Retinal Fundus Multi-Disease Image Classification using Hybrid CNN-Transformer-Ensemble Architectures

Deependra Singh[1*], Saksham Agarwal[2*], Subhankar Mishra[3,$]

[1]School of Physical Science, [2]School of Biological Science, [3]School of Computer Science, National Institute of Science Education and Research, Bhubaneswar, India
{[1]deependra.singh, [2]saksham.agarwal, [3]smishra}@niser.ac.in
[$] Corresponding author: Subhankar Mishra, smishra@niser.ac.in

Abstract. Our research is motivated by the urgent global issue of a large population affected by retinal diseases, which are evenly distributed but underserved by specialized medical expertise, particularly in non-urban areas. Our primary objective is to bridge this healthcare gap by developing a comprehensive diagnostic system capable of accurately predicting retinal diseases solely from fundus images. However, we faced significant challenges due to limited, diverse datasets and imbalanced class distributions. To overcome these issues, we have devised innovative strategies. Our research introduces novel approaches, utilizing hybrid models combining deeper Convolutional Neural Networks (CNNs), Transformer encoders, and ensemble architectures sequentially and in parallel to classify retinal fundus images into 20 disease labels. Our overarching goal is to assess these advanced models' potential in practical applications, with a strong focus on enhancing retinal disease diagnosis accuracy across a broader spectrum of conditions. Importantly, our efforts have surpassed baseline model results, with the C-Tran ensemble model emerging as the leader, achieving a remarkable model score of 0.9166, surpassing the baseline score of 0.9. Additionally, experiments with the IEViT model showcased equally promising outcomes with improved computational efficiency. We've also demonstrated the effectiveness of dynamic patch extraction and the integration of domain knowledge in computer vision tasks. In summary, our research strives to contribute significantly to retinal disease diagnosis, addressing the critical need for accessible healthcare solutions in underserved regions while aiming for comprehensive and accurate disease prediction.

Keywords: Retinal disease classification, CNN, Classification Transformers, IEViT, Ensemble, Hybrid

---

[*] The first two authors contributed equally to this work and should be regarded as Joint First Authors

1   Introduction

The human retina serves a critical role in the visual system by converting incoming light signals into electrical/chemical signals that are subsequently processed by the brain. Unfortunately, there are numerous retinal diseases that can lead to irreversible damage, resulting in permanent vision loss. Timely diagnosis and treatment are therefore essential to manage and prevent further deterioration. However, the availability of quality eye care can vary significantly within countries and regions, leading to limited access, particularly for individuals residing in remote areas [1].

Addressing this issue is crucial to ensure equitable access to quality eye care. The increasing prevalence of specific retinal diseases emphasizes the importance of early detection and efficient diagnostic systems. For instance, glaucoma is predicted to impact 111.8 million individuals by 2040 [2] predicted. [3] highlights that out of the 415 million people worldwide living with diabetes in 2015, 145 million experienced diabetic retinopathy (DR). Approximately 6.2 million individuals globally are affected by age-related macular degeneration (AMD) [4]. Various retinal diseases, including age-related macular degeneration (AMD), exhibit distinct pathophysiological features that can serve as valuable sources for the development of artificial intelligence (AI) diagnostic tools. Pathologies such as drusen or choroidal neovascularization (CNV), subretinal hemorrhage, and vascular leakage in dry or neovascular AMD provide potential photographic markers for the development of AI-based diagnostic algorithms [5]. These markers and other diagnostic indicators enable ophthalmologists to identify and differentiate between different retinal conditions [6]. The development of image-based diagnostic systems holds significant promise for addressing these challenges.

In recent years, there has been a growing interest in AI-based diagnostic systems that utilize fundus images. Initially, these systems targeted specific retinal diseases, such as diabetic retinopathy [7] and AMD [8]. However, a major drawback of these approaches is their narrow focus on individual diseases or disorders. Patients often experience multiple retinal diseases in real-world scenarios, rendering these software solutions less effective [9].To address this challenge, there has been a shift towards developing multi-disease classification systems for retinal diseases [10]. Prior studies have investigated deep-learning methods, mostly Convolutional neural networks (CNNs), both standalone [11] [12], or ensembled [13] [14], to diagnose retinal diseases using fundus images. Notably, [10] used CNNs to classify retinal diseases with impressive accuracies. [11] explored transfer learning, achieving 93.58% accuracy. Ensemble approaches, like [13], attained an AUC score of 0.97, even for limited labels. While CNNs have been effective, recent advancements in medical diagnosis have involved transformer-based models in increasing accuracy and targeting more disease labels [15] [16]. However, challenges persist in achieving better classification accuracy, enhancing model generalization and robustness, and deploying models in resource-limited environments; knowing the availability of large, properly labeled, and balanced datasets is scarce [11].

