CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF OVERVIEW EYE DISEASE DETECTION

In recent years, the application of deep learning techniques has revolutionized medical diagnostics, particularly in the field of ophthalmology. The project aims to develop a sophisticated system for eye disease detection using ResNet-50 algorithm. The approach not only improves diagnostic accuracy but also facilitates early intervention and treatment, potentially transforming eye care practices and outcomes. Deep learning techniques, particularly the ResNet-50, have shown great potential in accurately classifying retinal images for diagnosing these conditions. Python for advanced data processing, which will optimize lighting efficiency and convenience by dynamically responding to changes in ambient light levels and human activity.

1.2 PROCESSING OF IMAGES

In eye disease detection using the ResNet-50 algorithm, image processing plays a crucial role in preparing and enhancing retinal images for effective analysis. The goal of image processing in this context is to extract important features and remove noise, which is essential for accurate disease diagnosis. Image processing is typically carried out in eye disease detection using the ResNet-50 algorithm:

1. **Image Acquisition**: The first step involves acquiring high-quality retinal images, typically from fundus cameras or optical coherence tomography (OCT) scans. These images contain vital information about the structure of the eye and can be used to detect conditions like diabetic retinopathy, glaucoma, and macular degeneration.
2. **Preprocessing:** Image preprocessing is a crucial step to enhance the quality of the raw images and standardize them for analysis. This step may involve:

   • **Resizing:** Images are resized to a uniform dimension to ensure consistency across the dataset. This allows the model to process all images with the same input size.

   • **Grayscale Conversion:** In some cases, retinal images are converted to grayscale to focus on important features, especially when the color information is not essential for disease detection.

   • **Noise Reduction:** Random noise in the images, such as artifacts from the imaging device, is removed using techniques like Gaussian filtering or median filtering to ensure that the model focuses on relevant features.

   • **Data Augmentation:** To increase the diversity of the training data, various data augmentation techniques are applied. These can include random rotations, flips, zooming, and shifting of the images. Data augmentation helps in preventing overfitting and improving the generalization capability of the model.

   • **Region of Interest (ROI) Extraction**: In retinal images, certain areas such as the optic disc, macula, and blood vessels are more important for diagnosing specific diseases. Image segmentation techniques, such as thresholding or region-growing, are applied to extract the region of interest (ROI) from the images, which focuses the analysis on relevant areas.

- **Feature Extraction:** ResNet-50 uses residual learning with skip connections to extract hierarchical features from retinal images. This approach mitigates the vanishing gradient problem, allowing for deeper and more effective learning. ResNet-50 captures both low-level and high-level image features, enhancing the model's capability to detect subtle abnormalities in retinal images.

3. **Feeding into ResNet-50 Network:** After preprocessing and feature extraction, the processed images are fed into the ResNet-50 model. The network applies convolutional layers and residual blocks to analyze the spatial relationships and patterns in the images. ResNet-50's deep architecture enables it to achieve high accuracy in image classification tasks, making it suitable for identifying eye diseases.
4. **Post-Processing**: After the ResNet-50 network processes the image, the output is usually in the form of predictions regarding the presence or severity of a disease. Post-processing steps like thresholding can be used to convert the model's output into a binary classification (e.g., presence or absence of a disease) or to refine the predictions, depending on the problem.
5. **Evaluation and Validation:** Once the model completes the prediction, it undergoes an evaluation phase where various performance metrics are analyzed. Common metrics include accuracy, precision, recall, and F1-score. These metrics provide insights into how well the ResNet-50 model is performing in identifying eye diseases. Additionally, cross-validation techniques are applied to ensure model reliability across different datasets. Validation using external or unseen data helps to estimate the

model's generalization ability, making it more robust and effective in real-world clinical applications.

1.3 NEED FOR THE STUDY

The need of study in Detecting eye diseases at an early stage is essential to prevent vision loss and improve patient outcomes. Traditional diagnostic methods are often time-consuming and dependent on expert evaluation, leading to delays in treatment. Advanced computational models like ResNet-50 can analyze medical data efficiently and identify patterns related to various eye conditions. Automating disease detection helps reduce the workload of ophthalmologists while increasing diagnostic accuracy. Enhancing technology-driven techniques improves accessibility to eye care, especially in underserved areas. Developing intelligent systems for eye health can significantly improve early diagnosis and treatment planning.

1.4 OBJECTIVES OF STUDY

The objective of this study is to develop an effective system for early detection of eye diseases such as diabetic retinopathy, glaucoma, and cataracts using ResNet-50. By enhancing diagnostic accuracy, the model aims to identify patterns in medical data and improve the reliability of disease detection. Automating the identification process reduces dependence on manual examinations, leading to faster and more consistent diagnoses while assisting ophthalmologists in managing their workload. Efficient processing of medical image data helps recognize disease progression, enabling better treatment planning and patient care. The study also focuses on improving accessibility to eye care, particularly in underserved areas, by integrating the model into healthcare systems for real-time diagnosis.

CHAPTER 2

LITERATURE SURVEY ON EYE DISEASE DETECTION

Dawei Liu (2024) contributed significantly to the field of eye disease detection by exploring the use of advanced deep learning models for improving diagnostic accuracy and early detection. Their work focuses on leveraging convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) networks to analyze retinal images, providing a powerful tool for detecting a range of ocular diseases such as diabetic retinopathy, macular degeneration, and glaucoma. Liu et al. explored the potential of combining CNNs with LSTM models to effectively analyze temporal and spatial features in medical images, enabling more accurate and early identification of disease progression. Their research also highlights the importance of integrating multimodal data, such as patient history and environmental factors, into the diagnostic process. By applying these models to large-scale datasets, the study demonstrated the ability to improve diagnostic accuracy and reduce errors associated with traditional eye disease detection methods. Additionally, the integration of AI-based systems with telemedicine platforms was proposed to make eye care more accessible, especially in under-served regions. Liu et al.'s work reflects the growing trend of using AI to revolutionize eye care, offering tools that can assist clinicians in providing more personalized and timely treatments.

Bin Liu (2023) explored the application of unsupervised deep learning techniques for random noise attenuation in seismic data, a crucial aspect of improving seismic imaging and interpretation. Their research focuses on leveraging deep learning models, particularly autoencoders, to reduce noise interference in seismic datasets without requiring labeled data. This unsupervised approach is significant because it avoids the costly and time-consuming process of manual labeling, making it more scalable for large seismic datasets. Liu et al. demonstrated that their model could effectively identify and suppress random noise while preserving essential geological features, which is often a challenge with traditional filtering methods. The study also highlighted the advantages of using deep learning over conventional techniques like band-pass filtering, which may not adequately address complex noise patterns. The authors' approach relies on the inherent structure in seismic data, allowing the model to learn from raw data and enhance signal-to-noise ratios.

C. Rekha and K. Jayashree (2022) focused on predicting Hyphema, a condition involving the presence of blood in the anterior chamber of the eye, using deep learning techniques. Their research explored how machine learning models, particularly convolutional neural networks (CNNs), can be applied to detect and predict hyphema from medical imaging data, such as fundus and slit-lamp images. The study highlighted the potential of deep learning algorithms to identify subtle patterns in ocular images that may not be easily visible to the human eye, thereby improving early diagnosis and treatment of hyphema. Rekha and Jayashree's work is part of the growing trend of applying artificial intelligence (AI) in ophthalmology, where deep learning models have been shown to outperform traditional methods in terms of diagnostic accuracy. Their approach emphasizes the importance of using high-quality annotated datasets to train the models, ensuring they can accurately differentiate hyphema from other eye conditions with similar symptoms. Additionally, the researchers explored the integration of various data modalities, such as clinical and demographic information, to enhance prediction accuracy.

