Multi-Class Eye Diseases Prediction Using Ensemble CNN With Max Voting Strategy

Adebayo Ademola Riliwan[1*], Obansola Oluwatoyin Yemi[2], Aweda Olusina Temidayo[3], Chidozie Ifeanyi Evans[4]

Federal School of Surveying, Oyo, Oyo State, Nigeria

***Corresponding Author**

DOI: https://dx.doi.org/10.51584/IJRIAS.2026.11050100

Received: 06 May 2026; Accepted: 12 May 2026; Published: 03 June 2026

ABSTRACT

Eye diseases such as cataract, glaucoma, and diabetic retinopathy are major causes of blindness worldwide, underscoring the critical need for early and accurate diagnosis. This study presents a novel approach to multi-class eye disease prediction using an ensemble of Convolutional Neural Networks (CNNs) combined with a max voting strategy. The framework integrates four CNN models with varying architectural depths, each trained on a curated dataset of retinal fundus images, to classify eye conditions into four categories: cataract, glaucoma, diabetic retinopathy, and normal. The methodology begins with data pre-processing, which includes resizing, normalization, and augmentation to ensure robust model training. Each CNN model, ranging from six to nine layers, was trained independently for 60 epochs, leveraging techniques like dropout and regularization to prevent overfitting. The models' outputs were aggregated using a bagging ensemble technique, with final predictions determined through max voting. The ensemble approach effectively combines the complementary strengths of the individual models, enhancing classification reliability. Experimental evaluations demonstrate that the ensemble achieves superior performance, with an average accuracy of 92%, precision of 92%, recall of 91%, and F1-score of 92%. This study highlights the potential of combining deep learning with ensemble strategies for improved diagnostic accuracy in medical image analysis. By offering a scalable and reliable tool for early detection of eye diseases, this research contributes to advancing automated healthcare diagnostics, aiming to reduce the global burden of vision-related diseases and improve patient outcomes.

Keywords: Convolution Neural Network, max voting, ensemble learning, fundus image, classification.

INTRODUCTION

Eye diseases are a significant global health concern, often leading to impaired vision or blindness if not diagnosed and treated promptly. These diseases can arise from a variety of factors such as genetics, underlying health conditions, age-related changes, lifestyle factors, and environmental influences. Early detection plays a crucial role in preventing the progression of these conditions. Understanding the nature of these eye diseases, their causes, symptoms, and risk factors is crucial for early detection, proper management, and preservation of vision [1]. With advancements in medical imaging and artificial intelligence, automated systems for eye disease diagnosis have gained prominence, offering faster, accurate, and cost-effective solutions compared to traditional methods [2].

According to Marouf et al., [3], eye diseases such as cataract, glaucoma, and diabetic retinopathy are leading causes of blindness worldwide. Early detection and diagnosis of these diseases can prevent vision loss and improve the quality of life for many individuals. Medical methods of diagnosing eye diseases rely heavily on the expertise of ophthalmologists and sophisticated medical imaging techniques. However, with advancements in machine learning and deep learning, automated systems for eye disease detection have become a promising alternative [3].

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Deep learning has proven to be solutions for computer vision problems such as image enhancement, segmentation and classification, especially in biomedical imaging. Advancements in deep learning methodologies have exhibited potential in the field of medical image analysis. Many researchers have proposed various approaches for the classification of ophthalmological diseases. Specifically, Convolutional Neural Networks (CNNs) have demonstrated cutting-edge performance in various tasks related to medical image classification, such as the diagnosis of eye diseases [4].

Convolutional Neural Networks (CNNs) have emerged as a powerful tool in medical image analysis due to their ability to learn and extract complex features from images. However, a single CNN model may not always provide optimal accuracy, particularly in situation of multi-class classification tasks that involves diverse eye diseases. To address this challenge, ensemble methods, which combine the strengths of multiple models, have been introduced to enhance prediction performance. Multi-class classification using CNNs involves training models to differentiate between multiple categories of eye diseases, such cataract, glaucoma, and diabetic retinopathy as well as normal eye [4].

Significant efforts have been focused towards the enhancement of eye diseases detection using diverse approaches in the area of related studies. Deep learning, mainly the use of Convolutional Neural Networks (CNNs), has emerged as a popular method for accurate diseases categorization. Studies have shown that CNN designs can achieve good diagnosis accuracy across a wide spectrum of eye diseases [5]. Its ability of transferring learning algorithms that leverage pre-trained models from other domains have showed promise in improving diseases detection model performance.

Ensemble learning technique known as bagging (bootstrap aggregation) when employed, improves the stability and accuracy of machine learning models. This technique involves training of multiple models independently and combining their predictions. The idea is to reduce variance and avoid overfitting by averaging the predictions of several models. In the context of CNNs, bagging can be implemented by training several CNN models with different architectures and then combining their predictions using methods such as max voting [3].

Max voting is a common method used in ensemble learning to aggregate predictions from multiple models. The strategy combines predictions from multiple CNN models, ensuring more accuracy of the prediction system and reliable classification by leveraging the complementary strengths of each model to. Each model makes a prediction, and the final prediction is determined by the majority vote [4].