These works showcase deep learning's impact, though dataset limitations and lower accuracy on certain labels warrant attention. Our research seeks to pioneer a hybrid model that addresses these challenges, providing a scalable solution for comprehensive retinal disease diagnosis."

1.1   Contribution

To build our hybrid models combining CNN and transformers, we selected two baseline models from previous works: C-Tran [15], which was previously been tested on a fundus image dataset with 20 labels, and IEViT [16], which was previously been tried on chest X-ray images. Our work has the following contribution to current research:

1. We developed six novel models that combine deeper CNN backbones, transformer encoders, and ensemble architectures.
2. All six variants were tested on the dataset, showing improved performances over the baseline models.
3. Utilisation of the domain knowledge to highlight more relevant image regions with importance patch extraction.
4. Improvement in the general transformer-based learning using the IE concept.
5. For the purpose of reproducibility, the source code files can be accessed on the GitHub Repository.[1]

2   Dataset

Among the three publicly available datasets shown in the table below, we chose the MuReD dataset [17]. This decision was influenced by its utilization in the C-Tran paper we aimed to reproduce for establishing baselines of our model [15]. MuReD, a refined variant of the highly imbalanced RFMiD dataset, was preferred due to its relevance. In the future, we intend to incorporate the RFMiD 2.0 and JSIEC datasets in our analysis as it is the latest dataset out of three and encompasses all the essential labels of MuReD.

Table 1. List of Publicly available datasets

Dataset Name	Train	Validation	Test	Total	Labels
RFMiD dataset [18]	1920	640	640	3200	46
MuReD dataset [17]	1766	442	-	2208	20
RFMiD 2.0 [19]	516	344	344	860	49
JSIEC [12]	-	-	-	1000	39

---

[1] https://github.com/smlab-niser/23retinald

Table 2. List of Diseases to be detected [15]

Acronym	Full Name	Training	Validation	Total
DR	Diabetic Retinopathy	396	99	495
NORMAL	Normal Retina	395	98	493
MH	Media Haze	135	34	169
ODC	Optic Disc Cupping	211	52	263
TSLN	Tessellation	125	31	156
ARMD	Age-Related Macular Degeneration	126	32	158
DN	Drusen	130	32	162
MYA	Myopia	71	18	89
BRVO	Branch Retinal Vein Occlusion	63	16	79
ODP	Optic Disc Pallor	50	12	62
CRVO	Central Retinal Vein Oclussion	44	11	55
CNV	Choroidal Neovascularization	48	12	60
RS	Retinitis	47	11	58
ODE	Optic Disc Edema	46	11	57
LS	Laser Scars	37	9	46
CSR	Central Serous Retinopathy	29	7	36
HTR	Hypertensive Retinopathy	28	7	35
ASR	Arteriosclerotic Retinopathy	26	7	33
CRS	Chorioretinitis	24	6	30
OTHER	Other Diseases	209	52	261

3   Preprocessing of images

3.1   Sampling Techniques

Several sampling techniques have been employed in the literature, including LP ROS [15], Dynamic Random Sampling [12], and Weighted Random Sampler. We focused on two specific sampling techniques, Weighted Random Sampling and LP-ROS (Label Powerset Random Oversampling), listed in table 3. While the CTran paper used LP ROS for addressing the class imbalance, the Weighted Random Sampling technique, implemented in the code provided, demonstrated better performance. The decision to utilize Weighted Random Sampling was based on empirical evidence suggesting its efficacy in handling class imbalance within the CTran model.

Table 3. Sampling techniques used

Sampling Technique	Description
Weighted Random Sampling	Addresses class imbalance by assigning higher weights to under-represented classes during training.
LP-ROS (10%)	An oversampling technique based on the Label Powerset transformation, used to handle class imbalance. Here, the minority class samples are augmented by adding 10% more samples to balance the class distribution.

