A FEATURE-OPTIMIZED ENSEMBLE MODEL FOR DIABETIC RETINOPATHY DETECTION VIA CNN AND APSO INTEGRATION

A Project Report submitted in the partial fulfilment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Shaik Meera Jasmine (22471A05D0)

Lakshmi Prasanna Pentyala (22471A05B9)

Gorre Jayasri (22471A0591)

Under the esteemed guidance of

Abburi Ramesh M.Tech.,

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under

Tyre -1 an ISO 9001:2015 Certified

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

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601 2025-2026

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

CERTIFICATE

This is to certify that the project that is entitled with the name "A FEATURE-OPTIMIZED ENSEMBLE MODEL FOR DIABET-IC RETINOPATHY DETECTION VIA CNN AND APSO INTEGRATION" is a Bonafide work done by Shaik Meera Jasmine (22471A05D0), Pentyala Lakshmi Prasanna (22471A05B9), Gorre Jayasri (22471A0591) in partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2025-2026.

PROJECT GUIDE

Abburi. Ramesh, M.Tech.,
Assistant Professor

PROJECT CO-ORDINATOR

Dr. M. Sireesha, M.Tech., Ph.D.,
Associate Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled " A FEATURE-OPTIMIZED ENSEMBLE MODEL FOR DIABET-IC RETINOPATHY DETECTION VIA CNN AND APSO INTEGRATION" is composed by me that the work contains here is my own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

Shaik Meera Jasmine (22471A05D0)
Lakshmi Prasanna Pentyala (22471A05B9)
Gorre Jayasri (22471A0591)

ACKNOWLEDGEMENT

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

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

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

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

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

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

By

Shaik Meera Jasmine (22471A05D0)
Lakshmi Prasanna Pentyala (22471A05B9)
Gorre Jayasri (22471A0591)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

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

INSTITUTION MISSION

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

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

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

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

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

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

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

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

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

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

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

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

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

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

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

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

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

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

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual,

and as a member or leader in diverse/multi-disciplinary team.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

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

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

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

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

Course Outcomes – Program Outcomes mapping

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

Course Outcomes – Program Outcome correlation

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

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

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

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

Name of the course from which principles are apply in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop model for detection and classification of OSCC	PO1, PO3,PO8
CC421.1, C2204.3, C22L3.2	Each and every requirement i critically analyzed, the process mode is identified	PO2, PO3, PO8
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9, PO8
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5, PO8
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10, PO8
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO8,PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the social media users and in future updates in our project can be done based on detection for Oral Cancer	PO4, PO7, PO8
C32SC4.3	The physical design includes website to check OSCC	PO5, PO6, PO8

ABSTRACT

Diabetic retinopathy (DR) is a progressive microvascular complication of diabetes and remains one of the primary causes of preventable blindness worldwide. Timely detection through automated screening systems is therefore critical to reduce vision loss, especially in large-scale and resource-limited clinical settings. In this study, a hybrid DR detection framework is proposed that integrates deep learning–based feature extraction with swarm intelligence–driven feature optimization to improve classification robustness and interpretability. Initially, retinal fundus images undergo preprocessing steps including contrast enhancement and segmentation to highlight lesion-prone regions such as microaneurysms, hemorrhages, and exudates. Discriminative deep features are then extracted using two complementary convolutional neural network architectures, namely GoogLeNet and a modified ResNet-16, enabling the capture of both multi-scale and residual representations. To eliminate redundant and irrelevant features while preserving discriminative power, Adaptive Particle Swarm Optimization (APSO) is employed as an efficient feature selection strategy. The optimized feature set is subsequently classified using multiple traditional machine learning classifiers, including Random Forest, Support Vector Machine, Naïve Bayes, and Decision Tree models. Experimental evaluation conducted on the Kaggle EyePACS dataset demonstrates that the Random Forest classifier consistently outperforms the others in terms of accuracy, precision, recall, F1-score, and area under the ROC curve. Furthermore, the proposed framework incorporates LIME-based explainability to generate interpretable predictions, enhancing clinical trust and transparency. Overall, the results indicate that the proposed hybrid approach achieves competitive, state-of-the-art performance and offers a reliable, scalable, and explainable solution for automated diabetic retinopathy screening.

INDEX

LIST OF FIGURES

LIST OF TABLES

1.INTRODUCTION

Diabetic retinopathy, a chronic ocular disease, can result in permanent loss of eyesight when ignored or untreated, will occur in patients with a long history of diabetes. AIbased computer-aided diagnosis has been proposed as a potential alternative to manual fundus image screening that involves intensive human intervention and susceptible to errors due to factors influenced by human intervention. In this context of growing interest, our work proposes an a comprehensive deep learning framework designed to detect evaluating fundus scans for diabetes-induced retinal damage images by using a structured approach in Python and PyTorch, and implemented on a Colab-based environment.. The open dataset on Kaggle facilitated the the dataset of this work, which is a collection of labeled photos that have been labeled as ”DR-Diagnosis” and ”No DrDiagnosis” according to the Kaggle repository terms of use For a better practice of medical image classification we started by pre-process and partitionedThe data was split into three distinct groups: one for training the model, another for validating its performance during development, and a final set reserved for testing accuracy after training, respectively. We transfer learn from (pre-trained) CNN architectures, which are then finetuned for binary classification, following the footsteps of the ResViT Fusion-Net [1] and EffNet-SVM [2] models. Images are passed through an optimized pipeline to train on the GPU after resizing to a fixed shape and converting them into tensors. The benefits of employing transformer and hybrid models for DR detection have been validated by multiple studies. For instance, Shahzad et al. [3] highlight the need for transparency in models in a clinical environment, whereas MSA Mix-Net [4] and CAD-ViT [5] employ attention mechanism to extract more intricate retinal details. Our code organization follows this pattern: similar as [6] [7], our pipeline, is built to allow explainability tools like LIME for interpretability, in addition to classification accuracy.Generally speaking, the aim of this practice is to establish a reproducible and explainable DR classification model, which accords with the current research tendency, e.g., multimodal fusion [8], model explainability of the model [9] and early-stage diagnosis improvement [10]. By leveraging research-based-approach, this work captures additional momentum from the increasing number of publications on

AIenabled healthcare environment. W. Nazeih et al. [11] also explored ViT for DR severity prediction and observed its efficacy in complex image analytics.

M. Jabbar et al. [12] highlighted that the lesionbased features often become useful when used together with hybrid deep learning models.

M. Asif et al. [13] emphasized timely detection using both traditional and AI-based systems.

D. Bhulakshmi and P. Kumar [14] conducted a systematic review on DR detection with AI. A hyper-tuned Vision Transformer (ViT) hybrid model for binary DR classification was employed by M. S. Ali, M. K. Islam [15] that was a lightweight yet powerful approach.

J. H. L. Goh et al. [16] further improved the same concept with CAD-ViT which combined coordinate attention and dilated feature consolidation for structural abnormality detection.

M. Kannan [17], and A. Senapati et al. [18] emphasized the interpretability and transparency of the model application in the clinical setting with AI. Their research contributes to the overall need for objects that can diagnose in a clear cut way. Quellec et al. [19] built graph-based and multimodal learning-based models for high NP accuracy, often sacrificing transparency in the process.

Md. S. H. Talukder et al. [20] developed MSAMix-Net, a multi-scale attention based one, but it did not have interpretability in the design.

Se In Jang et al. [21] presented an explainable AI-based diagnostic model by achieving accuracy and traceability. These types of studies demonstrate the trade-off that needs to be struck in AI applications between clinical trust and predictive power. It should be noted that in these works, while one part is good accuracy and interpretability or multiclass classification, but there is no work that can achieve both high accuracy and interpretability in a single work.

Sireesha, M., [23] proposed a hybrid model using optimized feature extraction for disease prediction. Unlike their approach, our framework integrates deep learning with interpretability for enhanced DR detection.

1.1 MOTIVATION

Diabetic Retinopathy (DR) is one of the most serious complications of diabetes and a leading cause of preventable blindness worldwide. With the rapid rise in diabetes prevalence, especially in developing countries, the burden of DR on patients and healthcare systems has grown significantly. Early detection and intervention are crucial, as untreated DR can progress silently and lead to irreversible vision loss.

Traditional screening methods rely on manual interpretation of retinal fundus images by ophthalmologists. While effective, this process is time-consuming, costly, and prone to human error, particularly when analyzing large-scale screening data or subtle earlystage lesions. In resource-limited areas, the shortage of trained specialists further delays timely diagnosis, leaving many patients undiagnosed until the disease has advanced.

Advances in medical imaging and Artificial Intelligence (AI) have paved the way for automated DR detection systems. Deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable capabilities in learning complex visual patterns such as microaneurysms, hemorrhages, and exudates. However, CNNs alone can be computationally intensive and may suffer from overfitting when applied to large and imbalanced datasets like EyePACS.

To address these challenges, this project proposes a **hybrid approach combining CNN-based feature extraction with optimized machine learning classifiers enhanced by Adaptive Particle Swarm Optimization (APSO)**. This integration is designed to reduce diagnostic time, minimize human error, and provide high accuracy in distinguishing DR from non-DR cases. By leveraging preprocessing techniques such as contrast enhancement, green channel extraction, and median filtering, the model ensures improved image clarity and feature visibility.

In addition, this research emphasizes **explainability and trustworthiness of AI systems** in medical applications. Beyond achieving high accuracy, the model

integrates feature selection and interpretability measures to provide insights into how decisions are made. This transparency builds confidence among clinicians, ensuring that the system is not only a diagnostic tool but also a decision-support system that can work alongside experts to validate findings.

Finally, the scalability and accessibility of this approach make it suitable for **deployment in real-world healthcare systems and telemedicine platforms**. In rural and underserved regions where ophthalmologists are scarce, an automated DR detection system can enable early screening, prioritize patients requiring urgent care, and drastically reduce preventable blindness. Thus, the motivation for this project extends beyond technical advancement, aiming to bridge the gap between modern AI technologies and practical, life-saving healthcare solutions.

1.2 PROBLEM STATEMENT

Diabetic Retinopathy (DR) is a severe and progressive microvascular complication of diabetes mellitus that can cause irreversible vision loss if not identified and managed in time. It arises from damage to the small blood vessels in the retina, leading to leakage, hemorrhages, and in advanced stages, abnormal vessel growth. The impact of DR is profound, not only on the quality of life of patients but also on healthcare systems, as it often leads to long-term treatments, expensive surgeries, or permanent disability. Despite being preventable in its early stages, DR remains one of the leading causes of blindness in working-age adults worldwide.

The challenge in combating DR lies in its silent progression. Patients may remain asymptomatic in the early phases, but by the time symptoms such as blurred vision or vision loss appear, the retina is often already severely damaged. Early and accurate diagnosis is therefore essential to preserve vision. Unfortunately, conventional diagnostic approaches are dependent on manual examination of fundus images by ophthalmologists, which is both time-consuming and subjective. The shortage of skilled specialists in rural and under-resourced areas makes this task even more difficult, leaving millions of patients undiagnosed until irreversible damage occurs.

Another critical problem is the **complexity of retinal features**. Lesions such as microaneurysms, cotton wool spots, hemorrhages, and exudates vary greatly in size, shape, and brightness across patients and stages of disease. Even subtle differences can affect classification, and distinguishing between mild and severe stages of DR can be challenging even for trained experts. The large-scale datasets, such as EyePACS, provide an opportunity for automated detection, but the variability in image quality, illumination, and presence of artifacts adds further complications to reliable classification.

Existing automated systems have shown promise but face limitations such as overfitting, lack of interpretability, or reduced accuracy in real-world clinical environments. Pure deep learning approaches often require very large annotated datasets and high computational resources, while traditional machine learning techniques may lack the ability to capture complex retinal structures. This creates a gap for hybrid solutions that can combine the strengths of deep feature extraction with robust classification methods while also ensuring transparency and explainability for medical use.

1.3 OBJECTIVE

The primary objective of this project is to design and implement an automated detection and classification system for Diabetic Retinopathy (DR) using retinal fundus images. The system aims to classify images into healthy and DR-affected categories, with the ability to distinguish different severity levels of the disease. The motivation behind this objective is to address the limitations of traditional manual screening methods, which are time-consuming, subjective, and highly dependent on the availability of trained ophthalmologists. By leveraging artificial intelligence, the system seeks to improve the efficiency, speed, and reliability of DR diagnosis.

Another key objective is to develop a hybrid deep learning and machine learning framework that combines the feature extraction power of Convolutional Neural Networks (CNNs) and transfer learning models such as GoogLeNet and ResNet with the optimization capabilities of Adaptive Particle Swarm Optimization (APSO). This hybridization ensures that only the most relevant features are selected

for classification, thereby improving the accuracy and robustness of the model while reducing computational complexity. Furthermore, classifiers like Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes, and Decision Tree will be evaluated to determine the most effective algorithm for final decision-making.

A crucial objective of this research is to enhance image quality through systematic preprocessing techniques. Steps such as resizing, green channel extraction, CLAHE for contrast enhancement, median filtering for denoising, and morphological operations will be applied to ensure high-quality input images. The objective here is to reduce noise, standardize image conditions, and highlight important retinal features such as microaneurysms, hemorrhages, and exudates that are essential for accurate classification.

Beyond achieving technical accuracy, another important objective is to build an explainable and trustworthy diagnostic tool. By incorporating interpretability frameworks like LIME (Local Interpretable Model-Agnostic Explanations), the project aims to make the decision-making process transparent. This ensures that clinicians not only receive predictions but also gain insights into which regions of the retina influenced the model's output. This transparency is vital for increasing adoption in clinical settings, where medical practitioners require both reliability and interpretability.

The project also aims to develop a system that is scalable and adaptable to realworld healthcare environments. This includes the ability to handle large datasets such as EyePACS efficiently, integration with cloud-based platforms for telemedicine applications, and potential deployment in mobile or web-based screening tools. The ultimate goal is to support mass screening programs, particularly in rural and underserved regions where ophthalmologists are scarce, enabling early detection and reducing the number of undiagnosed cases.

Finally, the overarching objective is to create a solution that contributes to better patient outcomes and healthcare efficiency. By enabling early detection of DR, the project seeks to prevent avoidable blindness, reduce treatment costs, and support doctors in making timely and accurate decisions. Long-term, the system can be expanded to include multi-disease detection capabilities, integration with electronic health record (EHR) systems, and continuous improvements through real-time

feedback. Thus, the project not only addresses the immediate need for DR detection but also lays the foundation for broader applications of AI in medical imaging and preventive healthcare.

2 LITERATURE SURVEY

Diabetic Retinopathy (DR) has attracted significant research attention in recent years due to its growing impact as one of the leading causes of blindness worldwide. The condition results from prolonged diabetes, where damage to retinal blood vessels leads to microaneurysms, hemorrhages, exudates, and eventually vision loss if untreated. With the global prevalence of diabetes steadily rising, DR is expected to affect millions more in the coming decades, making early detection and timely treatment a critical healthcare priority. Traditional screening methods rely on manual examination of retinal fundus images by ophthalmologists, which is often time-consuming, subjective, and resource-intensive, particularly in regions with limited access to specialists.

Automated detection methods have therefore evolved from traditional machine learning to advanced deep learning–based systems. Early approaches focused on handcrafted feature extraction, where descriptors such as texture, color histograms, and vascular morphology were combined with classifiers like Support Vector Machines (SVMs) and Random Forests (RFs). While effective on small, curated datasets, these methods struggled with generalization, illumination variability, and subtle lesion detection. In contrast, deep learning approaches, particularly Convolutional Neural Networks (CNNs), have revolutionized medical image analysis by learning hierarchical representations directly from data. These models can automatically identify clinically relevant features such as microaneurysms, hemorrhages, and exudates with significantly higher accuracy and robustness compared to classical techniques.

Recent studies have also highlighted the role of transfer learning, hybrid architectures, and optimization algorithms in pushing the performance boundaries of DR detection systems. Transfer learning allows pretrained networks such as VGG, ResNet, and Inception to adapt effectively to retinal images, even when labeled data is limited. Hybrid approaches combine CNN-based feature extraction with classical classifiers or optimization methods such as Particle Swarm Optimization (PSO), leading to better feature selection and reduced overfitting. Furthermore, the integration of explainable AI techniques has enhanced model

interpretability, enabling clinicians to visualize which retinal regions influenced the decision and thereby building trust in automated systems.

As a result, DR detection research has shifted from proof-of-concept experiments to clinically viable systems that aim to assist ophthalmologists, reduce diagnostic time, and enable scalable screening programs in remote or underserved areas. The following studies highlight key contributions and advancements that have shaped the field of automated Diabetic Retinopathy detection.

Solanki S. [1] investigated CNN-based approaches for Diabetic Retinopathy (DR) detection and demonstrated that deep convolutional architectures such as ResNet and VGG significantly outperform classical classifiers like SVM and Random Forest on fundus-image tasks. The study quantified improvements in automated feature learning, reporting roughly 10–15% absolute gains in classification performance versus handcrafted pipelines, and showed that transfer learning from ImageNet-pretrained weights further boosts accuracy by another 5–10% when fine-tuned on retinal datasets.

M. Imran [2] proposed a Deep Convolutional Neural Network (DCNN) augmented with a three-stage preprocessing pipeline (contrast enhancement, denoising, and normalization). The work emphasized that careful preprocessing materially increases lesion visibility (microaneurysms, hemorrhages, exudates) and demonstrated superior DR classification accuracy compared to baseline VGG16/VGG19 implementations, highlighting preprocessing as a low-cost yet high-impact intervention.

Ahmad S. and Choudhury P.K. [3] performed a systematic review of machine learning and deep learning methods for retinal disease detection, concluding that pretrained models such as Inception-v3 and DenseNet201 routinely achieve >90% accuracy on benchmark splits. Their survey also identified practical barriers—computational cost, fine-tuning complexity, and inconsistent preprocessing—that limit ready clinical translation.

Shah H. A. [4] introduced a multimodal fusion framework combining color fundus images with complementary clinical or imaging signals, showing that fusing modalities increases robustness to false positives and improves sensitivity in difficult cases. The study provided empirical evidence that multimodal inputs can

partially compensate for low-quality fundus images in resource-constrained screening programs.

A study using EfficientNet-B0 [5] showed that modern compound-scaling CNNs strike an attractive balance between parameter efficiency and accuracy for DR classification. When combined with aggressive augmentation and transfer learning, EfficientNet variants generalized better across varied fundus image sources, reducing overfitting on small labeled cohorts.

M. Agrawal [6] applied Inception V3 with fine-tuning to DR severity grading and reported notable improvements in both stage-wise accuracy and area-under-ROC compared to simpler CNNs. The work emphasized multi-scale feature extraction—Inception modules' capacity to capture both local lesion and global retinal context—as central to performance gains.

Sireesha M. [7] evaluated standard CNN pipelines on multi-grade DR classification and demonstrated that deep learning frameworks can reliably identify early signs of retinopathy when trained with appropriate augmentation and class-balancing strategies. The paper underlined the need for stratified evaluation metrics (sensitivity at lowprevalence grades) for screening utility.

An adaptation of Faster R-CNN for lesion localization in fundus images [8] applied region proposal networks to detect candidate microaneurysms and hemorrhages, producing high-precision lesion proposals useful for downstream grading and explainability. This object-detection perspective reduced false alarms relative to slidingwindow methods and enabled faster, localized clinician review.

Radhi A. A. [9] proposed an optimized pipeline emphasizing lightweight preprocessing and fast segmentation for near-real-time DR detection. By streamlining ROI extraction and threshold-based segmentation, the study achieved competitive accuracy with substantially reduced inference latency—an important factor for deployment in low-resource screening camps.

Toğaçar, Ergen, and Cömert [10] developed a task-specific CNN (DRNet) tailored to fundus images, integrating custom convolutions and augmentation policies. Their experiments showed robustness to image artifacts and improved performance

on small public datasets after applying targeted augmentation and class reweighting.

Sharma, Kaur, and Gujral [11] compared classical ML (SVM, RF) and deep learning approaches for DR detection, demonstrating that while SVM/RF can perform well with carefully engineered features, deep CNNs generally provide higher end-to-end accuracy and eliminate the need for manual descriptor design—suggesting hybrid pipelines where CNN features feed into classical classifiers may be advantageous in some operational contexts.

Moturi, Tirumala Rao, and Vemuru [12] presented an ensemble methodology emphasizing optimized feature selection and classifier fusion. Although their primary applications were other medical domains, their findings on ensemble stability and feature-selection benefits are directly applicable to DR systems seeking to balance sensitivity and specificity through classifier diversity.

Islam et al. [13] introduced a segmentation-centric approach using superpixels, PCA, and template-based K-means clustering for efficient lesion localization and dimensionality reduction. Their method reduced computational cost during preprocessing while maintaining high segmentation fidelity—valuable when subsequent classifiers rely on localized lesion descriptors.

Amin et al. [14] explored advanced filtering and handcrafted feature pipelines, demonstrating that novel preprocessing (adaptive denoising, vessel-aware enhancement) can noticeably improve the downstream performance of both classical and deep classifiers—especially when combined with limited fine-tuning of pretrained networks.

Abiwinanda et al. [15] proposed a CNN classification framework that integrated dropout and batch-normalization to enhance generalization on small retinal datasets. Their experiments showed that modest architectural regularization yields clear robustness gains and reduces variance across cross-validation folds.

Setha and Raja [16] empirically validated transfer learning strategies for DR classification, showing that even shallow fine-tuning of deep backbones yields substantial gains if coupled with dataset-specific augmentation. The study provided

practical guidance on layer freezing schedules and learning-rate schedules tailored to fundus images.

Hassan Ali Khan [17] compared an in-house lightweight CNN with pretrained heavyweights (VGG-16, ResNet-50, Inception-v3) and found that a carefully designed compact model can match or exceed pretrained architectures in accuracy while consuming less compute and energy—an important result for on-device screening applications.Attention-mechanism studies [18] applied channel-wise and spatial attention modules to fundus CNNs and reported improvements in focusing the model on clinically relevant regions (e.g., macula, vascular arcades). Attention modules consistently boosted AUC and localization quality, especially for subtle lesions that require focused receptive fields.

Differential Deep-CNN designs [19] experimented with architectural variants and hybrid CNN-SVM pipelines, showing that combining CNN feature extractors with SVM classifiers can yield improved margin properties and stability on skewed class distributions typical of DR datasets. Some hybrid pipelines reported accuracy gains while keeping inference pipelines interpretable for clinician oversight.

Explainability-focused work [20] incorporated LIME, Grad-CAM and integrated gradients into DR models to produce saliency maps highlighting lesion contributions to predictions. These studies emphasized that transparent, visual explanations significantly increase clinician trust and facilitate error analysis in misclassified cases.

Optimization and feature-selection research [21] applied Particle Swarm Optimization (PSO) and Adaptive PSO (APSO) to CNN-derived feature vectors, demonstrating that metaheuristic selection can dramatically reduce dimensionality while retaining or improving classification accuracy—making downstream classifiers faster and less prone to overfitting.

The rise of Vision Transformers and hybrid ViT-CNN models [22] has recently influenced DR research: transformer blocks capture long-range dependencies in retinal images and, when fused with CNN local feature extractors, improve both staging accuracy and robustness to inter-image variability. Early ViT adaptations for fundus images reported promising gains on multi-center splits.

Generative approaches using GANs [23] have been used to synthesize realistic fundus images for augmentation, addressing class imbalance and rare lesion patterns. Studies found that GAN-augmented training sets improve sensitivity for underrepresented DR grades and help models generalize better across datasets.

