VIETNAM NATIONAL UNIVERSITY, HANOI

INTERNATIONAL SCHOOL

GRADUATION PROJECT

PROJECT NAME

DIABETIC RETINOPATHY DETECTION USING DEEP LEARNING

Student's name

Tran Van Duat

Hanoi - Year 2025

VIETNAM NATIONAL UNIVERSITY, HANOI

INTERNATIONAL SCHOOL

GRADUATION PROJECT

PROJECT NAME

DIABETIC RETINOPATHY DETECTION USING DEEP LEARNING

SUPERVISOR: TS Pham Thi Viet Huong

STUDENT: Tran van Duat

STUDENT ID: 19071591

COHORT: ICE2019A

SUBJECT CODE: INS4014

MAJOR: Informatics and Computer Engineering

Hanoi - Year 2025

ACKNOWLEDGEMENT

Over the past 5.5 years of my education and training at the International School - Hanoi National University, I've been fortunate to receive significant support and assistance from both teachers and friends. I express my profound and heartfelt gratitude to the faculty members at the Faculty of Applied Sciences - International School. Their dedication, knowledge, and passion have been instrumental in imparting valuable insights during my time at the school.

With the attentive, nurturing, and meticulous guidance of my teachers, I have been able to complete my graduation thesis on the topic **" Diabetic Retinopathy Detection using Deep Learning"** I would like to extend my sincere gratitude to TS Pham Thi Viet Huong for her diligent guidance during each meeting and discussion regarding the subject I was working on.

In addition, I would like to express my gratitude to the Board of Rectors of the International School and the Functional Offices, who provided direct and indirect assistance to me throughout the process of studying and researching this topic.

Last but not least, my parents, relatives, and friends, who have always inspired me to be more motivated to overcome problems and barriers, deeply care about me. I was inspired to study throughout my time in school by those encouraging words.

Given my limited time and experience, there might be shortcomings in my graduation project. I genuinely hope to receive guidance and valuable input from esteemed professors to enable me to improve, enhance my understanding, and better serve practical work in the future.

I sincerely thank you!

Hanoi, January 5th, 2025

Tran Van Duat

LETTER OF DECLARATION

I hereby confirm the completion of my graduation project titled **" Diabetic Retinopathy Detection using Deep Learning "** is an independent work and has not been previously published. This topic was completed under the guidance of TS Pham Thi Viet Huong. I commit that the my project to Diabetic Retinopathy Detection using Deep Learning is independent and has not been previously published. I utilized my own knowledge and skills to develop this Project, placing a high responsibility on ethical research activities.

Every aspect and functionality on the project has been built upon my personal research and analysis. All information, images, and descriptions have been generated authentically and reliably. I guarantee that all images and content used on the project adhere to copyright regulations and do not violate the intellectual property rights of any third party.

I accept full responsibility for ensuring the accuracy of information my graduation project.

Hanoi, January 5th, 2025

Tran Van Duat

ABSTRACT

Diabetic retinopathy (DR) is a leading cause of vision impairment globally. Early and accurate detection of DR severity is crucial for timely intervention and prevention of blindness. This study investigates the potential of two cutting-edge deep learning models, Swin Transformer and FastViT, for automated DR severity classification from fundus images. To enhance the visibility of pathological features crucial for accurate classification, a novel image preprocessing pipeline is proposed. This pipeline combines Contrast Limited Adaptive Histogram Equalization (CLAHE) with Top-hat and Black-hat morphological operations, aiming to improve contrast and effectively highlight salient features like hemorrhages and exudates. We conduct a rigorous comparative assessment of Swin Transformer and FastViT, applying them to the publicly available APTOS 2019 dataset and DDR dataset, utilizing identical training parameters and the proposed preprocessing technique. This allows for a definitive evaluation of their respective performance in this critical diagnostic task. By systematically evaluating these advanced models under controlled conditions, this research aims to identify the most effective approach for automated DR severity classification, ultimately contributing to the development of reliable diagnostic tools that can aid in reducing the risk of blindness associated with diabetes.

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
APTOS 2019	Asia Pacific Tele-Ophthalmology Society 2019
AUC	Area under the curve
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional neural network
CO2	Carbon dioxide
DDR	Dataset Diabetic Retinopathy
DR	Diabetic Retinopathy
FDA	Food and Drug Administration
MHSA	Multi-Head Self-Attention
MS COCO	Microsoft Common Objects in Context
OCT	Optical Coherence Tomography
RGB	Red Green Blue
SimMIM	A simple framework for masked image modeling

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPPTER I: INTRODUCTION

1.1 What is Diabetic Retinopathy (DR)?

Many studies estimate that by 2025 the number of diabetic retinopathy patients will increase from 382 million to 592 million. This is one of many eye diseases caused by diabetes. If not analyzed and treated promptly, it can lead to partial or total blindness due to changes in the retinal blood vessels. In some cases, abnormal new blood vessels grow on the outer retina or veins inside the retina can develop and release fluid or blood, leading to fluid retention in the retina. Unfortunately, in the early stages of diabetic retinopathy, there are often no visual impairments and no obvious symptoms that can only be detected by regular eye exams. Over time, the disease can cause serious disorders such as aneurysms, exudates, hemorrhages or small spots that obscure vision.

1.2 Why is Early Detection Crucial?

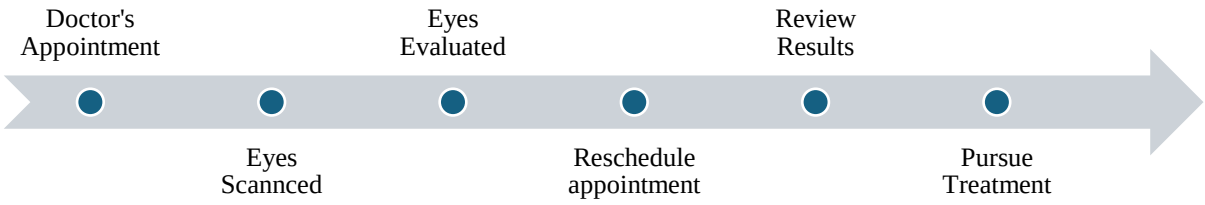

Figure 1.1: Medical examination and treatment process

According to medical research, diabetic retinopathy can occur at any age, the risk of developing the disease increases in the elderly. It is estimated that 93 million diabetic patients worldwide have diabetic retinopathy. Diabetic retinopathy usually has 4 stages (0 - No DR, 1 – Mild, 2 – Moderate, 3 – Severe, 4 - Proliferative DR). The process for diagnosing and detecting the current levels of diabetic retinopathy is quite cumbersome and time-consuming, from making an appointment with a doctor, then moving on to scanning the eyes for a preliminary diagnosis based on the doctor's personal experience, then scheduling a follow-up appointment, reviewing the results and proceeding with treatment.

So this project proposes a machine learning model that supports the early detection of diabetic retinopathy, helping doctors shorten the process and can then recommend appropriate advice to patients on the same day.

1.3 Deep learning-based solution

Deep Learning offers a powerful solution for improving diabetic retinopathy detection. Advantages of Transformers for Diabetic Retinopathy detection:

Superior Global Context Learning:

- Self-Attention is Key: The core mechanism of Transformers, self-attention, allows the model to attend to all regions of the retinal image simultaneously and learn the relationships between them, regardless of distance.
- Crucial for DR: Diabetic Retinopathy often manifests as multiple scattered lesions across the retina (hemorrhages, microaneurysms, exudates, etc.). The relationship and distribution of these lesions are crucial for diagnosis. Transformers can capture these global correlations more effectively.
- CNN Limitations: CNNs, with their local receptive fields, have more difficulty learning relationships between distant regions in the image, especially in the initial layers of the network.

Efficient Handling of High-Resolution Retinal Images:

- Detail-Rich Retinal Images: Retinal images are typically high-resolution to clearly show small blood vessels and subtle lesions.
- Swin Transformer Solves the Problem: Transformer variants like Swin Transformer use a shifted window mechanism to compute self-attention within local windows, gradually expanding the scope. This significantly reduces computational cost when processing high-resolution images while maintaining the ability to learn global relationships.
- Advantage over CNNs: Traditional CNNs struggle with high-resolution images due to a sharp increase in computational cost. Downsampling images to reduce this cost can lead to the loss of important details.

Flexibility and Potential for Transfer Learning:

- Easily Customizable Architecture: The Transformer architecture is flexible, allowing for easy adjustments, additions, and removals of components to fit the specific requirements of the DR task.
- Pre-trained Models: Large Transformer models, pre-trained on massive image datasets (like ImageNet), can be fine-tuned for the DR task. This helps to Improve Performance, Reduce Training Time and Reduce Data Requirements.

Research Trend: Transformers are becoming a major research trend in the field of computer vision and medical applications, indicating their great potential.

Transformers, with their ability to learn global relationships, process high-resolution images efficiently, their flexibility and potential for transfer learning, along with impressive experimental results, are a good and promising solution for improving the ability to detect diabetic retinopathy. They help overcome the limitations of traditional CNNs, opening up new avenues in automated diagnosis, assisting doctors in diagnosing the disease earlier and more accurately.

1.4 Research objectives

The objective of this research is:

- Investigate the application of Swin Transformer and FastViT, two advanced deep learning models, to accurately classify the severity of diabetic retinopathy from fundus images.
- Implement a novel image preprocessing approach, combining CLAHE with Top-hat and Black-hat morphological operations, to effectively highlight pathological features.
- Provide a definitive comparative assessment of both Swin Transformer and FastViT by subjecting them to identical parameters and preprocessing conditions, identifying the most effective model for this critical diagnostic application.

1.5 Assessment literature review

1.5.1 Diabetic Retinopathy for Swin Transformer V2 model

A.Dihin, Rasha & Alshemmary, Ebtesam & Al-Jawher, Waleed. (2023). Diabetic Retinopathy Classification Using Swin Transformer with Multi Wavelet. Journal of Kufa for Mathematics and Computer. 10. 167-172. 10.31642/JoKMC/2018/100225.

