanding success in image classification tasks across various domains, including medical imaging. These models excel at learning complex patterns from large datasets, making them highly promising for the detection of glaucoma.

The performance of the model is rigorously evaluated using metrics such as precision, recall, F1-score, confusion matrices, and ROC-AUC curves to ensure its reliability and robustness. Ultimately, the trained model is integrated into a web or mobile application, allowing healthcare professionals to upload images and conduct real-time, automated diagnosis. This innovation seeks to support early intervention, improve clinical outcomes, and significantly reduce the risk of blindness among pediatric glaucoma patients, while also making advanced diagnostic capabilities accessible even in resource-limited settings.

1.2 Aim and Objective

The aim of this project is to develop an automated, accurate, and efficient deep learning-based system for the early detection of pediatric glaucoma using retinal fundus and OCT images. The objective is to design and implement a lightweight convolutional neural network model, specifically MobileNetV2, enhanced with transfer learning techniques, to classify images as either healthy or glaucoma-affected. The system integrates data preprocessing and augmentation to improve model performance and generalization. It also features a user-friendly web interface for real time predictions, making it suitable for deployment in clinical settings, especially in resource-limited areas. This project ultimately seeks to support early diagnosis, reduce dependence on specialized expertise, and improve patient outcomes in pediatric glaucoma care.

1.3 Existing System

The Existing systems for detecting glaucoma primarily rely on traditional diagnostic methods, each with its own limitations. Visual field testing assesses peripheral vision loss but is highly subjective and depends on patient cooperation, making it challenging for young children. Optical Coherence Tomography (OCT) provides detailed imaging of the retina and optic nerve to detect structural changes; however, it requires manual interpretation by specialists. Similarly, retinal fundus photography captures images of the retina for optic disc evaluation, but its analysis is time-consuming and demands expert knowledge.

Some modern approaches use machine learning models for glaucoma detection, but they primarily focus on adult cases and lack robustness for pediatric glaucoma due to limited datasets. Additionally, mobile eye screening applications assist in eye examinations, but their diagnostic accuracy is generally lower, and they are not specialized for detecting pediatric glaucoma. These existing systems often rely on clinical expertise, are costly, and pose challenges in early diagnosis, particularly in pediatric cases, where early intervention is crucial to preventing vision loss.

1.4 Proposed Solution

The proposed system aims to develop an automated deep learning-based solution for the early detection of pediatric glaucoma using retinal fundus and Optical Coherence

Tomography (OCT) images. It leverages Convolutional Neural Networks (CNNs) to identify glaucoma-specific patterns in the optic nerve and surrounding retinal structures. Given the challenge of limited pediatric glaucoma data, the model incorporates transfer learning with pre-trained architectures like VGG16 and ResNet50 to enhance accuracy. Data augmentation techniques such as image rotation, flipping, brightness adjustments, and noise addition are employed to improve model generalization and performance.

To ensure reliability, the system is evaluated using key performance metrics, including accuracy, precision, recall, and F1-score. The trained model is integrated into a user-friendly web or mobile application, allowing healthcare professionals to upload and analyze retinal fundus and OCT images in real-time. Additionally, Training is conducted on Visual Studio, using the Adam optimizer and early stopping techniques to optimize efficiency and prevent overfitting. The model's performance is assessed using metrics such as accuracy. Validation on a separate test set ensures robust generalization. Ethical considerations, including patient privacy and data biases, are addressed, while iterative improvements informed by user feedback enhance the system’s usability and reliability for clinical deployment.

1.5 Motivation

The Pediatric glaucoma is a rare but serious eye disease that can lead to permanent vision loss if not detected early. Traditional diagnostic methods like visual field testing, Optical Coherence Tomography (OCT), and fundus photography rely on specialized equipment and expert interpretation, making early detection challenging, especially in remote areas with limited healthcare access. With recent advancements in deep learning and computer vision, artificial intelligence (AI) has shown promising results in medical image analysis. Convolutional Neural Networks (CNNs) have demonstrated high accuracy in disease classification, making them a powerful tool for automating glaucoma detection.

However, most AI models are trained on adult datasets, and pediatric glaucoma has unique characteristics that require specialized training. This project aims to bridge that gap by leveraging pre-trained deep learning models like VGG16, ResNet50, and MobileNetV2, combined with data augmentation techniques to improve accuracy on pediatric cases.

Chapter 2

LITERATURE SURVEY

2.1 Survey Paper:

[1] Cost-Effective Real-Time Eye Disease Detection and Classification Using Deep Learning Techniques [2024]

The research paper titled "Cost-Effective Real-Time Eye Disease Detection and Classification Using Deep Learning Techniques" was authored by Archana Chaudhari, Pranav Shelke, Saurabh Sandbhor, and Prathamesh Thombare and published in the 15th ICCCNT IEEE Conference at IIT-Mandi, India, in 2024. The study focuses on an efficient deep learning approach for detecting and classifying eye diseases using real-time image processing techniques.

The proposed architecture is based on VGG-19, a deep Convolutional Neural Network (CNN) that consists of multiple convolutional layers (3×3 filters), max pooling layers (2×2 filters), and fully connected layers. The model is optimized for medical image classification, enabling accurate detection of eye diseases from fundus images.

The methodology follows a preprocessing step to enhance image quality, followed by feature extraction using VGG-19. The extracted features are classified using a Softmax layer, which assigns probabilities to different disease categories. The system is trained on a dataset of retinal fundus images, using transfer learning and fine-tuning techniques to improve classification accuracy.

Despite its effectiveness, the study identifies certain limitations. The model requires a large amount of labeled data for accurate predictions, and its performance depends on high-quality input images. Additionally, computational requirements may limit real-time implementation in low-resource environments. Future improvements include integrating attention mechanisms and hybrid CNN-transformer models for enhanced accuracy and robustness.

[2] OptiNet: Leveraging Deep Learning Ensembles for Timely Detection of Visual Abnormalities in Children [2024]

The research paper titled "OptiNet: Leveraging Deep Learning Ensembles for Timely Detection of Visual Abnormalities in Children" was authored by Chethan Reddy, Hariprada, Nilkant Manik, Satish, and Prof. Shilpa S from PES University, Bengaluru, India. It was published in the 2024 IEEE International Conference (PICET) with ISBN 979-8-3503-6974-8/24. The study introduces a deep learning-based system for the early detection of pediatric visual abnormalities using retinal fundus images.

The proposed architecture consists of an ensemble of three pre-trained convolutional neural networks (CNNs)—ResNet-50, VGG-19, and EfficientNetB0. Each model plays a distinct role in feature extraction and classification. ResNet-50 is used for high-level feature extraction, VGG-19 enhances detailed feature representation, and EfficientNetB0 provides high-efficiency classification with fewer parameters. These models are combined through ensemble learning, where the individual outputs are weighted and aggregated to improve diagnostic accuracy.

The methodology follows a structured approach beginning with dataset collection from Bajwa Eye Hospital, which includes approximately 600 fundus images of both normal and diseased eyes. To enhance model generalization, data augmentation techniques such as rotation, flipping, and brightness adjustments are applied. The system uses transfer learning, leveraging the pre-trained models to extract meaningful features from retinal images. The ensemble model is trained on the processed dataset and deployed as a user-friendly web application using React for the frontend and FastAPI for backend processing, allowing for real-time, AI-assisted diagnosis. The weighted averaging ensemble technique ensures that the combined predictions of all three models enhance the overall diagnostic accuracy.

The study identifies certain limitations. The model performance is dependent on the quality and diversity of the dataset, and data scarcity poses a challenge for deep learning model training. Additionally, real-time deployment in low-resource clinical settings may require optimization due to computational constraints. Future improvements could involve dataset expansion, advanced ensemble strategies, and the integration of more robust deep learning architectures to further improve the system’s performance and clinical applicability.

[3] A Transfer Learning-Based Web App for Glaucoma Detection Using Low-Cost Ophthalm- oscopic Camera [2024]

The research paper titled "A Transfer Learning Based Web App for Glaucoma Detection Using Low-Cost Ophthalmoscopic Camera" was authored by Chethan B K, Sanjana Vishwanath Sambargi, Roopashree D, Prabhavathi K, Kiran Puttegowda, and Sunil Kumar D S. It was published in the 2024 Second International Conference on Networks, Multimedia, and Information Technology (NMITCON) by IEEE, with ISBN 979-8-3503-7289-2/24. The study focuses on leveraging transfer learning to develop a cost-effective web application for glaucoma detection using low-cost ophthalmoscopic cameras.

The proposed architecture is built on InceptionResNetV2, a deep convolutional neural network (CNN) optimized for medical image classification. The model consists of multiple layers, including convolutional blocks, residual connections, and batch normalization, ensuring high efficiency and accuracy in detecting glaucomatous and non-glaucomatous eyes. The system also incorporates an ophthalmoscopic camera with a NoIR sensor and a 20-dioptre lens to capture high-quality retinal fundus images for analysis.

The methodology begins with image acquisition using an ophthalmoscopic camera, followed by preprocessing techniques such as segmentation and noise reduction. The preprocessed images are then passed through the pre-trained InceptionResNetV2 model, where features are extracted and classified using a binary classification approach (glaucoma or non-glaucoma). The trained model is deployed as a web-based application that allows users to upload retinal images and receive instant diagnostic predictions. Additionally, a secure cloud-based database is used to store patient records and prediction results for further analysis.

Despite its advantages, the study highlights certain limitations. The model's accuracy is dependent on high-quality retinal images, and variations in lighting conditions or occlusions can affect detection performance. Additionally, the system requires continuous updates with new medical datasets to improve reliability and generalization. Future enhancements include integrating real-time telemedicine features, expanding the dataset with diverse eye conditions, and optimizing computational efficiency for faster processing in real-world clinical settings.

[4] Cataract Classification by Applying MobileNetV2 on Different Edge Detection Filters [2024]

The research paper titled "Cataract Classification by Applying MobileNetV2 on Different Edge Detection Filters" was authored by Sazzatul Islam Anik, Md. Samir Hasan, Hosney Jahan, Md Mahbubar Rahman, T. M. Shahriar Sazzad, and Jakia Afroz Jahan. It was published in the 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) at the Military Institute of Science and Technology (MIST), Dhaka, Bangladesh. The paper explores the effectiveness of different edge detection filters in enhancing fundus images for cataract classification using deep learning.

The proposed architecture employs MobileNetV2, a lightweight convolutional neural network (CNN) optimized for medical image classification. This model is known for its efficiency, utilizing depthwise separable convolutions to reduce computational complexity while maintaining high accuracy. The system integrates multiple edge detection filters, including Prewitt, Sobel, Zero Crossing, Roberts Cross, and Marr-Hildreth filters, to enhance image clarity before classification.

The methodology follows a structured pipeline, starting with data acquisition from the Combined Military Hospital (CMH), Dhaka, where fundus images were collected and labeled by

ophthalmologists. To improve classification performance, data augmentation techniques such as rotation, flipping, and center cropping were applied to balance the dataset. The preprocessed images were then passed through various edge detection filters, which helped improve retinal feature visibility. The MobileNetV2 model was fine-tuned using transfer learning, and its classification performance was evaluated using metrics such as Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). The study found that the Prewitt filter provided the highest image clarity and improved the model’s accuracy for cataract detection.

Despite its effectiveness, the study has some limitations. The dataset size was relatively small, which may limit the model’s generalization capability. Additionally, the model's performance is sensitive to variations in image quality, camera settings, and lighting conditions, which could impact real-world applicability. Future improvements include expanding the dataset, integrating more advanced deep learning architectures like InceptionV3 and ResNet, and testing the system on other ocular diseases such as diabetic retinopathy and glaucoma.

[5] Real-Time Smart Phone-Enabled Eyes Disease Classification [2024]

The research paper titled "Real-Time Smart Phone-Enabled Eye Disease Classification" was authored by Bollem Poojitha, Nikitha G.V, K.S Sampath, Neelima N, and T.V Smitha from Amrita School of Engineering, Bengaluru, India. It was published in the 2024 3rd International Conference for Advancement in Technology (ICONAT), Goa, India, with ISBN 979-8-3503-5417-1/24. This paper presents an AI-based real-time smartphone-enabled application for diagnosing eye diseases, making automated detection more accessible and user-friendly.

The proposed architecture utilizes a Convolutional Neural Network (CNN) trained on a curated dataset of eye fundus images, including normal, cataract, glaucoma, and diabetic retinopathy cases. The CNN model is trained in PyCharm and deployed using Streamlit, a cloud-based interface, allowing real-time classification via a smartphone. The system processes input images, extracts relevant features, and provides instant classification results, making it beneficial for remote and underserved areas.

The methodology follows a four-step approach: image acquisition, preprocessing, deep learning model training, and real-time visualization. Initially, fundus images are captured using smartphone-compatible ophthalmoscopic cameras and preprocessed using contrast adjustment, noise reduction, and feature extraction techniques. The CNN model is trained on these images using data augmentation for improved accuracy. The application enhances accessibility by providing a simple, user-friendly interface for healthcare professionals and patients.

Despite its advantages, the study highlights certain limitations. The model’s performance is influenced by lighting variations and image quality, which can impact classification accuracy. Additionally, real-time deployment in low-resource settings may face computational constraints. Future improvements include enhancing the dataset, integrating multi-modal imaging techniques, and optimizing the system for mobile devices to improve diagnostic reliability and efficiency.

[6] MobileNetV2-based Deep Learning for Retinal Disease Classification on a Mobile application [2023]

The research paper titled "MobileNetV2-based Deep Learning for Retinal Disease Classification on a Mobile Application" was authored by Pattarakorn Intaraprasit, Toan Huy Bui, and May Phu Paing from King Mongkut’s Institute of Technology Ladkrabang, Thailand, and Tokai University, Japan. It was published in the 2023 Biomedical Engineering International Conference (BMEiCON-2023), IEEE, with DOI 10.1109/BMEiCON60347.2023.10322079. The study focuses on using deep learning-based image classification for diagnosing retinal diseases through a mobile application, making diagnosis faster and more accessible.

The proposed architecture is based on MobileNetV2, a lightweight and efficient Convolutional Neural Network (CNN) optimized for mobile applications. MobileNetV2 uses depthwise separable convolutions, significantly reducing computational complexity while maintaining high classification accuracy. The model consists of 53 convolution layers and a global average pooling layer, enabling efficient feature extraction for classifying cataract, glaucoma, diabetic retinopathy, and normal retinal images.

The methodology follows a structured pipeline beginning with image acquisition from public databases, including High-Resolution Fundus (HRF), IDRiD, and Ocular Recognition Databases. The dataset was preprocessed using contrast enhancement and noise reduction techniques. Images were then classified into four categories: Cataract (CAT), Glaucoma (GLC), Diabetic Retinopathy (DR), and Normal Retina. The MobileNetV2 model was trained using transfer learning with ImageNet-pretrained weights and optimized with Adamax optimizer. The model was evaluated using 5-fold cross-validation, achieving good accuracy. The trained model was integrated into an Android application, allowing real-time retinal disease classification.

Despite its effectiveness, the study identifies certain limitations. The dataset used for training was limited to four disease categories, restricting its generalization to other retinal disorders. Additionally, the model's performance depends on high-quality fundus images, making real-time deployment in low-resolution mobile cameras challenging. Future improvements include expanding the dataset, optimizing the model for additional eye diseases, and integrating more advanced deep learning architectures.

[7] Early Diagnosis of Types of Glaucoma Using Multi Feature Analysis Based on DBN Classification [2023]

The research paper titled "Early Diagnosis of Types of Glaucoma Using Multi-Feature Analysis Based on DBN Classification" was authored by Likhitha Sunkara, Bhargavi Lahari Vema, Hema Lakshmi Prasanna Rajulapati, Avinash Mukkapati, and VBKL Aruna from VRSEC (JNTUK), Vijayawada, India. It was published in the 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS) with ISBN 979-8-3503-9846-5/23. The paper focuses on an early detection model for glaucoma using Deep Belief Networks (DBN), incorporating multi-feature analysis for enhanced classification accuracy.

