

Building a Lightweight and Time-Efficient Hybrid Model for Choroiditis Detection in Fundus Images: A Systematic Approach

Srishti Raj*

Computer Science & Engineering, BIT Mesra Campus, Birla Institute of Technology Mesra, Ranchi, India

E-mail: phdcs10051.20@bitmesra.ac.in

ORCID iD: <https://orcid.org/0000-0002-5775-2487>

*Corresponding author

Anup Kumar Keshri

Computer Science & Engineering, BIT Mesra Campus, Birla Institute of Technology Mesra, Ranchi, India

Email: anup_keshri@bitmesra.ac.in

ORCID iD: <https://orcid.org/0000-0001-9903-3109>

Received: July 22, 2025; Revised: May 25, 2026; Accepted: July 6, 2026; Published: August 8, 2026

Abstract: While the use of artificial intelligence in ophthalmology has increased over the past few years, inflammatory diseases like choroiditis remain widely unexplored. A delayed diagnosis in such cases can cause severe complications that can lead to visual disability, and hence, automated diagnosis systems can be of great importance. In this study, we present a systematic, evidence-driven methodology for designing a lightweight choroiditis classifier using a real-world small dataset with a constraint of low resource availability. This hybrid framework consists of two components. The first feature extraction component extracts features using the MobileNet-V3-Small model. The second component, namely the classification component, utilizes the Support Vector Machine as the binary classifier. This optimal combination was identified through systematic comparative experiments. Statistical testing confirms the robustness of the classifier selection. The model gives a cross-validation accuracy of 94% and a held-out test accuracy of 97.06% with a training time of approximately 3 minutes for the end-to-end pipeline on a carefully collected and previously introduced choroiditis dataset. Being lightweight and computationally efficient, this model is a suitable candidate for future development into a reliable computer-aided diagnosis tool that could assist experts in reviewing images, providing telemedicine care, and prioritizing patient appointments based on the initial results of the automated systems.

Index Terms: Choroiditis, Hybrid Framework, MobileNet-V3-Small, Support Vector Machine, Transfer Learning, Fundus Image Classification.

1. Introduction

Choroiditis is a specific form of posterior uveitis where the choroid gets inflamed. It is a vital vascular layer between the retina and the sclera and controls the flow of oxygen, blood, and other crucial nutrients to the retina. Hence, any impact on the choroid's health can have an immense effect on the retina and cause visual impairment [1,2].

Choroiditis occurrence can be infectious or non-infectious and may take acute, chronic, or recurrent forms. Inflammation can be localized to the choroid or involve retinal and vitreous structures, complicating the diagnosis and management. The common symptoms include blurred vision, redness, pain, floaters, photophobia, and impaired night vision. In the initial stages, the condition might remain asymptomatic, raising the risks of delayed diagnosis and irreversible vision loss. If left untreated, it can progress into sight-threatening secondary conditions such as cataracts, glaucoma, retinal atrophy, and macular edema [1]. Its diagnosis generally involves a physical examination of the dilated eye by an expert and supplementary imaging such as Optical Coherence Tomography (OCT), Fundus photography, and Fluorescein angiography. On the fundus images, the active form of choroiditis appears as yellow lesions with unclear borders, and the healed form manifests with a greyish appearance and well-defined boundaries, as depicted in Fig. 1 [3]. An intermediate regressing stage as defined in Section 3.1 is also represented in the dataset.

In the last few years, artificial intelligence (AI) has emerged as a promising tool for the timely diagnosis of ophthalmological conditions and for assisting specialists in decision-making. While there are numerous studies for common conditions such as diabetic retinopathy and glaucoma, studies related to inflammatory conditions such as choroiditis are less undertaken. This gap highlights an urgent need to develop automated diagnostic systems for such conditions to proceed towards AI-assisted healthcare.

In this work, we develop and evaluate a hybrid approach for the binary classification of fundus images into normal and choroiditis classes. The approach combines a lightweight deep learning model for feature extraction and a machine learning classifier for classification. We present a systematic approach for identifying this optimal combination for this task through multiple experiments.

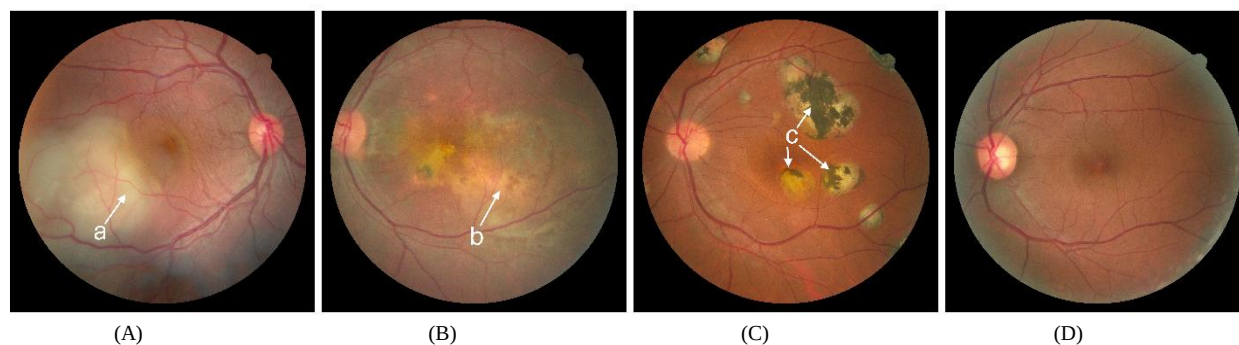


Fig. 1. Representative fundus images from the choroiditis dataset. (A) Active Choroiditis (B) Regressing Choroiditis (C) Healed Choroiditis (D) Normal Fundus. (a) active inflammatory lesion (b) regressing lesion (c) healed chorioretinal scars.

2. Literature Review

Several machine learning and deep learning algorithms have been recently developed to automate uveitis processes [4]. Several studies have aimed at studying the etiology [5-8] of uveitis, while several aimed at exploring the standardized classification of uveitis and its several sub-types utilizing clinical data and regression methodology under the Standardization of Uveitis Nomenclature (SUN) Project [9,10]. Some studies employed the use of Artificial Intelligence (AI) software to detect inflammation and cells in the anterior chamber [11-14] and vitreous haze grading [15] using fundus and OCT images. Haggag et al. [16] developed an automated system to grade vitreous inflammation using fused OCT images employing a U-net convolutional neural network (U-CNN) for segmentation and then a fully connected neural network for grading into five levels of severity. However, the data utilized in these works is small and is not publicly available. Parra et al. [17] evaluated the trustworthiness of deep learning models, namely, ResNet18, VGG16, and Vanilla CNN, in classifying fundus images into healthy and ocular toxoplasmosis with trust score as a metric, along with traditional evaluation metrics. In another work, Parra et al. [18] fine-tuned ResNet18 to detect ocular toxoplasmosis in fundus images. Alam et al. [19] benchmarked five pre-trained deep learning models, namely VGG16, DenseNet121, ResNet50, InceptionV3, and MobileNetV2, for the binary classification on the same dataset and MobileNetV2 emerged as the best performing model. Sun et al. [20] utilized the VGG-16 model via fine-tuning and hyperparameter adjustment for classifying experimental autoimmune uveitis (EAU) into three classes, namely, normal, trace, and disease, using the fundus photos. But the data is not publicly available, and the model's applicability to humans requires additional validation.

Although deep learning models and specialized datasets are used in recent research on automated uveitis diagnosis, these limitations recur across the literature: (i) use of heavyweight models, (ii) the choroiditis subtype receiving lesser attention compared to anterior uveitis and vitreous haze grading, (iii) model selection is rarely justified by systematic methodology and comparisons, with generally a single architecture reported without ablations, and (iv) external cohort validation is uncommon.

3. Materials and Methods

3.1. Data Description

This study uses choroiditis dataset [21] comprising 227 colored fundus images. It is a retrospective dataset comprising only images with no personal information or clinical metadata collected from a tertiary clinic of Eastern India.

Images having choroiditis lesions and completely healthy fundus were selected from the existing data repository of the clinic by an ophthalmologist and annotated as choroiditis or normal. To clearly distinguish the transitioning phase of active lesions towards healed; the regressing label was introduced. Regressing choroiditis signifies dull yellow lesions with lower inflammatory activity than active lesions, but where structural recovery is still ongoing. Some images in the dataset are multi-label implying they have more than one kind of choroiditis lesion. These labels and images were then

reviewed by the senior-most ophthalmologist at the clinic and confirmed. Any disagreement between the two specialists was discussed and resolved by consensus. For this binary classification work, choroiditis class comprises images containing active, regressing, and/or healed lesions, while the control class consists of normal fundus images. A total of 107 images belong to the positive (choroiditis) class, while 120 belong to the negative (normal) class. Table 1 gives the detailed characteristics of the dataset and Fig. 1 shows the examples of each type.

While the dataset is comparatively small by computer vision standards, it reflects the real-world scenario where the medical data for rare inflammatory conditions cannot be obtained in large numbers, subject to availability of patients and limitations of the research study and resources. To mitigate this data size limitation, techniques like transfer learning and data augmentation have been extensively utilized in the literature. Similarly, we also make use of such techniques while developing our final model.

Table 1. Dataset Characteristics.

Attribute	Description
Source	iRIS Super Speciality Eye Care Centre
Study Type	Retrospective, single-centre
Imaging Equipment	NIDEK AFC-330
Patient Ethnicity	Indian
Patient Age Range	Not recorded (adult patients)
Etiological subtype	Not recorded

Table 2. Distribution of Fundus Images.

Class Label	Number of Images
Active	38
Regressing	15
Healed	58
Normal	120

+ Individual images may contain lesions at more than one stage. Hence, the total count sums up to more than 107

3.2. Experimental Setup

All the experiments utilize Python sklearn [22] and PyTorch [23] and Torchvision [24] libraries on the Free version of Google Colab with default settings of no Graphics Processing Unit (GPU) and 12.7 GB Random Access Memory (RAM). All the experiments were executed on the CPU runtime of the platform to ensure hardware homogeneity for comparisons across models.

3.3. Proposed Methodology

This study proposes a systematic experimental strategy to derive an optimal framework for the choroiditis classification task. It involves three major experiments as follows:

1. Feasibility testing of traditional machine learning classifiers without feature learning.
2. Benchmark testing of popular transfer learning models under feature-extract, fine-tune, and scratch training configurations.
3. Designing and validating a hybrid framework that is fast yet accurate based on the results of experiments 1 and 2.
4. Comparing the hybrid framework against a few-shot learning baseline using prototypical networks.

Each phase is built on the limitations and outcomes of the previous one, forming a progressive, evidence-driven approach to model derivation. Fig. 2 shows the systematic approach followed.

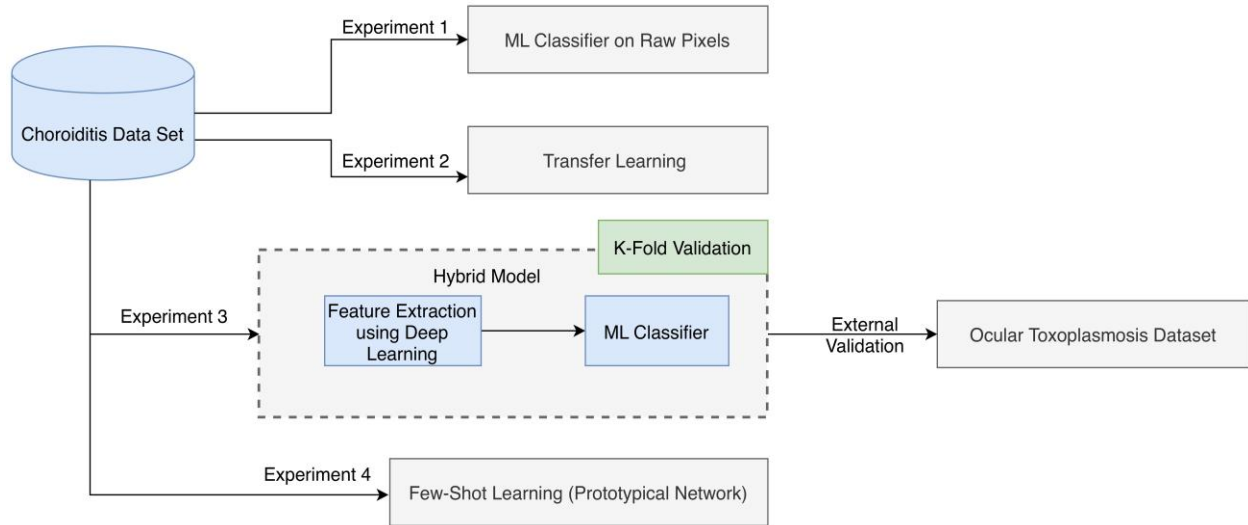


Fig. 2. Schematic Representation of the Experimental Framework.