Gauri Ramanathan, Diya Chakrabarti (2021) explore the significant advancements in applying artificial intelligence to the early diagnosis and management of various ocular conditions. Their work reviews a variety of machine learning techniques, particularly focusing on supervised learning models like support vector machines (SVMs), decision trees, and deep learning algorithms such as convolutional neural networks (CNNs). These methods have been widely used for detecting eye diseases like diabetic retinopathy, glaucoma, and macular degeneration from retinal fundus images, OCT scans, and other imaging modalities. The authors highlight how machine learning models have shown remarkable performance in automating the analysis of medical images, providing faster and more accurate diagnostics compared to traditional manual methods. They also discuss the challenges in using these models, including the need for large, high-quality labeled datasets, dealing with imbalanced data, and ensuring model interpretability for clinical settings. Additionally, the study emphasizes the importance of integrating multimodal data, such as patient history and lifestyle factors, to improve prediction accuracy and support personalized treatment plans. Overall, this literature survey reflects the growing role of machine learning in revolutionizing ophthalmology, offering a promising future for efficient, scalable, and accessible eye disease detection and management.

Ali Raza, and Sana Samer (2021) focus on advancing diagnostic techniques in ophthalmology through the application of artificial intelligence. Their work reviews various machine learning and deep learning models, particularly Inception-V4, which is known for its high accuracy in image classification tasks. The authors highlight the effectiveness of using DFI for capturing high-quality retinal images, which can be analyzed to detect a range of eye diseases, including diabetic retinopathy, glaucoma, and cataracts. They discuss the increasing adoption of convolutional neural networks (CNNs) like Inception-V4, which has proven to be particularly suitable for processing complex medical images and identifying subtle patterns associated with eye conditions. The study also reviews the performance of deep learning models in distinguishing cataracts from other eye diseases, noting the challenges in differentiating between various pathologies that share similar symptoms. Moreover, the authors point out the importance of large, annotated datasets in training these models to achieve high accuracy and reliability. The research contributes to the growing body of literature emphasizing the role of AI in ophthalmology, specifically in automating diagnosis, reducing clinician workload, and providing faster, more accurate results in the early detection of cataracts and other ocular diseases.

Xue Xia, and Jinhua Yan (2020) introduced a comprehensive approach to improving eye disease diagnosis through the creation of a large-scale dataset and the development of fundus image synthesis techniques. Their research addresses the challenges of limited annotated data in ophthalmology, which is crucial for training deep learning models to accurately diagnose eye diseases such as diabetic retinopathy, glaucoma, and macular degeneration. The authors propose the construction of a large-scale dataset of fundus images, which serves as a critical resource for training and evaluating machine learning algorithms. Additionally, they explore the use of image synthesis techniques to augment the dataset, allowing for better model generalization and improving diagnostic performance, particularly in cases where data scarcity could limit model accuracy. The study also presents a benchmark for evaluating the performance of various deep learning models in the context of eye disease diagnosis. By synthesizing fundus images and generating diverse datasets, Xia et al. enhance the ability of AI models to detect subtle features associated with different eye conditions.

CHAPTER 3

EYE DISEASE DETECTION SYSTEM

3.1 EYE DISEASE DETECTION USING SVM

Eye disease detection has evolved significantly with advancements in machine learning and deep learning, focusing on automated and efficient diagnosis through image analysis. Current systems often rely on Support Vector Machines (SVMs), a form of machine learning, to analyze retinal images such as fundus photographs, optical coherence tomography (OCT), and fluorescein angiography for detecting various conditions like diabetic retinopathy, macular degeneration, glaucoma, and cataracts. SVM-based models have demonstrated high accuracy in identifying retinal diseases by learning and extracting key features from images, making them invaluable for early diagnosis.

One of the most widely adopted systems is the use of machine learning models for diabetic retinopathy detection. Models like the Diabetic Retinopathy Detection System developed by Google Health use an SVM architecture to analyze retinal images, diagnosing the presence of diabetic retinopathy with an accuracy comparable to that of ophthalmologists. These systems have been trained on large datasets of labeled images and have undergone extensive validation, showcasing their potential in clinical environments. Furthermore, there have been advancements in glaucoma detection, where machine learning models analyze the optic disc and cup ratio in fundus images to assess glaucoma risk.

In addition to image-based systems, hybrid approaches combining machine learning with clinical data, such as patient history, have also been developed to improve diagnostic accuracy. For instance, multi-modal systems that incorporate both imaging and clinical data can enhance the prediction of conditions like macular degeneration or glaucoma, considering both the patient's retinal images and underlying risk factors such as age, family history, and medical history.

3.2 PYTHON AND CNN APPROACHES

Python is utilized as the primary programming language due to its robust support for machine learning and image processing. The project employs a multi-stage deep learning approach, leveraging the power of Vision Transformers (VITs) to analyze retinal fundus images for detecting various eye diseases like diabetic retinopathy, glaucoma, and macular degeneration. These models are designed to effectively capture long-range dependencies and intricate patterns in images, enabling precise detection and classification. To enhance the model's accuracy, transfer learning is incorporated, where pre-trained models like VITs are fine-tuned on eye disease datasets.

The Keras library is used to design, train, and optimize the deep learning models, providing essential tools for high-performance computations. For image preprocessing, OpenCV and Pillow are employed to improve image quality, resize images, and apply augmentations for better generalization. This multi-stage deep learning methodology, integrating Vision Transformers, allows for more accurate, scalable, and reliable eye disease detection, offering significant improvements in automated ophthalmology diagnostics.

3.3 DRAWBACKS OF EXISTING SYSTEMS

The existing eye disease detection systems using Support Vector Machines (SVMs) face several challenges that impact their clinical effectiveness. Unlike deep learning models, SVMs rely heavily on handcrafted features for image classification, making their performance highly dependent on the quality of feature extraction. Extracting relevant features from retinal images such as blood vessel patterns, optic disc measurements, and lesion characteristics requires significant domain expertise.

Additionally, SVMs may struggle with large and complex datasets, as their computational efficiency diminishes with an increasing number of features and data points. While SVMs can provide accurate results for well-defined, smaller datasets, their scalability is limited in scenarios requiring real-time diagnosis or large-scale screening.

Furthermore, the lack of interpretability in SVM decision boundaries poses challenges for gaining clinical trust, as healthcare professionals often prefer models that offer clear explanations for their predictions.

3.4 DATA PRIVACY AND ETHICAL CONCERNS

The use of Support Vector Machines (SVMs) for eye disease detection raises significant data privacy and ethical concerns. SVM-based models rely on medical records and eye scan images, which contain sensitive personal information that must be protected from unauthorized access, data breaches, or misuse.

Ensuring data security through encryption, restricted access, and proper storage is essential to maintain patient confidentiality. Ethical concerns also arise regarding potential biases in medical data, which can lead to inaccurate diagnoses and disproportionately affect certain populations, contributing to healthcare inequalities. Additionally, informed consent is crucial so that patients understand how their medical data is collected, stored, and used in SVM-based diagnostic systems.

Transparency in the decision-making process of SVMs helps build trust between patients and healthcare providers, ensuring that medical decisions are well understood. Furthermore, issues related to data ownership must be addressed to give patients control over their medical information and prevent exploitation. To uphold ethical standards, healthcare institutions must implement strict privacy measures, use diverse and representative data, and ensure fairness and accuracy in SVM-based disease detection methods.

