



Lightweight CNN Feature Extraction and PSO-Weighted Ensemble for Retinal OCT Classification

Neisa Hibatillah Alif¹, Anggun Dwi Rizkika², Feiticeira Zulkarnaen³, Nanik Suciati^{4*}
^{1,2,3,4}Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

*nanik@its.ac.id

Abstract

Retinal diseases are a primary cause of visual impairment, requiring accurate and efficient automated classification. This study proposes a multi-class classification approach for detecting retinal disease from Optical Coherence Tomography (OCT) images using the OCT-C8 dataset, which consists of 24,000 images across eight balanced classes. Pretrained lightweight convolutional neural networks (EfficientNet-B0, ShuffleNetV2, and RegNetY-400MF) are used as feature extractors to leverage transfer learning while reducing computational cost. Instead of end-to-end deep learning, extracted features are classified using traditional machine learning models (KNN, RF, SVM, and MLP), which are more efficient for moderate-sized datasets. To improve robustness, a weighted ensemble is applied, where classifier contributions are optimized using Particle Swarm Optimization (PSO). Experimental results show that the proposed method achieves a classification accuracy of 97.35%, outperforming conventional hard and soft voting methods, while maintaining a balance between computational efficiency and performance with potential for practical deployment.

Keywords: ensemble learning; lightweight CNN; OCT; PSO; retinal diseases classification

How to Cite: N. H. Alif, A. D. Rizkika, F. Zulkarnaen, and N. Suciati, "Lightweight CNN Feature Extraction and PSO-Weighted Ensemble for Retinal OCT Classification", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 3, pp. 597 - 608, Jun. 2026.

Permalink/DOI: <https://doi.org/10.29207/resti.v10i3.7451>

Received: January 7, 2026

Accepted: May 23, 2026

Available Online: June 8, 2026

This is an open-access article under the [CC BY 4.0 License](#)

Published by [Ikatan Ahli Informatika Indonesia](#)

1. Introduction

As the world's population ages, retinal disorders are becoming increasingly prevalent and remain one of the leading causes of blindness and visual impairment [1]. Early detection is essential because many retinal diseases, such as age-related macular degeneration (AMD), diabetic macular edema (DME), and choroidal neovascularization (CNV), are progressive and often asymptomatic in their early stages. Because Optical Coherence Tomography (OCT) allows for non-invasive and high-resolution imaging of retinal structures, it has become a standard diagnostic technique [2], [3].

However, manual interpretation of OCT images requires significant clinical expertise, is time-consuming, and is prone to inter-observer variability, which may lead to inconsistent diagnosis [4]. To address these limitations, deep learning techniques have been extensively applied in medical image analysis, including OCT imaging. By automatically learning

feature representations from image data, deep learning models can improve classification performance [5].

Recent studies on the Retinal OCT-C8 dataset have reported high classification accuracy. For example, Laouarem et al. [6] proposed HTC-Retina, which combines a Vision Transformer encoder with a Residual Depthwise-Pointwise ConvNet and achieved up to 97% accuracy. Similarly, Novely et al. [7] enhanced several pre-trained convolutional neural networks (CNNs) by incorporating attention mechanisms such as CBAM and skip connections, where DenseNet121 CBAM-Skip achieved 96.28% accuracy.

Other studies show a similar trend. Aykat and Senan [8] evaluated multiple deep architectures, including DenseNet121, EfficientNet, and InceptionV3, and identified DenseNet121 as the best-performing model. Qi et al. [9] proposed MSLI-Net based on ResNet50 with multi-scale and attention modules, achieving 96.72% accuracy, while Yang et al. [10] introduced

HyReti-Net, a hybrid model combining ResNet50 and Vision Transformer through feature fusion. Although these approaches achieve strong performance on the OCT-C8 dataset, many rely on large backbone networks, additional modules, or other architectural enhancements.

To improve robustness and decision quality, multimodal and ensemble methods have been explored. Zedadra et al. [11] developed VisionTrack by integrating OCT images with clinical data, demonstrating that multimodal inputs can enhance prediction performance. From the ensemble perspective, Pin et al. [12] proposed a hybrid approach combining multiple transfer learning models with classical classifiers, where the final decision is obtained through weighted soft voting, highlighting the advantages of combining heterogeneous models.

Nevertheless, two key gaps remain. First, many OCT classification methods still rely on computationally intensive architectures, limiting their practical deployment in low-resource settings [6], [8]. Second, although ensemble learning can improve robustness, most approaches use fixed or heuristic weighting strategies, which do not fully exploit the strengths of individual classifiers. While Particle Swarm Optimization (PSO) has shown effectiveness in optimizing model parameters and ensemble weights in

other domains, its application for adaptive weighting in OCT classification remains limited [13], [14].

To address these challenges, this study proposes an efficient framework for retinal disease classification using OCT images. The proposed method utilizes three lightweight CNN architectures as feature extractors: EfficientNet-B0, ShuffleNetV2, and RegNetY-400MF. The features extracted by these CNNs are then used as inputs to multiple machine learning classifiers, including K-Nearest Neighbor (KNN), Random Forest (RF), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP). These classifiers are combined through a weighted ensemble approach, where the ensemble weights are optimized using PSO to improve classification accuracy and stability. This approach aims to achieve a balance between high performance and low computational complexity while addressing the identified research gaps.

2. Methods

The research workflow consists of dataset preparation, preprocessing, transfer learning for feature extraction, classifier initialization, PSO with ensemble weight optimization, and performance evaluation. Figure 1 presents a flowchart of the proposed method.

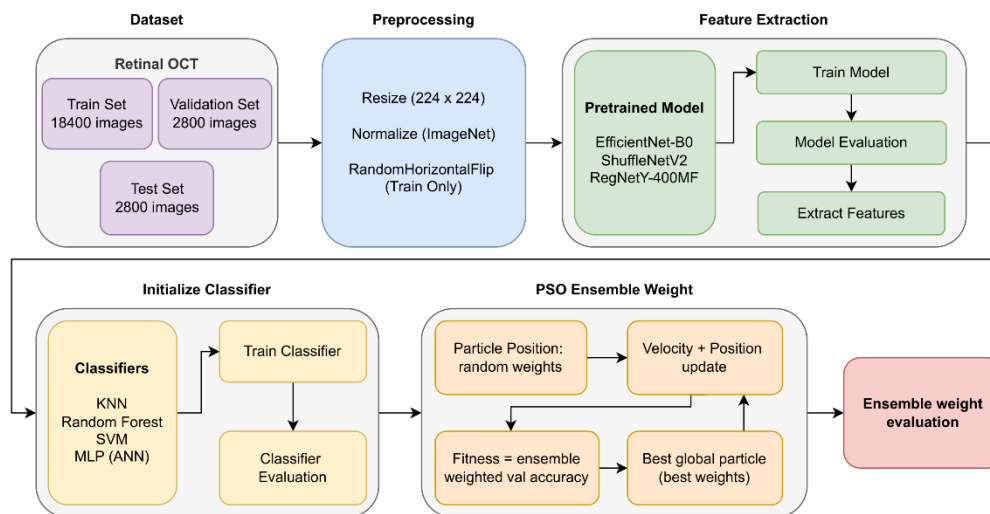


Figure 1. Proposed PSO-Weighted Ensemble Pipeline for Retinal OCT Classification

The retinal OCT-C8 dataset, which is openly accessible on Kaggle, was utilized in this study [15]. This dataset has a total of 24000 images across eight balanced classes that represent various retinal conditions: AMD (age-related macular degeneration), CNV (choroidal neovascularization), CSR (central serous retinopathy), DME (diabetic macular edema), DR (diabetic retinopathy), drusen (yellow deposits under the retina), MH (macular hole), and normal (healthy eyes with no abnormalities). These images are divided into 18400 training images (2300 per class), 2800 validation images, and 2800 test images (350 per class). The training set was utilized for model learning, while the

validation set was used to evaluate performance and guide the optimization process. The testing set was solely used to evaluate model performance and was not used as input during model training and optimization.