Objective: To develop a novel method for classifying diabetic retinopathy (DR) into five severity levels based on the Swin Transformer architecture combined with Multi Wavelet decomposition.

Preprocessing:

- Color Space Conversion: RGB retinal images are converted to the YCbCr color space.
- Y Channel Extraction: The Y channel (luminance) is extracted for further processing.
- Resizing: Images are resized to 224x224 pixels.
- Multi Wavelet Decomposition: Multi Wavelet decomposition is applied to the Y channel to extract features from different frequency bands.

Techniques:

- Swin Transformer: Used as the feature extractor. Swin Transformer's shifted window approach allows for efficient processing of high-resolution images.
- Classification: Features from the Swin Transformer are fed into a fully connected layer for classification into five severity levels of diabetic retinopathy.
- Optimization: The model is trained using the Adam optimizer and cross-entropy loss function.

Results:

- Dataset: APTOS 2019 Blindness Detection dataset.
- Accuracy: 98.57%
- Sensitivity: 97.68%

- Specificity: 98.91%
- F1-score: 98.13%
- AUC: 99.73%

Conclusion: This paper proposes an effective method for multi-class diabetic retinopathy classification using Swin Transformer and Multi Wavelet. It achieves impressive results, outperforming other methods on the same dataset. The combination of Swin Transformer and Multi Wavelet allows for robust feature extraction, contributing to high classification performance.

Li, Zhenwei & Han, Yanqi & Yang, Xiaoli. (2023). Multi-Fundus Diseases Classification Using Retinal Optical Coherence Tomography Images with Swin Transformer V2. Journal of Imaging. 9. 203. 10.3390/jimaging9100203.

Objective: To propose a method for classifying multiple fundus diseases using retinal Optical Coherence Tomography (OCT) images based on the Swin Transformer V2 model.

Preprocessing:

- Normalization: Pixel values of OCT images are normalized to the range [0, 1].
- Data Augmentation: Techniques like rotation, flipping, and shifting are applied to increase training set size and improve model generalization.
- Resizing: Not explicitly mentioned, but Swin Transformer V2 can handle different input sizes.

Techniques:

- Swin Transformer V2: Used as the main feature extractor. This improved version incorporates:
    - Residual-post-norm: For more stable training and improved accuracy.
    - Scaled cosine attention: Replaces dot-product attention to avoid extreme values.
    - Log-spaced continuous position bias: Allows the model to handle different image sizes more effectively.

- Classification: Features extracted by Swin Transformer V2 are passed to a fully connected layer for multi-class classification of fundus diseases.
- Optimization: The model is trained using the AdamW optimizer and cross-entropy loss function.

Results:

- Datasets:
  - OCT-C8 (8-class classification)
  - Kermany (4-class classification)
- Accuracy:
  - OCT-C8: 99.38%
  - Kermany: 99.53%
- F1-score:
  - OCT-C8: 99.38% (average)
  - Kermany: 99.54% (average)
- AUC:
  - OCT-C8: 99.93% (average)
  - Kermany: 99.98% (average)

Conclusion: This paper presents an effective method for classifying multiple fundus diseases using OCT images and Swin Transformer V2. The model achieves state-of-the-art results on two public datasets, demonstrating its potential for automated diagnosis of fundus diseases. The use of Swin Transformer V2 contributes significantly to the high performance and flexibility in handling different image sizes.

A.Dihin, Rasha & Alshemmary, Ebtesam & Al-Jawher, Waleed. (2023). Automated Binary Classification of Diabetic Retinopathy by SWIN Transformer. Journal of Al-Qadisiyah for Computer Science and Mathematics 15. 10.29304/jqcm.2023.15.1.1166.

Objective: To develop an automated system for binary classification of diabetic retinopathy (DR), distinguishing between images with DR and without DR, using the Swin Transformer model.

Preprocessing:

- Color Space Conversion: RGB retinal images are converted to the YCbCr color space.
- Y Channel Extraction: Only the Y channel (luminance) is used.
- Resizing: Images are resized to 224x224 pixels.
- Wavelet Decomposition: Multi-Level Wavelet decomposition is applied to the Y channel for feature extraction.

Techniques:

- Swin Transformer: Used as the feature extractor, similar to the first paper.
- Binary Classification: Features from the Swin Transformer are fed into a fully connected layer with a sigmoid activation function for binary classification (DR present or not).
- Optimization: The model is trained using the Adam optimizer and binary cross-entropy loss function.

Results:

- Dataset: Messidor-2 dataset.
- Accuracy: 98.95%
- Sensitivity: 98.48%
- Specificity: 99.23%
- F1-score: 98.86%
- AUC: 99.58%

Conclusion: This paper proposes a method for binary classification of diabetic retinopathy using Swin Transformer and Wavelet decomposition. The model achieves very high performance on the Messidor-2 dataset, highlighting its potential for automated DR screening. The combination of Swin Transformer and Wavelet analysis provides effective feature extraction from fundus images, resulting in excellent classification accuracy.

1.5.2 Diabetic Retinopathy for FastViT model

Vasu, Pavan & Gabriel, James & Zhu, Jeff & Tuzel, Oncel & Ranjan, Anurag. (2023). FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization. 10.48550/arXiv.2303.14189.

Objective: To introduce FastViT, a new hybrid Vision Transformer architecture that achieves high accuracy while maintaining a significantly faster inference speed compared to existing Vision Transformers and Convolutional Neural Networks. The authors aim to bridge the gap between high accuracy and fast inference in visual recognition tasks.

Preprocessing:

- Resizing: Images are resized to 256x256.
- Random Cropping: Random crops of size 224x224 are taken during training.
- Random Horizontal Flipping: Images are randomly flipped horizontally.
- Normalization: Pixel values are normalized using the ImageNet mean and standard deviation.

Techniques:

- Hybrid Architecture: FastViT combines the strengths of both Transformers and CNNs in a novel way:
    - Token Mixing with MHSA (Multi-Head Self-Attention): Used in the early and final stages for global context aggregation, similar to traditional Vision Transformers.
    - RepMixer (New Contribution): This replaces MHSA in the middle stages. RepMixer is the key innovation that uses structural reparameterization to achieve fast mixing of tokens. It leverages large kernel convolutions during training and efficiently folds them into smaller kernels during inference, reducing computational cost without sacrificing accuracy.
        - Training: Uses large kernel (e.g., 7x7) convolutions.

        - Inference: Folds these into smaller kernel (e.g., 3x3) convolutions and point-wise operations, significantly reducing computational cost.
    - Hierarchical Structure: The architecture follows a hierarchical structure, progressively downsampling the spatial resolution and increasing the channel dimension, similar to many CNNs and Vision Transformers.
- Structural Reparameterization: As mentioned above, this technique is applied within the RepMixer to achieve efficiency. It involves transforming the network's structure from a more complex form during training to a simpler, faster form during inference while preserving the learned representation.
- Optimization:
    - Optimizer: AdamW optimizer with a weight decay of 0.1.
    - Learning Rate: Initial learning rate of 4e-4 for the T-series models, and 1e-3 for the S, M, and L models, with cosine learning rate decay.
    - Batch Size: 4096 (across 64 TPU-v3 chips).
    - Training Epochs: 600 epochs.
    - Regularization: Mixup, Cutmix, RandAugment, and Repeated Augmentation are used for regularization.

Results:

- Dataset: Primarily ImageNet-1K for image classification. Limited experiments on MS COCO for object detection and ADE20K for semantic segmentation.
- Key Metrics:
    - Top-1 Accuracy: Measures the accuracy of the top prediction.
    - Throughput (images/sec): Measures the inference speed.
    - Latency (ms): Measures the time taken for a single inference.
- Main Findings:
    - State-of-the-art Trade-off: FastViT achieves a superior trade-off between accuracy and inference speed compared to existing models like Swin Transformers, ConvNeXts, and EfficientNets.

- High Accuracy: FastViT models reach up to 85.5% top-1 accuracy on ImageNet-1K.
- Fast Inference: FastViT models achieve significantly higher throughput and lower latency compared to other models with comparable accuracy. For example, FastViT-SA12 is 2.1x faster than ConvNeXt-T, while maintaining higher accuracy.
- Effectiveness of RepMixer: The RepMixer module is shown to be crucial for achieving fast inference while retaining high accuracy.
- Scalability: FastViT models of varying sizes (T, S, M, L) demonstrate consistent improvements in accuracy and efficiency as the model size increases.

Conclusion: The paper presents FastViT, a novel and efficient Vision Transformer architecture that leverages structural reparameterization through the RepMixer module. FastViT achieves a new state-of-the-art trade-off between accuracy and inference speed on ImageNet-1K, outperforming existing models. This makes it a promising architecture for real-world applications where both high accuracy and fast inference are critical. While the paper focuses primarily on ImageNet, the principles behind FastViT are likely applicable to other domains, including medical image analysis.

1.5.3 Using Top hat and Black hat for Diabetic Retinopathy

Hou, Yanli. (2014). Automatic Segmentation of Retinal Blood Vessels Based on Improved Multiscale Line Detection. Journal of Computing Science and Engineering. 8. 119-128. 10.5626/JCSE.2014.8.2.119.

Multidirectional Morphological Top-hat Transform: A multidirectional morphological White Top-hat transform with rotating structuring elements is applied. This step serves two primary purposes:

- Decrease Optic Disk Influence: The optic disk is a bright, circular region in the fundus image that can interfere with vessel segmentation. The Top-hat transform helps to suppress this bright region, reducing its influence on subsequent steps.

- Emphasize Vessels: The White Top-hat transform enhances bright, linear structures like blood vessels against the background. Using rotating structuring elements allows the method to detect vessels oriented in various directions.