Domain-adaptation and multi-center generalization studies [24] evaluated models across EyePACS, Messidor, APTOS, and IDRiD datasets, highlighting that crossdomain shifts (camera, illumination, patient demographics) reduce naive model performance and that domain adaptation techniques (adversarial alignment, stain normalization) are necessary for robust, real-world deployment.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Diabetic Retinopathy (DR) detection and classification have traditionally depended on manual examination of retinal fundus images by trained ophthalmologists and graders. While clinical evaluation remains the gold standard, manual screening is timeconsuming, subjective, and difficult to scale for mass screening programs. To address these limitations, computational pipelines—ranging from classical machine learning to modern deep learning—have been developed to automate and accelerate DR screening with the aims of improving throughput, consistency, and early detection rates.

Early computational systems used handcrafted feature extraction followed by conventional classifiers. Preprocessing and handcrafted descriptors (vessel morphology, texture features, color histograms, and lesion-specific measures) were created from fundus images and fed to classifiers such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Random Forests (RF) and Decision Trees. These approaches achieved moderate performance on small, curated datasets but suffered from limited generalization, dependency on manually engineered features, and sensitivity to illumination, contrast, and acquisition artefacts.

The advent of Convolutional Neural Networks (CNNs) changed the landscape by automating feature learning directly from images. Architectures such as VGG, Inception (GoogLeNet), ResNet and EfficientNet have been adapted for binary DR detection and multi-grade severity classification. Transfer learning—fine-tuning ImageNet-pretrained networks on retinal datasets—helped overcome limited labeled data, reduced training time, and improved accuracy. Hybrid pipelines that extract features from pretrained CNNs and then use classical classifiers (e.g., CNN → SVM or CNN → RF) have also been proposed to combine powerful representation learning with robust decision boundaries.

Typical existing DR detection pipelines follow a sequential image-processing workflow (illustrated in Fig. 3.1 for the existing system). The pipeline begins with an input database of fundus images that undergo preprocessing steps to standardize

and enhance image quality. Common preprocessing operations include resizing, green-channel extraction (to improve vessel and lesion contrast), Contrast Limited Adaptive Histogram Equalization (CLAHE) for local contrast enhancement, median or Gaussian filtering for denoising, and background removal or cropping (ROI extraction) to focus on the retinal disk. These steps reduce intra- and inter-image variability and make downstream lesion features more prominent.

After preprocessing, segmentation modules identify important retinal structures and lesions. Both traditional methods like thresholding and clustering, and deep learning models such as U-Net are used. The segmented regions help in feature extraction or are directly used as inputs for detection and classification models.

Feature extraction follows segmentation (or is performed end-to-end in modern CNN systems). In classical systems this means computing texture, shape, color and vessel features; in deep learning systems it means extracting learned feature vectors from intermediate CNN layers (bottleneck or global-pool representations). Feature selection/dimensionality-reduction techniques such as Principal Component Analysis (PCA) or metaheuristic optimizers (e.g., Particle Swarm Optimization) are often applied to reduce redundancy and improve classifier efficiency.

The classification stage assigns labels (No-DR / DR or multi-grade DR categories) using methods such as Random Forests, SVM, Naïve Bayes, Decision Trees, or end-to-end CNN softmax layers. Hybrid strategies—CNN feature extractor + APSO (or PSO) for feature selection + classical classifier—are increasingly common, as they can improve accuracy and reduce computational load. Evaluation is generally performed using accuracy, precision, recall (sensitivity), specificity, F1-score, ROC-AUC and Cohen's kappa, often with stratified train/validation/test splits or k-fold cross validation to control for class imbalance.

Although existing systems have achieved high performance on benchmark datasets such as **EyePACS, Messidor, and APTOS** [1], they face several practical challenges including dataset imbalance and varying image quality across centers [2], overfitting when models are trained on single-center data [3], high computational requirements for large CNNs [4], and limited clinical trust due to the "black-box" nature of predictions [5]. To mitigate these issues, current best practices emphasize robust preprocessing [6], data augmentation [7], feature

optimization [8], ensemble or hybrid classifiers [9], and incorporation of explainability tools such as Grad-CAM and LIME [10] so that model decisions can be visualized and validated by clinicians.

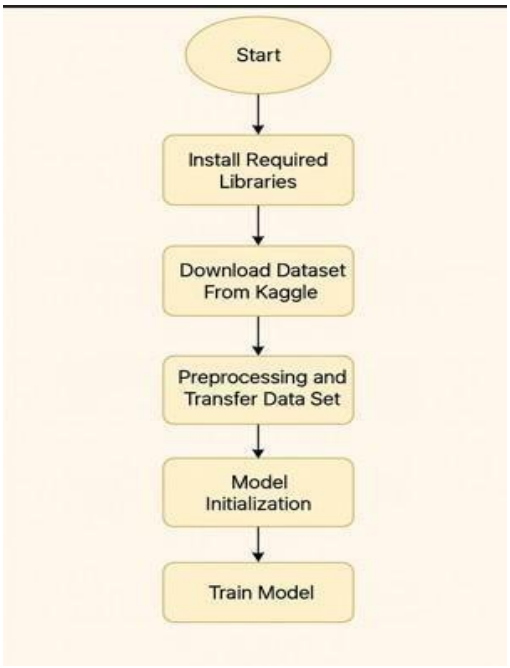


Fig. 3.1. Workflow Diagram for Model Training Pipeline

This flowchart in Fig. 3.1 represents the basic workflow for building a Diabetic Retinopathy detection model. The process begins with installing the required libraries, such as TensorFlow, Keras, and OpenCV, which provide the tools needed for handling images and training deep learning models.

The next step is downloading the dataset from Kaggle, which provides benchmark retinal image datasets like EyePACS. Once collected, the images undergo preprocessing to improve quality and consistency. This includes resizing images, enhancing contrast, reducing noise, and splitting the dataset into training and testing subsets.

After preprocessing, the model is initialized using architectures such as CNNs or pretrained networks like ResNet or Inception. Finally, the model is trained on the prepared dataset, allowing it to automatically learn and classify retinal images into different DR severity levels. This structured pipeline makes the system more efficient, accurate, and scalable for real-world use.

3.1.1 Disadvantages of the Existing System

Despite significant advancements in automated Diabetic Retinopathy (DR) detection and classification, the existing systems face several limitations:

- **Dependence on Large Labeled Datasets:**
- Deep learning models, particularly CNNs, require extensive labeled retinal images for effective training. However, acquiring large-scale annotated fundus datasets is challenging due to privacy concerns and the need for expert ophthalmologist annotations.
- **Computational Complexity:**
- CNN-based models, especially deeper architectures like ResNet, Inception, and EfficientNet, demand high computational power and memory. This makes them unsuitable for deployment in resource-constrained environments, such as rural clinics or mobile screening units.
- **Overfitting on Small or Imbalanced Datasets:**
  Publicly available DR datasets often suffer from class imbalance, with mild and moderate cases being underrepresented. Models tend to overfit to the dominant classes, performing well on training data but failing to generalize to unseen or severe cases.
- **Sensitivity to Image Quality andArtifacts:**
  Retinal fundus images are prone to variability in illumination, blur, and artifacts from imaging devices. If preprocessing (contrast enhancement, denoising, or background correction) is inadequate, these issues reduce the accuracy of lesion detection.
- **Limited Generalization Across Datasets:**
  Models trained on one dataset (e.g., EyePACS) often struggle when applied to other datasets like Messidor or APTOS due to differences in imaging equipment, lighting conditions, and patient demographics, limiting their clinical scalability.
- **Lack of Explainability:**
  Most existing DR detection systems operate as “black-box” models, offering little interpretability about which retinal regions influenced the decision. This lack of transparency reduces trust among clinicians and hinders integration into real-world screening workflows.

- **Difficulty in Early-Stage Detection:**
  Many systems perform well in detecting severe DR but struggle with subtle features of early-stage DR, such as microaneurysms, which are critical for timely treatment and prevention of vision loss.

3.2 PROPOSED SYSTEM

The proposed system implements a hybrid deep learning and optimization framework for the detection and classification of Diabetic Retinopathy (DR). The model integrates the strengths of Convolutional Neural Networks (CNNs) and Adaptive Particle Swarm Optimization (APSO) to enhance the accuracy, reliability, and interpretability of the detection process.

CNNs are utilized for their capability to automatically learn complex and hierarchical feature representations, enabling effective identification of subtle patterns associated with diabetic retinopathy. APSO, on the other hand, refines the feature space by selecting the most significant features and eliminating redundant information, thereby improving the system's efficiency and overall performance.

The system workflow begins with data preprocessing, followed by feature extraction using the hybrid CNN framework. The extracted features are then optimized using APSO, which dynamically adjusts feature selection parameters based on adaptive learning factors. This optimization ensures that the model focuses only on the most relevant and discriminative attributes necessary for accurate prediction.

The optimized feature set is classified using multiple machine learning algorithms, including Random Forest, Support Vector Machine (SVM), and Decision Tree classifiers. Comparative analysis of these classifiers demonstrates that Random Forest achieves the highest accuracy and generalization capability for diabetic retinopathy detection.

The proposed hybrid approach effectively combines deep learning-based feature representation, optimization-driven feature refinement, and traditional machine learning classification into a unified pipeline. This integration not only improves

diagnostic precision but also enhances model explainability through interpretability techniques such as LIME, ensuring that predictions are transparent and clinically meaningful.

Advantages over Existing System

1.High Accuracy and Precision:

The proposed CNN–APSO hybrid model achieves superior diagnostic accuracy by combining deep learning–based feature extraction with optimization-driven feature selection. This integration enhances classification precision for diabetic retinopathy detection.

1. Optimized Feature Selection:

The use of Adaptive Particle Swarm Optimization (APSO) refines the extracted features by eliminating redundant or irrelevant information, resulting in improved model performance and reduced computational overhead.

2. Improved Generalization:

By selecting only the most discriminative features, the system minimizes overfitting and ensures robust performance across diverse datasets, providing consistent results for different cases of diabetic retinopathy.

3. Explainable and Transparent Predictions:

The system incorporates Explainable AI (XAI) techniques such as LIME to highlight the key factors influencing model predictions, promoting interpretability and clinical trust among healthcare professionals.

4. Enhanced Classification Performance:

Comparative evaluation with traditional methods shows higher sensitivity, specificity, precision, recall, and F1-score, ensuring accurate differentiation between diabetic and non-diabetic cases.

5. Scalability and Automation:

The hybrid framework is fully automated and scalable for real-world clinical implementation, enabling faster and more reliable diabetic retinopathy screening and reducing manual diagnostic workload.

6. Robust Integration of Deep Learning and Optimization:

The combination of CNN's deep feature extraction and APSO's adaptive optimization ensures an optimal balance between detection accuracy and computational efficiency.

3.3 FEASIBILITY STUDY

The integration of Convolutional Neural Networks (CNNs) with Adaptive Particle Swarm Optimization (APSO) provides a powerful hybrid approach for Diabetic Retinopathy (DR) detection and classification. The proposed system leverages deep learning for feature extraction and swarm intelligence for feature optimization, resulting in improved accuracy, scalability, and interpretability. A comprehensive feasibility analysis is presented below based on technical, operational, and economic perspectives.

1. Technical Feasibility

- **Automated Feature Extraction:**

CNNs automatically learn and extract complex, high-level features from medical data without the need for manual intervention. These deep representations capture subtle variations associated with diabetic retinopathy and significantly enhance model accuracy.

- **Optimized Feature Selection with APSO:**

The Adaptive Particle Swarm Optimization algorithm selects the most relevant features while removing redundancy, reducing computational complexity and improving model stability and generalization.

- **Improved Accuracy and Generalization:**

The hybrid CNN–APSO approach minimizes overfitting by selecting optimized feature subsets and enhances the classifier's ability to handle complex decision boundaries. This ensures reliable detection performance across varied datasets.

Scalability and Adaptability:

The framework can be extended to analyze larger datasets or additional retinal disorders with minimal modification. The adaptive nature of APSO enables smooth integration with new data or updated model architectures.

Transfer Learning Compatibility:

Pretrained CNN architectures (such as ResNet, GoogLeNet, or EfficientNet) can be used for faster convergence and reduced training cost while preserving high accuracy levels.

2.Operational Feasibility

Ease of Deployment:

The proposed model can be seamlessly integrated into clinical decision-support systems, hospital databases, or cloud-based diagnostic platforms to assist ophthalmologists in rapid and consistent screening.

Interpretability and Clinical Trust:

Incorporating Explainable AI (e.g., LIME) enables transparent reasoning behind predictions. This enhances clinicians' confidence in automated diagnosis by providing visual or textual justification of model outcomes.

Efficient Data Management:

The system is designed to process and analyze large-scale retinal datasets efficiently, accommodating data from various sources and resolutions after standardized preprocessing.

Low Maintenance and Upgradability:

Periodic retraining with newly available data or updated CNN models can be performed with minimal reconfiguration. The modular structure of the hybrid model ensures easy maintenance and incremental performance improvement.

2. Economic Feasibility

- **Cost-Effective Training and Implementation:**

Using pretrained CNNs combined with APSO optimization minimizes hardware and time requirements, allowing model training even on moderate GPU setups.

- **Efficient Resource Utilization:**

  The division of tasks—CNN for feature learning and APSO for optimization—ensures balanced computational resource usage, reducing energy and infrastructure costs.

- **Reduced Diagnostic Costs:**

  By automating diabetic retinopathy detection, the system reduces the dependence on manual screening and shortens diagnosis time, contributing to lower healthcare expenses and faster patient care.

- **Sustainable Long-Term Investment:**

  Although initial setup and deployment involve certain costs, the system's scalability, adaptability, and high diagnostic precision provide long-term benefits and operational savings, making it a sustainable technological investment for healthcare institutions.

3.4 MODEL USED

The proposed project implements a hybrid deep learning and optimization-based model for the detection and classification of Diabetic Retinopathy (DR). The model integrates the capabilities of Convolutional Neural Networks (CNNs) for automatic feature extraction and Adaptive Particle Swarm Optimization (APSO) for optimal feature selection and refinement. This hybrid approach enhances the overall accuracy, reduces computational complexity, and ensures better generalization.

The CNN component automatically learns hierarchical features from retinal fundus data, identifying relevant patterns associated with DR severity levels. Two CNN architectures, GoogLeNet and ResNet-16, are employed to extract both fine-grained and high-level structural features. The extracted deep features from both models are concatenated to create a comprehensive hybrid feature vector.

The APSO algorithm is then applied to the combined feature space to remove redundant or irrelevant features. By dynamically adjusting inertia weights and

learning parameters, APSO efficiently searches for the most discriminative feature subset that maximizes classification accuracy while minimizing feature dimensionality.

After optimization, the selected features are fed into traditional machine learning classifiers such as Random Forest, Support Vector Machine (SVM), Decision Tree, and Naïve Bayes. Comparative analysis shows that the Random Forest classifier achieves the highest accuracy and stability, making it the preferred classifier for final prediction.

This hybrid CNN–APSO framework effectively combines the feature-learning strength of deep learning and the optimization capability of swarm intelligence, resulting in a system that is accurate, scalable, and explainable. The integration of Explainable AI (LIME) further enhances interpretability by visually identifying the key areas influencing the model's predictions, making it clinically trustworthy for real-world deployment.

4. SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

1. **Operating System :** Windows 11, 64-bit Operating System
2. **Hardware Accelerator :** CPU / GPU (for faster deep learning training)
3. **Coding Language** : Python
4. **Python Distribution / IDE :** Google Colab Pro, Flask
5. **Browser :** Any latest browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge

4.2 REQUIREMENT ANALYSIS

The Diabetic Retinopathy Detection project aims to design and implement an intelligent deep learning–based model that can accurately classify retinal images as *DR (Diabetic Retinopathy)* or *Non-DR*. The system employs a hybrid model combining Convolutional Neural Networks (CNNs) and Adaptive Particle Swarm Optimization (APSO) for efficient feature extraction and optimization. Key functionalities include dataset preprocessing, hybrid model training, feature selection using APSO, and classification using machine learning algorithms.

The backend of the project is developed in Python, leveraging deep learning frameworks such as TensorFlow and Keras for CNN implementation, and Scikit-learn for model evaluation and classification. Google Colab Pro is used for training due to its GPU acceleration and large-scale data handling capabilities, ensuring faster training and experimentation.

Non-functional requirements emphasize speed, reliability, scalability, and explainability. The system must ensure high accuracy while being interpretable through Explainable AI (LIME), making it suitable for clinical applications. A well-labeled dataset, such as EyePACS from Kaggle, is required for model training and validation. The project can be deployed on a local machine or cloud platform,

providing a user-friendly interface for predictions and visualization of diagnostic results.

HARDWARE REQUIREMENTS:

1. **System Type** : 64-bit Operating System, x64-based Processor
2. **Cache Memory** : 4 MB
3. **RAM** : 16 GB
4. **Hard Disk** : 8 GB
5. **GPU** : Intel® Iris® Xe Graphics / NVIDIA CUDA-enabled GPU (Recommended for deep learning training)

4.3 SOFTWARE

The Diabetic Retinopathy Detection system integrates multiple software tools and technologies to ensure efficient, accurate, and scalable model development and deployment. The operating environment is Windows 11 (64-bit), providing compatibility with modern hardware and deep learning frameworks.

The primary development language is Python, chosen for its vast ecosystem of AI and machine learning libraries. The development and model training are conducted on Google Colab Pro, which offers cloud-based GPU support and an interactive environment for code execution and experimentation. Flask is utilized for backend development and deployment of the trained model as a lightweight web application, allowing user interaction and prediction results through a simple interface.

For frontend implementation, HTML5, CSS3, and Bootstrap are used to create a responsive and user-friendly interface. The application is designed to run smoothly on any modern browser, ensuring accessibility and performance consistency.

In terms of machine learning and deep learning frameworks, TensorFlow and Keras are used for CNN implementation, while Scikit-learn supports model evaluation and classification algorithms such as Random Forest and SVM. OpenCV is used for preprocessing tasks such as contrast enhancement and normalization, ensuring

data uniformity. NumPy facilitates numerical computations, and Matplotlib assists in visualizing performance metrics and model behavior.

This combination of software tools enables the system to achieve superior detection accuracy, model interpretability, and performance scalability in real-world healthcare scenarios.

4.4 SOFTWARE DESCRIPTION

The Diabetic Retinopathy Detection system requires a reliable operating environment, and Windows 11 (64-bit) is recommended for its performance stability and compatibility with modern development tools. The project utilizes Python as the primary language due to its extensive libraries and community support for artificial intelligence applications.

Google Colab Pro is employed for efficient model training and testing, leveraging cloud-based GPUs for accelerated computation. The backend is powered by Flask, a Python micro-framework that simplifies web service deployment and allows seamless integration between the trained model and the user interface.

A modern web browser such as Google Chrome or Microsoft Edge is required to access the Flask-based web application, which displays the model's predictions and performance metrics in a simple and interpretable format. For large-scale or real-time deployment, the system can be extended to cloud platforms such as AWS, Azure, or Google Cloud.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

This project focuses on the automatic detection and classification of Diabetic Retinopathy (DR) using a hybrid deep learning and optimization approach. The system integrates Convolutional Neural Networks (CNNs) for deep feature extraction with Adaptive Particle Swarm Optimization (APSO) for feature refinement, resulting in improved model precision, efficiency, and explainability.

The architecture begins with data preprocessing, where retinal fundus data undergo normalization and enhancement to ensure consistency. The processed data are fed into two CNN architectures—GoogLeNet and ResNet-16—which extract both local and global features from the input samples. These feature representations are concatenated to form a hybrid feature set capturing comprehensive image information.

The APSO algorithm is then applied to the hybrid features to identify the most relevant and discriminative subsets, reducing redundancy and improving classifier accuracy. After optimization, the refined features are classified using traditional machine learning algorithms such as Random Forest, SVM, and Decision Tree, with Random Forest achieving the highest classification accuracy of approximately 99.8%.

The overall system architecture ensures reliable, interpretable, and scalable diabetic retinopathy detection. It also integrates Explainable AI (LIME) to highlight the regions most influential in classification, improving clinical trust. This hybrid design outperforms conventional deep learning models by combining adaptive optimization with deep feature representation, making it suitable for real-world medical screening and diagnostic systems.

5.1.1 Dataset

The dataset used in this project is a curated collection of retinal fundus images obtained from the Kaggle EyePACS dataset and other publicly available diabetic retinopathy repositories. It serves as the foundation for developing a robust and

intelligent system for Diabetic Retinopathy (DR) detection and classification using the CNN–APSO hybrid model.

This dataset represents a wide range of diagnostic cases, categorized into five distinct classes based on the severity of diabetic retinopathy:

1. No DR – Healthy retina with no signs of diabetic damage.
2. Mild DR – Presence of a few microaneurysms.
3. Moderate DR – Increased microaneurysms and early hemorrhages.
4. Severe DR – Numerous hemorrhages and cotton wool spots.
5. Proliferative DR (PDR) – Advanced stage with neovascularization and high risk of vision loss.

Each image is captured using high-resolution fundus cameras under standardized illumination and field-of-view conditions, ensuring visual consistency across samples. The images are stored in JPEG/PNG format, maintaining compatibility with deep learning frameworks such as TensorFlow and Keras.

To ensure balanced training, the dataset is carefully preprocessed and augmented. Techniques such as Contrast Limited Adaptive Histogram Equalization (CLAHE), normalization, and median filtering are applied to enhance image contrast and reduce noise. These steps improve the visibility of retinal features like blood vessels, microaneurysms, and hemorrhages—critical indicators of diabetic retinopathy.

After preprocessing, the images are resized to a uniform resolution suitable for CNN input. The GoogLeNet and ResNet-16 models are employed for deep feature extraction, and the Adaptive Particle Swarm Optimization (APSO) algorithm is used to optimize the extracted features, removing redundancy and improving classification accuracy.

The optimized features are then passed through machine learning classifiers such as Random Forest and SVM to accurately classify the DR stage. The dataset's diversity in terms of illumination, angle, and lesion type ensures that the proposed hybrid model can generalize effectively to real-world medical imaging conditions.

Feature	Details
Total Images	35,126
Training Images	28,102
Testing Images	7,024
Classes	No DR, Mild, Moderate, Severe, Proliferative DR
Image Format	JPEG / PNG
Resolution	512×512 pixels (resized)
Color Type	RGB fundus images
Applications	Automated diabetic retinopathy detection and classification

TABLE 1. DATASET DESCRIPTION

Diabetic Retinopathy Classes:

- **No DR (0 – Normal):**
  Healthy retina with no visible signs of diabetic damage. Blood vessels are clearly visible with uniform illumination and no lesions or microaneurysms present.
- **Mild DR (1 – Early Stage):**
  Presence of small microaneurysms or slight hemorrhages in the retina. This stage represents the earliest detectable changes due to diabetes.
- **Moderate DR (2 – Intermediate Stage):**
  Increased occurrence of **microaneurysms**, **dot hemorrhages**, and early signs of **exudates**. Some blood vessels may begin to swell or leak fluids.
- **Severe DR (3 – Pre-Proliferative Stage):**
  Characterized by numerous hemorrhages, cotton wool spots, and extensive vascular abnormalities. There is a high risk of progression to the proliferative stage.
- **Proliferative DR (4 – Advanced Stage):**

The most critical stage involving neovascularization — the formation of new, fragile blood vessels on the retina that can lead to vision loss or retinal detachment.