3.4. Model Evaluation

The models were evaluated using standard classification metrics [25,26]; namely, accuracy, precision, sensitivity (recall), specificity, F1-score, and area under the receiver operating characteristic curve (AUC). In this study, a True Positive (TP) is an image labelled choroiditis and correctly classified by the model, while a False Negative (FN) is a choroiditis image incorrectly classified as normal. On the other hand, a normal fundus image labelled correctly as normal is a True Negative (TN), while misclassified as choroiditis is a False Positive (FP). Sensitivity measures the model's ability to correctly identify choroiditis cases, which is clinically crucial for a model meant for disease detection. On the other hand, specificity measures the ability of the model to correctly identify healthy images and thus saving the healthy individuals from unnecessary stress. These evaluation metrics offer a well-rounded view of the performance of the models in practice. Thus, highlighting the employability and limitations of the system in real-world clinical settings. The formal definitions of all evaluation metrics are provided in Appendix A (Equations 3-9).

4. Results and Discussion

As outlined in the proposed methodology, we conducted three major experiments whose results and outcomes helped us arrive at the optimal model for choroiditis classification. These included evaluating traditional machine learning models, transfer learning approaches, and constructing a hybrid model. The following section discusses these experiments and results in detail.

4.1. Experiment 1: Application of Traditional Machine Learning Models

Algorithm 1. Traditional Machine Learning Pipeline

Train Phase:

1. For each class in the training dataset:
 - 1.1. For each image in the class folder:
 - Resize the image to 224×224 pixels.
 - Flatten the image into a 1D feature vector.
 - Store the flattened image and corresponding label.
2. Encode categorical labels using a label encoder
3. Define a dictionary for all the machine learning classifiers
4. For each model in the dictionary:
 - Train the model on the flattened training data.
 - Record training metrics

Test Phase:

5. Apply the same preprocessing (Step 1) to the test data.
 6. Use the trained models to predict on the test data
 7. Record the performance metrics
-

Traditional machine-learning models are often more suitable for smaller datasets [27]. Hence, we began our study by testing the employability of such models, namely SVM, Random Forest (RF), Naive Bayes (NB), Decision Tree (DT), XGBoost, LightGBM and CatBoost [28-32] on the dataset with a 70:30 train-test split based on patient-level data splitting

to prevent data leakage. All the classifiers except CatBoost use library default hyperparameters. CatBoost uses 100 iterations and depth 4, decided experimentally after doing multiple runs. All the Tree-based ensemble models (XGBoost, LightGBM, CatBoost) are restricted to single-threaded execution to ensure bit-exact reproducibility across machines. Table 22 in Appendix describes the hyperparameters in detail. No data class balancing techniques were used as the class imbalance ratio is mild (1:1.12). Algorithm 1 outlines the complete pipeline, and Table 3 summarizes the execution results.

Table 3. Performance of machine learning classifiers.

Classifier	Training Accuracy (%)	Test Accuracy (%)	Test AUC Score	Sensitivity	Specificity	Training Time (s)	Inference (ms/image)
SVM	100	79.41	0.88	0.8438	0.7500	45.11	15.159
RF	100	64.71	0.78	0.5312	0.7500	2.07	0.545
NB	70.44	55.88	0.55	0.3125	0.7778	0.70	1.953
DT	100	69.12	0.69	0.6562	0.7222	9.49	0.295
XGBoost	100	75.00	0.81	0.6562	0.8333	73.54	1.004
LightGBM	100	76.47	0.83	0.6562	0.8611	98.78	12.879
CatBoost	100	69.12	0.78	0.5625	0.8056	478.90	6.661

Table 3 shows that all models except Naive Bayes achieve a 100% training accuracy but have significantly lower test accuracies ranging from 55.88% to 79.41%, which is a sign of overfitting. In contrast, Naive Bayes exhibited inferior performance in both the training and test phases, probably because its simple model is unsuitable for straightforward application. SVM achieved the highest test accuracy (79.41%) and AUC (0.88), followed by LightGBM (76.47%, AUC 0.83) and XGBoost (75.00%, AUC 0.81). SVM also gave the most balanced sensitivity and specificity for the choroiditis class, both clinically relevant metrics for disease detection. On the one hand, models achieving accuracy greater than 60% signify that performance can be improved with additional components or tweaks; on the other hand, the overfitting and poor generalization trends highlight that applying machine learning models directly on raw pixel values causes the loss of spatial information, leading to poor performance on unseen data. Hence, despite having a small dataset, applying traditional machine learning algorithms directly to the image classification task is unsuitable. However, we can utilize these models more effectively with external features or components for the same task. Keeping these observations in mind, we perform the next set of experiments exploring the feasibility of pre-trained deep-learning models for the given task.

4.2. Experiment 2: Application of Transfer Learning Models

Transfer learning is a popular methodology for image-processing tasks with smaller datasets [33]. In this process, information gained, like weights and biases of the convolutional layers from training the model on one task (ImageNet dataset), is utilized for the other given task. This methodology removes the requirement of having large datasets for the convergence of deep learning models, thus reducing the time needed for training. Hence, it is a feasible option for domains like biomedical image processing, where available data is limited, making it challenging to build a model from scratch.

For this experiment, we implemented several pre-trained models [34-44] for 30 epochs with early stopping patience of 10 and a delta of 0.01 on validation loss. SGD optimizer with a learning rate of 0.001 and a momentum of 0.9 was utilized. The model checkpoint corresponding to the lowest validation loss was retained for the test set evaluation. Table 23 in Appendix describes the hyperparameters in detail. 70% of the original dataset was used for training, 15% for validation, and 15% for testing. The models were executed in three configurations: feature extraction, fine-tuning and scratch. In feature extraction, all layers except the final classification layer are frozen with pre-trained weights of ImageNet and are used to extract features. Only the final classification layer is modified to consist of two neurons for the binary classification task and retrained on this new dataset. In fine-tuning, all layers are initialized with the weights and biases of ImageNet but remain trainable; hence, they can update these parameters utilizing backpropagation. In the scratch configuration, random initialization of the parameters happens, equivalent to training the model from the beginning as if no pre-training had happened. Algorithm 2 outlines the complete pipeline for this experiment 2.

Table 4 and Table 5 show the classification performance of the most widely benchmarked models and their computational efficiency, respectively. The results of all the remaining architectures are provided in Tables 24 and 25. All training times and inference measurements were obtained on CPU to provide a hardware-homogeneous comparison across all experiments. Inference was measured as the mean forward-pass time over 500 runs (batch size = 1) following 30 warmup iterations, using a standardized benchmarking protocol applied uniformly to all architectures.

In Table 4, the models Inception-V3 and VGG-16 achieved a test accuracy of 97.06% in the feature extraction mode, exhibiting that pretrained ImageNet representation can transfer efficiently to choroiditis domain even when the model's backbone is fully frozen. The models gave similar or better results when fine-tuning, with ResNet18 achieving a 100% test accuracy. MobileNet-V3-Small achieved a test accuracy of 100% but has lower train accuracy signifying that the results on the small test size should be interpreted with caution. Unlike these two configurations, the models' performance was significantly lower when trained from scratch. These results confirm that the utilization of transfer

learning in small medical datasets is beneficial and better than training the models with random initialization, especially for the compact architectures.

Table 5 reveals substantial differences in the computational cost of these models. MobileNet-V3-Small requires only 1.52M parameters, 5.93 MB of storage, 532 MB of memory, and an inference time of 7.94 ms per image, whereas VGG-16 demands 134.27M parameters, 512.21 MB of storage, 1067 MB of memory, and an inference time of 771.11 ms per image. This amounts to VGG-16 being 88 times larger in parameters and nearly 97 times slower in inference than MobileNet-V3-Small. The training times further highlight this difference with MobileNet-V3-Small converging in 7 minutes under feature extraction, while VGG-16 takes over 208 minutes in the scratch configuration for the same task.

However, it is important to note that these results are based on a one-time static split. Such a one-time split restricts the reliability of the models as the results might become over-optimistic or pessimistic depending on the distribution of images going into the train, validation and test sets. These observations signify the importance of increasing the data size via augmentation techniques to expand the dataset. Further, applying K-Fold cross-validation becomes a necessity to build more reliable and generalizable models by testing and training on each data point. However, the training time listed in Table 4 for a single split demonstrates that executing such resource-intensive deep learning architectures across multiple folds with higher amounts of data would require higher computational resources and more training time. This requirement makes it challenging to use these models in a resource-constrained environment. On the other hand, the machine learning classifiers do not require extensive resources but lack the robustness of identifying the spatial patterns or features from the images and hence are insufficient when used singly. These limitations highlight the need for a lightweight yet efficient model that can work in a resource-limited environment.

As shown in Table 4, MobileNet-V3-Small takes the shortest training time and performs competitively. Therefore, we applied several augmentation techniques to increase the data and used MobileNet-V3-Small as the base feature extractor. The features extracted are subsequently fed to a machine learning classifier to obtain an optimal hybrid model for choroiditis classification.

Algorithm 2. Transfer Learning Pipeline

Train Phase:

1. For each pre-trained architecture:
 - 1.1. For each model configuration (Feature extraction, Training from scratch, Fine Tuning):
 - Initialize the model.
 - Replace the final layer with a two-neuron output layer matching the number of classes.
 - 1.2. Data Preparation:
 - For each class in the train directory:
 - For each image in the class folder:
 - Resize the image to the architecture's minimum required size.
 - Convert the image to a Tensor format.
 - Normalize the image using the mean and standard deviation.
 - Create data loaders for both the train and validation sets.
 - 1.3. Check and set the device to CPU.
 - 1.4. Initialize the optimiser and loss function
2. Train and Validation Loop:
 - For each epoch:
 - 2.1. For each phase in ['train', 'val']:
 - Set model mode (training or evaluation)
 - Initialize variables to track performance metrics.
 - Iterate over data batches:
 - Load images and labels.
 - Perform a forward pass to compute the model output.
 - Compute the loss.
 - If in train mode:
 - Clear parameter gradients.
 - Perform a backward pass.
 - Update model parameters using the optimizer.
 - Update running loss and accuracy metrics.
 - 2.2. Save model if current validation loss < previous best.
 - 2.3. If validation loss has not improved by δ (0.01) for 10 consecutive epochs, stop training.

Test Phase:

3. Load the best model.
 4. Perform the same transformations on the test dataset as in step 1.2.
 5. Evaluate and obtain performance metrics.
-

Table 4. Performance Evaluation of Popular Transfer Learning Models.

Model	Configuration	Best Train Accuracy (%)	Average Train Accuracy (%)	Best Validation Accuracy (%)	Average Validation Accuracy (%)	Test Accuracy (%)	Training Time
MobileNet-V3-Small	Feature Extraction	98.11	92.43	97.06	96.37	100.00	7m 3s
MobileNet-V3-Small	Scratch	69.18	60.15	52.94	52.94	52.94	5m 4s
MobileNet-V3-Small	Fine Tuning	100.00	96.83	94.12	90.65	97.00	13m 44s
EfficientNet-B0	Feature Extraction	95.60	89.54	97.06	93.34	91.18	19m 28s
EfficientNet-B0	Scratch	90.57	78.55	79.41	64.41	88.24	17m 14s
EfficientNet-B0	Fine Tuning	100.00	96.02	97.06	95.69	97.06	28m 20s
ResNet-18	Feature Extraction	96.86	91.16	97.06	93.2	91.18	10m 21s
ResNet-18	Scratch	94.34	81.50	82.35	74.39	85.29	10m 53s
ResNet-18	Fine Tuning	100.00	97.77	97.06	95.59	100.00	24m 55s
Inception-V3	Feature Extraction	93.08	85.83	94.12	88.34	97.06	41m 12s
Inception-V3	Scratch	78.62	66.04	70.59	57.11	91.00	72m 3s
Inception-V3	Fine Tuning	100.00	96.00	97.06	92.65	97.06	63m 31s
VGG-16	Feature Extraction	89.31	82.68	97.06	91.98	97.06	58m 17s
VGG-16	Scratch	92.45	70.92	85.29	65.00	88.24	208m 12s
VGG-16	Fine Tuning	100.00	95.52	97.06	94.85	97.06	68m 58s

Table 5. Computational Efficiency of Popular Transfer Learning Models.