3.2 Augmentation

In the context of image data preprocessing, augmentation techniques are commonly used to increase the diversity and variability of the training dataset. These techniques introduce modifications or transformations to the original images, allowing the model to learn from a wider range of data patterns. These augmentation techniques are applied as part of the data transformation pipeline using the torchvision.transforms module. They contribute to enhancing the model's robustness and ability to generalize patterns from the training data. The following table summarizes the augmentation techniques applied in this work:

Technique	Description
Random Horizontal Flip	Randomly flips the image horizontally with a probability of 0.5.
Random Vertical Flip	Randomly flips the image vertically with a probability of 0.5.
Random Rotation	Randomly rotates the image by 15 degrees.
Colour Jitter	Adjusts the brightness, contrast, saturation, and hue of the image with values of 0.4, 0.4, 0.4, and 0.1 respectively.
Resize	Resizes the image to the specified size.
ToTensor	Converts the image to a tensor.
Normalize	Normalizes the image tensor using pre-defined mean and standard deviation values of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225], respectively.

4 Models

This section provides an overview of the models used in our study and highlights their unique contributions and variations.

4.1 Deep Convolutional Neural Networks

We explored various deep CNN architectures as backbone models to identify the most effective model configuration. Our selection encompassed ResNet152d, DenseNet121, DenseNet201, and EfficientNetV2Small. These models were pre-trained on ImageNet and served as feature extractors, generating feature embeddings that were subsequently processed through transformer layers for classification. By evaluating different CNN architectures, we sought to capitalize on their unique feature extraction capabilities. Additionally, we utilized denser versions of DenseNet and ResNet compared to those employed in [15] (DenseNet121), aiming to enhance the performance of the baseline model. The performance of these architectures on our dataset is given in table 4.

Table 4. Performance of each CNN architecture on the dataset

Model	Loss	ML F1	ML AUC
DenseNet121	2.121	0.276	0.951
DenseNet201[a]	1.692	0.272	0.959
ResNet152d	1.815	0.410	0.947
EfficientNetV2-S	1.881	0.214	0.958

[a] CNNs in bold indicate the top two best-performing architectures

4.2   Classification Transformers C-Tran

The C-Tran model employed in this project combines DenseNet201 and ResNet152d backbones with a Transformer encoder, facilitating good performance for multi-label classification. The key components of the C-Tran model encompass feature extraction, positional encoding, and label prediction.

For each input image, visual features are extracted using the chosen CNN backbone, resulting in a feature embedding $\mathbf{X}_{\text{visual}}$ of dimensions $n \times b \times d$, where n is the number of classes, b batch size, and d is the embedding dimension. To incorporate positional information, the model utilizes positional encoding, denoted as $\mathbf{PE}_{2D}$, calculated using sine and cosine functions based on position in the height and width dimensions:

$$\mathbf{PE}_{2D}(i, j, k) = \begin{cases} \sin\left(\frac{i}{10000^{2k/d}}\right) & \text{if } k \text{ is even} \\ \cos\left(\frac{j}{10000^{2k/d}}\right) & \text{if } k \text{ is odd} \end{cases} \tag{1}$$

where i, j represents the position in the height and width dimensions, k represents the dimension of the positional encoding, and d is the embedding dimension of the visual features.

The 2D positional encoding matrix, $\mathbf{PE}_{2D}$, is element-wise added to the visual features after reducing it linearly to the dimensions of $\mathbf{X}_{\text{visual}}$, enriching spatial awareness:

$$\mathbf{X}_{\text{encoded}} = \mathbf{X}_{\text{visual}} + \mathbf{PE}_{2D} \tag{2}$$

where $\mathbf{X}_{\text{encoded}}$ represents the spatially enhanced visual features.

The Transformer encoder captures feature-label interactions using a self-attention mechanism, fostering the learning of diverse dependencies between features and labels. The final label predictions, $\mathbf{Y}_{\text{pred}}$, are generated using independent feed-forward networks for each label embedding:

$$\mathbf{Y}_{\text{pred}} = \sigma(\mathbf{E}_{\text{label}} \cdot \mathbf{W} + \mathbf{b}) \tag{3}$$

where σ represents the sigmoid activation function, $\mathbf{W}$ represents learned weights, and $\mathbf{b}$ represents biases. The C-Tran algorithm synergizes feature extraction, positional encoding, and self-attention mechanisms within a Transformer architecture to achieve state-of-the-art multi-label classification performance.

4.3   Ensemble Model

The ensemble model combines the predictions from two distinct backbone models, DenseNet201 and ResNet152D, utilizing a weighted ensemble strategy. The ensemble model is configured with two variants: one that employs separate transformer layers for each backbone model and another that utilizes a single transformer layer for the combined embeddings from both.