The proposed architecture is based on Deep Belief Networks (DBN), a type of deep learning model that consists of multiple layers of Restricted Boltzmann Machines (RBMs) stacked together. DBNs are capable of learning hierarchical features from input data, making them highly effective for medical image classification. The model processes retinal fundus images and analyzes various ocular characteristics, such as Retinal Nerve Fiber Layer (RNFL) thickness, Optical Coherence Tomography (OCT) scans, and retinal imaging, to classify different types of glaucoma (Open-Angle, Angle-Closure, Congenital, and Secondary Glaucoma).

The methodology follows a structured approach, starting with image acquisition from a glaucoma dataset repository, followed by preprocessing techniques such as resizing, grayscale conversion, and segmentation to enhance feature extraction. The GLCM (Gray-Level Co-occurrence Matrix) technique is used for feature extraction, capturing important textural and structural characteristics from the retinal images. A 70-30% data split is applied for training and testing the model. The Deep Belief Network (DBN) classifier is then trained to classify different types of glaucoma based on extracted features, using a multi-feature vector (MFV) approach that combines multiple retinal features for improved accuracy. The model is evaluated using accuracy, precision, recall, and F1-score metrics, achieving 89.24% accuracy in glaucoma classification.

Despite its effectiveness, the study highlights certain limitations. The model's performance is affected by variations in image quality and lighting conditions, which may impact feature extraction accuracy. Additionally, the dataset is relatively small, which may limit the model’s generalization capability. Future enhancements include expanding the dataset, integrating hybrid deep learning models, and improving real-time classification efficiency to make the system more clinically applicable.

[8] External validation of a deep learning detection system for glaucomatous optic neuropathy: a real-world multicentre study [2023]

The research paper titled "External Validation of a Deep Learning Detection System for Glaucomatous Optic Neuropathy: A Real-World Multicentre Study" was authored by Xu Qian, Song Xian, Su Yifei, Guo Wei, Hanruo Liu, Xi Xiaoming, Chunyan Chu, Yin Yilong, Yu Shuang, Ma Kai, Cheng Mei, and Qu Yi. It was published in the journal Eye (2023) 37:3813–3818, under The Royal College of Ophthalmologis. The study focuses on externally validating a deep learning-based system for diagnosing glaucomatous optic neuropathy (GON) using real-world multicentre data.

The proposed architecture is based on a deep learning system (DLS) for automated GON diagnosis, developed by Tencent Healthcare, Shenzhen. It incorporates Residual Neural Networks (ResNet-18) for image quality assessment and ResNet-34 for GON classification. The model first segments the optic disc using U-Net, then extracts features related to GON, such as cup-to-disc ratio (CDR), rim thinning, and retinal nerve fiber layer (RNFL) defects. The system integrates a multi-task deep learning framework that simultaneously predicts glaucoma and optic disc atrophy, improving diagnostic performance.

The methodology involves collecting a large-scale dataset from multiple hospitals in China, including Qilu Hospital of Shandong University (QHSDU) and three other medical institutions. The dataset consists of 3049 images in validation dataset 1, 7495 images in validation dataset 2, and 516 high myopia (HM) cases in validation dataset 3. Images undergo quality assessment, preprocessing, and segmentation before being classified by the deep learning model. The algorithm's performance is compared with manual grading by ophthalmologists, and its generalization capability is tested across varying image qualities and retinal comorbidities. The system achieves accuracy of 83.18% and 81.40% in datasets 1 and 2, with AUC scores of 85.17% and 86.64%, outperforming human graders.

Despite its strong performance, the study identifies several limitations. The model struggles with highly variable fundus image quality, which affects classification reliability. Additionally, retinal comorbidities such as diabetic retinopathy (DR) and age-related macular degeneration (AMD) cause misclassifications, leading to false positives and false negatives. The dataset primarily consists of Chinese patients, limiting its applicability to diverse populations. Future improvements include expanding datasets, refining deep learning architectures, and integrating multimodal clinical data (e.g., OCT scans and intraocular pressure measurements) to enhance diagnostic accuracy and generalization.

[9] Deep-learning approach to detect childhood glaucoma based on periocular photograph [2023]

The research paper titled "Deep-learning approach to detect childhood glaucoma based on periocular photographs" was authored by Yoshiyuki Kitaguchi, Rina Hayakawa, Rumi Kawashima, Kenji Matsushita, Hisashi Tanaka, Ryo Kawasaki, Takahiro Fujino, Shinichi Usui, Hiroshi Shimojyo, Tomoyuki Okazaki, and Kohji Nishida. It was published in Scientific Reports (2023) 13:10141 under Springer Nature. The study focuses on developing a deep learning model for detecting childhood glaucoma based on periocular photographs, aiming to improve early diagnosis and referral decisions.

The proposed architecture is built on RepVGG, a convolutional neural network (CNN) optimized for structural re-parameterization. The model transforms a multi-branch training structure into a plain CNN during inference, allowing for high accuracy with reduced computational complexity. The RepVGG architecture enables faster inference speed compared to traditional CNNs while maintaining high accuracy in image classification tasks.

The methodology follows a structured pipeline beginning with data collection from Osaka University Hospital, where 65,534 periocular images from 8,125 pediatric patients were analyzed. After preprocessing, including image resizing, normalization, and augmentation (flipping, rotation, and contrast adjustment), the dataset was used to train and validate the RepVGG deep learning model. The model underwent fivefold cross-validation and was tested against diagnoses made by pediatric ophthalmologists and glaucoma specialists. The performance was evaluated using standard classification metrics (accuracy, sensitivity, specificity, and AUC scores). The model achieved an AUC of 0.95, accuracy of 87%, sensitivity of 75%, and specificity of 90%, surpassing human examiners in detecting childhood glaucoma in cases without corneal opacity.

Despite its promising results, the study highlights certain limitations. The model's performance is restricted to the dataset from a single institution, which may impact its generalizability across diverse populations. Additionally, the model struggles with other pediatric corneal diseases (e.g., Peters' anomaly, corneal dystrophies), which were not included in the dataset. The system also requires further external validation to confirm its reliability in real-world clinical settings. Future enhancements include expanding the dataset, integrating multimodal imaging (e.g., OCT scans), and optimizing deep learning architectures for better interpretability and robustness.

[10] Childhood glaucoma profile in a Southwestern Ethiopia tertiary care center - a retrospective study [2023]

The research paper titled "Childhood Glaucoma Profile in a Southwestern Ethiopia Tertiary Care Center: A Retrospective Study" was authored by Tarekegn Mulugeta, Guteta Gebremichael, and Sufa Adugna from Jimma University, Ethiopia. It was published in 2023 under the Childhood Glaucoma Research Network (CGRN) classification. The study focuses on the epidemiology, clinical features, and management strategies for childhood glaucoma in Southwestern Ethiopia.

The proposed architecture follows the CGRN classification system, categorizing childhood glaucoma into primary congenital glaucoma (PCG), juvenile open-angle glaucoma (JOAG), secondary glaucoma, and glaucoma suspect cases. The classification is based on factors such as age at onset, intraocular pressure (IOP), visual acuity, and structural optic nerve damage. The study emphasizes surgical interventions, particularly combined trabeculotomy and trabeculectomy (CTT), as the primary treatment for PCG cases.

The methodology involved a retrospective hospital-based cross-sectional study, analyzing medical records of 105 childhood glaucoma patients (181 eyes) diagnosed between September 2019 and August 2022 at Jimma Medical Center (JMC), Ethiopia. The study extracted patient data, including demographics, family history, clinical assessments (IOP, visual acuity), and treatment modalities (medication vs. surgery). IOP was measured using an iCare tonometer, and in uncooperative children, it was assessed under anesthesia. Data analysis was conducted using SPSS software, applying statistical models to assess treatment success rates.

Despite its significance, the study has several limitations. The late presentation of PCG cases remains a critical challenge, as most children are diagnosed after significant visual impairment has already occurred. Additionally, lack of proper documentation on family history and consanguinity limited the study’s ability to assess genetic factors. The study suggests future prospective research and larger multi-center studies to improve understanding and treatment outcomes for childhood glaucoma in resource-limited settings.

[11] Profile of Childhood Glaucoma Attending a Tertiary Eye Care Center in Northern India [2023]

The research paper titled "Profile of Childhood Glaucoma Attending a Tertiary Eye Care Center in Northern India" was authored by Suneeta Dubey, Kanika Jain, Julie Pegu, and Saptarshi Mukherjee. It was published in the Journal of Current Glaucoma Practice (JOCGP), Volume 17, Issue 2, April–June 2023, Pages 68–74. The study investigates the prevalence, clinical features, and management of childhood glaucoma in a tertiary eye care hospital in Northern India, emphasizing the importance of early diagnosis and intervention.

The study follows the Childhood Glaucoma Research Network (CGRN) classification, categorizing childhood glaucoma into primary congenital glaucoma (PCG), juvenile open-angle glaucoma (JOAG), and secondary glaucoma associated with acquired conditions. The majority of primary glaucoma cases were PCG (79%), while secondary glaucoma was mostly caused by trauma, post-vitreo-retinal surgery, and retinoblastoma.

The methodology involved a retrospective review of 405 children (584 eyes) diagnosed with childhood glaucoma between April 2014 and March 2019. Data collection included intraocular pressure (IOP) measurements, optic nerve cupping analysis, corneal diameter assessments, and visual field tests. Medical and surgical treatments were analyzed, with trabeculectomy and trabeculotomy being the primary surgical interventions for PCG, while secondary glaucoma cases were managed using a combination of medical and surgical approaches. The study used SPSS software for statistical analysis, summarizing data through mean values, standard deviation, and percentage distributions.

Despite its contributions, the study has several limitations. The retrospective design may introduce selection bias, and the single-center study limits generalizability to broader populations. Additionally, the lack of long-term follow-up data makes it challenging to assess treatment effectiveness over time. Future recommendations include multicenter studies, improved screening programs, and parental counseling to enhance childhood glaucoma management in resource-limited settings.

[12] Current Surgical Techniques for the Management of Pediatric Glaucoma: A Literature Review India [2023]

The research paper titled "Current Surgical Techniques for the Management of Pediatric Glaucoma: A Literature Review" was authored by Zeynep Aktas and Gokcen Deniz Gulpinar Ikiz. It was published in Ophthalmology, in 2023. This review provides a comprehensive analysis of surgical interventions for pediatric glaucoma, emphasizing advancements in techniques. The paper discusses various surgical architectures, including goniotomy, trabeculotomy, trabeculectomy, and

glaucoma drainage devices (GDDs). Goniotomy and trabeculotomy are preferred for primary congenital glaucoma (PCG), while trabeculectomy with adjunctive mitomycin C is recommended for older children. The use of glaucoma drainage devices (GDDs) is suggested for refractory cases, but they carry risks such as hypotony and tube-related complications. Additionally, minimally invasive glaucoma surgery (MIGS) is gaining attention as a potential adjunctive or standalone treatment.

The methodology involves a literature review of the latest advancements in pediatric glaucoma surgeries. The study evaluates the success rates, complications, and effectiveness of different surgical interventions by analyzing past research findings. The focus is on assessing long-term outcomes, postoperative complications, and patient suitability for specific procedures. The review also discusses the role of adjunctive therapies like mitomycin C in enhancing surgical success.

Despite the advancements, the study highlights several limitations. Goniotomy and trabeculotomy have variable success rates depending on patient age and disease severity. Trabeculectomy carries risks such as bleb-related infections, while GDDs have higher complication rates in children than adults. MIGS, although promising, lacks extensive long-term studies. Future research is needed to explore enhanced surgical techniques, patient-specific treatment approaches, and long-term monitoring strategies to improve pediatric glaucoma management.

[13] UNet Mobilenetv2: Medical Image Segmentation Using Deep Neural Network (DNN) [2023]

The research paper titled "UNET MobileNetV2: Medical Image Segmentation Using Deep Neural Network (DNN)" was authored by Bikash Chandra Bag, Hirak Kumar Maity, and Chaitali Koley. It was published in the Journal of Mechanics of Continua and Mathematical Sciences (JMCMS), Volume 18, Issue 1, January 2023, Pages 21-29. The paper focuses on polyp segmentation in colonoscopy images using a deep learning model, aiming to improve the detection of abnormal tissue growth and reduce the chances of missing cancerous polyps.

The proposed architecture is based on U-Net and MobileNetV2, combining the strengths of U-Net’s encoder-decoder structure for segmentation and MobileNetV2’s lightweight depth-wise separable convolutions for efficient feature extraction. The U-Net structure consists of an encoder-decoder path, where MobileNetV2 serves as the encoder. The model is trained on the CVC-612 dataset, which contains 612 polyp images with ground truth segmentations.

The methodology includes several key steps. First, data preprocessing is applied, including pixel normalization and data augmentation to improve model generalization. The U-Net-MobileNetV2 model is trained using TensorFlow and Keras frameworks, with a learning rate of 0.0001 and a batch size of 16. The evaluation is conducted using Dice Coefficient and Intersection over Union (IoU) metrics, achieving a Dice score of 89.71% and an IoU score of 81.64%, outperforming other state-of-the-art models

Despite its high accuracy, the study identifies certain limitations. The model's performance depends on high-quality colonoscopy images, and real-world applications may face challenges due to variations in lighting, camera settings, and motion artifacts. Additionally, the small dataset size (CVC-612) limits the model's generalization ability. Future improvements include training on larger datasets, optimizing segmentation for different medical imaging modalities, and integrating real-time processing for clinical use.

[14] An Iterative Constraint Spectral Model for Ophthalmic Disease Detection Using Transfer Learning [2022]

The research paper titled "An Iterative Constraint Spectral Model for Ophthalmic Disease Detection Using Transfer Learning" was authored by Dr. Shanmuganathan C, Anish T P, Mary Joseph, and Thamizharasi M from SRM Institute of Science and Technology and RMK College of Engineering and Technology, India. It was published in the 2022 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI), IEEE, with ISBN 979-8-3503-3384-8/22. The study presents a deep learning-based approach using transfer learning for detecting ophthalmic diseases using retinal fundus images to assist in automated diagnosis.

The proposed architecture is based on Convolutional Neural Networks (CNNs) integrated with transfer learning. The model utilizes a pre-trained CNN (such as ResNet or VGG) as a feature extractor, reducing the need for extensive labeled data. The study employs a novel loss function to improve classification performance. The classification process includes feature extraction using CNN, followed by classification using a Support Vector Machine (SVM) for improved accuracy.

The methodology involves multiple steps, starting with image preprocessing, including grayscale conversion, normalization, data augmentation, and noise removal to enhance image quality. The dataset is then split into training and testing sets, where transfer learning is applied to fine-tune the pre-trained CNN model for ophthalmic disease detection. The model is trained using supervised learning algorithms, and performance is evaluated based on classification accuracy, precision, recall, and F1-score. The study compares the proposed deep learning model against existing datasets and traditional machine learning approaches, demonstrating improved efficiency and reduced diagnosis time.

Despite its promising performance, the study identifies several limitations. The model's accuracy is influenced by image quality variations and the availability of large, diverse datasets. Additionally, manual annotations by ophthalmologists are required for training, which is a time-consuming process. Another limitation is that the model struggles with noisy or low-resolution images, leading to false positives or misclassifications. Future work includes expanding the dataset, integrating real-time deployment in clinical settings, and improving model interpretability using explainable AI techniques.

[15] Gaze Exploration Index (GE i)-Explainable Detection Model for Glaucoma [2022]

The research paper titled "Gaze Exploration Index (GE-i)-Explainable Detection Model for Glaucoma" was authored by Sajitha Krishnan, J. Amudha, and Sushma Tejwani. It was published in IEEE Access, Volume 10, Pages 74334-74349, July 2022. This study introduces an explainable AI-based Computer-Aided Detection system that analyzes eye gaze patterns and visual field loss in glaucoma patients using an eye-tracking device.

The proposed architecture is based on deep neural networks (DNNs) combined with eye-tracking data analysis. The system utilizes the Gaze Exploration Index (GE-i), which incorporates visual exploration tasks to assess eye movement behavior in glaucoma patients. The architecture consists of four main modules: (1) Visual Exploration Tasks, which assess eye movement patterns during various activities, (2) Estimation of Extensive Gaze and Performance (EXGP) Module, which extracts 28 gaze and performance features, (3) EXGP Feature Analysis, which validates eye gaze data using statistical tests and clinical measures, and (4) Explainable Detection Model, which employs a Deep Neural Network (DNN) to classify patients as glaucomatous or normal based on gaze behavior.