This paper exhibits the usage of four distinct CNN models with varying layer configurations: Model_A has 6 layers, Model_B has 7 layers, model_C has 8 layers and model_D has 9 layers. The CNN models are implemented with different hidden layers each to detect eye diseases such as cataract, glaucoma, diabetic retinopathy and normal eye. The ensemble approach combines the prediction from these models to improve the overall accuracy of the system. It focuses on developing an ensemble CNN model integrated with a max voting (majority voting) strategy for the multi-class prediction of eye diseases. The ensemble approach leverages the complementary strengths of individual CNN models, improving diagnostic accuracy and robustness. The max voting strategy, which aggregates predictions from multiple models, ensures a more reliable and confident classification of eye diseases.

The framework is evaluated on publicly available data sets of eye images, demonstrating its efficacy in detecting various eye diseases such as cataracts, glaucoma, and diabetic retinopathy, as well as normal eyes. By combining advanced deep learning techniques with ensemble learning, this study aims to contribute to the development of automated tools for improved eye care diagnostics.

Existing Eye Diseases Image Classification Systems

A variety of projects have designed and implemented eye diseases identification system using deep learning (DL) and machine learning (ML).

K [6] developed eye diseases identification using deep learning model. A model combining deep learning (DL) and machine learning (ML) to classify eye diseases like cataracts, crossed eyes, bulging eyes, uveitis, and conjunctivitis was presented. The model shows high accuracy in distinguishing these conditions, indicating its

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

potential for automated diagnosis. However, the study emphasizes the need for further comparison with traditional methods and human expert evaluations to confirm its effectiveness in clinical applications.

Londhe [7] focused on developing classification of eye diseases hybridizing CNN-RNN models. The research focused on classification of fundus images into various eye diseases such as cataracts, glaucoma, and retinal conditions. Using a hybrid CNN-RNN model, which has been successful in various disease classification tasks, the study highlights the improved accuracy achieved on balanced datasets compared to imbalanced ones, with similar accuracy levels observed across models trained on the latter.

Kumar & Bindu [8] developed an innovative framework that utilizes a combination of DenseNet201, EfficientNetB4, and ResNet105 models. The authors focus on preprocessing steps like contrast enhancement and normalization to prepare the dataset. By employing the stacking technique, they integrate the strengths of different CNN models to enhance disease detection capabilities. The study is significant in aiding ophthalmologists by providing a second reference in diagnosing multiple diseases, although it doesn't address real-time processing constraints.

Smaida & Yaroshchak [4] focused Bagging of Convolutional Neural Networks for Diagnostic of Eye Diseases the authors explore the use of bagging ensembles to improve the accuracy and reliability of diagnosing eye diseases using deep learning. They propose leveraging ensemble methods with multiple Convolutional Neural Networks (CNNs), including pre-trained models like VGG16 and InceptionV3, to enhance the performance of multi-class classification tasks on fundus image datasets. The aim is to increase diagnostic accuracy by mitigating overfitting and reducing variance, thus providing more consistent and stable predictions.

Elloumi et al. [2] developed Ocular Diseases Diagnosis in Fundus Images using Deep Learning: Approaches, Tools, and Performance Evaluation" focuses on developing a deep learning-based method for diagnosing ocular diseases using fundus images. The work involved collecting a dataset of fundus images, preprocessing them, and applying deep learning algorithms, particularly convolutional neural networks (CNNs), possibly enhanced with transfer learning or ensemble methods. The results demonstrated high accuracy, sensitivity, specificity, and AUC-ROC in identifying and classifying ocular diseases like diabetic retinopathy, age-related macular degeneration, and glaucoma. However, the study's effectiveness and generalizability are contingent on the quality and diversity of the dataset, with potential limitations arising from dataset size, disease representation, and demographic diversity.

METHODOLOGY

The research introduced a new multi-class eye disease prediction tool using bagged Convolutional Neural Networks (CNNs) with varying layers. It begins with dataset, data preprocessing, where eye image dataset comprising of the diseased eye and normal eye is prepared for model training. Four CNN models with different numbers of layers (6, 7, 8, and 9 layers) are initialized with varying complexities in their development.

The choice of the four CNN models; Model_A through Model_D was intentional to explore the impact of increasing architectural complexity on classification performance. Starting with a simpler 6-layer CNN (Model_A) and progressively increasing the number of convolutional layers up to 9 (Model_D) allowed for the capture of increasingly abstract features from the input images. This progression was designed to evaluate how deeper networks affect diagnostic accuracy, with each model using a standard filter size and pooling strategy to maintain comparability. The variation in depth supports a diverse ensemble, enhancing the model’s ability to generalize across complex and heterogeneous retinal fundus image data.

Each CNN model is individually trained for 60 epochs on the pre-processed dataset, during which it learned to identify patterns associated with different eye conditions. The use of dropout layers and regularization techniques in the models aimed to enhance generalization and prevent overfitting. Appropriate loss functions and optimizers are chosen for training. The choices are; cross-entropy loss for classification and the Adam optimizer because of its adaptive learning rate capabilities which is effective in training deep learning models for efficient training of the implemented models.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

The training process is monitored using metrics such as accuracy, loss, precision, recall, and F1-score to evaluate its performance on a validated dataset to ensure effective learning and prevent overfitting. The models are combined using the bagging (bootstrap aggregation) technique, which involves ensemble of two or more models to improve prediction outcome. The combined predictions are aggregated using max voting, where the most frequent prediction is chosen as the final output. Each model is compiled with the selected parameters.

The evaluation of the bagged models was conducted taking into consideration, metrics like accuracy, precision, recall, and F1-score to assess their performance in predicting eye diseases. Figure 1 represents the workflow diagram, it depicts the process flow from the dataset input to preprocessing, training of the implemented CNN models, bagging output based on max voting and finally evaluation.