Moreover, regulatory frameworks and compliance standards must be established to govern the use of SVMs in eye disease detection. Institutions should adhere to legal regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR) to ensure data protection and ethical use of medical information. Continuous monitoring and auditing of SVM systems can help identify potential biases or errors, promoting accountability and reliability. Collaboration between AI developers, medical practitioners, and regulatory bodies is essential to create transparent and equitable AI-driven diagnostic systems that prioritize patient welfare and privacy.

3.5 RECENT ADVANCEMENTS AND FUTURE DIRECTIONS

Recent advancements in eye disease detection using Support Vector Machines (SVMs) have significantly improved the accuracy and efficiency of diagnosing conditions such as diabetic retinopathy, glaucoma, and age-related macular degeneration. By leveraging powerful classification capabilities, SVM models can process medical imaging data, retinal scans, and clinical data to detect disease patterns effectively.

Recent developments have integrated SVMs with advanced feature extraction techniques, enabling more precise and early-stage disease detection. Additionally, advancements in federated learning and secure data sharing have allowed SVM models to be trained on diverse datasets while preserving data privacy. Cloud-based and edge computing solutions have also enhanced real-time disease detection, making these models more accessible for remote and telemedicine applications.

Future directions in eye disease detection using SVMs focus on increasing interpretability, optimizing feature selection, and improving dataset diversity to eliminate biases. The integration of multimodal data, including genetic and lifestyle factors, could enhance predictive accuracy.

Researchers are also exploring hybrid machine learning architectures that combine SVMs with ensemble methods to improve decision-making transparency. With ongoing advancements, SVM-based models have the potential to revolutionize early diagnosis, personalized treatment, and global accessibility to eye care, ultimately reducing the burden of preventable blindness.

The integration of SVMs with Internet of Things (IoT) devices and wearable technology enables continuous eye health monitoring. Smart contact lenses and portable retinal imaging devices equipped with SVM-powered analytics can provide real-time assessments, allowing for early detection and intervention in high-risk patients.

CHAPTER 4

EYE DISEASE DETECTION USING RES-NET 50

4.1 PROPOSED METHODOLOGY

The proposed system for eye disease detection using ResNet-50 focuses on leveraging deep feature extraction and rule-based analysis instead of machine learning-based classification. This system follows a structured approach to process retinal images, extract meaningful features, and classify eye diseases based on predefined medical criteria. By utilizing ResNet-50 purely as a feature extractor, the system avoids complex AI-driven learning mechanisms while still benefiting from deep feature representations.

The system begins with image acquisition, where high-resolution retinal images are collected using fundus cameras or Optical Coherence Tomography (OCT) scans. These images serve as the primary input for the detection process. Following this, preprocessing techniques such as resizing, grayscale conversion, noise reduction, and contrast enhancement are applied to improve image quality and standardize input data. This ensures that the extracted features remain consistent and relevant for further analysis.

In the feature extraction phase, ResNet-50 is employed as a pre-trained deep learning model to extract hierarchical features from retinal images. The extracted features—such as texture patterns, vessel structures, and lesion characteristics are analyzed using predefined medical rules. These rules are designed based on ophthalmological knowledge, defining thresholds for various disease indicators like hemorrhages, exudates, or optic nerve damage.

The rule-based classification system compares the extracted features against established medical criteria. For example, if certain patterns of blood vessel damage exceed predefined thresholds, the system identifies the possibility of diabetic retinopathy. Similarly, optic nerve cup-to-disc ratio measurements can indicate the presence of glaucoma. This approach eliminates the need for AI-based decision-making while still providing automated disease detection capabilities.

Finally, the report generation and decision support phase provides diagnostic insights in an interpretable format. The system generates reports highlighting detected abnormalities, severity levels, and recommended medical actions. These reports can assist ophthalmologists in confirming diagnoses, reducing their workload, and improving efficiency in clinical settings. Additionally, integration with telemedicine platforms can enhance accessibility to eye care, particularly in remote areas where specialized ophthalmologists are scarce.

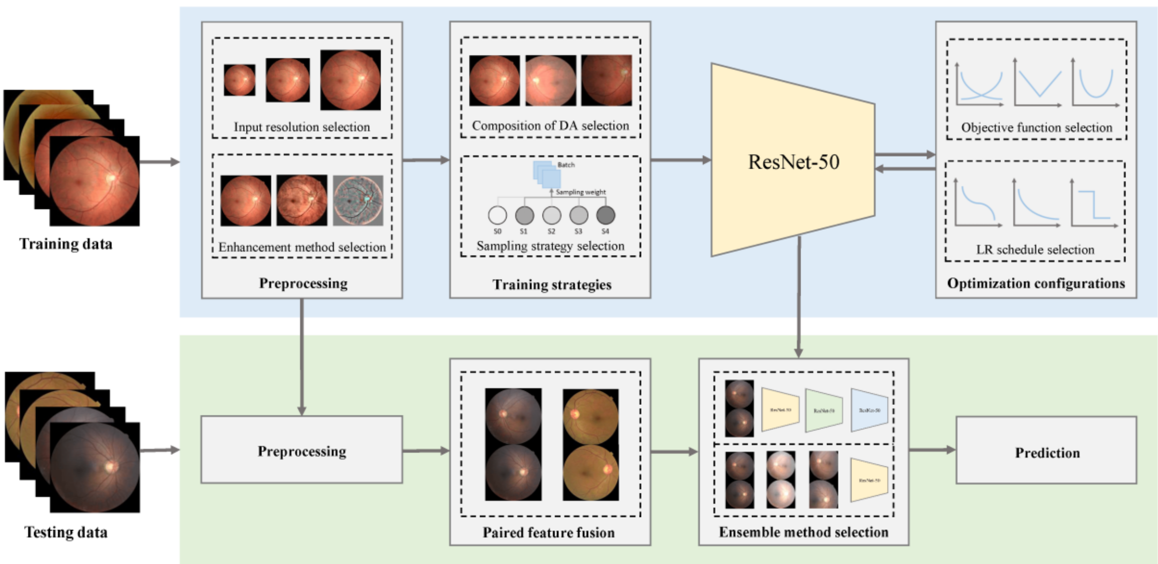

Fig 4.1 ResNet-50 architecture on eye diseases

In Fig 4.1 the use of ResNet-50 for feature extraction ensures that even subtle variations in retinal images are captured, facilitating the identification of early-stage eye diseases. The detailed feature maps generated by ResNet-50 provide insights into abnormalities that may go unnoticed during manual inspections.

By combining this advanced feature extraction capability with a robust rule-based framework, the system minimizes false positives and false negatives, leading to more reliable diagnoses. This dual approach of automated assistance and human expertise enhances the overall effectiveness of eye disease detection, ultimately contributing to improved patient outcomes and better management of ocular health.

THE LIMITATIONS ARE:

1. Longer Training Time
2. High Computational Complexity
3. Overfitting Risk
4. Lack of Explainability
5. Limited Spatial Feature Extraction.

4.2 RES-NET 50 OVERVIEW

The proposed system for eye disease detection using ResNet-50focuses on leveraging deep feature extraction and rule-based analysis instead of machine learning-based classification. This system follows a structured approach to process retinal images, extract meaningful features, and classify eye diseases based on predefined medical criteria. By utilizing ResNet-50 purely as a feature extractor, the system avoids complex AI-driven learning mechanisms while still benefiting from deep feature representations.

The system begins with image acquisition, where high-resolution retinal images are collected using fundus cameras or Optical Coherence Tomography (OCT) scans. These images serve as the primary input for the detection process. Following this, preprocessing techniques such as resizing, grayscale conversion, noise reduction, and contrast enhancement are applied to improve image quality and standardize input data.

