TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

Dissertation

On

Comparing Vision Transformers and CNNs for Accurate Retinal Disease Classification

Submitted By

Binod Paudyal

Roll No. 540/076

T.U. Regd. No. 3-2-373-188-2003

Under The Supervision of

Asst. Prof. Sarbin Sayami

Submitted To

Central Department of Computer Science and Information Technology

In partial fulfillment for the requirement of the Master's Degree in Computer Science and Information Technology (M.Sc. CSIT)

March, 2025

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

STUDENT'S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than the ones listed have been used in this work.

……..

Binod Paudyal

12th February, 2025

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

SUPERVISOR'S RECOMMENDATION

I hereby declare that the dissertation prepared under my supervision by **Mr. Binod Paudyal** entitled "**Comparing Vision Transformers and CNNs for Accurate Retinal Disease Classification**" in the partial fulfillment of the requirements for the degree of M.Sc. CSIT in the Central Department of Computer Science and Information Technology will be processed for the final evaluation.

……..

Asst. Prof. Sarbin Sayami

Central Department of Computer Science & Information Technology,

IOST, TU, Kirtipur

12th February, 2025

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of a Master's Degree in Computer Science and Information Technology.

Evaluation Committee

……..

Asst. Prof. Sarbin Sayami

(Head Of Department)

Central Department of Computer Science and Information Technology

Tribhuvan University, Kirtipur, Nepal

……..

Asst. Prof. Sarbin Sayami

(Supervisor)

Central Department of Computer Science and Information Technology

Tribhuvan University, Kirtipur, Nepal

……..

Assoc. Prof. Deo Narayan Yadav

External Examiner

Patan Multiple Campus

Tribhuvan University, Nepal

……..

Assoc. Prof. Nawaraj Paudel

Internal Examiner

Central Department of Computer Science and Information Technology

Tribhuvan University, Nepal

Date: 6th March, 2025

Acknowledgement

I am sincerely grateful to my esteemed supervisor, **Asst. Professor Sarbin Sayami**, for his unwavering support, motivation, and profound academic guidance throughout the course of this research.

My deepest gratitude to the Central Department of Computer Science and Information Technology, its faculty, and staff for their guidance and resources. I am hugely indebted to my colleagues for their constructive feedback and support.

Finally, I can't remain without thanking my family for helping me with daily chores throughout this journey.

Binod Paudyal (540/076)

Abstract

Retinal diseases, such as Age-Related Macular Degeneration (AMD) and Diabetic Macular Edema (DME) significanty contribute to vision impairment in global scale. An early diagnosis and timely treatment can save a lot of people form blindness. This research focuses on leveraging Optical Coherence Tomography (OCT) images for the classification of retinal diseases using advanced deep learning models. Specifically, we explore the capabilities of Vision Transformers (ViTs), Convolutional Neural Networks (CNNs), and a proposed Hybrid CNN-Transformer model (HybridCNNViT).

The HybridCNNViT model was developed by combining the local feature extraction strengths of CNNs with the global context modeling capabilities of Transformers. Comparative evaluations of accuracy, precision, and computational efficiency revealed that HybridCNNViT outperforms standalone ViTs and CNNs for retinal disease classification. As it offers a promising approach to improve healthcare outcomes in ophthalmology, can be further improved and used in applications of automated retinal disease detection and clinical diagnostics.

Keywords: Vision Transformers, Convolutional Neural Networks, HybridCNNViT, Retinal Disease Classification, Optical Coherence Tomography, Deep Learning in Healthcare, Medical Image Analysis

Table of Contents

List of Abbreviations

AI - Artificial Intelligence

AUC - Area Under the Curve

AMD - Age-Related Macular Degeneration

CNN - Convolutional Neural Network

CNV - Choroidal Neovascularization

DME - Diabetic Macular Edema

ERM - Epiretinal Membrane

OCT - Optical Coherence Tomography

RAO - Retinal Artery Occlusion

ReLU - Rectified Linear Unit

ROC - Receiver Operating Characteristic

RVO - Retinal Vein Occlusion

SGD - Stochastic Gradient Descent

SMOTE - Synthetic Minority Over-sampling Technique

VID - Vitreomacular Interface Disease

ViT - Vision Transformer

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

About 2.2 billion people suffer from some kind of visual impairments according to the The World Health Organization (WHO). If treated timely nearly 1 billion people can be prevented from going blind [1]. Research suggests that Retinal diseases especially Age-Related Macular Degeneration (AMD), Diabetic Macular Edema (DME), and Retinal Vein Occlusion (RVO) are likely to cause vision impairment and blindness. If these conditions are timely diagnosed, a large population can be saved from possible blindness [2] [3].

Optical Coherence Tomography (OCT) is a non-invasive imaging technology in ophthalmology, which offer a detailed and high-resolution image of the retina. [4]. Ophthalmologists, then closely examine retinal structures, using these images which helps then diagnose the conditions with a great precision. As analysis of OCT manually is a challenging job and consumes a lot of time, automating the process will be beneficial to both clinicians and patients.

Deep learning has excelled in significantly advanced fields such as speech recognition, visual object recognition, and object detection. Convolutional Neural Networks (CNNs), in particular has been used in medical image analysis tasks such as classification and segmentation [5]. Especially in the case of medical image data analysis, CNN architectures like VGG, ResNet, and EfficientNet have shown great success [6][7][8]. However, CNNs primarily focus on extracting local features, which leads to missing global contextual information in most cases, which is pivotal for comprehensive analysis particularly in medical imaging.

Vision Transformers (ViTs) has been looked upon as an alternative to CNN models as it leverages self-attention mechanisms to model long-range dependencies, treating images as sequences of patches [9]. Much superior on capturing global relationships ViTs look promising but the downside is their dependency on large datasets and computational resources. In medical imaging where there is low availability of labeled

data, it faces a bigger challenge [10]. Thus, giving rise to Hybrid models as the potential solution, which combines the strengths of CNNs and Transformers by providing solutions to these limitations, effectively capturing both local and global features [11].

1.2 Problem Definition

In any medical image classification using CNNs, the key challenge to capture both fine-grained local features and global structural context. In the case of retinal disease classification using OCT images though CNNs excel in extracting localized patterns, they often miss broader contextual relationships. However, ViTs look good in the task of global feature modeling but often require extensive data and computational resources. In order to overcome drawbacks of both CNNs and ViTs, models that can balance these capabilities, particularly hybrid architectures that integrate CNNs and ViTs can prove to be pivotal.

As there is a lack of comparative studies evaluating the performance of CNNs, ViTs, and hybrid models on OCT datasets, the necessity for this research further underscores. If gaps are addressed, an automated diagnostic systems for retinal diseases can be designed from the findings of this research.

1.3 Objectives

The primary objectives of this research are:

1. To customize a Vision Transformer (ViT) model and develop a Hybrid CNN-Transformer model (HybridCNNViT) optimized for OCT image classification.
2. To perform a quantitative comparison of the HybridCNNViT, customized ViT, ResNet-50, EfficientNet-B0, and VGG16 models.

1.4 Dissertation Organization

This dissertation is organized as follows:

- Chapter 1 introduces the background, problem statement, objectives, and organization of the dissertation.
- Chapter 2 provides a comprehensive review of relevant literature.

- Chapter 3 details the methodology, including dataset collection, model architectures, and evaluation metrics.
- Chapter 4 describes the implementation process, including tools, libraries, and workflows.
- Chapter 5 discusses findings from experiments and performance comparisons.
- Chapter 6 concludes the research and offers recommendations for future work.

Chapter 2: Literature Review

2.1 Convolutional Neural Networks in Medical Imaging

Convolutional Neural Networks (CNNs) reduce the work loads of clinicians around the world by automating feature extraction. CNNs consist of layers that perform convolutional operations to capture spatial hierarchies in images which cuts the time and efforts of a clinician if done manually[5]. With the turn of the century, deep learning models a subset of machine learning have shown great prospects in medical imaging. Deep CNN architectures in particular, such as VGG, ResNet, and EfficientNet, has revolutionized medical imaging by offering unparalleled accuracy in image classification tasks [6][7][8].

The availability of larger dataset paved way for the deep CNNs architectures. One of these models, the VGG network, introduced by Simonyan and Zisserman [6], uses small 3×3 filters in deep architectures, enabling fine-grained feature extraction. The simple architecture proved appealing to researchers but the computational cost is high as well as the memory requirements limits its scalability. It took a while for addressing this issue but few years later another deep CNN model, ResNet, utilized residual connections to solves the vanishing gradient problem. This model made the training process of deeper networks easier [7]. As the huge number of parameters pose problem in deep learning by consuming a lot of computational resources, EfficientNet scales CNN architectures efficiently by optimizing depth, width, and resolution, achieving better accuracy with fewer parameters [8]. These advancements have established CNNs as the backbone of medical image analysis.

CNNs have been used widely in the field of medicine. These include the classification of retinal diseases, organ segmentation, and tumor detection. CNNs have the potential to increase early detection rates, as evidenced by Esteva et al.'s demonstration of their efficacy in diagnosing skin cancer with dermatologist-level accuracy [12]. The accuracy of one of the model, when applied to the BreakHis dataset, ranged from 98.87% to 99.34% for binary classification and from 90.66% to 93.81% for multi-class classification, indicating that CNNs are useful for accurately and automatically

detecting cancer [13]. These uses demonstrate CNNs' versatility in a range of medical imaging tasks.

CNNs have intrinsic limitations despite their success. When it comes to comprehending complicated medical images, their dependence on local feature extraction frequently leaves out global patterns [14]. CNNs also need large labeled datasets to avoid overfitting, which is a major drawback in medical imaging since annotated data is expensive and hard to get by [15]. Although their efficacy varies depending on the application and dataset quality, data augmentation, transfer learning, and the integration of synthetic data production have emerged as answers to these problems.