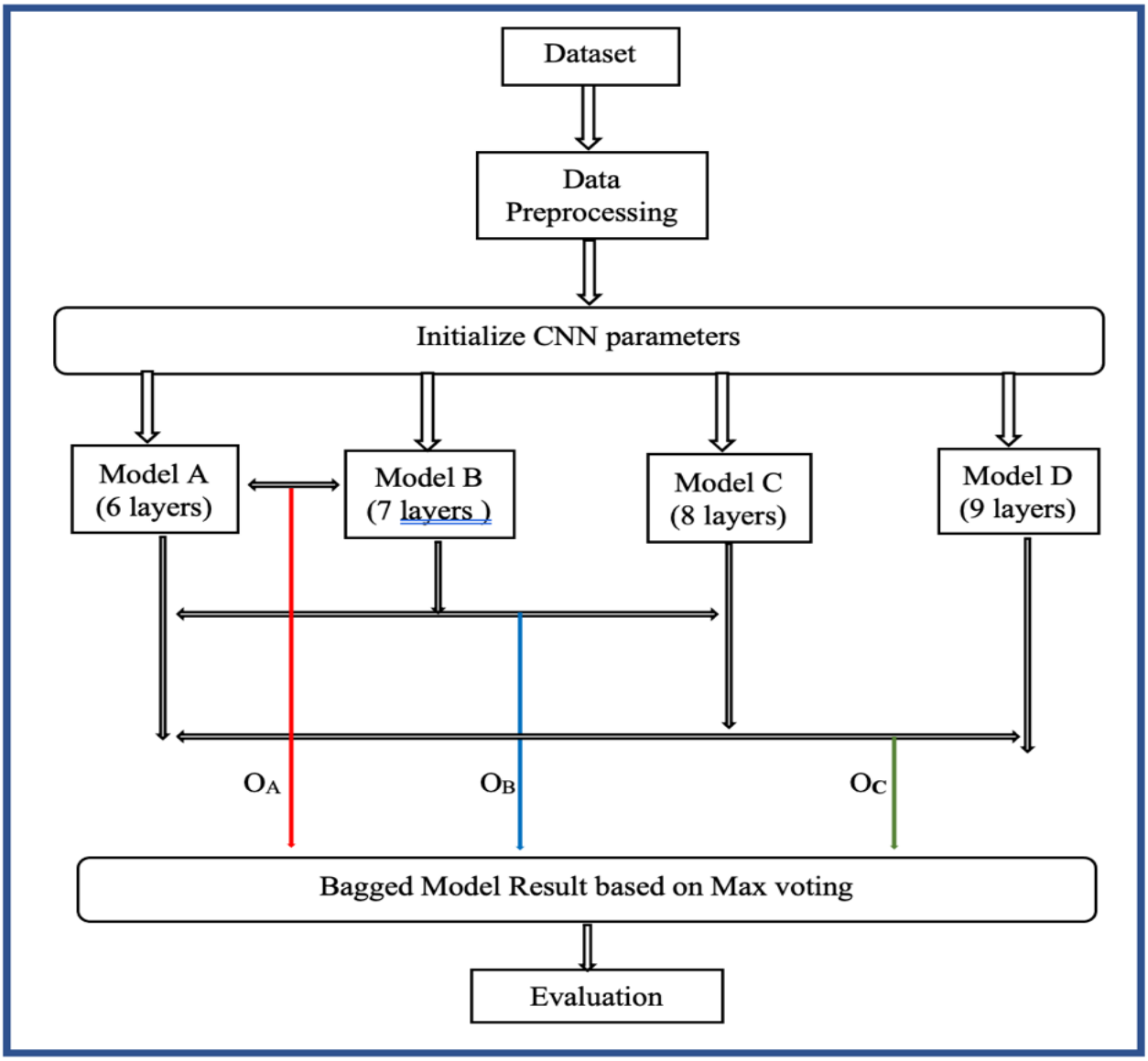


Figure 1: Workflow Diagram

As shown in figure 1, four CNNs (Model A–D, 6–9 convolutional blocks) trained independently for 60 epochs with dropout 0.3, Adam optimizer. Predictions aggregated via confidence-weighted max voting:

vote = argmax Σ wi·pi…….equation 1

where:

wi is model validation F1.

pi is the prediction of model.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Dataset

Dataset is a collection of data that is structured and organized for training the models or processing. In this context, the data type is image. A comprehensive dataset is gathered, containing images of diseased and normal eyes. The dataset used was acquired from Kaggle, a data science website that contains a variety of dataset for research purpose. It has an extensive dataset with a variety of fundus images capturing different eye diseases form the basis of the study to guarantee robustness and representativeness. The dataset is divided into subsets for training, validation, and testing, enabling objective model evaluation. The dataset contains 4217 retinal fundus images that has been curated from IDRiD, Ocular Disease Intelligent Recognition (ODIR) dataset, High-Resolution Fundus (HRF) Image Database. Images of Glaucoma, Cataract, Diabetic-retinopathy, and Normal were among the four eye conditions depicted. Each photograph has the proper illness labelled on it by ophthalmologist. Source: https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification.