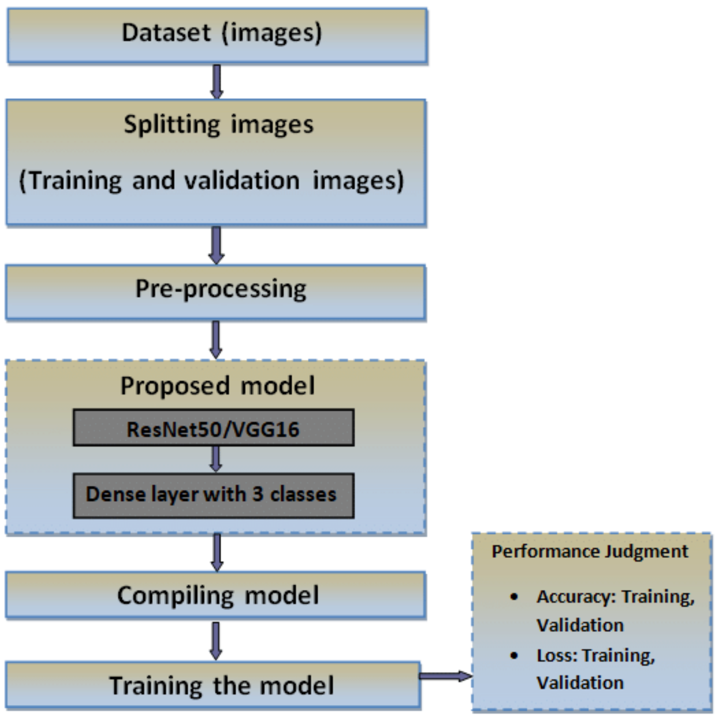

Fig 4.2 Flowchart of ResNet 50

It provides a cost-effective and scalable solution for preliminary disease detection, allowing healthcare providers to prioritize patients requiring urgent medical attention. The proposed approach also offers flexibility for further customization by medical professionals, allowing them to adjust classification rules based on evolving diagnostic standards and regional health data.

In addition, the use of ResNet-50 for feature extraction ensures that even subtle variations in retinal images are captured, facilitating the identification of early-stage eye diseases.

4.3 METHODS IN RES-NET50

The primary methods used in ResNet-50 include convolutional operations, batch normalization, ReLU (Rectified Linear Unit) activation, and skip connections. Convolutional operations are applied to extract hierarchical features from input images, capturing patterns like edges, textures, and shapes. Batch normalization is used to stabilize and accelerate the training process by normalizing the inputs of each layer, reducing internal covariate shifts. ReLU activation introduces non-linearity to the model, allowing it to learn complex patterns effectively. One of the key innovations of ResNet-50 is the use of skip connections, which create shortcuts that bypass one or more layers. This residual learning mechanism ensures better gradient flow, prevents the vanishing gradient problem, and allows the model to train deeper networks without a significant drop in performance. Additionally, global average pooling is employed to reduce the dimensionality of feature maps, summarizing the spatial information efficiently before passing it to the fully connected layer.

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 / 8/ 10 /11
- Language : Python
- IDE : Jupyter Notebook
- Libraries : Tensor flow , Kenras , numpy

5.2 HARDWARE REQUIREMENTS

- RAM : 8 GB
- PROCESSOR : i5 intel core
- MEMORY : 256 SSD and 1TB HDD

5.3 DEVELOPMENT ENVIRONMENT

The development environment comprises tools and platforms that facilitate efficient project development, code management, and model deployment. This section details the setup and tools used

- Jupyter IDEs: Writes, compiles, and uploads code. Library: Pre-written code for specific tasks. Debugger: Debugs code and identifies errors.
- Python IDEs: For larger-scale code development, IDEs like PyCharm or VS Code are recommended. These IDEs offer features such as syntax highlighting, error detection, and Git integration, which streamline code development and version control.

5.4 PYTHON: The Chosen Programming Language

Python is the primary programming language used in this Parkinson’s Disease detection project, chosen for its versatility, simplicity, and extensive support for machine learning and data science libraries. Python has become one of the most popular languages in data science and machine learning due to its rich ecosystem and readability, which helps streamline complex processes.

Python's syntax is intuitive, resembling plain English, making it accessible for beginners while being powerful enough for advanced programmers. The language’s readability simplifies the development process, allowing developers to focus more on problem-solving and model accuracy than on code syntax intricacies. Python also supports object-oriented, procedural, and functional programming paradigms, providing flexibility in code organization and structure.

Key Python Libraries

1. TensorFlow (tensorflow) – Used for building and training the convolutional network model.

2. keras(tensorflow.keras.preprocessing.image)–Used for image preprocessing and data augmentation.

3. NumPy (numpy) – Used for handling arrays and numerical operations. Computer Vision & Image Processing

4. Pillow (PIL) (PIL.Image, ImageTk) – Used for opening and displaying images in the GUI.

5. Matplotlib (matplotlib.pyplot) – Used for plotting model accuracy and loss graphs.

GUI Development

7. Tkinter (tkinter, filedialog, messagebox) – Used for building the graphical user interface (GUI).

PDF Generation & File Handling

8. FPDF (fpdf) – Used for generating PDF reports.

9. OS (os) – Used for file path handling and retrieving the working

Web Browser

10. Web browser (webbrowser) – Used to open the generated PDF report automatically.

5.5 FEATURES OF PYTHON

Python is a versatile and powerful programming language with unique features that make it well-suited for machine learning, data science, and other computational projects. Its features have contributed to its popularity, especially in research and data analysis fields. Key features of Python include:

PYTHON FEATURES

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation.

1) Easy to Learn and Use

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block.

2) Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

3) Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable. Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language.

It enables programmers to develop the software for several competing platforms by writing a program only once.

4) Free and Open Source

Python is freely available for everyone . It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

5) Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

6) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

7) Large Standard Library

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

8) GUI Programming Support

Graphical User Interface is used for the developing Desktop application. PyQT5,Tkinter, Kivy are the libraries which are used for developing the web application.

9) Integrated

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C,C++ Java. It makes easy to debug the code.

10) Embeddable

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well.It can embed other language into our code.

11) Dynamic Memory Allocation

In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time.

CHAPTER 6

IMPLEMENTATION AND RESULTS

6.1 DATA SPLITTING

In the process of eye disease detection using the ResNet-50 algorithm, data splitting plays a crucial role in ensuring the model's accuracy and generalization. The dataset, consisting of retinal images or ophthalmic medical records, is typically divided into three main subsets: training, validation, and testing.

The training set, usually comprising around 70-80% of the data, is used to train the ResNet-50 model by extracting deep hierarchical features and learning patterns that indicate the presence of diseases such as diabetic retinopathy, glaucoma, or macular degeneration. The validation set, accounting for approximately 10-15% of the data, is essential for fine-tuning hyperparameters and preventing overfitting by evaluating the model's performance during training.

6.2 DATA PREPROCESSING

In eye disease detection using the ResNet-50 algorithm, the data preprocessing stage is essential to ensure that the input images are clean, consistent, and suitable for effective model training. The process begins with Data Collection, where high-resolution retinal images are gathered from medical databases or clinical sources, including fundus photographs or Optical Coherence Tomography (OCT) scans. Following this, Image Preprocessing Techniques are applied to enhance image quality by performing noise reduction, contrast adjustment, and color normalization.

These steps help in improving image clarity and making the model more sensitive to subtle patterns indicative of diseases such as diabetic retinopathy, glaucoma, or macular degeneration. Since ResNet-50 requires fixed-size input images, typically 224x224 pixels, the images are resized while maintaining aspect ratios to minimize distortion.