The methodology follows a structured pipeline that begins with data acquisition from 117 participants (50 glaucoma patients, 48 normal, and 19 excluded cases) at Narayana Nethralaya, Bengaluru. The system collects eye-tracking data during three tasks: dot detection, visual search, and free-viewing. Features such as fixation count, saccade rate, reaction time, and scan path length are extracted. The EXGP module processes eye-tracking data, which is then fed into a DNN classifier trained using SHAP (SHapley Additive exPlanations) feature selection to determine the most significant parameters for glaucoma detection. The final GE-i screening index is computed using a weighted regression model.

Despite its advantages, the study highlights several limitations. The model is trained on a relatively small dataset, which may limit generalizability. Additionally, individual differences in gaze behavior and compensatory eye movements among young glaucoma patients introduce variability. Another limitation is the requirement for eye-tracking hardware, which may not be readily available in all clinical settings. Future improvements include expanding the dataset, refining the GE-i model, and integrating multimodal ophthalmic imaging (OCT and fundus photography) for enhanced glaucoma screening.

[16] Melanoma image classification based on MobileNetV2 network [2022]

The research paper titled "Melanoma Image Classification Based on MobileNetV2 Network" was authored by Rarasmaya Indraswari, Rika Rokhana, and Wiwiet Herulambang. It was published in Procedia Computer Science, Volume 197, Pages 198–207, 2022, under Elsevier B.V. The study explores deep learning-based melanoma classification using the MobileNetV2 architecture for benign and malignant lesion detection. The motivation behind the study is to enhance computer-aided diagnosis (CAD) tools for skin cancer detection.

The proposed architecture is built upon MobileNetV2, a lightweight convolutional neural network (CNN) designed for efficient computation. The head model includes a global average pooling layer, two fully connected layers, and a softmax activation function to classify melanoma images. The MobileNetV2 backbone utilizes depthwise separable convolutions making it ideal for mobile-based applications.

The methodology involves transfer learning, where a pre-trained MobileNetV2 model (trained on ImageNet) is fine-tuned using four melanoma datasets: ISIC-Archive, ISBI 2016, MED-Node, and PH2 Database. The data preprocessing includes image resizing (224×224), augmentation (rotation, flipping, and zooming), and normalization. The Adam optimizer and binary cross-entropy loss function are used to train the model, with 20 epochs and a batch size of 32. Performance evaluation metrics include accuracy, sensitivity, specificity, and precision. The experimental results demonstrate that MobileNetV2 achieves up to 85% accuracy on ISIC-Archive, outperforming alternative CNN models such as ResNet50V2, InceptionV3, and InceptionResNetV2.

Despite its advantages, the study acknowledges several limitations. The model struggles with imbalanced datasets, leading to lower sensitivity for malignant cases in datasets like ISBI 2016 and PH2. Additionally, the reliance on high-quality dermoscopic images restricts its applicability in real-world scenarios where image quality varies. Future research aims to address class imbalance, explore weighted loss functions, and integrate real-time deployment for mobile applications.

[17] Deep Learning-Based Glaucoma Detection with Cropped Optic Cup and Disc and Blood Vessel Segmentation [2022]

The research paper titled "Deep Learning-Based Glaucoma Detection with Cropped Optic Cup and Disc and Blood Vessel Segmentation" was authored by Mir Tanvir Islam, Shafin T. Mashfu, Abrar Faisal, Sadman Chowdhury Siam, Intisar Tahmid Naheen, and Riasat Khan. It was published in IEEE Access, Volume 10, Pages 2828-2841, January 2022. The study presents a deep learning-based approach for glaucoma detection using retinal fundus images, emphasizing optic cup and disc segmentation as well as blood vessel segmentation. The main objective of the research is to develop an automated and efficient glaucoma classification technique that reduces human errors and dependency on expensive clinical examinations.

The proposed architecture incorporates various convolutional neural networks (CNNs), including EfficientNet-b3, MobileNet, DenseNet, and GoogLeNet, to classify fundus images as either glaucomatous or non-glaucomatous. The researchers created two datasets: Dataset-1, which consists of cropped optic cup and disc images, and

Dataset-2, which includes blood vessel-segmented fundus images obtained through a U-Net-based segmentation approach. The EfficientNet-b3 model demonstrated the best classification accuracy of 90.52% on Dataset-1, while MobileNet v3 achieved 83.48% accuracy on Dataset-2.

The methodology follows a structured process beginning with dataset preparation, where 634 color fundus images were collected and annotated by ophthalmologists at Bangladesh Eye Hospital. These images were preprocessed using grayscale conversion, contrast enhancement, and augmentation techniques (rotation, flipping, and brightness adjustment) to improve model robustness. The second step involved segmentation, where the U-Net model was trained on the High-Resolution Fundus Image (HRF) dataset to extract blood vessel features from fundus images. Finally, deep learning-based classification was performed using various CNN models, optimized with Adam optimizer, batch normalization, and cross-entropy loss function, with the evaluation based on accuracy, precision, recall, F1-score, and ROC-AUC metrics.

Despite its effectiveness, the study presents several limitations. The dataset size is relatively small, which may impact model generalization in real-world scenarios. Additionally, variations in image quality and imbalanced class distribution can lead to misclassification in some cases. The requirement for manual annotations by ophthalmologists for training the model also poses a challenge. Future work suggests expanding the dataset, improving segmentation techniques, and incorporating additional ophthalmic imaging modalities, such as Optical Coherence Tomography (OCT), for more accurate and comprehensive glaucoma diagnosis.

[18] Detection of Glaucoma on Fundus Images Using Deep Learning on a New Image Set Obtained with a Smartphone and Handheld Ophthalmoscope [2022]

The research paper titled "Detection of Glaucoma on Fundus Images Using Deep Learning on a New Image Set Obtained with a Smartphone and Handheld Ophthalmoscope" was authored by Clerimar Paulo Bragança, José Manuel Torres, Christophe Pinto de Almeida Soares, and Luciano Oliveira Macedo. The study introduces a novel approach for glaucoma detection using deep learning models on fundus images acquired with a smartphone-attached panoptic ophthalmoscope, aiming to enhance early glaucoma screening accessibility.

The proposed architecture utilizes Convolutional Neural Networks (CNNs), including DenseNet, MobileNet, ResNet50v2, InceptionV3, and Xception, trained through transfer learning. The ensemble model, formed by combining the outputs of individual classifiers, enhances classification accuracy.

The methodology follows a structured pipeline: Data Acquisition – capturing images using a smartphone-attached panoptic ophthalmoscope; Preprocessing – cropping images to focus on the optic disc, contrast enhancement, and augmentation techniques; Model Training – applying transfer learning on CNNs, freezing pre-trained layers while training new layers on the BrG dataset; Evaluation – five-fold cross-validation, computing accuracy, sensitivity, specificity, and area under the ROC curve (AUC). The final ensemble model achieved a classification accuracy of 90%, showing promising results for smartphone-based glaucoma detection.

Despite its success, the study highlights several limitations. The dataset is limited in diversity, as it originates from a single region, which may reduce model generalizability. Additionally, external lighting conditions and image quality variations affect classification performance. The use of a panoptic ophthalmoscope presents resolution limitations compared to conventional fundus cameras. Future research aims to expand the dataset, refine the segmentation process, and incorporate additional ophthalmic imaging modalities (such as Optical Coherence Tomography - OCT) for improved glaucoma diagnosis.

[19] Identifying Those at Risk of Glaucoma: A Deep Learning Approach for Optic Disc and Cup Segmentation and Their Boundary Analysis [2022]

The paper titled "Identifying Those at Risk of Glaucoma: A Deep Learning Approach for Optic Disc and Cup Segmentation and Their Boundary Analysis" was authored by Jongwoo Kim, Loc Tran, Tunde Peto, and Emily Y. Chew. The study presents a deep learning-based method for automated segmentation of the optic disc and cup from fundus images to estimate key measures for glaucoma screening. The primary objective is to develop an efficient and reliable screening tool for early detection of glaucoma, reducing reliance on manual clinical assessments.

The proposed architecture consists of a three-step approach utilizing Mask R-CNN and Multiscale Average Pooling Net for segmentation. The first step involves Region of Interest detection using Mask R-CNN, which identifies the optic disc area in fundus images. The second step employs MAPNet, a deep learning-based segmentation algorithm, to segment the optic disc and cup using multiple feature extraction layers. The third step calculates key glaucoma indicators, such as the cup-to-disc ratio, CD area ratio, and rim-to-disc (RD) ratio, to evaluate potential glaucomatous damage.

The methodology follows a structured pipeline: Data Acquisition, using publicly available datasets (REFUGE and RIGA); Preprocessing, including image normalization, augmentation, and ROI extraction; Model Training, applying Mask R-CNN for ROI detection and MAPNet for segmentation, followed by ensemble learning to improve segmentation accuracy; and Performance Evaluation, using metrics such as Jaccard Index, Sensitivity, and Specificity. The results indicate that the proposed approach significantly enhances segmentation reliability and diagnostic precision.

Despite its advantages, the study highlights several limitations. The high computational cost of deep learning models like Mask R-CNN requires significant hardware resources, making real-time implementation challenging. Additionally, the dataset imbalance affects segmentation accuracy for certain optic disc and cup shapes, leading to over-segmentation or under-segmentation errors.

[20] Automated glaucoma detection from fundus images using wavelet-based denoising and machine learning [2021]

The research paper titled "Automated Glaucoma Detection from Fundus Images Using Wavelet-Based Denoising and Machine Learning" was authored by Sibghatullah I. Khan, Shruti Bhargava Choubey, Abhishek Choubey, Abhishek Bhatt, Pandya Vyomal Naishadhkumar, and Mohammed Mahaboob Basha. It was published in Concurrent Engineering: Research and Applications, spanning pages 1–12, in 2021. The study proposes a computer-aided detection (CAD) system that utilizes machine learning and wavelet-based denoising techniques for automated glaucoma detection in fundus images, aiming to enhance early screening and classification of the disease.

The proposed architecture follows a five-stage image processing approach, incorporating spatial feature extraction, wavelet-based denoising, feature selection, and classification using Least Squares Support Vector Machine (LS-SVM). The RGB components of fundus images are individually processed, and key image features such as contrast, homogeneity, entropy, standard deviation, and energy are extracted. A Neighborhood Component Analysis (NCA) method is applied to select the most relevant features before feeding them into the LS-SVM classifier, which is tested with multiple kernel functions, including linear, polynomial, and radial basis function (RBF) kernels.

The methodology consists of preprocessing fundus images from the MIAG dataset (455 images: 200 glaucomatous and 255 normal images), applying Discrete Wavelet Transform (DWT) for noise reduction, and performing feature extraction across RGB channels. The images undergo classification using LS-SVM, where the performance is evaluated using accuracy, sensitivity, specificity, and Matthews Correlation Coefficient (MCC). The experimental results show that LS-SVM with an RBF kernel achieves the highest classification accuracy of 91.22%, outperforming other existing approaches.

Despite its promising results, the study presents several limitations. The dataset size is relatively small, restricting its generalizability to real-world clinical applications. Additionally, the model's performance depends on high-quality images, meaning it may struggle with low-resolution or noisy fundus images. Furthermore, the study does not compare deep learning approaches, which could potentially offer higher accuracy. Future research aims to expand the dataset, incorporate deep learning-based segmentation methods, and improve real-time diagnostic capabilities for clinical use.

2.1 Survey Findings

Challenges Identified:

1. Variability in Image Quality

- Fundus and OCT image quality is affected by differences in lighting, resolution, and imaging devices across datasets.
- Patient movement and improper focus during image capture introduce

blurring and artifacts, reducing detection accuracy.

- Image contrast variations make it difficult to differentiate between optic disc, cup, and surrounding retinal structures.

2. Limited Annotated Datasets

- Deep learning models require large, high-quality labeled datasets, but annotated glaucoma images are scarce.
- Manual labeling by ophthalmologists is time-consuming and expensive, leading to dataset imbalance.
- Some models perform well on specific datasets but fail when tested on new populations, limiting real-world effectiveness.

3. Lack of Generalization and Robustness

- Most models are trained on limited datasets from specific populations, affecting their ability to generalize across ethnicities and age groups.
- Variations in eye conditions, such as cataracts, diabetic retinopathy, or myopia, interfere with glaucoma detection, leading to false positives.
- Differences in optic disc size and shape across patients require adaptive learning models, which many studies lack.

4. Limited Clinical Validation and Real-World Testing

- While many models show high accuracy on benchmark datasets, few have been tested in real-world clinical environments.
- Regulatory approval for AI-based glaucoma screening is still in its early stages, limiting adoption.
- There is a lack of integration with hospital systems and electronic medical record (EMR) for real-time diagnosis.

Chapter 3

REQUIREMENT SPECIFICATION

3.1 Stakeholders

The potential stakeholders for the project are:

- Team Members
- Project Guide
- End users
- Technology providers / Tools used
- Medical and research Communities
- Faculty Department
- College Management

3.2 Functional Requirements

1. **Image Input:** The system must accept retinal images (fundus and OCT images) as input**.**
2. **Data Preprocessing:** Preprocess the input images through:
    - Resizing
    - Normalization
    - Data augmentation (like rotation, flipping, brightness adjustment)
3. **Classification:** Classify the input images into two categories such as Healthy and Glaucoma - affected.
4. **Training and Testing:** Support both the training phase and testing phase.
5. **Performance Evaluation:** Evaluate system performance based on Accuracy, Precision, Recall, F1-Score and AUC-ROC.
6. **Real-time Detection:** Provide real-time diagnosis once a new image is uploaded.
7. **User Interface:** Develop a user-friendly interface that allows:
    - Uploading retinal images
    - Displaying classification results
    - Showing confidence scores for predictions.

8. **Data Storage:** Securely store the images and results for future use or auditing purposes.
9. **Error Handling:** Gracefully handle errors during input upload, preprocess - ing, model prediction, or display.

3.3 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Non-functional requirements are often called qualities of a system.

1. Scalability: The system should be able to handle increasing amounts of data without significant degradation in performance.
2. Efficiency: Ensure low latency for predictions and optimize training time using lightweight architectures like MobileNetV2.
3. Reliability: The system should perform consistently under varying conditions and with different datasets.
4. Usability: The interface should be intuitive and accessible for non-technical users, including clinicians and researchers.
5. Portability: Support deployment on various platforms, including cloud services and edge devices.

3.4 External Interface Requirements

3.4.1 Hardware Requirements

- **Processor:** Intel Core i5 or higher (for handling image processing and model training tasks).
- RAM: Minimum 8 GB (16 GB recommended for optimal performance during deep learning model training).
- Storage: At least 100 GB free space for storing dataset and model files.
- GPU: Nvidia GPU (e.g., GTX 1050 or higher) for local model training (VS Code can be used).

3.4.2 Software Requirements

- **Operating System:** Windows

- **Visual Studio:** Lightweight code editor used for writing, testing, and debugging Python scripts in the project with GPU Support.
- **Python 3.x:** Programming language for model development.
- **TensorFlow/Keras:** Deep learning libraries for building and training the MobileNetV2 model.

3.4.2.1 Visual Studio Code

Visual Studio Code, commonly known as VS Code, is a lightweight, open-source code editor developed by Microsoft. It provides a flexible and user-friendly environment for writing, editing, and debugging code across various programming languages, including Python. With features such as IntelliSense for smart code completion, integrated terminal, Git version control, and support for a wide range of extensions, VS Code significantly enhances productivity. Its seamless integration with Jupyter notebooks and remote development tools makes it ideal for developing and testing deep learning models locally before deploying them to cloud platforms. Due to its efficiency, adaptability, and rich ecosystem, VS Code is widely adopted by developers, data scientists, and researchers in machine learning and artificial intelligence.

3.4.2.2 Tensor Flow

TensorFlow, an open-source machine learning framework developed by Google, has emerged as a leading platform for building and deploying deep learning models. It provides a flexible and scalable environment for developing various machine learning applications, ranging from image recognition and natural language processing to reinforcement learning and generative modeling. In this comprehensive software description, we delve into the architecture, features, and capabilities of TensorFlow in Python, exploring its core components, workflow, and best practices for effective utilization.