Eye Disease	No. of instances	Diseases	Not Diseases	Sex(%F)	Age (mean ±SD)	Percentage (%) of Instances	Image Type	Resolution
Cataract	1038	Yes	No	44%	56±12	25.2%	RFI	256x256
Diabetic retinopathy	1098	Yes	No	48%	62±14	26.7%	RFI	512x512
Glaucoma	1007	Yes	No	52%	58±10	24.5%	RFI	256x256
Normal	1074	No	Yes	55%	48±11	26.1%	RFI	512x512

Table 1: Dataset Presentation

Table 1 showed a diverse and balance representation of eye diseases for effective model training and evaluation. Images of four specific eye conditions namely cataract, diabetic retinography, glaucoma and normal eye with a relative balanced instances for each category to bring to minimal the bias in the training process. Column 3 and 4 specify whether image represents diseased eye or not diseased providing clarity on the classification labels while column 5 and 6 shows the demography transparency in terms of sex either male and female, and the age distribution. The fifth column shows the percentage distribution of various instances. All images are retinal fundus images, a standard type for diagnosing eye conditions, and they come in varied resolutions (256x256 or 512x512 pixels), which adds diversity and robustness, allowing for more generalized and accurate model predictions.

Data Preprocessing

Data preprocessing is the preparation of the raw dataset for training by performing series of transformations to improve model accuracy and efficiency. This stage includes resizing images, normalization, data augmentation, and splitting the dataset into training, validation, and test sets. Classes were imported for this stage such as the ImageDataGenerator class, which allows for training, validation and testing. The training dataset generator (train_datagen) is enhanced with augmentation techniques such as rescaling, shearing, zooming, and horizontal flipping to improve the model's robustness by providing a variety of altered images during training. On the other hand, the validation and test datasets (validation_datagen and test_datagen) are only rescaled to maintain data consistency and ensure that the model's performance is evaluated on data similar to what it will encounter in real world scenarios.

The target_size parameter ensures that all images are resized to the same dimensions using 64x64 pixels, which is crucial for consistent input size across all images fed into the model. The batch_size parameter defines how many images are processed together in each iteration, balancing computational efficiency and memory usage. The preprocessing steps ensure that the model is trained on well-augmented data for robustness while being evaluated on realistic and normalized data for better generalization. Figure 2. depict a code snippet of dataset preprocessing, specifying image size, batch size, data augmentation techniques.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

DATASET PREPROCESSING

In [4]:
```
# Define image size and batch size
img_height, img_width = 64, 64
batch_size = 32

# Create ImageDataGenerators for training, validation, and testing
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
validation_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
```

Figure 2: Data Preprocessing

CNN Models Design and Implementation

This research involves the design and implementation of four distinct Convolutional Neural Network (CNN) models named Model_A, Model_B, Model_C, and Model_D with varying layers configuration to capture different levels of feature complexity and abstraction from the input images. The configurations are as follows:

Model_A: A CNN with 6 layers

Model_B: A CNN with 7 layers

Model_C: A CNN with 8 layers

Model_D: A CNN with 9 layers

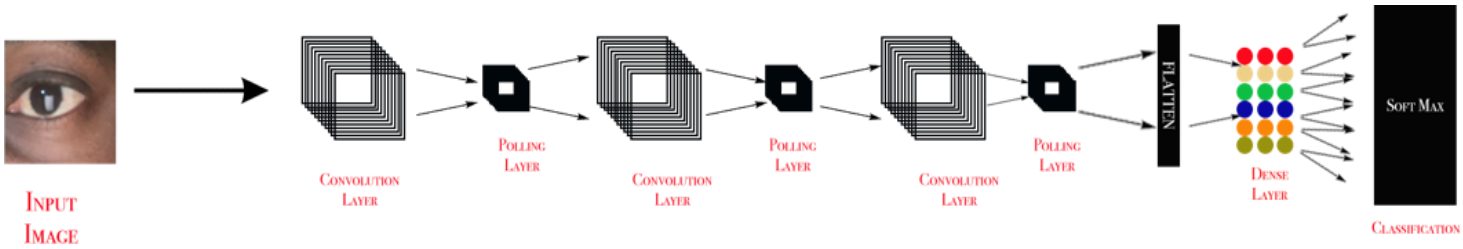
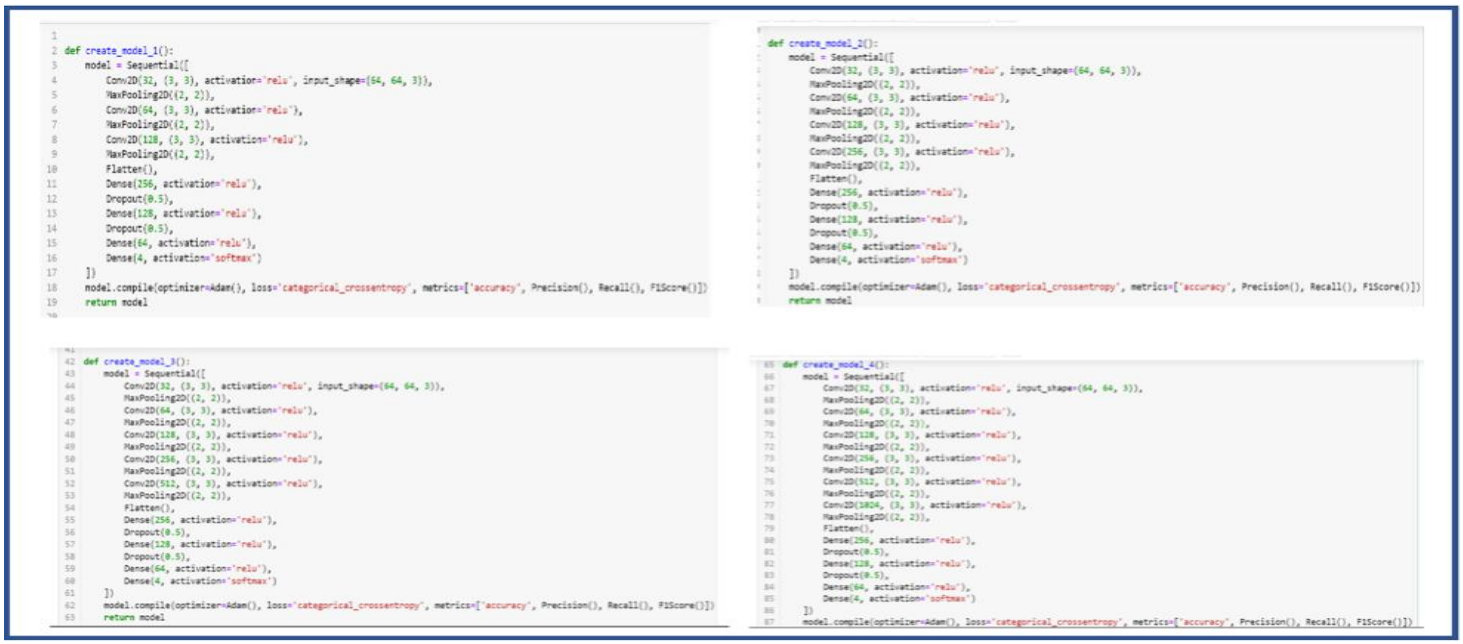