Additional preprocessing steps include Data Augmentation using techniques like rotation, flipping, and brightness adjustments to expand the dataset, reduce overfitting, and improve model generalization. Furthermore, Normalization is applied by scaling pixel values to a range of [0, 1] or standardizing them based on the mean and standard deviation of the dataset. This ensures stable and efficient model convergence during training.

By performing these comprehensive preprocessing steps, ResNet-50 can effectively extract meaningful features from retinal images, leading to accurate and reliable eye disease detection.

6.3 CLASSIFICATION MODELS

Classification models for eye disease detection using the ResNet-50 algorithm are designed to analyze retinal images and accurately identify conditions such as diabetic retinopathy **(DR),** glaucoma **(GL),** and cataracts **(CAT)**. ResNet-50, a deep Convolutional Neural Network **(CNN)**, is particularly effective in extracting hierarchical features from images through its residual learning framework, which allows for the training of deeper networks without the risk of vanishing gradients. The model processes retinal images by passing them through multiple convolutional layers, capturing both low-level and high-level visual features.. The classification process involves training the model on labeled datasets, where ResNet-50 learns to distinguish between healthy and diseased cases. Evaluation metrics such as accuracy **(ACC),** precision **(PRE),** recall **(REC),** and F1-score **(F1)** are commonly used to measure the model's effectiveness. Additionally, techniques like cross-validation and confusion matrix analysis are applied to ensure robustness and minimize misclassifications. These classification models offer a reliable and automated approach for early detection of eye diseases, providing valuable support to ophthalmologists **(OPH**) in making timely and accurate clinical decisions.

SUMMARY OF MODEL COMPARISON

The comparison of eye disease detection using the Vision Transformer **(VIT)** algorithm involves evaluating different machine learning (ML) and deep learning **(DL)** models based on performance metrics such as accuracy (ACC), precision **(PRE)**, recall **(REC)**, and F1-score **(F1).** VITs, unlike traditional models, excel in capturing long-range dependencies and complex spatial relationships by using self-attention mechanisms. This makes them particularly effective in analyzing high-resolution retinal images for detecting diseases like diabetic retinopathy **(DR),** glaucoma **(GL),** and cataracts **(CAT).** Compared to conventional models such as Support Vector Machines **(SVM),** Random Forest **(RF),** and K-Nearest Neighbors **(KNN),** VITs demonstrate higher accuracy in image classification tasks due to their global feature extraction capabilities. While Convolutional Neural Networks **(CNNs)** are effective for local feature extraction, VITs outperform them in capturing intricate patterns across an entire image.

6.4 SOURCE CODE

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image
import ImageDataGenerator
from tensorflow.keras.applications
import ResNet50 from tensorflow.keras.layers
import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.models
import Model from tensorflow.keras.optimizers
import Adam from tensorflow.keras.callbacks
import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint

# Set Parameters
IMAGE_SIZE = (224, 224)
BATCH_SIZE = 32
EPOCHS = 30
# Increased for better training

# Data Augmentation
train_datagen = ImageDataGenerator(
rescale=1./255,
rotation_range=30,
width_shift_range=0.2,
height_shift_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
validation_split=0.2  # 80% train, 20% validation
)
```

```
# Load Data
train_data = train_datagen.flow_from_directory(
"fundus_dataset/",
# Change to your dataset path
target_size=IMAGE_SIZE,
batch_size=BATCH_SIZE,
class_mode="categorical",
subset="training"
)
val_data = train_datagen.flow_from_directory(
"fundus_dataset/",
target_size=IMAGE_SIZE,
batch_size=BATCH_SIZE,
class_mode="categorical",
subset="validation"
)
# Load Pretrained ResNet-50
base_model=ResNet50
(weights="imagenet",include_top=False,
input_shape=(224, 224, 3))
base_model.trainable = False
# Freeze base layers initially

# Add Custom Layers
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation="relu")(x)
```

```
x = Dropout(0.5)(x)
output_layer = Dense(train_data.num_classes, activation="softmax")(x)

# Create Model
model = Model(inputs=base_model.input, outputs=output_layer)
# Compile Model
model.compile(optimizer=Adam(learning_rate=0.0001),
loss="categorical_crossentropy", metrics=["accuracy"])

# Callbacks for Better Training
callbacks = [
EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True),
ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3,
min_lr=1e-6),
ModelCheckpoint("best_fundus_model.h5",
save_best_only=True, monitor='val_accuracy', mode='max')
]

# Train Model
history = model.fit(train_data, validation_data=val_data, epochs=EPOCHS,
callbacks=callbacks)

# Save Final Model
model.save("fundus_resnet50.h5")
print("✅ Model training complete and saved!")
```

GUI CODE

```
import tkinter as tk
from tkinter
import filedialog, messagebox from PIL import Image, ImageTk
import tensorflow as tf
import numpy as np
import cv2 from fpdf
import FPDF
import webbrowser
# Load the trained model
model = tf.keras.models.load_model("Myproject.keras")
# Define a dictionary to map class indices to disease names
class_names = {
0: "ACRIMA",     # Class 0 is Healthy
1: "cataract",    # Class 1 is Cataract
2: "Glaucoma",    # Class 2 is Glaucoma
3: "ODIR-5K",
4: "ORIGA",
5: "retina_disease",# Class 3 is Diabetic Retinopathy
# Add more diseases if applicable
}
# Function to preprocess image
from keras.preprocessing
import image
import numpy as np
def preprocess_image(img_path):
img = image.load_img(img_path, target_size=(150, 150))
  # Resize image to 150x150
```

```
img_array = image.img_to_array(img)  # Convert image to array
img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension
img_array /= 255.0  # Normalize the image
return img_array
# Function to predict disease
def predict_disease(img_path):
processed_img = preprocess_image(img_path)
 # Preprocess the image
prediction = model.predict(processed_img)
# Get prediction
disease_index = np.argmax(prediction)  # Get the index of the class with the
highest probability return class_names.get(disease_index, "Unknown")
# Return the corresponding disease name
# Function to generate PDF report
def generate_pdf(name, age, sex, phone, result):
pdf = FPDF()
pdf.add_page()
pdf.set_font("Arial", size=16)
pdf.cell(200, 10, "Eye Disease Detection Report", ln=True, align='C')
pdf.ln(10)
pdf.set_font("Arial", size=12)
pdf.cell(200, 10, f"Name: {name}", ln=True)
pdf.cell(200, 10, f"Age: {age}", ln=True)
pdf.cell(200, 10, f"Sex: {sex}", ln=True)
pdf.cell(200, 10, f"Phone: {phone}", ln=True)
pdf.cell(200, 10, f"Result: {result}", ln=True)
pdf.output("eye_disease_report.pdf", "F")
print("☑ PDF generated successfully!")  # Debugging print
```

```
messagebox.showinfo("Success", "PDF Report Generated Successfully!")
webbrowser.open("eye_disease_report.pdf")
# Function to show report window
def show_report_window(name, age, sex, phone, result):
report_window = tk.Toplevel(root)
report_window.title("Eye Disease Report")
report_window.geometry("400x300")
tk.Label(report_window, text="Eye Disease Detection Report", font=("Arial",
14, "bold")).pack(pady=10)
tk.Label(report_window, text=f"Name: {name}", font=("Arial", 12)).pack()
tk.Label(report_window, text=f"Age: {age}", font=("Arial", 12)).pack()
tk.Label(report_window, text=f"Sex: {sex}", font=("Arial", 12)).pack()
tk.Label(report_window, text=f"Phone: {phone}", font=("Arial", 12)).pack()
tk.Label(report_window, text=f"Result: {result}", font=("Arial", 12, "bold"),
fg="blue").pack(pady=10)
# GUI Setup
root = tk.Tk()
root.title("Eye Disease Detection")
root.geometry("500x600")
root.configure(bg="#f0f0f0")
# UI Elements
label_name = tk.Label(root, text="Enter Name:", bg="#f0f0f0", font=("Arial",
12))
label_name.pack()
entry_name = tk.Entry(root, font=("Arial", 12))
entry_name.pack()
label_age = tk.Label(root, text="Enter Age:", bg="#f0f0f0", font=("Arial", 12))
label_age.pack()
```