2.2 Vision Transformers in Computer Vision

Using the Transformer architecture originally created for natural language processing, Vision Transformers (ViTs), first presented by Dosovitskiy et al. [9], represent a paradigm shift in computer vision. In contrast to CNNs, ViTs split a picture into fixed-size patches, which a Transformer encoder flattens and processes as sequences. This method overcomes CNNs' intrinsic constraints by allowing ViTs to record contextual information and global dependencies. The original ViT model was pre-trained on JFT-300M, a dataset containing 300 million images, and fine-tuned on ImageNet, where it achieved a top-1 accuracy of 88.55%, outperforming traditional CNNs like ResNet-152, which achieved 77.8% accuracy under similar conditions [9].

ViTs have outperformed CNNs in situations that call for a comprehensive grasp of visual patterns and have shown state-of-the-art performance in large-scale picture categorization tasks. For instance, the Data-efficient Image Transformer (DeiT) achieved 83.1% top-1 accuracy on ImageNet-1K while utilizing significantly fewer training resources compared to the original ViT model [10]. These findings underscore ViTs’ potential to revolutionize medical imaging applications.

ViTs' dependence on huge datasets and heavy processing requirements, however, make it difficult to use them in medical imaging. When trained solely on ImageNet (1.3 million images) without additional pre-training, ViTs underperformed compared to CNNs, demonstrating their reliance on large-scale data for effective feature learning [16]. To improve flexibility, techniques including using domain-specific augmentations, embedding pre-trained ViTs on medical datasets, and using smaller

patch sizes to preserve spatial hierarchies have been looked into [17]. Combining these approaches is essential to expanding the use of ViTs to smaller datasets, which are common in medical imaging.

2.3 Hybrid CNN-Transformer Models

Hybrid models that combine CNNs and Transformers have been proposed to bridge the gap between local and global feature extraction. These models seek to capitalize on the advantages of both architectures: Transformers' global attention mechanism and CNNs' local feature extraction capacity. In order to enable global context modeling without appreciably increasing computational complexity, Bottleneck Transformers (BoTNet), for example, substitute multi-head self-attention layers for some of the convolutional levels in ResNet achieving a top-1 accuracy of 84.7% on the ImageNet benchmark. This approach not only improves performance but also reduces parameters, with minimal latency overhead. [11]. Similarly, by utilizing local information, Convolutional Vision Transformers (ConViTs) improve ViTs' performance on datasets like CIFAR-10 and CIFAR-100 by incorporating soft convolutional inductive biases [16].

Convolutional neural networks and vision transformers are combined in CMT (Convolutional Neural Networks Meet Vision Transformers), a hierarchical architecture that processes images at various scales to capture coarse and fine-grained Characteristics. CMT-S outperformed previous CNN-based and transformer-based models, achieving 83.5% top-1 accuracy on ImageNet while being significantly more computationally efficient [17]. By skillfully combining the advantages of both models, this architecture provides enhanced performance for tasks like organ segmentation and the categorization of retinal diseases. Hybrid models are appropriate for complicated medical imaging problems due to their resilience in managing unbalanced datasets and utilizing multi-scale feature extraction.

2.4 Applications in Retinal Disease Classification

Research on the classification of retinal diseases using OCT images is crucial because it has the potential to completely transform early diagnosis and therapy planning. By utilizing their deep architectures and feature extraction capabilities, traditional CNN-

based techniques, including those using ResNet and EfficientNet, have demonstrated remarkable performance in the classification of retinal disorders [15].

Because of their complex structures and great resolution, OCT pictures pose special difficulties. Despite their effectiveness, CNNs frequently fail to capture the global dependencies needed for thorough analysis. ViTs overcome this constraint by modeling global context through self-attention techniques. Ma et al. demonstrated that a hybrid CNN-Transformer model outperformed classic CNNs in OCT image classification, achieving overall accuracies of 91.56% and 86.18% on two public retinal OCT datasets, respectively, exhibiting improved accuracy and resilience in detecting retinal abnormalities [18].

The strengths of CNNs and Transformers are used in hybrid models to further improve classification performance. For example, hybrid architectures have been used to accurately diagnose diabetic retinopathy and macular degeneration by taking use of their ability to collect both local and global information [16]. Even with these developments, problems still exist. Training deep learning models, especially ViTs, is restricted by the amount of OCT datasets. Deploying these models in clinical settings still requires striking a balance between computational resources and model complexity.

Chapter 3: Methodology

In order to classify diseases using the OCTDL dataset, this study focuses on creating a hybrid CNN-ViT model and evaluating its performance against Vision Transformer (ViT) and various CNN architectures, such as ResNet, VGG16, and EfficientNet.

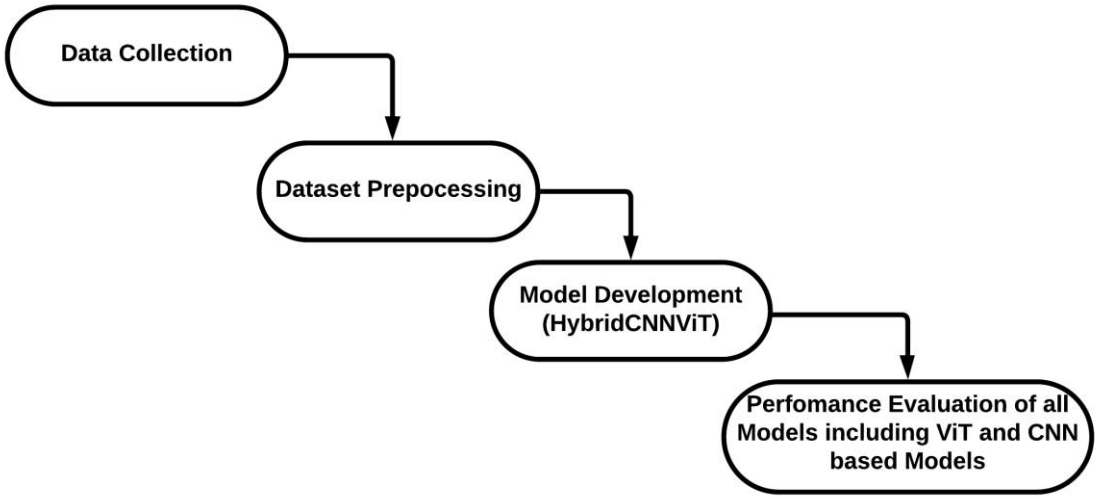

Figure 3.1: The sequence of steps carried out in the experiments of this research

At first, the CNN and ViT models' baseline performances were assessed without any changes made to their architectures. These assessments served as benchmarks for comparison. After that, a Hybrid CNN-ViT model was created by fusing embeddings from a pre-trained Vision Transformer with features that were taken from a ResNet backbone. The goal of the hybrid design was to combine the advantages of transformers (global context awareness) with CNNs (local feature extraction).

In order to enhance performance and avoid overfitting, the Hybrid CNN-ViT model was developed by integrating data preprocessing techniques, applying early stopping, and extensively adjusting hyperparameters. To guarantee a thorough comparison, performance metrics including accuracy, F1-score, AUC, ROC, and confusion matrix were calculated for each model.

Lastly, the efficiency of the Hybrid CNN-ViT model in optical coherence tomography image classification was examined and contrasted with the CNN and ViT models used alone. The study sought to determine which model, in terms of accuracy, computing efficiency, and generalizability, was most appropriate for classifying diseases.

3.1 Dataset Collection

High-resolution images from seven retinal disease classes—Age-Related Macular Degeneration (AMD), Diabetic Macular Edema (DME), Epiretinal Membrane (ERM), Normal, Retinal Artery Occlusion (RAO), Retinal Vein Occlusion (RVO), and Vitreomacular Interface Disease (VID)—are included in the Optical Coherence Tomography (OCT) dataset, which was obtained from a publicly accessible repository [19]. There are 2,064 images in the dataset, and each class is sufficiently represented to guarantee balanced training and testing splits. To eliminate duplicates, corrupted files, and incorrectly classified samples, the dataset was subjected to thorough quality checks.

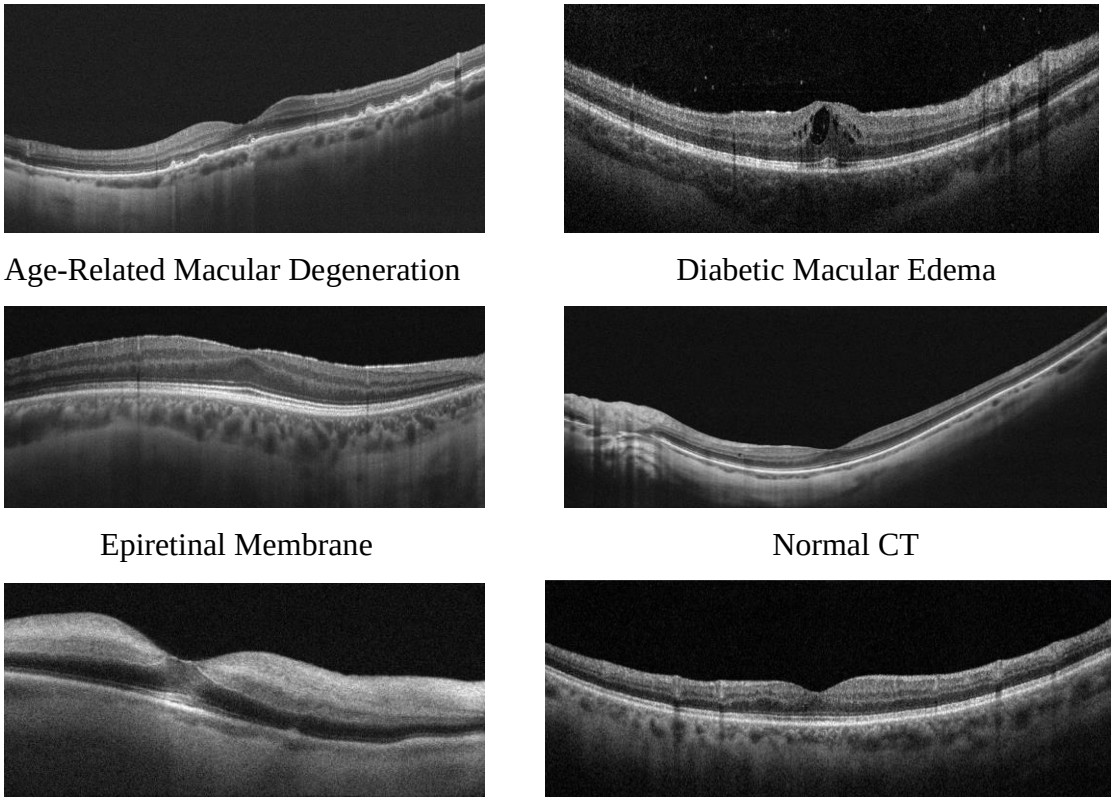

Figure 3.2: Sample images from the dataset.