3.4.2.3 MobileNetV2

MobileNetV2 is a lightweight deep learning architecture specifically designed for mobile and embedded vision applications. Developed by Google researchers, MobileNetV2 builds upon the success of its predecessor, MobileNetV1, by introducing novel architectural improvements aimed at achieving higher efficiency, accuracy, and speed while maintaining a compact model size. In this comprehensive software description, we explore the design principles, architecture, features, and applications of MobileNetV2, highlighting its significance in the field of computer vision and its widespread adoption in various real-world applications.

3.4.2.4 HTML

HTML (HyperText Markup Language) is the standard language used for creating and structuring content on the web. It provides the basic building blocks for web pages by defining an elements such as headings, paragraphs, links, images, and forms. In this project, HTML is used to design the structure of the user interface, enabling users to upload images and view an glaucoma prediction results in a clear and accessible format.

3.4.2.5 CSS

CSS (Cascading Style Sheets) is a styling language used to control the visual presentation of HTML elements. It allows developers to apply styles such as colors, fonts, layouts, and responsiveness to web pages. In the context of this project, CSS is used to enhance the user interface by ensuring a clean, professional, and responsive design that improves the user experience when interacting with the glaucoma detection system.

3.4.2.6 JavaScript

JavaScript is a high-level scripting language that enables dynamic behavior and interactivity in web applications. It is widely used to manipulate the Document Object Model (DOM), handle events, and communicate with backend servers asynchronously. In this project, JavaScript plays a crucial role in managing form submissions, displaying prediction results, and enhancing the responsiveness of the user interface, making the web-based diagnostic tool interactive and user-friendly.

Chapter 4

PROBLEM STATEMENT

Pediatric glaucoma is a rare but serious eye condition that, if left undiagnosed, can lead to irreversible blindness. Traditional diagnostic methods rely on manual examination by ophthalmologists, which can be subjective, time-consuming, and dependent on clinical expertise. The lack of standardized and automated screening tools often results in delayed detection, reducing the chances of effective early intervention. Additionally, the limited availability of large-scale pediatric glaucoma datasets further complicates the development of reliable diagnostic systems.

To address these challenges, this project proposes the development of an automated deep learning-based system for detecting pediatric glaucoma using retinal fundus and Optical Coherence Tomography (OCT) images. By leveraging Convolutional Neural Networks (CNNs), the system aims to accurately identify glaucoma-specific patterns, ensuring faster and more accessible diagnosis. The integration of transfer learning with pre-trained models like VGG16 and ResNet50 enhances the model’s accuracy even with limited pediatric glaucoma datasets. This solution is particularly beneficial for resource-limited settings, where access to specialized ophthalmologists is scarce, ultimately contributing to early detection, timely treatment, and improved clinical outcomes.

4.1 Objectives

- Develop a deep learning-based system using CNNs for automatic detection of pediatric glaucoma from retinal fundus and OCT images.
- Leverage transfer learning with pre-trained models like VGG16 and ResNet50 to handle limited pediatric glaucoma data.
- Use data augmentation techniques to improve model generalization and performance.
- Evaluate the model's accuracy and reliability using metrics such as precision, recall, and F1-score.

4.2 Expected Output

- Enables early and accurate detection of Pediatric Glaucoma using retinal and OCT images.
- Reduces dependency on clinical expertise by providing automated diagnosis.
- Improves diagnostic speed and consistency through deep learning-based image analysis.
- Achieves high performance with key metrics: Accuracy, Precision, Recall, F1-Score and AUC-ROC
- Utilizes MobileNetV2 for lightweight, efficient model deployment suitable for mobile and cloud platforms.
- Provides a user-friendly interface for healthcare professionals to upload images and receive real-time results.
- Minimizes costs by leveraging open-source tools (TensorFlow, VS Code) and reducing the need for specialized equipment.
- Maintains patient data security and privacy in compliance with healthcare standards.
- Facilitates scalability and accessibility, especially in resource-limited or remote areas.

Chapter 5

SYSTEM ANALYSIS AND DESIGN

5.1 System Analysis:

The Pediatric Glaucoma Detection System involves understanding the requirements, constraints, and objectives of the project to develop an effective solution. This includes identifying the need for early and accurate detection of pediatric glaucoma, a condition that can lead to irreversible vision loss if not diagnosed timely. The system focuses on automating the detection process using deep learning, thereby minimizing reliance on manual clinical diagnosis and expensive equipment. Additionally, system analysis involves studying existing diagnostic methods, analyzing their limitations, conducting feasibility studies, and evaluating the technical, operational, and financial aspects of implementing the proposed system. Through systematic analysis, the project aims to ensure that the final system meets clinical needs, is technically feasible, affordable, and provides a reliable solution for enhancing early glaucoma diagnosis in children.

This analysis encompasses several key areas:

- **Identification of Requirements:** Understand the need for an automated deep learning system that can detect pediatric glaucoma accurately from retinal fundus and OCT images. Identify the limitations of traditional methods and the necessity for quicker, scalable solutions that can assist healthcare professionals.

- **Assessment of Objectives:** Define the primary objective of providing early and accurate glaucoma detection using deep learning. Secondary objectives include minimizing false positives/negatives, enabling real-time predictions, and maintaining high model generalization across different datasets.

- **Functionality and Features:** Define desired features, including the ability to upload retinal images through a user-friendly interface, perform real-time diagnosis, and display confidence scores. Incorporate preprocessing, image normalization, data augmentation, and lightweight MobileNetV2-based prediction to ensure efficiency and reliability.

- **Study of Existing Solutions:** Review existing glaucoma detection methods and an machine learning models, analyzing their strengths and limitations. Most current solutions target adult glaucoma and require expensive imaging devices and specialist interpretation.

- **Feasibility Studies:** Conduct feasibility studies to assess technical feasibility through the use of frameworks like TensorFlow, Keras, and Visual Studio for training. Operational feasibility is evaluated through the system's simple interface and easy integration into clinical practice. Financial feasibility is considered by utilizing open-source tools and cloud platforms to minimize costs, making the system affordable and scalable.

- **Technical Evaluation:** Evaluate technical aspects such as hardware requirements (basic GPUs, cloud access), software dependencies (Python, TensorFlow, Keras, VS Code), and integration with future healthcare platforms. Focus on factors like model efficiency, prediction speed, data privacy, and adaptability to different clinical environments.

5.2 Detailed Design:

Detailed design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. The Purpose of this design document is to explore the logical view of architecture design, data flow diagrams, sequence diagram and an overview of the proposed system for performing the operations such as signature capturing, pre-processing, feature extraction and validation which when combined to give the desired output. The design activity module consists three outputs.

- Architecture design.
- High level design.
- Low level design.

5.3 System Architecture

The system architecture of the pediatric glaucoma detection project describes how different parts of the system work together to achieve the final goal. It starts with collect the retinal images, which are stored in a database. These images are then preprocessed by resizing, normalizing, and enhancing them to improve quality. After preprocessing, the data is divided into training and testing sets. A deep learning model, specifically MobileNetV2, is trained on the training data to learn how to detect glaucoma. Once trained, the model makes predictions on the test images. Finally, the system compares the predictions with actual results to calculate accuracy and performance.

The system architecture of our Deep Learning Approach To Detect Pediatric Glaucoma is given below in fig 5.3.1. The system architecture consists of the following steps:

1. **Database:** Collects and stores data relevant for training and testing.
2. **Preprocessing:** Input data is pre-processed to remove noise, normalize features, and prepare the dataset for model training.
3. **Build Training and Test Datasets:** The dataset is split into two subsets:
   - **Training Dataset:** Used to train the deep learning model.
   - **Test Dataset:** Used to evaluate the model's performance.
4. **Model Learning:** Deep learning algorithms are applied to the training dataset to learn and create the model.
5. **Prediction:** The learned model is applied to the test dataset to generate predictions.
6. **Accuracy Comparison and Conclusions:** Predictions are compared to actual results to evaluate the model's accuracy and Conclusions are drawn based on the model's performance metrics.

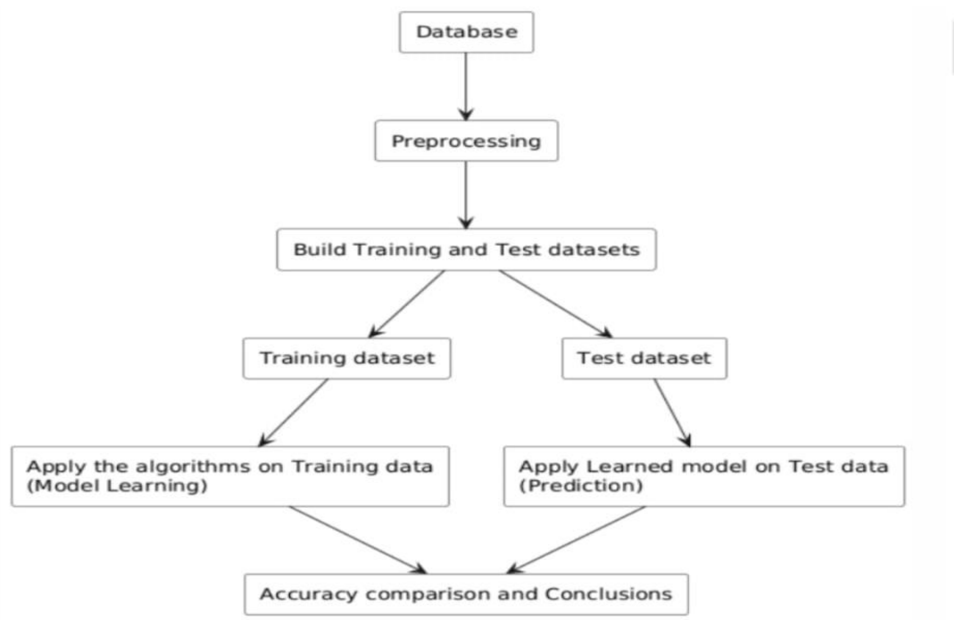


Fig 5.3.1: System Architecture

5.4 High Level Design

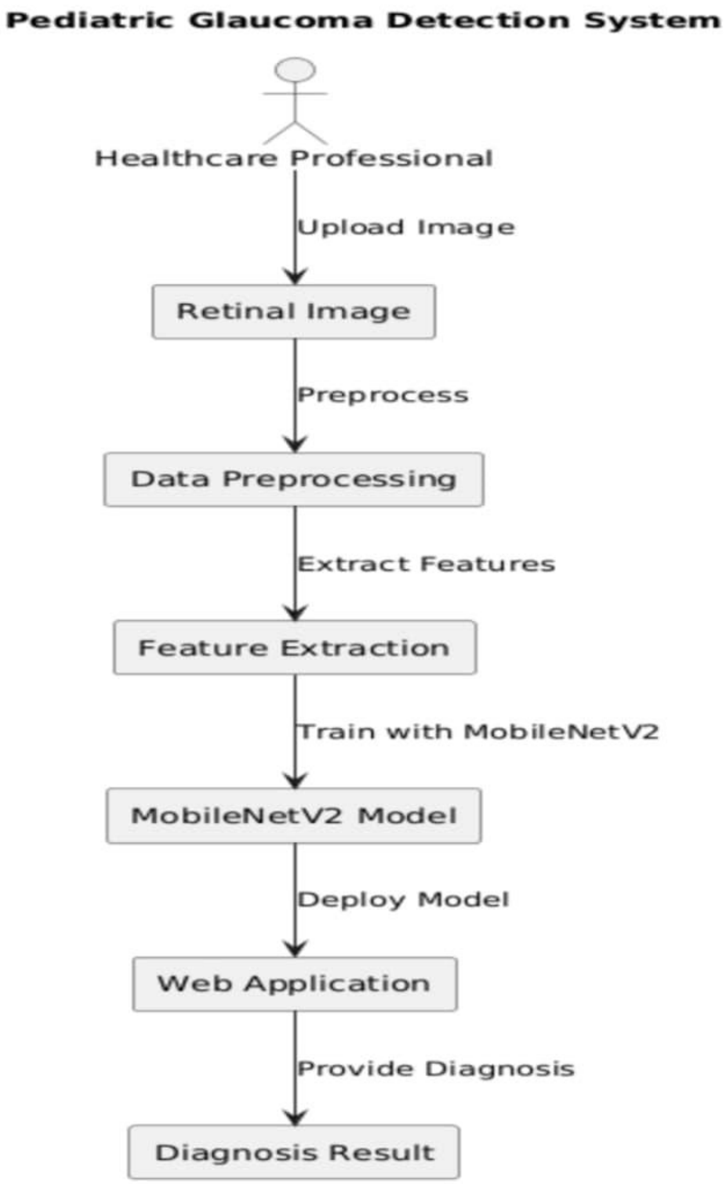


Fig 5.4.1 Proposed System

The proposed system is a deep learning-based solution aimed at addressing the critical challenge of detecting pediatric glaucoma, a condition that requires early diagnosis to prevent irreversible vision loss. Unlike traditional diagnostic methods, which often rely heavily on specialized clinical expertise and are time-intensive, this system leverages the power of artificial intelligence to provide an automated, efficient, and scalable solution. By analyzing retinal images, the system detects subtle signs of glaucoma that may not be apparent to the human eye, thus enhancing diagnostic accuracy and consistency.

The system utilizes MobileNetV2, a lightweight and efficient convolutional neural network known for its high performance in image classification tasks. MobileNetV2's architecture is particularly well-suited for medical applications where computational efficiency and speed are critical. To further optimize the model, transfer learning is employed, leveraging pre-trained weights from ImageNet. This reduces the need for extensive training and ensures that the system can achieve high accuracy even with a limited dataset.

Data preprocessing plays a pivotal role in the proposed system. Retinal images are resized to a uniform dimension, normalized to standardize pixel values, and augmented to increase dataset diversity. These steps ensure the model can generalize effectively to new, unseen data, improving its robustness in real-world applications. The use of techniques like rotation, flipping, and brightness adjustments in augmentation helps the model learn from varied perspectives of the same data, which is particularly important in detecting complex medical conditions like glaucoma.

A user-friendly interface is a key component of the system, designed to facilitate its adoption by healthcare professionals. The interface allows clinicians to upload retinal images and receive real-time predictions along with confidence scores. This not only simplifies the diagnostic process but also empowers medical practitioners with accurate and timely insights, aiding in early intervention. The system's deployment on a cloud or server ensures scalability, making it accessible to multiple users across different locations while maintaining quick response times.

Overall, the proposed system combines advanced deep learning techniques with practical design considerations to provide a comprehensive solution for pediatric glaucoma detection. By automating the diagnostic process, it not only reduces the dependency on specialized clinical expertise but also makes early detection more accessible, particularly in resource-limited settings. This system represents a significant step forward in leveraging technology to improve healthcare outcomes for children at risk of glaucoma.

5.5 Low Level Design

The proposed Deep Learning Approach To Detect Pediatric Glaucoma give above in fig 5.4.1. The system consists of key modules such as:

1. **Data Collection Module**
   - **Purpose:** Collects and organizes retinal image datasets from reliable sources, such as the Eagle dataset.
   - **Key Features:**
     - Data acquisition from medical repositories.
     - Annotation and categorization of images into classes (e.g., healthy, glaucoma).
     - Storage and indexing for easy access.
2. **Data Preprocessing and Augmentation Module**
   - **Purpose:** Prepares the raw retinal images for model training by ensuring uniformity.
   - **Key Features:**
     - Image resizing (e.g., to 224x224 pixels).
     - Normalization of pixel values to standardize input data.
     - Augmentation techniques (e.g., rotation, flipping, brightness adjustment) to improve model generalization.
3. **Model Development Module**
   - **Purpose:** Implements the deep learning model for glaucoma detection.
   - **Key Features:**
     - Selection of MobileNetV2 architecture for efficient image classification.
     - Integration of transfer learning using pre-trained ImageNet weights.
     - Fine-tuning of the model to adapt it for pediatric glaucoma detection.
4. **Training and Evaluation Module**
   - **Purpose:** Trains the model and evaluates its performance using appropriate metrics.
   - **Key Features:**
     - Model training with optimal hyperparameters (batch size, epochs, learning rate).
     - Evaluation metrics, such as accuracy, precision, recall, F1-score, and AUC-ROC.