```
entry_age = tk.Entry(root, font=("Arial", 12))
entry_age.pack()
label_sex = tk.Label(root, text="Enter Sex:", bg="#f0f0f0", font=("Arial", 12))
label_sex.pack()
entry_sex = tk.Entry(root, font=("Arial", 12))
entry_sex.pack()
label_phone= tk.Label
(root, text="Enter Phone:", bg="#f0f0f0", font=("Arial", 12))
label_phone.pack()
entry_phone = tk.Entry(root, font=("Arial", 12))
entry_phone.pack()
label_image=tk.Label(root,text="UploadImage:",bg="#f0f0f0", font=("Arial",
12))
label_image.pack()
# Function to upload an image
def upload_image():
file_path = filedialog.askopenfilename()
if file_path:
img = Image.open(file_path)
img = img.resize((200, 200))
img = ImageTk.PhotoImage(img)
panel.config(image=img)
panel.image = img
panel.file_path = file_path  # Store file path in panel object
btn_upload=tk.Button
(root,text="Upload Image",font=("Arial", 12),
bg="#4CAF50", fg="white", command=upload_image)
btn_upload.pack(pady=5)
```

```
panel = tk.Label(root, bg="#f0f0f0")
panel.pack()
# Function to analyze image
def analyze_image():
name = entry_name.get()
age = entry_age.get()
sex = entry_sex.get()
phone = entry_phone.get()
# if not name or not age or not sex or not phone or not
hasattr(panel, 'file_path'):
messagebox.showerror("Error", "Please enter all details and upload an image.")
returnresult = predict_disease(panel.file_path)
# Generate and save PDF
generate_pdf(name, age, sex, phone, result)
# Show report in a new window
show_report_window(name, age, sex, phone, result)
btn_analyze=tk.Button
(root,text="Analyze & Generate Report", font=("Arial", 12), bg="#008CBA",
fg="white", command=analyze_image)
btn_analyze.pack(pady=10)
root.mainloop()
```

6.5 RESULT

OUTPUT 1: Figure 6.1 demonstrates to fill the patients details

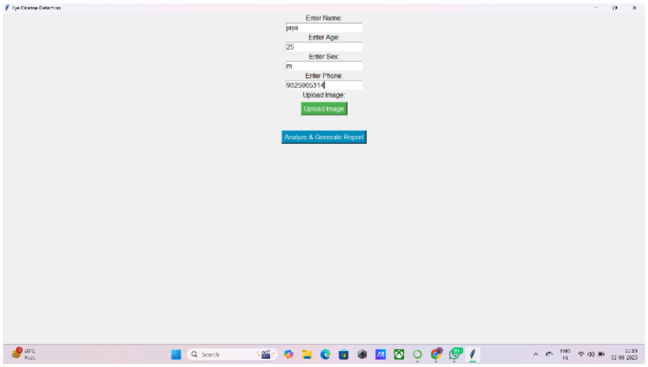

Figure: 6.1 process of upload details

OUTPUT 2: Figure 6.2 demonstrates to upload the image of the eye

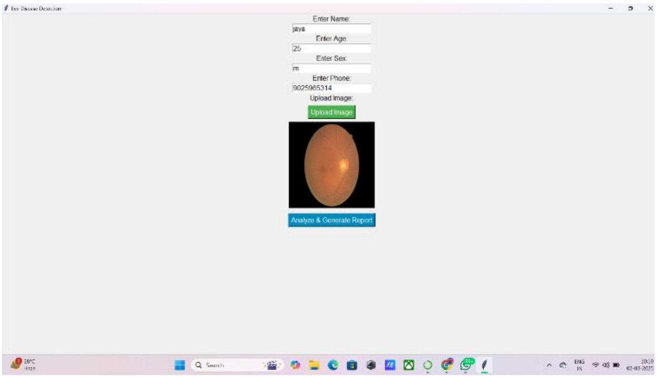

Figure : 6.2 image uploading process

OUTPUT 3: Figure 6.3 demonstrates to generate pdf report by uploaded Image

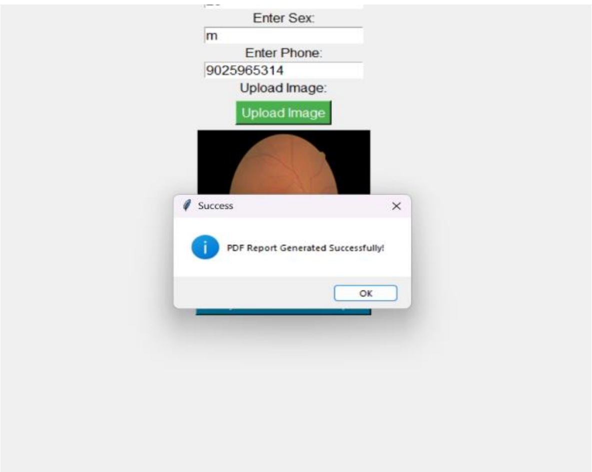

Figure : 6.3 Generated pdf

OUTPUT 4: Figure 6.4 demonstrates eye disease detection report

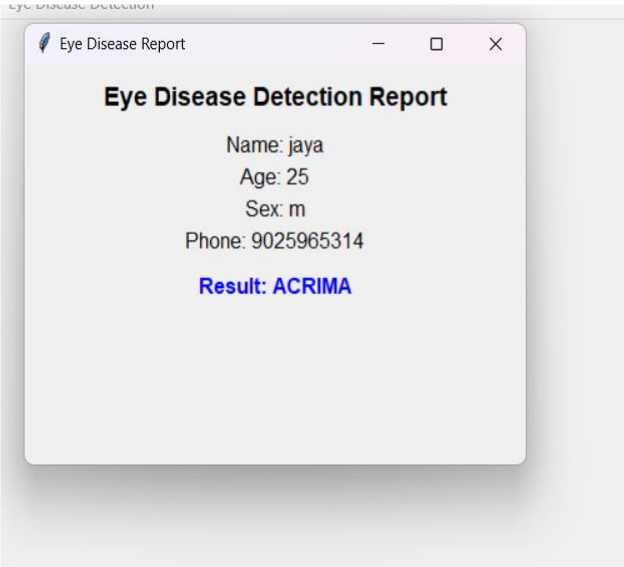

Figure: 6.4 minimized report for eye disease detection

OUTPUT 5 : Figure 6.5 demonstrates the report and result of the uploaded image

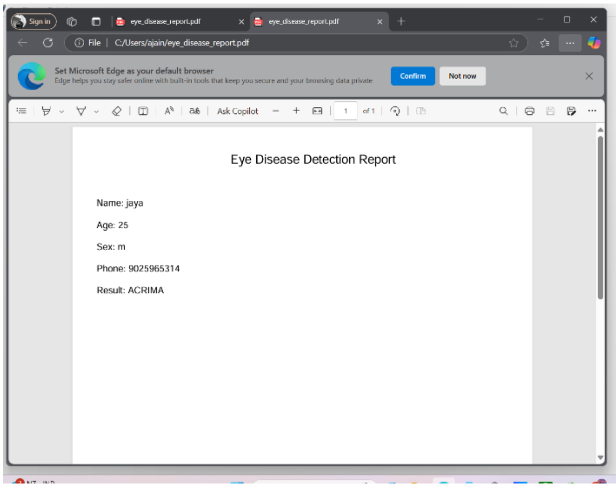

Figure: 6.5 report for Eye Disease detection

- **Figure 6.1:** Screenshot of a computer screen showing the picture of demonstrates to fill the patients details in the user interface.
- **Figure 6.2:** It represents the uploading of eye image in the report form. It is the process to identify whether it is defected .
- **Figure 6.3:** It demonstrates to generate pdf report by uploaded Image.
- **Figure 6.4:** It is the form of minimized report for eye disease detection
- **Figure 6.5:** It demonstrates the report and result of the uploaded image and give accurate result .

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In conclusion, using ResNet-50 algorithms for eye disease detection offers a promising future for proactive eye care. By analyzing these data, the future of eye disease detection using ResNet-50 holds significant potential for further advancements in accuracy, efficiency, and accessibility. One key area of enhancement is the integration of ResNet-50 with multimodal data, incorporating not only retinal images but also patient demographics, genetic data, and medical history.

The comprehensive approach can provide a deeper understanding of a patient's health, leading to more accurate and personalized diagnoses. Additionally, researchers are focusing on optimizing ResNet-50 models by developing lightweight versions that reduce computational complexity, enabling real-time disease detection on edge devices like smartphones and portable medical scanners. Another promising direction involves using transfer learning and fine-tuning techniques with pre-trained ResNet-50 models, minimizing the need for large annotated datasets. This allows the model to leverage prior knowledge from large-scale datasets like ImageNet, improving performance even with limited labeled medical data. These advancements in ResNet-50-based models are set to revolutionize eye disease detection, supporting ophthalmologists in delivering faster and more accurate diagnoses.

7.2 FUTURE ENHANCEMENT

Future enhancements for eye disease detection using ResNet-50 can significantly improve the accuracy, efficiency, and applicability of these systems. One major enhancement is the integration of ResNet-50 with multimodal data, combining retinal images with patient medical history, genetic information, and lifestyle data to provide a more comprehensive diagnosis.

Additionally, optimizing ResNet-50 through model compression and quantization techniques can reduce computational complexity, enabling real-time analysis on portable medical devices and smartphones for early disease detection. Another key enhancement involves improving the interpretability of ResNet-50 models using visualization methods such as Class Activation Maps (CAM) and Gradient-weighted Class Activation Mapping (Grad-CAM). These techniques can highlight regions of interest in retinal images, offering more transparent and explainable diagnostic decisions, thereby increasing trust among ophthalmologists and medical practitioners.