Model	Configuration	Params (M)	Size (MB)	FLOPs (GFLOP)	Memory (MB)	Inference (ms/image)
MobileNet-V3-Small	Feature Extraction	1.52	5.93	0.111	532.19	7.94
MobileNet-V3-Small	Scratch	1.52	5.93	0.111	530.18	11.13
MobileNet-V3-Small	Fine Tuning	1.52	5.93	0.111	530.03	12.77
EfficientNet-B0	Feature Extraction	4.01	15.58	0.769	554.98	76.63
EfficientNet-B0	Scratch	4.01	15.58	0.769	556.82	62.38
EfficientNet-B0	Fine Tuning	4.01	15.58	0.769	554.92	61.83
ResNet-18	Feature Extraction	11.18	42.71	3.627	575.36	100.09
ResNet-18	Scratch	11.18	42.71	3.627	571.92	67.94
ResNet-18	Fine Tuning	11.18	42.71	3.627	573.61	95.19
Inception-V3	Feature Extraction	24.35	93.21	11.423	681.77	364.82
Inception-V3	Scratch	24.35	93.21	11.422	647.07	348.29
Inception-V3	Fine Tuning	24.35	93.21	11.423	682.30	346.40
VGG-16	Feature Extraction	134.27	512.21	30.932	1067.1	765.58
VGG-16	Scratch	134.27	512.21	30.932	1051.4	771.11
VGG-16	Fine Tuning	134.27	512.21	30.932	1063.8	457.22

4.3. Experiment 3: Hybrid Deep and Machine Learning Model with Augmentation and Cross Validation

The model workflow comprises three primary tasks: Data preparation, Model building (Training and Validation) and Evaluation (Testing).

4.3.1. Data Preparation

In this step, the data is first divided into a training set and a test set. The test set is the same as that of Experiment 2, comprising 15% of the data. The remaining 85% of the data is divided into five folds. In each fold, the data is separated into training and validation sets using a patient-level stratified group K-fold split. This strategy ensures that no image from the same patient appears in both the training and validation sets of any fold. All the images are resized to 224×224 pixels. Then, in each fold, only the training data is augmented while the validation data remains unchanged. Augmentation not only increases the dataset's size but also introduces several variations to it. It also helps eliminate the class imbalance problem and control the problem of overfitting, as models built over augmented data have a more thorough training set and can be generalised better. Fig. 3 depicts the data augmentation process followed. The augmentation techniques of horizontal and vertical flip, saturation enhancement, Gaussian noise addition, affine transformation, per-channel multiplication, linear contrast adjustment, cropping, horizontal and vertical shearing, elastic and perspective transformations, gamma contrast adjustment, Gaussian blur, sharpening and adaptive equalisation are employed as presented in Appendix Table 26.

After augmentation, each class has 1092 images as training data in each fold, resulting in a balanced set for training. The original test set is kept as an independent set for model evaluation with 18 and 16 images in the normal and choroiditis classes to represent a real-world scenario. After this step, we proceed to feature extraction where each image is normalized using the mean and standard deviation of ImageNet data.

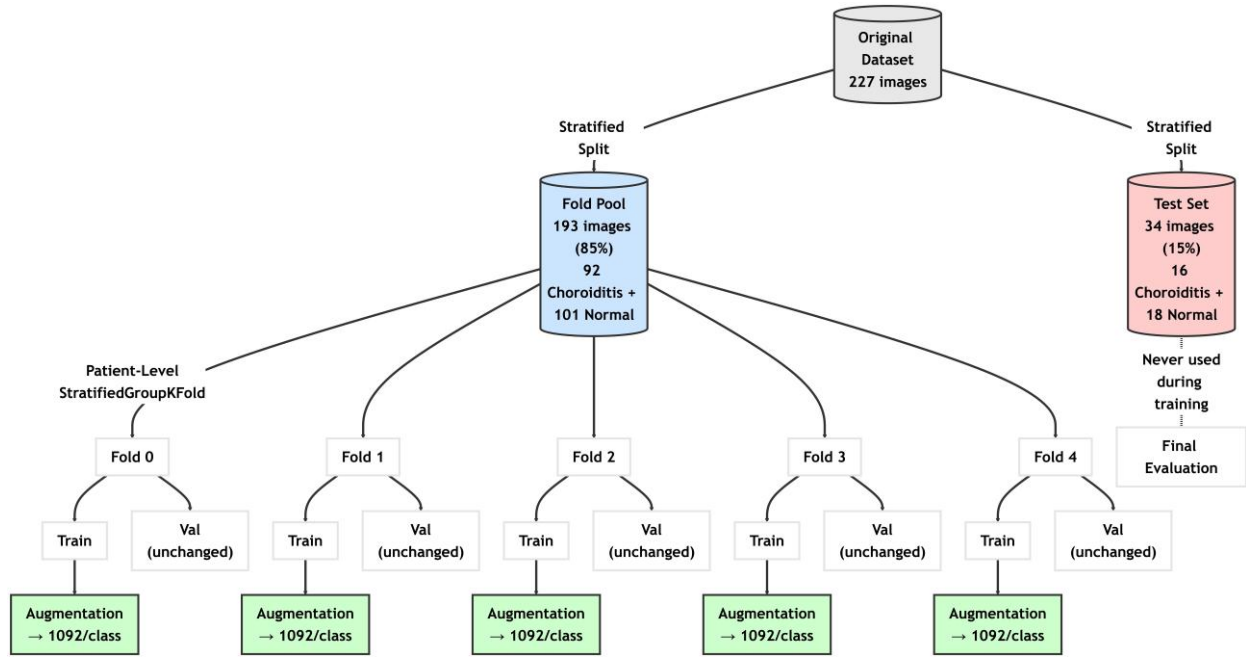


Fig. 3. Data Split and Augmentation Process.

4.3.2. Model Building and Evaluation

From Table 4, it is evident that MobileNet-V3-Small (Feature Extraction setting) during transfer learning takes the shortest amount of time with a decent performance. MobileNet-V3-Small [40,41] belongs to the MobileNet Convolutional Neural Networks family. It combines hardware-aware neural architecture search (NAS) with the NetAdapt algorithm. This combination enables it to operate in low-resource environments with minimum computational cost, such as mobile applications. This model is lightweight, faster and maintaining or improving accuracy compared to its predecessors, such as MobileNetV2. It employs depth-wise separable convolutions to minimize computational costs and parameters. The hard-swish activation function is also selectively employed across layers, balancing computational efficiency and accuracy.

The hard-swish function is given as in equation (1) below:

$$\text{hard-swish}(x) = x \cdot \frac{\text{ReLU6}(x+3)}{6} \quad (1)$$

where x is the output from one of the convolutional layers of MobileNet-V3-Small, representing the features extracted from the input fundus images before activation, and $\text{ReLU6}(x+3)$ is a bounded Rectified Linear Unit (ReLU) function capped to a value of 6, enhancing non-linear properties while controlling the range of the activation output.

Hence, to build a time-efficient and reliable model, we propose a hybrid architecture utilizing MobileNet-V3-Small as the feature extractor and then employing traditional machine learning classifiers for a K-fold cross-validation. This application of K-fold cross-validation ensures that each data point of the dataset gets the opportunity to participate in both training and testing across the folds during the model-building process. This, in turn, reduces model bias and variance, leading to a more generalized and robust model. Fig. 4 displays the overall architecture of the framework and Algorithm 3 describes the complete procedure.

Machine learning classifiers of experiment 1 are trained on the data after the feature extraction process. Tables 6 and 7 showcase their performance with 95% confidence intervals (Student's t formula) across the five stratified cross-validation folds. Excluding Naive Bayes, all classifiers had achieved perfect training accuracy but showed considerable differences in validation and test performance. A training accuracy of 1, but a massive gap between the training and validation/test accuracy of Decision Tree (18.2 percentage points) and XGBoost (12.1 percentage points) implies that these models are learning the training data more and overfitting on it, signifying that the generalization is poor and more memorization is happening. At the same time, Naive Bayes has a significantly lower performance with a wider confidence interval, exhibiting that the model is not suitable for this dataset. The sensitivity and specificity also exhibit poor performance for these models, as shown in Table 7. These models and the LightGBM model exhibit poor reliability in recognizing the disease's presence. SVM and Random Forest perform better than other models and require less time for training. These two models have high specificity in recognizing the disease, an important metric due to possible negative consequences (psychologically and physically stressful therapy consisting of potent corticosteroids) of making a false positive prediction. Apart from specificity, SVM shows the highest validation accuracy, AUC, and test

accuracy values (97.06%) with narrower confidence intervals for each metric compared to other models. The Random Forest emerges as the second-best model.

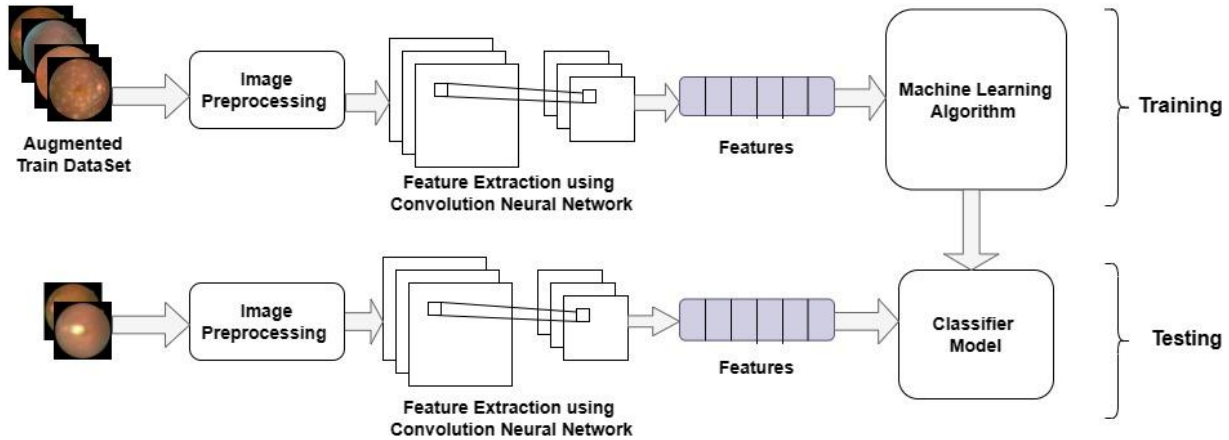


Fig. 4. Model Architecture.

Algorithm 3. Choroiditis Classification Using Proposed Methodology

Training

1. Initialize MobileNet-V3-Small with pretrained weights on ImageNet.
Exclude the top classification layer for using the model as a Feature Extractor.
2. For each category in the train directory:
For each image in the category's folder:
Resize the image to a size of 224×224 pixels.
Convert the image to a tensor and transform to match MobileNetV3's expectations.
Normalize the image using mean and standard deviation of ImageNet data
Append labels to each image.
3. For each category:
Apply Label Encoder to encode labels as 0 and 1.
4. Training and Validation Loop:
For each classifier:
4.1. For each fold in cross-validation:
Load augmented training and unaugmented validation data.
Reshape the data according to the classifier requirements.
Train the classifier on training data, record time and evaluate.
Evaluate the model on validation data.
Store results for each fold.

Testing

5. Prepare test data as in Step 2.
 6. Run the five-fold models, average the probabilities and evaluate the performance metrics.
-

Performing a pairwise exact two-sided binomial McNemar test [45] with a significance level of $\alpha = 0.05$ reveals the same as shown in Table 8. This test is conducted on the performance of models on the validation dataset of all folds ($n=193$). From the table, it is evident that the Decision Tree model differs significantly from most of the other models, including SVM, RF, LightGBM, and CatBoost. SVM and Random Forest, having a p-value of 0.5488, are similar. The XGBoost model also displays a difference with respect to SVM, RF, and CatBoost models. LightGBM differs from the Decision Tree and CatBoost models. At the same time, some models display the absence of a major statistical difference between them, namely the pairs of SVM-RF (0.5488), SVM-LightGBM (0.2379), SVM-CatBoost (0.7744), RF-NB (0.2266), RF-LightGBM (0.5488), NB-XGBoost (0.6072), and NB-LightGBM (0.8145). These results, when combined with the results of Tables 6 and 7, give SVM and RF as the closest candidates for hybrid model selection. Even a DeLong test applied to aggregated validation set probabilities across all five folds yields $z = 0.813$, $p = 0.416$, which renders the two classifiers statistically equivalent. However, the five-fold validation results shown in Tables 9 and 10 signify that Random Forest has wider confidence intervals and a wider per-fold spread compared to SVM. Hence, we proceed with SVM as the final selected classifier for the hybrid framework on this dataset.