- Validation and test sets to monitor performance and prevent overfitting.

5. **User Interface (UI) Module**

- **Purpose:** Ensures ease of use for healthcare professionals interacting with the system.
- **Key Features:**
  - Simple interface for image uploads.
  - Display of real-time predictions and confidence levels, Accessibility via web or mobile platform.

Chapter 6

IMPLEMENTATION

6.1 Overview

The goal of the programming or implementation phase is to translate the design of the system produced during the design phase into code in a given programming language, which can be executed by the computer and that performs the computation specified by the design. Implementation of any software is always preceded by important decisions regarding selection of the platform, the language used etc. These decisions are often influenced by several factors such as real environment in which the system works, the speed that is required, the security concerns, other implementation specific details etc. This chapter discusses the step-by-step process for developing the automated pediatric glaucoma detection system. The implementation involves preparing the dataset, building the deep learning model, preprocessing the data, training the model, and deploying the system with a user-friendly interface.

The system is composed of several interconnected steps:

1. **System Model:**

- **Data Acquisition and Preprocessing**: Collect retinal fundus and OCT images, followed by resizing, normalization, and data augmentation.
- **Model Design:** Utilize MobileNetV2 architecture with transfer learning for efficient image classification.
- **Training:** Train the model using the Eagle dataset on Visual Studio with GPU support.
- **Evaluation:** Assess the model’s performance using metrics like accuracy, precision, recall, and F1-score.
- **Deployment:** Implement the system with an interactive UI using HTML, CSS, JavaScript allowing users to upload images for real-time diagnosis.

2. **Data Preparation:**

The Eagle dataset is the primary source of annotated retinal images. Key preprocessing steps include:

- **Resizing:** Ensures uniform input dimensions for the model (e.g., 224x224 pixels).
- **Normalization:** Scales pixel values to a [0, 1] range for faster convergence.
- **Data Augmentation:** Enhances the dataset by applying transformations like

rotation, flipping, and zooming to improve generalization.

3. **Model Implementation:**

The MobileNetV2 architecture was chosen for its lightweight design and efficiency in image classification tasks. Key implementation steps include:

- Model Initialization:
  - Load the pre-trained MobileNetV2 model with ImageNet weights.
  - Modify the output layer to match the binary classification task (healthy or glaucoma-affected).
- Optimizer and Loss Function:
  - Use the Adam optimizer for efficient gradient descent.
  - Binary cross-entropy as the loss function.
- Training Configuration:
  - Batch size: 26
  - Epochs: 50 (with early stopping to prevent overfitting)
  - Learning rate: 0.001

4. **Training and Validation:**

The training process involves feeding the pre-processed images into the model and optimizing its weights to minimize loss. Validation is performed on a separate test set to evaluate model generalization. The training process is monitored using**:**

- Accuracy: Measures correct predictions.
- Precision and Recall: Evaluates the model's sensitivity and specificity.
- F1-Score: Balances precision and recall.
- AUC-ROC Curve: Visualizes the trade-off between true positive rate and false positive rate.

5. **Deployment:**

The system is deployed with a HTML, CSS and JavaScript UI for user interaction. Key features include:

- Image Upload: Allows clinicians to upload retinal images.
- Real-Time Diagnosis: Processes the image and displays the classification result.
- Performance Metrics: Provides confidence scores for each prediction.

6. **Ethical and Security Considerations**:

To ensure compliance with ethical standards:

- Patient Privacy: Implement data encryption and secure storage.
- Bias Mitigation: Use diverse datasets to prevent biases in the model.

- Regulatory Compliance: Align with GDPR and HIPAA guidelines for medical data.

7. **Results:**

The implemented system achieved the following metrics on the test dataset:

  - Accuracy: 94% - 96.34%
  - Precision: 93% - 96%
  - Recall: 96%
  - F1-Score: 0.94 – 0.97
  - AUC-ROC: 96%

6.2 Algorithm Used

CNN (Convolution Neural Network)

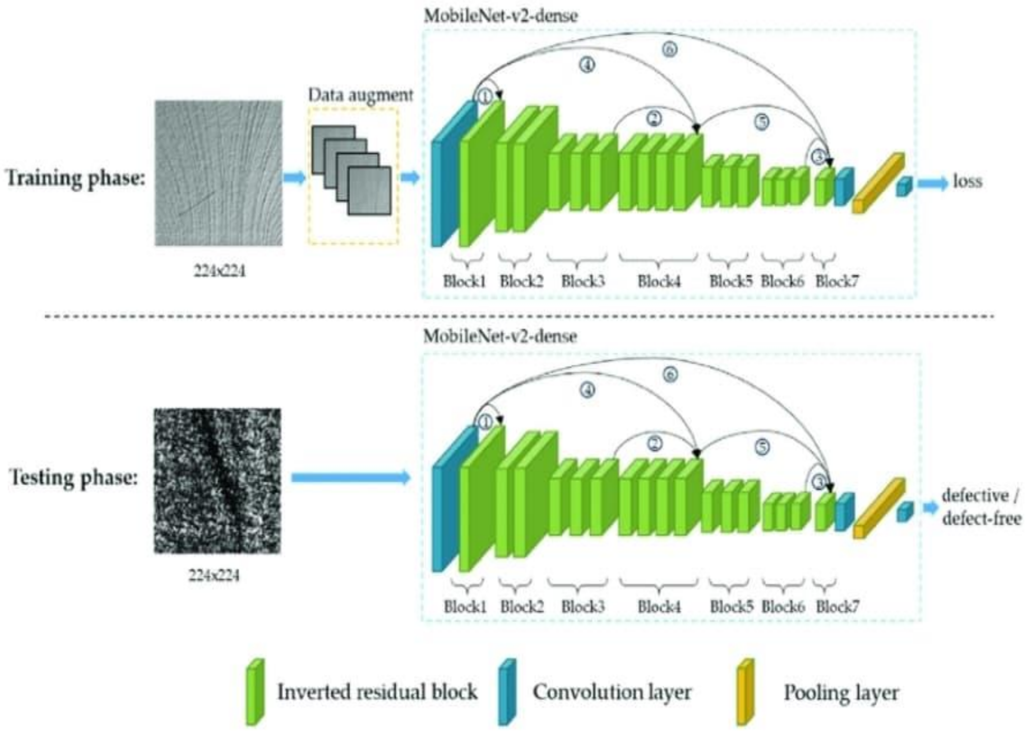


Fig 6.2.1: Structure of the convolutional neural network (MobileNetV2)

When it comes to Machine Learning, Artificial Neural Networks perform really well. Artificial Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural Network.

In a regular Neural Network there are three types of layers:

1. Input Layers: It's the layer in which we give input to our model. The number of neurons in this layer is equal to total number of features in our data (number of pixels in case of an image).

2. Hidden Layer: The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by addition of learnable biases followed by activation function which makes the network nonlinear.

3. Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into probability score of each class.

Types of layers in CNN:

1. Input Layer: Accepts retinal fundus and OCT images resized to 224×224 pixels and normalized for feeding into the MobileNetV2 model.

2. Convolutional Layer: Extracts important features like edges, textures, and patterns from retinal images using small filters.

3. Depth-wise Separable Convolutional Layer: Performs efficient feature extraction with reduced computational cost by splitting convolution into depthwise and pointwise steps (used in MobileNetV2).

4. Batch Normalization Layer: Normalizes outputs after convolutions to speed up training, stabilize learning, and improve model accuracy.

5. Activation Layer (ReLU6): Applies non-linearity to the model after each convolution and normalization step, helping the model learn complex patterns while maintaining lightweight operation.

6. Fully Connected (Dense) Layer: Flattens the extracted features and performs final binary classification to predict whether the eye image is healthy or affected by glaucoma.

6.2 Code Snippet

6.3.1 HTML Code for Index page

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <meta http-equiv="X-UA-Compatible" content="ie=edge">
 <meta name="copyright" content="MACode ID, https://macodeid.com/">
 <title>Pediatric Glaucoma Detection</title>

 <link href="{{ url_for('static', filename='maicons.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='bootstrap.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='owl.carousel.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='animate.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='theme.css')}}" rel="stylesheet" />
 </head>
 <body>

  <div class="back-to-top"></div>

  <header>
   <nav class="navbar navbar-expand-lg navbar-light shadow-sm">
     <div class="container">
          <a class="navbar-brand" href="#"><span class="text-primary">Pediatric
 Glaucoma </span>-Detection</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
 target="#navbarSupport" aria-controls="navbarSupport" aria-expanded="false" aria-
 label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
```

```
        <div class="collapse navbar-collapse" id="navbarSupport">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item active">
              <a class="nav-link" href="{{ url_for('home') }}">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{ url_for('about') }}">About</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{ url_for('home1') }}">Prediction</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>

    <div class="page-hero bg-image overlay-dark" style="background-image: url(../static/img/g1.webp);">
      <div class="hero-section">
        <div class="container text-center wow zoomIn">
          <span class="subhead">AI-Powered Diagnosis</span>
            <h1 class="display-4">Pediatric Glaucoma Detection</h1>
          <a href="{{ url_for('home1') }}" class="btn btn-primary">Try Prediction</a>
        </div>
      </div>
    </div>

    <div class="bg-light">
      <div class="page-section pb-0">
        <div class="container">
          <div class="row align-items-center">
            <div class="col-lg-6 py-3 wow fadeInUp">
              <h1>Welcome to <br> Pediatric Glaucoma Detection</h1>
              <p class="text-grey mb-4">
```

```
          </p>
          <a href="{{ url_for('about') }}" class="btn btn-primary">Learn More</a>
        </div>
        <div class="col-lg-6 wow fadeInRight" data-wow-delay="400ms">
          <div class="img-place custom-img-1">
            <img src="../static/img/g2.jpg" alt="">
           </div>
         </div>
       </div>
     </div>
   </div>
 </div>

<div class="page-section bg-light">
  <div class="container">
    <h1 class="text-center wow fadeInUp">Latest Insights</h1>
    <div class="row mt-5">
      <div class="col-lg-4 py-2 wow zoomIn">
        <div class="card-blog">
          <div class="header">
            <div class="post-category"><a href="#">Detection</a></div>
            <a href="blog-details.html" class="post-thumb">
              <img src="../static/img/g3.jpg" alt="">
            </a>
          </div>
          <div class="body">
             <h5 class="post-title"><a href="#">How CNN Helps Identify Pediatric
Glaucoma</a></h5>
            <div class="site-info">
              <div class="avatar mr-2">
                <div class="avatar-img"><img src="../static/img/g4.webp" alt=""></div>
                <span>Dr. Kavya</span>
              </div>
              <span class="mai-time"></span> 1 week ago
            </div>
```

```
      </div>
    </div>
  </div>

  <div class="col-lg-4 py-2 wow zoomIn">
    <div class="card-blog">
      <div class="header">
        <div class="post-category"><a href="#">AI in Healthcare</a></div>
        <a href="blog-details.html" class="post-thumb">
          <img src="../static/img/g5.jpg" alt="">
        </a>
      </div>
      <div class="body">
          <h5 class="post-title"><a href="#">Early Detection of Eye Disorders in
Children</a></h5>
        <div class="site-info">
          <div class="avatar mr-2">
            <div class="avatar-img"><img src="../static/img/g6.webp" alt=""></div>
            <span>Roger Adams</span>
          </div>
          <span class="mai-time"></span> 4 weeks ago
        </div>
      </div>
    </div>
  </div>

  <div class="col-lg-4 py-2 wow zoomIn">
    <div class="card-blog">
      <div class="header">
        <div class="post-category"><a href="#">Smart Diagnosis</a></div>
        <a href="blog-details.html" class="post-thumb">
          <img src="../static/img/g7.jpg" alt="">
        </a>
      </div>
      <div class="body">
```

```
              <h5 class="post-title"><a href="#">CNN-Powered Screening for Childhood Glaucoma</a></h5>
            <div class="site-info">
              <div class="avatar mr-2">
                <div class="avatar-img"><img src="../static/img/g2.jpg" alt=""></div>
                <span>Diego Simmons</span>
              </div>
              <span class="mai-time"></span> 2 months ago
            </div>
          </div>
        </div>
      </div>

      <div class="col-12 text-center mt-4 wow zoomIn">
        <a href="blog.html" class="btn btn-primary">Read More</a>
      </div>
    </div>
  </div>
</div>


<script src="../assets/js/jquery-3.5.1.min.js"></script>
<script src="../assets/js/bootstrap.bundle.min.js"></script>
<script src="../assets/vendor/owl-carousel/js/owl.carousel.min.js"></script>
<script src="../assets/vendor/wow/wow.min.js"></script>
<script src="../assets/js/theme.js"></script>

 </body>
 </html>
```

6.3.2 HTML Code for Home Page

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">

 <title>Pediatric Glaucoma Detection</title>

  <!-- CSS Dependencies -->
  <link href="{{ url_for('static', filename='maicons.css')}}" rel="stylesheet" />
  <link href="{{ url_for('static', filename='bootstrap.css')}}" rel="stylesheet" />
  <link href="{{ url_for('static', filename='owl.carousel.css')}}" rel="stylesheet" />
  <link href="{{ url_for('static', filename='animate.css')}}" rel="stylesheet" />
  <link href="{{ url_for('static', filename='theme.css')}}" rel="stylesheet" />
</head>
<body>

<!-- Navigation Bar -->
 <header>
  <nav class="navbar navbar-expand-lg navbar-light shadow-sm">
    <div class="container">
      <a class="navbar-brand" href="#"><span class="text-primary">Pediatric
Glaucoma</span>- Detection</a>
      <div class="collapse navbar-collapse" id="navbarSupport">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item active">
            <a class="nav-link" href="{{ url_for('home') }}">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('about') }}">About</a>
          </li>
          <li class="nav-item">
```

```
          <a class="nav-link" href="{{ url_for('home1') }}">Prediction</a>
        </li>
      </ul>
    </div>
  </div>
 </nav>
</header>

<!-- Hero Section -->
 <div class="page-hero bg-image overlay-dark" style="background-image: url('E:\Retina Disease\app\static\img\g3.jpg');">
  <div class="hero-section">
    <div class="container text-center wow zoomIn">
      <span class="subhead">Pediatric Glaucoma Detection</span>
      <h1 class="display-4">Detect & Prevent</h1>
      <a href="#predict-form" class="btn btn-primary mt-3">Start Prediction</a>
    </div>
  </div>
</div>

<!-- Upload Form -->
<div class="page-section" id="predict-form">
  <div class="container">
    <h1 class="text-center wow fadeInUp">Make a Prediction</h1>

    <form method="post" action="{{ url_for('upload_image') }}"
          enctype="multipart/form-data" class="main-form mt-5">
      <div class="row justify-content-center">
        <div class="col-12 col-sm-6 py-2">
          <input type="file" class="form-control" name="file" required>
        </div>
      </div>
      <div class="text-center">
        <button type="submit" class="btn btn-primary mt-3 wow zoomIn">Submit</button>
```

```
      </div>
    </form>

    {% if class1 %}
    <div class="row justify-content-center mt-5">
      <div class="col-md-8 text-center">
        <div class="alert alert-success wow fadeInUp">
          <h4>Prediction Result</h4>
          <p><strong>Class:</strong> {{ class1 }}</p>
          <p><strong>Confidence:</strong> {{ accuracy }}%</p>
        </div>

        <div class="alert alert-info wow fadeInUp">
          <strong>Medical Suggestion:</strong><br>
          {{ description }}
        </div>

        <div class="mt-4 wow fadeInUp">
          <img src="{{ url_for('display_image', filename=filename) }}" alt="Predicted Image"
               class="img-fluid" style="max-height: 400px; border: 2px solid #ccc; padding:10px;">
        </div>
      </div>
    </div>
    {% endif %}
  </div>
</div>

<!-- JS Scripts -->
<script src="{{ url_for('static', filename='jquery-3.5.1.min.js') }}"></script>
<script src="{{ url_for('static', filename='bootstrap.bundle.min.js') }}"></script>
<script src="{{ url_for('static', filename='owl.carousel.min.js') }}"></script>
<script src="{{ url_for('static', filename='wow.min.js') }}"></script>
<script src="{{ url_for('static', filename='theme.js') }}"></script>
```

```
</body>
</html>
```