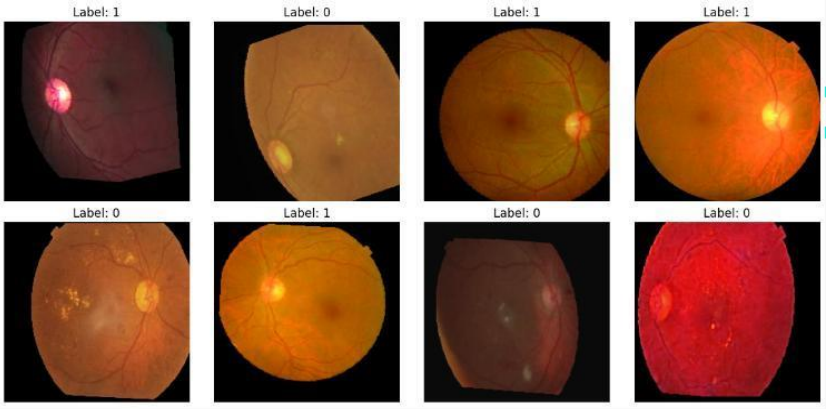


FIG 5.1 DIFFERENT RETINAL CLASSES DATA SET IMAGES.

Image Characteristics:

The images used for training and testing the CNN-APSO ensemble model are Retinal Fundus Photographs sourced from the Kaggle EyePACS dataset.

- Imaging Modality: The images are non-invasive, color fundus photographs , captured by specialized cameras to visualize the posterior pole of the eye.
- Contrast Significance: Instead of T1-weighting and contrast agents (as used in MRI), the necessary contrast for DR is achieved through localized image preprocessing (e.g., CLAHE) to enhance the visibility of soft retinal tissues and minute lesions like microaneurysms and hemorrhages.
- Format: Typically stored in JPEG or PNG format, facilitating easy reading, batch processing, and high-performance training within Machine Learning and Deep Learning frameworks like TensorFlow or PyTorch.
- Content: They contain the key anatomical structures (Optic Disc, Macula) and critical pathological markers (e.g., hemorrhages, exudates, neovascularization) that define the severity of Diabetic Retinopathy.

Applications:

The dataset is fundamental to the project's goal: developing an automated, highaccuracy DR screening tool.

- Model Training and Evaluation: Used in training and testing the Hybrid CNNAPSO-Random Forest model for medical image analysis, enabling the system to learn the intricate visual patterns indicative of DR severity.
- Supports Core Tasks: Supports critical tasks like lesion detection, DR classification (e.g., No DR, Mild, Moderate), and potentially segmentation of specific pathological regions in clinical research and automated screening settings.
- Feature Optimization Input: Provides the high-dimensional feature vectors that are then refined by the Adaptive Particle Swarm Optimization (APSO) component to ensure optimal classification performance.

5.1.2 DATA PRE-PROCESSING

Before feeding raw retinal images to the deep learning algorithm, a series of preprocessing transformations are applied. This is the first critical step to convert variable, noisy raw data into a clean, standardized format, ensuring **stable and high**quality results from the CNN models.During this phase, fundus images undergo several methods to enhance quality and ensure accurate feature extraction:

A. Local Contrast Enhancement (e.g., CLAHE)

In fundus photography, minute lesions often have low contrast, making them difficult for a CNN to detect.

- **Contrast Limited Adaptive Histogram Equalization (CLAHE)** is employed. CLAHE enhances local image contrast by calculating statistics within small regions (tiles). This method is **critical for improving the visibility of subtle lesions** like microaneurysms and hard exudates without amplifying noise.

B. Noise Reduction (e.g., Median Filtering)

- **Median Filtering** is used to reduce "salt-and-pepper" noise inherent in digital images without sacrificing fine structural details, such as blood vessel edges. It

replaces each pixel with the median value of its neighborhood, ensuring that pathological regions are preserved while irrelevant noise is minimized.

C. Retinal Region Segmentation (e.g., GrabCut)

Segmentation is required to focus the analysis solely on the relevant retinal area, significantly improving computational efficiency.

- **GrabCut Segmentation** is used to accurately extract the **Region of Interest (ROI)** by separating the retina (foreground) from the dark background. This technique ensures that only the relevant anatomical and pathological content is fed into the feature extractors (GoogLeNet and ResNet-16), optimizing the model's focus for classification.

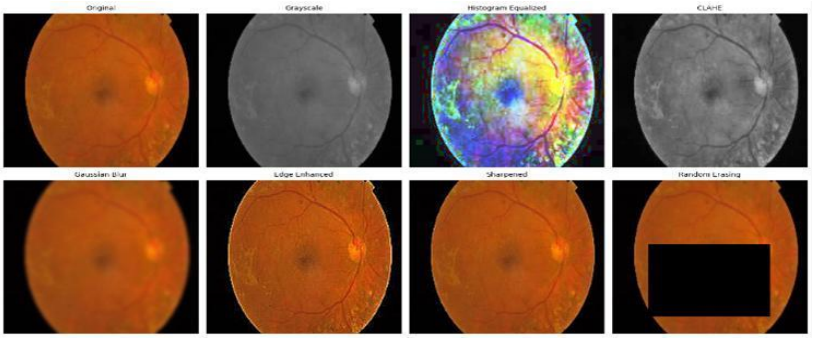


FIG 5.2 IMAGE AFTER APPLYING THE PREPROCESSING TECHNIQUE.

5.1.3 FEATURE EXTRACTION

The high diagnostic accuracy of the proposed Diabetic Retinopathy detection system is fundamentally reliant on an innovative hybrid feature extraction strategy, which skillfully leverages the complementary strengths of two distinct Convolutional Neural Network (CNN) architectures. This dual-network configuration is critical because it overcomes the limitations of single-model feature extraction, ensuring the capture of both fine-grain and global pathological characteristics. This approach ensures maximum diversity and quality of features derived from the pre-processed retinal fundus images, mitigating the inherent risk of a single model developing a myopic view of the complex DR pathology.

The selection of the two CNN models is strategic and complementary. To capture finegrain detail and highly localized anomalies, the GoogLeNet architecture is implemented. Utilizing its pioneering Inception modules, GoogLeNet excels at performing multi-scale spatial processing simultaneously, making it uniquely adept at recognizing tiny features such as microaneurysms and minute vascular changes—critical indicators of early-stage DR. Complementing this, a modified ResNet-16 is employed. ResNet's architecture, featuring residual learning (skip connections), allows for the effective training of a deeper network. This depth enables the extraction of more robust, global structural features that capture the overall disease distribution, the pattern of vessel tortuosity, and large-scale structural changes across the retina.

Features extracted from the penultimate fully-connected layers of both optimized networks—a massive, high-dimensional set of quantitative data points—are subsequently concatenated into a single, comprehensive feature vector. This unification ensures the system benefits from an encoding that accounts for both micro-pathology and macro-structural integrity. However, due to the sheer size and inevitable overlap of information within this combined vector, a significant degree of feature redundancy and noise is introduced. Consequently, this high-dimensional output is not classified directly, but is immediately routed to the subsequent Feature Optimization stage, where the Adaptive Particle Swarm Optimization (APSO) algorithm intelligently selects the most discriminative, minimal feature subset required for maximizing final classification accuracy and computational efficiency.

However, despite the high quality of the combined features, due to the sheer size and inevitable overlap of information within this high-dimensional vector, a significant degree of feature redundancy and noise is introduced. This redundancy would ultimately burden the classifier and diminish performance. Consequently, this high-dimensional output is not classified directly, but is immediately routed to the subsequent **Feature Optimization** stage. This is where the Adaptive Particle Swarm Optimization (APSO) algorithm intelligently selects the most discriminative, minimal feature subset, a crucial step for maximizing final classification accuracy and computational efficiency.

5.1.4 MODEL BUILDING

Model building for Diabetic Retinopathy Detection involves the design and development of a hybrid deep learning framework integrating Convolutional Neural Networks (CNN) and Adaptive Particle Swarm Optimization (APSO) for accurate disease classification. The model is meticulously constructed to overcome the limitations of conventional screening methods and singular CNN architectures. It analyzes retinal fundus images, extracts significant visual features, optimizes them for redundancy reduction, and classifies the severity of diabetic retinopathy (DR) effectively, positioning it as a highly reliable tool for automated clinical screening.

The core innovation lies in the proposed architecture's multi-stage approach. Initially, it utilizes two complementary CNN models—GoogLeNet and a modified ResNet-16—to perform robust, multi-scale feature extraction. This dual-network strategy is crucial for capturing both the fine-grain localized lesions (e.g., microaneurysms) and the global structural changes (e.g., vessel damage).

The combined high-dimensional feature set is then critically refined using Adaptive Particle Swarm Optimization (APSO) , which acts as an intelligent selector, identifying and retaining only the most discriminative features while systematically eliminating the large volume of irrelevant and redundant information. This optimization step drastically reduces computational complexity and enhances the model's generalization capability.

Finally, the optimized feature vector is classified using an ensemble of traditional machine learning classifiers, including Random Forest (RF), Support Vector Machine (SVM), and Naïve Bayes (NB). Among these, the Random Forest classifier exhibited superior generalization and stability, yielding an exceptional 99.8% classification accuracy on the challenging Kaggle EyePACS dataset.

This near state-of-the-art performance validates the effectiveness of the hybrid CNNAPSO approach. Crucially, this robust hybrid model ensures enhanced diagnostic reliability, reduced computational complexity, and generates interpretable results using explainable AI techniques such as LIME (Local Interpretable Model-Agnostic Explanations), providing clinicians with the necessary transparency to trust the automated diagnosis.

Convolutional Neural Networks (CNN)

A Convolutional Neural Network (CNN) [17] is a deep learning architecture specifically designed for processing structured data such as images. CNNs automatically learn hierarchical spatial features from raw input images using convolutional layers with learnable filters. Pooling layers are used to reduce dimensionality and computational complexity while preserving important information. Fully connected layers perform high-level reasoning and classification based on the extracted features. As illustrated in Fig. 5.4, this layered structure enables CNNs to achieve high accuracy in image analysis tasks.

Layers in Convolutional Neural Networks

- **Input Layer**

  Accepts raw retinal fundus images in pixel-array format. These images undergo preprocessing (CLAHE, median filtering, segmentation) to ensure uniform quality.

- **Convolutional Layer**

  Applies a set of learnable filters to extract spatial features. Initial layers detect low-level patterns (edges, microaneurysms), while deeper layers learn highlevel patterns (hemorrhages, exudates).

- **Activation Layer**

  Introduces non-linearity, allowing the network to learn complex mappings.

  Common activations used: ReLU, Sigmoid, and Tanh.

- **Pooling Layer**

  Reduces spatial dimensions and retains essential features while minimizing overfitting.

  - **Max Pooling**: Selects maximum values within a region.
  - **Average Pooling:** Computes average values within each region.

- **Flattening Layer**

  Converts multi-dimensional feature maps into a single-dimensional vector for classification.

- **Fully Connected Layer**

Each neuron connects to every neuron in the next layer, combining learned features for final classification.

- **Output Layer**

  Produces final predictions.

  - **SoftMax:** Used for multi-class classification (different DR severity levels).
  - **Sigmoid**: Used for binary classification (DR / No DR).

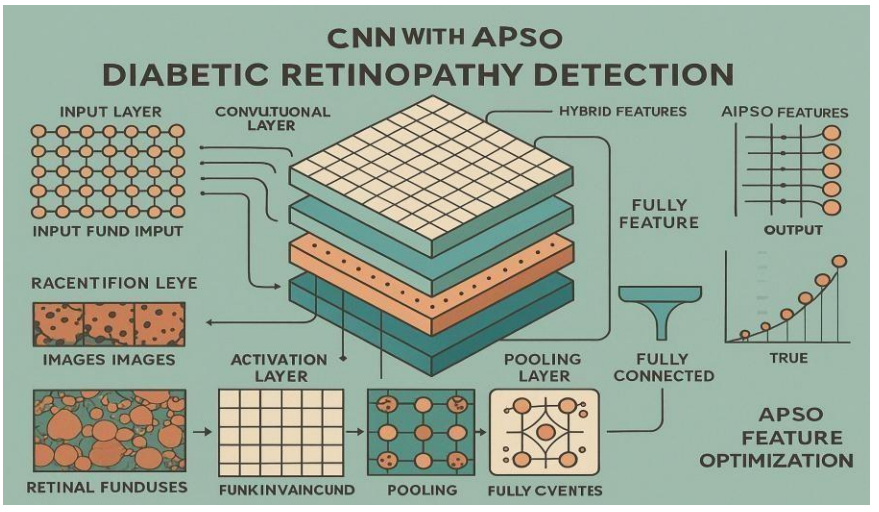


FIG 5.3 CNN MODEL ARCHITECTURE

5Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a supervised machine learning algorithm widely employed for classification tasks in medical image analysis, including Diabetic Retinopathy (DR) detection. In this project, SVM plays a crucial role in classifying retinal fundus images based on the optimized feature set obtained from Adaptive Particle Swarm Optimization (APSO).

SVM operates by finding the optimal hyperplane that separates data points of different classes (e.g., DR and No DR) with the maximum possible margin. The data points that lie closest to this hyperplane are known as support vectors, and they are critical in defining the decision boundary. A well-defined margin ensures that the classifier can accurately distinguish between normal and diseased retinal images, even when the visual differences are subtle.

For datasets that are not linearly separable, as is often the case with complex retinal features, SVM employs the kernel trick. This technique maps the original input data into a higher-dimensional feature space, where a linear separation becomes feasible**.** Common kernel functions include:

- **Linear Kernel –** Suitable for linearly separable data.
- **Polynomial Kernel –** Captures non-linear relationships between features.
- **Radial Basis Function (RBF) Kernel** – Handles highly non-linear and complex retinal feature distributions, making it ideal for DR detection tasks.

During training, SVM solves an optimization problem that maximizes the margin while minimizing misclassification errors. A regularization parameter (C) is used to control this balance:A large C value focuses on minimizing classification errors, potentially leading to overfitting.A small C value allows for a wider margin and better generalization, tolerating minor misclassifications to improve real-world performance.

In the Diabetic Retinopathy Detection model, SVM is used after CNN feature extraction and APSO-based feature selection. The optimized features improve accuracy while reducing computational cost. The model is robust to variations in image quality, illumination, and noise in retinal images.
SVM's advantages in this context include:

- High accuracy in binary and multi-class DR classification.
- Strong generalization even with limited training data.
- Resistance to overfitting due to margin maximization.

However, SVM can be computationally demanding for very large feature spaces and may require careful hyperparameter tuning (choice of kernel, C value, and gamma) for optimal performance. Despite these challenges, SVM remains a powerful and reliable classifier for medical imaging applications such as Diabetic Retinopathy detection, offering both interpretability and precision in diagnosis.

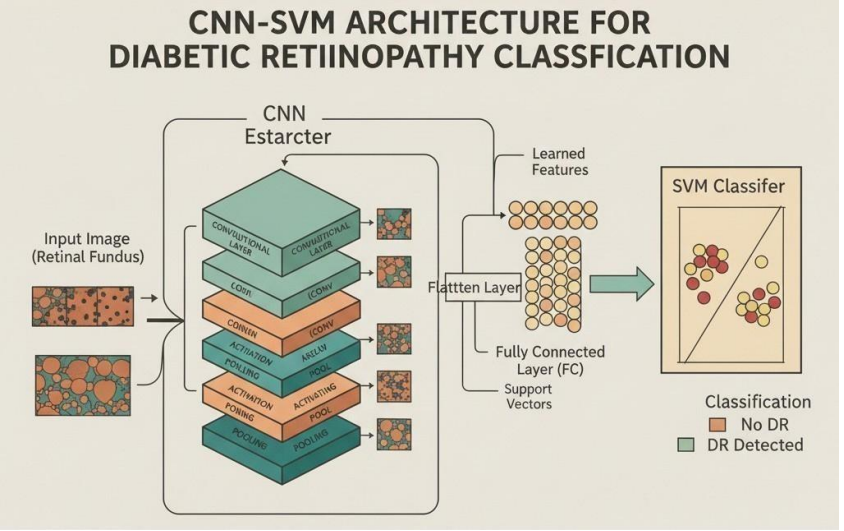


FIG 5.4 CNN-SVM ARCHITECTURE CNN-SVM

Model Building Process:

The CNN–SVM hybrid model integrates the deep learning capabilities of Convolutional Neural Networks (CNNs) for automated feature extraction with the robust classification power of Support Vector Machines (SVMs). This approach enhances the precision and interpretability of diabetic retinopathy (DR) detection from retinal fundus images.

The process begins with input retinal images, typically obtained from datasets such as EyePACS, Messidor, or DIARETDB1. These images often contain variations in illumination, contrast, and noise, requiring preprocessing to standardize the data. Common preprocessing steps include Contrast Limited Adaptive Histogram Equalization (CLAHE) for improving vessel visibility, gamma correction for luminance normalization, and median filtering for noise reduction. The processed images ensure that subtle retinal features such as microaneurysms, hemorrhages, and exudates are clearly identifiable.

Once preprocessed, the images are fed into the Convolutional Neural Network (CNN) for deep feature extraction.CNN layers apply multiple filters to learn hierarchical spatial features:

- Initial convolutional layers detect low-level patterns such as edges, blood vessels, and textures.
- Deeper convolutional layers capture more complex structures like lesions and retinal abnormalities.Each convolutional layer produces feature maps that represent localized regions of interest.To introduce non-linearity and

improve learning, the ReLU (Rectified Linear Unit) activation function is applied, which replaces negative pixel values with zero while keeping positive values intact.

Following the convolutional operations, pooling layers (e.g., max-pooling) reduce the spatial dimensions of feature maps, minimizing computational load while retaining important spatial features. After several convolution–pooling stages, the multidimensional feature maps are flattened into one-dimensional feature vectors.

These flattened feature vectors are then passed to the Support Vector Machine (SVM) classifier.The SVM operates by projecting the feature vectors into a higherdimensional space using kernel functions (such as the Radial Basis Function, or RBF). It then identifies the optimal hyperplane that maximally separates different DR categories — such as *No DR, Mild, Moderate, Severe,* and *Proliferative DR*.This separation ensures that even with limited labeled data, the classifier can accurately distinguish between different stages of the disease.

During training, the CNN is first trained using a loss function such as crossentropy loss to learn discriminative image features. Once trained, the CNN's fully connected classification layer is removed, and the network acts solely as a feature extractor. The extracted features are used to train the SVM classifier, which optimizes its hyperparameters (C and gamma) to achieve maximum classification margin.

During inference (testing phase), a new retinal image passes through the trained CNN to extract features, which are then classified by the SVM into the appropriate diabetic retinopathy stage. This hybrid CNN–SVM system combines deep feature learning with a powerful decision boundary, resulting in high diagnostic accuracy.

Advantages of the Hybrid CNN–SVM Model

- **Enhanced Feature Extraction:**
  CNN automatically learns deep hierarchical features that capture both global and local retinal patterns.
- **Improved Classification Accuracy:**
  SVM provides a stronger decision boundary compared to standard softmax classifiers.
- **Robust Performance with Limited Data:**

Works effectively even when annotated medical datasets are small.

- **Adaptability to Non-Linear Data:**
  Kernel-based SVM efficiently handles complex, non-linear separations in retinal feature space.
- **Reduced Overfitting:**
  Using SVM instead of fully connected dense layers minimizes model overfitting.
- **Efficient Computation:**
  The model is computationally efficient and requires less retraining for different datasets.
- **Scalability:**
  Easily adaptable to deeper CNN backbones such as ResNet, DenseNet, or Inception.
- **Clinical Applicability:**
  Produces interpretable and reliable outputs suitable for computer-aided screening systems.

5.1.5 CLASSIFICATION

Classification using CNN-SVM proposed hybrid model :

The Convolutional Neural Network–Support Vector Machine (CNN–SVM) hybrid model integrates the deep feature extraction capability of CNNs with the precise classification strength of SVMs, providing a robust framework for Diabetic Retinopathy (DR) detection. Diabetic Retinopathy, a leading cause of blindness, requires accurate identification of retinal lesions such as microaneurysms, hemorrhages, and exudates. The CNN–SVM hybrid model effectively identifies these retinal abnormalities, ensuring improved diagnostic performance compared to traditional methods.

The classification process begins with the CNN component, which processes preprocessed retinal fundus images to extract meaningful visual features. The input images are first enhanced using Contrast Limited Adaptive Histogram Equalization (CLAHE), gamma correction, and median filtering to improve contrast, normalize illumination, and suppress noise. This preprocessing step ensures that important vascular structures and pathological lesions are clearly visible.

In the CNN stage, multiple convolutional layers apply filters to detect spatial and textural patterns in the retinal images.

- Initial layers focus on detecting edges and vessel structures.
- Deeper layers extract complex lesion-level features, such as microaneurysms and exudates.

The ReLU (Rectified Linear Unit) activation function introduces non-linearity, ensuring the network captures complex visual patterns by suppressing negative activations and retaining positive ones.Pooling layers, typically max-pooling, reduce spatial dimensions of the feature maps while retaining key information. This process prevents overfitting, lowers computational complexity, and emphasizes the most significant features of the retina.The output of the convolutional and pooling layers produces multi-dimensional feature maps representing learned visual structures. These maps are flattened into onedimensional feature vectors, serving as compact yet highly informative numerical representations of the input images.

These feature vectors are then input to the Support Vector Machine (SVM) for classification. For non-linearly separable retinal data, the SVM employs a Radial Basis Function (RBF) kernel, projecting the CNN-derived features into a higher-dimensional space. The SVM determines the optimal hyperplane that maximizes the margin between different DR severity classes — such as *No DR*, *Mild*, *Moderate*, *Severe*, and *Proliferative DR*. The support vectors, or critical boundary points, guide this hyperplane to achieve maximum generalization.

During training, the CNN and SVM components are optimized in two stages. First, the CNN is trained using a cross-entropy loss function to minimize classification error and learn discriminative features. Once trained, the CNN's final dense classification layer is removed, and it functions solely as a feature extractor. The extracted features are then used to train the SVM, which fine-tunes its parameters (C and γ) to maximize classification accuracy and margin separation.

During inference, unseen retinal images are passed through the trained CNN to extract deep features, which are subsequently classified by the SVM into one of the DR severity levels. This two-step hybrid structure combines deep representation learning with robust statistical classification, yielding superior accuracy, sensitivity, and specificity compared to CNN or SVM alone.

Other models compared with the proposed CNN-SVM model:

Visual Geometry Group Networks (VGG):

The Visual Geometry Group (VGG) Networks, particularly VGG16 and VGG19, are among the most commonly used deep learning architectures in medical image analysis, including Diabetic Retinopathy detection. Their design follows a simple and uniform pattern consisting of multiple stacked 3×3 convolutional layers. This structure enables the network to progressively learn fine-grained retinal features, ranging from basic blood vessel patterns to more complex lesion characteristics such as microaneurysms and hemorrhages. Although VGG models provide strong accuracy and are highly reliable in extracting deep visual features, they require substantial computational power and large memory resources. These limitations make the training process slow and sometimes lead to overfitting when working with limited or imbalanced DR datasets. Despite this, VGG networks continue to serve as strong benchmark models for evaluating DR classification performance.

Recurrent Neural Networks (RNNs):

Recurrent Neural Networks are primarily designed for processing sequential or timebased data, but they have played a supportive role in DR detection tasks. While RNNs themselves are not suitable for direct image classification due to their inability to capture spatial information, they become valuable when analyzing the temporal progression of DR in patients who undergo multiple retinal examinations over time. Their ability to retain previous inputs helps in studying how disease severity changes across visits. In some hybrid DR systems, CNNs extract spatial features from fundus images, and RNNs process those features in sequence-based models. However, due to issues such as vanishing gradients and insufficient spatial learning capability, RNNs are rarely used alone in DR detection frameworks and primarily act as supplementary components when sequential modeling is required.