Complementing the performance and statistical comparison of the models, Table 11 presents the computing cost of all the 7 classifiers on the shared backbone of the MobileNet-V3-Small (FLOPs 0.111 GFLOP, Feature Extraction Time 27.45 ms/image, Model Size 5.96 MB, all backbone parameters frozen). The average train time is the per-fold average time. The approximate total training time across all folds can be calculated as $5 \times$ average time of each fold. The

end-to-end inference (E2E inference) is the per-image inference time for the complete pipeline, including the shared backbone and the classifier head. In comparison, the Prediction time is the per-image classifier inference time without the backbone. After Naive Bayes, which is the poorest performer among all the classifiers, SVM is the fastest (1.88 s average per fold) and smallest (1.53 MB) model to train. The E2E time of 28.24 ms/image indicates that the SVM classifier head adds minimal cost on top of feature extraction using the MobileNet model (27.45 ms/image). In contrast, Random Forest is almost five times slower (8.85 s/fold) to train and ten times slower at the per-image predict step (7.90 ms) than SVM. Other models are all slower to train than SVM as well.

Table 12 and Fig. 5 and Fig. 6, show a more detailed analysis of the performance on the held-out test data. Total number of features extracted for each image by the MobileNet model is a 1024-dimensional feature vector per image, and the full hybrid pipeline with the components of feature extraction, and 5-fold SVM training takes a total time of approximately 3 minutes on our default free settings of Google Colab as described previously. Hence, the total time for this hybrid classifier on a larger augmented dataset is a considerable improvement compared to the other models' training time benchmarked on a much smaller non-augmented dataset. Thus, these efficiency characteristics, combined with the higher fold-wise and test performance of SVM, support its selection as the final classifier in the proposed hybrid framework.

Table 6. General Performance Metrics of Individual Classifiers at $k = 5$.

Classifier	Training Time (s)	Average Training Accuracy (95% CI)	Test Accuracy (%)	Average Validation Accuracy (95% CI)	Average Validation AUC (95% CI)	Test AUC Score
SVM	9.40	100	97.06	93.80 (90.94–96.67)	0.9798 (0.9588–1.0000)	1.0000
RF	44.26	100	94.12	91.86 (87.61–96.11)	0.9713 (0.9388–1.0000)	1.0000
NB	0.52	89.42 (87.26–91.59)	88.24	89.06 (82.51–95.61)	0.9520 (0.9146–0.9894)	0.9340
DT	36.21	100	82.35	81.79 (70.55–93.03)	0.7998 (0.6530–0.9466)	0.9688
XGBoost	37.01	100	91.18	87.85 (83.60–92.09)	0.9713 (0.9443–0.9983)	1.0000
LightGBM	87.67	100	91.18	91.61 (86.46–96.77)	0.9644 (0.9314–0.9974)	1.0000
CatBoost	66.49	100	94.12	91.80 (86.56–97.04)	0.9741 (0.9478–1.0000)	1.0000

Table 7. Class-Specific Metrics of Individual Classifiers.

Classifier	Class	Sensitivity (95% CI)	Specificity (95% CI)	Precision (95% CI)	F1-Score (95% CI)
SVM	Choroiditis	0.9086 (0.8151–1.0000)	0.9694 (0.9092–1.0000)	0.9587 (0.8882–1.0000)	0.9295 (0.8987–0.9603)
SVM	Normal	0.9694 (0.9092–1.0000)	0.9086 (0.8151–1.0000)	0.9179 (0.8504–0.9855)	0.9412 (0.9105–0.9719)
RF	Choroiditis	0.8549 (0.6911–1.0000)	0.9456 (0.8493–1.0000)	0.9461 (0.8543–1.0000)	0.8907 (0.7942–0.9872)
RF	Normal	0.9456 (0.8493–1.0000)	0.8549 (0.6911–1.0000)	0.9085 (0.8845–0.9325)	0.9250 (0.8828–0.9671)
NB	Choroiditis	0.7603 (0.4783–1.0000)	0.9573 (0.9024–1.0000)	0.9496 (0.8715–1.0000)	0.8247 (0.6157–1.0000)
NB	Normal	0.9573 (0.9024–1.0000)	0.7603 (0.4783–1.0000)	0.8624 (0.8113–0.9135)	0.9068 (0.8651–0.9484)
DT	Choroiditis	0.7279 (0.4857–0.9701)	0.8718 (0.7594–0.9841)	0.7935 (0.5319–1.0000)	0.7535 (0.5137–0.9933)
DT	Normal	0.8718 (0.7594–0.9841)	0.7279 (0.4857–0.9701)	0.8198 (0.7337–0.9059)	0.8418 (0.7709–0.9126)
XGBoost	Choroiditis	0.7596 (0.4756–1.0000)	0.9351 (0.8461–1.0000)	0.9351 (0.8473–1.0000)	0.8150 (0.6214–1.0000)
XGBoost	Normal	0.9351 (0.8461–1.0000)	0.7596 (0.4756–1.0000)	0.8602 (0.7860–0.9343)	0.8926 (0.8663–0.9189)
LightGBM	Choroiditis	0.8493 (0.5960–1.0000)	0.9365 (0.8470–1.0000)	0.9401 (0.8602–1.0000)	0.8761 (0.7251–1.0000)
LightGBM	Normal	0.9365 (0.8470–1.0000)	0.8493 (0.5960–1.0000)	0.9169 (0.8338–1.0000)	0.9236 (0.8771–0.9701)
CatBoost	Choroiditis	0.8283 (0.6519–1.0000)	0.9673 (0.9059–1.0000)	0.9643 (0.9033–1.0000)	0.8848 (0.7767–0.9928)
CatBoost	Normal	0.9673 (0.9059–1.0000)	0.8283 (0.6519–1.0000)	0.8939 (0.8363–0.9516)	0.9280 (0.8880–0.9680)

Table 8. McNemar's Test p-values at $\alpha = 0.05$ (Pairwise two-sided binomial).

	SVM	RF	Naive Bayes	Decision Tree	XGBoost	LightGBM	CatBoost
SVM	-	0.5488	0.0768	0.0000	0.0127	0.2379	0.7744
RF	0.5488	-	0.2266	0.0009	0.0215	0.5488	0.0625
Naive Bayes	0.0768	0.2266	-	0.0241	0.6072	0.8145	0.0129
Decision Tree	0.0000	0.0009	0.0241	-	0.0755	0.0081	0.0000
XGBoost	0.0127	0.0215	0.6072	0.0755	-	0.0625	0.0002
LightGBM	0.2379	0.5488	0.8145	0.0081	0.0625	-	0.0386
CatBoost	0.7744	0.0625	0.0129	0.0000	0.0002	0.0386	-

Table 9. RF 5-fold performance.

Fold	Accuracy	Sensitivity	Specificity	Precision	F1-Score	AUC
0	0.8824	0.8667	0.8947	0.8667	0.8667	0.9667
1	0.8947	0.95	0.8333	0.8636	0.9048	0.9278
2	0.9062	0.625	1	1	0.7692	0.9844
3	0.9574	0.92	1	1	0.9583	0.9936
4	0.9524	0.913	1	1	0.9545	0.984
Mean (95% CI)	0.9186 (0.8761–0.9611)	0.8549 (0.6911–1.0000)	0.9456 (0.8493–1.0000)	0.9461 (0.8543–1.0000)	0.8907 (0.7942–0.9872)	0.9713 (0.9388–1.0000)

Table 10. SVM 5-fold performance.

Fold	Accuracy	Sensitivity	Specificity	Precision	F1-Score	AUC
0	0.9118	0.8	1	1	0.8889	0.9544
1	0.9211	0.95	0.8889	0.9048	0.9268	0.9806
2	0.9688	1	0.9583	0.8889	0.9412	0.974
3	0.9362	0.88	1	1	0.9362	0.9945
4	0.9524	0.913	1	1	0.9545	0.9954
Mean (95% CI)	0.9380 (0.9094–0.9667)	0.9086 (0.8151–1.0000)	0.9694 (0.9092–1.0000)	0.9587 (0.8882–1.0000)	0.9295 (0.8987–0.9603)	0.9798 (0.9588–1.0000)

Table 11. Computational Efficiency of the seven classifiers.

Classifier	Average Train Time (s)	Classifier Size (KB)	Classifier Prediction Time (ms)	E2E Inference (ms)
SVM	1.88	1525.7	0.79	28.24
Random Forest	8.85	1212.6	7.90	35.35
Naive Bayes	0.1	57.2	0.56	28.02
Decision Tree	7.24	35.6	0.50	27.96
XGBoost	7.4	135.2	1.06	28.51
LightGBM	17.53	433.1	4.46	31.91
CatBoost	13.3	107.7	4.19	31.65

Table 12. Classification Report for SVM on Test Data.

Class	Precision	Recall	Specificity	F1-score	Support
Choroiditis	0.9412	1	0.9444	0.9697	16
Normal	1	0.9444	1	0.9714	18

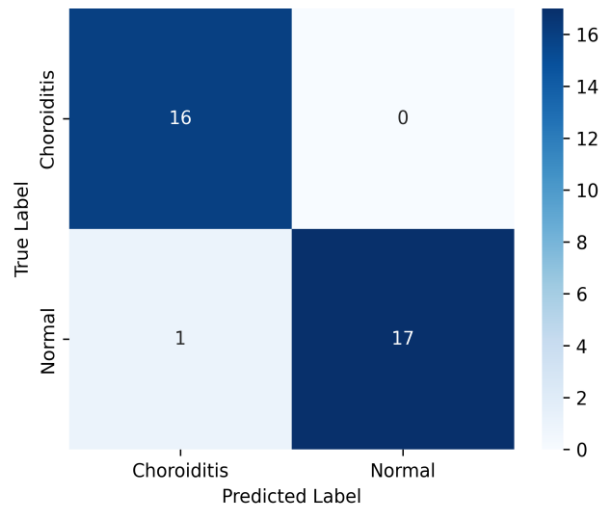


Fig. 5. Confusion Matrix for SVM on Test Data.

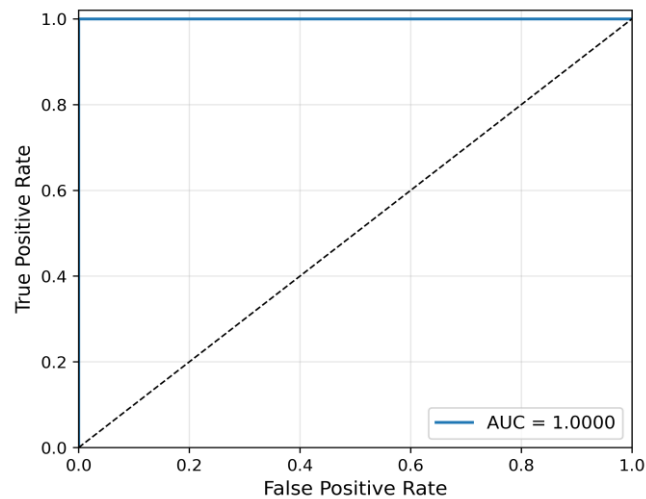


Fig. 6. ROC Curve for SVM on Test Data.

In this hybrid framework, MobileNet-V3-Small extracts the features as high-dimensional vectors storing essential spatial and other information cues that help distinguish between the two classes.

The feature vector can be given as $f = \text{MobileNetV3Small}(x)$, where MobileNet-V3-Small is an abstract function comprising all operations in the model's architecture for feature extraction. It includes convolutional layers, activation functions, pooling layers, and normalisation layers, and x denotes an input fundus image, and f is the extracted feature vector. Fig. 7 shows the t-Distributed Stochastic Neighbor Embedding (t-SNE) [46] projection of the features extracted from the held-out test set. It provides visual confirmation that the frozen backbone learns discriminative feature representations. The two classes occupy largely separate regions in the 1024-dimensional feature space, which directly explains the high separability achieved by the linear SVM classifier.

Next, the SVM [47,48] classifier receives this feature vector f and categorizes it as "Normal" or "Choroiditis". The training data's learned weights and bias form the basis of this decision, which optimizes the separation margin between the two classes.

The SVM decision function can be denoted as below in equation (2):

$$y = \text{sign}(\sum_{i=1}^N \alpha_i y_i K(f_i, f) + b) \quad (2)$$

where,

y represents the predicted label, normal or choroiditis, for an image x ,

N represents the total number of support vectors,

α_i represents the support vectors' coefficient learned during training,

y_i represents the label of these support vectors,

$K(f_i, f)$ represents the linear kernel function used for the SVM. Even when the data is not linearly separable in the original input space, this kernel technique enables the SVM to function in a modified feature space and identify a linear separation.

b represents the bias that adjusts the hyperplane to accommodate the distribution of data points better for more accurate classification.