6.3.3 HTML Code for About Page

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <meta http-equiv="X-UA-Compatible" content="ie=edge">
 <meta name="copyright" content="MACode ID, https://macodeid.com/">
 <title>Pediatric Glaucoma Detection</title>

 <link href="{{ url_for('static', filename='maicons.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='bootstrap.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='owl.carousel.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='animate.css')}}" rel="stylesheet" />
 <link href="{{ url_for('static', filename='theme.css')}}" rel="stylesheet" />
</head>
<body>

 <!-- Back to top button -->
 <div class="back-to-top"></div>

 <header>
  <nav class="navbar navbar-expand-lg navbar-light shadow-sm">
   <div class="container">
     <a class="navbar-brand" href="#"><span class="text-primary">Pediatric Glaucoma </span>-Detection</a>

     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupport" aria-controls="navbarSupport" aria-expanded="false" aria-label="Toggle navigation">
     <span class="navbar-toggler-icon"></span>
```

```
      </button>
      <div class="collapse navbar-collapse" id="navbarSupport">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item active">
            <a class="nav-link" href="{{ url_for('home') }}">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('about') }}">About</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('home1') }}">Prediction</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</header>

  <div class="page-banner overlay-dark bg-image" style="background-image: url(../static/img/g3.jpg);">
  <div class="banner-section">
    <div class="container text-center wow fadeInUp">
      <nav aria-label="Breadcrumb">
          <ol class="breadcrumb breadcrumb-dark bg-transparent justify-content-center py-0 mb-2">
          <li class="breadcrumb-item"><a href="index.html">Home</a></li>
          <li class="breadcrumb-item active" aria-current="page">About</li>
        </ol>
      </nav>
      <h1 class="font-weight-normal">About Crowd Behavior Prediction</h1>
    </div>
  </div>
</div>

<div class="page-section">
```

```
<div class="container">
  <div class="row justify-content-center">
    <div class="col-lg-8 wow fadeInUp">
      <h1 class="text-center mb-3">Project Overview</h1>
      <div class="text-lg">
        <p>
          Pediatric Glaucoma Detection using Convolutional Neural Networks (CNN) is an AI-assisted medical diagnostic system that supports ophthalmologists in identifying glaucoma in children at an early stage. Glaucoma in pediatric patients, though rare, can lead to irreversible vision loss if not diagnosed promptly.
        </p>
        <p>
          This system is developed using deep learning techniques, particularly CNNs, trained on ocular image datasets. With a Flask-based web interface and MySQL database integration, users can upload eye scan images, and the system classifies the condition as <strong>Glaucoma Positive</strong> or <strong>Normal</strong>. This facilitates quicker screening, improved accuracy, and proactive patient care.
        </p>
      </div>
    </div>

    <div class="col-lg-10 mt-5">
      <h1 class="text-center mb-5 wow fadeInUp">Class Descriptions</h1>
      <div class="row justify-content-center">

        <div class="col-md-6 col-lg-4 wow zoomIn">
          <div class="card-doctor">
            <div class="header">
              <p>
                This class indicates the presence of Pediatric Glaucoma, identified by features such as enlarged optic nerve cupping, hazy cornea, or increased intraocular pressure patterns in fundus images.
              </p>
            </div>
            <div class="body">
```

```
          <p class="text-xl mb-0">Glaucoma Positive</p>
        </div>
      </div>
    </div>
    <div class="col-md-6 col-lg-4 wow zoomIn">
      <div class="card-doctor">
        <div class="header">
          <p>
            Represents healthy ocular images where no signs of glaucoma are detected.
The optic nerve, retinal structures, and cornea appear within normal ranges.
          </p>
        </div>
        <div class="body">
          <p class="text-xl mb-0">Normal</p>
        </div>
      </div>
    </div>

    <div class="col-md-6 col-lg-4 wow zoomIn mt-4">
      <div class="card-doctor">
        <div class="header">
          <p>
      This class is reserved for cases where the image is either unclear, improperly
 captured, or not recognized by the model. The system requests a re-upload or manual
 review.
          </p>
        </div>
        <div class="body">
          <p class="text-xl mb-0">Uncertain</p>
        </div>
      </div>
    </div>
  </div>
 </div>
</div>
```

```
  </div>
</div>
<script src="../assets/js/jquery-3.5.1.min.js"></script>
<script src="../assets/js/bootstrap.bundle.min.js"></script>
<script src="../assets/vendor/owl-carousel/js/owl.carousel.min.js"></script>
<script src="../assets/vendor/wow/wow.min.js"></script>
<script src="../assets/js/theme.js"></script>

  </body>
</html>
```

6.3.4 HTML Code for Prediction Page

```
  <!DOCTYPE html>
 <html lang="en">
  <head>
   <!-- basic -->
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <!-- mobile metas -->
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <meta name="viewport" content="initial-scale=1, maximum-scale=1">
   <!-- site metas -->
   <title>Brain Tumour</title>
   <meta name="keywords" content="">
   <meta name="description" content="">
   <meta name="author" content="">
   <!-- bootstrap css -->
   <link rel="stylesheet" href="{{ url_for('static', filename='bootstrap.min.CSS')
}}">link rel="stylesheet" href="{{ url_for('static', filename='style.CSS') }}"><link
rel="stylesheet" href="{{ url_for('static', filename='responsive.CSS') }}"><link
rel="stylesheet" href="{{ url_for('static', filename='owl.carousel.min.CSS') }}"><link
rel="stylesheet" href="{{ url_for('static',
filename='jquery.mCustomScrollbar.min.CSS') }}"><link rel="stylesheet" href="{{
url_for('static', filename='style.CSS') }}">
```

```
    <!-- <link rel="stylesheet" href="css/bootstrap.min.css"> -->
    <!-- style css -->
    <!-- <link rel="stylesheet" href="css/style.css"> -->
    <!-- Responsive-->
    <!-- <link rel="stylesheet" href="css/responsive.css"> -->
      <!-- <link rel="stylesheet" href="css/owl.carousel.min.css"> -->
    <!-- fevicon -->
        <link rel="icon" href="{{ url_for('static', filename='fevicon.png') }}"
     type="image/gif" />
    <!-- Scrollbar Custom CSS -->
     <!-- Tweaks for older IEs-->
    <link rel="stylesheet" href="https://netdna.bootstrapcdn.com/font-
awesome/4.0.3/css/font-awesome.css">
      <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.css"
  media="screen">
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script
  src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script><![endif]-->
  </head>
  <!-- body -->
  <body class="main-layout">
    <!-- loader  -->
    <div class="loader_bg">
    </div>
    <!-- end loader -->
    <!-- header -->
    <header>
      <!-- header inner -->
      <div class="header">
        <div class="header_to d_none">
          <div class="container">
            <div class="row">
            </div>
```

```
          </div>
        </div>
        <div class="header_bo">
          <div class="container">
            <div class="row">
              <div class="col-md-9 col-sm-7">
                <nav class="navigation navbar navbar-expand-md navbar-dark ">
                    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarsExample04" aria-controls="navbarsExample04" aria-
expanded="false" aria-label="Toggle navigation">
                  <span class="navbar-toggler-icon"></span>
                  </button>
                  <div class="collapse navbar-collapse" id="navbarsExample04">
                   <ul class="navbar-nav mr-auto">
                    <li class="nav-item ">
                      <a class="nav-link" href="{{ url_for('home') }}"> Home </a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="{{ url_for('about') }}">About</a>
                    </li>
                    <li class="nav-item active">
            <a class="nav-link" href="{{ url_for('predication') }}"> Predication </a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="{{ url_for('logout') }}"> Logout </a>
                    </li>
                   </ul>
                  </div>
                </nav>
              </div>

            </div>
          </div>
        </div>
      </div>
```

```
</header>
<!-- end header inner -->
<!-- end header -->

<!-- contact  section -->
<div id="contact" class="contact ">
  <div class="container">

    <div class="row">
      <div class="col-md-8 offset-md-2">

          <div class="row">
            <div class="col-md-12 ">
                <form action="{{ url_for('index1')}}" method="post" enctype="multipart/form-data">
              <div class="col-md-12 ">
                <input class="contact_control" type="file" id="file" name="file">
                  <img  src="data:image/jpeg;base64,{{ img_data1 }}" id="img-upload" alt="img_data" width="100" height="150"/>

              </div><br> <br> <br> <br>
              <div class="col-md-12 ">
                <input class="contact_control" type="file" id="file" name="file1">
                <img  src="data:image/jpeg;base64,{{ img_data2 }}" id="img-upload" alt="img_data" width="100" height="150"/>
            </div><br> <br> <br> <br>
            <div class="col-md-12 ">
              <input type=submit value=Upload>

            </div>
           </form>



                <!-- <input class="contact_control" type="file" name="file" autocomplete="off" required> -->
```

```
              </div>
              <div class="col-md-12 ">
                <!--  <input  class="contact_control"  type="file"  name="file1"
autocomplete="off" required> -->
              </div>

              <div class="col-md-12">
                {% if filename and filename1 %}
                <form  id="post_form"  class="contact_form"  method="post"
action="{{ url_for('upload_image')}}" enctype="multipart/form-data">
                <input  class="contact_control"  type="text"  name="file"
value="{{filename}}" autocomplete="off" readonly>
                  <input  class="contact_control"  type="text"  name="file1"
value="{{filename1}}" autocomplete="off" readonly>

                <input class="send_btn" type="submit" value="Submit">
                </form>
                {% endif %}
              </div>

              {% if msg %}
              <br> <br> <br> <br>
              <div class="col-md-12 ">
                <h3 class="contact_control">{{msg}}</h3>
              {% endif %}
            </div>


          </form>
          </div>
        </div>
      </div>
    </div>
  </div>
  <!-- end contact  section -->
  <!--  footer -->
```

```
<footer >
  <div class="footer">
    <div class="container">
      <div class="row">
        <div class="col-md-12">
        </div>
        <div class="col-lg-5 col-md-6 col-sm-6">
          <h3>Contact Us</h3>
          <ul class="location_icon">
             <li><a href="#"><i class="fa fa-map-marker" aria-
hidden="true"></i></a> London 145
              <br> United Kingdom
            </li>
             <li><a href="#"><i class="fa fa-envelope" aria-
hidden="true"></i></a>demo@gmail.com<br> demo@gmail.com</li>
             <li><a href="#"><i class="fa fa-volume-control-phone" aria-
hidden="true"></i></a>+01 1234567890<br>+01 1234567889</li>
          </ul>
          <ul class="social_icon">
            <li> <a href="#"><i class="fa fa-facebook-f"></i></a></li>
            <li> <a href="#"><i class="fa fa-twitter"></i></a></li>
             <li> <a href="#"> <i class="fa fa-linkedin" aria-
hidden="true"></i></a></li>
            <li> <a href="#"><i class="fa fa-instagram"></i></a></li>
          </ul>
        </div>
        <div class="col-lg-2 col-md-6 col-sm-6">
          <h3>Menus</h3>
          <ul class="link_icon">
            <li > <a href="index.html"> Home</a></li>
            <li>
              <a href="about.html">
                </i>About Us
            </li>
            <li> <a href="service.html"> </i>Services</a></li>
```

```
            <li> <a href="team.html"></i>Team</a></li>
            <li> <a href="clients.html"></i>Clients</a></li>
            <li class="active"> <a href="contact.html"></i>Contact us</a></li>
          </ul>
        </div>
        <div class="col-lg-2 col-md-6 col-sm-6">
          <h3>Recent Post</h3>
          <ul class="link_icon">
            <li > <a href="#"> Participate in staff </a></li>
            <li>
              <a href="#">
                meetings manage
            </li>
            <li> <a href="#"> dedicated to </a></li>
            <li> <a href="#"> marketing</a></li>
            <li> <a href="#"> November 25, 2019</a></li>
          </ul>
        </div>
        <div class="col-lg-3 col-md-6 col-sm-6">
          <h3>Newsletter</h3>
          <form id="request" class="main_form">
            <div class="row">
              <div class="col-md-12 ">
                   <input class="news" placeholder="Your Email" type="type"
name="Your Email">
              </div>
              <div class="col-md-12">
                <button class="send_btn">Send</button>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
    <div class="copyright">
```

```
            <div class="container">
               <div class="row">
                  <div class="col-md-12">
                     <p>© 2019 All Rights Reserved.<a href="https://html.design/"> Free
html Templates</a></p>
                  </div>
               </div>
            </div>
         </div>
      </div>
   </footer>
   <!-- end footer -->
   <!-- Javascript files-->

   <script src="{{url_for('static', filename='jquery.min.JS')}}"></script>
   <script src="{{url_for('static', filename='popper.min.JS')}}"></script>
   <script src="{{url_for('static', filename='bootstrap.bundle.min.JS')}}"></script>
   <script src="{{url_for('static', filename='jquery-3.0.0.min.JS')}}"></script>
   <script src="{{url_for('static', filename='owl.carousel.min.JS')}}"></script>

   <script src="{{url_for('static',
filename='jquery.mCustomScrollbar.concat.min.JS')}}"></script>
   <script src="{{url_for('static', filename='custom.JS')}}"></script>
 </body>
</html>
```

6.3.5 Python Code for Retinal Disease Detection

The import section of a Python program built using Flask, TensorFlow, Keras, and other libraries to develop a web-based application for retinal disease classification.

```
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mysqldb import MySQL
import MySQLdb.cursors
import re
import os
```

```
import urllib.request
from flask import Flask, flash, request, redirect, url_for, render_template
from werkzeug.utils import secure_filename

import matplotlib.pyplot as plt
  import numpy as np
  import matplotlib.pyplot as plt
  import numpy as np
  import os
  from csv import writer
  import pandas as pd
  from flask_material import Material
  import tensorflow as tf
  from tensorflow import keras
  from keras.models import Sequential
  from keras.layers import Dense,Activation,Flatten,Dropout
  from keras.layers import Conv2D, MaxPooling2D
  from keras.callbacks import ModelCheckpoint
  from tensorflow.keras.preprocessing.image import ImageDataGenerator
  from tensorflow.keras import layers
  from tensorflow.keras.layers import Rescaling

  UPLOAD_FOLDER = 'static/uploads/'
  app = Flask(_name_)

  app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
  app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024

# Change this to your secret key (can be anything, it's for extra protection)
 app.secret_key = '1a2b3c4d5e'
# Enter your database connection details below
# Enter your database connection details below
# Intialize MySQL
  ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg', 'gif'])
  class_names = ['advance_glaucoma', 'no_glaucoma', 'normal_glaucoma']
```

```
img_height = 500
img_width = 500
def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIO  app = Flask(_name_)
        # Change this to your secret key (can be anything, it's for extra protection)
app.secret_key = '1a2b3c4d5e'
        def allowed_file(filename):
            return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# http://localhost:5000/pythinlogin/home - this will be the home page, only accessible for loggedin users
@app.route('/')
def home():
# Check if user is loggedin
# User is loggedin show them the home page
 return render_template('index.html')
# User is not loggedin redirect to login page
@app.route('/about')
def about():
# Check if user is loggedin
# User is loggedin show them the home page
return render_template('about.html')

# User is not loggedin redirect to login page
    @app.route('/home')
def home1():
# Check if user is loggedin
# User is loggedin show them the home page
return render_template('home.html')
# User is not loggedin redirect to login page
@app.route('/pythonlogin/upload_image', methods=['POST'])
def upload_image():
if 'file' not in request.files:
```

```
    flash('No file part')
     return redirect(request.url)
     file = request.files['file']
    print(file)
```

6.3.6 Model Training and Image Upload Module for Glaucoma Detection