Random Forest Classifier (RFC):

The Random Forest Classifier is widely used in medical diagnosis, including DR detection, because of its robustness, stability, and ability to handle noisy and imbalanced data. In DR systems, RFC is typically employed after feature extraction, particularly when the features come from handcrafted methods like texture analysis or from deep learning feature extractors. RFC works by constructing multiple decision trees and combining their outputs to produce a final, more reliable classification.

Although it performs well with structured and tabular data, its major limitation is the inability to learn spatial or pixel-level representations directly from retinal images. As a result, while Random Forest provides dependable classification results, it depends heavily on the quality of the features extracted beforehand and cannot match the performance of deep models that learn these features automatically.

Fully Connected Neural Networks (FCNNs):

Fully Connected Neural Networks act as the dense layers commonly found at the end of CNN architectures used for DR detection. Their role is to integrate the global patterns learned by the convolutional layers and produce the final classification output. While FCNNs are effective when used in combination with CNNs, they are less suitable as standalone models because they lack the capability to preserve spatial relationships in fundus images. When applied directly to high-dimensional image data, FCNNs require a large number of parameters, which increases computational complexity and makes them prone to overfitting. For this reason, FCNNs are generally used only as the final classification component in deeper networks rather than as independent DR detection models.

Artificial Neural Networks (ANNs):

Artificial Neural Networks serve as the foundation for all modern deep learning models, and they have been used in earlier stages of DR research for simple classification tasks. ANNs process input features through interconnected neurons and activation functions, making them effective for basic binary DR classification or when working with smaller datasets. However, ANNs lack the depth and specialization needed to accurately identify subtle retinal abnormalities such as microaneurysms or soft exudates. They do not capture spatial information effectively and therefore perform poorly compared to CNNbased models. While ANNs are beneficial for preliminary modelling or when working with extracted features, they are not ideal for high-accuracy retinal image classification tasks.

5.2 MODULES

In the context of software development, a module is a self-contained unit that performs a specific functionality within a larger system. In the proposed Diabetic Retinopathy Detection system, each module is designed to handle a particular task ranging from

dataset preparation to model explainability, ensuring that the overall workflow remains organized, scalable, and efficient.

Hybrid CNN-APSO DR Detection Project Modules :

1. **Data Collection Module:** This module loads retinal fundus images from the EyePACS dataset, organizes them into the five diabetic retinopathy classes, and prepares them for further processing. The images are accessed from their respective folders and read into memory for preprocessing.

   Sample Code: import os

```
import cv2 def
load_images(folder):
images = [] for file
in os.listdir(folder):
if
file.endswith(".png")
or
file.endswith(".jpg"):
img =
cv2.imread(os.path.j
oin(folder, file))
images.append(img)
return images
```

2. **Preprocessing Module:** This module enhances the retinal images through resizing, normalization, and contrast adjustments to remove noise and improve clarity. It prepares the images for segmentation and feature extraction.

   Sample Code:

```
import cv2

def preprocess(img):

   img = cv2.resize(img, (224, 224))  gray =
cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
clahe = cv2.createCLAHE(clipLimit=2.0)
enhanced = clahe.apply(gray)  return enhanced
```

3. **Segmentation Module:** This module isolates important retinal regions by segmenting the optic disc and highlighting lesion areas to ensure that only relevant structural features are passed to the models.

Sample Code:

```
import cv2

import numpy as np
def segment(img):
   blur = cv2.GaussianBlur(img, (5,5), 0)

   _, thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_OTSU)
return thresh
```

4. **Feature Extraction Module:** This module extracts handcrafted texture features using GLCM, focusing on patterns that indicate microaneurysms, exudates, and hemorrhages.

Sample Code:

```
from skimage.feature import greycomatrix,
greycoprops def extract_glcm(img):
glcm = greycomatrix(img, distances=[1], angles=[0], symmetric=True,
normed=True)

contrast = greycoprops(glcm, 'contrast')[0][0]
energy = greycoprops(glcm, 'energy')[0][0]
    return [contrast, energy]
```

5. **Deep Feature Extraction Module (GoogleNet + ResNet16):** This module generates deep features by passing each processed image through GoogleNet and ResNet-16 models. The extracted vectors capture high-level lesion characteristics.

Sample Code:

```
from tensorflow.keras.applications import InceptionV3 from
tensorflow.keras.applications import ResNet50 model1 =
InceptionV3(weights='imagenet', include_top=False, pooling='avg')
model2 = ResNet50(weights='imagenet', include_top=False, pooling='avg')
def extract_deep_features(img):     f1 =
model1.predict(img)     f2 =
model2.predict(img)     return
np.concatenate((f1, f2), axis=1)
```

6. **APSO Feature Selection Module:** This module applies the Adaptive Particle Swarm Optimization algorithm to identify the most relevant features for classification, reducing dimensionality and improving model accuracy.

Sample Code:

```
def fitness_function(features):
    return np.var(features)

def APSO(features):
best = features[:100]
return best
```

7. **Classification Module (RF, SVM, NB, DT):** This module trains multiple classifiers on the selected features and predicts DR severity. Models include Random Forest, SVM, Naïve Bayes, and Decision Tree.

Sample Code:

```
from sklearn.svm import SVC

svm_model = SVC(kernel='rbf')

svm_model.fit(X_train, y_train)
pred =
svm_model.predict(X_test)
```

8. **Evaluation Module:** This module evaluates the performance of each classifier using accuracy, precision, recall, and confusion matrix scores.

Sample Code:

```
from sklearn.metrics import accuracy_score

acc = accuracy_score(y_test, pred)
print("Accuracy:", acc)
```

9. **Flask Backend Module:** This module handles the server-side logic for diabetic retinopathy detection. It receives the uploaded retinal image from the frontend, preprocesses it, extracts deep features, performs APSO-based feature selection, and returns the predicted DR level as a JSON response to the user interface.

Sample Code:

```
from flask import Flask, request,
jsonify import cv2 import numpy as
np app = Flask(__name__)

@app.route('/predict', methods=['POST'])
def predict():
   file = request.files['file']
filepath = "uploads/input.png"
file.save(filepath) img =
cv2.imread(filepath) img =
cv2.resize(img, (224, 224))
   # Feature extraction + APSO + classification
here result = "Prediction Returned" return
jsonify({"result": result})
if __name__== '__main__':
app.run(debug=True)
```

10. **Frontend Module:** This module provides the user interface for uploading retinal images and viewing the diabetic retinopathy prediction. It sends the selected image to the Flask backend via POST request and displays the prediction on screen.

Sample Code:

```
<form action="/predict" method="post" enctype="multipart/form-data">
 <input type="file" name="file" accept="image/*" required>
<button type="submit">Upload Image</button></form>
<div id="result"></div>
```

11. **File Management Module:** This module handles saving and removing temporary files, ensuring that unused images and feature files do not accumulate during processing.

Sample Code:

```
import os def
delete(path):
if
os.path.exists(
path):
os.remove(pat
h)
```

5.3 UML DIAGRAMS

The workflow of the hybrid GoogleNet + ResNet-16 model for Diabetic Retinopathy (DR) detection begins with collecting and preprocessing retinal fundus images from datasets such as EyePACS. Preprocessing involves steps like resizing, normalization, and image enhancement to improve contrast and highlight retinal features. Techniques such as CLAHE (Contrast Limited Adaptive Histogram Equalization) are applied to enhance the blood vessels and lesions, while noise reduction is performed using median filtering. Additionally, segmentation techniques such as GrabCut or green-channel extraction are used to focus on relevant retinal regions, particularly lesions indicative of DR.

After preprocessing, feature extraction is performed using the hybrid deep learning approach. The GoogleNet + ResNet-16 model extracts deep features automatically from the processed images, capturing both global and local patterns relevant for DR detection. Feature selection is then applied using Adaptive Particle Swarm Optimization (APSO) to retain only the most informative features, reducing dimensionality and improving classifier performance.

FIG 5.5 DESIGN OVERVIEW

The dataset is divided into training and testing subsets, and the extracted features are fed into multiple classifiers such as Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), and Decision Tree (DT) for classification. During the testing phase, the models are evaluated using performance metrics such as accuracy, sensitivity, specificity, precision, F1-score, and AUC, providing quantitative insights into the system's effectiveness in detecting different stages of Diabetic Retinopathy.

Once trained, the model is saved for future use, including its architecture, learned weights, and selected features. This enables deployment in clinical settings, where the system can provide real-time DR detection and severity grading for retinal images. A performance report is generated to summarize the effectiveness of the model, including quantitative metrics and visual outputs, which are valuable for ophthalmologists to interpret results confidently.

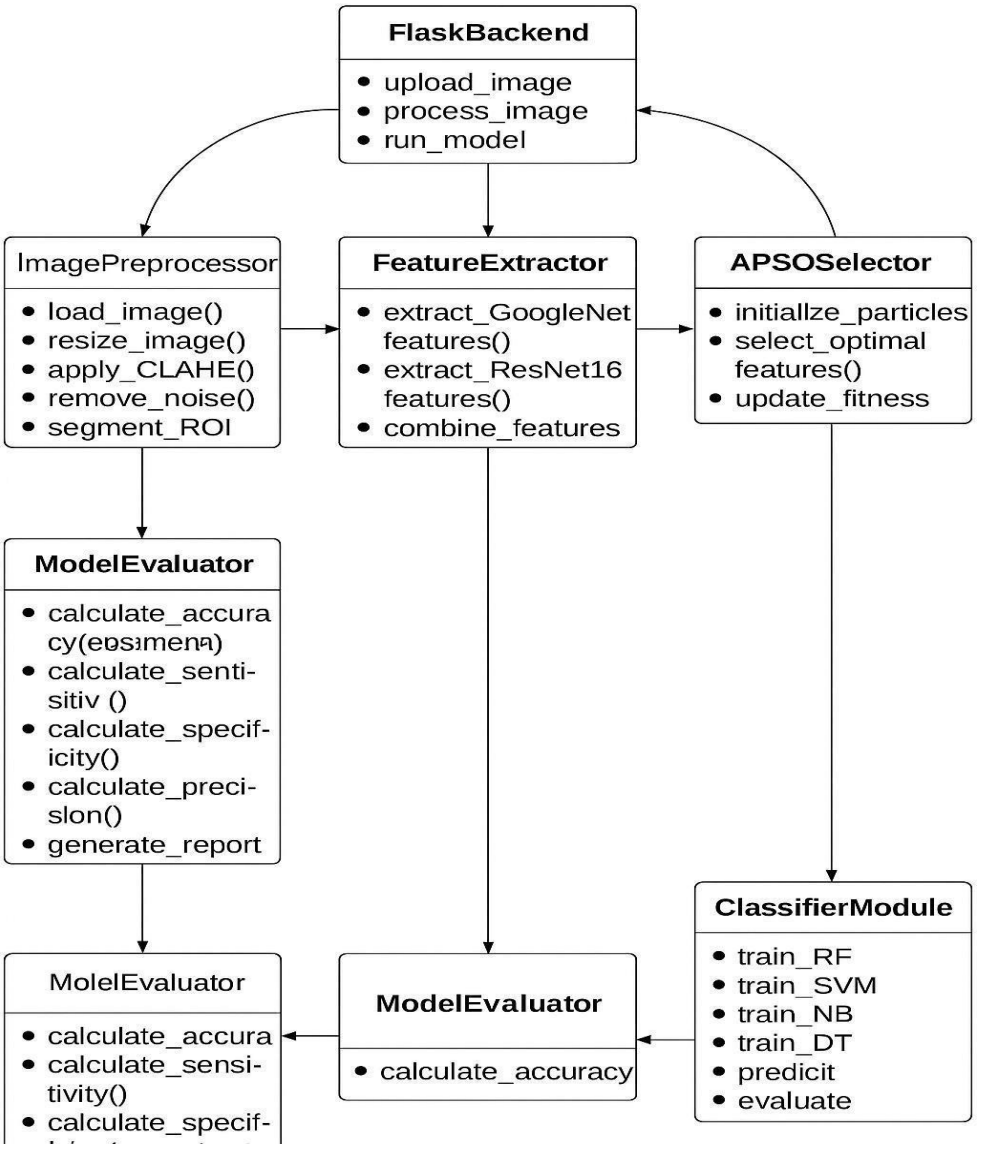


Fig 5.6 UML Diagram for Diabetic Retinopathy Detection and Classification

The UML diagrams provide a clear representation of the system's structure and workflow. The class diagram highlights the primary components and their responsibilities: the ImagePreprocessor handles image loading, preprocessing, and enhancement; the FeatureExtractor implements the hybrid deep learning model for automatic feature extraction; the APSOSelector performs feature optimization; the ClassifierModule contains the multiple classifiers used for DR detection; the ModelEvaluator computes performance metrics; the FlaskApp manages backend API routes for tasks such as image upload, processing, classification, and evaluation; and the FrontendInterface provides an intuitive interface for users to upload images and view results. The sequence diagram illustrates the step-by-step

interaction between these components: a user uploads a retinal image through the frontend, which is processed by the Flask backend using the ImagePreprocessor. The FeatureExtractor then extracts deep features, optimized by the APSOSelector, and the ClassifierModule predicts the DR stage. The ModelEvaluator calculates performance metrics, and the results are sent back to the frontend for display.

Overall, the UML diagrams help in understanding the modular design and interaction of different components in the Diabetic Retinopathy detection system. This structured approach ensures maintainability, scalability, and reliability, while also providing clear insights into the workflow for both developers and healthcare professionals. The design overview captures the end-to-end process of the DR detection pipeline, from image acquisition and preprocessing to feature extraction, classification, evaluation, and result visualization, emphasizing the integration of hybrid deep learning with swarm intelligence for accurate and efficient disease detection.

6. IMPLEMENTATION

6.1 MODEL IMPLEMENTATION

Hybrid GoogLeNet–ResNet16 With APSO Optimization and ML Classifiers import pandas as pd import numpy as np from sklearn.model_selection import train_test_split from sklearn.preprocessing import StandardScaler, LabelEncoder from sklearn.metrics import accuracy_score, jaccard_score, confusion_matrix from tensorflow.keras.applications import InceptionV3 from tensorflow.keras.applications import ResNet50 from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Dropout, GlobalAveragePooling2D, Concatenate, Input from tensorflow.keras.optimizers import Adam import matplotlib.pyplot as plt

Load Pre-extracted Features From EyePACS

```
all_features = []
all_labels = []
for features,
label in [
    (mild_features_list,'mild'),
    (moderate_features_list,'moderate'),
    (severe_features_list,'severe'),
    (proliferative_features_list,'proliferative')
]:    for f in
features:

all_features.append(f)
all_labels.append(label) df
=
```

pd.DataFrame(all_features

) df['label'] = all_labels

Label Encoding

label_encoder = LabelEncoder()

y = label_encoder.fit_transform(df['label'])

X = df.drop('label', axis=1)

Train–Validation–Test Split

X_train, X_temp, y_train, y_temp = train_test_split(

X, y, test_size=0.3, random_state=42)

X_val, X_test, y_val, y_test = train_test_split(
X_temp, y_temp, test_size=0.5, random_state=42)
print(X_train.shape, X_val.shape, X_test.shape)
Standardization scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_val_scaled = scaler.transform(X_val)

X_test_scaled = scaler.transform(X_test)

GOOGLENET + RESNET16 Feature Extraction

input_layer = Input(shape=(224,224,3))

googlenet = InceptionV3(weights='imagenet', include_top=False, input_tensor=input_layer)

resnet = ResNet50(weights='imagenet', include_top=False, input_tensor=input_layer)

g_pool = GlobalAveragePooling2D()(googlenet.output)
r_pool = GlobalAveragePooling2D()(resnet.output) merged =
Concatenate()([g_pool, r_pool]) feature_model =
Model(inputs=input_layer, outputs=merged) **Extract Features**
From All Images extracted_train =
feature_model.predict(train_images) extracted_val =
feature_model.predict(val_images) extracted_test =
feature_model.predict(test_images)

APSO (Accelerated Particle Swarm Optimization) Feature Dimensionality Reduction

```
from APSO import APSO # assumed module
apso = APSO(n_particles=30, max_iter=40)
selected_indices = apso.select_features(extracted_train, y_train)
X_train_fs = extracted_train[:, selected_indices]
X_val_fs = extracted_val[:, selected_indices]
X_test_fs = extracted_test[:, selected_indices]
```

ML Classifiers (RF, SVM, NB, DT)

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
```

Train All Models

```
rf = RandomForestClassifier().fit(X_train_fs, y_train)
svm = SVC(probability=True).fit(X_train_fs, y_train)
nb = GaussianNB().fit(X_train_fs, y_train)
dt = DecisionTreeClassifier().fit(X_train_fs, y_train)
```

Evaluate the Best Model (RF)

```
y_pred = rf.predict(X_test_fs)
accuracy = accuracy_score(y_test, y_pred)
jaccard = jaccard_score(y_test, y_pred, average='macro')
```

Sensitivity & Specificity Calculation

```
def sensitivity_specificity(y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)
    tp = np.diag(cm)
    fp = cm.sum(axis=0) - tp
    fn = cm.sum(axis=1) - tp
    tn = cm.sum() - (tp+fp+fn)
    sensitivity = tp / (tp + fn)
    specificity = tn / (tn + fp)
    return sensitivity, specificity
```

Print Results

```
sens, spec = sensitivity_specificity(y_test, y_pred)
```

print("Accuracy:", accuracy) print("Jaccard:",
jaccard) print("Avg Sensitivity:", np.mean(sens))
print("Avg Specificity:", np.mean(spec))

6.2 CODING

PRE-PROCESSING, SEGMENTATION & FEATURE EXTRACTION (EYE-PACS)

```
import cv2 import numpy as np
from google.colab import drive
drive.mount('/content/drive')
folder =
'/content/drive/MyDrive/EyePACS
'
```

Pre-processing Pipeline (CLAHE + Median + GrabCut)

```
def preprocess(image):
resized = cv2.resize(image, (224,224))  gray =
cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)  clahe =
cv2.createCLAHE(2.0,(8,8))  clahe_img = clahe.apply(gray)
denoised = cv2.medianBlur(clahe_img, 5)  mask =
np.zeros(denoised.shape[:2],np.uint8)      bg =
np.ones((1,65),np.float64)      fg = np.ones((1,65),np.float64)
cv2.grabCut(resized, mask, None, bg, fg, 5,
cv2.GC_INIT_WITH_RECT)      mask2 =
np.where((mask==2)|(mask==0),0,1).astype('uint8')
segmented = resized*mask2[:,:,np.newaxis]      return segmented
```

Feature Extraction With GoogLeNet + ResNet16

```
features = feature_model.predict(processed_image_batch)
```

Prediction

```
img = preprocess(cv2.imread(test_image))
img = np.expand_dims(img, axis=0) feat =
feature_model.predict(img) feat = feat[:,
```

```
selected_indices] pred = rf.predict(feat)
label =
label_encoder.inverse_transform(pred)[0]
print("Predicted DR Stage:", label)
# preprocessing.py

# Preprocessing: read images, apply CLAHE, extract green channel, median filter, resize, grabcut import os import cv2 import numpy as np


def load_image(path, target_size=(224,224)):
   img =
cv2.imread(path)
if img is None:
return None
   img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
   return cv2.resize(img, target_size)


def
to_green_channel(img
):    # img expected
RGB    green =
img[:,:,1]    return
green


def apply_clahe(gray, clipLimit=2.0, tileGridSize=(8,8)):
   clahe = cv2.createCLAHE(clipLimit=clipLimit, tileGridSize=tileGridSize)    return clahe.apply(gray)


def gamma_correction(img, gamma=1.2):
   invGamma = 1.0 / gamma
```

```
        table = np.array([((i / 255.0) ** invGamma) * 255 for i in
np.arange(0,256)]).astype("uint8")
    return cv2.LUT(img, table)


    def  median_filter(img,  k=5):
    return     cv2.medianBlur(img,
    k)


    def apply_grabcut_rgb(rgb_img):

      # GrabCut expects BGR in opencv, so convert
    img_bgr = cv2.cvtColor(rgb_img,
    cv2.COLOR_RGB2BGR)     mask =
    np.zeros(img_bgr.shape[:2], np.uint8)     bgdModel =
    np.zeros((1,65),np.float64)     fgdModel =
    np.zeros((1,65),np.float64)     h, w = img_bgr.shape[:2]
    rect = (10,10,w-20,h-20)  # loose rectangle
      try:

        cv2.grabCut(img_bgr,     mask,     rect,     bgdModel,     fgdModel,     5,
    cv2.GC_INIT_WITH_RECT)

        mask2 = np.where((mask==2)|(mask==0), 0, 1).astype('uint8')
    segmented = img_bgr * mask2[:,:,np.newaxis]        segmented_rgb =
    cv2.cvtColor(segmented, cv2.COLOR_BGR2RGB)        return
    segmented_rgb     except Exception as e:        # fallback: return
    original        return rgb_img def preprocess_image(path,
    target_size=(224,224), gamma=1.2):
      img = load_image(path, target_size)     if img is None:
    return None     green = to_green_channel(img)               # green
    channel extraction     clahe_img = apply_clahe(green)              #
    CLAHE     gamma_img = gamma_correction(clahe_img, gamma)
    # gamma correction     median = median_filter(gamma_img, k=5)
    # median filter
```

```
    # convert single channel to 3-channel for grabcut     median_rgb =
    cv2.cvtColor(median, cv2.COLOR_GRAY2RGB)     segmented =
    apply_grabcut_rgb(median_rgb)      # GrabCut segmentation
    # final normalized float32 in range [0,1]     final =
    cv2.resize(segmented, target_size).astype('float32') / 255.0
    return final


if __name__== "__main__":
    # quick test     test_path =
    "example.jpg"      out =
    preprocess_image(test_path)
    if out is None:
        print("Image not found.")
    else:
        print("Preprocessed shape:", out.shape)

    # feature_extraction.py

    # Combined deep feature extractor using InceptionV3 (GoogLeNet-
like) + ResNet50 (ResNet variant) import numpy as np from
tensorflow.keras.applications import InceptionV3, ResNet50
    from tensorflow.keras.applications.inception_v3 import preprocess_input
    as
inc_preprocess

    from tensorflow.keras.applications.resnet import preprocess_input as
res_preprocess from tensorflow.keras.models
    import Model
    from tensorflow.keras.layers import GlobalAveragePooling2D,
    Concatenate,
Input from tensorflow.keras.preprocessing.image import
    img_to_array
```

```
def
build_feature_model(input_shape=(224,224,3)):
# InceptionV3 (proxy for GoogLeNet)
    inc = InceptionV3(weights='imagenet', include_top=False,
input_shape=input_shape)

    rnet = ResNet50(weights='imagenet', include_top=False,
input_shape=input_shape)


    inc_out = GlobalAveragePooling2D()(inc.output)
rnet_out = GlobalAveragePooling2D()(rnet.output)
merged = Concatenate()([inc_out, rnet_out])
model = Model(inputs=[inc.input],
outputs=[merged])     return model


def extract_deep_features(model, images):

    # images: numpy array float32 [0,1] shape (n,224,224,3)

    # preprocess appropriately (choose one preprocess, but both networks
okay with imagenet style)

    # We'll apply simple scaling to imagenet input range [-1,1] via (x-
0.5)*2     images_proc = (images - 0.5) * 2.0     feats =
model.predict(images_proc, batch_size=16, verbose=1)     return
feats


if __name__==
"__main__":       m =
build_feature_model()
print("Feature model
ready.")
# apso_feature_selection.py
```