The retinal image dataset was preprocessed to improve quality and consistency. The images were scaled to 224×224 pixels. For training set, RandomHorizontalFlip was used as a lightweight augmentation technique for better model generalization. All images were normalized using the ImageNet mean and standard deviation to match the pretrained CNN backbone [16].

In the proposed hybrid framework, CNN is an important feature extractor. CNNs are capable to learn complex visual patterns that are essential for distinguishing classes. Feature extraction was performed using three pre-trained lightweight models: EfficientNet-B0, ShuffleNetV2, and RegNetY-400MF. These three models were selected to ensure fairness of the experiment across computational scales with similar ranges of parameters and FLOPs.

EfficientNet-B0 is a lightweight CNN that uses a compound scaling technique based on a simple set of coefficients to uniformly scale the network's depth, width, and resolution [17]. This strategy allows the model to reach high performance while maintaining low computational costs, which makes it ideal for computationally constrained environments [18], [19]. Despite its small size, EfficientNet-B0 maintains its robust performance and has become one of the most popular baseline models for feature extractor.

ShuffleNetV2 is a CNN architecture specifically designed for fast and hardware-friendly computation. This approach focuses on reducing memory access costs and unnecessary operations. Its key idea, channel shuffle, enables effective information mixing between channel groups with very low computational overhead. Due to its fast and hardware-friendly design, ShuffleNetV2 is a suitable backbone for lightweight feature extraction [20].

RegNetY-400MF is part of the RegNet family, which was created based on the idea that high-performing CNNs follow simple and regular patterns. By modeling these patterns, the authors developed a variety of efficient architectures that scale smoothly across different sizes. The “Y” variants, including RegNetY-400MF, add squeeze-and-excitation (SE) blocks to improve channel attention. Its combination of compact design and high performance makes RegNetY-400MF an effective feature extractor in the proposed framework [21].

The proposed ensemble method consists of four different machine learning classifiers: KNN, RF, SVM, and MLP. In order to maximize its own accuracy on the extracted features, each classifier was initialized and trained independently. Then, the trained models are evaluated on the validation and test sets to determine baseline performance before being combined into an ensemble.

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the collective behavior observed in nature, such as bird flocks and fish schools [22]. In PSO, each particle represents a candidate solution and iteratively updates its position according to its personal best solution and the global best solution found by the swarm. This cooperative search strategy enables PSO to efficiently balance exploration and exploitation of the search space [23], [24]. The mathematical process of updating the velocity and position of each particle within the PSO is shown in Equations 1 and 2.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest_i - x_i^t) + c_2 r_2 (gbest - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

In these equations, x_i^{t+1} and v_i^{t+1} denote the position and velocity of particle i at iteration $t + 1$, respectively. The variables $pbest_i$ and $gbest$ represent the personal best position of particle i and the global best position of the swarm. The r_1 and r_2 are random coefficient within the interval $[0, 1]$. The parameter ω denotes the inertia weight, which controls the momentum influence of the previous velocity, while the c_1 and c_2 are the cognitive and social acceleration coefficients that determine the influence of the personal and global best positions on the particle's movement.

To mitigate the weaknesses of each classifier and achieve strong generalization, the weighted ensemble technique was implemented. PSO was used to optimize the weight vector $\mathbf{w} = [w_1, w_2, \dots, w_N]$, where N is the number of base classifiers [14]. Each particle position represented a candidate weight vector and the fitness of each particle was evaluated using the ensemble's validation accuracy.

At each iteration, the velocity and position of the particle were updated using the inertia, cognitive, and social components. The resulting particle position vector is transformed using the softmax function to make sure that all weights were positive and summed to one ($\sum w_i = 1$). The global best particle represented the optimal combination of classifier weights during the optimization process.

The final ensemble prediction was calculated using a weighted soft voting method, where the PSO-optimized weights are applied to the probability outputs of individual classifiers [25]. The combined probability vector for the ensemble ($P_{ensemble}$) is calculated using Equation 3.

$$P_{ensemble} = \sum_{i=1}^N w_i P_i \quad (3)$$

where P_i is the probability vector output by classifier i . The maximum value of the combined vector is then used to get the final class prediction y , as shown in Equation 4.

$$y = \arg \max(P_{ensemble}) \quad (4)$$

The optimized weights from PSO determined the contribution of each classifier. This method allowed the ensemble to focus on high-performing classifiers while reducing the influence of weaker classifiers. The performance of the final PSO-based ensemble was evaluated on the test set using evaluation metrics.

Subsequently, the evaluation metrics used to assess the performance of the proposed framework are presented and discussed. The overall performance was evaluated using accuracy, precision, recall, and F1-score. The equations for each metric are shown in Equations 5-8 [26].

TP (True Positives): Number of positive instances correctly classified as positive

TN (True Negatives): Number of negative instances correctly classified as negative

FP (False Positive): Number of negative instances incorrectly classified as positive

FN (False Negative): Number of positive instances incorrectly classified as negative

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

$$F1 - \text{score} = \frac{2 (\text{Precision} \cdot \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (8)$$

In addition to quantitative metrics, confusion matrices are also used to provide detailed analysis of classification results. Confusion matrices summarize the number of correct and incorrect predictions for each class by comparing the predicted labels with the actual labels [27]. This allows for the examination of classification errors per class. The label order for the eight classes (0 to 7) across all confusion matrices in this study is as follows: (0) AMD, (1) CNV, (2) CSR, (3) DME, (4) DR, (5) drusen, (6) MH, and (7) normal.

3. Results and Discussions

In this section, the evaluation results from each step are presented and discussed. To ensure the robustness of the results, each pipeline was repeated five times using different random seeds (0, 21, 42, 123, and 2024). The

reported performances correspond to the average and standard deviation across the five runs.

All experiments were conducted using the Kaggle computational environment using an NVIDIA Tesla P100-PCIE GPU with 16 GB of VRAM and 34 GB of RAM. The models were developed using PyTorch 2.9.0 with CUDA 12.6 support.

3.1 Transfer Learning Performance

In this step, the experiment focused on training the CNN models for image feature extraction. The models were trained for 5 epochs with 32 batch sizes, using the Adam optimizer with a learning rate of 0.0001.

The results of the backbone performance testing are shown in Table 1. EfficientNet-B0 achieved the highest average accuracy with a score of 96.96%, followed by RegNetY-400MF with 96.72% and ShuffleNetV2 with 96.11%. These scores illustrate that all three lightweight architectures are capable of providing good classification results. However, EfficientNet-B0 is slightly more effective at extracting features relevant to retinal images. These results are considered very good, even though the models were only trained on relatively low epochs. This confirms that the pre-trained models were quite effective in producing high-quality feature vectors necessary for the next classifier training phase.