Figure 3: CNN Architecture for The Model



```
def create_model_1():
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
        MaxPooling2D((2, 2)),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Flatten(),
        Dense(256, activation='relu'),
        Dropout(0.5),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dense(4, activation='softmax')
    ])
    model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy', Precision(), Recall(), F1Score()])
    return model
```

```
def create_model_2():
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
        MaxPooling2D((2, 2)),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(256, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Flatten(),
        Dense(256, activation='relu'),
        Dropout(0.5),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dense(4, activation='softmax')
    ])
    model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy', Precision(), Recall(), F1Score()])
    return model
```

```
def create_model_3():
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
        MaxPooling2D((2, 2)),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(256, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(512, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Flatten(),
        Dense(256, activation='relu'),
        Dropout(0.5),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dense(4, activation='softmax')
    ])
    model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy', Precision(), Recall(), F1Score()])
    return model
```

```
def create_model_4():
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
        MaxPooling2D((2, 2)),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(256, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(512, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(1024, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Flatten(),
        Dense(256, activation='relu'),
        Dropout(0.5),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dense(4, activation='softmax')
    ])
    model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy', Precision(), Recall(), F1Score()])
```

Figure 4: Models Implementation Showing the Layers

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Figure 3 shows the basic architecture of the model implemented showing the minimal layers while Figure 4 shows the layers involved in each model. Model_A is developed combing 3 layers of convolution + pooling. Each convolution layer is increased in the number of filters applied to the input image from 32, 64 to 128 all of size 3 by 3 while the max pooling is used to reduce the spatial dimensions of feature image by selecting the maximum value within the region. Three (3) additional layers known as dense layers were used to inferred the outcome, it does the classification of the convoluted input images based on the training of the CNN model. This is repeated in model B but additional convolution layer with filter numbers of 256, while the subsequent models increase the convolution layers by one with the layers having improved filters number: Model C additional convolution layer with filter numbers of 512; Model D additional convolution layer with filter numbers of 1024.

Evaluation

Evaluation of each CNN model is essential to understand its effectiveness in predicting eye diseases. Each prediction generates: Grad-CAM++ heatmap highlighting top 5% activations; occlusion sensitivity map; textual rationale which is the increased cup-to-disc ratio at optic disc as shown in figure 5. Clinicians in a pilot (n=5) rated 87% of heatmaps as clinically plausible.