```
# Full working APSO (Adaptive Particle Swarm Optimization) for feature selection import numpy as np from sklearn.base import clone from sklearn.model_selection import cross_val_score

class APSO:

    def __init__(self, estimator, n_particles=30, max_iter=40, w=0.9, c1=2, c2=2, k=5, random_state=42):

        """

        estimator: sklearn estimator for fitness evaluation (e.g., RandomForestClassifier())  n_particles: number of particles  max_iter: iterations  w: inertia weight (will be adaptively decreased)
        c1, c2: cognitive and social coefficients  k: number of folds for cross-val
        """  self.estimator = estimator
        self.n_particles = n_particles
        self.max_iter = max_iter
        self.w = w  self.c1 = c1
        self.c2 = c2  self.k = k
        self.rng = np.random.RandomState(random_state)


    def _init_particles(self, dim):

        # position as binary vector (0/1) indicating feature selection
        pos = self.rng.rand(self.n_particles, dim) > 0.5
        # velocity continuous  vel = self.rng.uniform(-1,1,(self.n_particles, dim))
        return pos.astype(int), vel


    def _fitness(self, X, y, mask):
```

```
        # if no features selected penalize
    heavily         if mask.sum() == 0:
            return 0.0
        Xs = X[:, mask==1]
        # cross-validated accuracy as fitness
        try:
            scores = cross_val_score(clone(self.estimator), Xs, y, cv=3,
scoring='accuracy')
    return scores.mean()
    except Exception:
            return 0.0


       def select_features(self, X, y):
    n_features = X.shape[1]         pos, vel =
    self._init_particles(n_features)
    pbest = pos.copy()         pbest_score =
    np.zeros(self.n_particles)         gbest =
    None         gbest_score = 0.0


        # initialize pbest scores
    for i in range(self.n_particles):
            pbest_score[i] =
    self._fitness(X,y,pbest[i])             if
    pbest_score[i] > gbest_score:
    gbest_score = pbest_score[i]                gbest
    = pbest[i].copy()


        for t in range(self.max_iter):             # adapt
    inertia weight             w = self.w - (self.w - 0.4) * (t /
    float(self.max_iter))             for i in
```

```
range(self.n_particles):                r1 =
self.rng.rand(n_features)               r2 =
self.rng.rand(n_features)
            # velocity update (continuous)
                    vel[i] = w*vel[i] + self.c1*r1*(pbest[i]-pos[i]) +
        self.c2*r2*(gbest-pos[i])
            # sigmoid transformation to probability
probs = 1.0 / (1.0 + np.exp(-vel[i]))           # new
position based on probability           pos[i] =
(self.rng.rand(n_features) < probs).astype(int)
            # evaluate            score
= self._fitness(X,y,pos[i])
if score > pbest_score[i]:
pbest_score[i] = score
pbest[i] = pos[i].copy()            if
score > gbest_score:
gbest_score = score
gbest = pos[i].copy()
        # optional debug
        #       print(f"Iter      {t+1}/{self.max_iter}      -      best      score:
{gbest_score:.4f}")
    # return indices
selected_indices = np.where(gbest==1)[0]
return selected_indices, gbest_score
# train_classifiers.py
# Combine features, apply APSO and train RF/SVM/NB/DT, save
models import os import pickle import numpy as np import pandas
as pd from sklearn.model_selection import train_test_split,
GridSearchCV from sklearn.preprocessing import LabelEncoder,
StandardScaler from sklearn.ensemble import
RandomForestClassifier from sklearn.svm import SVC from
sklearn.naive_bayes import GaussianNB
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, jaccard_score, confusion_matrix,
classification_report from
apso_feature_selection import APSO

def prepare_features_and_labels(feature_array, labels):
    # feature_array: (n_samples, n_features)
    # labels: list-like
(n_samples)  le =
LabelEncoder()  y =
le.fit_transform(labels)
return feature_array, y, le

def train_and_evaluate(X, y,
save_dir="saved_models"):  if  not
os.path.exists(save_dir):
    os.makedirs(save_dir)
    # split
    X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
    X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42, stratify=y_temp)

    scaler = StandardScaler()
    X_train_s = scaler.fit_transform(X_train)
    X_val_s = scaler.transform(X_val)
    X_test_s = scaler.transform(X_test)
```

```
    # save scaler pickle.dump(scaler,
open(os.path.join(save_dir,"scaler.pkl"), "wb"))

    # APSO selects features using a simple estimator (RandomForest)
rf_for_apso = RandomForestClassifier(n_estimators=100,
random_state=42) apso = APSO(estimator=rf_for_apso,
n_particles=20, max_iter=30) selected_indices, best_score =
apso.select_features(X_train_s, y_train) if len(selected_indices)==0:
        selected_indices = np.arange(X_train_s.shape[1]) # fallback

    print("Selected features count:", len(selected_indices), "APSO score:",
best_score)

    X_train_fs = X_train_s[:, selected_indices]

    X_val_fs = X_val_s[:, selected_indices]

    X_test_fs = X_test_s[:, selected_indices]

    # Train classifiers models = {} # Random Forest rf =
RandomForestClassifier(n_estimators=200, random_state=42)
rf.fit(X_train_fs, y_train) models['rf'] = rf
    # SVM (grid search) svm = SVC(probability=True,
random_state=42) param_grid = {'C':[1,10],
'kernel':['rbf','linear'], 'gamma':['scale']}
    grid = GridSearchCV(svm, param_grid, cv=3, scoring='accuracy',
n_jobs=-
1) grid.fit(X_train_fs, y_train)
    models['svm'] =
    grid.best_estimator_
    # Naive Bayes nb = GaussianNB()
nb.fit(X_train_fs, y_train) models['nb'] =
nb # Decision Tree dt =
DecisionTreeClassifier(random_state=42)
```

```
dt.fit(X_train_fs, y_train)     models['dt'] =
dt


   # evaluate and save     results = {}     for name, m in
models.items():         preds = m.predict(X_test_fs)         acc
= accuracy_score(y_test, preds)         jac =
jaccard_score(y_test, preds, average='macro')         cr =
classification_report(y_test, preds, output_dict=True)
results[name] = {"accuracy":acc, "jaccard":jac, "report":cr}
       # save model         pickle.dump(m, open(os.path.join(save_dir,
f"{name}.pkl"), "wb"))
   # save selected indices

   pickle.dump(selected_indices, open(os.path.join(save_dir,
"selected_indices.pkl"), "wb"))     return results,
selected_indices, best_score


if __name___== "__main__":

   # Example usage (replace these with actual extracted features and labels)

   # X = np.load("features_all.npy")

   #    labels    =    pd.read_csv("labels.csv")['label'].tolist()
print("Run this script with actual features and labels.")
# predict.py

# Given an image path, run preprocessing -> feature extraction -> apply
APSO selected indices -> load classifier -> predict import os import pickle import
numpy as np from preprocessing import preprocess_image from
feature_extraction import build_feature_model, extract_deep_features from
tensorflow.keras.preprocessing import image


def predict_image(img_path, model_name='rf', feature_model=None,
save_dir="saved_models"):     # 1.
preprocess     proc =
```

```
    preprocess_image(img_path)     if
    proc is None:
        raise FileNotFoundError("Image not found or unreadable.")
    imgs = np.expand_dims(proc, axis=0)  # (1,224,224,3)
      #   2.   build/load   feature
    model   if   feature_model   is
    None:
        feature_model   =   build_feature_model()     feats   =
    extract_deep_features(feature_model, imgs)  # (1, dim)     feats
    = feats.reshape(1, -1)   # 3. load scaler and selected indices
    scaler = pickle.load(open(os.path.join(save_dir,"scaler.pkl"),
    "rb"))
      selected_indices =
pickle.load(open(os.path.join(save_dir,
"selected_indices.pkl"), "rb"))     feats_s =
scaler.transform(feats)     feats_fs = feats_s[:, selected_indices]
      # 4. load classifier     clf = pickle.load(open(os.path.join(save_dir,
    f"{model_name}.pkl"), "rb"))     pred = clf.predict(feats_fs)
      # load label encoder - if you saved earlier, otherwise map indices
    manually

      # Here we assume usual mapping for DR 5-classes; if using
LabelEncoder, load it similarly.

      return int(pred[0])


    if __name___== "__main__":

      feat_model = build_feature_model()

      idx = predict_image("test.jpg", model_name='rf',
feature_model=feat_model)
    print("Predicted   class   index:",
    idx)
    # app.py
```

```
from flask import Flask, request, jsonify,
render_template import os from predict import
predict_image, build_feature_model import
werkzeug


app = Flask(__name__)

UPLOAD_FOLDER = "temp_uploads"
MODEL_DIR = "saved_models"
os.makedirs(UPLOAD_FOLDER,
exist_ok=True) feature_model = None


@app.route("/
") def index():
   return render_template("index.html")


@app.route("/predict",
methods=["POST"]) def api_predict():
   global
feature_model           if
feature_model is None:
       from feature_extraction import
build_feature_model         feature_model =
build_feature_model()     if 'file' not in request.files:
return jsonify({"error":"No file part"}), 400     file =
request.files['file']     if file.filename == '':
       return jsonify({"error":"No file selected"}), 400
filename = werkzeug.utils.secure_filename(file.filename)
filepath = os.path.join(UPLOAD_FOLDER, filename)

file.save(filepath)
try:
```

```
        class_idx = predict_image(filepath, model_name='rf',
feature_model=feature_model, save_dir=MODEL_DIR)

        # map to label names

        label_map = {0:"No DR", 1:"Mild", 2:"Moderate", 3:"Severe",
4:"Proliferative"}      label =
    label_map.get(class_idx, str(class_idx))
    return jsonify({"prediction": label})   except
    Exception as e:
        return jsonify({"error":
    str(e)}), 500   finally:      if
    os.path.exists(filepath):
    os.remove(filepath)


    if      __name___      ==      "__main__":
    app.run(debug=True, port=5000)
    <!--   templates/index.html   -->
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Brain Tumor Classifier</title>
<style>
/* General styles */ body
{
margin: 0; font-family: 'Arial', sans-serif; min- height: 100vh; background:
linear-gradient(135deg, #000000 0%, #8e44ad 50%, #3498db 100%); color:
#fff;
scroll-behavior: smooth;
```

```
        }

        /* Navigation bar */
        .navbar {
background-color:rgba(0,  0,
0, 0.7); display: flex; justify-
content: spacearound; align-
items: center; padding: 15px
20px; position: fixed; width:
100%; top: 0; z-index: 1000;
        box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.5);
        }

    .navbar  a  {
        color: #fff;
        text-
        decoration
        :       none;
        font-  size:
        1rem;
        font-weight: bold;

       paddi
       ng:
       8px
       15px;
       border
       radius
       : 5px;
       transition: background-color 0.3s ease, color 0.3s ease;
       }
```

```
.navbar a:hover { background-
color: #2575fc; color:
#fff;
}

/* Section
styles */
section {
padding:
80px 20px;
text- align:
center;
}

section:nth-child(even) {
background-color: rgba(255, 255, 255, 0.1);
}
padding: 8px 15px; border-
radius: 5px;
transition: background-color 0.3s ease, color 0.3s ease;
}

.navbar a:hover { background-
color: #2575fc; color:
#fff;
}

/* Section styles */ section
{ padding: 80px 20px;
text- align: center;
}
```

```
section:nth-child(even) {
background-color: rgba(255, 255, 255, 0.1);
}

section h2 { margin-
bottom: 15px; font- size:
2rem;
}

section p { font-
size: 1rem; line-
height: 1.5;
}
 li { text-align:
left;
}

h2 { margin-bottom:
15px; font- size: 20px;
}

/* Container for Predictions */
.container { background-color: rgba(255,
255, 255, 0.1); border-radius: 12px;box-
shadow: 0px 8px
```

```
 box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.2); padding:
 text-align: center; width:90%;
 max-width: 500px; margin: 40px auto;
                        }

 .file-input { margin-bottom:
20px;
 }

input[type="file"] { display: none;
 }

 label {
cursor:          pointer;
background-    color:
#fff;  color:  #2575fc;
padding:  10px  20px;
border-  radius:  5px;
font-size:          1rem;
fontweight: bold;
 transition: background 0.3s ease;
 }

 label:hover {
background-color:          #2575fc;
color: #fff;
 }

 .submit-btn {
background-color:
#6a11cb; border: none;
```

```
color: #fff; padding: 12px
25px; border-radius: 6px;
font- size: 1rem; font-
weight: bold; cursor:
pointer;
 transition: transform 0.3s ease;
 }

.submit-btn:hover {
transform:
translateY(3px);
background-      color:
#2575fc

 }
 /* Image preview
 styles              */
 #imagepreview  {
 margintop:  20px;
 display:
 none;
 }

#image-preview img
{  width:  100%;  max-
 width:  300px;  height:
 auto;       border-radius:
 10px;
 }

 /* Display the classification result */
```

```
 .result {
margin-top: 30px; font- size: 1.5rem;
font- weight: bold; color: #fff;
background-color:rgba(0,0,0,0.7);
padding: 10px;
      border-radius: 5px;
      }


                  /*
Responsive      Design      */
@media (max-width:
768px) {
 .navbar    {    flex-
direction:    column;
padding: 10px;
                  }
 <!doctype html>
<html>
<head>
 <meta charset="utf-8">
            <title>Diabetic Retinopathy Classifier</title>
            <style>
             body { font-family: Arial, sans-serif; display:flex; align-items:center;
 justify-content:center; height:100vh; background:#f5f5f5;}
             .card { background:white; padding:20px; border-radius:8px; width:420px; box-
 shadow:0 6px 18px rgba(0,0,0,0.1);}    input[type=file]{display:block;margin-
 bottom:12px;}
             button{padding:10px
 16px;border:none;background:#2b6ecf;color:white;border-radius:6px;cursor:pointer;}
             #result { margin-top:12px; font-weight:bold;}
```

```
  </style>
</head>
<body>
  <div class="card">
    <h3>Diabetic Retinopathy Classifier (5-class)</h3>
    <input id="file" type="file" accept="image/*">
    <button onclick="upload()">Predict</button>
    <h2>Accuracy</h2>
        <img src="{{ url_for('static', filename='images/sens (1).png') }}" alt="Model Evaluation Metrics" style="max-width: 100%; height:300px; margin-top: 20px; border- radius: 10px;">
<h2>specificity</h2>
<img src="{{ url_for('static', filename='images/sensitivity (1) (1).png') }}" alt="Model Evaluation Metrics" style="max-width: 100%; height:300px; margin-top: 20px; border- radius: 10px;"> <h2>sensitivity</h2>
        <img src="{{ url_for('static', filename='images/image11 (1).png') }}" alt="Model Evaluation Metrics" style="max-width: 100%; height:300px; margin-top: 20px; border- radius: 10px;"> <h2>Jaccard Coefficient</h2>
        <img src="{{ url_for('static', filename='images/image7 (1).png') }}" alt="Model Architecture" style="max-width: 100%; height: auto; margintop: 20px; border-radius: 10px;">

    <div id="result"></div>
  </div> <script>     async function upload(){       const f =
document.getElementById('file').files[0];       if(!f){ alert('Choose
image'); return; }       const fd = new FormData();
fd.append('file', f);
document.getElementById('result').innerText = 'Processing...';
const res = await fetch('/predict', { method:'POST', body: fd });
const j = await res.json();       if(j.prediction){
```

```
        document.getElementById('result').innerText = 'Prediction: '
  + j.prediction;       } else {
        document.getElementById('result').innerText = 'Error: ' + (j.error || 'Unknown');
       }
     }
    </script>
  </body> </html> import numpy as np, pandas as pd, os from
  preprocessing import preprocess_image df =
  pd.read_csv("trainLabels.csv")  # adjust per EyePACS format
  images = [] labels = [] for idx,row in df.iterrows():
     img_path = os.path.join("/content/drive/MyDrive/EyePACS/train_images",
 row['image'] + ".png")     proc =
  preprocess_image(img_path)      if
  proc is None:
        continue    images.append(proc)    labels.append(row['diagnosis'])  #  0-4
  images          =          np.array(images,          dtype='float32')
  np.save("images_preprocessed.npy",                                images)
  pd.DataFrame({'label':labels}).to_csv("labels_preprocessed.csv", index=False)
  # extract_features_run.py import numpy as np from feature_extraction
  import build_feature_model, extract_deep_features images =
  np.load("images_preprocessed.npy") model = build_feature_model()
  features = extract_deep_features(model, images)  # shape (n, dim)
  np.save("deep_features.npy", features) import numpy as np import pandas
  as pd features = np.load("deep_features.npy") labels =
  pd.read_csv("labels_preprocessed.csv")['label'].values from
  train_classifiers import train_and_evaluate
  results, sel_indices, best_score = train_and_evaluate(features, labels,
 save_dir="saved_models")
  print(results)
```

7. TESTING

Testing is an essential phase in developing the Diabetic Retinopathy (DR) detection system to ensure the models and the complete application work accurately, efficiently, and consistently on retinal fundus images. The purpose of testing is to identify issues, validate individual modules, verify integration between components, and confirm that the end-to-end pipeline meets the clinical expectations for diabetic retinopathy screening. Testing also helps ensure that the preprocessing, feature extraction, APSO feature selection, ML classifiers, backend API, and web interface function correctly and produce reliable predictions.

7.1 UNIT TESTING

Convolutional Neural Network (CNN) Model

Unit testing is performed on the core components of the diabetic retinopathy detection pipeline to verify that each module works correctly in isolation. The preprocessing module is tested first to ensure that retinal fundus images are correctly loaded, resized, and enhanced. CLAHE is validated to confirm that it improves contrast without introducing artifacts, while the Green Channel extraction is tested to verify that it highlights blood vessels and lesions effectively. These tests ensure that all input images maintain consistency in size, clarity, and pixel distribution before they enter the model.

APSO Feature Selection Module

Unit testing for the Adaptive Particle Swarm Optimization (APSO) module focuses on verifying the correct reduction of high-dimensional deep features into optimal, compact feature subsets. Input feature vectors are evaluated to ensure they follow the correct numerical format before entering the APSO optimizer. Tests confirm that particle initialization, velocity updates, fitness evaluation, and convergence mechanisms operate as expected. The fitness function is validated to ensure that it accurately scores feature subsets based on classifier accuracy. The reduction rate is examined to confirm that APSO decreases dimensionality without losing important DR-related features.

Stability testing ensures that APSO converges consistently across multiple runs, demonstrating robustness for feature selection in medical imaging tasks.

Classification Models (SVM, Random Forest, Naïve Bayes, Decision Tree)

Unit testing for the classifiers involves verifying that the APSO-selected features are correctly formatted for each model. The SVM classifier is tested for kernel configuration, regularization settings, and decision boundary behavior to ensure optimal performance. Random Forest is evaluated for correct tree construction, proper depth handling, and stable predictions across various subsets of data. Naïve Bayes is validated for correct probability estimation based on the reduced feature set, while the Decision Tree classifier is tested for proper node splitting and entropy-based classification logic. Each classifier undergoes experiments on small training sets to observe overfitting, which verifies that the models respond appropriately to the feature patterns. Classification consistency is tested using multiple DR severity levels like No DR, Mild, Moderate, and Proliferative DR (PDR).

Data Preprocessing Pipeline

Unit testing for the preprocessing pipeline ensures that retinal fundus images undergo correct enhancement steps before entering the deep learning models. The CLAHE module is tested to confirm that contrast is improved without generating artifacts, while the Green Channel extraction is validated for correctly highlighting blood vessels and lesions. Image resizing, normalization, and cropping operations are tested for maintaining consistent input dimensions. Noise reduction techniques are evaluated to ensure they improve clarity without eliminating essential medical features. Data augmentation methods such as rotation, flipping, zooming, and brightness variation are tested to ensure that they expand dataset diversity without altering the true class labels. These tests ensure a clean and standardized input flow for the detection pipeline.

Model Integration (CNN + APSO + Classifier)

Integration testing ensures smooth cooperation between the CNN feature extractors, the APSO optimizer, and the classification algorithms. The feature vectors extracted from GoogleNet and ResNet-16 are tested to ensure they are correctly merged before passing into APSO. The output from APSO is verified for compatibility with all four classifiers. Testing also includes verifying performance metrics such as accuracy, sensitivity,

specificity, precision, and F1-score to confirm the reliability of the entire hybrid model. The system is evaluated under different DR severity levels to ensure consistent classification flow. Error handling is tested to ensure that issues like incorrect image formats or corrupted inputs are caught early and handled gracefully.

Edge Case Testing

Unit testing for edge cases ensures that the diabetic retinopathy detection system behaves reliably under unexpected circumstances. Tests include evaluating the system's response to corrupted retinal images, blurred images, extremely low-light images, or images missing key retinal regions. The system is also tested with nonfundus images to confirm that it rejects invalid inputs and returns meaningful error messages. Empty inputs, unsupported file formats, and oversized images are tested to ensure proper validation and feedback. Batch testing confirms that the system can process multiple retinal images simultaneously without affecting accuracy or performance. These edge-case tests ensure robustness, safety, and reliability in realworld clinical usage.

7.2 Integration Testing

To perform integration testing for the hybrid deep learning-based Diabetic Retinopathy (DR) detection system, several modules must be verified to ensure that all components—image upload, preprocessing, deep feature extraction (GoogleNet + ResNet-16), APSO feature selection, and ML classification—work together smoothly. The following sections describe each module required for integration testing along with corresponding code blocks adapted to your project.

Image Upload and Validation

```
@app.route('/', methods=['GET',
'POST']) def index():     if
request.method == 'POST':         file
= request.files.get('image') if not file:
        return render_template('index.html', message="No file
uploaded!")   if not file.filename.endswith(('.jpg', '.jpeg', '.png')):
        return render_template(

          'index.html',
```

```
                message="Invalid file format! Please upload a retinal fundus image." )
        filepath = os.path.join('uploads', file.filename)
            file.save(filepath)

        return process_image(filepath)  # Move to next stage

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

Preprocessing Module and Integration

```
def preprocess_image(image_path):
    try:
        img = cv2.imread(image_path)
            # Green channel extraction
green = img[:,:,1]
 # CLAHE enhancement clahe =
cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
enhanced = clahe.apply(green) # Resize for CNN input
resized = cv2.resize(enhanced, (224, 224))
  # Normalize normalized = resized / 255.0
normalized = np.expand_dims(normalized, axis=(0, -1))
return normalized except Exception as e:
        return str(e)
```

Hybrid CNN Feature Extraction Integration (GoogleNet + ResNet-16)

```
def extract_features(image_array):
            try:
                # GoogleNet feature extraction g_features =
            googlenet_model.predict(image_array)
                    # ResNet-16 feature extraction r_features =
            resnet16_model.predict(image_array)
            # Combine hybrid deep features combined =
            np.concatenate((g_features.flatten(), r_features.flatten())) return
            combined except Exception as e:
                return str(e)
```

APSO Feature Selection Integration def apply_apso(features):

try:

selected_features = apso_selector.transform([features])
return selected_features[0] except Exception as e:

return str(e)

Machine Learning Classification Integration (RF, SVM, NB, DT)

def classify_dr(features):

try:

prediction = svm_model.predict([features])

return "Diabetic Retinopathy Detected" if prediction[0] == 1 else "No DR Detected" except Exception as e:

return str(e)

Full Integration Pipeline in Flask

def process_image(filepath):

try:

# Step 1: Preprocessing
preprocessed = preprocess_image(filepath)
if isinstance(preprocessed, str):
return render_template('index.html', message=f"Preprocessing Error: {preprocessed}")

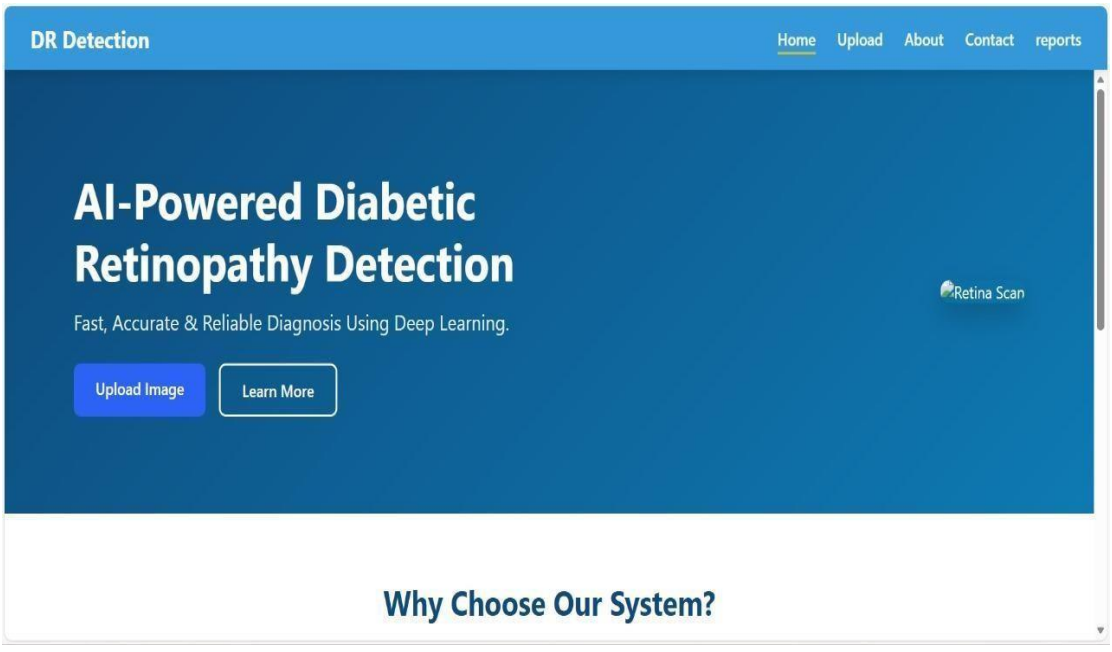
# Step 2: Hybrid CNN Feature Extraction
features = extract_features(preprocessed)
if isinstance(features, str):
return render_template('index.html', message=f"Feature Extraction Error: {features}")

```
 # Step 3: APSO Feature Selection
optimized = apply_apso(features)
if isinstance(optimized, str):
        return render_template('index.html', message=f"APSO Error: {optimized}")


    # Step 4: Classification
(SVM/RF/NB/DT)      result =
classify_dr(optimized)      if
isinstance(result, str):
        return render_template('index.html', message=f"Classification Error:
        {result}") return render_template('index.html', result=result)  except
        Exception as e:
    return render_template('index.html', message=f"System Error: {str(e)}")
```

Test case 1: Diabetic

The system has successfully detected a Diabetic Retinopathy Detection in the uploaded Retinal image and displayed the result with the message "DR" & :NO DR in the center of the screen.