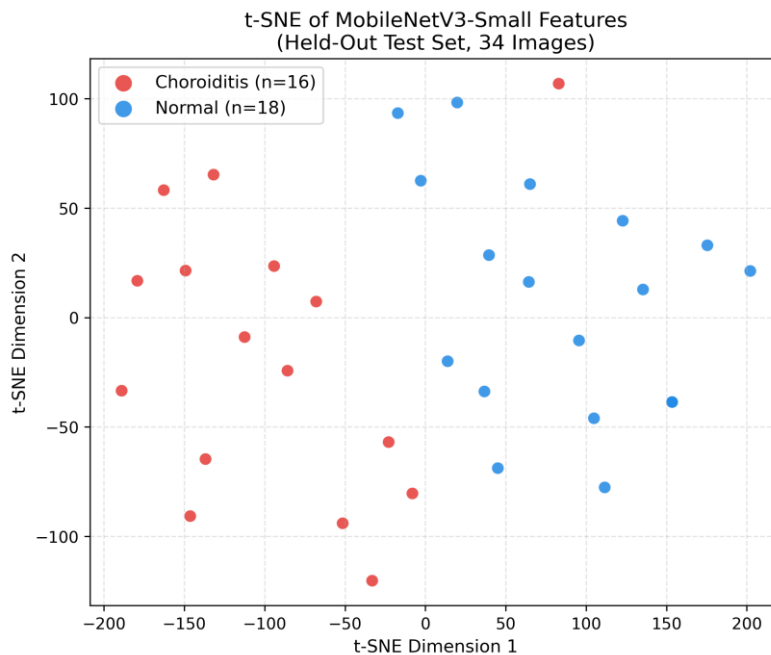


Fig. 7. t-SNE projection of MobileNet-V3-Small features from the held-out test set (34 images).

4.3.3. Error Analysis

This section breaks down the Choroiditis-class predictions on the held-out test set by the subtypes and analyses the accuracy of all seven classifiers sharing the MobileNet backbone. Table 13 reports this per-subtype classification rate. Each cell shows the number of correctly classified images out of the total images mentioned in the header.

For the Normal class, the SVM ensemble produces a single false positive. This same Normal image is misclassified as Choroiditis by all six other classifiers. Fig. 8 shows the Grad-CAM [49] of SVM for this case, and the Appendix Fig. 10 shows the corresponding Grad-CAMs for each of the seven classifiers. These figures reveal that for this particular normal image, the models focus on the black-shadowed region and the yellowish light appearing in the posterior pole of the fundus image, confusing it with pigments of healed lesions, thus resulting in wrong predictions.

The same image misclassified by all classifiers and their Grad-CAMs spanning kernel methods, tree ensembles, gradient boosting, and a generative model indicates that the visual features driving the false positive are coming from the features extracted by the shared frozen MobileNet-V3-Small backbone and not from any individual classifier's decision boundary.

For the Choroiditis class, the chosen SVM hybrid classifier correctly classifies all the images irrespective of the subtype, as depicted in Fig. 9. From these Grad-CAMs, it is evident that the model's attention shifts appropriately with the subtypes. The model concentrates on active inflammatory lesion borders, on granular hypopigmented patches in healed lesions, on mixed yellowish active lesions and pigmented residual patterns in recurrent disease (having more than one kind of lesion), and on resolving lesion margins in regressing disease. Thus, it indicates that the learned representation generalizes across the morphological range of the disease rather than relying on a single visual representation. Similarly, Random Forest, XGBoost, LightGBM, and CatBoost identify all choroiditis test images correctly. The two weakest classifiers, namely Decision Tree and Naive Bayes, fail specifically on the Active subtype (DT misses 2 of 4 Active images, NB misses 1 of 4). The Active class, misdiagnosed by these two models, is a non-trivial case which is diagnosed correctly by the SVM hybrid model.

Table 13. Per-subtype Choroiditis Performance analysis on the held-out test set.

Classifier	Correct Active (n=4)	Correct Healed (n=9)	Correct Recurrent (n=2)	Correct Regressing (n=1)	Total Choroiditis
SVM	4	9	2	1	16
RF	4	9	2	1	16
NB	3	9	2	1	15
DT	2	9	2	1	14
XGB	4	9	2	1	16
LightGBM	4	9	2	1	16
CatBoost	4	9	2	1	16

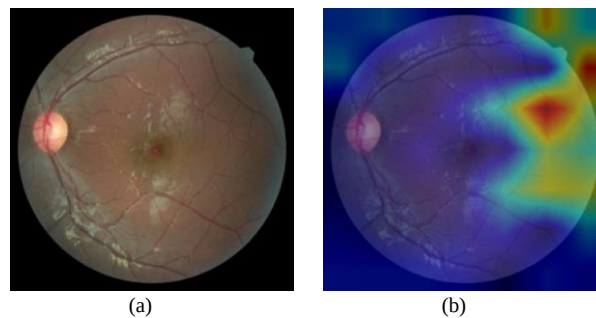


Fig. 8. SVM hybrid pipeline failure case on the held-out test set. (a) shows the original Normal-class fundus image that the SVM ensemble misclassified as Choroiditis. (b) shows the Grad-CAM overlay computed for the SVM's predicted class.

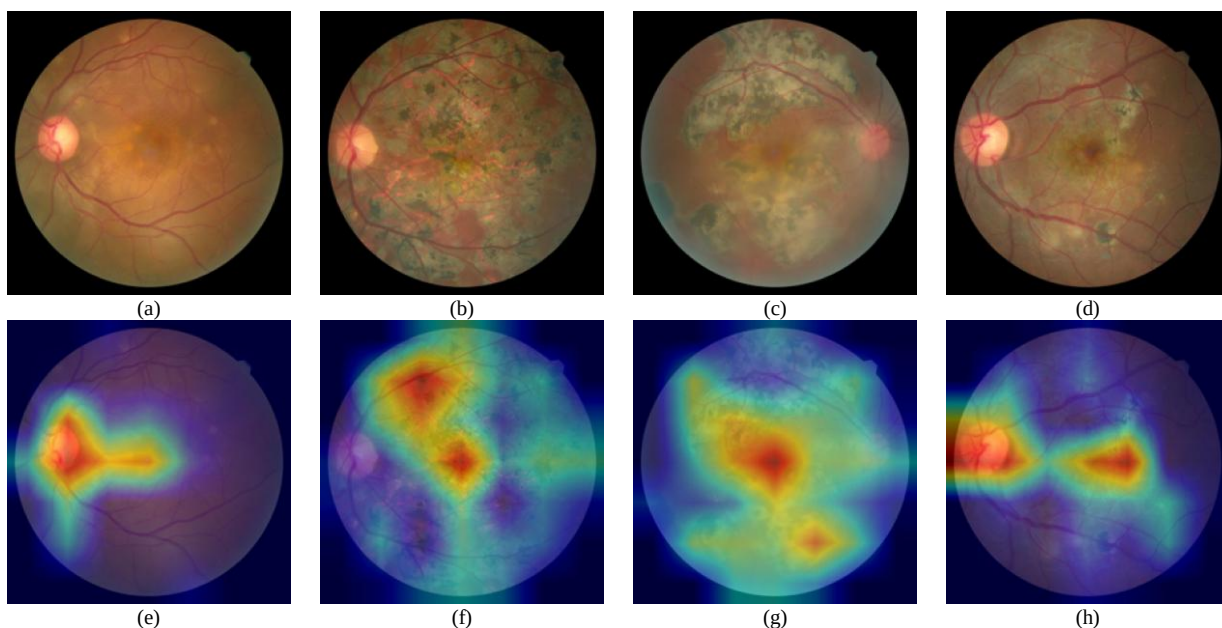


Fig. 9. Grad-CAM visualization for the SVM hybrid model. Fundus Images with (a) Active Lesions, (b) Healed Lesions, (c) Recurrent (with prior Healed and current active) Lesions, and (d) Regressing Lesions. (e), (f), (g), (h) - corresponding Grad-CAM overlays.

4.3.4. External Validation on Independent Cohort

This section assesses the generalizability of the proposed hybrid model by conducting an external validation analysis on the public Ocular Toxoplasmosis (OT) dataset [50], consisting of 412 colored fundus images from two independent hospital centres. Of these, 132 images are healthy, and 280 have ocular toxoplasmosis (active, inactive or both). This dataset is selected because ocular toxoplasmosis manifests as choroiditis and chorioretinitis lesions, thus making it a relevant test set for cross-cohort transferability.

For this binary classification task, the healthy images belong to the negative (Normal) class, and all toxoplasmosis images belong to the positive (Disease) class. The five SVM hybrid models of the five folds from Experiment 3, trained on the choroiditis data, are evaluated on the OT dataset. Then the disease-class probabilities of all folds' models are averaged out to give a single probability, which decides the final label. A probability of 0.5 or greater signifies a diseased image; otherwise, it belongs to the normal class.

Table 14 presents the performance of each fold model on this external dataset. The accuracy across each fold ranges from 0.7961 to 0.8495, while AUC ranges from 0.8526 to 0.9185. These results indicate the model is stable and exhibits only moderate variability with no dramatic differences, even when the cohort changed. This variability is mitigated with the averaging of probabilities, as depicted in Table 15. It gives a sensitivity of 89.64%, correctly identifying 251 out of 280 diseased images, alongside an overall accuracy of 83.50% and an AUC of 0.9059. These results show that the MobileNet-V3-Small backbone learned features that are generalisable and transfer effectively across the two posterior uveitis cohorts (choroiditis and ocular toxoplasmosis). However, the specificity is 70.45%, which means 39 out of 132 healthy fundus images were misclassified as diseased images by the model. This behaviour is expected with zero-shot external cohort evaluation because while the inflammatory conditions have the same lesion features across the datasets, normal images can have different features coming from different imaging equipment, clinical centres, acquisition protocols, and demographic populations with differences in ethnic retinal pigmentation (Indian vs Paraguayan). While the features of normal images from the external cohort differ, each fold's StandardScaler was fitted on the source choroiditis training features, which results in a normalization mismatch that shifts some healthy-fundus feature vectors toward the disease side of the linear decision boundary of the SVM.

Nevertheless, an AUC of 0.9059 on the external cohort indicates a balanced discriminative performance for both classes. When compared to the Alam et al [19] study, MobileNetV2 achieved the best cross-validation accuracy of 0.989 ± 0.006 and an AUC of 0.97 ± 0.02 on this cohort. However, comparisons of the proposed hybrid model with their model are not straightforward, as they differ in disease type and dataset. The hybrid model, achieving a test accuracy of 97.06% and an AUC of 1.00 on choroiditis fundus images, is competitive with their results, while also achieving a decent performance on the OT dataset. This indicates the framework has genuine potential as a screening tool for posterior inflammatory conditions. To further improve the specificity for the external cohorts, domain adaptation or fine-tuning on the target population would be needed, which is a direction for future work.

Table 14. Per-fold external validation results on the independent ocular toxoplasmosis cohort.

Fold	Accuracy	Sensitivity	Specificity	F1-Score	AUC
0	0.8107	0.8464	0.7348	0.8587	0.8863
1	0.8495	0.9643	0.6061	0.8970	0.8801
2	0.8325	0.8786	0.7348	0.8770	0.9033
3	0.8325	0.8250	0.8485	0.8701	0.9185
4	0.7961	0.8714	0.6364	0.8531	0.8526

Table 15. External validation results without retraining.

Accuracy	Sensitivity	Specificity	F1-Score	AUC
0.8350	0.8964	0.7045	0.8807	0.9059

4.4. Experiment 4: Few-Shot Learning Baseline with Prototypical Networks

In recent years, few-shot learning has emerged as a widely used technique with competitive performance, especially for biomedical tasks with limited training data [51]. Unlike traditional deep learning models, it aims to classify unseen images from very few labelled samples. Hence, to evaluate our hybrid model further, we compare it with the performance of a prototypical few-shot learner built on the same MobileNet-V3-Small architecture as the backbone.

A prototypical network is a type of few-shot learning method that creates a prototype representation of each class, by calculating the mean feature vector from its support samples. The label for an unseen image is then predicted by calculating the Euclidean distance of the image to each prototype. The class at minimum distance from the test image is assigned as the predicted label. The only difference between the two models is the usage of distance in few-shot learning and a decision boundary in the SVM classifier. Algorithm 4 describes the complete procedure followed. Table 16 shows the prototype network performed consistently across all K-shot settings employed during evaluation. The average accuracy is approximately 88% with a narrow 95% confidence interval (Student's t) of $\pm 0.18\%$. The negligible difference between the performance of 1-shot (88.77%) and full-shot (88.82%) implies that the model performance is

not based on the number of support samples for training but on the quality of embedding space created. Table 17 presents the full-shot test evaluation and Table 18 presents the Performance and Computational Cost to facilitate direct comparison with Experiments 2 and 3. All architecture hyperparameters are provided in the Appendix Table 27.