3.2 Data Preprocessing

1. **Resizing:** Images were resized to 224×224 pixels to ensure uniformity across models.

2. **Normalization:** Pixel intensity values were normalized to a range of [0, 1] using the mean and standard deviation of the ImageNet dataset, as many models were pre-trained on it.
3. **Data Augmentation:** Augmentation techniques, such as rotation (±15 degrees), horizontal flipping, contrast and brightness adjustments, Gaussian noise addition, and random zooming, were applied to enhance data diversity and reduce overfitting. In addition to these techniques a cutting-edge CutMix augmentation technique is aimed to improve a model's capacity to generalize to previously unknown input. [20] It works by substituting a matching patch from another image in the same batch for a random area of one image. This technique offers several advantages. It improves a model's localization ability by exposing it to mixed features during training. Additionally, it enhances generalization by creating novel samples, which helps reduce overfitting

4. **Label Encoding:** Disease classes were encoded as numerical labels for multi-class classification tasks.

3.3 Various Deep Learning Architectures

3.3.1 Vision Transformer (ViT)

The Vision Transformer (ViT), introduced by Dosovitskiy et al. [9], takes an innovative approach for image recognition by utilizing the Transformer architecture, which was originally designed for natural language processing. In contrast to conventional convolutional neural networks (CNNs), which use convolutional filters to extract local features from pictures, ViTs view an image as a series of patches, so turning the image recognition issue into a sequence modeling problem.

The image is separated into fixed-size patches, usually 16×16 pixels, in order to prepare the input. In order to generate patch embeddings, each patch is first flattened into a 1D vector and then run through a linear projection layer, which converts the vectors into a higher-dimensional space. Understanding the spatial context of the image depends on the model maintaining the relative positional information of each patch, which is made possible via positional encodings.

A Transformer encoder receives these enhanced patch embeddings. The encoder is made up of feed-forward neural networks and several layers of multi-head self-attention processes. The model can comprehend both local and long-range links in the image thanks to the self-attention mechanism, which enables it to capture global dependencies across all patches. This is one of ViTs' main advantages over CNNs, which are best at identifying local patterns.

Before the sequence of patch embeddings is sent through the encoder, a unique learnable class token is added. This token uses the self-attention method to compile data from all patches. The class token functions as a summary representation of the full image at the encoder's output, where it is subsequently utilized for classification tasks. Complex visual recognition tasks require the ability to capture global relationships and contextual information in images, which the ViT's architecture makes it very good at. But because it doesn't take advantage of CNNs' built-in inductive biases, including translational invariance and locality, it needs a large amount of training data to operate at its best.

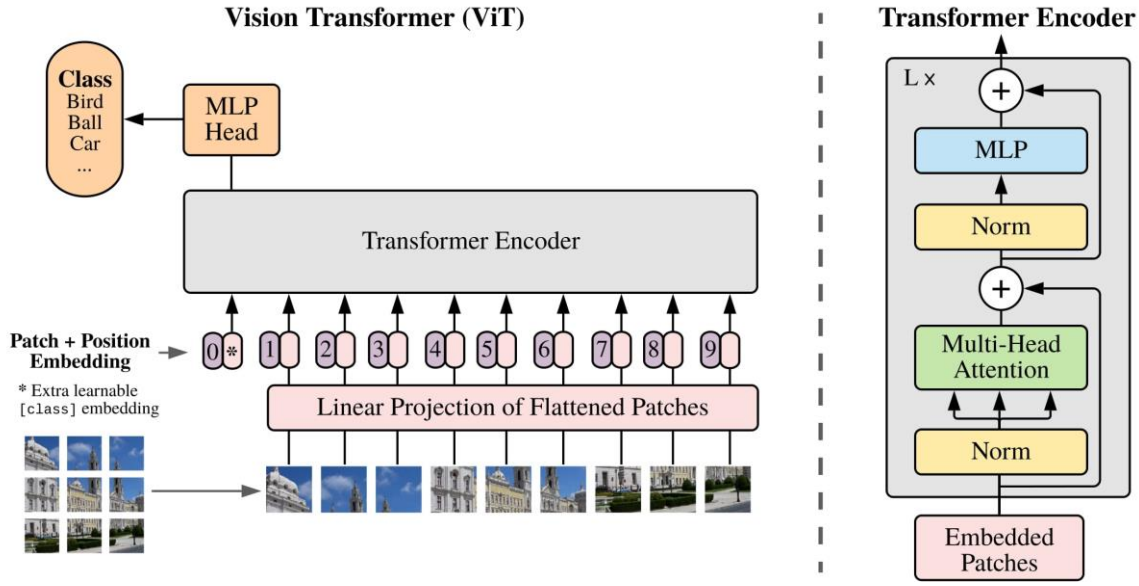

Figure 3.3: VIT Architecture

(Image Source: Transformers for image recognition at scale [9])

3.3.2 ResNet-50

He et al. [7] proposed ResNet-50, which uses a residual learning framework to overcome the vanishing gradient problem and other difficulties in training very deep neural networks. As the gradient decreases during backpropagation, traditional deep networks frequently falter, making it challenging for the weights of previous layers to update efficiently. By adding residual blocks with skip (or shortcut) connections, ResNet (Residual Network) addresses this problem.

Residual Learning Framework

The key concept of ResNet is the usage of residual blocks, in which a block's input is appended directly to its output. A residual block can be expressed mathematically as :

$$y = F(x, \{W_i\}) + x$$

Here, x is the input, $F(x, \{W_i\})$ represents the transformation applied by the block (e.g., convolutional layers), and y is the output of the block. The skip connection ensures that the network learns a residual mapping $F(x)$ rather than the direct mapping $H(x) = F(x) + x$. This approach simplifies optimization and helps mitigate the vanishing gradient problem by allowing gradients to flow directly through the skip connections.

Architecture

ResNet-50 is a 50-layer deep neural network made up of the following layers:

- **Convolutional Layers:** These layers use convolution operations to extract features from the input images.
- **Batch Normalization**: To stabilize and speed up training, this method normalizes the output of convolutional layers.
- **Activation Functions**: To add non-linearity to the network, ReLUs (Rectified Linear Units) are utilized.

- **Pooling Layers**: The most important features are preserved and computational complexity is decreased by using pooling layers, which shrink the spatial dimensions of feature maps.

- **Fully Connected Layer**: This last layer generates class scores for tasks involving image categorization.

ResNet-50's 50 layers are separated into a number of stages, each of which consists of several residual blocks. Each residual block's bottleneck architecture is a crucial component of ResNet-50. The amount of calculations is decreased by the bottleneck block by:

1. Applying a 1×1 convolution to reduce the feature dimensions.
2. Using a 3×3 convolution to extract spatial features.
3. Applying another 1×1 convolution to restore the feature dimensions.

This bottleneck design allows ResNet-50 to be computationally efficient while maintaining its representational power.

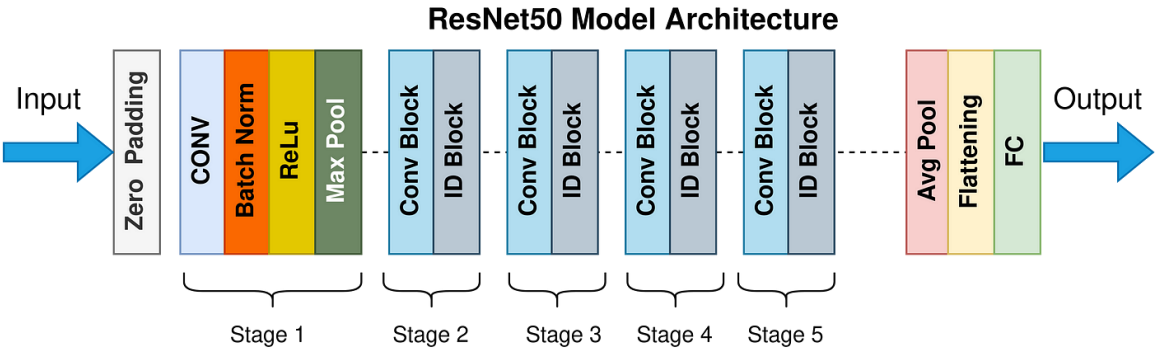

Figure 3.4: ResNet50 Architecture

3.3.3 EfficientNet-B0

Tan and Le [8] presented EfficientNet, a novel compound scaling strategy for optimizing three important architectural dimensions: depth (number of layers), breadth (number of channels in each layer), and resolution (size of input images). Conventional scaling techniques frequently modify one of these dimensions separately, which may result in less-than-ideal performance or wasteful computing usage. In order to provide

a balanced and effective model design, EfficientNet's compound scaling methodically scales these dimensions together using a well selected set of scaling coefficients.