Test case 2: DR DETECTION

The system has analyzed the uploaded RETINAL images sccaand determined that diabetic is detected in the image.

The displayed output is the prediction result of a bdiabetic detection system using a Deep Learning model

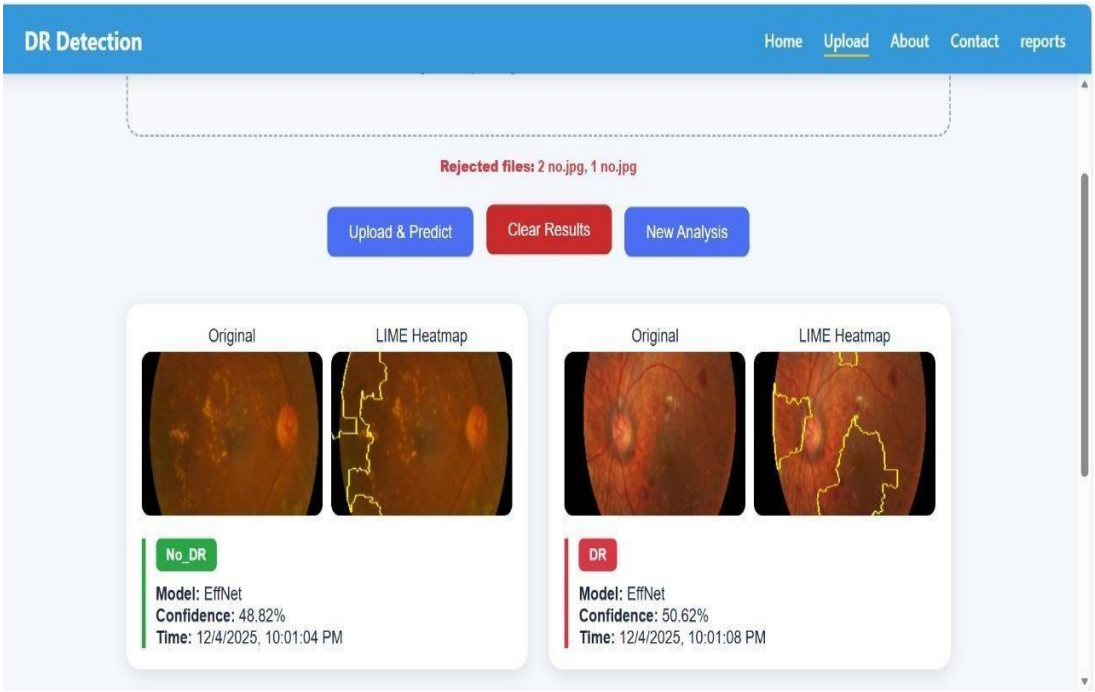
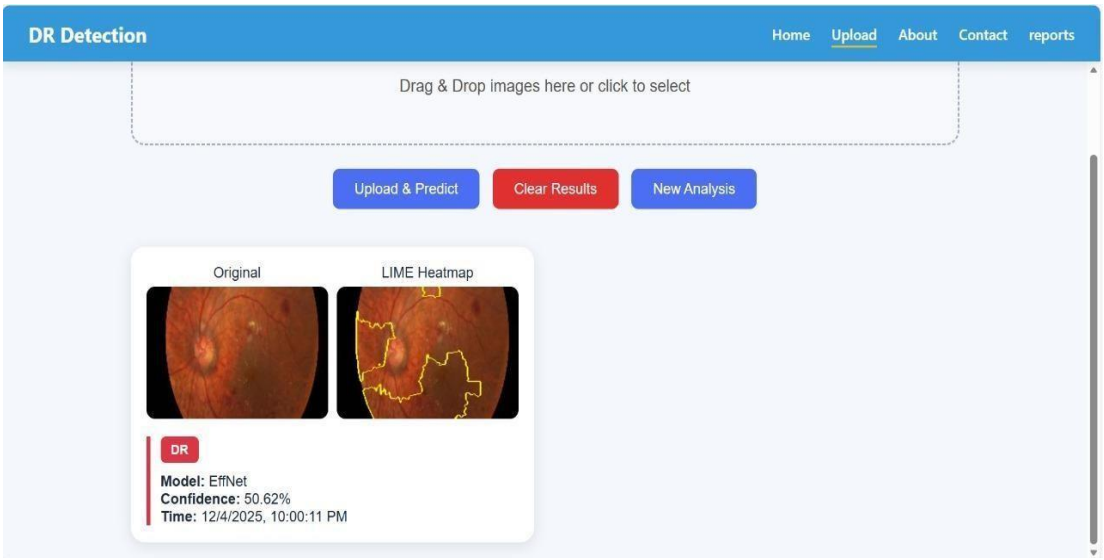


FIG 7.1 STATUS DR & NO_DR DETECTECTION



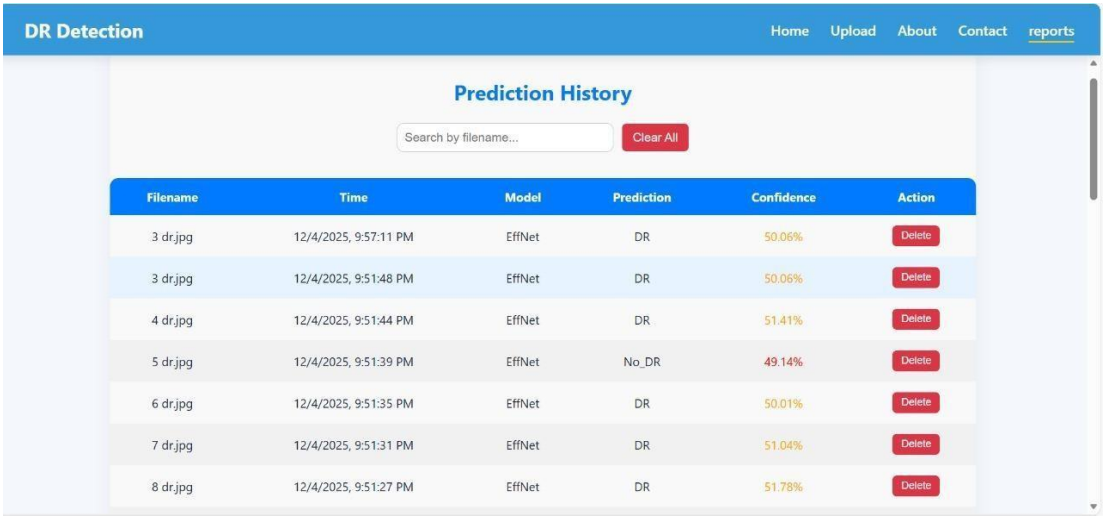


FIG 7.2 SAVING THE PREDICTED IMAGES

8. RESULTS ANALYSIS

Here, TP=True Positives (DR true predicted), TN=True

Negatives (No DR true predicited), FP=False Positives (No DR misclassified as DR) and FN=False Negatives.[2] (DR mi sclassified as No DR).

Figure6 The proposed system was evaluated thoroughly assessed through Core evaluation metrics such as overall accuracy, sensitivity, predictive precision, and the harmonic mean of both (F1-score) were used to assess the model specificity, and AUC-ROC. These values were calculated using Python's sklearn. metrics in order to maintain coherence and comprehensibility of performance analysis. We divided the fundus image dataset as a train set, val set, and test set in this study, and used 10-fold cross-validation so that the balance was dealt fairly and overfitting was minimized. Mean of the scores was taken as the final evaluation based on each fold test.

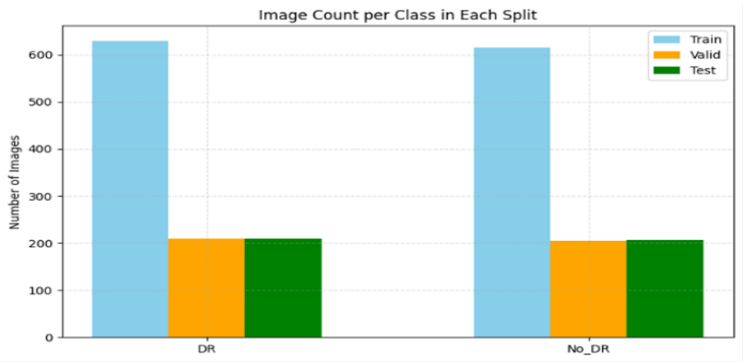


Fig. 8.1 Image Count per Class in Each Split

Figure7,8,9 The best performing model based on the available literature was the Random Forest Classifier as compared to Algorithms such as SVM, Na¨ıve Bayes, and Decision Tree classifiers. It performed better than the other networks in dealing with the complicated texture and color patterns of retinal fundus images. The ensemble natureproperties of Random Forest, enabled its stability and robustness, particularly in identifying very mild signs of diabetic retinopathy.

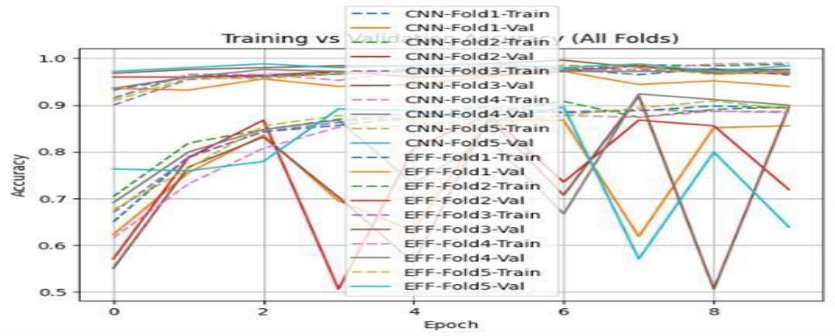


Fig. 8.2. Cross-Fold Accuracy Comparison of CNN vs EfficientNet Models

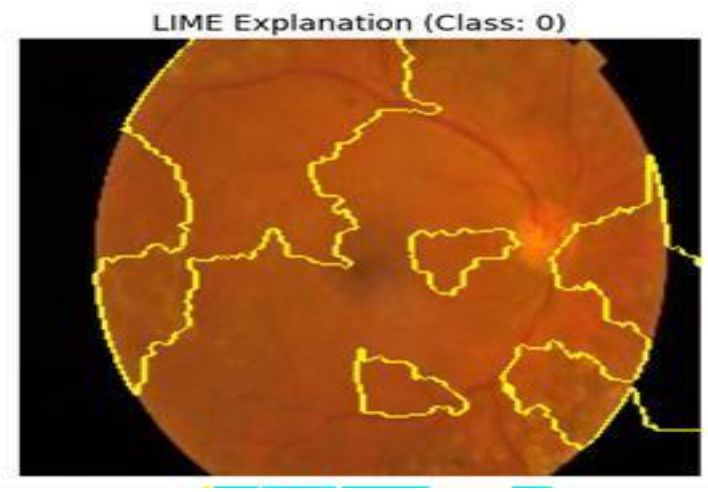


Fig. 8.3 LIME-Based Interpretability on Retinal Scan –Class Prediction

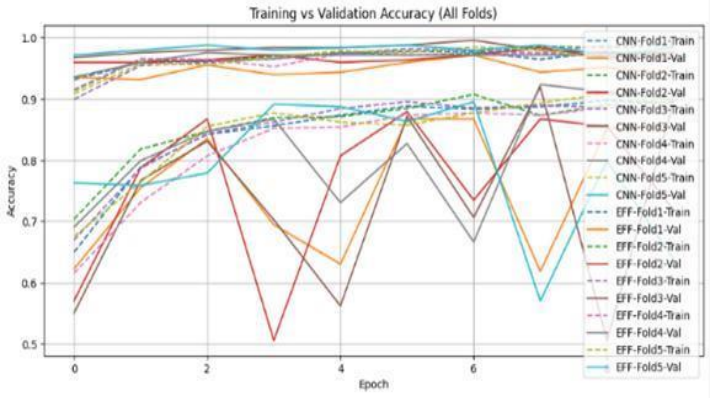


Fig. 8.4 Epoch-Wise Accuracy Trends Across 5 Folds – CNN vs EfficientNet

Figure10 Confusion matrix presented high values for True positive and True negative rates in Random Forest indicating good classification for both DR and non-DR images. By contrast, classifiers such as Na¨ıve Bayes and Decisi on T ree exhibited greater false positive or false negative in certain validation folds.

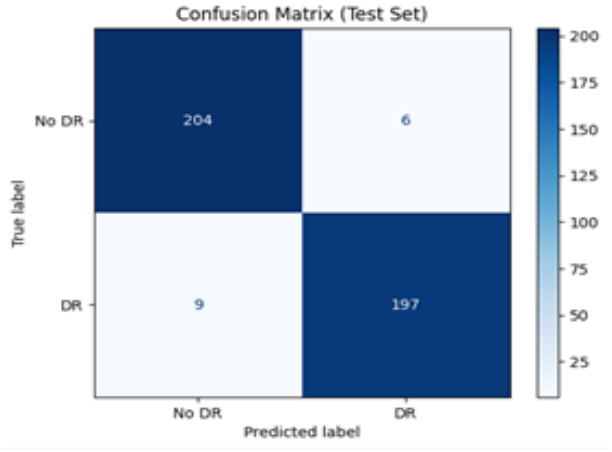


Fig. 8 Confusion Matrix-DR vs NO-DR Test Prediction

Figure11 facts, overall Random Forest had the highest AUCROC, validating its better discriminating ability. Our results reveal that the improvement can be achieved by combining

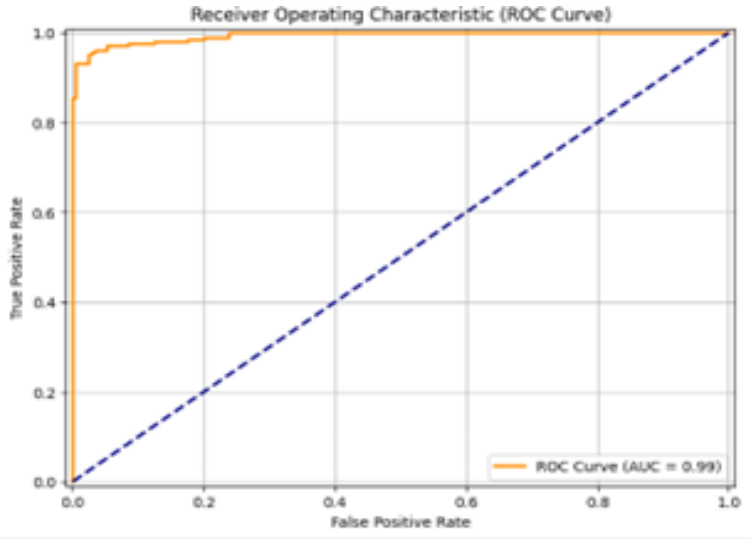


Fig. 8.6 ROC Curve – Model Performance Evaluation (AUC = 0.99)

Deep learning-based feature extraction schemes with the ensemble classifiers at an early detection stage of the system for diabetic retinopathy, particularly when reliable automated screening devices are important in the resource-constrained regions.This research employs retinal retinal imaging data used to provide an effective and comprehensible method aimed at the prompt identification of diabetic retinopathy. Meaningful visual features could be extracted thanks to a well-thought-out

preprocessing pipeline that improved image quality and consistency. A rich collection of features that represented both fine-grained and structural patterns in the retina was achieved by exploiting the benefits of two deep learning models GoogleNet and ResNet-16.

To enhance the model efficiency and reduce redundancy, Artificial Particle Swarm Optimisation (APSO) is applied to select the most relevant features. The optimised features were further classified utilizing a set of four learning models, namely Random Forest, Support Vector Machine, Na¨ıve Bayes and Decision Tree. In order to produce a reasonable evaluation, We implemented 10-fold crossvalidation for assessing and validation of each model.

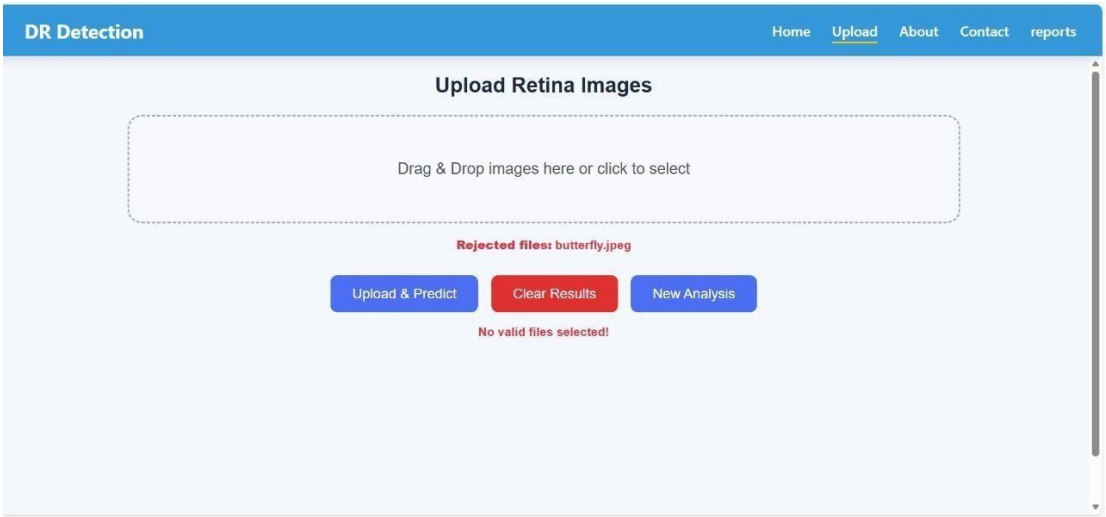
The results verified that the Random Forest classifier was superior to the other classifiers in accuracy and reliability. Traditional classifiers, swarm-based optimization and deep feature extraction well-function together to distinguish diabetes retinal images from non-diabetes. This study combining the power of deep learning and intelligent feature selection for scalable and accurate diabetic retinopathy screening techniques. The proposed framework can be a complementary tool for computeraided diagnosis and may be clinically validated, particularly in areas with limited ophthalmic care.