Figure 2 illustrates the confusion matrices for the classification results obtained from the transfer learning models using the best-performing run of each backbone. For EfficientNet-B0, the model achieves perfect recognition for AMD, CSR, DR, and MH, indicating that these classes contain highly distinctive visual patterns.

Table 1. CNN Backbones Accuracy Performance Across 5 Runs (%)

Backbone	Run 1	Run 2	Run 3	Run 4	Run 5	Average	SD
EfficientNet-B0	97.00	97.14	96.96	96.53	97.17	96.96	0.26
ShuffleNetV2	96.29	95.92	96.54	95.86	95.92	96.11	0.30
RegNetY-400MF	96.75	96.89	96.75	96.93	96.29	96.72	0.26

Misclassifications occur in CNV, DME, drusen, and normal classes, which shows higher interclass similarity in OCT images. In ShuffleNetV2, a similar pattern is observed. Although the model maintains strong performance for AMD, CSR, DR, and MH, increased confusion is visible in CNV, drusen, and normal, suggesting that the lightweight architecture may struggle to fully separate texture variations in certain disease patterns. Meanwhile, RegNetY-400MF shows improved class separation, particularly for DME, but overall class confusion remains, indicating that feature extraction alone is insufficient to resolve overlapping patterns among clinically related classes.

The strong performance of all backbone models, particularly EfficientNet-B0, indicates that the use of pretrained CNNs provides highly discriminative feature representations. This demonstrates that transfer learning is effective for OCT image classification, even with lightweight architectures.

3.2 Machine Learning Classifier Performance

CNN-generated deep features were used to train and evaluate four classifiers: KNN, RF, SVM, and MLP. To ensure fairness and comparability between different CNN models, this study used a set of general hyperparameters for all classifiers across all CNN models, as shown in Table 2.

The performance of four classifiers is evaluated based on features extracted from three lightweight backbones. These results are summarized in Table 3. For the EfficientNet-B0 features representation, the highest average accuracy was achieved by KNN classifier with 97.29%, followed by SVM and RF, while the lowest accuracy was the MLP with 97.13%. For the ShuffleNetV2 feature representation, the highest average accuracy was achieved by SVM with 96.09%, followed closely by MLP and RF, while the lowest was KNN classifier with 96.03%. Meanwhile, for the RegNetY-400MF feature representation, RF achieved

the best average accuracy at 96.94%, followed by SVM and MLP, while the lowest was the KNN classifier with 96.81% accuracy.

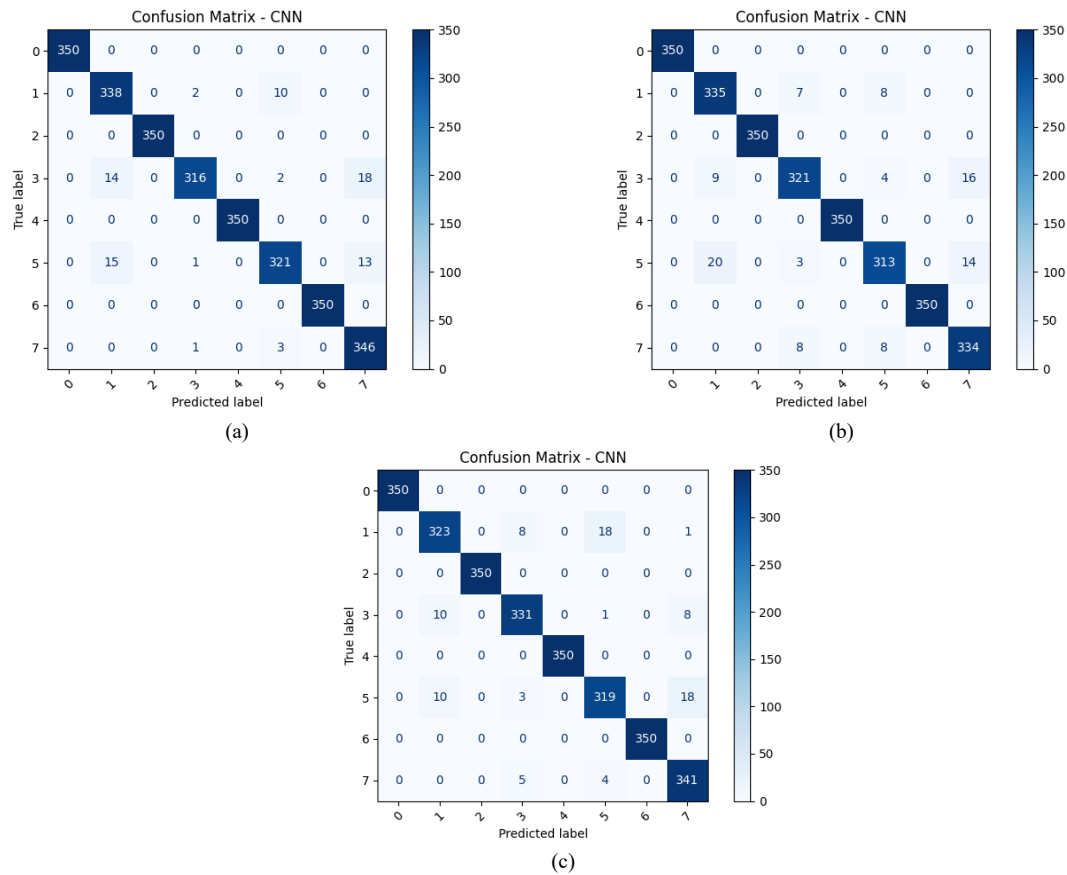


Figure 1. The Best Confusion Matrices Performance of (a) EfficientNet-B0 (Seed 2024), (b) ShuffleNetV2 (Seed 42), and (c) RegNetY-400MF (Seed 123)

Table 2. Hyperparameters of Different Classifiers

Classifier	Hyperparameter
KNN	$n_{neighbors} = 5$ $metric = 'minkowski'$
RF	$n_{estimators} = 100$ $criterion = 'gini'$ $C = 1$
SVM	$kernel = 'rbf'$ $probability = True$
MLP	$hidden\ layer\ sizes = (256,)$ $max\ iter = 500$

Figure 3 presents the confusion matrices obtained by combining the best CNN model, EfficientNet-B0's

feature representation, with four classical ML classifiers at their best performance (all from seed 2024). For all four classifiers, AMD, CSR, DR, and MH remain perfectly classified. However, noticeable confusion still occurs among CNV, DME, drusen, and normal classes. This result shows that while traditional classifiers can improve class separation, their performance is still limited by the quality of the extracted features. The limitation is no longer primarily in the classifier itself, but in the degree of feature separability between visually similar classes.

Table 3. CNN Feature Extractor and Individual Classifier Accuracy Performance Across 5 Runs (%)

Backbone	Classifier	Run 1	Run 2	Run 3	Run 4	Run 5	Average	SD
EfficientNet-B0	KNN	97.43	97.14	97.32	97.00	97.54	97.29	0.22
	RF	97.18	97.04	97.18	97.00	97.54	97.19	0.21
	SVM	97.29	97.18	97.29	96.93	97.54	97.25	0.22
	MLP	97.00	97.21	97.11	96.93	97.39	97.13	0.18
ShuffleNetV2	KNN	96.36	95.89	96.11	95.82	95.96	96.03	0.21
	RF	96.25	96.04	96.11	95.86	96.04	96.06	0.14
	SVM	96.25	95.96	96.18	96.25	95.79	96.09	0.20
	MLP	96.32	95.86	96.11	96.11	96.00	96.08	0.17
RegNetY-400MF	KNN	96.64	96.93	97.00	96.89	96.61	96.81	0.18
	RF	97.11	96.89	97.00	97.11	96.61	96.94	0.21
	SVM	97.00	96.89	97.00	97.00	96.46	96.87	0.23
	MLP	96.64	96.61	97.11	97.11	96.61	96.82	0.27

The results show that all combinations of backbones and classifiers can provide stable classification performance on the Retinal OCT-C8 dataset. The combination of EfficientNet-B0 with KNN provides the best overall results, achieving the highest accuracy

while using a lightweight backbone for feature extraction. These results show that the choice of combination between the backbone and the classifier affects the classification results.