EfficientNet-B0, the baseline model, serves as the cornerstone for the EfficientNet family. It uses a number of advanced techniques to attain great accuracy with fewer parameters:

1. **Depthwise Separable Convolutions**: These drastically cut down on the amount of parameters and operations required for standard convolutions by factorizing them into depthwise and pointwise operations.

2. **Squeeze-and-Excitation Modules**: These modules re-calibrate channel-specific feature responses by modeling channel interdependencies, allowing the network to focus on more informative features.

3. **Swish Activation Function**: EfficientNet employs the Swish activation function $(x \cdot \text{sigmoid}(x))$, which enhances the non-linear representational capacity of the network compared to ReLU.

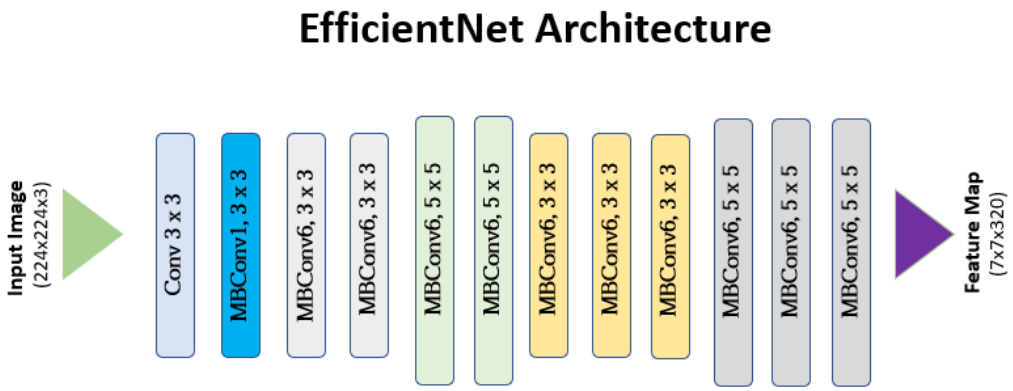

Figure 3.5: EfficientNet-B0 Architecture

3.3.4 VGG16

VGG16, proposed by Simonyan and Zisserman [6], uses a simple and highly structured deep learning architecture of 16 layers, including 13 convolutional layers and 3 fully linked layers. Its design uses tiny 3×3 convolutional filters all around the network,

emphasizing consistency and simplicity. These filters, which have a stride of 1, capture fine-grained spatial information, allowing for accurate feature extraction.

Convolutional layers are followed by ReLU activation functions to add non-linearity to the block-structured design. The spatial dimensions of the feature maps are reduced by interspersing max-pooling layers with a 2×2 filter after each block, eliminating computational cost while preserving crucial characteristics. As the depth grows, the network may learn more abstract representations thanks to this hierarchical structure. VGG16 is very good at image classification tasks because the fully connected layers at the end of the network convert the retrieved characteristics into class probabilities. Notwithstanding its simplicity, VGG16 includes a lot of parameters (around 138 million), which raises its memory and processing requirements. Its outstanding performance and strong generalization on big datasets come at the expense of this.

Because of its performance in image classification tasks and its ability to extract features for transfer learning applications, VGG16 has established itself as a benchmark model in computer vision. It is a fundamental architecture in deep learning research because of its consistent design and modular nature.

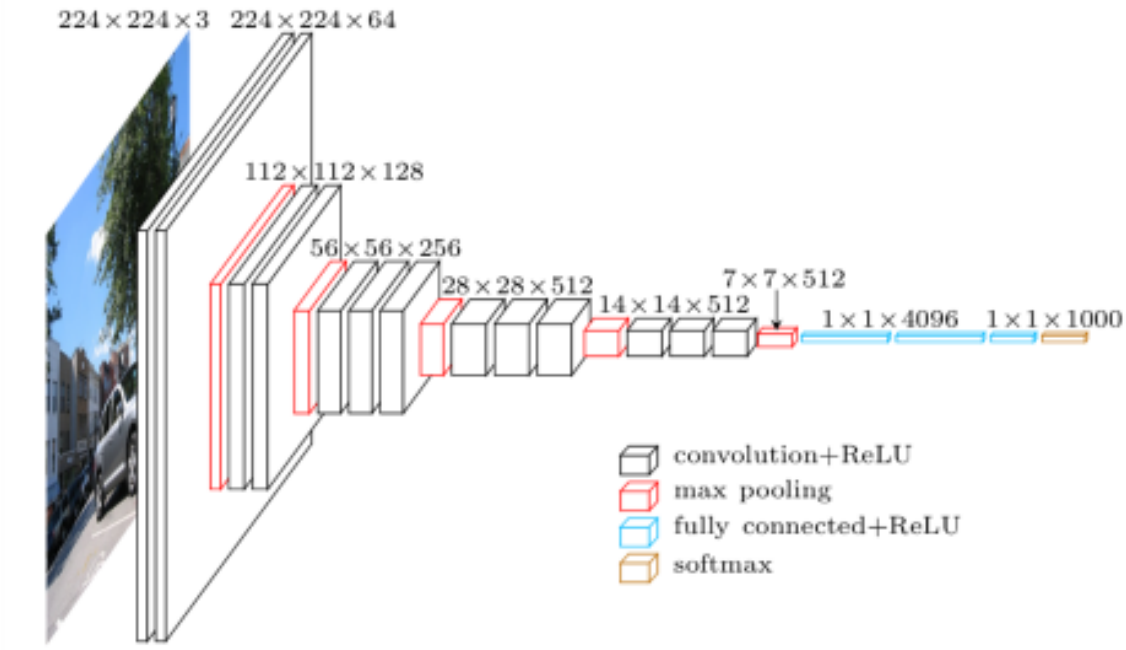

Figure 3.6: VGG16 Architecture

3.3.5 Hybrid CNN-Transformer Model (HybridCNNViT)

The HybridCNNViT model combines CNNs with Vision Transformers to take advantage of the strengths of each architecture. Here is a full discussion of how the HybridCNNViT model works:

1. **ResNet-50 Backbone:**
    - ResNet-50 is used to extract localized spatial features from OCT pictures. The image is processed by the model's convolutional layers to extract fine-grained information, and dimensional reduction is guaranteed by global average pooling.
    - An identity layer (nn.Identity()) is used in place of the fully connected layer of ResNet-50 to provide a high-dimensional feature map (2048 features).
2. **Vision Transformer (ViT):**
    - The ViT component divides the input image into patches (16×16 pixels), which are then flattened and passed through a linear projection layer to form patch embeddings.
    - Positional encodings are added to the embeddings to retain spatial relationships.
    - A multi-head self-attention mechanism processes the embeddings to capture global dependencies. The output from ViT is a 768-dimensional feature vector.
3. **Feature Fusion:**
    - The feature vectors from ResNet-50 and ViT are concatenated along the feature dimension, resulting in a combined feature vector of size 2816.
    - This concatenation enables the model to utilize both local and global features effectively, improving classification performance.

4. **Fully Connected Layers:**

    - The concatenated feature vector is passed through a fully connected layer that reduces the dimensionality to 256 features.

    - A dropout layer is applied to prevent overfitting, followed by another fully connected layer that maps the reduced features to the number of retinal disease classes.

5. **Forward Pass:**

    - During inference, the input image is simultaneously processed through the ResNet-50 backbone and ViT. Their outputs are concatenated, and the resulting feature vector is passed through the fully connected layers to produce final predictions.

The HybridCNNViT model effectively combines the strengths of ResNet-50 for extracting detailed local features and ViT for capturing global contextual information, ensuring comprehensive feature representation. Its modular design allows seamless integration of pre-trained models, leveraging transfer learning for improved performance. Despite its sophisticated architecture, the use of pre-trained components ensures computational efficiency and feasibility. Furthermore, the model's scalability enables adaptation to diverse datasets and tasks through fine-tuning of its CNN and ViT components. By harnessing pre-trained features, the HybridCNNViT demonstrates strong generalization and delivers robust performance across various image classification tasks.

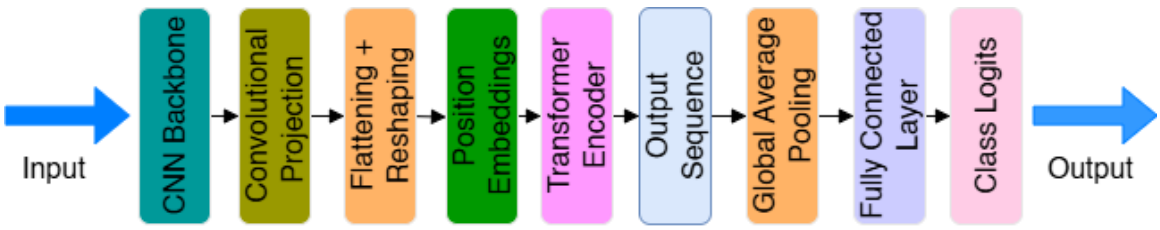

Figure 3.7: HybridCNNViT Architecture

3.4 Training and Validation

1. **Split Ratio:** The dataset was divided into training (60%), validation (20%), and testing (20%) sets.

2. **Optimizer:** AdamW optimizer was used for training, with an initial learning rate of 0.0001.

3. **Loss Function:** Categorical cross-entropy loss was employed to minimize the discrepancy between predicted probabilities and actual labels.

4. **Early Stopping:** Training was halted if the validation loss did not improve for 15 consecutive epochs, preventing overfitting.

5. **Epochs and Batch Size:** Models were trained for 50 epochs with a batch size of 32.