Algorithm 4. Prototypical Network Pipeline

Training Phase:

1. Initialize MobileNet-V3-Small with pretrained weights on ImageNet.
Replace the final classification layer with an identity layer.
Freeze the backbone of MobileNet-V3-Small.
2. Initialize the projection head.
3. Set the device to CPU.
4. Extract features for the train and validation sets using the frozen backbone.
5. Episodic Training Loop:
For each episode:
Sample a 2-way 5-shot episode (5 support and 15 query images for each class).
Calculate the class prototypes as the mean embedding of each class's support set.
Calculate the Euclidean distance of the query images with the prototypes of each class.
Assign query images to the nearest prototype class.
Compute loss and update projection head parameters.
Evaluate on validation episodes and save projection head if validation loss decreases.
If validation loss has not improved for 100 consecutive episodes, stop training.

Test Phase:

6. Load the best projection head.
7. Extract features for the test set using the frozen backbone.
8. Compute class prototypes from the full training set embeddings.
9. Classify test images to the nearest prototype class.
10. Evaluate across K-shot settings (1, 5, 10, full) over 600 episodes each.

Record performance metrics.

Table 16. K-Shot Episodic Evaluation of a 5-Shot Trained Prototypical Network (600 evaluation episodes per setting).

K-Shot Setting	Average Episodic Accuracy (95% CI)	Best Episode	Worst Episode
1-shot	88.77 $\pm$ 0.18	96.67%	86.67%
5-shot	88.74 $\pm$ 0.18	96.67%	86.67%
10-shot	88.84 $\pm$ 0.18	96.67%	86.67%
Full shot	88.82 $\pm$ 0.18	96.67%	86.67%

Table 17. Full-Shot Test Set Evaluation of Prototypical Network.

Class	Precision	Sensitivity	F1-Score	Specificity	Support
Choroiditis	0.8000	1.0000	0.8889	0.7778	16
Normal	1.0000	0.7778	0.8750	1.0000	18

Table 18. Performance and Computational Cost of Prototypical Network.

Test Accuracy (%)	AUC	FLOPs (GFLOP)	Inference (ms/image)	Memory (MB)	Model Size (MB)	Total Params	Trainable Params	Frozen Params	Training Time
88.24	0.9896	0.111	12.42	668.16	7.10	1,813,664	295,808	1,517,856	~4 s

The model achieved a test accuracy of 88.24% with an AUC of 0.9896 on the full test set. Noticeably, the sensitivity score is 1.0 for class choroiditis, meaning the model identified all the diseased images in the test set correctly. However, this perfect sensitivity is achieved at the cost of reduced specificity (0.78), meaning the model flagged 4 healthy fundus images as having choroiditis. This trade-off contrasts with the hybrid model which achieves higher accuracy and a balanced sensitivity-specificity under 5-fold cross validation. While the prototype model is faster in training (4 seconds) than the hybrid model, it lags in the classification task by a significant margin. Moreover, the prototypical model can be biased with the test dataset, while the hybrid model is more rigorously tested for generalizability via the cross-fold validation.

These results suggest that for the small choroiditis dataset of this scale, transfer learning combined with traditional classifiers is more effective than distance-based few-shot learning. This difference can be attributed to the full training set utilization helping the model identify a robust SVM decision boundary, whereas the Euclidean distance metric acts as a less flexible classifier.

4.5. Ablation

An ablation study is a model analysis approach where one component of a system is swapped or removed to determine whether the component is the right choice or not. In this paper, the backbone (MobileNet-V3-Small as feature extractor) is kept constant, and ablation is performed on the classifier head between Softmax classifier, SVM and Few-shot prototypical network. All three methods receive the same input of 1024-dimensional feature vectors extracted by the MobileNet-V3-Small backbone, normalized by the same StandardScaler and evaluated under the same five-fold patient-stratified cross-validation and on the same 34-image held-out test set. Table 19 shows these ablation results on the Test set. The Few-Shot network clearly lags behind the SVM and Softmax classifier. It is an expected outcome because the prototypical networks work well for cases with very few labelled examples per class (typically 1 to 5 shots). This dataset has approximately a hundred examples per class, so the episodic training paradigm is not a good match, and a conventional linear classifier trained on all available data works better. The SVM and Softmax classifier performance is the same on the test set. Table 20 shows the cross-validation performance of the two classifier heads, with SVM having slightly higher choroiditis sensitivity and F1-score. This value is clinically important as it means improved detection of diseased fundus images and better balance between recall and precision. Given the clinical importance of minimizing missed choroiditis cases, these factors motivated the selection of SVM as the preferred classifier.

Table 19. Approach Ablation: SVM vs SoftMax vs Few-Shot Network.

Approach	CV Average Accuracy (95% CI)	CV Average AUC (95% CI)	Test Accuracy (%)	Test Sensitivity	Test Specificity	Test AUC	Total CV Train (s)
SVM (linear)	93.80 (90.94-96.67)	0.9798 (0.9588-1.0000)	97.06	1.0000	0.9444	1.0000	9.40
SoftMax (Logistic Regression)	92.53 (87.44-97.62)	0.9780 (0.9515-1.0000)	97.06	1.0000	0.9444	1.0000	1.79
Few-Shot Prototypical Network	- (episodic)	- (episodic)	88.24	1.0000	0.7778	0.9896	~4

Table 20. Per-class cross-validation breakdown: SVM vs SoftMax (Logistic Regression).

Classifier	Class	Sensitivity (95% CI)	Specificity (95% CI)	Precision (95% CI)	F1-Score (95% CI)
SVM	Choroiditis	0.9086 (0.8151-1.0000)	0.9694 (0.9092-1.0000)	0.9587 (0.8882-1.0000)	0.9295 (0.8987-0.9603)
SVM	Normal	0.9694 (0.9092-1.0000)	0.9086 (0.8151-1.0000)	0.9179 (0.8504-0.9855)	0.9412 (0.9105-0.9719)
SoftMax (LR)	Choroiditis	0.8783 (0.7723-0.9842)	0.9609 (0.9330-0.9889)	0.9400 (0.8817-0.9983)	0.9066 (0.8315-0.9817)
SoftMax (LR)	Normal	0.9609 (0.9330-0.9889)	0.8783 (0.7723-0.9842)	0.9078 (0.8398-0.9757)	0.9328 (0.8932-0.9725)

Table 21. SVM Kernel Ablation: Linear vs. RBF (5-Fold Cross-Validation).

Metric	Linear Kernel	RBF Kernel
Average Accuracy	93.80%	91.76%
95% CI (Accuracy)	90.94% – 96.67%	86.27% – 97.26%
Average AUC	0.9798	0.9717
95% CI (AUC)	0.9588 – 1.0000	0.9490 – 0.9944
CI Width (Accuracy)	±2.87	±5.50
Model file size	1.5 – 1.6 MB /fold	4.3 – 4.8 MB / fold

5. Limitations

Although this work is a substantial effort in the direction of choroiditis detection, there are several limitations and challenges requiring exploration in future work. First, the dataset comprises 227 fundus images from a single clinical centre with patients of Indian ethnicity. The generalizability of the trained model to other populations, ethnicities, or imaging equipment needs further efforts. The external validation analysis on the Ocular Toxoplasmosis dataset is a preliminary transferability examination rather than absolute evidence of generalizability. Second, the dataset is a retrospective collection of fundus images and lacks clinical metadata such as patient age, gender, disease duration, treatment history, or etiology (infectious, autoimmune or idiopathic). In the absence of such metadata, subgroup analyses are not possible. Third, the current work focuses on identifying diseased fundus images and does not separately identify the grade of the disease as active, regressing or healed. Fourth, the ground-truth annotations were established through a two-tier consensus protocol involving two expert graders with adjudication by a senior ophthalmologist, ensuring a high standard of annotation quality. However, the labelling lacks a formal inter-rater reliability metric like Cohen's kappa, thus preventing quantitative assessment of annotation consistency. Fifth, on the external Ocular Toxoplasmosis dataset, the model gave an AUC of 0.9059 and a sensitivity of 89.64%, but the specificity was only 70.45% on healthy fundus images. This behaviour on external validation indicates a residual cohort shift. Validating the model on a multi-centre choroiditis dataset with standardized acquisition protocols is required for clinical deployment. Despite these limitations, this work demonstrates the feasibility and potential of the lightweight hybrid framework for choroiditis detection.

6. Conclusion and Future Work

The hybrid framework promises a significant advancement in the automated detection of choroiditis from fundus images, combining the strengths of MobileNet-V3-Small for feature extraction and SVM for classification. This hybrid approach achieved a five-fold cross-validation accuracy of 93.80% (AUC: 0.9798) and a held-out test accuracy of 97.06% (AUC: 1.00) with a total pipeline runtime of approximately three minutes, exhibiting high computational efficiency, rendering it suitable for real-world applications. Notably, the framework demonstrates a test specificity of 0.9444 and a cross-validation specificity of 0.9694, correctly identifying the majority of healthy individuals and protecting them from the unnecessary stress of a false diagnosis.

This model is small (5.96 MB) and requires minimal computational resources (0.111 GFLOPs). Because of this, it has strong potential for development into a lightweight clinical mobile application in the future. Its deployment in tertiary clinics can assist ophthalmologists with appointment scheduling on a priority basis and in early diagnoses, which will not only enhance the diagnostic capability of the health industry but also help in the holistic management of such patients. To achieve this, first, the model would require conversion into a mobile-friendly format such as ONNX or TensorFlow Lite. Next, it would have to be tested across different devices to ensure it functions correctly after the conversion. Finally, a complete mobile application would need to be built around the model. From a regulatory perspective, building such AI tools would fall under the Software as a Medical Device (SaMD) category. In the US, it would be considered an FDA Class II device. If a similar device already exists, the tool can go through the 510(k) pathway. If no such device exists, it would instead need to go through the De Novo pathway. In the European Union, the equivalent process is CE marking under the Medical Device Regulation (MDR 2017/745). Here, the tool would fall under Class IIa, as it is meant to support diagnosis and not replace the experts. Fulfilling this regulation requires validation of the model on a larger, multi-centre dataset and a prospective study in the future. Beyond deployment, several promising directions exist for this study to advance further. The future stages must also consist of the expansion of the model to grade the severity of choroidal lesions, extending its coverage to other ocular disorders, and enhancing the datasets with a broader range of demographic data. Furthermore, adopting advanced techniques like ensemble methods and improving the model might enhance its efficacy and promote user confidence in such tools. Ultimately, more advancements and research in this area will open the door to more advanced, readily available tools and improve the quality of life for both patients and medical experts.

All the Declarations and Statements

Author Contributions Statement

Srishti Raj – Conceptualization, methodology, data curation, software implementation, model training and validation, formal analysis, visualization, and writing of the original draft preparation.

Anup Kumar Keshri – Supervision, project administration, writing, review, and editing.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This study is not funded by any person or organization.

Data Availability Statement

The dataset used in this study is part of an on-going research effort and is planned for public release through a dedicated data descriptor in future work. Due to ethical considerations and institutional data-sharing restrictions associated with clinical data, the dataset cannot be made fully publicly available at this stage. However, the data can be made available for research and academic purposes upon reasonable request to the corresponding author, subject to a formal data sharing agreement and approval to ensure compliance with patient privacy and institutional guidelines.

Ethical Declarations

This study was conducted using a retrospective collection of de-identified fundus images. No personal patient information was collected or processed. Ethical approval for the use of clinical data was obtained from iRIS Super Speciality Eye Care Centre, Ranchi, India (Letter No. iRIS/HR-EC/24012) and Kusum Independent Ethics Committee (CT/2025/SR) under the Indian Rules for usage of retrospective data for AI research purposes. No prospective patient recruitment or intervention was involved.

Acknowledgments

The authors thank the expert team of iRIS Eye Hospital (iRIS Eye Care Center) for providing the choroiditis dataset (Letter No. – iRIS/HR-EC/24012).

Declaration of Generative AI in Scholarly Writing

AI-assisted technologies are used at very few places to strengthen the language and readability.