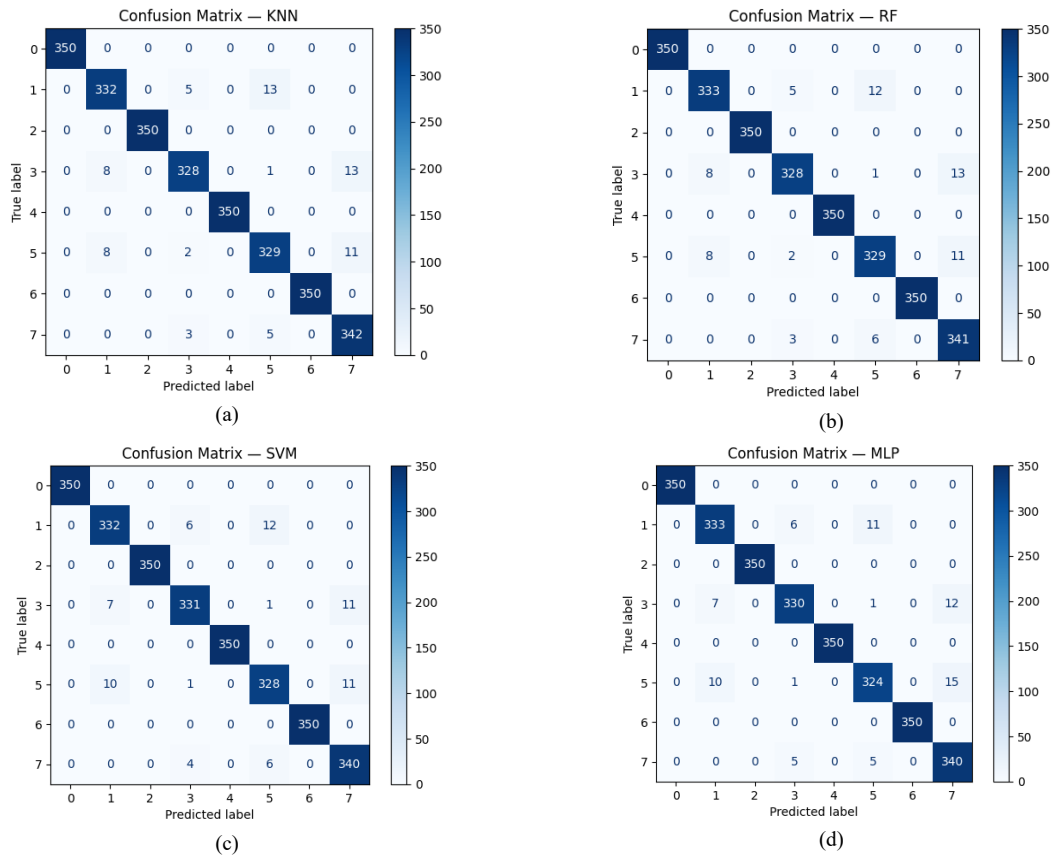


Figure 2. The Best Confusion Matrices using EfficientNet-B0 Features with (a) KNN, (b) RF, (c) SVM, and (d) MLP Classifier, All from Seed 2024

3.3 PSO-Weighted Ensemble Performance

The ensemble experiments combined CNN feature extraction, traditional classifiers, and PSO for optimal ensemble weighting. PSO was used to determine the best classifier weights set that maximized validation accuracy. Standard hyperparameters are utilized to train PSO, as shown in Table 4 [28], [29].

Table 3. Hyperparameters of PSO

Algorithm	Hyperparameter
PSO	$n_{particles} = 100$
	$max\ iter = 50$
	$\omega = 0.9$
	$c_1, c_2 = 2$
	$r_1, r_2 = range[0, 1]$

Table 4. PSO-weighted Ensemble Accuracy Performance Across 5 Runs (%)

Backbone	Run 1	Run 2	Run 3	Run 4	Run 5	Average	SD
EfficientNet-B0	97.43	97.39	97.32	97.04	97.57	97.35	0.20
ShuffleNetV2	96.39	96.00	96.18	96.18	96.64	96.28	0.24
RegNetY-400MF	97.11	97.04	97.14	97.14	96.75	97.04	0.17

To further analyze the optimization behavior, the convergence of the PSO algorithm from the best ensemble model is illustrated in Figure 4. In general, the convergence curves show that all three models reached

As shown in Table 5, the PSO-based ensemble consistently achieved high classification accuracy across all experiments and backbones. EfficientNet-B0 achieved the highest performance, with an accuracy range of 97.04% to 97.57%. Similarly, RegNetY-400MF demonstrated stable performance, achieving accuracies of 96.75%–97.14%, while ShuffleNetV2 produced lower but still competitive results, ranging from 96.00% to 96.64%. The relatively small variation across experiments indicates that the PSO optimization process is stable and not highly sensitive to initialization. However, choosing a good CNN backbone still affects the final results.

a stable solution well before the maximum limit of 50 iterations, as the optimization process was terminated when there was no improvement in accuracy exceeding the threshold (0.0001) for 10 consecutive iterations. In

EfficientNet-B0, accuracy increased around the 1st and 5th iteration before stabilizing until the end. For ShuffleNetV2, an increase in accuracy was observed at the beginning, with no further change in the subsequent iterations. Meanwhile, RegNetY-400MF showed two noticeable increases in accuracy, one at the beginning and another around the 7th iteration. This pattern indicates that PSO mainly searched for the optimal solution in the early iterations, while subsequent iterations maintained the best solution already found.

In addition to performance, the distribution of ensemble weights provides insight into how the PSO algorithm combines different classifiers. Table 6 presents the average and standard deviation of each classifier's weights across the various tests. For EfficientNet-B0, KNN received the highest average weight (0.42 ± 0.32), followed by SVM (0.28 ± 0.17), while RF and MLP contributed less. Similarly, for ShuffleNetV2, SVM obtained the highest average weight (0.31 ± 0.26), although the distribution remained relatively balanced across classifiers. For RegNetY-400MF, RF showed the highest average contribution (0.39 ± 0.45), while MLP received the lowest weight.

These results demonstrate a general alignment between individual classifier performance and the weights assigned by PSO. For each backbone, the classifier with the highest accuracy tends to receive the largest average weight within the ensemble. This result indicates that PSO is able to identify and prioritize stronger classifiers during the optimization process.