3.5 Evaluation Metrics

To comprehensively evaluate the performance of the models, several metrics were used, each with a specific formula to quantify key aspects of model performance:

1. **Accuracy:** The proportion of correctly classified instances out of the total predictions.

$$\mathbf{Accuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

TP: True Positives
TN: True Negatives
FP: False Positives
FN: False Negatives

2. **Precision:** The ratio of true positives to the sum of true positives and false positives, indicating the correctness of positive predictions.

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

3. **Recall (Sensitivity):** The ratio of true positives to the sum of true positives and false negatives, measuring the model's ability to detect positive instances.

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}}$$

4. **F1 Score:** The harmonic mean of precision and recall, providing a single measure of model performance that balances the two metrics.

$$\textbf{F1 Score} = 2 \times \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}$$

5. **Confusion Matrix:** A tabular representation of the model's performance, showing the counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each class.

Predicted / **Actual**	**True**	**False**
True	**TP**	**FN**
False	**FP**	**TN**

Chapter 4: Implementation

4.1 Environment and Hardware

The implementation was carried out using Google Colab, leveraging its computational resources to train and evaluate the models efficiently. The following hardware was utilized:

- **GPU:** NVIDIA Tesla T4, providing accelerated computation for deep learning.
- **CPU:** Intel Xeon, for preprocessing and auxiliary tasks.
- **RAM:** 12 GB, adequate for handling the OCT dataset and model parameters.
- **Storage:** Google Drive integration for dataset storage and checkpoints.

4.2 Programming Language and Libraries

4.2.1 Programming Language

Python was used as the primary programming language for the implementation due to its versatility and extensive support for deep learning libraries.

4.2.2 Libraries and Packages

The implementation and training of the deep learning models, including ResNet-50, EfficientNet-B0, VGG16, and Vision Transformers (ViT), were carried out using PyTorch, which served as the primary deep learning framework. Pre-trained Vision Transformer (ViT) models were utilized through the Transformers library, while TorchVision was employed for data augmentation, access to pre-trained models, and related utilities. Numerical operations and dataset preprocessing were facilitated by NumPy, and Matplotlib and Seaborn were used for visualizing data and plotting evaluation metrics. Additionally, Scikit-learn was instrumental in computing evaluation metrics and generating confusion matrices to assess model performance comprehensively.

4.3 Steps of Implementation

The research was carried out in the following flow which can be divided into two major phases. One was dataset preparation and image pre-processing and another was model

preparation and model compilation. After these phases were complete, the processed images were used for model training and the model's performance was evaluated.

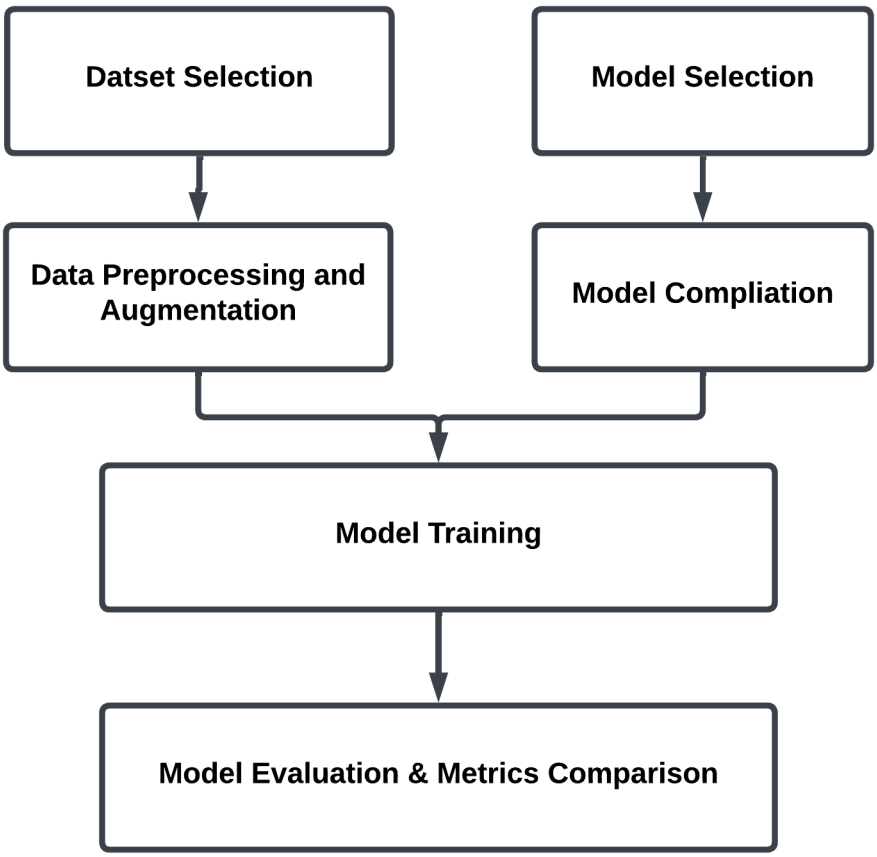

Figure 4.1: The overall framework of the technical portion of this research work of this dissertation

4.3.1 Dataset Preparation

1. **Loading:** The OCT dataset was loaded from its repository and unzipped into appropriate directories.
2. **Splitting:** The dataset was split into training (60%), validation (20%), and testing (20%) subsets.
3. **Organizing:** Data was organized into folders based on class labels for compatibility with data loaders.

4.3.2 Image Preprocessing

1. **Resizing:** All images were resized to 224×224 pixels to ensure uniform input dimensions.

2. **Augmentation:** Data augmentation techniques, including horizontal flipping, random rotation, contrast adjustments, and Gaussian noise addition, were applied during training to improve generalization.

3. **Normalization:** Images were normalized using the mean and standard deviation of the ImageNet dataset to align with pre-trained model requirements.

4.3.3 Cut- Mix Augmentation

As part of the implementation, CutMix was utilized to enhance the diversity of the training dataset and improve model generalization. During the training phase, the `cutmix` function was applied to each batch of training data.

4.3.4 Image Normalization

Normalization was performed to scale pixel intensity values to a range of [0, 1] using the following formula:

$$X_{norm} = X - \mu \sigma X_{norm} = \frac{x - \mu}{\sigma}$$

where μ and σ represent the mean and standard deviation, respectively, calculated from the ImageNet dataset.

4.3.5 Model Preparation

1. **ResNet-50 Backbone:** A pre-trained ResNet-50 model was loaded, and its fully connected layer was replaced with an identity layer to output feature embeddings.

2. **Vision Transformer (ViT):** The ViT-B16 model, pre-trained on ImageNet-21k, was used to extract global dependencies from the input images.

3. **HybridCNNViT Model:** The ResNet-50 backbone with the ViT model. Then the concatenated feature vectors were passed through fully connected layers to generate class predictions.

4.3.6 Model Training and Evaluation

1. **Optimizer:** AdamW optimizer with a learning rate of 0.0001 was used for training.

2. **Loss Function:** Cross-entropy loss was employed for multi-class classification.

3. **Early Stopping:** Training was halted if the validation loss did not improve for 15 epochs.

4. **Metrics:** Accuracy, precision, recall, F1 score, and AUC-ROC were computed to evaluate model performance.

5. **Epochs and Batch Size:** Training was conducted for 50 epochs with a batch size of 32.

6. **Logging:** Training and validation losses were logged after each epoch, and checkpoints were saved for the best-performing model.

Chapter 5: Findings and Discussion

5.1 Performance of the Models

5.1.1 HybridCNNViT

The HybridCNNViT model achieved an accuracy of **95.16%**, combining the local feature extraction of ResNet-50 with the global contextual understanding of ViT. It demonstrated robust performance across all classes, effectively balancing the strengths of CNNs and Transformers.

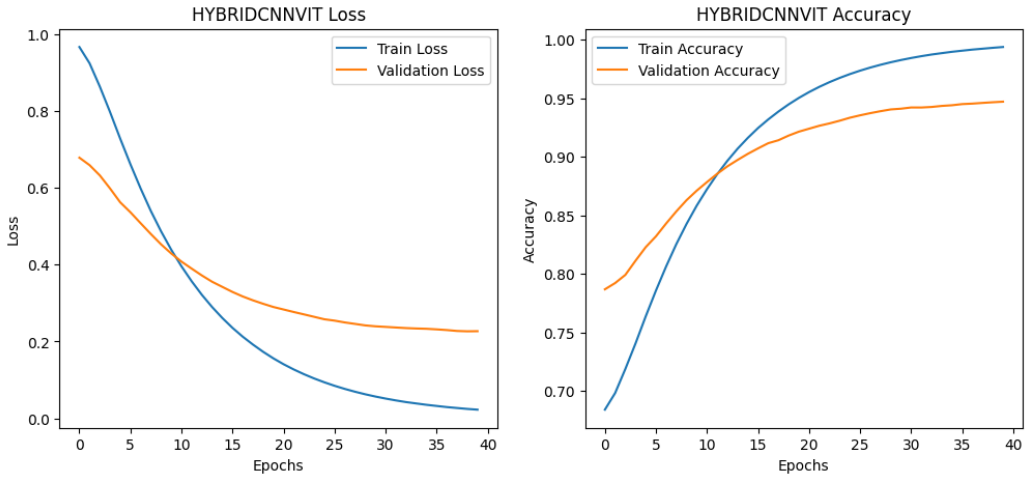

Figure 5.9: Training and Validation Loss And Accuracy for HybridCNNViT model

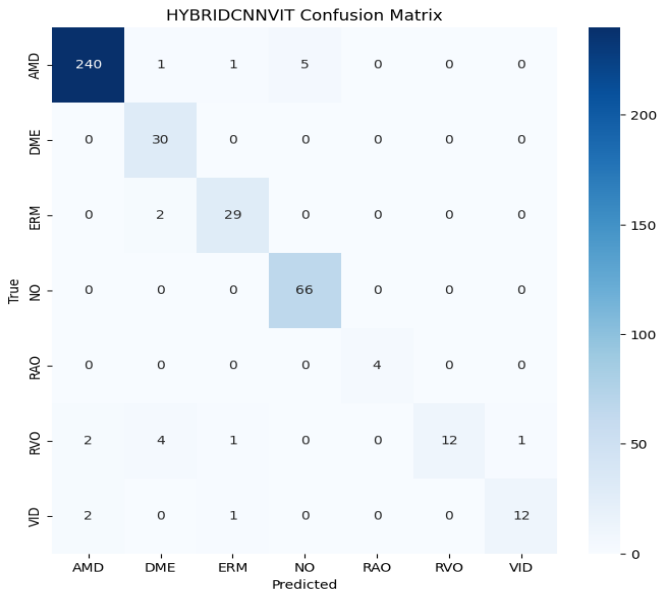

Figure 5.10: Confusion Matrix for HybridCNNViT model.