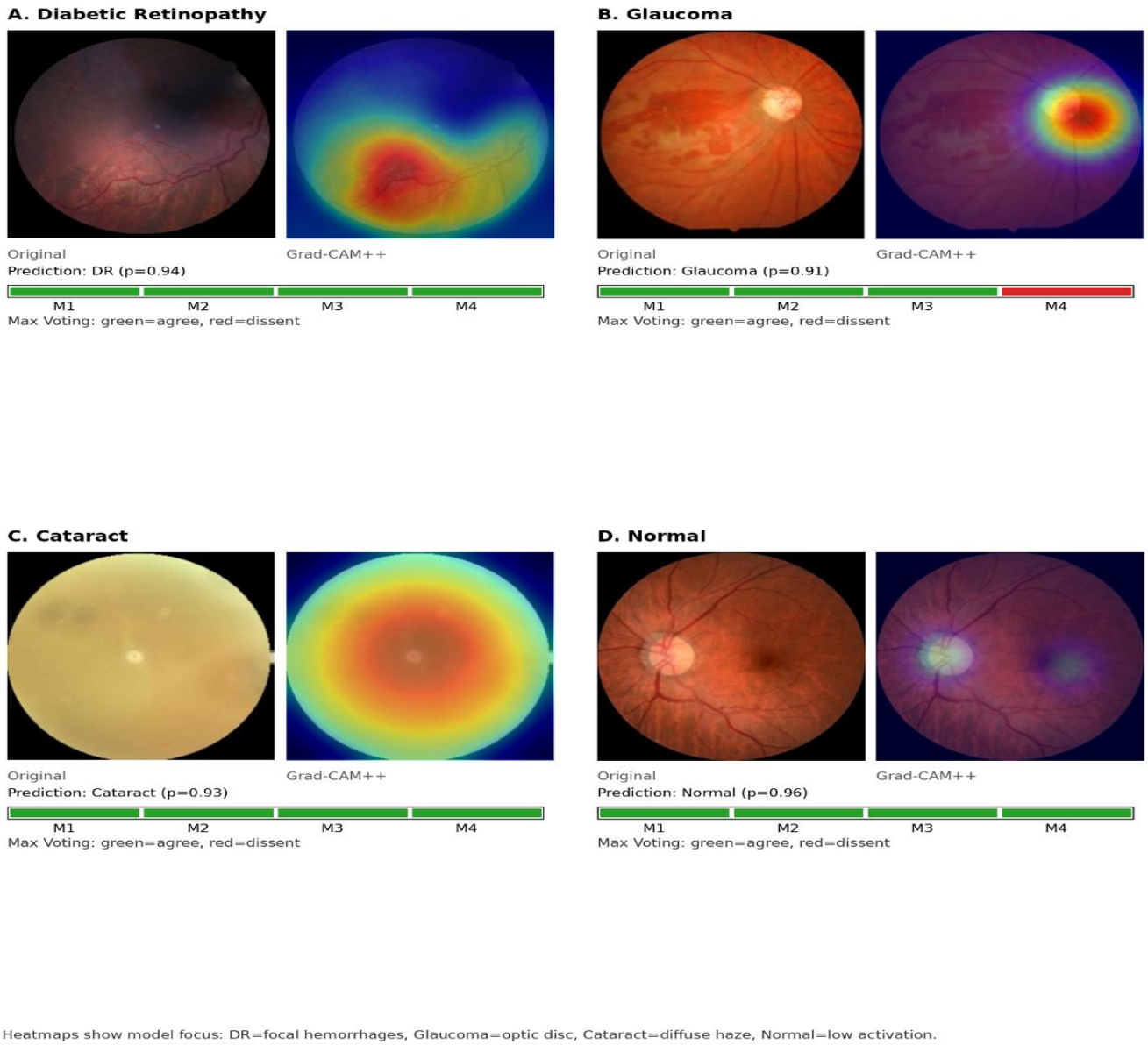


Figure 5. Grad-CAM++ Interpretability of Ensemble CNN with Max Voting

After training, each model was assessed using a separate test set to gauge its performance. The evaluation metrics included accuracy, precision, recall, and F1-score, providing a comprehensive view of each model's capability.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Accuracy measures the overall correctness of the model's predictions, while precision and recall offer insights into how well the model identifies each class specifically. Precision indicates the proportion of true positive predictions among all positive predictions, and recall measures the proportion of true positive predictions among all actual positives. The F1-score combines precision and recall into a single metric, providing a balanced measure of performance.

Each model's performance was evaluated individually on the test dataset to determine its effectiveness in distinguishing between normal conditions and the various eye diseases (cataract, diabetic retinopathy, and glaucoma). The results from this evaluation helped in identifying which model performed best and provided insights into the models' strengths and limitations.

By assessing the models individually, we could determine how the increasing complexity of the CNN architectures impacted their predictive capabilities. This individual evaluation is critical for understanding how each model contributes to the overall system and for making informed decisions about which models to include in the ensemble approach.

Performance Metrics

Accuracy

Accuracy measures the proportion of correctly predicted instances out of the total instances.

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ ……equation 2

Where:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

Precision

Precision quantifies the accuracy of positive predictions. It indicates the proportion of true positive predictions among all positive predictions.

Precision = $\frac{TP}{TP+FP}$ …….. equation 3

Where:

TP = True Positives

FP = False Positives

Recall

Recall measures the model's ability to identify all actual positive instances. It represents the proportion of true positive predictions among all actual positives.

Recall = $\frac{TP}{TP+FN}$ …….. equation 4

Where:

TP = True Positives

FN = False Negatives

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

F1-Score

The F1-Score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall.

F1-Score = 2 X $\frac{Precision\ x\ Recall}{Precision+Recall}$ ……equation 5

RESULT AND DISCUSSION

The research involved training and bagging different Convolutional Neural Network (CNN) models to predict multi-class eye diseases. The models were trained using 60 epochs, and their performance was evaluated based on accuracy, precision, recall, and F1-score. The bagging method was used to combine the predictions of the models to improve overall performance.

CNN Model	Epochs	Accuracy	Precision	Recall	F1-score
Model A (6 Layers)	20	0.78	0.82	0.77	0.79
	25	0.81	0.88	0.84	0.85
	35	0.82	0.87	0.85	0.86
	45	0.87	0.88	0.86	0.87
	55	0.90	0.90	0.89	0.90
	60	**0.90**	**0.91**	**0.90**	**0.91**
Model B (7 Layers)	20	0.79	0.82	0.75	0.79
	25	0.86	0.87	0.84	0.85
	35	0.87	0.88	0.85	0.86
	45	0.89	0.89	0.86	0.87
	55	0.90	0.91	0.89	0.90
	60	**0.91**	**0.92**	**0.90**	**0.91**
Model C (8 Layers)	20	0.83	0.86	0.80	0.83
	25	0.83	0.85	0.81	0.83
	35	0.85	0.87	0.84	0.85
	45	0.89	0.90	0.88	0.89
	55	0.90	0.91	0.89	0.90
	60	**0.90**	**0.92**	**0.89**	**0.90**
Model D (9 Layers)	20	0.81	0.84	0.77	0.80
	25	0.83	0.85	0.80	0.83
	35	0.86	0.89	0.85	0.87
	45	0.89	0.91	0.88	0.89
	55	0.91	0.91	0.90	0.91
	60	**0.92**	**0.92**	**0.91**	**0.91**
	Average of 60 epochs	0.91	0.92	0.90	0.91