However, the relatively large standard deviation values as shown in Table 6 indicate that this relationship is not entirely deterministic. Rather than consistently assigning a single best classifier a dominant weight, PSO explores multiple weight configurations that yield comparable performance. In some experiments, a single classifier dominated the ensemble weight, while in others, the weights were more evenly distributed. This behavior suggests that PSO not only considers individual classifier accuracy but also leverages the strengths between classifiers to improve overall performance. Furthermore, despite the variability in weight allocation, the overall classification performance remains stable, as shown in Table 5. This indicates that the ensemble optimization produced several near-optimal solutions and that PSO reliably identified them.

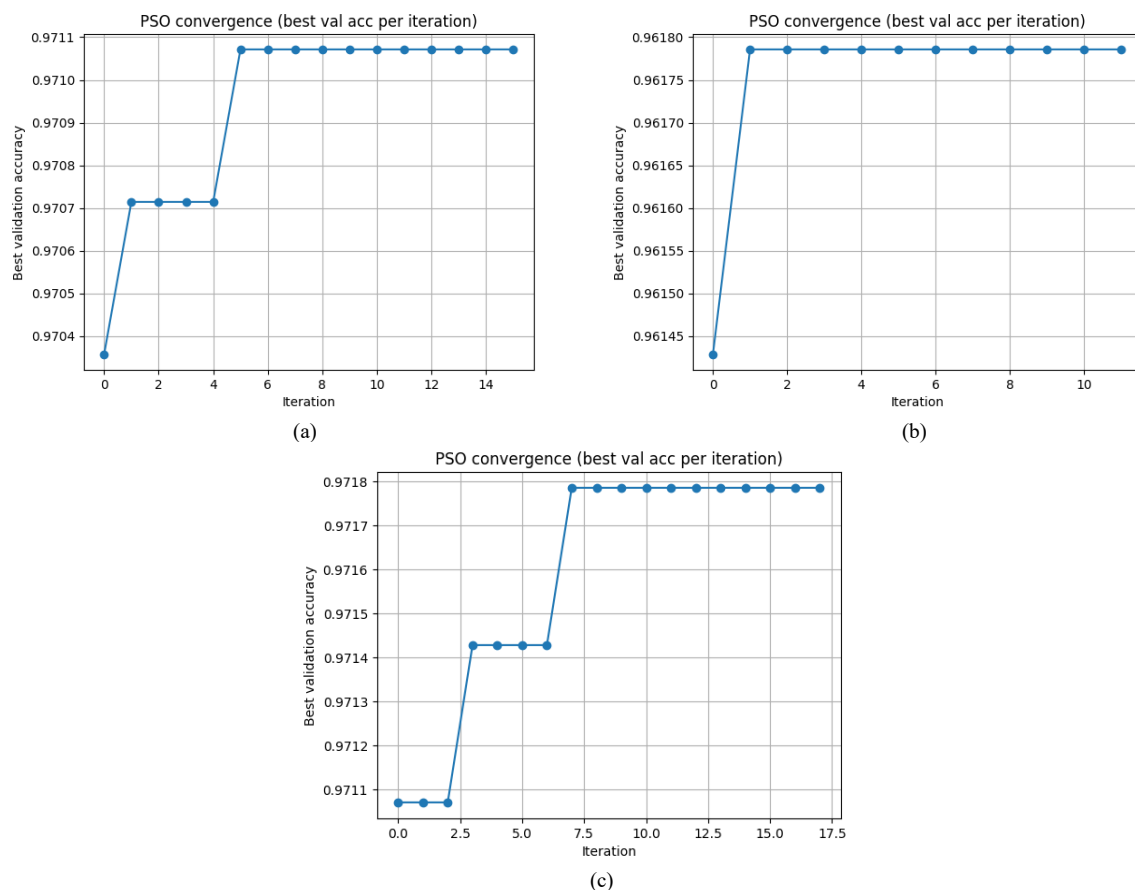


Figure 4. PSO Convergence Curves from the Best PSO-weighted Ensemble Performance for (a) EfficientNet-B0 (Seed 2024), (b) ShuffleNetV2 (Seed 2024), and (c) RegNetY-400MF (Seed 42)

Table 5. Normalized Ensemble Weights for Different Classifier Across 5 Runs (Average $\pm$ SD)

Backbone	KNN	RF	SVM	MLP
EfficientNet-B0	0.42 ± 0.32	0.19 ± 0.16	0.28 ± 0.17	0.11 ± 0.12
ShuffleNetV2	0.27 ± 0.19	0.21 ± 0.20	0.31 ± 0.26	0.22 ± 0.26
RegNetY-400MF	0.25 ± 0.26	0.39 ± 0.45	0.27 ± 0.30	0.09 ± 0.06

Figure 5 illustrates the confusion matrix of the best PSO-weighted ensemble of each feature representation combined with multiple ML classifiers. Compared to the standalone CNN and the individual CNN-ML classifier combinations, the ensemble demonstrates a stronger diagonal dominance and a more balanced distribution of predictions across classes. This improvement indicates that PSO effectively combines complementary information from multiple classifiers, thereby improving robustness.

However, despite this improvement, confusion between CNV, DME, drusen, and normal classes persists. This suggests that the limitations are not solely due to

ensemble performance but also reflect feature overlap and clinical similarity between the classes. The results indicate that while ensemble optimization improves overall stability and separability, it cannot completely eliminate ambiguity in more detailed medical classification tasks.

Overall, these results confirm that combining multiple classifiers can improve the robustness and performance of the classification system. While these findings demonstrate that PSO produces reliable results, it is also important to evaluate the proposed method against conventional ensemble strategies, as discussed in the next section.