5.1.2 ResNet-50

ResNet-50 achieved an accuracy of **92.74%** on the test dataset. It demonstrated a strong ability to classify retinal images correctly, particularly excelling in classes with abundant samples. However, precision and recall scores for less-represented classes, such as RVO and VID, were slightly lower due to the inherent class imbalance.

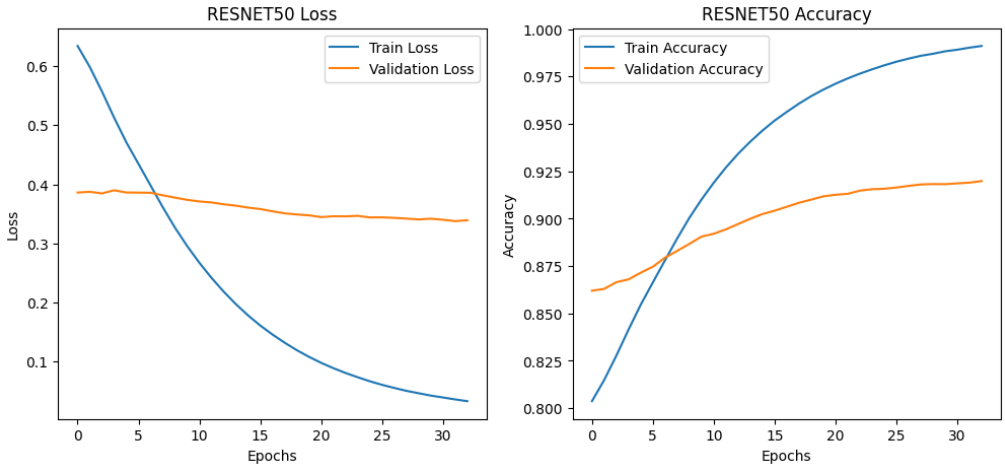

Figure 5.11: Training and Validation Loss And Accuracy for ResNet-50 model

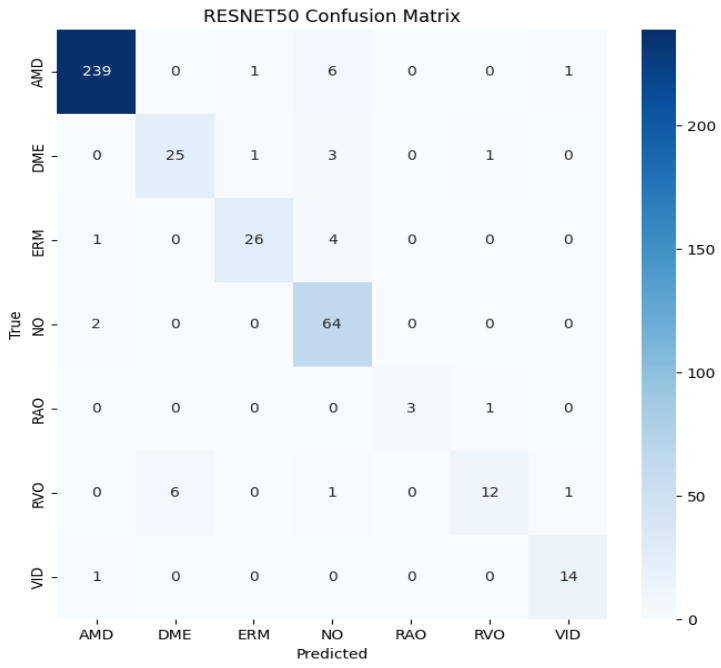

Figure 5.12: Confusion Matrix for Resnet50 model

5.1.3 VGG16

The VGG16 model outperformed the other CNN-based models with an accuracy of **93.46%**. Its consistent performance across all classes highlights its ability to extract detailed spatial features, particularly in complex retinal disease images.

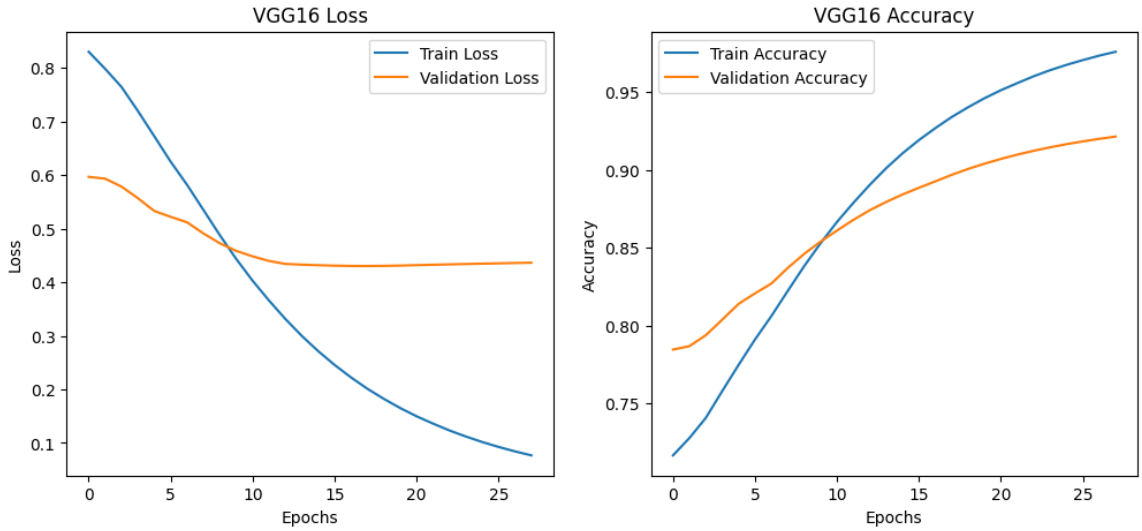

Figure 5.13: Training and Validation Loss And Accuracy for VGG16 model

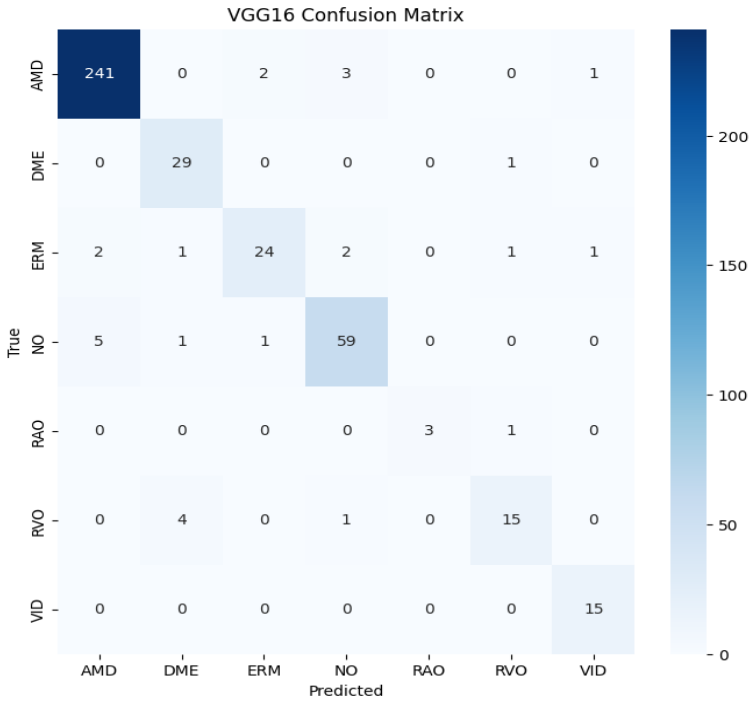

Figure 5.14: Confusion Matrix for VGG16 model

5.1.4 EfficientNet-B0

EfficientNet-B0 achieved an accuracy of **91.77%**, demonstrating a balance between computational efficiency and performance. It performed well in identifying rare classes, such as RAO and VID, with precision scores exceeding **90%**.