Variant 1: Separate Transformer Layers: In this variant, the input image undergoes two separate C-Tran paths. First, it passes through the DenseNet201 backbone model, generating feature embeddings. These embeddings then traverse through a set of transformer layers, as in C-Tran. The sigmoid function is applied to the output to produce the raw predictions denoted as $\mathbf{P}_{\mathrm{dn}}$. Simultaneously, the input image also traverses the ResNet152d backbone model, producing feature embeddings that pass through their respective transformer layers, and raw predictions are generated via sigmoid denoted as $\mathbf{P}_{\mathrm{rn}}$.

We employ weighted ratios, denoted as a and b, to combine the predictions from both backbone models. These ratios determine the relative importance of each backbone model's predictions. The combined output embeddings, $\mathbf{P}_{\mathrm{c}}$, are calculated as follows:

$$\mathbf{P}_{\mathrm{c}} = a \cdot \mathbf{P}_{\mathrm{dn}} + b \cdot \mathbf{P}_{\mathrm{rn}} \tag{4}$$

To form the combined loss L, the weighted average of these individual losses is computed:

$$L = a \cdot L_{\mathrm{dn}} + b \cdot L_{\mathrm{rn}} \tag{5}$$

This combined loss, L, is then used for gradient descent optimization and model parameter updates. Additionally, evaluation metrics are computed using the sigmoid-activated predictions to assess model performance.

Variant 2: Single Transformer Layer: In this variant, the input image is passed through both the DenseNet201 and ResNet152D backbone models. The feature embeddings from both backbones are then combined and fed into a single transformer layer. The output embeddings from both backbone models, denoted as $\mathbf{X}_{\mathrm{dn}}$ and $\mathbf{X}_{\mathrm{rn}}$, are concatenated along the n dimension (dimension 0), creating a unified feature embedding, $\mathbf{X}_{\mathrm{c}}$:

$$\mathbf{X}_{\mathrm{c}} = [\mathbf{X}_{\mathrm{dn}}, \mathbf{X}_{\mathrm{rn}}] \tag{6}$$

where [] operation signifies concatenation. This combined feature embedding, $\mathbf{X}_{\mathrm{c}}$, is subsequently processed by a single C-Tran transformer encoder layers, capturing the contextual information from both backbone models. The remaining steps, such as positional encoding, projection layer, and classification layer, remain the same as in the previous variant. This variant allows the model to leverage the strengths of both backbone models in a unified transformer layer.

Algorithm 1 C-Tran Ensemble Variants Forward Pass

Require: x: Input tensor of shape [batch, channel, height, width]
Ensure: p: Output tensor of shape [batch, num_classes]
 1: Initialize network components
 2: $x_1 \leftarrow \text{Backbone1}(x)$
 3: $x_2 \leftarrow \text{Backbone2}(x)$
 4: **if** Variant 1 **then**
 5:    Add positional encodings: $x_1, x_2 \leftarrow (x_1 + P_1, \ x_2 + P_2)$
 6:    $p_1 \leftarrow \text{MLP}_1(\text{Transformer}_1(x_1))$
 7:    $p_2 \leftarrow \text{MLP}_2(\text{Transformer}_2(x_2))$
 8:    $p = a \cdot p_1 + b \cdot p_2$
 9: **else**
10:    $x \leftarrow \text{Concatenate}(x_1, x_2)$
11:    Add positional encoding $P : x \leftarrow x + P$
12:    $p \leftarrow \text{MLP}(\text{Transformer}(x))$
13: **end if**
14: Train model and update weights
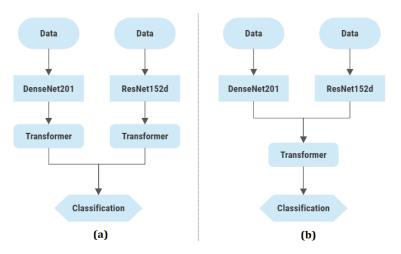15: **return** p

---

Fig. 1. Models Representation for (a) Variant 1, (b) Variant 2

These two variants provide different approaches for integrating multiple backbone models and transformer layers within the ensemble model, enabling the model to leverage diverse features and capture complex patterns in the input data.

4.4   IEViT Model

The proposed IEViT (Iterative Expansion Vision Transformer) model is an extension of the ViT (Vision Transformer) architecture, incorporating concepts inspired by the ResNet (Residual Network). It introduces an iterative process in which the original input image is iteratively added to the output of each Transformer encoder layer. To facilitate this, a convolutional block is designed in parallel with the ViT network. The CNN block processes the entire input image and generates an embedding of the image denoted as $\mathbf{x}_{img}$. This embedding is then concatenated to the output of each Transformer encoder layer, denoted as $\mathbf{z}_l$, thereby incorporating the complete image information throughout the encoding process.