CHAPPTER II: RESEARCH APPROACH AND METHODOLOGY

2.1 Deep learning

Deep learning is a more specialized self-learning method in Machine Learning with specific characteristics and high complexity. In fact, self-learning in phase 1 cannot help Artificial Intelligence (AI) solve complex problems.

Deep Learning involves multiple hidden layers in a neural network and the last layer is the one that goes through the number of layers and complexity of the network is called depth.

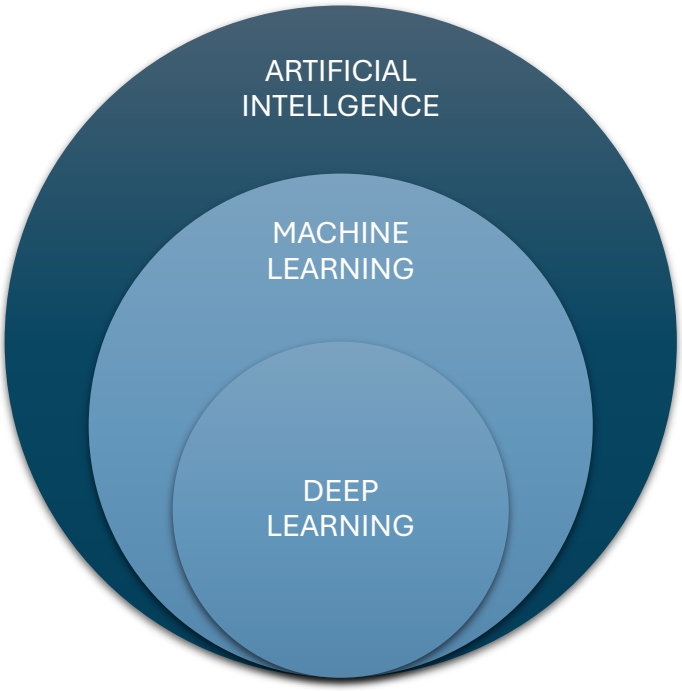

Figure 2.1: AI vs Machine learning vs Deep learning

A significant advancement in deep learning involves the substantial increase in neural network depth, transitioning from a few layers to hundreds. More depth means the ability to recognize larger patterns, with a larger pool of information increasing the ability to pick up on objects that are broader and more detailed.

Deep learning has been particularly effective in medical imaging, due to the availability of high-quality data and the ability of convolutional neural networks to classify images. For example, deep learning is as effective as or more effective than dermatologists in classifying skin cancer. Several vendors have received Food and Drug Administration (FDA) approval for deep learning algorithms for diagnostic

purposes, including image analysis for oncology and retinal diseases. Deep learning has also made significant contributions to improving the quality of healthcare by using data from electronic health records to predict medical events.

For example, when recognizing a photo, AI self-learning in stage 1 can only distinguish simple details such as the light level of the photo. This processing does not help identify objects in the photo. Then it is necessary to develop AI's self-learning ability at a higher stage called deep learning so that AI can process important information from the large amount of data provided. Retinal image diagnostics, for instance, demand sophisticated processing. This involves the identification and prioritization of key features such as hemorrhages, aneurysms, and exudates. Precise diagnostic conclusions are subsequently derived from this refined data analysis

2.2 Transformer

2.2.1 History of Transformers

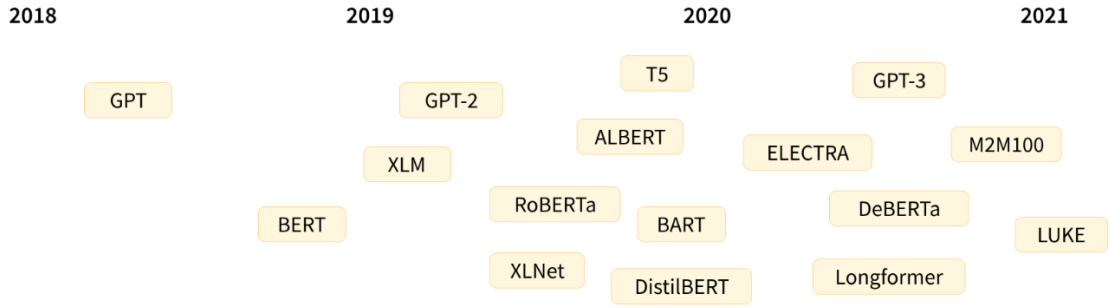

Figure 2.2 A brief timeline of the development of Transformer models

The Transformer architecture was introduced in June 2017. The initial research focus was on translation tasks. This was followed by the introduction of several influential models.

2.2.2 Transformers are large models

Beyond a few exceptions (like DistilBERT), the general strategy for achieving better performance has been to increase the size of the models as well as the amount of data they are pre-trained on.

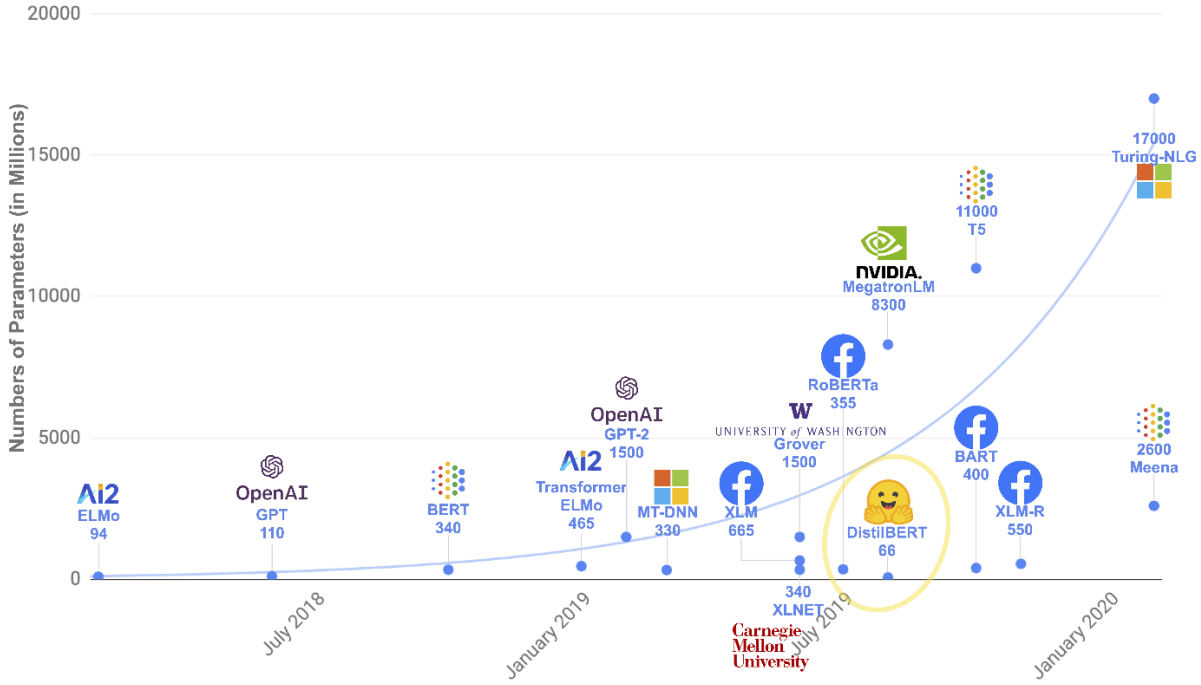

Figure 2.3 Transformers are large models

Unfortunately, training a model, especially a large one, requires massive amounts of data. This becomes very expensive in terms of time and computational resources. It even translates into an environmental impact, as can be seen in the following chart.

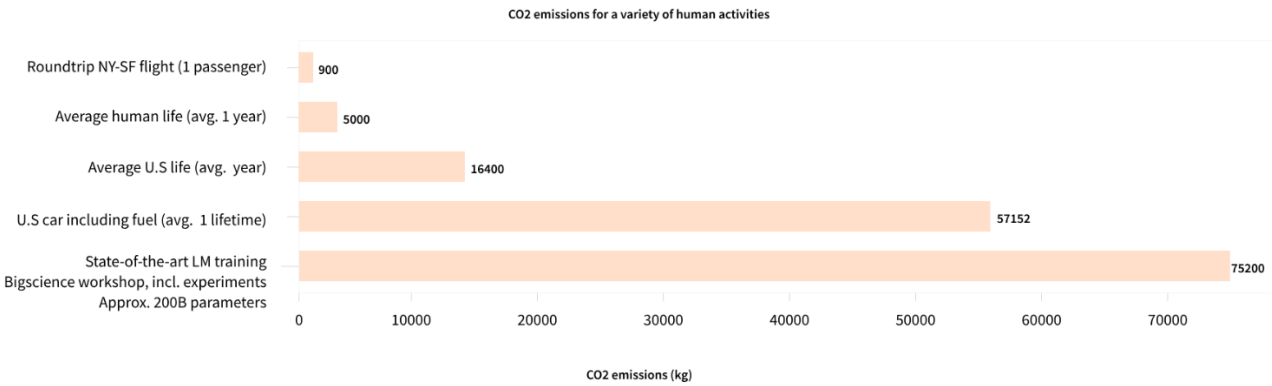

Figure 2.4 CO2 emissions for a variety of human activities

2.2.3 Architecture for Transformer

The model is composed of two blocks:

- Encoder (left): The encoder receives an input and builds a representation of it (its features). This means that the model is optimized to acquire an understanding from the input.

- Decoder (right): The decoder uses the encoder's representation (features) along with other inputs to generate a target sequence. This means that the model is optimized for generating outputs.