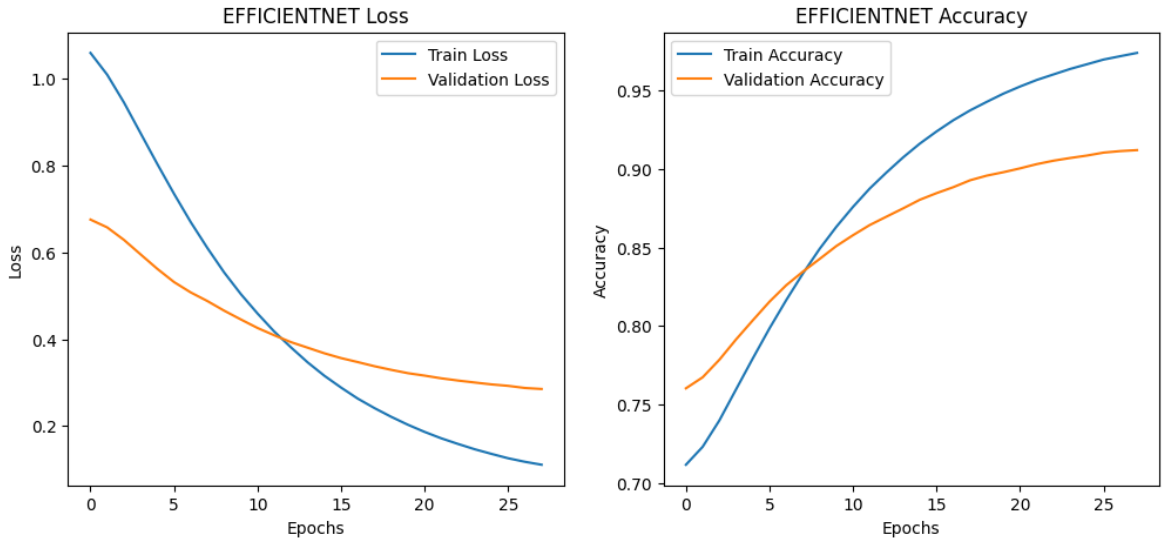

Figure 5.15: Training and Validation Loss And Accuracy for EfficientNet-B0 model

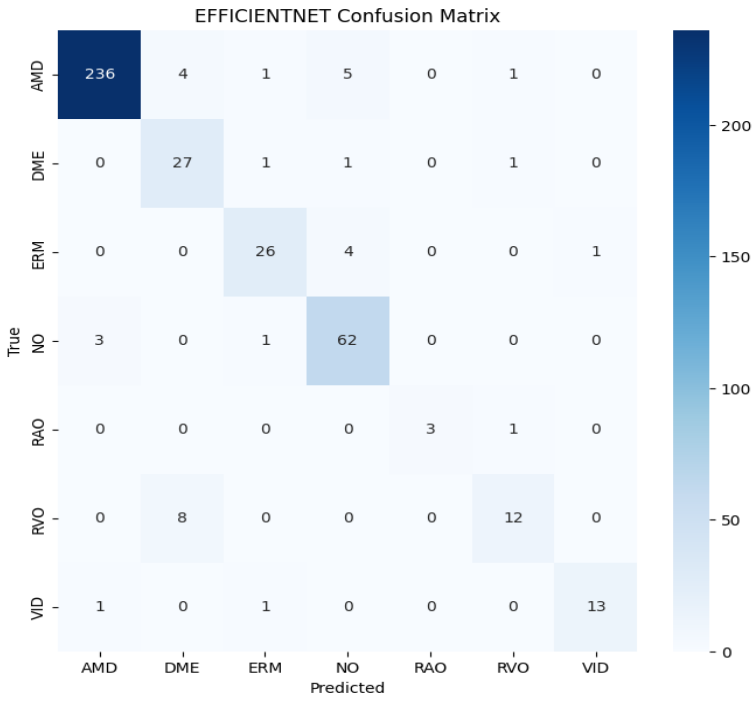

Figure 5.16: Confusion Matrix for EfficientNet-B0 model

5.1.5 Vision Transformer (ViT)

ViT achieved an accuracy of **90.56%**, leveraging its ability to capture global dependencies. While effective in identifying classes with unique features, it showed marginally lower performance in handling noisy or overlapping datasets.

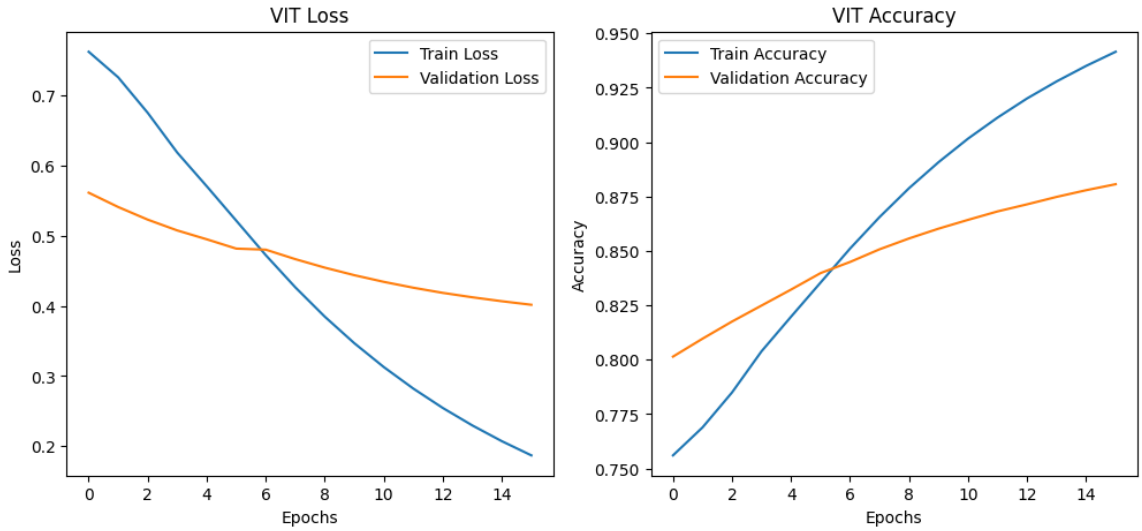

Figure 5.17: Training and Validation Loss And Accuracy for ViT model

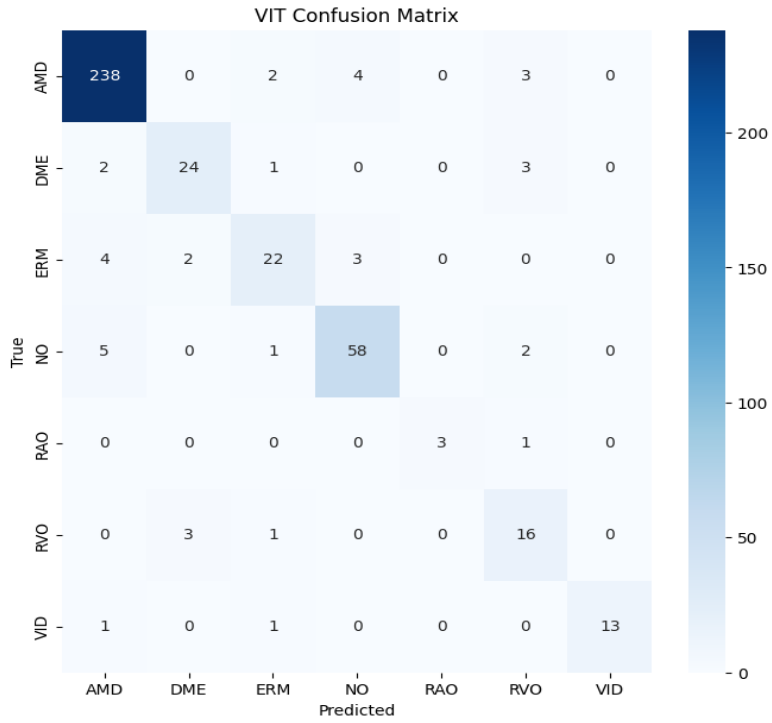

Figure 5.18: Confusion Matrix for ViT model

5.2 Comparison of Model Performance

A comprehensive comparison of the models highlights VGG16 as the best-performing model based on accuracy, precision, and recall metrics. However, the HybridCNNViT model closely followed, showcasing its potential for tasks requiring combined local and global feature extraction.

Table 1: Comparison of Metrics all Models

Model	Accuracy	Precision	Recall	F1_score	Cohen_kappa
VGG16	0.9346	0.9352	0.9346	0.9338	0.8912
RESNET50	0.9274	0.9306	0.9274	0.9265	0.8798
EFFICIENTNET	0.9177	0.9232	0.9177	0.9181	0.8649
VIT	0.9056	0.9080	0.9056	0.9058	0.8422
HYBRIDCNNVIT	0.9516	0.9554	0.9516	0.9499	0.9199

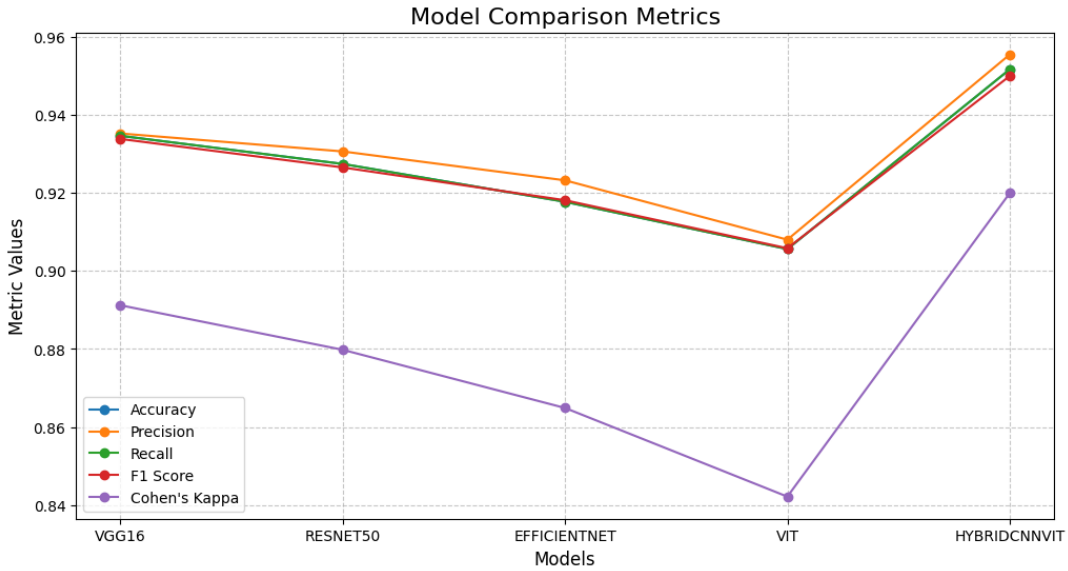

Figure 5.19: Comparison of metrics of all models.