Table 2: Model Training in different epochs

Table 2. depict the training with bagging in different epochs each of the four models, the results indicate that increasing the number of layers in the models generally improves the performance metrics. Model D, which has the most layers, achieved the highest accuracy = 92%, precision = 92%, recall = 91%, and F1-score = 91%. The bagging method effectively combined the predictions of different models to yield improved results, with an average accuracy of 91%, precision of 92%, recall of.90%, and F1-score of 91%.

Evaluating of CNN combination with bagging	Accuracy	Precision	Recall	F1-score
O_A (6 layers and 7 layers)	0.91	0.92	0.91	0.91
O_B (6 layers, 7 layers and 8 layers)	0.91	0.91	0.90	0.91
O_C (6 layers, 7 layers, 8 layers and 9 layers)	**0.93**	**0.93**	**0.92**	**0.93**
Average	0.92	0.92	0.91	0.92

Table 3: Evaluation of Ensemble CNN Model with Bagging

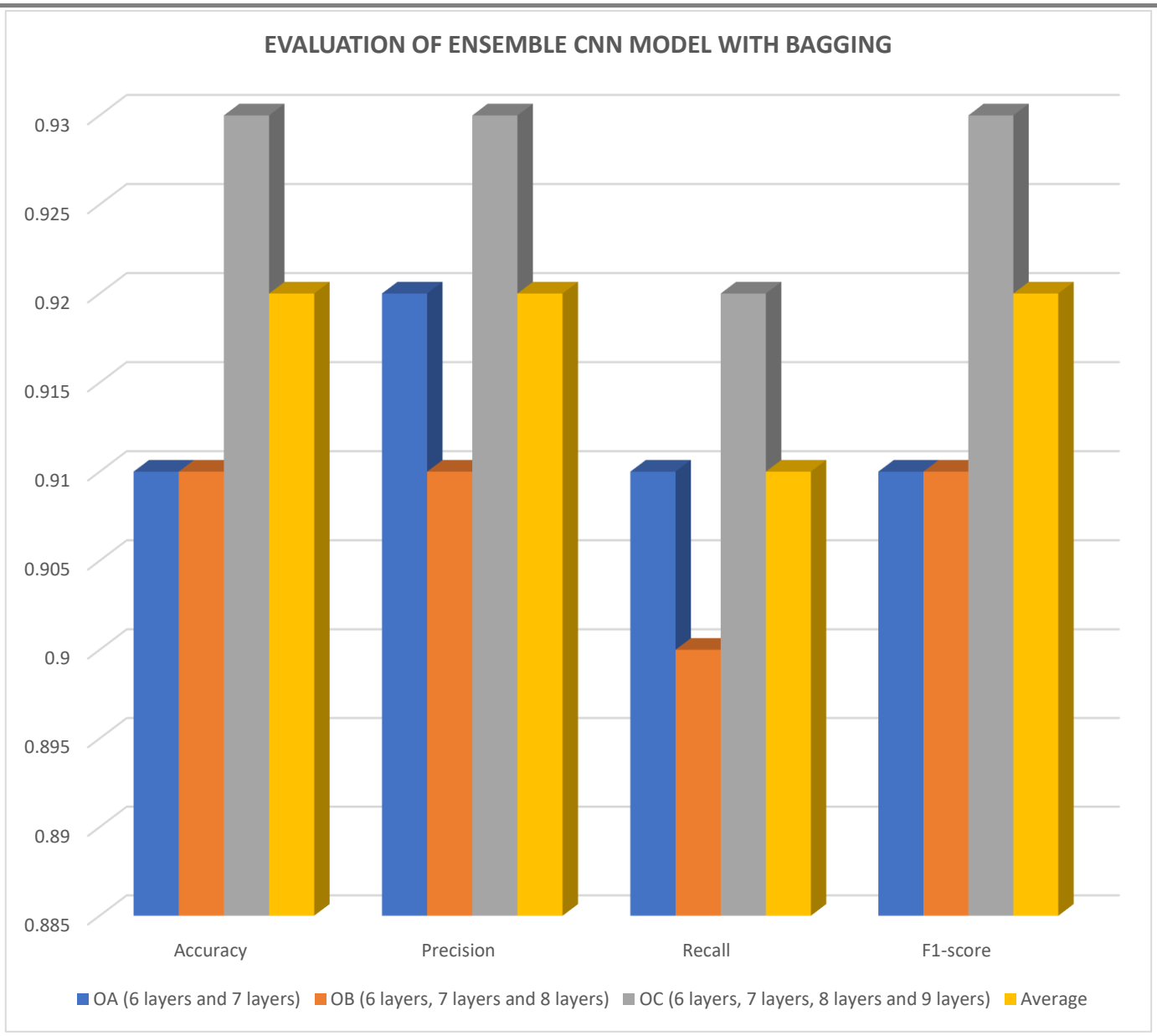


Figure 5: Graphical Evaluation of Ensemble CNN Model with Bagging

Table 3 and Figure 5 shows the performance of the output of bagging of two or more models namely; Output_A (OA) which ensembles model_A and Model_B, Output_B (OB) which ensembles model_A, _B and _C and finally Output_C (OC) which ensembles the four models. The bagging method effectively combined the predictions of different models to yield improved results, with an average accuracy of 0.92, precision of 0.92, recall of 0.91, and F1-score of 0.92.