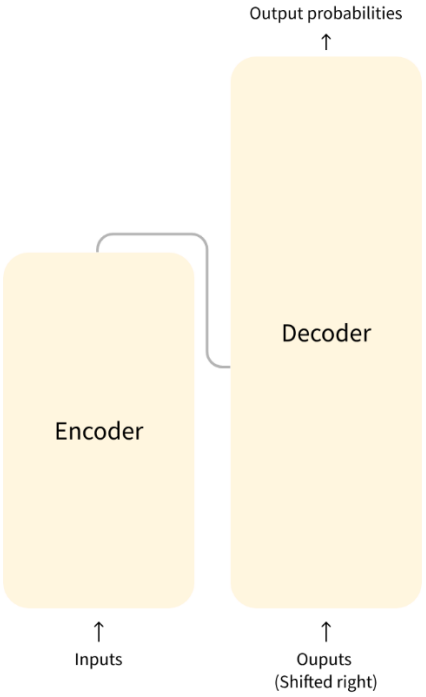

Figure 2.5 Two Blocks of Transformer

Can be used independently, depending on the task:

- Encoder-only models: Good for tasks that require understanding of the input, such as sentence classification and named entity recognition.
- Decoder-only models: Good for generative tasks like text generation.
- Encoder-decoder models or sequence-to-sequence models: Good for generative tasks that require an input, such as translation or summarization.

Attention Layers: A key feature of Transformer models is that they are built with special layers called attention layers.

Architecture (Example in Natural Language Processing):

The original Transformer architecture was designed for translation. During training, the encoder receives an input (sentence) in a given language, while the decoder receives the same sentences in the desired target language. In the encoder, the attention layers can use all the words in a sentence. However, the decoder works

sequentially and can only pay attention to the words in the sentence that it has already translated.

To speed things up during training, the decoder is fed the whole target, but it's not allowed to use future words.

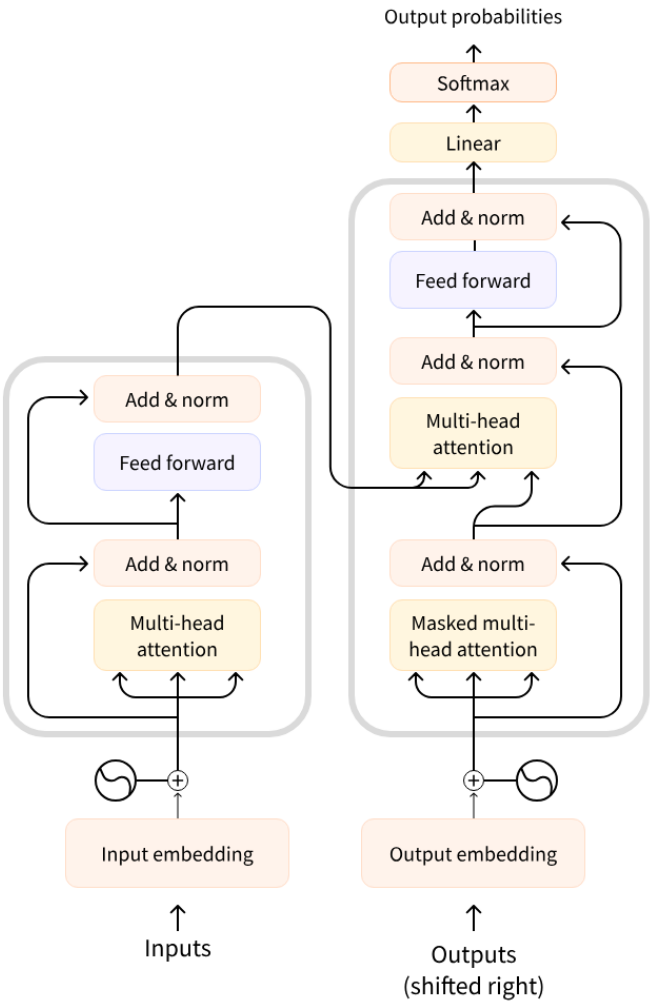

Figure 2.6 Architecture Transformer

The first attention layer in the decoder pays attention to all (past) inputs to the decoder, but the second attention layer uses the output of the encoder. Therefore, it can access the whole input sentence to best predict the current word.

The attention mask can also be used in the encoder/decoder to prevent the model from paying attention to some special words.

2.3 Swin Transformer V2 Model

Swin Transformer v2 model pre-trained on ImageNet-1k at resolution 256x256. It was introduced in the paper Swin Transformer V2: Scaling Up Capacity and Resolution by Liu et al.

Model description

The Swin Transformer is a type of Vision Transformer. It builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. In contrast, previous vision Transformers produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally.

Swin Transformer v2 adds 3 main improvements:

- A residual-post-norm method combined with cosine attention to improve training stability.
- A log-spaced continuous position bias method to effectively transfer models pre-trained using low-resolution images to downstream tasks with high-resolution inputs.
- A self-supervised pre-training method, SimMIM, to reduce the needs of vast labeled images.

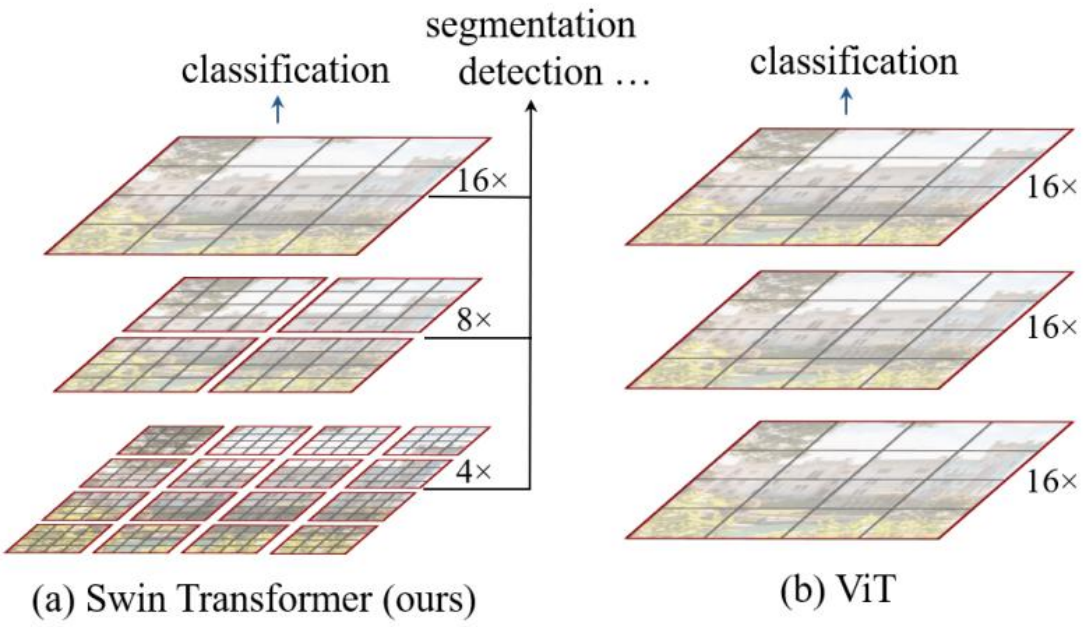

Figure 2.7 Architecture Swin Transformer v2

2.4 FastViT Model

Model description

FastViT is a new hybrid vision transformer architecture that combines elements of transformer and convolutional designs, providing an optimal balance between accuracy and efficiency. It introduces a novel token mixing operator called RepMixer, which lowers memory access costs by eliminating skip-connections. The model also uses train-time overparametrization and large kernel convolutions to enhance accuracy without significantly affecting latency.

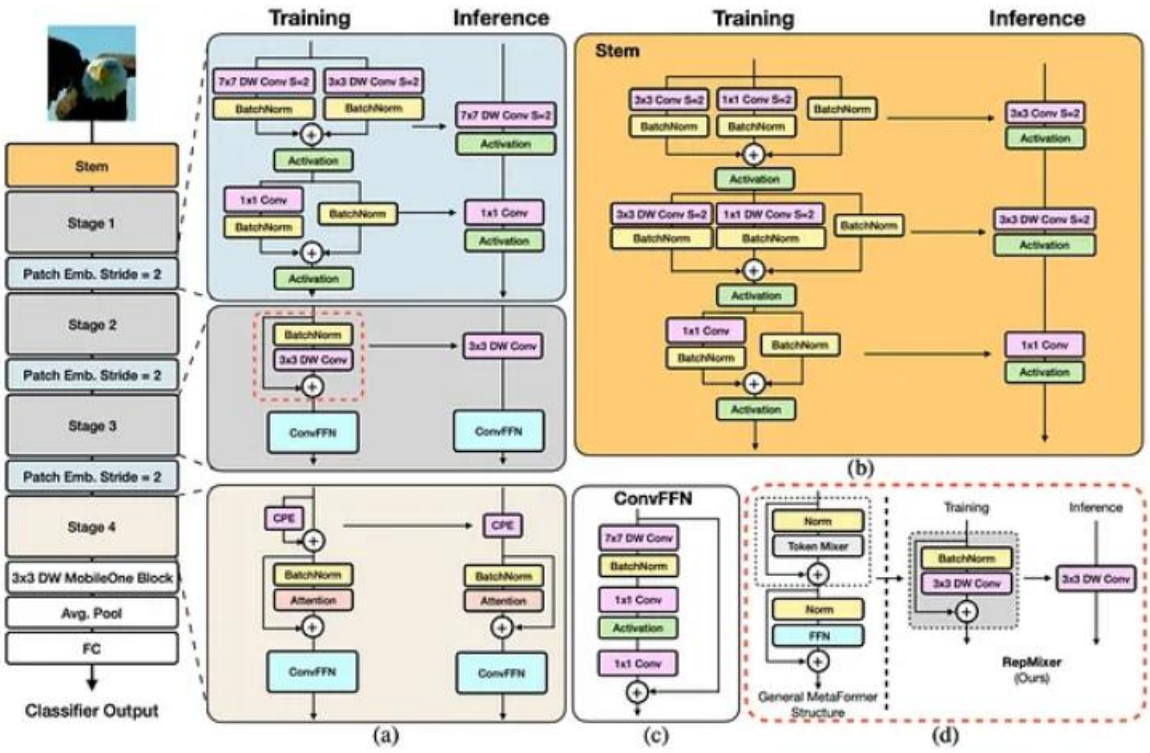

Figure 2.8 Architecture FastViT Model

2.5 Performance metrics

To measure the performance of model, different metrics are used to classification of eye diseases. In this section, we will provide a brief overview of these metrics.