Federated learning can also be applied to ensure privacy-preserving model training across multiple healthcare institutions without exposing sensitive patient data. Furthermore, transfer learning and fine-tuning with domain-specific datasets can further enhance model performance, reducing the reliance on large annotated datasets. Incorporating continuous learning mechanisms will also allow ResNet-50 models to adapt to new data and evolving disease patterns, ensuring long-term accuracy. These advancements will make ResNet-50-based eye disease detection systems more robust, accessible, and reliable for early diagnosis and personalized treatment planning.

APPENDICES

PYTHON DEFINITION

Python is a high-level programming language designed to be easy to read and simple to implement. It is open source, which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to Java and .NET virtual machines.

Python is considered a scripting language, like Ruby or Perl and is often used for creating Web applications and dynamic Web content. It is also supported by a number of 2D and 3D imaging programs, enabling users to create custom plug-ins and extensions with Python. Examples of applications that support a Python API include GIMP, Inkscape, Blender, and Autodesk Maya.

Scripts written in Python (.PY files) can be parsed and run immediately. They can also be saved as a compiled programs (.PYC files), which are often used as programming modules that can be referenced by other Python programs.

PYTHON FEATURES

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation.

JUPYTER NOTEBOOK:

Interactive development environment:

Jupyter Notebook provides an interactive coding environment that supports live code execution, visualization, and documentation within a single interface. It is widely used for data analysis, machine learning, and research due to its ability to combine code, markdown, and outputs in a seamless manner.

Version control support:

Jupyter Notebook can be integrated with version control systems like Git, allowing users to track changes, manage different versions of notebooks, and collaborate on projects efficiently. Tools such as nbdime enhance the review process by providing better diff and merge capabilities for Jupyter notebooks.

Extensibility and customization:

Jupyter supports the development of extensions that enhance functionality, such as custom widgets, themes, and integrations with external tools. Custom extensions can be built to improve compatibility with code review tools, enabling a more structured and efficient review process.

Collaborative features:

Jupyter Notebook supports real-time collaboration through platforms like JupyterHub and JupyterLab, allowing multiple users to work on the same notebook simultaneously. Commenting features, shared execution environments, and interactive widgets facilitate smooth communication and feedback during the code review process.

Integration with machine learning frameworks:

Jupyter Notebook seamlessly integrates with popular machine learning frameworks such as TensorFlow, PyTorch, and Scikit-learn. This allows developers and researchers to experiment with models, visualize training progress, and fine-tune hyperparameters within the same interactive environment.

Interactive data visualization:

Jupyter Notebook supports rich data visualization capabilities through libraries like Matplotlib, Seaborn, Plotly, and Bokeh. These tools enable users to create interactive charts, graphs, and dashboards, making it easier to analyze and interpret complex datasets.

REFERENCES

1. M. A. Hossain, T. A. Asa, F. Huq , and M. A. Moni , ‘‘Eye disorders in Town Bangladesh A hospital-based descriptive study ,’’ J. Biomed. Anal., vol. 2 , Mar. 2019.
2. I. Sutradhar , P. Gayen , M. Hasan , R. D. Gupta , T. Roy , and M. Sarker , topic ‘Eye diseases :The neglected health condition among urban slum and Population of Dhaka , Bangladesh ,’’ BMC Ophthalmology , vol. 19 , no.1, Dec. 2019.
3. M. Yusufu , J. Bukhari , X. Yu , T. P. H. Lin , D. S. C. Lam , and N. Wang, ‘‘Challenges in eye care in the Asia–Pacific region ,’’ Asia – Pacific of J. Ophthalmology , vol. 10 , Sep. 2021.
4. D. Kermany , K. Zhang, and M. Goldbaum. (2018). Large Dataset of Labled Optical Coherence Tomography OCT and Chest X-ray Images.

5. L. Fang, Y. Jin, L. Huang, S. Guo, G. Zhao, and X. Chen, ‘‘Iterative fusion convolutional neural networks for classification of the OCT images,’’ J. Vis. Commun. Image Represent., vol. 59,Feb. 2019.
6. D. S.Kermany et al., ‘‘Identifying medical diagnoses and treatable disease by image-based deep learning,’’ Cell, vol. 172, ,Feb. 2018.
7. S. Zagoruyko and N. Komodakis,‘‘Identify Wide residual networks,’’ 2017.
8. V. Das, S. Dandapat, and P. K. Bora, ‘‘Multi-scale deep feature fusion for automated classification of macular pathologies from images and Biomed Signal Process. Control, vol. 54, Sep. 2019.
9. Healthcare Radius. (2040). Glaucoma Eye Disorder is Expected To Double in India By 2040. Accessed: Oct. 2, 2024.
10. A. S. M. Miah, Md. R. Islam, and M. K. I. Molla, ‘‘Motor imagery of the classification using sub band tangent space mapping ,’’ in Proc . 20th Int. Conf. Computer. Inf. Technol. (ICCIT), Dec. 2017.
11. A. S. M. Miah, S. R. A. Ahmed, M. R. Ahmed, O. Bayat,‘‘Motor-imagery BCI task classification using Riemannian geometry and averaging with a mean absolute deviation,’’ in Proc. Sci. Meeting Elect.-Electron. Boimed. Eng. Comput. Sci. (EBBT),Apr. 2019.

12. A. S. M. Miah, M. R. Islam, and M. K. I. Molla, “ EEG classification for MI-BCI using CSP with averaging covariance matrices: An experimental study,” in Proc. Int. Conf. Compute., Commun., Chem., Mater. Electron. Eng., Jul. 2019.