The IEViT architecture comprises stacked 2D convolutional layers, followed by a 1D global maximum pooling layer within the CNN block. The resulting tensor is a feature embedding, $\mathbf{x}_{\mathrm{img}}$ of dimension D. Additionally, the image is divided into $N = \frac{H \cdot W}{P^2}$ patches, which are flattened to patches x_p of size $N \times (P^2 \cdot C)$. Subsequently, the patch embeddings z_l are created by mapping the patches to D dimensions using 2D convolution layers. Position embeddings denoted as $\mathbf{E}_{\mathrm{pos}}, E_{\mathrm{pos}} \in \mathbb{R}^{(N+1)\times D}$, are introduced to encode the positional information of each patch in the original image.

Firstly, the initial patch embedding $\mathbf{z}_{\mathrm{p}}$ and the positional encodings $\mathbf{E}_{\mathrm{pos}}$ are element-wise added. An optional class token might also be concatenated with z_p.

$$\mathbf{z}_0 = \mathbf{z}_{\mathrm{p}} + \mathbf{E}_{\mathrm{pos}} \tag{7}$$

where $\mathbf{z}_0$ represents the combined embeddings. The core iterative expansion process in IEViT occurs during the transformation of image embeddings through stacked Transformer encoder layers. At each layer l, the image embedding, $\mathbf{x}_{\mathrm{img}}$, is fused with the encoder output, $\mathbf{z}_l$, resulting in a combined representation denoted as $\mathbf{z}^{\hat{l}}$. This process is mathematically represented as:

$$\mathbf{z}^{\hat{l}} = [\mathbf{z}_l, \mathbf{x}_{\mathrm{img}}], \quad \text{where } \mathbf{z}^{\hat{l}} \in \mathbb{R}^{(N+1+l)\times D}. \tag{8}$$

The iterative expansion in IEViT enhances its capability to capture spatial information and maintain global context, contributing to superior performance across diverse computer vision tasks.

We propose two novel variants to enhance the patch extraction step in IEViT, leveraging domain knowledge of the 20 diseases targeted for classification [16]. These variants address the importance of prioritizing patches from near the center of the image, where crucial features are expected to be prominent for the specific disease labels we're targeting.

Variant 1- Unequal Patches: In this variant, we manually set the patch dimensions to be extracted from the image for forming the patch of embedding of dimension D. We extract patches with decreasing dimensions as we move closer to the center. This approach allocates more space to patches near the center, where crucial features are expected to be prominent. The patch dimensions, manually defined as 32x32, 16x16, and 8x8, progressively decrease towards the center.

Variant 2- Dynamic Patch Decomposer (DPD): This approach utilizes normalized learnable parameter importance weights for the initial 144 patches, each with a dimension of 32×32. The method dynamically determines which patches to subdivide based on their importance, using the formula for the new patch dimension (NPD):

$$NPD = 32 \times \left(\frac{w_{avg}}{w_i} \right)^k$$

where w_i represents the weight of the i-th patch, w_{avg} is the average weight, and k is a hyperparameter. Patches with $NPD > 32$ remain unchanged, while others are subdivided into dimensions $NPD \times NPD$ after rounding NPD to the nearest factor of 32. The new patches are extracted, and their embedding z_n is appended to the former patch embedding tensor z_l at the position of the original patches that were divided or merged, as represented by the equation:

$$z_l' = z_l \oplus z_n$$

A similar adjustment is applied to maintain positional encoding and adapt it for subdivided or merged patches. Specifically, the positional encoding is extended to accommodate new patches while preserving the existing ones, ensuring that spatial information is accurately represented throughout the encoding process. The positional encoding for the new patches is inserted at the position of the original patch in the encoding tensor. This extension is expressed by:

$$E_{pos}' = E_{pos} \oplus E_{new}$$

Additionally, the MLP (Multi-Layer Perceptron) head parameters are dynamically adjusted to handle the input changes due to patch subdivision and merging. Parameter sharing is employed to efficiently accommodate the new patches without a significant increase in model parameters, ensuring that the MLP weight matrix adapts to the varying input dimensions. Weight splicing is used when the patches are merged. This adaptive approach enhances the model's ability to capture and utilize original and subdivided patch information.