5.3 Challenges and Limitations

1. **Class Imbalance:** Despite data augmentation techniques, imbalanced class distribution affected the performance of certain models.

2. **Training Complexity:** HybridCNNViT required more training time and resources due to its complex architecture.

3. **Generalization:** ViT's reliance on large datasets limits its performance on smaller or noisier datasets.

5.5 Recommendations

1. **Hybrid Model Optimization:** Further fine-tuning of the HybridCNNViT model could improve its performance, though it still outperformed all other CNN models and ViT.

2. **Class Balancing Techniques:** Advanced balancing methods, such as SMOTE, could address class imbalance issues.

3. **Scalability:** Lightweight models like EfficientNet-B0 can be optimized for real-time applications.

Chapter 6: Conclusion and Future Recommendation

6.1 Conclusion

This study aimed to classify retinal diseases using Optical Coherence Tomography (OCT) images by evaluating the performance of multiple deep learning models, including ResNet-50, VGG16, EfficientNet-B0, Vision Transformer (ViT), and a HybridCNNViT model. Through extensive experimentation and analysis, the following key findings were observed:

1. **Model Performance:** HybridCNNViT achieved the highest accuracy (95.16%), closely followed by the VGG model (93.46%). It demonstrated robust performance by effectively combining the local feature extraction capabilities of ResNet-50 with the global context modeling of ViT.
2. **Advantages of Hybrid Models:** The HybridCNNViT model highlighted the benefits of integrating CNNs and Transformers, excelling in feature extraction and classification across diverse retinal disease classes.
3. **Challenges:** Class imbalance and limited dataset size were primary challenges. Despite data augmentation and transfer learning, certain models exhibited reduced precision and recall for underrepresented classes.

Overall, the findings underscore the potential of hybrid architectures like HybridCNNViT for medical imaging tasks, demonstrating that leveraging complementary strengths of CNNs and Transformers can improve classification performance.

6.2 Future Recommendation

With the goal to classify retinal diseases using OCT images, this study compared Vision Transformers, CNNs, and a proposed HybridCNNViT model. Although specific methodologies and training sets have been studied , there are many opportunities for more research. To maximize performance, for example, other hyperparameters like learning rate schedules, optimizer variations, and layer-specific modifications should be looked into. Additional opportunities for improvement are provided by structural adjustments, such as adjusting the feature fusion technique in HybridCNNViT or

experimenting with various patch sizes in ViT. To ascertain their effect on model accuracy and efficiency, combinations of these hyperparameters and structural changes could also be methodically explored. This investigation may be expanded to other medical imaging datasets in future research, allowing for the evaluation of robustness and generalizability..

References

[1] World Health Organization, "World report on vision," 2019. [Online]. Available: https://www.who.int/publications-detail-redirect/world-report-on-vision.

[2] P. Mitchell, G. Liew, B. Gopinath, and T. Y. Wong, "Age-related macular degeneration," *The Lancet*, vol. 392, no. 10153, pp. 1147–1159, 2018. DOI: 10.1016/S0140-6736(18)31550-2.

[3] P. Romero-Aroca, "Managing diabetic macular edema: The leading cause of diabetes blindness," *World Journal of Diabetes*, vol. 2, no. 6, pp. 98–104, 2011. DOI: 10.4239/wjd.v2.i6.98.

[4] R. F. Spaide, J. G. Fujimoto, and N. K. Waheed, "Optical coherence tomography angiography," *Progress in Retinal and Eye Research*, vol. 64, pp. 1–55, 2018. DOI: 10.1016/j.preteyeres.2017.11.003.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[8] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 6105–6114. DOI: 10.48550/arXiv.1905.11946.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. DOI: 10.48550/arXiv.2010.11929.

[10] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 10347–10357. DOI: 10.48550/arXiv.2012.12877.

[11] A. Srinivas, N. Linzer, J. Tan, Q. Lei, and T. Darrell, "Bottleneck transformers for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16519–16529. DOI: 10.48550/arXiv.2101.11605.

[12] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, and J. Dean, "Deep learning-enabled medical computer vision," *npj Digital Medicine*, vol. 4, p. 5, 2021. DOI: 10.1038/s41746-020-00376-2.

[13] Y. Jiang, L. Chen, H. Zhang, and X. Xiao**,** "Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module," *PLoS ONE*, vol. 14, no. 3, p. e0214587, 2019. DOI: 10.1371/journal.pone.0214587.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. DOI: 10.1145/3065386.

[15] H. Chen, Y. Wang, and H. Xie, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12299–12310. DOI: 10.48550/arXiv.2012.00364.

[16] S. d'Ascoli, H. Touvron, M. Cord, and P. Rodriguez, "ConViT: Improving vision transformers with soft convolutional inductive biases," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 2286–2296. DOI: 10.48550/arXiv.2103.10697.

[17] J. Guo, H. Touvron, P. Bojanowski, M. Douze, and M. Cord, "CMT: Convolutional neural networks meet vision transformers," *arXiv preprint arXiv:2107.06263*, 2021. DOI: 10.48550/arXiv.2107.06263.

[18] Z. Ma, Y. Zhang, and Y. Wang, "HCTNet: A hybrid ConvNet-transformer network for retinal optical coherence tomography image classification," *Biosensors*, vol. 12, no. 7, p. 542, 2022. DOI: 10.3390/bios12070542.

[19] M. Kulyabin, A. Kostromin, D. Turlapov, and E. Kharin, "OCTDL: Optical coherence tomography dataset for image-based deep learning methods," *IEEE Dataport*, 2023. DOI: 10.21227/9g8e-1w55.

[20] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct. 2019, pp. 6022–6031. DOI: 10.1109/ICCV.2019.00612.

Appendix 1

An example code of model creation using Pytorch .

```
##############################################
# Section 2: Define Models
##############################################
import torch
import torch.nn as nn
from torchvision.models import resnet50, ResNet50_Weights
from transformers import ViTModel


class HybridCNNViT(nn.Module):
    def __init__(self, num_classes, image_size=224):
        super(HybridCNNViT, self).__init__()

        # ResNet50 Backbone
        self.resnet = resnet50(weights=ResNet50_Weights.DEFAULT)
        self.resnet.fc = nn.Identity()  # Remove fully connected
layer

        # Vision Transformer (ViT-B16)
        self.vit = ViTModel.from_pretrained('google/vit-base-
patch16-224-in21k')

        # Combine ResNet and ViT features
        resnet_feature_dim = 2048  # Output from ResNet50 global
pooling
        vit_feature_dim = self.vit.config.hidden_size
        combined_feature_dim = resnet_feature_dim +
vit_feature_dim

        # Fully Connected Layers
        self.fc1 = nn.Linear(combined_feature_dim, 256)
        self.dropout = nn.Dropout(0.5)
        self.fc2 = nn.Linear(256, num_classes)
```

```
    def forward(self, x):
        # ResNet Features
        resnet_features = self.resnet(x)

        # ViT Features
        vit_features = self.vit(pixel_values=x).pooler_output

        # Combine Features
        combined_features = torch.cat((resnet_features,
vit_features), dim=1)

        # Fully Connected Layers
        x = torch.relu(self.fc1(combined_features))
        x = self.dropout(x)
        x = self.fc2(x)

        return x

# Define the function to create the model instance
def create_hybridcnnvit_model(num_classes):
    return HybridCNNViT(num_classes)
```

Appendix 2

Example code to plot and evaluate the models.

```
##############################################
# Section 5: Plot and Evaluate
##############################################
def plot_and_evaluate(model_name, model, train_loss, val_loss,
train_acc, val_acc, dataloaders, dataset_sizes):
    plt.figure(figsize=(12, 5))

    # Plot Loss
    plt.subplot(1, 2, 1)
    plt.plot(train_loss, label='Train Loss')
    plt.plot(val_loss, label='Validation Loss')
    plt.title(f'{model_name} Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()

    # Plot Accuracy
    plt.subplot(1, 2, 2)
    plt.plot(train_acc, label='Train Accuracy')
    plt.plot(val_acc, label='Validation Accuracy')
    plt.title(f'{model_name} Accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()

    plt.show()

    # Confusion Matrix and Metrics
    y_true, y_pred = [], []
    model.eval()
    with torch.no_grad():
        for inputs, labels in dataloaders['val']:
            inputs, labels = inputs.to(device), labels.to(device)
            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)
            y_true.extend(labels.cpu().numpy())
```

```
            y_pred.extend(preds.cpu().numpy())

    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 8))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=class_names, yticklabels=class_names)
    plt.title(f'{model_name} Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()

    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred,
average='weighted', zero_division=0)
    recall = recall_score(y_true, y_pred, average='weighted')
    f1 = f1_score(y_true, y_pred, average='weighted')
    cohen_kappa = cohen_kappa_score(y_true, y_pred)

    print(f"\nMetrics for {model_name}:")
    print(f"Accuracy: {accuracy:.4f}")
    print(f"Precision: {precision:.4f}")
    print(f"Recall: {recall:.4f}")
    print(f"F1_score: {f1:.4f}")
    print(f"Cohen_kappa: {cohen_kappa:.4f}")

    print("\nClassification Report:")
    print(classification_report(y_true, y_pred,
target_names=class_names))

    return {
        'model_name': model_name,
        'accuracy': accuracy,
        'precision': precision,
        'recall': recall,
        'f1_score': f1,
        'cohen_kappa': cohen_kappa
    }
```