Abbreviations

The following abbreviations are used in this manuscript:

AI - Artificial Intelligence
AUC - Area Under the Receiver Operating Characteristic Curve
CAD - Computer-Aided Diagnosis
CI - Confidence Interval
CNN - Convolutional Neural Network
CPU - Central Processing Unit
CV - Cross Validation
DT - Decision Tree
EAU - Experimental Autoimmune Uveitis
FDA - Food and Drug Administration
GFLOPs - Giga Floating Point Operations per Second
Grad-CAM - Gradient-weighted Class Activation Mapping
LightGBM - Light Gradient Boosting Machine
MDR - Medical Device Regulation
NAS - Neural Architecture Search
NB - Naive Bayes
OCT - Optical Coherence Tomography
ONNX - Open Neural Network Exchange
OT - Ocular Toxoplasmosis
RAM - Random Access Memory
RBF - Radial Basis Function
RF - Random Forest
ROC - Receiver Operating Characteristic
SaMD - Software as a Medical Device
SGD - Stochastic Gradient Descent
SUN - Standardization of Uveitis Nomenclature
SVM - Support Vector Machine
t-SNE - t-Distributed Stochastic Neighbor Embedding
XGB - XGBoost (Extreme Gradient Boosting)

Appendix

To support reproducibility, this appendix provides the complete implementation details, metrics used in the study and all the results.

A. Model Evaluation

Several essential performance metrics derived from the confusion matrix elements mentioned in Section 3.4 are shown in equations 3-9 below:

Accuracy indicates the model's ability to correctly classify images for both classes among the total number of images evaluated.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Precision indicates the model's ability to correctly classify images for a particular class among all the images that it predicted belonged to the class. A high precision for choroiditis signifies that more patients would receive timely medical attention after diagnosis.

$$\text{Precision}_{\text{Choroiditis}} = \frac{TP}{TP+FP} \quad (4)$$

$$\text{Precision}_{\text{Normal}} = \frac{TN}{TN+FN} \quad (5)$$

Sensitivity or recall indicates the model's ability to correctly classify images for a particular class from the total number of images in that class. A high recall for choroiditis means the model rarely misses patients with this condition, which is vital to prevent vision loss.

$$\text{Sensitivity}_{\text{Choroiditis}} = \frac{TP}{TP+FN} \quad (6)$$

Specificity indicates the model's ability to identify negative cases for a class correctly. A high specificity for the normal class means the model rarely misidentifies healthy individuals as having choroiditis, which is desirable to protect them from the unnecessary stress of a wrong diagnosis.

$$\text{Specificity}_{\text{Normal}} = \frac{TN}{TN+FP} \quad (7)$$

The *F1-score* indicates the model's combined power of precision and recall, calculated as their harmonic mean. It suggests the model's ability to make correct predictions for the positive cases while minimizing the false positives. A high F1-score means that most of the time the model neither fails to identify a patient with the disease nor misdiagnoses a healthy person.

$$\text{F1-Score}_{\text{Choroiditis}} = 2 \times \frac{\text{Precision}_{\text{Choroiditis}} \times \text{Recall}_{\text{Choroiditis}}}{\text{Precision}_{\text{Choroiditis}} + \text{Recall}_{\text{Choroiditis}}} \quad (8)$$

The *Receiver Operating Characteristic curve (ROC)* [25] indicates the model's ability to distinguish between the labels at various threshold values graphically. It plots the correctly identified choroiditis cases (True Positive Rate) against the normal images incorrectly classified as choroiditis (False Positive Rate).

The *Area Under the Curve (AUC)* indicates the model's ability to differentiate between choroiditis and normal cases across all the classification thresholds. AUC is a scalar value between 0 and 1, calculated as the area under the two-dimensional region of the ROC curve. A higher AUC means that the model can detect choroiditis cases effectively while reducing the misclassification of healthy individuals.

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(x)) \, dx \quad (9)$$

B. Hyperparameters for Experiment 1

Table 22. Hyperparameter details for reproducibility of Experiment 1.

Classifier	Hyperparameters
SVM	kernel='linear', C=1.0, gamma='scale', probability=True, random_state=42
Random Forest	n_estimators=100, criterion='gini', max_depth=None, random_state=42
Naive Bayes	GaussianNB (deterministic; no tuneable hyperparameters)
Decision Tree	criterion='gini', max_depth=None, random_state=42
XGBoost	max_depth=6, learning_rate=0.3, n_estimators=100, n_jobs=1, random_state=42
LightGBM	num_leaves=31, learning_rate=0.1, n_estimators=100, n_jobs=1, random_state=42
CatBoost	iterations=100, depth=4, loss_function='Logloss', eval_metric='AUC', thread_count=1, random_seed=42

C. Hyperparameters for Experiment 2 and Results of Remaining Models

Table 23. Hyperparameter details for reproducibility of Experiment 2.

Training	Early Stopping	Data	Model Selection
SGD (lr=0.001, momentum=0.9)	Patience: 10	Input: 224×224 (299×299 for Inception-V3)	Criterion: Lowest Val Loss
Cross-Entropy Loss	Delta (δ): 0.01	Split: 159/34/34 (train/val/test)	Configs: Feature Extraction, Scratch, Fine Tuning
Batch Size: 8	Monitor: Val Loss	Augmentation: None	Classes: 2
Max Epochs: 30		Preprocessing: Resize, ToTensor, ImageNet Normalize	Seed: 0

Tables 24 and 25 show the classification performance and computational efficiency of the nine architectures not included in the main paper. The following observations supplement the discussion in Section 4 (Experiment 2).

The scratch configuration reveals a clear correlation between the model's architecture and its training ability on the small choroiditis dataset. The VGG family (88.24% test accuracy) and ResNet family (ResNet-18: 85.29%, ResNet-50: 82.35% test accuracy) of models performed better than the lighter models in the scratch mode. This is possible because these models are able to learn even from random initialization due to their large, redundant convolutional capacity. Their straightforward stacked convolutional structures provide sufficient features from only

159 training images. DenseNet-121 (82.35%) exhibited similar performance, benefiting from its dense connectivity pattern that helps facilitate gradient flow during scratch training.

Table 24. Classification Performance of Remaining Transfer Learning Architectures.

Model	Configuration	Best Train Accuracy (%)	Average Train Accuracy (%)	Best Validation Accuracy (%)	Average Validation Accuracy (%)	Test Accuracy (%)	Training Time
ShuffleNet	Feature Extraction	61.01	52.48	67.65	55.06	52.94	8m 20s
ShuffleNet	Scratch	71.70	62.07	67.65	55.70	44.12	7m 25s
ShuffleNet	Fine Tuning	64.78	53.98	79.41	60.00	61.76	13m 59s
SqueezeNet1.0	Feature Extraction	94.97	89.41	97.06	96.08	94.12	4m 8s
SqueezeNet1.0	Scratch	57.86	52.72	64.71	55.35	61.76	7m 48s
SqueezeNet1.0	Fine Tuning	100.00	95.89	97.06	95.14	82.35	18m 41s
MobileNetV2	Feature Extraction	96.86	89.26	97.06	93.04	100.00	16m 14s
MobileNetV2	Scratch	79.87	65.35	67.65	51.34	52.94	8m 17s
MobileNetV2	Fine Tuning	100.00	95.34	97.06	95.75	97.06	18m 11s
MobileNetV3-Large	Feature Extraction	99.37	93.67	97.06	91.28	94.12	11m 45s
MobileNetV3-Large	Scratch	76.73	65.75	52.94	52.94	52.94	6m 20s
MobileNetV3-Large	Fine Tuning	100.00	96.70	97.06	92.82	91.18	8m 51s
AlexNet	Feature Extraction	90.57	82.79	94.12	88.50	76.47	5m 7s
AlexNet	Scratch	52.83	52.83	52.94	52.94	52.94	8m 6s
AlexNet	Fine Tuning	100.00	92.32	94.12	87.4	91.18	10m 51s
DenseNet-121	Feature Extraction	98.74	92.01	97.06	92.59	88.24	23m 50s
DenseNet-121	Scratch	91.82	81.73	79.41	72.76	82.35	43m 5s
DenseNet-121	Fine Tuning	100.00	96.85	97.06	95.00	91.18	32m 40s
ResNet-50	Feature Extraction	96.86	91.85	97.06	96.57	88.24	29m 11s
ResNet-50	Scratch	84.91	67.57	85.29	65.10	82.35	66m 14s
ResNet-50	Fine Tuning	100.00	96.32	100.00	94.41	91.18	38m 58s
VGG-11	Feature Extraction	89.94	82.34	97.06	92.53	94.12	18m 3s
VGG-11	Scratch	93.08	73.63	85.29	66.37	88.24	94m 22s
VGG-11	Fine Tuning	100.00	95.79	97.06	95.70	94.12	45m 59s
VGG-19	Feature Extraction	88.68	81.92	97.06	93.63	94.12	33m 59s
VGG-19	Scratch	88.05	67.09	76.47	59.44	88.24	119m 4s
VGG-19	Fine Tuning	100.00	94.20	97.06	91.81	88.24	120m 28s

Table 25. Computational Efficiency of Remaining Transfer Learning Architectures.

Model	Configuration	Params (M)	Size (MB)	FLOPs (GFLOP)	Memory (MB)	Inference (ms/image)
ShuffleNet	Feature Extraction	1.26	4.95	0.288	532.26	28.31
ShuffleNet	Scratch	1.26	4.95	0.288	528.61	25.66
ShuffleNet	Fine Tuning	1.26	4.95	0.288	532.14	22.25
SqueezeNet1.0	Feature Extraction	0.74	2.83	1.465	526.29	46.23
SqueezeNet1.0	Scratch	0.74	2.83	1.465	528.45	78.85
SqueezeNet1.0	Fine Tuning	0.74	2.83	1.465	526.83	74.04
MobileNetV2	Feature Extraction	2.23	8.73	0.599	545.35	46.12
MobileNetV2	Scratch	2.23	8.73	0.599	545.50	51.04
MobileNetV2	Fine Tuning	2.23	8.73	0.599	544.32	45.02
MobileNetV3-Large	Feature Extraction	4.2	16.23	0.431	545.97	38.18
MobileNetV3-Large	Scratch	4.2	16.23	0.431	545.69	39.70
MobileNetV3-Large	Fine Tuning	4.2	16.23	0.431	554.49	19.90
AlexNet	Feature Extraction	57.01	217.49	1.42	739.92	48.84
AlexNet	Scratch	57.01	217.49	1.42	734.42	58.66
AlexNet	Fine Tuning	57.01	217.49	1.42	735.83	68.55
DenseNet-121	Feature Extraction	6.96	27.11	5.666	585.15	183.80
DenseNet-121	Scratch	6.96	27.11	5.666	577.27	219.51
DenseNet-121	Fine Tuning	6.96	27.11	5.666	587.63	163.84
ResNet-50	Feature Extraction	23.51	89.99	8.174	657.08	206.11
ResNet-50	Scratch	23.51	89.99	8.174	642.61	218.94
ResNet-50	Fine Tuning	23.51	89.99	8.174	653.16	202.40
VGG-11	Feature Extraction	128.77	491.24	15.21	1052.88	335.42
VGG-11	Scratch	128.77	491.24	15.21	1039.57	336.93
VGG-11	Fine Tuning	128.77	491.24	15.21	1040.07	375.71
VGG-19	Feature Extraction	139.58	532.46	39.256	1080.27	850.83
VGG-19	Scratch	139.58	532.46	39.256	1080.2	547.38
VGG-19	Fine Tuning	139.58	532.46	39.256	1114.75	908.06

In contrast, the lightweight architectures; MobileNetV2 (52.94%), MobileNet-V3-Small (52.94%), and MobileNetV3-Large (52.94%) performed as if making a chance guess in the scratch configuration. This is in tandem with their architecture that relies on depth-wise separable convolutions and inverted residual blocks. Their small

structure lacks the redundancy required to learn discriminative features from scratch on limited data.

Notably, AlexNet also failed in the scratch configuration (52.94%) despite having 57.01M parameters, the fourth largest among all the architectures tested. This phenomenon is due to AlexNet's parameter distribution. The majority of AlexNet's parameters are in fully connected layers rather than convolutional layers, giving it very limited feature extraction capacity relative to its total parameter count.

ShuffleNet performed the worst among all architectures, achieving only 61.76% even in its best configuration (fine-tuning) and dropping to 52.94% under feature extraction. This is likely because of the architecture relying on group convolutions and shuffle operations that limit how information flows between channels. While this design works well for large datasets like ImageNet, it struggles to capture the subtle differences between choroiditis and normal fundus images when trained on limited data.

D. Hyperparameters and Augmentation for Experiment 3

For experiment 3, all the classifiers use the same hyperparameters as Experiment 1, listed in Table 22. A global random seed of 42 is applied across all libraries (Python, NumPy, PyTorch) for deterministic reproducibility. The augmentation techniques used during training are listed in Table 26.

Table 26. Augmentation Details.