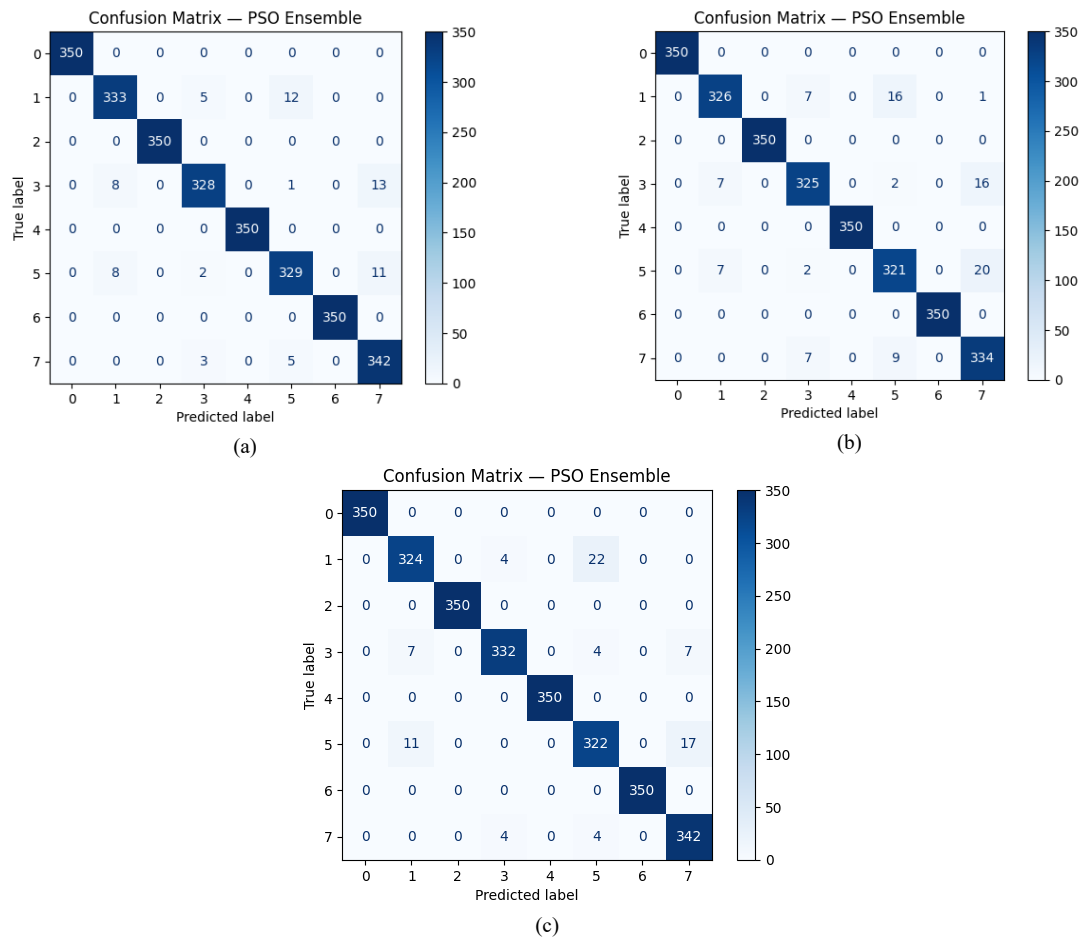


Figure 5. The Best Confusion Matrices Performance of PSO-weighted Ensemble using (a) EfficientNet-B0 (Seed 2024), (b) ShuffleNetV2 (Seed 2024), and (c) RegNetY-400MF (Seed 42) Feature Extractor

3.4 Comparison with Other Voting Ensemble Method

To further evaluate the effectiveness of the proposed method, its performance was compared with conventional hard and soft voting methods using the same set of CNN features and ML classifiers. Hard voting or majority voting rule treats all classifiers as equally important, as the final class label is determined by the majority vote of individual classifier predictions [30]. On the other hand, soft voting combines the class probability outputs of each classifier using a function (commonly an average function) and selects the class with the highest value from the function applied to the probabilities [31]. Both approaches rely on uniform

weighting and do not adapt to variations in each classifier's performance.

As shown in Table 7, the proposed method achieved the highest performance compared to hard and soft voting across all CNN backbones. For EfficientNet-B0, the PSO-weighted method achieved 97.35% accuracy, 97.37% precision, and 97.35% recall, outperforming all backbones and both conventional voting methods.

A similar trend was observed for both ShuffleNetV2 and RegNetY-400MF backbones, with the PSO-based ensemble yielding the highest average metrics among the evaluated methods. These results indicate that assigning optimized weights to classifiers provides an

advantage over conventional voting methods, which assume equal or fixed contributions.

To determine whether these improvements are statistically significant, a paired t-test was conducted using the F1-scores from the five runs, as the F1-score provides a more balanced measure. The results from Table 8 show that the improvement of the PSO-

weighted ensemble method is statistically significant for EfficientNet-B0 ($p < 0.05$) and RegNetY-400MF ($p < 0.05$ when compared to hard voting). However, no statistically significant difference was found for ShuffleNetV2, indicating that the effectiveness of PSO-based weighting may depend on the underlying backbone architecture.

Table 6. Performance Comparison of Ensemble Methods Across 5 Runs (Average $\pm$ SD, in terms of %)

Backbone	Method	Accuracy	Precision	Recall	F1-score
EfficientNet-B0	Hard Voting	97.25 $\pm$ 0.19	97.26 $\pm$ 0.20	97.25 $\pm$ 0.19	97.25 $\pm$ 0.18
	Soft Voting	97.28 $\pm$ 0.21	97.30 $\pm$ 0.20	97.28 $\pm$ 0.21	97.28 $\pm$ 0.21
	Proposed Method	97.35 $\pm$ 0.20	97.37 $\pm$ 0.20	97.35 $\pm$ 0.20	97.35 $\pm$ 0.20
ShuffleNetV2	Hard Voting	96.10 $\pm$ 0.16	96.12 $\pm$ 0.15	96.10 $\pm$ 0.16	96.10 $\pm$ 0.16
	Soft Voting	96.04 $\pm$ 0.19	96.07 $\pm$ 0.18	96.04 $\pm$ 0.19	96.04 $\pm$ 0.19
	Proposed Method	96.28 $\pm$ 0.24	96.31 $\pm$ 0.25	96.28 $\pm$ 0.25	96.28 $\pm$ 0.25
RegNetY-400MF	Hard Voting	96.90 $\pm$ 0.14	96.91 $\pm$ 0.14	96.90 $\pm$ 0.14	96.90 $\pm$ 0.14
	Soft Voting	96.87 $\pm$ 0.16	96.88 $\pm$ 0.16	96.87 $\pm$ 0.16	96.86 $\pm$ 0.16
	Proposed Method	97.04 $\pm$ 0.17	97.05 $\pm$ 0.17	97.04 $\pm$ 0.17	97.03 $\pm$ 0.17

Table 7. Statistical Significance (Paired t-Test Results) of the F1-score Between PSO-weighted Ensemble and Conventional Voting Methods

Backbone	Proposed Method vs Hard Voting	Proposed Method vs Soft Voting
EfficientNet-B0	0.036	0.005
ShuffleNetV2	0.201	0.117
RegNetY-400MF	0.019	0.079

Figure 6 presents the confusion matrices of the ensemble model using EfficientNet-B0 features with hard voting and soft voting. Although the differences between the PSO-weighted and conventional voting methods are relatively small, the confusion matrices reveal consistent patterns in class-level performance. As shown in Figures 5 and 6, the PSO-based method

shows a more stable separation between clinically similar features in the CNV, drusen, and normal classes compared to hard and soft voting. In contrast, the conventional voting method performs slightly better in distinguishing the DME class. This suggests that each approach exhibits different strengths depending on the type of interclass similarity.

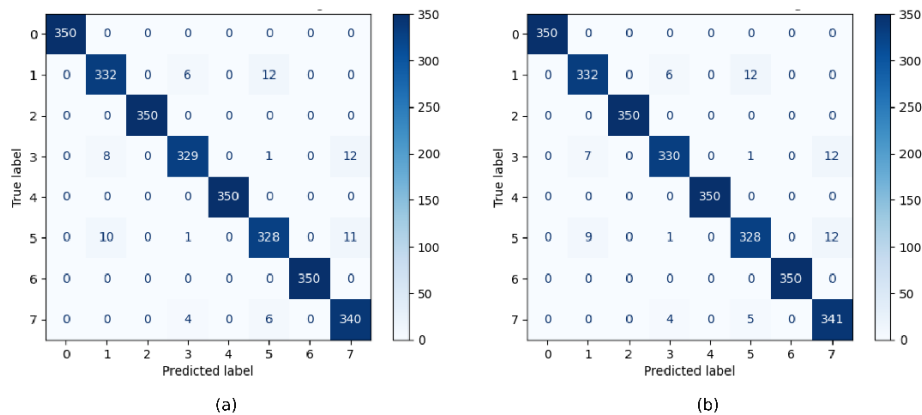


Figure 6. Confusion Matrices of The Best Ensemble Model using EfficientNet-B0 Features: (a) Hard Voting and (b) Soft Voting, All from Seed 2024

These observed differences can be explained by the underlying mechanisms of the ensemble methods. The superior confusion matrix of the PSO-weighted ensemble can be attributed to its ability to assign optimized weights to each classifier based on its contribution to the overall performance. Unlike conventional voting methods that treat all classifiers equally, PSO dynamically emphasizes models that are more effective for specific feature patterns. This adaptive weighting allows the ensemble to better capture differences between visually similar classes. This results in better class discrimination and a more balanced confusion matrix.