2.5.1 Accuracy

The percentage of correct predictions (both positive and negative) out of the total predictions.

```python
@torch.no_grad()
def accuracy(output, target):
    with torch.no_grad():
        _, predicted = output.max(1)  # Get the class index with the highest score
        correct = predicted.eq(target).sum().item()  # Count correct predictions
        total = target.size(0)  # Total number of samples
        accuracy = 100.0 * correct / total  # Compute accuracy percentage
    return accuracy

# ... inside train_step and val_step functions ...
metrics["accuracy"] = 100.0 * correct_predictions / total_samples
```

Figure 2.9 Image Code Accuracy

Formula: (Number of Correct Predictions) / (Total Number of Predictions) * 100%

Significance: This metric indicates the overall correctness of the model's classifications.

Calculated for both the training set (train_accuracy) and the validation set (val_accuracy) after each epoch.

2.5.2 Loss

A function that measures the difference between the model's predicted values and the actual values. Lower loss values indicate a better model.

Type of Loss Function Used: **FocalLoss** (a variation of CrossEntropyLoss)

```
from sklearn.utils.class_weight import compute_class_weight
# Extract the class labels (diagnosis column)
class_labels = train_df['diagnosis'].values

# Compute class weights using sklearn
unique_classes = np.unique(class_labels)
class_weights = compute_class_weight('balanced', classes=unique_classes, y=class_labels)

# Create a dictionary that maps class labels to their corresponding weights
class_weight_dict = {class_label: weight for class_label, weight in zip(unique_classes, class_weights)}

print("Class Weights:", class_weight_dict)
```

```
Class Weights: {0: 0.6028808864265928, 1: 1.0882, 2: 1.0892892892892894, 3: 2.536596736596737, 4: 0.9008278145695364}
```

Figure 2.10 Image code Class Weights

Class Weights: {0: 0.6028808864265928, 1: 1.0882, 2: 1.0892892892892894, 3: 2.536596736596737, 4: 0.9008278145695364}

```
import torch.nn.functional as F
class FocalLoss(nn.Module):
    def __init__(self, class_weights, gamma=1.5, reduction='mean'):
        super(FocalLoss, self).__init__()
        self.alpha = torch.tensor(class_weights).float().to(device)
        self.gamma = gamma
        self.reduction = reduction

    def forward(self, inputs, targets):
        targets = targets.long() # Đảm bảo targets là kiểu Long
        ce_loss = F.cross_entropy(inputs, targets, reduction='none')
        pt = torch.exp(-ce_loss)
        focal_loss = self.alpha[targets] * (1 - pt)**self.gamma * ce_loss
        if self.reduction == 'mean':
            return focal_loss.mean()
        elif self.reduction == 'sum':
            return focal_loss.sum()
        else:
            return focal_loss

criterion = FocalLoss(class_weights=list(class_weight_dict.values()), gamma=2, reduction='mean').to(device)
```

Figure 2.11 Image code Focal loss

- **Focal Loss** is employed to address class imbalance in the dataset. It focuses on hard-to-classify samples by down-weighting the contribution of easy-to-classify samples. The gamma parameter adjusts the level of focus on hard examples. class_weights are computed to balance the influence of classes with different numbers of samples

Significance: Indicates the magnitude of the model's error. Lower loss suggests the model is learning better.

Calculated for both the training set (train_loss) and the validation set (val_loss) after each epoch.

2.5.3 Confusion Matrix

A table that shows the counts of true positive, true negative, false positive, and false negative predictions for each class.1 Rows represent the actual class, and columns represent the predicted class.

```
cm = confusion_matrix(y_true_val, y_pred_val)

# Plot confusion matrix using seaborn
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=[label_title[str(i)] for i in range(cm.shape[0])],
            yticklabels=[label_title[str(i)] for i in range(cm.shape[0])])
# ...
```

Figure 2.12 Image code Confusion Matrix

Significance: Provides a detailed view of the model's performance on each class, helping to identify classes where the model is struggling.

Calculated and displayed after training is complete.

2.5.4 Classification Report

A report that provides key metrics for each class

Precision: (True Positives) / (True Positives + False Positives). Out of all the samples predicted as positive, how many were actually positive?

Recall: (True Positives) / (True Positives + False Negatives). Out of all the actual positive samples, how many were correctly predicted?

F1-score: The harmonic mean of Precision and Recall. (2 * Precision * Recall) / (Precision + Recall). Balances Precision and Recall.

Support: The number of actual samples in each class.

```
from sklearn.metrics import classification_report
# ...
report = classification_report(y_true_val, y_pred_val, digits=2)
print("\nClassification Report:\n", report)
```

Figure 2.13 Image Code Classification

Significance: Provides a more comprehensive evaluation of the model's performance on each class compared to just using accuracy. Calculated and printed after training is complete.

2.5.5 Error Rate per Class

The percentage of misclassified samples for each class.

Significance: Helps identify classes where the model is performing poorly and needs improvement.

CHAPPTER III: BUILDING DR DETECTION SYSTEM

3.1 Dataset

3.1.1 Dataset APTOS 2019

This is a dataset taken from the 2019 Kaggle competition provided by Aravind Eye Hospital to diagnose diseases based on retinal images collected in India. According to the project information provided, this is a dataset taken by hospital technicians traveling to remote rural areas, then reviewed and diagnosed by highly skilled doctors. The dataset includes 5590 high-resolution images. Since it is a competition dataset, 3662 images are labeled according to their disease severity. The dataset includes five disease severity levels on a scale of 0 – 4:

0 - No DR

1 - Mild

2 - Moderate

3 - Severe

4 - Proliferative DR

Diabetic Retinopathy Stage	**Number of Examples**
0 - No DR	1805
1 - Mild	370
2 - Moderate	999
3 - Severe	193
4 - Proliferative DR	295

Table 3.1 Number image for layer in APTOS 2019 dataset

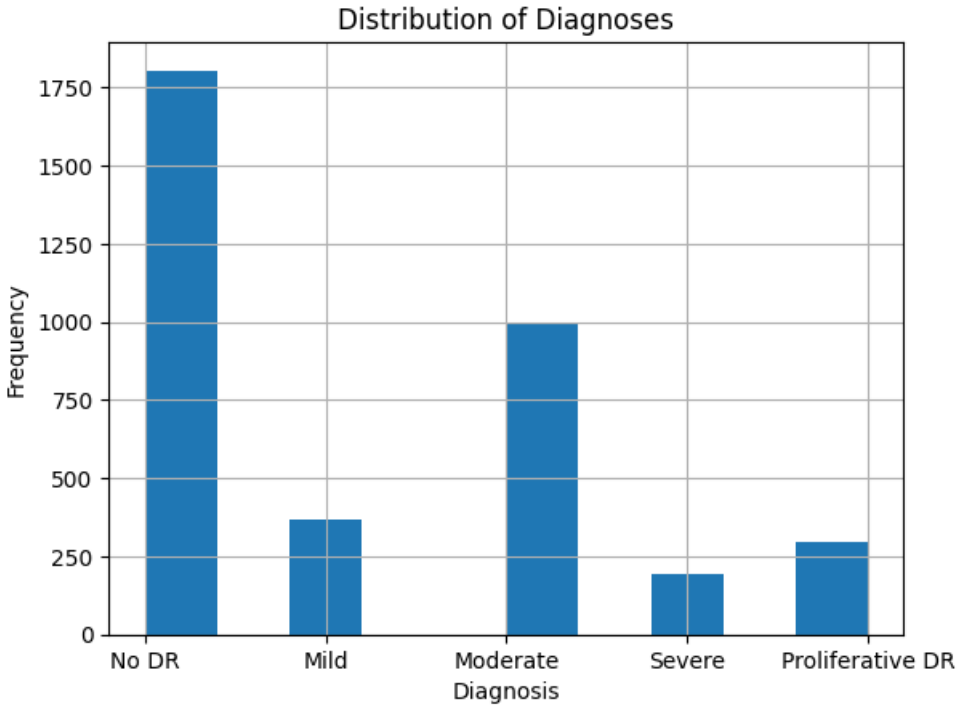

Figure 3.1 Distribution of Diagnoses for APTOS 2019 dataset

Link dataset: https://www.kaggle.com/competitions/aptos2019-blindness-detection/data

3.1.2 Dataset DDR dataset

The DDR dataset is a collection of patient data sourced from diverse locations worldwide, and has been evaluated and classified by experts. The dataset comprises 1779 fundus images and focuses on three levels of disease severity:

1 - Mild

3 - Severe

4 - Proliferative DR

Diabetic Retinopathy Stage	**Number of Examples**
1 - Mild	630
3 - Severe	236
4 - Proliferative DR	913