13. M. R. Rahman, M. T. Hossain, N. Nawal, M. S. Sujon, A. S. M. Miah,and M. M. Rashid , “ A comparative review of detecting Alzheimer’s disease using various methodologies,” BAUST J., vol. 2, Jun. 2020.

14. K. Atai Kibria , A. Sarker Noman , M. Abir Hossain , M. Shohi dul Islam Bulbul , M. Mamunur Rashid , and A. Saleh Musa Miah , “Creation of a cost - efficient and effective personal assistant robot using Arduino and machine learning algorithm,” in Proc. IEEE RegionSymp. (TENSYMP), Jun. 2020.

15. T. Zobaed, S. R. A. Ahmed, A. S. M. Miah, S. M. Binta, M. R. A. Ahmed, and M. Rashid, “Real time sleep onset detection from single channel EEG signal using block sample entropy ,” IOP Conf. Ser. Mater. Sci. Eng., vol. 928, May 2020.

16. Z. Li, Y. Han , and X. Yang , “Multi - fundus diseases classification using retinal optical coherence tomography images with Swin transformer V2 ,” J. Image., vol. 9, Sep. 2023.

17. T. He, Q. Zhou, and Y. Zou**,** ‘‘Automatic detection of age-related macular degeneration based on deep learning and local outlier factor algorithm ,’’ Diagnostics, vol. 12, Feb. 2022.

18. M. R. Rahman, M. T. Hossain, N. Nawal, M. S. Sujon, A. S. M. Miah, and M. M. Rashid , ‘‘ A comparative review of detecting Alzheimer’s disease using various methodologies, ’’ BAUST J., vol. 2, Jun. 2020.

19. M. S. Ali, J. Mahmud, S.M. F. Shahriar, S. Rahmatullah, and A.S. M. Miah ‘‘Potential disease detection using naive Bayes and random forest approach ,’’ BAUST J., vol. 1, 2022.

LIST OF PUBLICATIONS

1. Dr. R. Kavitha , A. Jain Raj , S. Kirthik , R. Jayachandran has presented A paper entitled "EYE DISEASE DETECTION USING DEEP LEARNING " in International conference on Artificial intelligence in health and medical Science (ICAIHMS' 2025) held on 22nd march 2025 organized by the Department of ECE and CSE , Karpagam institute of Technology , Coimbatore .
   PAPER ID: KIT276

KARPAGAM INSTITUTE OF TECHNOLOGY
Inspiring Innovation

Autonomous - Approved by UGC and Anna University, Chennai
Approved by AICTE and Affiliated to Anna University, Chennai
Accredited by NAAC with 'A++'
Accredited by NBA - (CSE, IT and ECE)
An ISO 9001:2015 and ISO 14001:2015 Certified Institution

ICAIHMS'25

CERTIFICATE OF PARTICIPATION

Certificate No. : KIT273

This is to certify that Mr./ Ms./ Dr. **Dr. R. Kavitha ,Associate Professor** of **P.A.COLLEGE OF ENGINEERING AND TECHNOLOGY** has participated and presented a paper titled **EYE DISEASE DETECTION USING DEEP LEARNING** in the AICTE sponsored Interational Conference on Artificial Intelligence in Health and Medical Science (ICAIHMS'25) organized by Department of ECE & CSE, Karpagam Institute of Technology, Coimbatore held on March 21 & 22, 2025.

Dr. K. KARUNAMBIGA	Dr. S. GOPINATH	Dr. P . MANIMARAN
Co-Convener	Convener	Principal

KARPAGAM
INSTITUTE OF TECHNOLOGY
Inspiring Innovation

Autonomous - Approved by UGC and Anna University, Chennai
Approved by AICTE and Affiliated to Anna University, Chennai
Accredited by NAAC with 'A++'
Accredited by NBA - (CSE, IT and ECE)
An ISO 9001:2015 and ISO 14001:2015 Certified Institution

ICAIHMS'25

CERTIFICATE OF PARTICIPATION

Certificate No. : KIT274

This is to certify that Mr./ Ms./ Dr. **Jain Raj A** of **P.A.COLLEGE OF ENGINEERING AND TECHNOLOGY** has participated and presented a paper titled **EYE DISEASE DETECTION USING DEEP LEARNING** in the AICTE sponsored Interational Conference on Artificial Intelligence in Health and Medical Science (ICAIHMS'25) organized by Department of ECE & CSE, Karpagam Institute of Technology, Coimbatore held on March 21 & 22, 2025.

Dr. K. KARUNAMBIGA	Dr. S. GOPINATH	Dr. P . MANIMARAN
Co-Convener	Convener	Principal

KARPAGAM
INSTITUTE OF TECHNOLOGY
Inspiring Innovation

Autonomous - Approved by UGC and Anna University, Chennai
Approved by AICTE and Affiliated to Anna University, Chennai
Accredited by NAAC with 'A++'
Accredited by NBA - (CSE, IT and ECE)
An ISO 9001:2015 and ISO 14001:2015 Certified Institution

ICAIHMS'25

CERTIFICATE OF PARTICIPATION

Certificate No. : KIT275

This is to certify that Mr./ Ms./ Dr. **Kirthik S**

of **P.A.COLLEGE OF ENGINEERING AND TECHNOLOGY** has participated and presented a paper titled **EYE DISEASE DETECTION USING DEEP LEARNING** in the AICTE sponsored International Conference on Artificial Intelligence in Health and Medical Science (ICAIHMS'25) organized by Department of ECE & CSE, Karpagam Institute of Technology, Coimbatore held on March 21 & 22, 2025.

Dr. K. KARUNAMBIGA
Co-Convener

Dr. S. GOPINATH
Convener

Dr. P . MANIMARAN
Principal

KARPAGAM
INSTITUTE OF TECHNOLOGY
Inspiring Innovation

Autonomous - Approved by UGC and Anna University, Chennai
Approved by AICTE and Affiliated to Anna University, Chennai
Accredited by NAAC with 'A++'
Accredited by NBA - (CSE, IT and ECE)
An ISO 9001:2015 and ISO 14001:2015 Certified Institution

ICAIHMS'25

CERTIFICATE OF PARTICIPATION

Certificate No. : KIT276

This is to certify that Mr./ Ms./ Dr. **Jayachandran R** of **P.A.COLLEGE OF ENGINEERING AND TECHNOLOGY** has participated and presented a paper titled **EYE DISEASE DETECTION USING DEEP LEARNING** in the AICTE sponsored Interational Conference on Artificial Intelligence in Health and Medical Science (ICAIHMS'25) organized by Department of ECE & CSE, Karpagam Institute of Technology, Coimbatore held on March 21 & 22, 2025.

Dr. K. KARUNAMBIGA	**Dr. S. GOPINATH**	**Dr. P . MANIMARAN**
Co-Convener	Convener	Principal