Overall, this study shows that PSO-based weighted voting is better at capturing the strengths of individual classifiers. The proposed method overcomes the limitations of fixed voting technique by learning classifier contributions adaptively and provides a more flexible and effective ensemble mechanism for CNN-based retinal disease classification.

3.5 Computational Efficiency Analysis

In addition to classification performance, the computational efficiency of the evaluated CNN backbones was analyzed to evaluate the practicality of the proposed method. The efficiency analysis considers

the number of model parameters, floating-point operations per second (FLOPs), model size, and inference time. These metrics can provide insight into the computational cost and deployment feasibility of the models, specifically for resource-constrained environments.

The measurements were obtained using PyTorch with an input resolution of 224×224 pixels. The number of parameters and FLOPs were computed directly from the network architecture. The model size was obtained by saving the trained model weights and calculating the file size in megabytes. Lastly, inference time was measured by averaging the prediction time per image during forward propagation.

The efficiency evaluation is summarized in Table 9. Among the evaluated architectures, ShuffleNetV2 requires the lowest computational cost with approximately 0.30 GFLOPs and 1.78 million parameters, making it the most lightweight model. In contrast, RegNetY-400MF has approximately 4.13 million parameters and 0.82 GFLOPs, resulting in higher computational complexity but achieving better performance than ShuffleNetV2. Meanwhile, EfficientNet-B0 offers a balance between computational cost and prediction accuracy, with 3.76 GFLOPs and 0.74 million parameters.

As shown in Table 9, all backbones have fewer than 5 million parameters and fewer than 1 GFLOPs. These lightweight sizes confirm their suitability for computationally efficient retinal OCT classification. In comparison, widely used architectures, such as VGG16 and ResNet-50, require approximately 138 million parameters with 15 GFLOPs and 25 million parameters with 4 GFLOPs respectively, significantly higher than the lightweight CNNs evaluated in this study [32], [33], [34].

In terms of runtime performance, the average inference time per image was 7.71 ms for EfficientNet-B0, 5.89 ms for ShuffleNetV2, and 9.49 ms for RegNetY-400MF. These results indicate that the lightweight CNN architectures can perform real-time inference efficiently on GPU hardware. Furthermore, the size analysis shows that the models require 14.64 MB, 6.97 MB, and 16.01 MB of storage, respectively.

Table 8. Computational Efficiency Comparison of CNN Backbones

Backbone	Params (M)	FLOPs (G)	Model Size (MB)	Average Inference Time (ms)
EfficientNet-B0	3.76	0.74	14.64	7.71
ShuffleNetV2	1.78	0.30	6.97	5.89
RegNetY-400MF	4.13	0.82	16.01	9.49

Overall, the computational efficiency results confirm that the evaluated lightweight CNN backbones provide a good balance between classification performance and computational cost. While EfficientNet-B0 achieves the highest classification performance, ShuffleNetV2 offers the lowest computational requirements, making

it a more practical choice for deployment in resource-limited environments.

3.6 Limitations and Future Work

Despite the strong performance achieved, this study has several limitations. First, the evaluation was conducted only on the retinal OCT-C8 dataset, which is balanced and collected under controlled conditions. Similar to many previous studies, this limits the ability to assess generalizability of the proposed method to more diverse, imbalanced, or real-world datasets. Therefore, future research should include cross-dataset evaluation to better understand the model's robustness under varying data distributions and clinical scenarios.

Second, although this study proposes a lightweight framework, computational evaluation was conducted only on GPU-based hardware. In contrast to studies that utilize computationally intensive architectures or consider deployment-related constraints [6], [8], this work does not include hardware-level validation. Therefore, the actual performance of the proposed approach in real-world clinical environments remains uncertain. Future research should investigate implementation on low-power or edge devices to validate its applicability in real-world settings.

Third, this study uses only two-dimensional OCT images. Unlike previous studies that incorporate multimodal data to improve diagnostic performance [11], the proposed approach relies solely on single-modality inputs. This limitation may reduce the model's ability to capture more complex or complementary clinical information. Future research could explore integrating volumetric OCT data or multimodal inputs to enhance performance.

Fourth, the PSO-based ensemble weighting is optimized using validation accuracy as the main objective. While this approach improves performance, it may introduce bias toward the validation set and does not explicitly consider robustness to noise or uncertainty, unlike some recent methods. Furthermore, the observed variation in ensemble weights across experiments suggests that the optimization process may converge to several near-optimal solutions. Future studies could investigate multi-objective optimization, regularization strategies, or alternative fitness functions to improve ensemble robustness and consistency.

Finally, similar to many deep learning approaches, the model operates as a black box. Although other studies have explored explainability techniques, interpretability is not addressed in this work. Future research should incorporate explainable AI methods to increase transparency.

4. Conclusions

This study proposes an efficient framework for multi-class retinal disease classification from OCT images by combining lightweight pretrained CNN models with traditional machine learning classifiers and a PSO-

based weighted ensemble strategy. The experimental results show that the proposed method consistently achieves high classification performance across multiple runs. In particular, the PSO-weighted ensemble using EfficientNet-B0 features achieves the best overall performance, with an accuracy of 97.35% along with strong precision, recall, and F1-score, outperforming conventional hard and soft voting methods. These results indicate that adaptive weighting through PSO is effective in capturing the strengths of individual classifiers and improving overall prediction performance.

In addition to accuracy, the proposed framework maintains relatively low computational complexity compared to conventional deep learning approaches, making it suitable for resource-limited settings. However, the study is limited to a single dataset and GPU-based evaluation and does not yet address real-world deployment or interpretability. Future work will focus on extending the proposed approach to more diverse datasets and real-world applications, deploying it on real clinical or edge devices, integrating multimodal data, and improving model robustness and transparency to enhance its practical applicability further.

Acknowledgements

This research was supported by the Institut Teknologi Sepuluh Nopember (ITS), under the Strategic Research Grant ITS 2026 with contract number 1660/PKS/ITS/2026.