Augmentation Function	Parameters
Horizontal Flip	Swap left-to-right positions.
Vertical Flip	Swap top-to-bottom positions.
Saturation	Factor $\in [0.8, 1.5]$ (HSV space)
Gaussian Noise	Standard deviation ≈ 10.2
Rotation	Random angle $\in [-20^\circ, 20^\circ]$
Per-Channel Brightness	Factor per channel $\in [0.8, 1.2]$
Linear Contrast	$\alpha \in [0.85, 1.25]$
Random Crop followed by Resize	Crop border $\in [0, 10\%]$. Resize to 224×224 using bilinear interpolation
Horizontal Shear	angle $\in [-10^\circ, 10^\circ]$
Vertical Shear	angle $\in [-10^\circ, 10^\circ]$
Elastic Transform	$\alpha = 40$, $\sigma = 5$
Perspective Transform	Scale $\in [0.01, 0.05]$
Gamma Contrast	Gamma $\in [0.7, 1.5]$
Gaussian Blur	Sigma $\in [0.5, 1.5]$
Sharpen	Strength $\in [0.5, 1]$. $\sigma = 1.0$
CLAHE	clipLimit = 2.0. tileGridSize = 8×8 .

E. Error Analysis of all ML Classifiers on Test Data

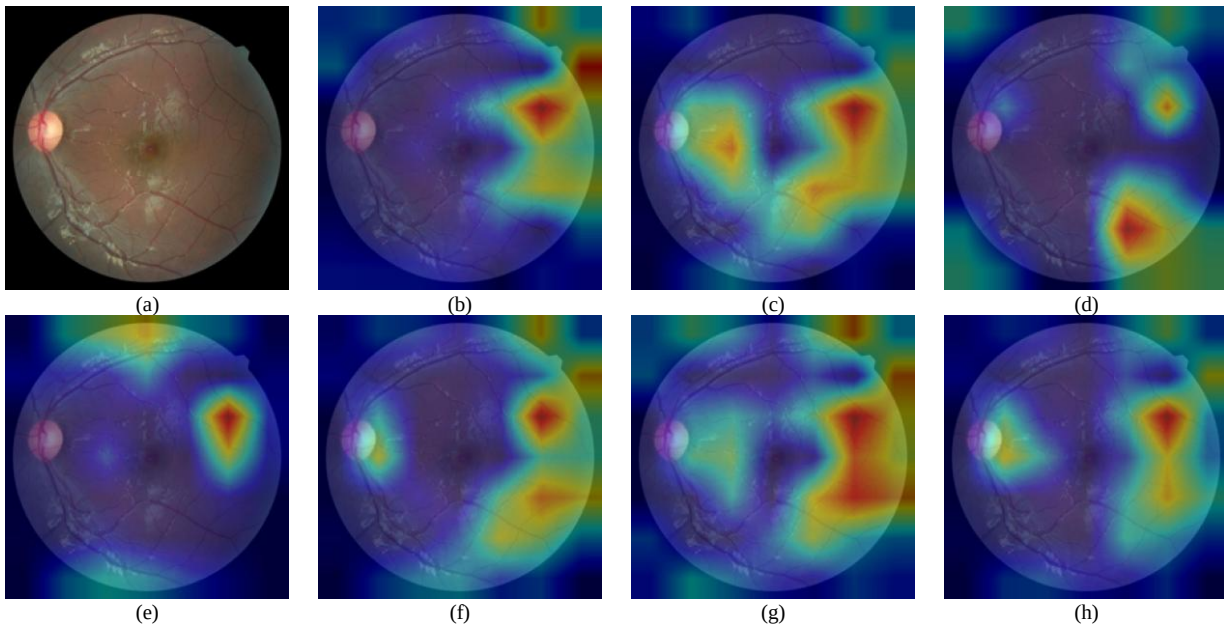


Fig. 10. Universal failure case across all seven classifiers. (a) the original Normal-class image. Grad-CAM overlay produced by each classifier: (b) SVM (c) Random Forest (d) Naive Bayes (e) Decision Tree (f) XGBoost (g) LightGBM (h) CatBoost.

F. Hyperparameters for Experiment 4

Table 27. Hyperparameter details for reproducibility of Experiment 4.

Architecture	Training	Episodic Setup	Evaluation
Backbone: MobileNet-V3-Small (ImageNet, frozen)	Adam (lr=0.001)	N-Way: 2	Evaluation Episodes: 600/setting
Feature Dimension: 1024	Max Episodes: 1,000	K-Shot: 5	K-Shot Settings: 1, 5, 10, full
Projection Head: Linear(1024→256) → BatchNorm → ReLU → Linear (256→128) → L2-Normalization	Actual Episodes: 250 (early stopping triggered)	N-Query: 15/class	Evaluation N-Query: 15/class
Trainable Parameters: 295,808	Early Stop Patience: 100	Validation Frequency: Every 50 episodes	Model Size: 7.1 MB
Frozen Parameters: 1,517,856	Monitor: Validation Loss	Validation Episodes: 200	FLOPs: 0.111 GFLOP
Device: CPU	Seed: 42	Classes: 2	Training Time: ~4 s

References

- [1] U. Krishna, D. Ajanaku, A. K. Denniston, and T. Gkika, "Uveitis: a sight-threatening disease which can impact all systems," *Postgrad. Med. J.* 93, 766 (2017).
- [2] C. E. Willoughby, D. Ponzin, S. Ferrari, A. Lobo, K. Landau, and Y. Omid, "Anatomy and physiology of the human eye: effects of mucopolysaccharidoses disease on structure and function—a review," *Clin. Exp. Ophthalmol.* 38, 2 (2010).
- [3] R. Geetha and K. Tripathy, "Chorioretinitis," in *StatPearls* [Internet], StatPearls Publishing, 2022.
- [4] L. F. Nakayama et al., "Artificial intelligence in uveitis: A comprehensive review," *Surv. Ophthalmol.* 68, 669 (2023).
- [5] J. J. González-López et al., "Development and validation of a Bayesian network for the differential diagnosis of anterior uveitis," *Eye* 30, 865 (2016).
- [6] A. M. Mutawa and M. A. Alzuwawi, "Multilayered rule-based expert system for diagnosing uveitis," *Artif. Intell. Med.* 99, 101691 (2019).
- [7] Y. Jamilloux et al., "Development and validation of a Bayesian network for supporting the etiological diagnosis of uveitis," *J. Clin. Med.* 10, 3398 (2021).
- [8] I. Mellakh, T. Wang, R. Jacquot, N. Moalla, and R. Youssef-Douss, "Enhancing the Etiological Diagnosis of Uveitis with Artificial Intelligence: A Predictive Approach," 2023 15th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 1 (2023).
- [9] B. Trusko et al., "The standardization of uveitis nomenclature (SUN) project," *Methods Inf. Med.* 52, 259 (2013).
- [10] A. A. Okada and D. A. Jabs, "The standardization of uveitis nomenclature project: the future is here," *JAMA Ophthalmol.* 131, 787 (2013).
- [11] Y. Li, C. Lowder, X. Zhang, and D. Huang, "Anterior chamber cell grading by optical coherence tomography," *Invest. Ophthalmol. Vis. Sci.* 54, 258 (2013).
- [12] S. Sharma et al., "Automated analysis of anterior chamber inflammation by spectral-domain optical coherence tomography," *Ophthalmology* 122, 1464 (2015).
- [13] E. Baghdasaryan et al., "Analysis of ocular inflammation in anterior chamber—involving uveitis using swept-source anterior segment OCT," *Int. Ophthalmol.* 39, 1793 (2019).
- [14] M. A. Sorkhabi et al., "Assessment of anterior uveitis through anterior-segment optical coherence tomography and artificial intelligence-based image analyses," *Transl. Vis. Sci. Technol.* 11, 7 (2022).
- [15] C. L. Passaglia, T. Arvaneh, E. Greenberg, D. Richards, and B. Madow, "Automated method of grading vitreous haze in patients with uveitis for clinical trials," *Transl. Vis. Sci. Technol.* 7, 10 (2018).
- [16] S. Haggag et al., "An automated CAD system for accurate grading of uveitis using optical coherence tomography images," *Sensors* 21, 5457 (2021).
- [17] R. Parra et al., "A trust-based methodology to evaluate deep learning models for automatic diagnosis of ocular toxoplasmosis from fundus images," *Diagnostics* 11, 1951 (2021).
- [18] R. Parra et al., "Automatic Diagnosis of Ocular Toxoplasmosis from Fundus Images with Residual Neural Networks," *Stud. Health Technol. Inform.* 281, 173 (2021).
- [19] S. S. Alam, S. B. Shuvo, S. N. Ali, F. Ahmed, A. Chakma, and Y. M. Jang, "Benchmarking Deep Learning Frameworks for Automated Diagnosis of Ocular Toxoplasmosis: A Comprehensive Approach to Classification and Segmentation," *IEEE Access*, vol. 12, pp. 22759–22775. (2024).
- [20] J. Sun et al., "Identifying mouse autoimmune uveitis from fundus photographs using deep learning," *Transl. Vis. Sci. Technol.* 9, 59 (2020).
- [21] S. Raj and A. K. Keshri, "Evaluation of machine learning classifiers for choroiditis prediction from fundus images," *Grenze Int. J. Eng. Technol.*, vol. 9, no. 2, pp. 2269–2274, (2023).
- [22] Scikit-learn developers, "User guide," Scikit-learn. [Online]. Available: https://scikit-learn.org/stable/user_guide.html. Accessed: Jan. 2024.
- [23] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [24] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," *Proc. 18th ACM Int. Conf. Multimedia*, pp. 1485–1488, 2010.

- [25] A. D. Forbes, "Classification-algorithm evaluation: Five performance measures based on confusion matrices," *J. Clin. Monit.* 11, 189 (1995).
- [26] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.* 27, 861 (2006).
- [27] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Comput. Sci.* 2, 420 (2021).
- [28] C. M. Bishop, "Pattern recognition and machine learning," Springer, 2, 645 (2006).
- [29] T. Hastie, R. Tibshirani, and J. H. Friedman, "The elements of statistical learning: data mining, inference, and prediction," Vol. 2, 1-758, Springer, New York (2009).
- [30] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 785 (2016).
- [31] G. Ke et al., "Lightgbm: A highly efficient gradient boosting decision tree," *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [32] L. Prokhorenkova et al., "CatBoost: unbiased boosting with categorical features," *Adv. Neural Inf. Process. Syst.* 31 (2018).
- [33] H. E. Kim et al., "Transfer learning for medical image classification: A literature review," *BMC Med. Imaging* 22, 69 (2022).
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 770 (2016).
- [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.* 25 (2012).
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556* (2014).
- [37] F. N. Iandola et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size," *arXiv preprint arXiv:1602.07360* (2016).
- [38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 4700 (2017).
- [39] C. Szegedy et al., "Rethinking the inception architecture for computer vision," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2818 (2016).
- [40] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861* (2017).
- [41] A. Howard et al., "Searching for mobilenetv3," *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 1314 (2019).
- [42] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 6848 (2018).
- [43] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *Int. Conf. Mach. Learn.*, 6105 (2019).
- [44] C. Szegedy et al., "Going deeper with convolutions," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1 (2015).
- [45] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923. (1998).
- [46] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605. (2008).
- [47] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.* 20, 273 (1995).
- [48] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods," Cambridge University Press (2000).
- [49] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 618–626. (2017).
- [50] O. Cardozo et al., "Dataset of fundus images for the diagnosis of ocular toxoplasmosis," *Data Brief*, vol. 48, p. 109056. (2023).
- [51] E. Pachetti and S. Colantonio, "A systematic review of few-shot learning in medical imaging," *Artif. Intell. Med.*, vol. 156, p. 102949. (2024)

Authors' Profiles



Srishiti Raj is a research scholar and is pursuing a PhD in the Department of Computer Science and Engineering at Birla Institute of Technology, Ranchi, India. She has done B. Tech. and M. Tech in Computer Science and Engineering from Kalinga Institute of Industrial Technology, India. Her areas of interest are Digital Image Processing and Neural Networks.



Anup Kumar Keshri is an assistant professor at Birla Institute of Technology, Ranchi, India. His qualifications are a Ph.D. in Computer Science and Engineering, an ME and a BE in Software Engineering. He has a total of 14 years of teaching experience and 16 years of research experience. He has a total of 20 publications in highly esteemed journals and conferences.

How to cite this paper:

Srishiti Raj, Anup Kumar Keshri, "Building a Lightweight and Time-Efficient Hybrid Model for Choroiditis Detection in Fundus Images: A Systematic Approach", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.18, No.4, pp. 1-24, 2026. DOI:10.5815/ijigsp.2026.04.01