Table 3.2 Number image for layer in DDR dataset

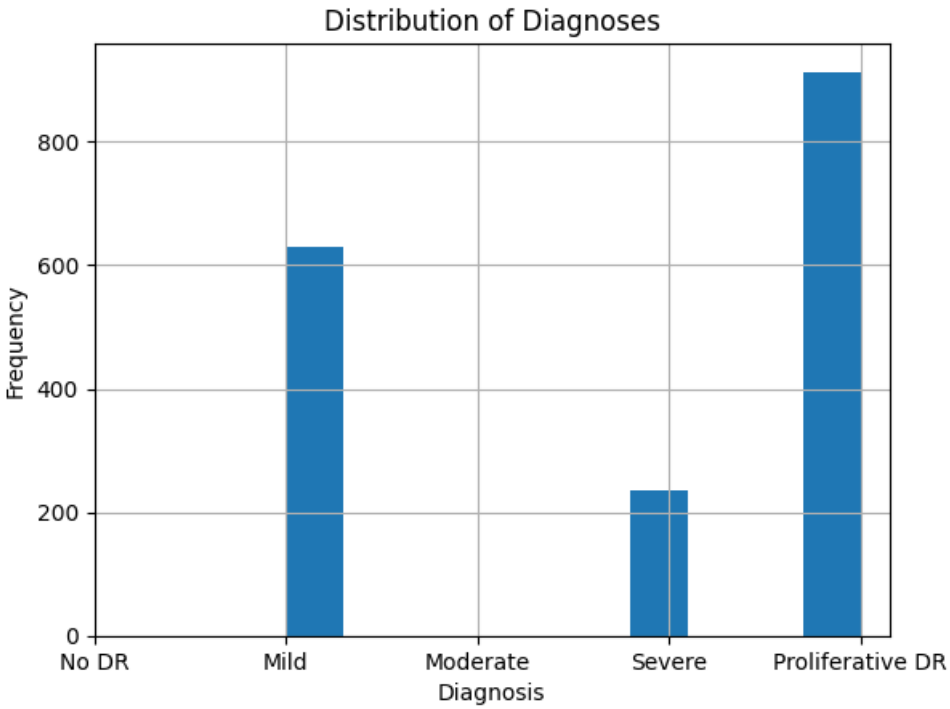

Figure 3.2 Distribution of Diagnoses for DDR dataset

Link dataset: https://www.kaggle.com/datasets/duttrnvn/datasets-dr/data

3.1.3 Combine datasets

Diabetic Retinopathy Stage	**Number of Examples**
0 - No DR	1805
1 - Mild	1000
2 - Moderate	999
3 - Severe	429
4 - Proliferative DR	1208

Table 3.3 Number image for layer in New Dataset

The combined dataset, resulting from merging the two original datasets, provides a richer and more extensive training set, improving model performance and reducing the potential for overfitting.

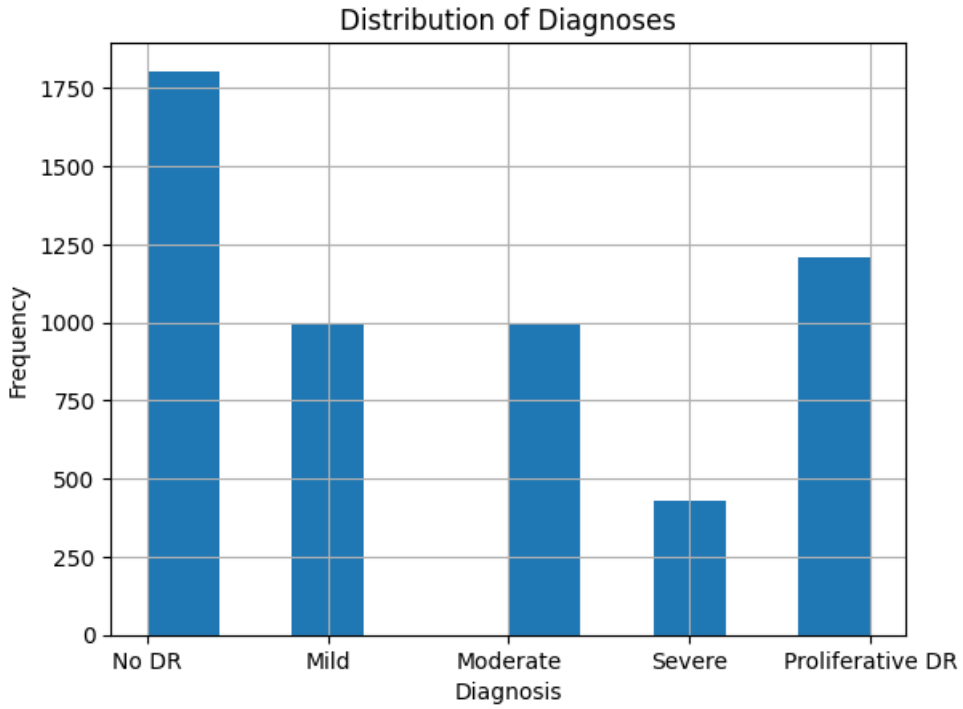

Figure 3.3 Distribution of Diagnoses for New Dataset

The bar chart visualizes the distribution of diagnoses in a dataset related to diabetic retinopathy. The x-axis delineates five levels of severity, ranging from No DR to Proliferative DR, while the y-axis indicates the frequency of each diagnosis. The most striking feature of this distribution is the significant overrepresentation of the No DR category, with approximately 1805 cases, compared to Mild and Moderate DR with approximately 1000 cases each. The Severe category, at around 429 cases, and Proliferative DR, at over 1208 cases, are less frequent. This skewed distribution highlights the challenges in training a balanced and accurate diagnostic model, as the model might become biased towards the dominant No DR class. Addressing this class imbalance will be crucial for developing a clinically useful model.

The following are some representative images taken from the newly merged dataset:

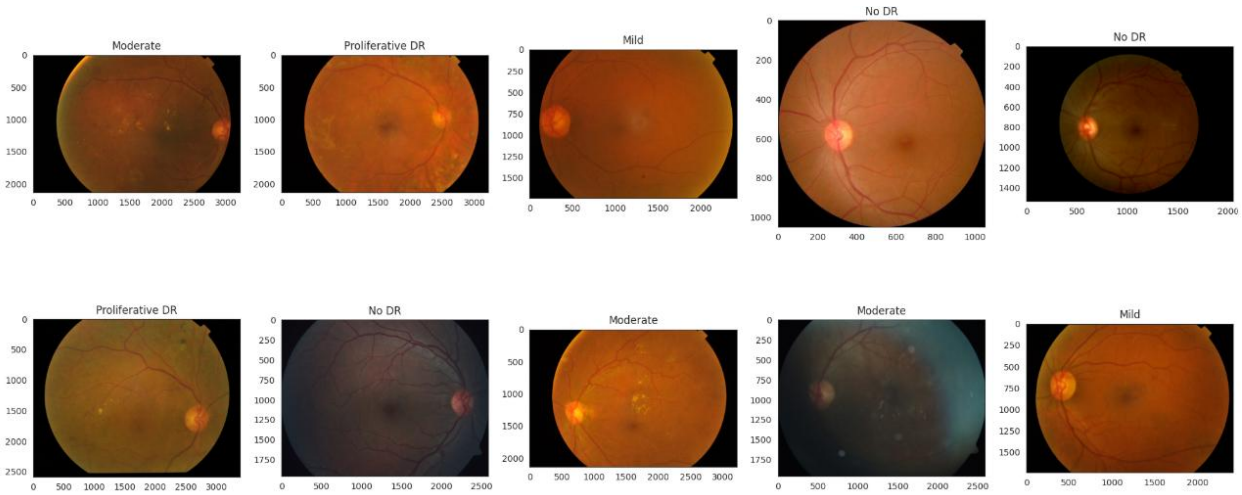

Figure 3.4 Image in Dataset

While the image quality is generally satisfactory, there is a lack of uniformity in image dimensions, and a significant portion of the images appear to be underexposed.

3.2 Workflow

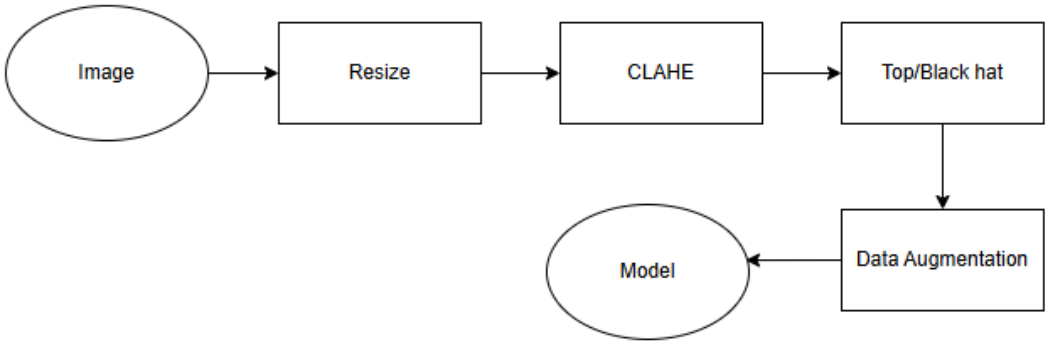

Figure 3.5 Workflow

In this workflow, the input data will undergo the following process:

- Images will be resized to 256x256 pixels.
- Images will be split into separate color channels to apply CLAHE.
- The red and green channels will be further processed using Top-hat and Black-hat transformations to enhance key image details for detecting diabetic retinopathy.
- The dataset will be divided into training and validation sets with a ratio of 80% for training and 20% for validation.

- Data augmentation will be applied to the training set to diversify the training data.
- Finally, the processed data will be fed into models A and B for training." or "Finally, the resulting data will be used to train models swinv2_small_window16_256 and fastvit_s12.

3.3 Data Pre-processing

3.3.1 Resize

The purpose of resizing in image processing for machine learning, is to standardize the input size of the images, to reduce data dimensionality and improve the model's performance.

Standardizing Input Size:

- Consistent Input for the Model: Deep Learning models typically require inputs of a fixed size. Images in real-world datasets often come in varying dimensions. Resizing all images to a uniform size (e.g., 256x256) ensures that the model can process all images in the dataset.
- Compatibility with Model Architecture: Certain network architectures, like the Swin Transformer or Fastvit Transformer used in this case, are designed to perform optimally with a specific input size. Resizing images to that size allows the model to leverage its full capabilities.

Reducing Data Dimensionality:

- Decreased Computational Complexity: High-resolution images contain a large number of pixels, leading to significant computational cost during model training. Resizing images to a smaller size reduces the number of pixels, thus decreasing computational complexity and speeding up the training process.
- Reduced Storage Space: Smaller image sizes require less storage space.
- Noise Reduction: In some cases, reducing the image size can help eliminate noise or irrelevant details, focusing on the main features of the image.