References

- [1] S. R. Flaxman *et al.*, "Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis," *Lancet Global Health*, vol. 5, no. 12, pp. e1221–e1234, 2017, doi: [10.1016/S2214-109X\(17\)30393-5](https://doi.org/10.1016/S2214-109X(17)30393-5).
- [2] G. F. Attizzani, L. Patrício, and H. G. Bezerra, "Optical coherence tomography assessment of calcified plaque modification after rotational atherectomy," *Catheterization and Cardiovascular Interventions*, vol. 81, no. 3, pp. 558–561, 2013, doi: [10.1002/ccd.23385](https://doi.org/10.1002/ccd.23385).
- [3] R. Chopra, S. K. Wagner, and P. A. Keane, "Optical coherence tomography in the 2020s—outside the eye clinic," *Eye (Basingstoke)*, vol. 35, no. 1, pp. 236–243, 2021, doi: [10.1038/s41433-020-01263-6](https://doi.org/10.1038/s41433-020-01263-6).
- [4] N. Prakash, "Eye Anatomy," *Handbook of Optometry and Eye Disorders*, pp. 1–1, 2006, doi: [10.5005/jp/books/10343_1](https://doi.org/10.5005/jp/books/10343_1).
- [5] D. S. Kermany *et al.*, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, no. 5, pp. 1122–1131.e9, 2018, doi: [10.1016/j.cell.2018.02.010](https://doi.org/10.1016/j.cell.2018.02.010).
- [6] A. Laouarem, C. Kara-Mohamed, E. B. Bourennane, and A. Hamdi-Cherif, "HTC-retina: A hybrid retinal diseases classification model using transformer-Convolutional Neural Network from optical coherence tomography images," *Computers in Biology and Medicine*, vol. 178, no. June, p. 108726, 2024, doi: [10.1016/j.combiomed.2024.108726](https://doi.org/10.1016/j.combiomed.2024.108726).
- [7] N. Novelty, S. M. Shuvo, and Md. F. Faruk, "Improving Pre-Trained CNNs with CBAM and Skip Connections for Multi-Class Retinal Diseases Classification using OCT Images," in *Proceedings of the 3rd International Conference on Computing Advancements*, ACM, 2024, pp. 946–953, doi: [10.1145/3723178.3723304](https://doi.org/10.1145/3723178.3723304).
- [8] Ş. Aykat and S. Senan, "Using Machine Learning to Detect Different Eye Diseases from OCT Images," *International Journal of Computational and Experimental Science and Engineering*, vol. 9, no. 2, pp. 62–67, 2023, doi: [10.22399/ijcesen.1297655](https://doi.org/10.22399/ijcesen.1297655).
- [9] Z. Qi *et al.*, "MSLI-Net: retinal disease detection network based on multi-segment localization and multi-scale interaction," *Front. Cell Dev. Biol.*, vol. 13, no. June, pp. 1–18, 2025, doi: [10.3389/fcell.2025.1608325](https://doi.org/10.3389/fcell.2025.1608325).
- [10] J. Yang *et al.*, "HyReti-Net: hybrid retinal diseases classification and diagnosis network using optical coherence tomography," *Front. Med.*, vol. 12, no. September, pp. 1–13, 2025, doi: [10.3389/fmed.2025.1660920](https://doi.org/10.3389/fmed.2025.1660920).
- [11] A. Zedadra, M. Y. Salah-Salah, O. Zedadra, and A. Guerrieri, "Multi-Modal AI for Multi-Label Retinal Disease Prediction Using OCT and Fundus Images: A Hybrid Approach," *Sensors*, vol. 25, no. 14, pp. 1–20, 2025, doi: [10.3390/s25144492](https://doi.org/10.3390/s25144492).
- [12] K. Pin, J. W. Han, and Y. Nam, "Retinal diseases classification based on hybrid ensemble deep learning and optical coherence tomography images," *Electronic Research Archive*, vol. 31, no. 8, pp. 4843–4861, 2023, doi: [10.3934/era.2023248](https://doi.org/10.3934/era.2023248).
- [13] A. M. Mutawa, K. Sabti, B. S. Sundaram Thankaleela, and S. Raizada, "Bio-Inspired Optimization of Transfer Learning Models for Diabetic Macular Edema Classification," *AI*, vol. 6, no. 10, pp. 1–30, 2025, doi: [10.3390/ai6100269](https://doi.org/10.3390/ai6100269).
- [14] N. Verma, N. Kumar, G. Kumar, and K. Singh, "A hybrid ensemble framework with particle swarm optimization for network anomaly detection," *Discover Applied Sciences*, vol. 7, no. 8, Aug. 2025, doi: [10.1007/s42452-025-07419-x](https://doi.org/10.1007/s42452-025-07419-x).
- [15] O. S. Naren, "Retinal OCT Image Classification - C8" [Dataset]. Kaggle, 2021. doi: [10.34740/KAGGLE/DSV/2736749](https://doi.org/10.34740/KAGGLE/DSV/2736749).
- [16] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009, pp. 248–255. doi: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [17] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the International Conference on Machine Learning*, PMLR, 2019, pp. 6105–6114, doi: [10.48550/arXiv.1905.11946](https://doi.org/10.48550/arXiv.1905.11946).
- [18] X. Long *et al.*, "EfficientNetB0-Based End-to-End Diagnostic System for Diabetic Retinopathy Grading and Macular Edema Detection," *Diabetes, Metabolic Syndrome and Obesity*, vol. Volume 18, pp. 1311–1321, Apr. 2025, doi: [10.2147/DMSO.S506494](https://doi.org/10.2147/DMSO.S506494).
- [19] F. F. Jiven and R. Rumini, "MRI Classification of Brain Tumors Using EfficientNetB0 Feature Extraction and Machine Learning Methods," *Journal of Applied Informatics and Computing*, vol. 9, no. 6, pp. 3394–3404, Dec. 2025, doi: [10.30871/jaic.v9i6.10363](https://doi.org/10.30871/jaic.v9i6.10363).
- [20] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," in *Computer Vision – ECCV*, Springer, 2018, pp. 122–138. doi: [10.1007/978-3-030-01264-9_8](https://doi.org/10.1007/978-3-030-01264-9_8).
- [21] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollar, "Designing Network Design Spaces," in *Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2020, pp. 10425–10433. doi: [10.1109/CVPR42600.2020.01044](https://doi.org/10.1109/CVPR42600.2020.01044).
- [22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, 1995, pp. 1942–1948. doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, IEEE, 1998, pp. 69–73. doi: [10.1109/ICEC.1998.699146](https://doi.org/10.1109/ICEC.1998.699146).
- [24] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, Oct. 2007, doi: [10.1007/s11721-007-0002-0](https://doi.org/10.1007/s11721-007-0002-0).
- [25] J. Cao, S. Kwong, R. Wang, X. Li, K. Li, and X. Kong, "Class-specific soft voting based multiple extreme learning machines ensemble," *Neurocomputing*, vol. 149, no. Part A, pp. 275–284, Feb. 2015, doi: [10.1016/j.neucom.2014.02.072](https://doi.org/10.1016/j.neucom.2014.02.072).
- [26] S. A. Hicks *et al.*, "On evaluation metrics for medical applications of artificial intelligence," *Sci. Rep.*, vol. 12, no. 1, p. 5979, Apr. 2022, doi: [10.1038/s41598-022-09954-8](https://doi.org/10.1038/s41598-022-09954-8).

- [27] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: [10.1016/j.aci.2018.08.003](https://doi.org/10.1016/j.aci.2018.08.003).
- [28] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, Dec. 2015, doi: [10.1016/j.chemolab.2015.08.020](https://doi.org/10.1016/j.chemolab.2015.08.020).
- [29] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Population size in Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 58, Nov. 2020, doi: [10.1016/j.swevo.2020.100718](https://doi.org/10.1016/j.swevo.2020.100718).
- [30] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar. 1998, doi: [10.1109/34.667881](https://doi.org/10.1109/34.667881).
- [31] R. Delgado, "A semi-hard voting combiner scheme to ensemble multi-class probabilistic classifiers," *Applied Intelligence*, vol. 52, no. 4, pp. 3653–3677, Mar. 2022, doi: [10.1007/s10489-021-02447-7](https://doi.org/10.1007/s10489-021-02447-7).
- [32] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2015. doi: [10.48550/arXiv.1409.1556](https://doi.org/10.48550/arXiv.1409.1556).
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Conference on Computer Vision and Pattern Recognition*, IEEE, 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [34] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: [10.1007/s10462-020-09825-6](https://doi.org/10.1007/s10462-020-09825-6).