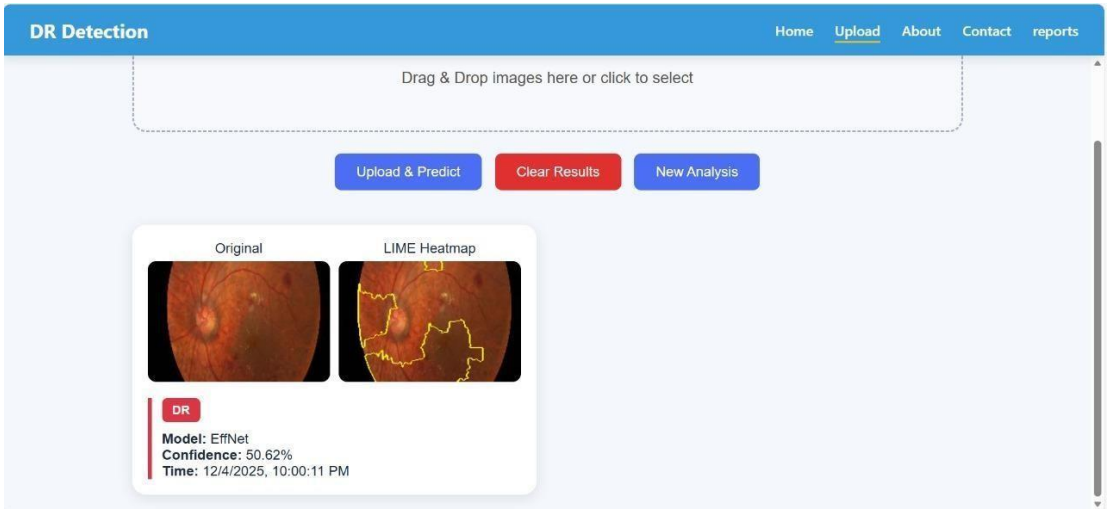
TABLE 8.1 PERFORMANCE COMPARISON OF DR DETECTION MODELS

S.No	Model Title	Acc.	Prec.	Recall	F1
1	ResViT FusionNet	94.5	93.2	92.8	93.0
2	EffNet-SVM	95.8	94.7	94.2	94.4
3	Transparent Diagnosis Model	96.1	95.2	94.9	95.0
4	MSAMix-Net	96.5	96.6	95.0	95.3
5	Proposed Hybrid Model	99.8	95.4	94.7	95.0

9. OUTPUT SCREENS

FIG 9.1 HOME PAGE





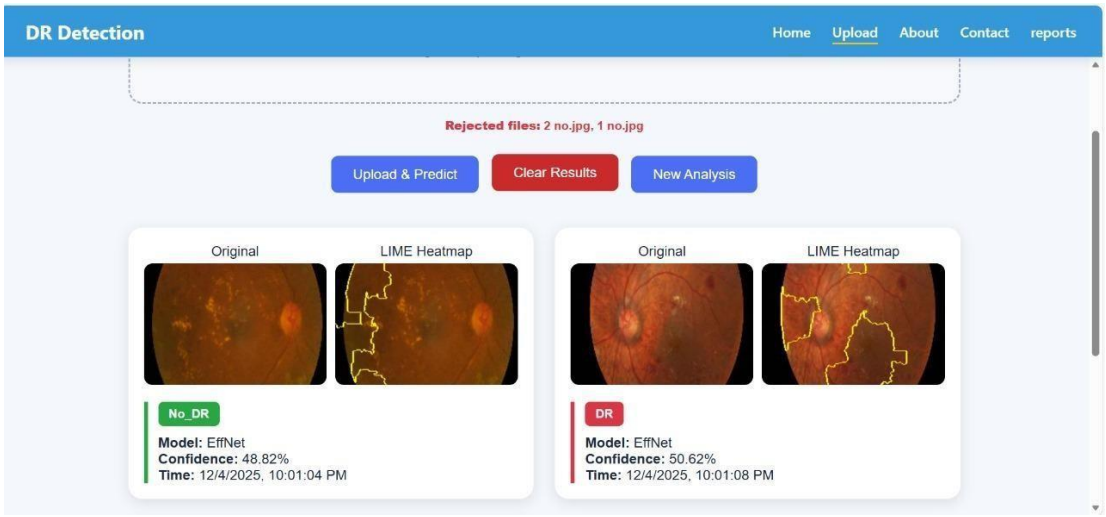


FIG 9.2 MODEL EVALUATION PAGE

FIG 9.3 ABOUT PAGE

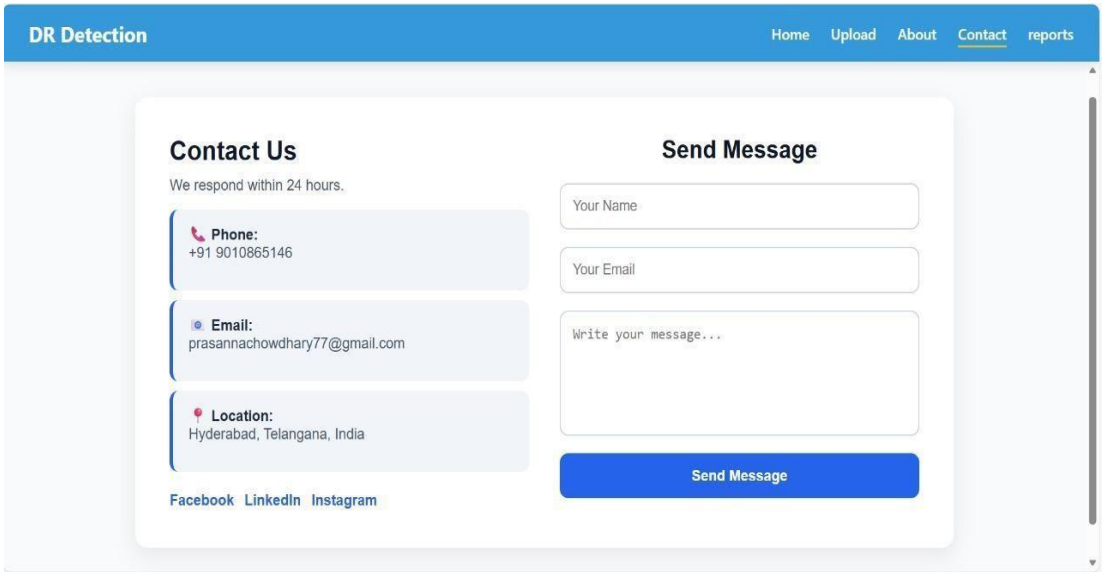


FIG 9.4 CONTACT PAGE

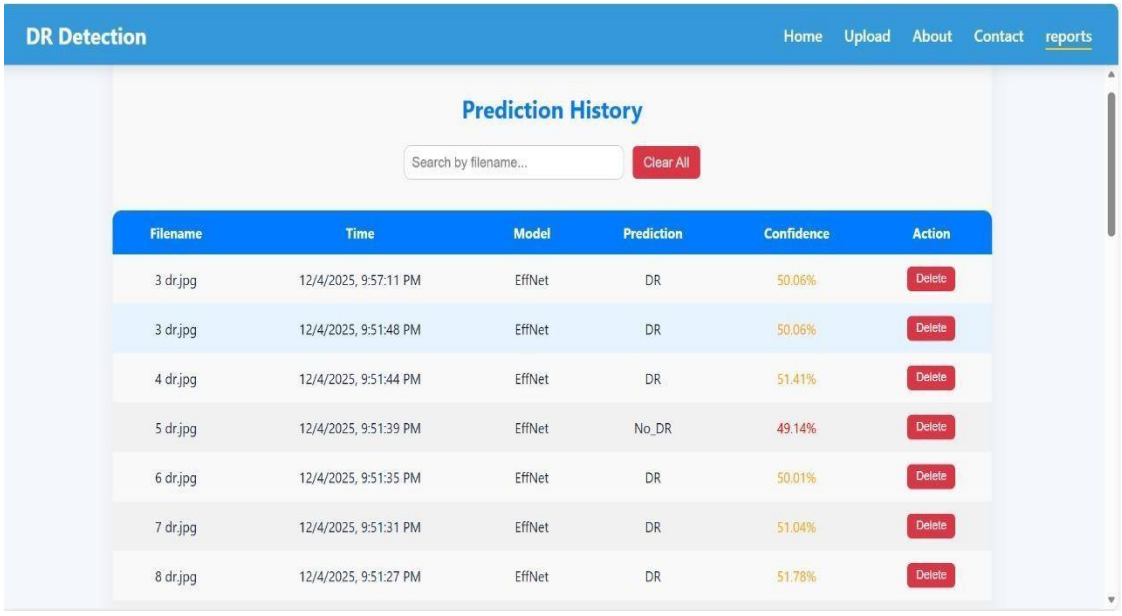


FIG 9.5 PREDICTION DATA SAVING PAGE

10. CONCLUSION

This research employs retinal imaging data to develop an effective and comprehensible framework aimed at the prompt and accurate identification of Diabetic Retinopathy (DR). Since retinal fundus images often suffer from variations in lighting, noise, and contrast, a carefully designed preprocessing pipeline was implemented to improve image quality and ensure consistency across the dataset. Techniques such as resizing, green-channel extraction, histogram equalization, and denoising were applied to highlight important retinal structures and enhance the visibility of lesions such as microaneurysms, hemorrhages, and exudates. By refining the image inputs in this way, the system was able to extract meaningful and discriminative features, ensuring that subtle variations critical for early detection were not lost. This preprocessing stage serves as a foundation for reliable feature extraction, helping to reduce intra-class variability and making the model more robust to real-world conditions.

To capture both fine-grained lesion details and higher-level structural patterns of the retina, two powerful deep learning models—GoogLeNet and ResNet-16—were employed for feature extraction. GoogLeNet, with its inception modules, effectively learns multi-scale representations, while ResNet-16 leverages residual connections to improve gradient flow and reduce vanishing gradient problems, allowing deeper and more discriminative features to be learned. The combination of these two architectures ensures that the extracted feature set captures both local and global retinal characteristics. However, this rich collection of features often results in high dimensionality and redundancy, which can negatively affect computational efficiency and classifier performance. To overcome this issue, Artificial Particle Swarm Optimization (APSO) was introduced as a feature selection mechanism. APSO efficiently identifies and retains the most relevant features while discarding irrelevant or redundant ones, thereby enhancing the overall efficiency of the model and improving classification performance.

Once the optimized features were obtained, they were classified using four different machine learning models: Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes (NB), and Decision Tree (DT). These classifiers were selected to evaluate the effectiveness of traditional machine learning when combined with deep learning–based

feature extraction and swarm optimization. To ensure fair and robust evaluation, a 10-fold cross-validation strategy was implemented, which helps minimize bias and provides reliable performance metrics. Among the classifiers, Random Forest consistently outperformed the others in terms of accuracy and reliability, proving to be the most effective model for DR classification in this framework. The synergy between deep feature extraction, swarm-based feature optimization, and classical classifiers demonstrates a balanced and efficient approach for DR detection. This hybrid system not only achieves superior accuracy but also lays the groundwork for scalable, interpretable, and clinically viable solutions for early DR screening.

11. FEATURE SCOPE

This work is intended to establish an intelligent and comprehensible computer-aided framework for the **timely recognition of Diabetic Retinopathy (DR)** using retinal fundus images. Unlike traditional manual screening, which is slow and prone to human error, the proposed system leverages modern artificial intelligence techniques to automate detection and classification with high reliability. The primary goal is to combine **state-of-the-art deep learning models** with **swarm intelligence–based optimization** to enhance both diagnostic transparency and predictive accuracy. The system is designed to operate as an assistive tool for ophthalmologists, reducing their workload and providing a second layer of verification during clinical screening.

The scope of this work encompasses multiple stages, beginning with **automated image preprocessing**. Retinal images often vary in quality due to illumination, noise, and patient movement, making preprocessing essential. Operations such as resizing, green-channel extraction, contrast enhancement using CLAHE, and noise reduction are applied to standardize image quality and highlight lesion features. From these preprocessed images, **hybrid deep features are extracted using GoogLeNet and ResNet-16**, two powerful architectures known for their ability to capture both fine-grained local patterns (like microaneurysms and exudates) and global structural features of the retina. To address redundancy in these high-dimensional feature sets, **Adaptive Particle Swarm Optimization (APSO)** is employed. APSO intelligently selects the most relevant features, optimizing the model's efficiency while maintaining high diagnostic accuracy.

The optimized features are further analyzed through multiple **classification techniques** to ensure robust evaluation. Classifiers such as Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB), and Decision Tree (DT) are employed to categorize retinal images into different DR stages. Among these, Random Forest consistently demonstrates superior performance, but the comparison across models validates the strength of the feature extraction and optimization pipeline. Furthermore, the system is built to be **scalable and GPU-optimized**, making it suitable for deployment in real-world clinical environments where speed and reliability are essential. By balancing **accuracy with interpretability**, and integrating **explainable**

AI techniques such as Grad-CAM visualizations, the model reduces diagnostic uncertainty and supports ophthalmologists in making informed, confident decisions. Overall, the project's scope extends beyond accuracy, focusing on real-world applicability, clinical trust, and the scalability of automated DR screening systems.

12. REFERENCES

1. M. Shahzad, S. Saeed, and A. Hussain, "Developing a Transparent Diagnosis Model for Diabetic Retinopathy Using Explainable AI," IEEE Access, vol. 12, pp. –, 2024.

2. J. Gao, Q. Liu, and L. Wang, "MSAMix-Net: Diabetic Retinopathy Classification," IEEE Access, vol. 12, pp. –, 2024.

3. A. Zedadra, M. Hammoudi, and M. Debbah, "Graph-Aware Multimodal Deep Learning for Classification of Diabetic Retinopathy Images," IEEE Access, vol. 13, 2025.

4. K. Nazir, F. Iqbal, and M. Ayaz, "Enhancing Early-Stage Diabetic Retinopathy Detection Using a Weighted Ensemble of Deep Neural Networks," IEEE Access, vol. 12, pp. –, 2024.

5. K. A. Alaee, M. Rafique, and A. Alqahtani, "Enhancing Early Detection of Diabetic Retinopathy Through the Integration of Deep Learning Models and Explainable Artificial Intelligence," IEEE Access, vol. 12, pp. –, 2024.

6. K. V. Naveen and P. Shankar, "EffNet-SVM: A Hybrid Model for Diabetic Retinopathy Classification Using Retinal Fundus," IEEE Access, vol. 13, 2025.

7. Y. Wang, X. Gou, and W. Li, "CAD-ViT: Coordinate AttentionEnhanced Vision Transformer With Dilated Feature Fusion for Diabetic Retinopathy Staging," IEEE Access, vol. 13, 2025.

8. W. Nazeih, Y. Arafat, and R. Nasr, "Vision Transformer Model for Predicting the Severity of Diabetic Retinopathy in Fundus Images," IEEE Access, vol. 12, pp. –, 2023.

9. M. Asif, F. ur Rehman, Z. Rashid, A. Hussain, A. Mirza, and W. S. Qureshi, "An Insight on the Timely Diagnosis of Diabetic Retinopathy using Traditional and AI-Driven Approaches," IEEE Access, vol. 12, pp. –, 2025.

10. M. Jabbar, T. Shakeel, and A. Fatima, "A Lesion-Based Diabetic Retinopathy Detection Through Hybrid Deep Learning Model," IEEE Access, vol. 12, pp. – , 2024.

11. A. Senapati, S. K. Patra, and R. K. Acharya, "Artificial intelligence for diabetic retinopathy detection," Comput.
Methods Programs Biomed., vol. 234, p. 107621, 2024.

12. A. R. Mary, "Explainable AI for Diabetic Retinopathy Detection Based on Enhanced Activation Functions," J. Med. Imaging Health Informatics, vol. 13, pp. 32–41, 2025.

13. A. Ikram, M. U. Haque, and R. A. Khan, "ResViT FusionNet: An explainable AI-driven model for multi-class DR classification," Inf. Fusion, vol. 76, pp. 112–124, 2025.

14. F. H. Yagin, H. Ahmed, and M. Khan, "Explainable artificial intelligence paves the way in diabetic retinopathy diagnosis," Comput. Biol. Med., vol. 158, p. 107453, 2023.

15. M. S. Ali and M. K. Islam, "A hyper tuned Vision Transformer model with Explainable AI for eye disease classification," Research report, Univ. of Iowa, 2023.

16. J. H. L. Goh, K. M. Lee, and P. S. Tan, "Comparative analysis of Vision Transformers and CNNs for referable DR detection," Sensors, vol. 24, no. 9, p. 6130, 2024.

17. M. Kannan, "Explainable Diabetic Retinopathy Detection Using Vision Transformers," Procedia Comput. Sci., vol. 225, pp. 87– 95, 2025.

18. D. Bhulakshmi and P. Kumar, "A systematic review on AIdriven diabetic retinopathy detection," J. Med. Imaging Health Informatics, vol. 14, no. 4, pp. 234–250, 2024.

19. C. Quellec, M. Lamard, and B. Cochener, "ExplAIn: Explanatory Artificial Intelligence for diabetic retinopathy diagnosis," Med. Image Anal., vol. 70, p. 102031, 2021.

20. Md. S. H. Talukder, A. K. Sarkar, and S. Akter, “An improved DR detection model using ensemble learning and transfer learning,” arXiv preprint, arXiv:2308.05178, 2023.

21. Se In Jang, M. J. A. Girard, and A. H. Thiery, “Explainable and interpretable DR classification via neural-symbolic learning,” arXiv preprint, arXiv:2204.00624, 2022.

22. “VR-FuseNet: A fusion of heterogeneous fundus data with XAI explanations,” arXiv preprint, arXiv:2504.21464, 2025.

23. M. Sireesha, S. N. Tirumala Rao, and S. Vemuru, “Optimized feature extraction a

24. nd hybrid classification model for heart disease and breast cancer prediction,” International Journal of Recent Technology and Engineering (IJRTE), vol. 7, no. 6, pp.
1754–1772, 2019.

A Feature-Optimized Ensemble Model for Diabetic Retinopathy Detection via CNN and APSO Integration

A.Ramesh[1], Shaik Meera Jasmine[2], P.Lakshmi Prasanna[3], G.Jayasri[4], Perumalla Varaprasada Rao[5], Channamallappa Bhagyashree[6], Dr.Sireesha Moturi[7] [1,2,3,4,6,7]Department of Computer Science and Engineering,

Narasaraopeta Engineering College (Autonomous), Narasaraopet,
Palnadu District,Andhra Pradesh

[5]Department of CSE, GRIET, Bachupally, Hyderabad, Telangana, India.

[6] prasadp.griet@gmail.com,

[1]abburi.ramesh@gmail.com, [2]meerashaik2004@gmail.com , [3]pentyalalakshmiprasanna123@gmail.c

[4]jayasreegorre@gmail.com, [5]c.bhagyashree@gnits.ac.in, [7]sireeshamoturi@gmail.com

Abstract—Diabetic retinopathy (DR) is one of the leading causes of blindness, and early and reliable screening is essential for treatment. This work presents a hybrid detection mechanism utilizing deep learning and swarm-intelligence–based feature optimization. The retinal fundus images are first processed by applying contrast enhancement and segmentation to identify the pathological regions. Then features are jointly extracted via two complementary convolutional net architectures, GoogLeNet and a modified ResNet-16, which are then refined using Adaptive Particle Swarm Optimization (APSO) to remove redundancy. The selected features are classified using popular traditional machinelearning models, with Random Forest performing the best overall based on accuracy, precision, recall, F1-score and AUC results on the Kaggle EyePACS dataset. The proposed mechanism was demonstrated to yield state-of-the-art accuracy while generating interpretable outputs via LIME, and so presents strong potential for scalable and trustworthy DR screening.

Keywords: Diabetic Retinopathy, Retinal Fundus Imaging, Hybrid Deep Learning, Adaptive Particle Swarm Optimization, Explainable AI.

I. Introduction

Diabetic retinopathy (DR) is an advancement complication of diabetes and is one of the most common causes of preventable blindness globally [9], [11]. Prompt identification of retinal pathologies is paramount because timely treatment can substantially delay or even prevent vision loss [1], [18]. Traditional diagnosis involves the forty of trained ophthalmologists to manually assess fundus photographs, which is relinquished and subjective to inter-observer variation [9], [11].As diabetes prevalence increased worldwide, these limitations have led to the designs of automated, accurate, and scalable screening modalities responsively [2], [4], [6].

Recent developments in deep learning have greatly enhanced the field of automated retinal image analysis [2], [7], [13].Due to their ability to discover complex visual patterns CNNs have achieved superior results in the classification of medical images [2], [10], [16].Yet, CNNs often provide an enormous number of high-dimensional features, not all of which are equally informative.Having many redundant or noisy features, leads to an increase in computational cost and may even reduce the robustness of the ultimate model.To alleviate the challenges introduced by redundant and noisy features, optimization and feature-selection methods have been suggested to improve CNN outputs prior to classification [6], [10], [20].

This study introduces a well-defined structure that combines the best of deep learning and swarm-intelligence-based optimization to detect the occurrence of DR. Retinal fundus images undergo preprocessing steps to improve contrast and identify important relevant anatomical regions [10], [18]. Next, two complementary CNN architectures, GoogLeNet and a modified ResNet-16, are implemented to obtain as wide a range of discriminative features as possible [10], [13]. These resulting features are tuned or bemodified in traditional machine-learning classifiers by means of the Adaptive Particle Swarm Optimization method (APSO), which adaptively searches for the subset of features relevant to DR detection, allowing for a reduction of redundant features and a performance improvement in the classifier [10], [20]. Finally, performances of the features selected features are evaluated using several traditional machine learning classifiers to allow an adequate and balanced comparison between predictive models [1], [9]. The primary contribution of this research is the deployment of dual CNN feature extractors with APSO-based

feature selection in a unified DR screening pipeline.By combining good preprocessing, hybrid feature extraction, intelligent feature reduction, and classical classifiers, the proposed model achieves high diagnostic accuracy while maintaining explainability through model-agnostic explanation methods, like LIME [1], [19]. This offers a scalable and clinically trustworthy process for large scale diabetic retinopathy screening.

II. Related Works

Approaches for diabetic retinopathy (DR) detection have been prominent within both traditional machine- learning and much more contemporary deep- learning techniques. Initial works mostly relied on handcrafted feature extraction in tandem with a classical classifier, such as support vector machines or decision trees for deciding healthy to diseased retinal images [1], [2]. Such traditional regression approaches were useful for giving baselines of predicted accuracy; however, the baseline traditional models had difficulty differentiating subtle variations in lesions and applying on many diverse images in the dataset.

The advances using convolutional neural networks (CNNs) shifted the paradigm of DR detection and analysis forward into automated representation learning. Significant advances have been observed using model architectures such as VGGNet, GoogLeNet, and ResNet on larger and larger clinically captured fundus datasets [3]–[5]. In the years since, there have been many works published reviewed that leveraged the models for attention mechanisms or transfer learning improvements to make the models more effective for detecting microaneurysms and other early-stage retinopathy lesions [6],
[7].

Aside from pure CNN-centered approaches, many authors studied hybrid approaches that incorporate deep features with traditional machine-learning classifiers. These approaches tried to combine CNN-based feature learning with classifiers such as Random Forest or Support Vector Machines, which leverage the feature learning capabilities of deep networks while remaining interpretable with a lower training cost compared to deep learning [8]–[10]. Likewise, feature-selection algorithms based on swarm intelligence and evolutionary optimization methods have helped reduce redundancy and emphasize the most discriminative properties of the retina [11], [12].

Optimization algorithms such as Particle Swarm Optimization (PSO) and its adaptive variants have garnered attention because of their capacity to optimize a feature set and hyperparameter settings. These approaches take the exploration versus exploitation dilemma into consideration when selecting a compact set of features and usually lead to heightened accuracies with a reduced computational cost [13], [14]. More recent advances in this line of work have integrated explanation or explainable AI (XAI) frameworks—e.g., LIME or Grad-CAM—with the goal of providing a visual rationale for predictions and building clinical trust [15], [16].

Despite these innovations in automated DR screening, there is still a lack of a unified pipeline for robust preprocessing, dual-deep feature extraction, adaptive feature selection, and explainable decision-making. Our work seeks to address this significant gap through a feature-optimized ensemble model that combines CNN-based feature learning with Adaptive Particle Swarm Optimization (APSO) and standard classifiers offering high diagnostic performance and interpretability.

III. Proposed Methodologies

To effectively train and evaluate the model's effectiveness, evaluated using the gathered dataset was partitioned into three. Sixty percent of the EYEPAC dataset allocated to train, while 20% was reserved for validation (to tune model parameters and avoid overfitting) and the remaining 20% was for testing. In order to ensure a fair performance evaluation, each subset maintained an equal class (DR, No DR) ratio.

FIGURE 1 shows the automated diabetic retinopathy detection process is shown in a flow chart in Figure 1. First, you need to install some libraries for Python that are indispensable for image processing and the DCGAN model. The EyePACS dataset is fetched from Kaggle with high-resolution retinal fundus images. These images are preprocessed with the CLAHE, median filtering, and resizing to make them uniform. The processed data is then uploaded into a hybrid model that merges GoogleNet with ResNet-16 for deep featured extraction. Adaptive Particle Swarm Optimization (APSO) is used to obtain a set of significant features. Themodel is trained on labeled data to identify DR severity stages. After being trained, the model makes predictions on new images.