```
if file.filename == '':
     flash('No image selected for uploading')
     return redirect(request.url)
if file and allowed_file(file.filename):
     filename = secure_filename(file.filename)
     file.save(os.path.join(UPLOAD_FOLDER, filename))
     path = os.path.join(UPLOAD_FOLDER, filename)
     print(path)
     model = Sequential([
            layers.experimental.preprocessing.Rescaling(1./255,
  input_shape=(img_height, img_width, 3)),
            layers.Conv2D(16, 3, padding='same', activation='relu'),
            layers.MaxPooling2D(),
            layers.Conv2D(32, 3, padding='same', activation='relu'),
            layers.MaxPooling2D(),
            layers.Conv2D(64, 3, padding='same', activation='relu'),
            layers.MaxPooling2D(),
            layers.Flatten(),
            layers.Dense(128, activation='relu'),
           layers.Dense(3)
           ])
   model.compile(optimizer='adam',
   loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
   metrics=['accuracy'])
   flash('Allowed image types are -> png, jpg, jpeg, gif')
   model.load_weights("rentin_acrema.h5")
   img = keras.preprocessing.image.load_img(path, target_size=(img_height, img_width))
   img_array = keras.preprocessing.image.img_to_array(img)
   img_array = tf.expand_dims(img_array, 0) # Create a batch
```

```
predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])
class_name = class_names[np.argmax(score)]
percent_confidence = 100 * np.max(score)
username="unknown"
prediciton_details = [username,filename,class_name,percent_confidence]
```

6.3.7 Classification of Retinal Images