Algorithm 2 DPD Patch Division and Weight Handling

1: Calculate $NPD = 32 \times \left(\frac{w_{avg}}{w_i}\right)^k$
2: **for** patch in patch embedding z_0 **do**
3:     **if** $NPD > 32$ **then**
4:         *No patch subdivision required*
5:     **else**
6:         *Subdivide the patch into $NPD \times NPD$ sized patches*
7:         Extract the embeddings for the new patches
8:         $z'_l = z_l \oplus z_n$
9:         Adjust positional encoding: $E'_{pos} = E_{pos} \oplus E_{new}$
10:     **end if**
11: **end for**
12: **if** MLP input dimension increases **then**
13:     Expand MLP head parameters with weight sharing
14: **else if** MLP input dimension decreases **then**
15:     Slice MLP head parameters
16: **else**
17:     MLP head parameters remain unchanged
18: **end if**
19: Continue with standard processing

4.5   CTran Variants with IEViT Concept

In this subsection, we present two variants of the CTran model that incorporate the Iterative expansion concept from IEViT.

IECTe- Iterative Expansion C-Tran Ensembler: The IECTe variant combines ensemble model variant 2 with the IE concept by appending the reduced feature embeddings from the two CNN backbones x and y to each output of transformer layers. Concatenating feature embeddings along the *num_classes* dimension (dim = 0):

$$z_0 = [x, y]$$

$$\mathbf{z}^{\hat{l}} = [\mathbf{z}_l, \mathbf{z_0}]$$

By incorporating these additional feature embeddings, the model gains a holistic understanding of the input image from multiple perspectives. This enables the model to leverage the strengths of both ResNet201 and DenseNet201 architectures, leading to enhanced representation learning and improved performance. The iterative nature of the IEViT approach ensures that the network retains knowledge of the entire input image throughout the Transformer encoding process. By iteratively concatenating the feature embeddings to the Transformer output, the IECTe model continuously "remembers" the full image information, facilitating effective information fusion and promoting robust representations.

IeECT: Iterative Ensemble Expansion C-Tran- The IeECT variant focuses on reducing the overall model complexity for IECTe. In this variant, the feature embedding from DenseNet201 serves as the primary input to the Transformer ($z_0 = x$), while the feature embedding from ResNet152 is appended to the Transformer output after each layer.

$$\mathbf{z}^{\hat{l}} = [\mathbf{z}_l, \mathbf{y}]$$

By employing this design, the IeECT model benefits from the expressive power of DenseNet201, which captures intricate image details and relationships. Simultaneously, the inclusion of the ResNet152 feature embedding allows the model to incorporate information from a different architectural perspective, enriching the representations with complementary insights. Using the IEViT concept in IeECT ensures that the model retains knowledge of the original image throughout the encoding process while maintaining fewer parameters. [1]

Algorithm 3 IE pipeline Variants for CTran

1: Initialize network components, L transformer encoder layers
2: $x_1, x_2 \leftarrow \text{Backbone1}(x), \text{Backbone2}(x)$
3: **if** Variant 1 **then**
4:     $z_0, z \leftarrow \text{Concatenate}(x_1, x_2)$
5: **else**
6:     $z_0, z \leftarrow x_1, x_2$
7: **end if**
8: Add positional encoding $P : z_0 \leftarrow z_0 + P$
9: $z_1 \leftarrow \text{Transformer\_Layer}_0(z_0)$
10: **for** layer l in transformer encoder $l \neq 0$ **do**
11:     $z_l \leftarrow \text{Concatenate}(z_l, z)$
12:     $z_(l+1) \leftarrow \text{Transformer}(z_1)$
13: **end for**
14: **return** $p \leftarrow \text{MLP}(z_L)$
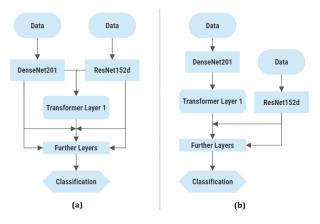

Fig. 2. Model Representation for:(a) IECTe, (b) IeECT

5   Model Training

5.1   Hyperparameters

Table 5. List of hyperparameters used for model optimization

Hyperparameter	Value
Number of Epochs	200
Batch Size	8, 12, 16, 32
Embedding Dimensions	960
Optimizer	AdamW
Scheduler	Cosine Annealing Warm Restarts, Learning Rate: 5e-5, Weight Decay: 1e-6
Loss Function	BCEwithLogitLoss
Transformer Layers	6, 12

5.2   Metrics

To evaluate the performance of our models, we used the same metrics as used in [15] for comparison on the same dataset. These metrics include F1-score, AUC, and mAP. ML mAP, ML F1, and ML AUC are calculated by averaging scores for each label, excluding the "NORMAL" label- having the AUC and F1-score represented by Bin AUC and Bin F1 metric.