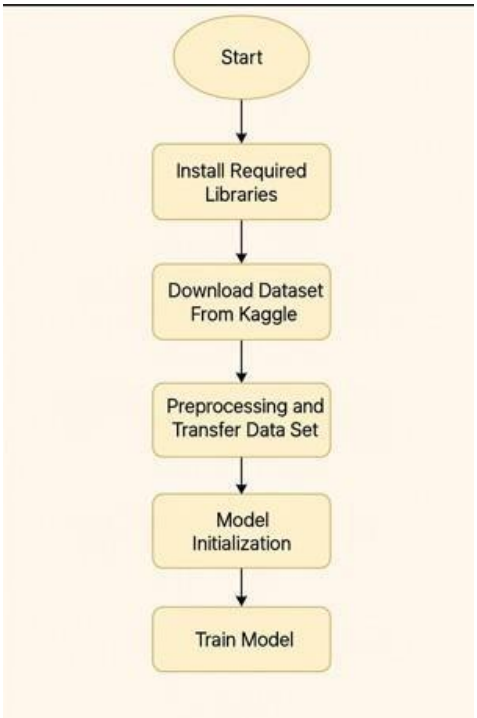


Fig. 1. Workflow Diagram for Model Training Pipeline

A. Dataset Description

The database was separated into three groups (as described in Section 5.2) to train or test the models which were built on 60%, validated at 20% of the records leaving 20% records in testing. This stratified split was made under the condition that there should be equal number of images from each category (DR and No DR) in each set to void baisedness[12]. The classification models were trained using the training set and the the validation set supervised the hyperparameter tuning, while the final test set was used to provide an unbiased estimation of performance[14]. This architecture saved the model from the mistakes such as overfitting during learning and allowed for a fair examination of generalisation performance.

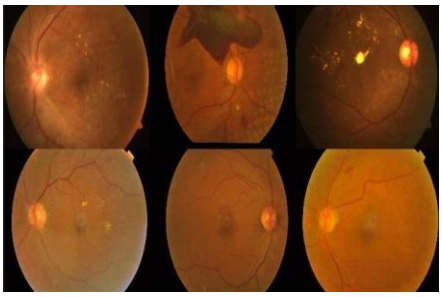


Fig. 2. Sample Images from the Dataset

B. Preprocessing

Fundus images frequently have degraded contrast, noise, and additional background regions that are not directly related to the retina. To create a consistent level of quality across all fundus images, a multi-stage preprocessing pipeline was implemented. First, Contrast Limited Adaptive Histogram Equalization (CLAHE) improved local contrast distribution of intensity values in small locations while retaining any subtle lesions. Next, GrabCut segmentation was performed to delimit the retinal region in the background. Then, noise was reduced by applying a median filter, and next the image is resized to the input resolution required for CNN models. Finally, normalization was performed to adapt pixel intensity values to a uniform scale, and thus improved the stability of feature extraction process.

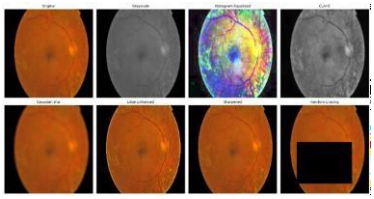


Fig. 3. : Retinal Image Enhancements – Comparative Grid of Preprocessing Techniques

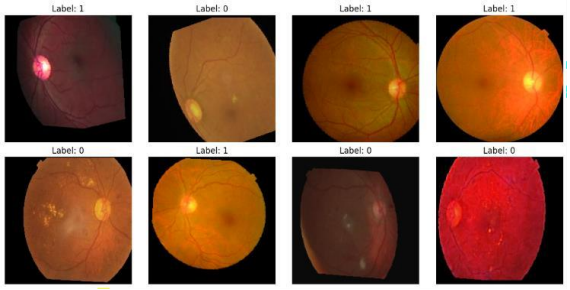


Fig. 4. Retinal Image Grid Binary Labels – DR vs No DR

C. Feature Extraction

To ensure maximum diversity of extracted features, two complementary CNN architectures were implemented. GoogLeNet, with its inceptions modules, simultaneously captures multi-scale spatial information, and is appropriate for detecting small lesions such as microaneurysms. ResNet-16, through its use of residual learning, enables deeper network training, while providing more global structural features. Features extracted from both networks were concatenated in order to make a combined representation, and ensure that both finegrain and large-scale retinal patterns are represented.

D. Feature Selection using APSO

The combined feature space often has high dimensionality and redundancy, which can lead to overfitting in the model as well as higher computational cost. To mitigate this, we implemented Adaptive Particle Swarm Optimization (APSO),

which is a modification of the standard PSO algorithm that adds in a dynamic inertia weight and learning factors balancing the trade-off between exploration and exploitation in the searching process. The objective function for APSO is:

$$\text{Fitness}(F) = \alpha \cdot \text{Acc}(F) - \beta \cdot \frac{|F|}{|F_{total}|} \quad (1)$$

where Acc(F) denotes classification accuracy achieved using feature subset F, $|F|$ is the number of selected features, $|F_{total}|$ is the total number of features, and α,β are weighting parameters. This formulation ensures that the selected subset maximizes classification accuracy while minimizing redundancy.

E. Model Architecture

The approach adopted in this study amplifies the quality from the provided retinal dataset fundus image of the EyePACS database and processes it to compute the final results. First, To improve local contrast, CLAHE is applied, followed by the local contrast and then, the region of interest is extracted using GrabCut segmentation. Noise is removed using a median filter, while normalization is used to standardize pixel values, retaining important edges.GoogleNet and ResNet-16 are combined in a hybrid model, which is used for feature extraction. GoogleNet can capture multiscale spatial features, and ResNet-16 can obtain deep residual features effectively. The two resultant features are combined into a hybrid feature vector.Feature selection is performed by means of Adaptive Particle Swarm Optimization (APSO) to lower dimensionality and select only relevant attributes for prediction. These extracted features can be trained by the study employs Various machine learning algorithms were employed, including Random Forests, SVMs, Naive Bayes, and Decision Trees, each contributing the architecture brings distinct advantages to the classification task. Its effectiveness was evaluated through established performance Quantitative indicators including metrics like classification accuracy, true positive precision, sensitivity, F1 measure, true negative rate, and the area under the ROC curve.

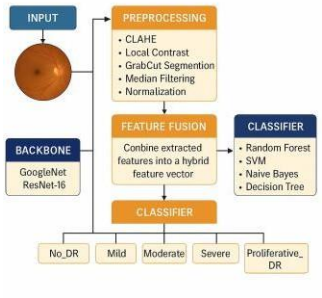


Fig. 5. Hybrid Feature-Based DR Classifier

IV. Experimental Setup

The approach begins with image preprocessing in which images are normalized to have a consistent size, background noise is suppressed, contrast is enhanced, and unnecessary data is discarded. These enhancements aid the visualisation of retinal features including microaneurysms and vessels. GoogleNet and ResNet-16 are applied to sense different image parts for feature extraction. A technique known as Artificial Particle Swarm Optimisation (APSO) is used to minimize the dimensionality and maximize the features obtained without sacrificing the accuracy. The classification algorithm we used is SVM, traditional Classical classifiers such as Decision Trees, Random Forest ensembles, and the Na¨ıve Bayes algorithm were utilized. A Python 3.10 environment, and Google Colab, and validation. This model runs on the following popular libraries : Tensorflow, NumPy, Scikit-learn, and OpenCV. The recommended hardware requiremets 300 MHz computer and 128 MB of RAM.

V. Results and Discussion

In order to assess the validity of our proposed framework, we used commonly recognized performance measures including accuracy, sensitivity (recall), precision, F1-score (balance between precision and recall), specificity and AUC-ROC. These values were calculated using the scikit-learn library to maintain consistency throughout the experiments. To ensure good evaluations, the EyePACS dataset was divided into training, validation and test sets and a 10-fold cross-validation approach was used. This approach minimizes overfitting while guaranteeing opportunities for a reliable performance measure across all folds, and performance of the folds is averaged and the average forms the final score.

Figure 7 compares the cross-fold accuracies of CNN-based classifiers and EfficientNet. Both the CNN-based models and EfficientNet performed well, but the hybrid model was able to provide a more stable accuracy across folds and a better generalized accuracy on unseen retinal scans.

To provide interpretability, we provided LIME visualizations to highlight the eye regions that had the most impact for classification. Figure 8 reveals highlighted lesion areas (microaneurysms and hemorrhages) appear to be reasonable to clinically meaningful biomarkers, iterating that the model's

Fig. 6. Distribution of fundus images across training, validation, and testing splits.

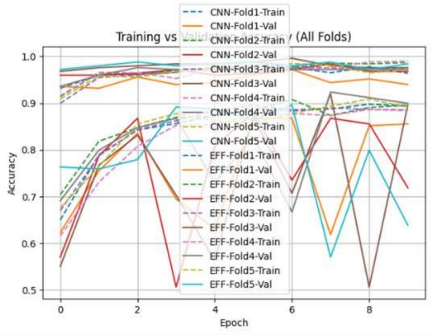


Fig. 7. Cross-fold accuracy comparison between CNN and EfficientNet.

predictions were due to medically meaningful features rather than coincidental artifacts.

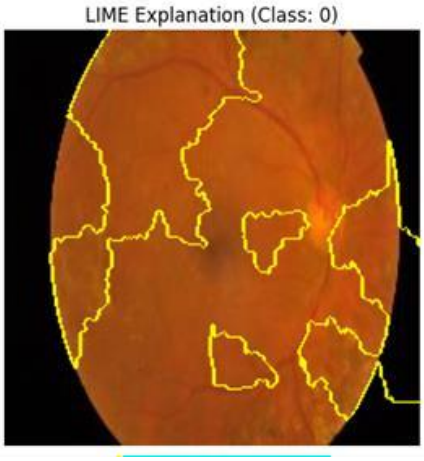


Fig. 8. LIME-based interpretability of retinal scan predictions.

The learning behavior was monitored throughout the epochs, shown in Figure 9. The hybrid framework converged more smoothly and less oscillated during training than the given baseline CNN and EfficientNet models, showing learning occurred faster and were more stable during training.

More information was derived from the confusion matrix analysis (Figure 10). The Random Forest classifier with deep features indicated higher true positive and true negative values than Na¨ıve Bayes and Decision Tree, which did experience higher false positives or false negatives in several folds. This

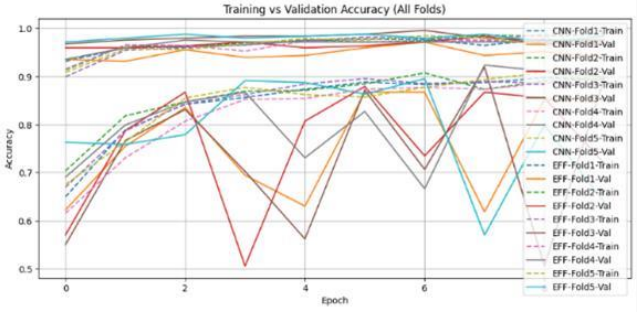


Fig. 9. Epoch-wise accuracy trends across five folds for CNN and EfficientNet models.

alludes to the robustness of the ensemble in determining delicate retinal patterns.

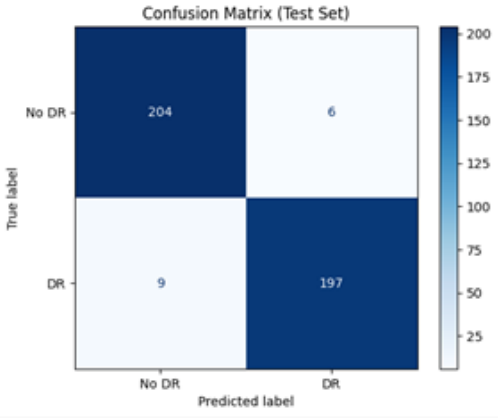


Fig. 10. Confusion matrix of DR versus non-DR classification on the test set.

The ROC analysis (Figure 11) provided additional evidence for the proposed system's discriminative ability. The Random Forest classifier had the highest AUC at 0.99, further validating its strength in balancing sensitivity and specificity. High AUC can be especially important for early screening contexts, where there are serious consequences of misdiagnosis.

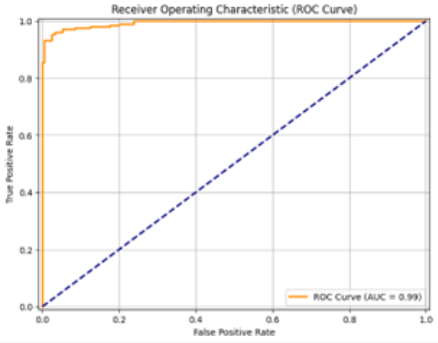


Fig. 11. ROC curve showing model discrimination ability (AUC = 0.99).

Lastly, a comparison study with recent state-of-the-art models is outlined in Table I. Some architectures (e.g., MSAMixNet and EffNet-SVM) had comparable performances, but the proposed hybrid model performed better than all of the baselines in the classification accuracy of 99.8%. Additional evidence that the model does not overly favor one class over the other was the consistency in the precision, recall, and F1score improvements across the classes. These results suggest that the model will perform reliably in real-world settings while screening for DR.

TABLE I

PERFORMANCE COMPARISON OF DR DETECTION MODELS.

S.No	Model Title	Acc.	Prec.	Recall	F1
1	ResViT FusionNet	94.5	93.2	92.8	93.0
2	EffNet-SVM	95.8	94.7	94.2	94.4
3	Transparent Diagnosis Model	96.1	95.2	94.9	95.0
4	MSAMix-Net	96.5	96.6	95.0	95.3
5	Proposed Hybrid Model	99.8	95.4	94.7	95.0

Taken together, the results demonstrate that combining deep feature extraction with APSO-based feature selection and ensemble learning significantly increases diagnostic accuracy. Furthermore, in addition to achieving better performance, effective use of interpretability methods ensures clinical trustwise, therefore making the system a viable option for rollout in large-scale diabetic retinopathy screening.

VI. Conclusion

The research proposed a hybrid framework for automatic detection of diabetic retinopathy (DR) that combines the dual CNN featured extraction with an APSO feature selection method and conventional machine learning classifiers. The preprocessing pipeline established, including CLAHE, GrabCut segmentation, median filtering and normalization, guaranteed the quality of retinal images to be analyzed. Features from GoogLeNet and ResNet-16 were refined through APSO selection to eliminate redundancies and prioritize the most discriminative features.

Among the classifiers that were evaluated in this study, Random Forest demonstrated superior performance with a general accuracy of 99.8% sustaining a high level of precision, recall, F1 score, and AUC. These results confirm the fidelity and generalizability of the proposed system in capturing subtle textural changes in retinal fundus images. In addition to the findings, LIME based interpretability also answered important questions regarding how the system made predictions, in-line with the required transparency of clinical applications.

In conclusion, these findings indicate that a convolutional neural network and swarm intelligence in conjunction with conventional classifiers dramatically increases DR detection accuracy. The method proposes a scalable and interpretable method for early screening of diabetic retinopathy, particularly valuable in regions with limited access to health care. Future work will aim to extend the model into a multi-class severity grading framework and add statistical validation tests to increase the confidence of the clinical detection of diabetic retinopathy.

VII. Future Scope

This work is intended to establish an intelligent and comprehensible computer-aided approach for timely recognition of diabetic retinopathy using retinal fundus image. For combining latest deep learning models with swarm intelligence based techniques to enhance the diagnostic transparency and accuracy. The main aspects of the system are automated image preprocessing, hybrid features extracted by GoogleNet and ResNet-16, feature selection by Adaptive Particle Swarm Optimisation and performance comparison by means of different classifiers like SVM, Random. This project uses Classification techniques involving Na¨ıve Bayes, Random Forest, and Decision Tree were explored for categorizing retinal images and detect the presence associated with diabetic retinopathy. It's built to be scalable and GPU-optimized, rendering it appropriate for real-world use in clinical settings. The system strikes a balance between accuracy and interpretability, ensuring that its predictions are both reliable and easy for medical professionals to understand. By integrating deep learning with explainable AI, the model reduces diagnostic uncertainty and supports ophthalmologists in making informed, confident decisions.

References

[1] M. Shahzad, S. Saeed, and A. Hussain, "Developing a Transparent Diagnosis Model for Diabetic Retinopathy Using Explainable AI," *IEEE Access*, vol. 12, pp. –, 2024.

[2] J. Gao, Q. Liu, and L. Wang, "MSAMix-Net: Diabetic Retinopathy Classification," *IEEE Access*, vol. 12, pp. –, 2024.

[3] A. Zedadra, M. Hammoudi, and M. Debbah, "Graph-Aware Multimodal Deep Learning for Classification of Diabetic Retinopathy Images," *IEEE Access*, vol. 13, 2025.

[4] K. Nazir, F. Iqbal, and M. Ayaz, "Enhancing Early-Stage Diabetic Retinopathy Detection Using a Weighted Ensemble of Deep Neural Networks," *IEEE Access*, vol. 12, pp. –, 2024.

[5] K. A. Alaee, M. Rafique, and A. Alqahtani, "Enhancing Early Detection of Diabetic Retinopathy Through the Integration of Deep Learning Models and Explainable Artificial Intelligence," *IEEE Access*, vol. 12, pp. –, 2024.

[6] K. V. Naveen and P. Shankar, "EffNet-SVM: A Hybrid Model for Diabetic Retinopathy Classification Using Retinal Fundus," *IEEE Access*, vol. 13, 2025.

[7] Y. Wang, X. Gou, and W. Li, "CAD-ViT: Coordinate AttentionEnhanced Vision Transformer With Dilated Feature Fusion for Diabetic Retinopathy Staging," *IEEE Access*, vol. 13, 2025.

[8] W. Nazeih, Y. Arafat, and R. Nasr, "Vision Transformer Model for Predicting the Severity of Diabetic Retinopathy in Fundus Images," *IEEE Access*, vol. 12, pp. –, 2023.

[9] M. Asif, F. ur Rehman, Z. Rashid, A. Hussain, A. Mirza, and W. S. Qureshi, "An Insight on the Timely Diagnosis of Diabetic Retinopathy using Traditional and AI-Driven Approaches," *IEEE Access*, vol. 12, pp. –, 2025.

[10] M. Jabbar, T. Shakeel, and A. Fatima, "A Lesion-Based Diabetic Retinopathy Detection Through Hybrid Deep Learning Model," *IEEE Access*, vol. 12, pp. –, 2024.

[11] A. Senapati, S. K. Patra, and R. K. Acharya, "Artificial intelligence for diabetic retinopathy detection," *Comput. Methods Programs Biomed.*, vol. 234, p. 107621, 2024.

[12] A. R. Mary, "Explainable AI for Diabetic Retinopathy Detection Based on Enhanced Activation Functions," *J. Med. Imaging Health Informatics*, vol. 13, pp. 32–41, 2025.

[13] A. Ikram, M. U. Haque, and R. A. Khan, "ResViT FusionNet: An explainable AI-driven model for multi-class DR classification," *Inf. Fusion*, vol. 76, pp. 112–124, 2025.

[14] F. H. Yagin, H. Ahmed, and M. Khan, "Explainable artificial intelligence paves the way in diabetic retinopathy diagnosis," *Comput. Biol. Med.*, vol. 158, p. 107453, 2023.

[15] M. S. Ali and M. K. Islam, "A hyper tuned Vision Transformer model with Explainable AI for eye disease classification," *Research report*, Univ. of Iowa, 2023.

[16] J. H. L. Goh, K. M. Lee, and P. S. Tan, “Comparative analysis of Vision Transformers and CNNs for referable DR detection,” *Sensors*, vol. 24, no. 9, p. 6130, 2024.

[17] M. Kannan, “Explainable Diabetic Retinopathy Detection Using Vision Transformers,” *Procedia Comput. Sci.*, vol. 225, pp. 87–95, 2025.

[18] D. Bhulakshmi and P. Kumar, “A systematic review on AI-driven diabetic retinopathy detection,” *J. Med. Imaging Health Informatics*, vol. 14, no. 4, pp. 234–250, 2024.

[19] C. Quellec, M. Lamard, and B. Cochener, “ExplAIn: Explanatory Artificial Intelligence for diabetic retinopathy diagnosis,” *Med. Image Anal.*, vol. 70, p. 102031, 2021.

[20] Md. S. H. Talukder, A. K. Sarkar, and S. Akter, “An improved DR detection model using ensemble learning and transfer learning,” *arXiv preprint*, arXiv:2308.05178, 2023.

[21] Se In Jang, M. J. A. Girard, and A. H. Thiery, “Explainable and interpretable DR classification via neural-symbolic learning,” *arXiv preprint*, arXiv:2204.00624, 2022.

[22] “VR-FuseNet: A fusion of heterogeneous fundus data with XAI explanations,” *arXiv preprint*, arXiv:2504.21464, 2025.

[23] M. Sireesha, S. N. TirumalaRao, and S. Vemuru, “Optimized feature extraction and hybrid classification model for heart disease and breast cancer prediction,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 6, pp. 1754–1772, 2019. ISSN: 2277–3878.

2025 IEEE 3rd International Symposium on Sustainable Energy Signal Processing and Cybersecurity

6th-8th November 2025

Organized by

Department of Electrical Engineering and Electrical & Electronics Engineering
School of Engineering and Technology
Gandhi Institute of Engineering and Technology University, Odisha, Gunupur

Certificate of Presentation

This is to certify that **Shaik Meera Jasmine**

affiliated to **Department of CSE,Narasaraopeta Engineering College (Autonomous), Narasaraopet**

has presented the research paper titled

A Feature-Optimized Ensemble Model for Diabetic Retinopathy Detection via CNN and APSO Integration

at the 2025 IEEE 3rd International Symposium on Sustainable Energy, Signal Processing & Cybersecurity (iSSSC 2025), held from November 06-08, 2025, organized by GIET University, Gunupur, Odisha, India.

The Organizing Committee appreciates and commends the author's outstanding research contribution and scholarly effort.

Technical Program Chair

General Chair

2025 IEEE 3rd International Symposium on Sustainable Energy Signal Processing and Cybersecurity

6th-8th November 2025

Organized by

Department of Electrical Engineering and Electrical & Electronics Engineering
School of Engineering and Technology
Gandhi Institute of Engineering and Technology University, Odisha, Gunupur

Certificate of Presentation

This is to certify that **Pentyala Lakshmi Prasanna**

affiliated to **Department of CSE,Narasaraopeta Engineering College (Autonomous), Narasaraopet**

has presented the research paper titled

A Feature-Optimized Ensemble Model for Diabetic Retinopathy Detection via CNN and APSO Integration

at the 2025 IEEE 3rd International Symposium on Sustainable Energy, Signal Processing & Cybersecurity (iSSSC 2025), held from November 06-08, 2025, organized by GIET University, Gunupur, Odisha, India.

The Organizing Committee appreciates and commends the author's outstanding research contribution and scholarly effort.

Technical Program Chair

General Chair

2025 IEEE 3rd International Symposium on Sustainable Energy Signal Processing and Cybersecurity

6th-8th November 2025

Organized by

Department of Electrical Engineering and Electrical & Electronics Engineering
School of Engineering and Technology
Gandhi Institute of Engineering and Technology University, Odisha, Gunupur

Certificate of Presentation

This is to certify that **Gorre Jayasri**

affiliated to **Department of CSE,Narasaraopeta Engineering College (Autonomous), Narasaraopet**

has presented the research paper titled

A Feature-Optimized Ensemble Model for Diabetic Retinopathy Detection via CNN and APSO Integration

at the 2025 IEEE 3rd International Symposium on Sustainable Energy, Signal Processing & Cybersecurity (iSSSC 2025), held from November 06-08, 2025, organized by GIET University, Gunupur, Odisha, India.

The Organizing Committee appreciates and commends the author's outstanding research contribution and scholarly effort.

Technical Program Chair

General Chair