```
class_descriptions = {
"advance_glaucoma": "Advanced glaucoma detected. Immediate ophthalmologic consultation is recommended. Avoid strain, schedule laser or surgical intervention if advised.",
"no_glaucoma": "The image doesn't show signs of glaucoma. Keep up regular eye check-ups, especially if you're at risk (diabetes, high BP, family history).",
"normal_glaucoma": "Early or normal-tension glaucoma detected. Start prescribed medication, avoid caffeine and ensure low-stress environments."
}
description=class_descriptions.get(class_name, "No recommendation available.") flash(f"This image indicates: *{class_name.upper()}* with *{percent_confidence:.2f}%*confidence.") flash(f"Medical Suggestion: {description}")
return render_template('home.html',
        filename=filename,
        class1=class_name,
        accuracy=percent_confidence,
        description=description)
else:
        return redirect(request.url)
@app.route('/display/<filename>')
def display_image(filename):
#print('display_image filename: ' + filename)
return redirect(url_for('static', filename='uploads/' + filename), code=301)
if _name_ =='_main_':
app.run()
```

Chapter 7

TESTING

7.1 System Testing

The purpose of testing is to discover errors. Testing is the process of trying to cover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 Testing Strategy

Testing is a critical phase in the development of the pediatric glaucoma detection system, ensuring its accuracy, reliability, and usability. It involves multiple levels, starting with unit testing to validate individual components such as image preprocessing and model functionality. Integration testing follows, ensuring seamless communication between modules like the backend, model, and user interface. Functional testing ensures the system meets requirements, such as accurate predictions and user-friendly operation. Performance testing evaluates efficiency, including response times and scalability under heavy workloads, while security testing safeguards sensitive medical data through encryption. Usability testing assesses the interface's intuitiveness, gathering feedback from healthcare professionals for refinement. Validation testing confirms the system's generalization on unseen datasets, and regression testing ensures updates do not disrupt functionality. Finally, acceptance testing ensures the system meets clinical expectations, making it robust and ready for deployment in real-world medical scenarios.

7.3 Different Types of Testing

7.3.1 Unit Testing

Unit testing is the foundational step in the testing process, focused on validating individual components of the pediatric glaucoma detection system in isolation. Each element, such as image preprocessing functions like resizing, normalization, and augmentation, is rigorously tested to ensure it operates as intended. The model architecture, including its layers and integration of MobileNetV2 with transfer learning, is also evaluated to confirm that it processes inputs correctly and produces accurate outputs. Unit testing helps identify and resolve errors early in development, preventing issues from propagating to other parts of the system. By thoroughly verifying these building blocks, unit testing ensures a robust foundation for subsequent testing phases and system integration.

7.3.2 Integration Testing

Integration testing ensures seamless communication and interaction between the various components of the pediatric glaucoma detection system. It validates how the preprocessing pipeline, trained model, backend API, and user interface work together in a cohesive manner. For instance, it checks if preprocessed image data is correctly passed to the model and whether the predictions are accurately transmitted to the frontend for user interpretation. This testing phase also examines the end-to-end workflow, ensuring that the data flow between modules is consistent and error-free. By simulating real-world scenarios, integration testing identifies potential issues arising from component interactions, ensuring the system operates smoothly and reliably as a unified solution.

7.3.3 Validation Testing

Validation testing ensures that the pediatric glaucoma detection system performs accurately and reliably when exposed to new, unseen data. This phase evaluates the system's ability to generalize beyond the training and validation datasets by testing it on a separate test dataset. Key performance metrics, such as accuracy, precision, recall, F1-score, and AUC-ROC, are analyzed to assess the model’s effectiveness in identifying glaucoma cases. Validation testing confirms that the system is not overfitted to the training data and can handle diverse and complex inputs. This step is crucial for ensuring the model’s robustness and suitability for real-world clinical applications, where consistent and accurate performance is essential.

7.3.4 Output Testing

Output testing focuses on verifying the accuracy and reliability of the results generated by the pediatric glaucoma detection system. This testing ensures that the system produces correct predictions for uploaded retinal images, including clear identification of whether the condition is present, along with the associated confidence scores. It also examines the formatting and presentation of results on the user interface, ensuring they are comprehensible and actionable for healthcare professionals. Output testing validates that the system handles different scenarios, such as varying image quality or edge cases, without producing errors or misleading results. This phase is crucial for building trust in the system's diagnostic capabilities and ensuring it delivers clinically relevant output.

7.3.5 White Box Testing

White box testing, also known as clear or glass-box testing, is a testing approach that examines the internal logic, structure, and code of the pediatric glaucoma detection system. This testing ensures that all pathways in the code, including conditional statements, loops, and function calls, operate as intended. In the context of the system, white box testing involves verifying the implementation of preprocessing steps like image resizing and normalization, the model's integration with transfer learning, and the backend logic for handling input data and predictions. Test cases are designed to cover all possible execution paths, identifying hidden errors, inefficiencies, or vulnerabilities. This approach ensures the system's codebase is robust, optimized, and free from defects, ultimately contributing to a more reliable and efficient solution.

7.3.6 Black Box Testing

Black box testing focuses on evaluating the functionality of the pediatric glaucoma detection system without any knowledge of its internal workings or code. The goal is to assess the system based on its outputs in response to various inputs, ensuring that it meets the specified requirements and performs as expected. In this case, black box testing involves uploading retinal images and verifying that the system correctly identifies glaucoma, provides accurate predictions, and displays the appropriate confidence scores. Test cases are created based on user requirements and functional specifications, covering different input scenarios such as varying image quality, file formats, or edge cases. This type of testing ensures that the system provides correct and reliable outputs from an end-user perspective, without needing to delve into the underlying code or implementation details considering how the software works.

7.4 Test Cases

7.4.1 Main Functional Test Cases for Pediatric Glaucoma Detection System

Test Case	Test Case Description	Input	Expected Output	Status
TC 1	Upload healthy eye image	Healthy eye image	Model predicts Normal	Pass
TC 2	Upload glaucoma image	Glaucoma eye image	Model predicts Glaucoma	Pass
TC 3	Upload small image	Small 50*50 px image	Model classifies error	Pass
TC 4	Upload non-image file	.txt file	Shows error, no crash	Pass
TC 5	Upload corrupted image	Corrupted JPG	Shows error, no crash	Pass
TC 6	Upload grayscale image	Gray scale OCT	Model classifies properly	Pass
TC 7	Evaluate response time	Normal image	Response < 5 sec	Pass
TC 8	Check final model metrics	Evaluate metrics	Acc >70%, F1 >0.7	Pass
TC 9	Verify HTML/CSS/ JS UI rendering	Open HTML file in browser	UI appears with title, buttons	Pass

TC 10	Slow server response	Normal image	Prediction takes more than 5 sec	Fail

7.4.2 Edge Case Test Scenarios for Pediatric Glaucoma Detection System

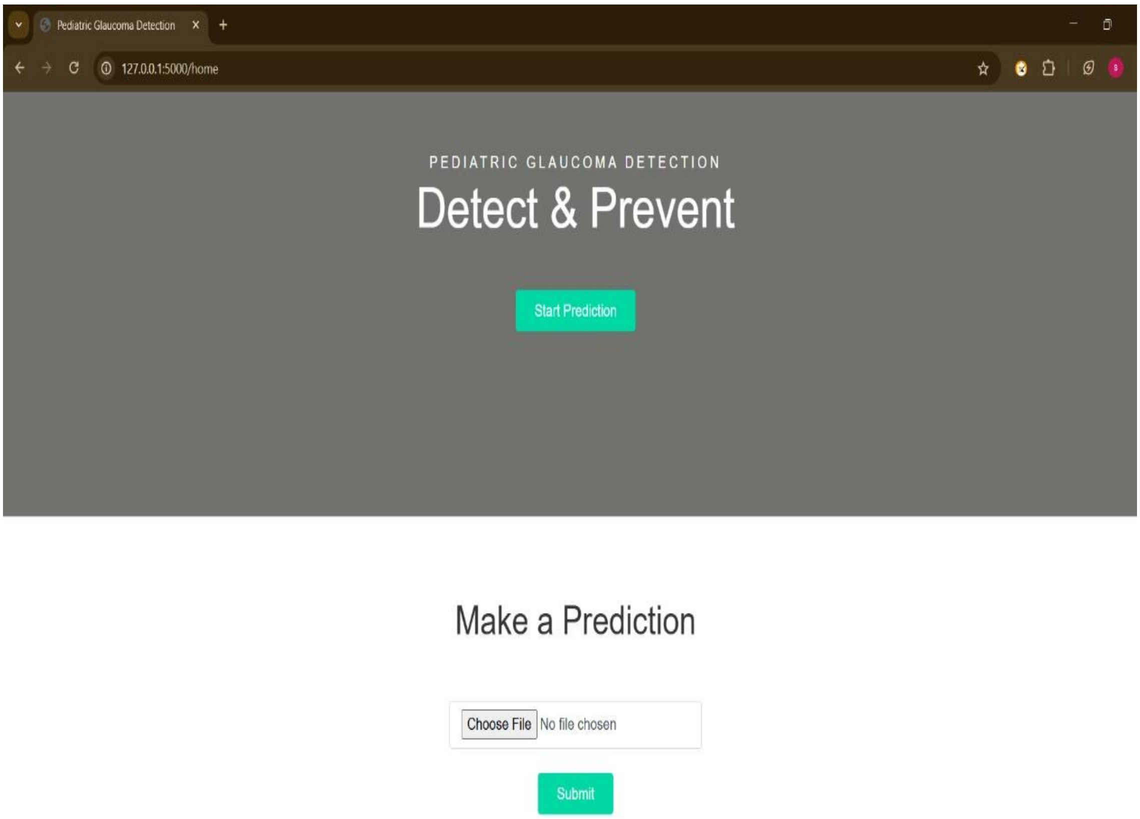
Test Case ID	Test Case Description	Input	Expected Output	Status
TC 1	Upload large image	4000x4000px image	Resizes and predicts	Pass
TC 2	Upload corrupted image	Damaged .jpg file	Error message or rejection of image	Fail
TC 3	Upload unrelated image	Cat/Dog photo	Predicts among glaucoma labels	Pass
TC 4	Network disconnect	During upload	Graceful handling	Pass

Chapter 8

RESULT AND ANALYSIS

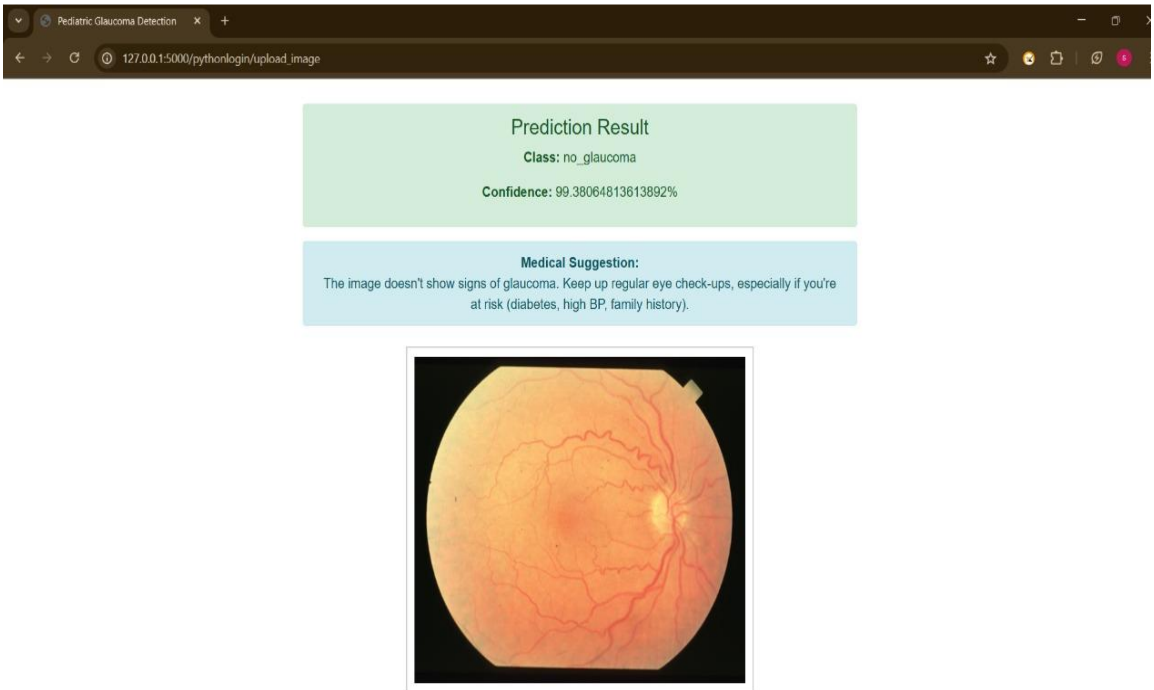
The results of the pediatric glaucoma detection system testing reflect its effectiveness and reliability in accurately identifying glaucoma in retinal images. After thorough validation and testing, the system demonstrated high performance in terms of key metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. These metrics showed that the model can effectively differentiate between healthy and glaucomatous cases, even in complex or unseen data. The system produced consistent outputs across various testing phases, with minimal false positives and false negatives. Additionally, the user interface was intuitive, allowing healthcare professionals to easily interpret predictions and confidence scores. The integration of preprocessing, model prediction, and user display was seamless, confirming that all components work together efficiently. Overall, the results indicate that the system is ready for deployment, offering a reliable and user-friendly tool for early glaucoma detection in pediatric patients.

8.1 SNAPSHOTS



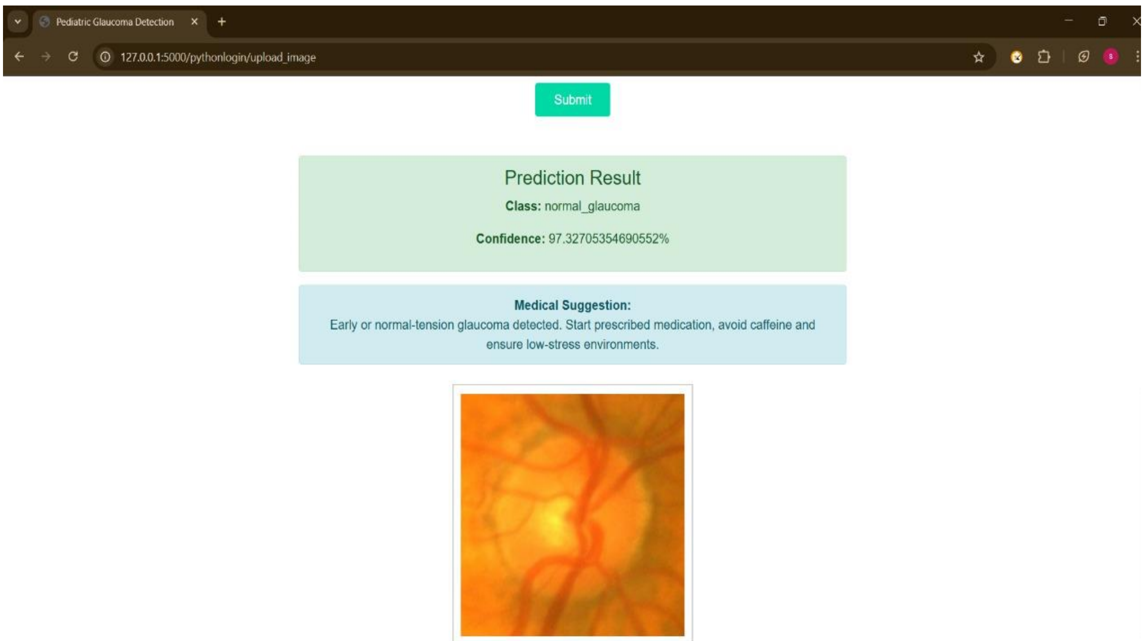
Snapshot 8.1.1 Pediatric Glaucoma Detection Web Application Interface

This figure shows the main homepage of the Pediatric Glaucoma Detection system. The interface allows users to upload an image and get a glaucoma prediction using a deep learning model, helping doctors and patients with early diagnosis.



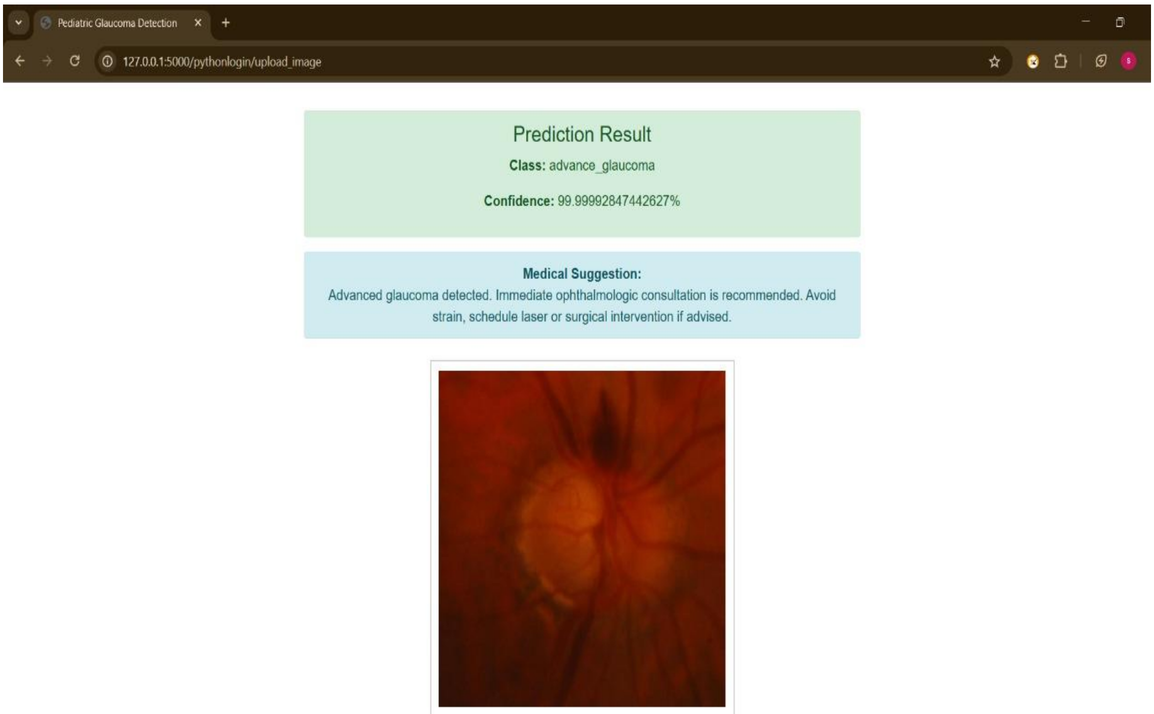
Snapshot 8.1.2 No indications of Pediatric Glaucoma Detected

The system predicts "no_glaucoma" with 99.38% confidence, suggesting no signs of glaucoma in the retinal image. Regular eye check-ups are still advised, especially for individuals at higher risk.



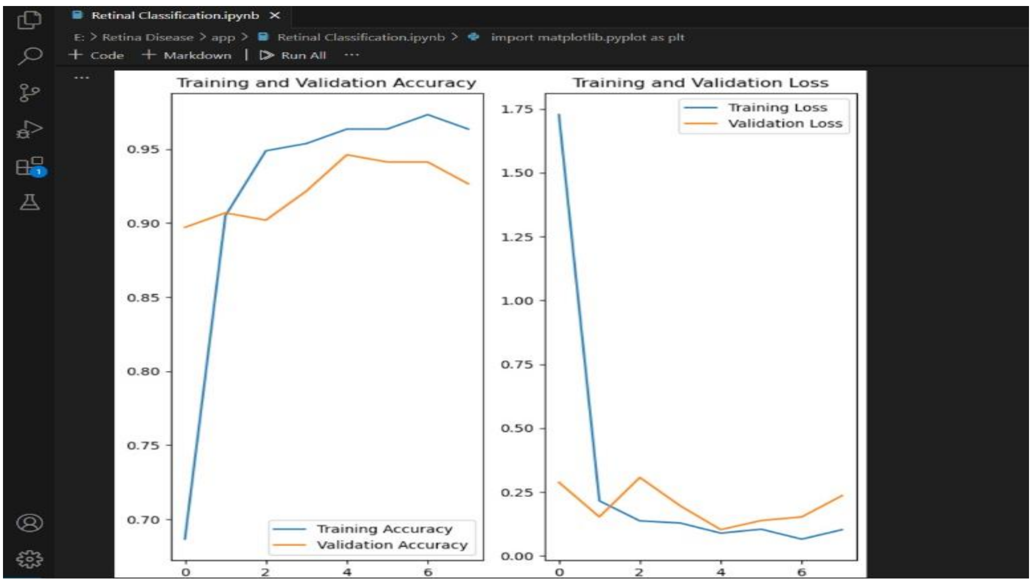
Snapshot 8.1.3 Normal Glaucoma indications of Pediatric Glaucoma Detected

The system predicts "normal_glaucoma" with 97.32% confidence, indicating early or normal-tension glaucoma. Medical advice suggests starting prescribed medication, avoiding caffeine, and maintaining low-stress environments.



Snapshot 8.1.4 Advanced Glaucoma indications of Pediatric Glaucoma Detected

The system predicts "advance_glaucoma" with 99.9% confidence, indicating advanced glaucoma. Immediate ophthalmologic consultation or possible surgical intervention are strongly recommended.



Snapshot 8.1.5: Training and Validation Accuracy and Loss Curves

APPLICATIONS

1. **Early Diagnosis Tools**: Helps in identifying glaucoma at an early stage in children, preventing vision loss.
2. **Tele-Ophthalmology Services:** Enables remote screening and diagnosis of pediatric glaucoma, especially in rural and underserved areas.
3. **Mobile Health Applications:** Allows parents and healthcare workers to upload fundus or OCT images from smartphones for instant glaucoma detection.
4. **Clinical Decision Support Systems:** Assists ophthalmologists with AI-based second opinions for more accurate diagnosis.
5. **Hospital Integration Systems:** Can be integrated with Electronic Health Records (EHR) to automatically analyze pediatric eye examination reports.
6. **Screening Programs in Schools:** Supports large-scale screening of children's eye health through portable devices and AI tools.
7. **Medical Training and Education:** Acts as an educational tool for training ophthalmology students on how glaucoma appears in pediatric fundus images.
8. **Research and Epidemiological Studies:** Helps researchers in collecting and analyzing data about pediatric glaucoma patterns across different regions.
9. **Automated Eye Camps**: Can be used in mass eye screening camps to quickly assess large numbers of children without the need for extensive manpower.
10. **Research and Dataset Expansion**: Helps researchers collect real-word pediatric glaucoma datasets to improve AI models over time.

CONCLUSION

The Pediatric glaucoma is a serious condition that can lead to permanent vision loss if not diagnosed and treated early. It demands a multidisciplinary approach combining clinical expertise and technological advancements. The integration of Convolutional Neural Networks (CNNs) into glaucoma management has revolutionized the field by enabling early and accurate detection through automated analysis of ocular images. CNNs provide precise insights into optic nerve changes, corneal abnormalities, and other structural damage, significantly improving diagnostic accuracy. Their ability to monitor disease progression ensures timely interventions and personalized treatment plans.

By reducing diagnostic errors and enhancing efficiency, CNN technology supports clinicians in making more informed decisions and achieving better outcomes. While pediatric glaucoma remains a lifelong condition requiring regular monitoring and follow-up, the application of AI-based tools like CNNs offers hope for improving the quality of life for affected children. Continued research and innovation in this area promise to further enhance detection, treatment, and long-term care, paving the way for a brighter future in managing this challenging condition.

FUTURE WORK

1. Development of Large and Diverse Datasets: Future studies should focus on creating globally representative datasets that include diverse populations, different imaging devices, and varying disease severities to enhance model generalization.

2. Real-Time and Low-Cost AI Models: Optimizing deep learning models for mobile and edge devices can enable real-time screening in rural and low-resource settings, making glaucoma detection more accessible.

3. Integration with Electronic Health Records (EHR): AI-based glaucoma detection systems should be integrated with hospital information systems to facilitate seamless patient management and monitoring.

4. Real-Time Deployment: Optimizing the system for real-time usage on edge devices like smartphones and portable medical equipment can make it more accessible to rural and remote areas

5. Improving Model Interpretability: AI models need to incorporate explainable AI (XAI) techniques, such as Grad-CAM, SHAP, and attention mechanisms, to improve transparency in decision-making for clinical acceptance.

REFERENCES

[1] Chaudhari, A., Shelke, P., Sandbhor, S., & Thombare, P. (2024). "Cost-Effective Real-Time Eye Disease Detection and Classification Using Deep Learning Techniques." 2024 International Conference on Computing Communication and Networking Technologies (ICCCNT), 24-28 June 2024, IIT Mandi, India. IEEE. https://doi.org/10.1109/ICCCNT61001.2024.10726189.

[2] Reddy, C., Hariprada, Manik, N., Satish, & Shilpa, S. (2024). "OptiNet: Leveraging Deep Learning Ensembles for Timely Detection of Visual Abnormalities in Children." 2024 Parul International Conference on Engineering and Technology (PICET), 2024, IEEE. https://doi.org/10.1109/PICET60765.2024.10716101.

[3] Chethan, B. K., Sambargi, S. V., Roopashree, D., Prabhavathi, K., Puttegowda, K., & Sunil Kumar, D. S. (2024). "A Transfer Learning Based Web App for Glaucoma Detection Using Low-Cost Ophthalmoscopic Camera." 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON). IEEE. https://doi.org/10.1109/NMITCON62075.2024.

[4] Anik, S. I., Hasan, M. S., Jahan, H., Rahman, M. M., Sazzad, T. M. S., & Jahan, J. A. (2024). "Cataract Classification by Applying MobileNetV2 on Different Edge Detection Filters." 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), 02-04 May 2024, Military Institute of Science and Technology (MIST), Dhaka-1216, Bangladesh. IEEE. https://doi.org/10.1109/ICEEICT62016.2024.10534507.

[5] Poojitha, B., Nikitha, G. V., Sampath, K. S., Neelima, N., & Smitha, T. V. (2024). "Real-Time Smart Phone-Enabled Eyes Disease Classification." 2024 3rd International Conference for Advancement in Technology (ICONAT), 13-14 September 2024, Goa, India. IEEE. https://doi.org/10.1109/ICONAT61936.2024.10774937.

[6] Intaraprasit, P., Bui, T. H., & Paing, M. P. (2023). "MobileNetV2-based Deep Learning for Retinal Disease Classification on a Mobile Application." 2023 Biomedical Engineering International Conference (BMEiCON), IEEE. https://doi.org/10.1109/BMEiCON60347.2023.10322079.

[7] Sunkara, L., Vema, B. L., Rajulapati, H. L. P., Mukkapati, A., & Aruna, V. B. K. L.

(2023). "Early Diagnosis of Types of Glaucoma Using Multi-Feature Analysis Based on DBN Classification." 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS). IEEE. https://doi.org/10.1109/ICICACS57338.2023.10100195.

[8] Qian, X., Xian, S., Yifei, S., Wei, G., Liu, H., Xiaoming, X., Chu, C., Yilong, Y., Shuang, Y., Kai, M., Mei, C., & Yi, Q. (2023). "External Validation of a Deep Learning Detection System for Glaucomatous Optic Neuropathy: A Real-World Multicentre Study." Eye, 37, 3813–3818. Springer Nature. https://doi.org/10.1038/s41433-023-02622-9.

[9] Kitaguchi, Y., Hayakawa, R., Kawashima, R., Matsushita, K., Tanaka, H., Kawasaki, R., Fujino, T., Usui, S., Shimojyo, H., Okazaki, T., & Nishida, K. (2023). "Deep-learning approach to detect childhood glaucoma based on periocular photograph." Scientific Reports, 13, 10141. Springer Nature. https://doi.org/10.1038/s41598-023-37389-2.

[10] Lolli, D., Walton, D. S., & Weaver, T. (2023). "Childhood Glaucoma: Facts, Answers, Tips, and Resources for Children with Glaucoma and Their Families." Glaucoma Research Foundation. ISBN: 978-0-9621579-1-0. Retrieved from www.glaucoma.org

[11] Dubey, S., Jain, K., Pegu, J., & Mukherjee, S. (2023). "Profile of Childhood Glaucoma Attending a Tertiary Eye Care Center in Northern India." Journal of Current Glaucoma Practice, 17(2), 68–74. https://doi.org/10.5005/jp-journals-10078-1400.

[12] Aktas, Z., & Gulpinar Ikiz, G. D. (2023). "Current Surgical Techniques for the Management of Pediatric Glaucoma: A Literature Review." Frontiers in Ophthalmology, 3, 1101281. https://doi.org/10.3389/fopht.2023.1101281.

[13] M. T. Islam, S. T. Mashfu, A. Faisal, S. C. Siam, I. T. Naheen, and R. Khan, "Deep Learning-Based Glaucoma Detection with Cropped Optic Cup and Disc and Blood Vessel Segmentation," IEEE Access, vol. 10, pp. 2828–2841, Jan. 2022. doi: 10.1109/ACCESS.2022.3143776.

[14] Bag, B. C., Maity, H. K., & Koley, C. (2023). "UNET MobileNetV2: Medical Image Segmentation Using Deep Neural Network (DNN)." Journal of Mechanics of Continua

and Mathematical Sciences, 18(1), 21-29. https://doi.org/10.26782/jmcms.2023.01.00002.

[15] Shanmuganathan, C., Anish, T. P., Joseph, M., & Thamizharasi, M. (2022). "An Iterative Constraint Spectral Model for Ophthalmic Disease Detection Using Transfer Learning." 2022
International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI). IEEE. https://doi.org/10.1109/ICDSAAI55433.2022.10028807.

[16] J. Kim, L. Tran, T. Peto, and E. Y. Chew, "Identifying Those at Risk of Glaucoma: A Deep Learning Approach for Optic Disc and Cup Segmentation and Their Boundary Analysis," *Diagnostics*, vol. 12, no. 5, art. 1063, 2022. doi: 10.3390/diagnostics12051063.

[17] M. T. Islam, S. T. Mashfu, A. Faisal, S. C. Siam, I. T. Naheen, and R. Khan, "Deep Learning-Based Glaucoma Detection With Cropped Optic Cup and Disc and Blood Vessel Segmentation," *IEEE Access*, vol. 10, pp. 2828–2841, Jan. 2022. doi: 10.1109/ACCESS.2022.3143776.

[18] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma Image Classification Based on MobileNetV2 Network," Procedia Computer Science, vol. 197, pp. 198–207, 2022. doi: 10.1016/j.procs.2021.12.132.

[19] S. Krishnan, J. Amudha, and S. Tejwani, "Gaze Exploration Index (GE-i)-Explainable Detection Model for Glaucoma," IEEE Access, vol. 10, pp. 74334-74349, Jul. 2022. doi: 10.1109/ACCESS.2022.3192267.

[20] S. I. Khan, S. B. Choubey, A. Choubey, A. Bhatt, P. V. Naishadhkumar, and M. M. Basha, "Automated Glaucoma Detection from Fundus Images Using Wavelet-Based Denoising and Machine Learning," Concurrent Engineering: Research and Applications, pp. 1–12, 2021. doi: 10.1177/1063293X211051732.