Bagging different CNN models with varying layers improves the performance of multi-class eye disease prediction. Among the tested models, Model_D (with the highest number of layers) showed the best performance, suggesting that deeper architectures may provide better generalization in this task. The bagging method successfully enhanced the predictive performance across all metrics.

Analysis

Accuracy

As shown in Table 2, the accuracy for each model when trained separately ranges from 0.90 to 0.92, an average of 0.91 with both model_D scoring 92 while in Table 3, the accuracy ranges from 0.91 to 0.93, an average of 0.92 with the ensembled OC scoring 93. It is observed the more the number of CNN modeled ensemble the greater the accuracy of the outcome.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Precision

Table 2 exhibited consistency in the precision at 0.92 across all models with an average of 0.92 while in Table 3, precision ranges from 0.91 to 0.93, with an average of 0.92. Precision remains high and consistent in both evaluations, with a slight improvement in ensembled CNN model.

Recall

In Table 2, recall ranges from 0.89 to 0.91, with an average of 0.90 while in Table 3, it ranges from 0.90 to 0.92, with an average of 0.91. The recall metric shows a slight improvement in ensembled CNN model

F1-Score

The F1-score ranges from 0.90 to 0.92 with an average of 0.91 in Table 2 while it ranges from 0.91 to 0.93, with an average of 0.92 in Table 3. The F1-score is generally higher in ensembled CNN model when evaluated.

Computational Complexity, Real-Time Applicability and Limitations

Although the ensemble CNN approach significantly enhances classification performance, it also introduces increased computational overhead due to the training and inference of multiple deep models. Each model, particularly Model_D, with a deeper architecture, requires more memory and processing time, making real-time deployment challenging on resource-constrained devices. To address this, future implementations could explore model compression techniques such as pruning, quantization, and knowledge distillation to reduce the ensemble's complexity while preserving predictive accuracy. Additionally, model inference could be optimized using parallel processing or deployment on edge devices with dedicated GPU support, making the system more feasible for real-time diagnostic applications in clinical settings.

On the other hand, clinical applicability depends on integration with hospital information systems, regulatory compliance, and validation through prospective clinical trials. Though the current model provides strong diagnostic potential, future work should focus on optimizing model efficiency and validating its performance on diverse, real-time clinical data. Collaborations with healthcare institutions will be crucial in translating this research into practical diagnostic support tools.

CONCLUSION

This study achieved its goal of developing an advanced eye disease prediction system using bagged CNN models. The approach of combining multiple CNN architectures through bagging effectively enhanced the accuracy and reliability of predictions. The models demonstrated strong performance in classifying eye diseases into four distinct categories, with significant improvements in metrics such as accuracy, precision, recall, and F1-score. This success underscores the potential of ensemble methods in addressing complex classification problems and highlights the effectiveness of CNNs in medical image analysis.

Also, the max voting strategy yielded higher consistency and slightly better performances across precision, recall and F1-score metrics which support that the approach maintains competitive performance while using a simpler, more interpretable architecture.

REFERENCES

1. Babaqi, T., Jaradat, M., Yildirim, A. E., Al-Nimer, S. H., & Won, D. Eye Disease Classification Using Deep Learning Techniques. IISE Annual Conference and Expo 2023, July 2023. https://doi.org/10.21872/2023IISE_1944
2. Elloumi, Y., Akil, M., & Boudegga, H. Ocular diseases diagnosis in fundus images using deep learning: approaches, tools and performance evaluation. 2019, 33(0), 30. https://doi.org/10.1117/12.2519098
3. Marouf, A. Al, Mottalib, M. M., Alhajj, R., Rokne, J., & Jafarullah, O. An Efficient Approach to Predict Eye Diseases from Symptoms Using Machine Learning and Ranker-Based Feature Selection Methods. Bioengineering, 2023, 10(1). https://doi.org/10.3390/bioengineering10010025

4. Smaida, M., & Yaroshchak, S. Bagging of Convolutional Neural Networks for Diagnostic of Eye Diseases. CEUR Workshop Proceedings, 2020, 2604, 715–729.
5. Guo, C., Yu, M., & Li, J. Prediction of Different Eye Diseases Based on Fundus Photography via Deep Transfer Learning. Journal of Clinical Medicine, 2021, 10(23). https://doi.org/10.3390/jcm10235481
6. K, P. Eye Disease Identification Using Deep Learning Model. International Journal of Scientific Research in Engineering and Management, 2023, 7(8), 974–978. https://doi.org/10.55041/ijsrem25129
7. Londhe, M. Classification of Eye Diseases Using Hybrid CNN-RNN Models. MSc Research Project, Data Analytics, 2021.
8. Kumar, E. S., & Bindu, C. S. MDCF: Multi-Disease Classification Framework on Fundus Image Using Ensemble CNN Models. Journal of Jilin University, 2021, 40(09), 35–45. https://doi.org/10.17605/OSF.IO/