$$\text{ML Score} = \frac{\text{ML mAP} + \text{ML AUC}}{2}$$

The Model Score metric evaluates the overall performance by considering both disease classification and normalcy detection.

$$\text{Model Score} = \frac{\text{ML Score} + \text{Bin AUC}}{2}$$

6   Results and Discussion

The results of our experiments are presented in Table 6. As expected, the deeper DenseNet201 model surpassed the baseline C-Tran model that used DenseNet161. The ensemble models with deeper CNN backbones also demonstrated improved scores. The Strong Densenet and Weak Resnet ensemble variant 1, in particular, consistently achieved a model score above 0.91, surpassing the rest. It surpassed the baseline in 5 out of 8 metrics, scoring AUC less than 0.9 only in 1 label compared to 2 of the baseline. Fig. 3 shows its ROC curve. As seen in Table 6, this model also surpassed the state-of-the-art (SOTA) model, which had a score of 0.9. As for the ViT-based models, all of its variants showed superior performance compared to the C-Tran baseline model while maintaining lower computational complexity. Notably, the IEViT v2 DPD variant displayed slightly better performance among the IEViT models for 32×32 dimension patch. The IE variants

of C-Tran also exhibited slight improvements over the baseline, showcasing that the IE concept can be advantageous even in generic transformer architectures while maintaining lower complexity.
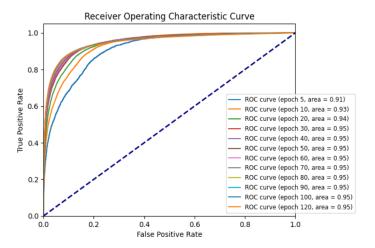

Fig. 3. ROC curve for Ensemble model

Table 6. Comparative performance of the proposed hybrid models with the state-of-the-art (SOTA) model

Model	ML mAP	ML F1	ML AUC	ML Score	Model Score
SOTA	0.573	0.685	0.962	0.824	0.9 [10]
Proposed Models					
C-Tran DenseNet201	0.6909	0.6394	0.9221	0.8065	**0.9032**[a]
C-Tran ResNet152d	0.6580	0.6042	0.9315	0.7951	0.8973
EV1 (S: RN, W: DN) [b]	0.6686	0.5974	0.9279	0.7982	0.8991
EV1 (S: DN, W: RN)	0.7120	0.6831	0.9544	0.8332	**0.9166**
EV2	0.6904	0.6538	0.9303	0.8103	**0.9052**
IEViT	0.6713	0.6015	0.9532	0.8122	**0.9061**
IEViT v1 UP	0.6781	0.6070	0.9509	0.8145	**0.9073**
IEViT v2 DPD	0.6893	0.6258	0.9420	0.8157	**0.9078**
IECTe	0.6801	0.5993	0.9420	0.8111	**0.9055**
IeECT	0.6725	0.6365	0.9299	0.8012	**0.9006**

[a]Bold numbers show improved model scores (> 0.9)

[b]S means strong, W means weak

7   Conclusion

In summary, our research underscores the effectiveness of hybrid models in classifying retinal images despite the limitations of a small, imbalanced dataset. While we have made commendable progress in multi-label retinal disease classification, we acknowledge the constraints imposed by data size and class imbalances, which could lead to randomness and potential overfitting.

Despite these challenges, our work represents a significant step in advancing deep learning-based strategies for accurate retinal disease classification. It also opens avenues for further refinement. We plan to focus on improving model interpretability to address these limitations and enhance practical utility. In our future work, we aim to explore the integration of Shapley values to unravel the causal relationships behind our model's predictions, making them more transparent and understandable.

In conclusion, our study highlights the potential of combining transformers and ensemble learning to tackle retinal disease classification challenges. Through our ongoing efforts to enhance interpretability and delve into advanced techniques, we aim to pave the way for more robust, interpretable, and effective models that can significantly benefit the field of medical diagnostics.

References

1. Dong, Li, Wanji He, Ruiheng Zhang, Zongyuan Ge, Ya Xing Wang, Jinqiong Zhou, Jie Xu et al. "Artificial intelligence for screening of multiple retinal and optic nerve diseases." JAMA Network Open 5, no. 5 (2022): e229960-e229960.
2. Tamim N, Elshrkawey M, Nassar H. Accurate diagnosis of diabetic retinopathy and glaucoma using retinal fundus images based on hybrid features and genetic algorithm. Applied Sciences. 2021;11(13):6178.
3. Chelaramani S, Gupta M, Agarwal V, Gupta P, Habash R. Multi-task learning for fine-grained eye disease prediction. In: Pattern Recognition: 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26–29, 2019, Revised Selected Papers, Part II. Springer; 2020. p. 734-749.
4. Muchuchuti S, Viriri S. Retinal Disease Detection Using Deep Learning Techniques: A Comprehensive Review. Journal of Imaging. 2023;9(4):84.
5. Heo TY, Kim KM, Min HK, Gu SM, Kim JH, Yun J, Min JK. Development of a deep-learning-based artificial intelligence tool for differential diagnosis between dry and neovascular age-related macular degeneration. Diagnostics. 2020;10(5):261.
6. Bernardes R, Serranho P, Lobo C. Digital ocular fundus imaging: a review. Ophthalmologica. 2011;226(4):161-181.
7. Gulshan, Varun, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs." Jama 316, no. 22 (2016):2402-2410.
8. Kermany, Daniel S., Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown et al. "Identifying medical diagnoses and treatable diseases by image-based deep learning." cell 172, no. 5 (2018): 1122-1131.
9. Bai J, Wan Z, Li P, Chen L, Wang J, Fan Y, Chen X, Peng Q, Gao P. Accuracy and feasibility with AI-assisted OCT in retinal disorder community screening. Frontiers in Cell and Developmental Biology. 2022;10.
10. Kim, Kyoung Min, Tae-Young Heo, Aesul Kim, Joohee Kim, Kyu Jin Han, Jaesuk Yun, and Jung Kee Min. "Development of a fundus image-based deep learning diagnostic tool for various retinal diseases." Journal of Personalized Medicine 11, no. 5 (2021): 321.
11. Das, Amrit, Rohan Giri, Gunjan Chourasia, and A. Anilet Bala. "Classification of retinal diseases using transfer learning approach." In 2019 International conference on communication and electronics systems (ICCES), pp. 2080-2084. IEEE, 2019.
12. Cen, Ling-Ping, Jie Ji, Jian-Wei Lin, Si-Tong Ju, Hong-Jie Lin, Tai-Ping Li, Yun Wang et al. "Automatic detection of 39 fundus diseases and conditions in retinal photographs using deep neural networks." Nature communications 12, no. 1 (2021): 48
13. Abbas, Ramsha, Syed Omer Gilani, and Asim Waris. "Ensemble Based Multi-Retinal Disease Classification and Application with Rfmid Dataset Using Deep Learning."
14. Müller, Dominik, Iñaki Soto-Rey, and Frank Kramer. "Multi-disease detection in retinal imaging based on ensembling heterogeneous deep learning models." In German Medical Data Sciences 2021: Digital Medicine: Recognize–Understand–Heal, pp. 23-31. IOS Press, 2021.
15. Rodriguez, M. A., H. AlMarzouqi, and P. Liatsis. "Multi-label Retinal Disease Classification Using Transformers." IEEE Journal of Biomedical and Health Informatics (2022)

16. Okolo, Gabriel Iluebe, Stamos Katsigiannis, and Naeem Ramzan. "IEViT: An enhanced vision transformer architecture for chest X-ray image classification." Computer Methods and Programs in Biomedicine 226 (2022): 107141.
17. Mendeley Data (2022). Multi-Label Retinal Diseases (MuReD) Dataset. Mendeley. https://data.mendeley.com/datasets/pc4mb3h8hz/2
18. Pachade, Samiksha, Prasanna Porwal, Dhanshree Thulkar, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, Luca Giancardo, Gwenolé Quellec, and Fabrice Mériaudeau. "Retinal fundus multi-disease image dataset (rfmid): A dataset for multi-disease detection research." Data 6, no. 2 (2021): 14.
19. Panchal, Sachin, Ankita Naik, Manesh Kokare, Samiksha Pachade, Rushikesh Naigaonkar, Prerana Phadnis, and Archana Bhange. "Retinal Fundus Multi-Disease Image Dataset (RFMiD) 2.0: A Dataset of Frequently and Rarely Identified Diseases." Data 8, no. 2 (2023): 29.
20. Zheng, Quan, Ziwei Wang, Jie Zhou, and Jiwen Lu. "Shap-CAM: Visual Explanations for Convolutional Neural Networks Based on Shapley Value." In European Conference on Computer Vision, pp. 459-474. Cham: Springer Nature Switzerland, 2022.