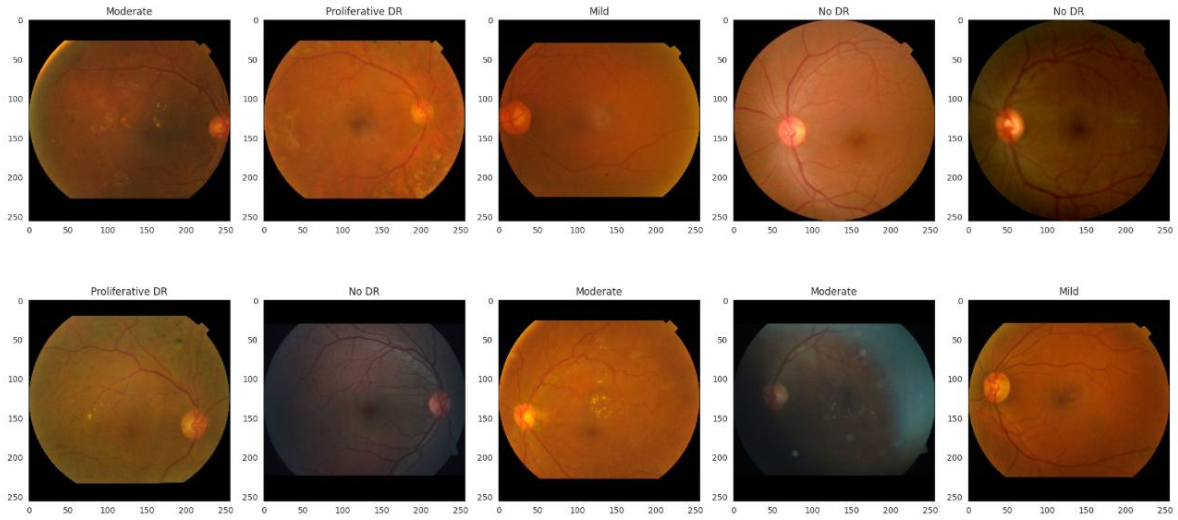

Figure 3.6 Resize image

3.3.2 CLAHE

CLAHE is a technique used to enhance the contrast of an image, highlighting important details, and improving the performance of machine learning models.

```
red, green, blue = cv2.split(image)

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))

red = clahe.apply(red)

green = clahe.apply(green)

blue = clahe.apply(blue)
```

The function process_image(image, img_size=256) uses CLAHE to enhance contrast.

cv2.createCLAHE(clipLimit = 2.0, tileGridSize = (8, 8)) creates a CLAHE object with:

- clipLimit = 2.0: The contrast limit threshold. The histogram of each tile will be clipped at this threshold.
- tileGridSize=(8, 8): The size of the tiles (8x8).

clahe.apply(red), clahe.apply(green), clahe.apply(blue): Applies CLAHE separately to each color channel (red, green, blue) of the image. Applying it to each channel individually enhances color contrast more effectively.

Apply CLAHE for Channel Red

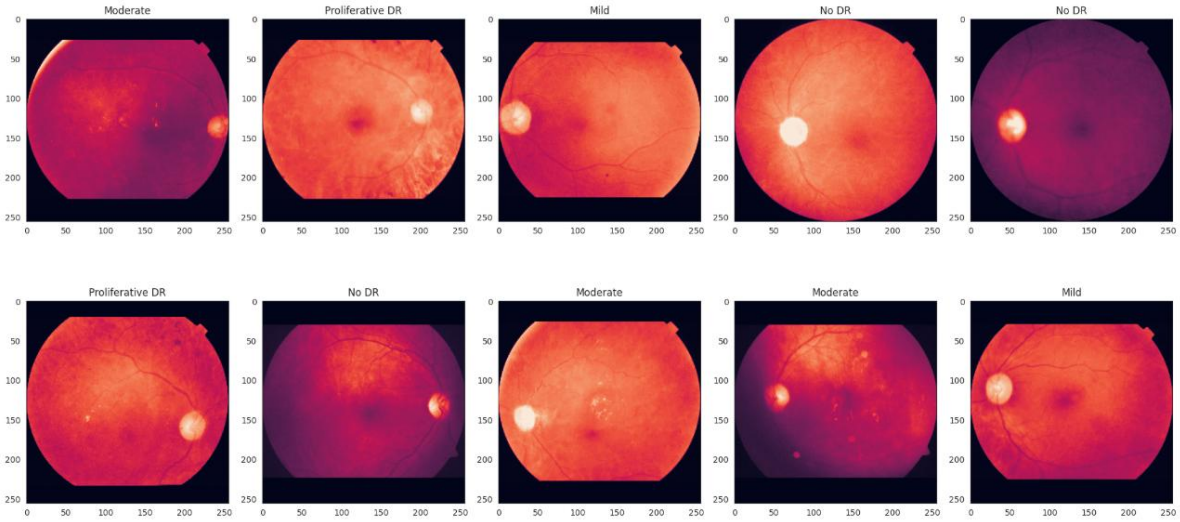

Figure 3.7 Channel Red

Apply CLAHE for Channel Red:

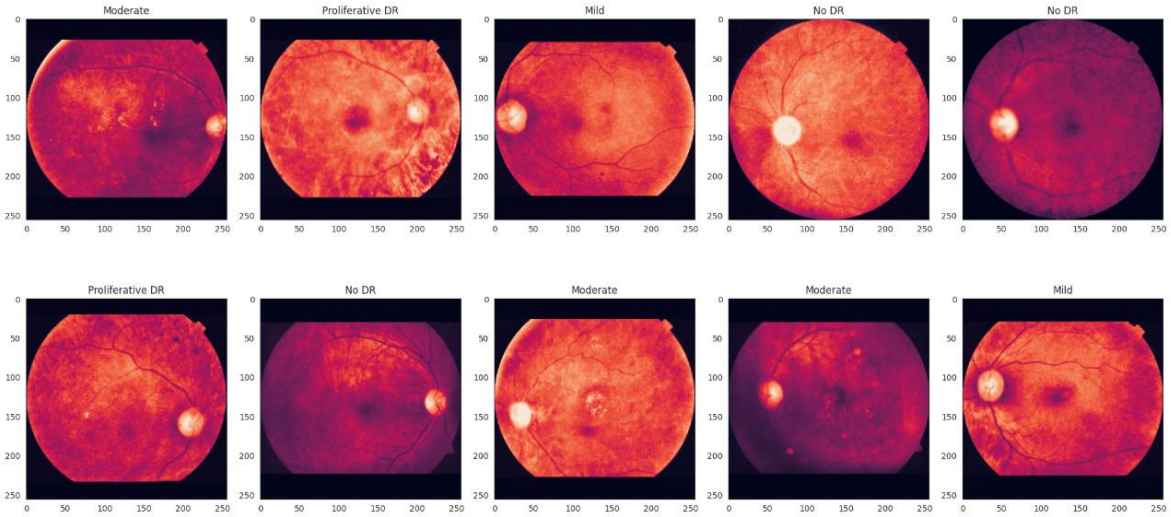

Figure 3.8 Apply CLAHE for Channel Red

Apply CLAHE for Channel Green

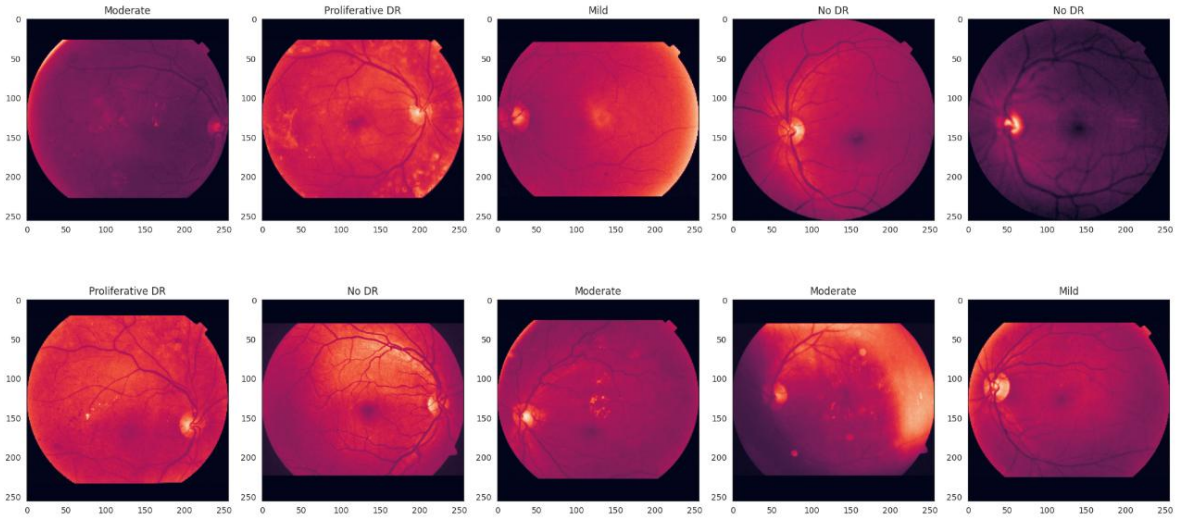

Figure 3.9 Channel Green

Apply CLAHE for Channel Green:

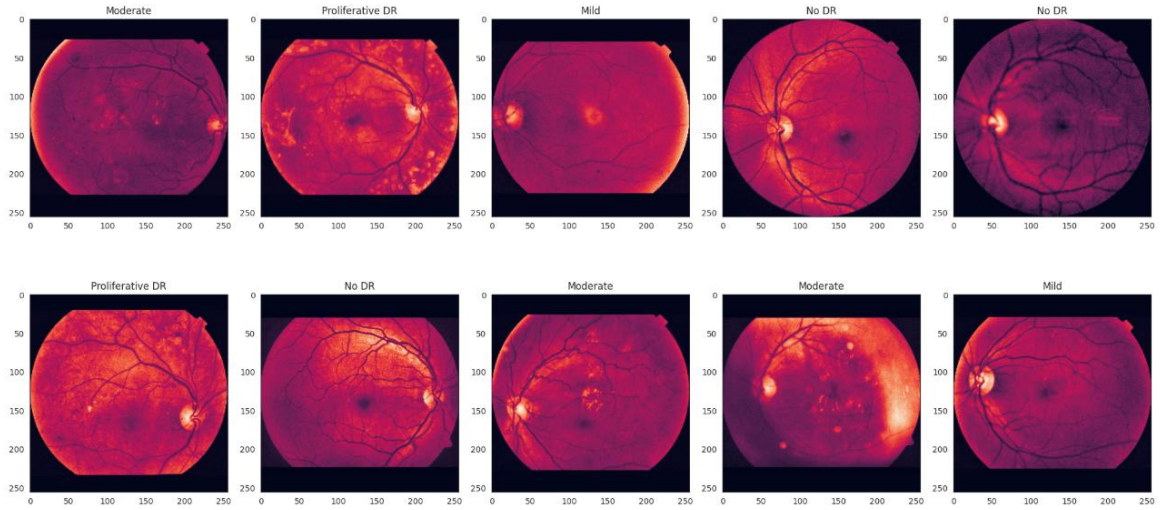

Figure 3.10 Apply CLAHE for Channel Green

Apply CLAHE for Channel Blue

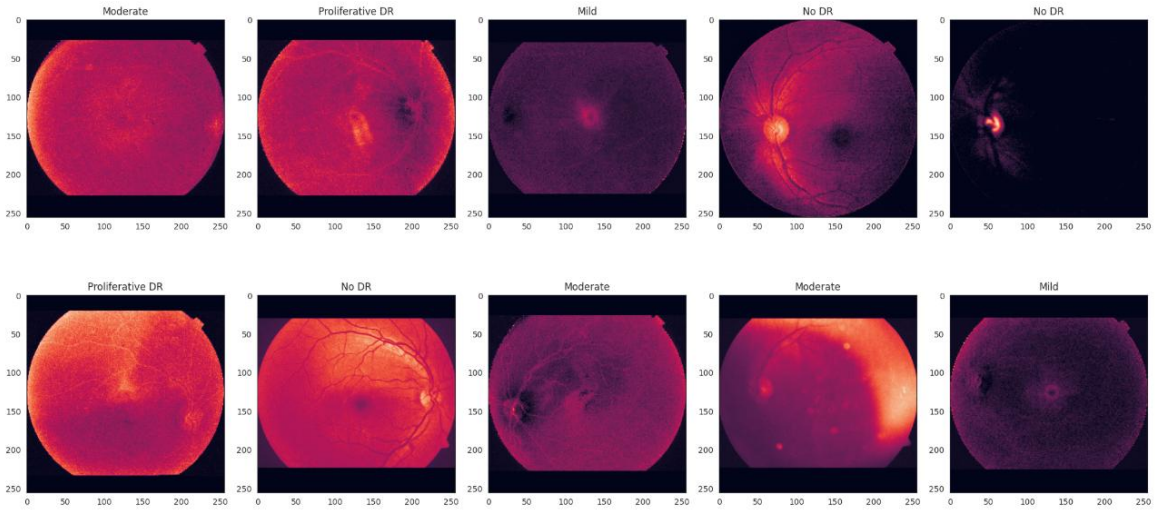

Figure 3.11 Channel Blue

Apply CLAHE for Channel Blue:

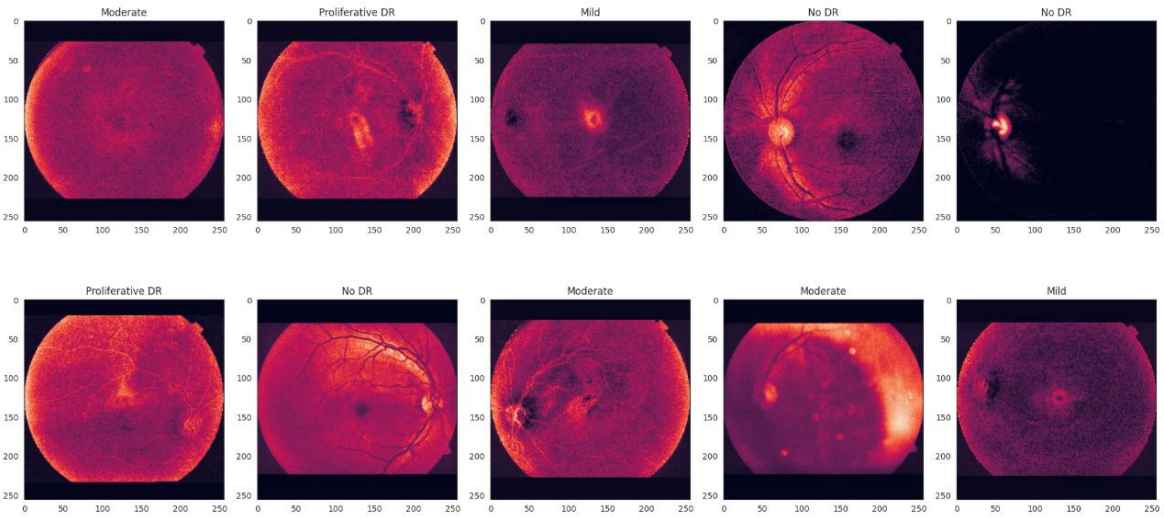

Figure 3.12 Apply CLAHE for Channel Blue

3.3.3 Top/Black hat

This is a morphological transformation on binary images. It is used to extract small details and elements from the retinal images in the dataset. With Top-hat we can enhance the brightness of bright objects of interest in a dark background, which helps to enhance very small details and remove them with Top-hat.

In retinal images, Top-hat helps enhance bright areas of interest such as blood stains, specks, etc., thereby highlighting the objects we want to be interested in to proceed with feature extraction.

$$Top - hat(f) = (f) - (f \circ \mathrm{g})$$

Furthermore, the Black-hat transform enhances dark objects of interest against a bright background. Here, the Black-hat operation is applied to highlight dark features such as hemorrhages and microaneurysms, which are important indicators in the diagnosis of diabetic retinopathy. Specifically, the Black-hat transform is employed with three different kernels (kernel_horizontal, kernel_vertical, and a circular kernel) to extract dark features in multiple orientations. The results from these three operations are then combined by taking the pixel-wise maximum, stored in the image_blackhat variable, aiming to comprehensively capture these crucial dark features.

$$Bot - hat(f) = (f \bullet g) - (f)$$

Code Kernel and Top hat, Black hat:

```
kernel_horizontal = np.array([[1, 1, 1, 1, 1]], dtype=np.uint8)

kernel_vertical = np.array([[1],

                            [1],

                            [1],

                            [1],

                            [1]], dtype=np.uint8)

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (9, 9))


#Top/Black hat for Channel Red

blackhat_h = cv2.morphologyEx(red, cv2.MORPH_BLACKHAT,
kernel_horizontal)

blackhat_v = cv2.morphologyEx(red, cv2.MORPH_BLACKHAT, kernel_vertical)

tophat = cv2.morphologyEx(red, cv2.MORPH_TOPHAT, kernel)

blackhat = cv2.morphologyEx(red, cv2.MORPH_BLACKHAT, kernel)

image_tophat = cv2.add(red, tophat)
```

```
image_blackhat = np.maximum(np.maximum(blackhat_h, blackhat_v), blackhat)

image_red = cv2.subtract(image_tophat, image_blackhat)


#Top/Black hat for Channel Green

blackhat_h = cv2.morphologyEx(green, cv2.MORPH_BLACKHAT,
kernel_horizontal)

blackhat_v = cv2.morphologyEx(green, cv2.MORPH_BLACKHAT,
kernel_vertical)

tophat = cv2.morphologyEx(green, cv2.MORPH_TOPHAT, kernel)

blackhat = cv2.morphologyEx(green, cv2.MORPH_BLACKHAT, kernel)

image_tophat = cv2.add(green, tophat)

image_blackhat = np.maximum(np.maximum(blackhat_h, blackhat_v), blackhat)

image_green = cv2.subtract(image_tophat, image_blackhat)


image = cv2.merge([image_red, image_green, blue])
```

Top-hat and Black-hat transforms are used to highlight lesions in retinal images.

- kernel_horizontal, kernel_vertical, kernel: These are the structuring elements used.
- cv2.morphologyEx(image, cv2.MORPH_TOPHAT, kernel): Performs the Top-hat transform.
- cv2.morphologyEx(image, cv2.MORPH_BLACKHAT, kernel): Performs the Black-hat transform.
- The Top-hat and Black-hat transforms are applied separately to the red and green color channels after CLAHE has been applied.

I used Top and Black hat to detect:

- Bright Lesions: Lesions like exudates are often brighter than the background. Top-hat helps extract these bright regions.
- Dark Lesions: Lesions like microaneurysms and hemorrhages are often darker than the background. Black-hat helps extract these dark regions.

The results of Top-hat and Black-hat are combined to highlight both bright and dark lesions:

- image_tophat = cv2.add(red, tophat): Adds the Top-hat image to the original image to highlight bright lesions.
- image_blackhat = np.maximum(np.maximum(blackhat_h, blackhat_v), blackhat): Takes the pixel-wise maximum between Black-hat images with horizontal, vertical, and circular kernels to capture dark features in different orientations.
- image_red = cv2.subtract(image_tophat, image_blackhat): Subtracts the Black-hat image from the Top-hat image to remove noise and retain the desired features.

So, Top-hat and Black-hat are useful tools for extracting bright and dark features in images DR, helping to highlight lesions in retinal images.

Apply Top-hat and Black-hat to the Red channel:

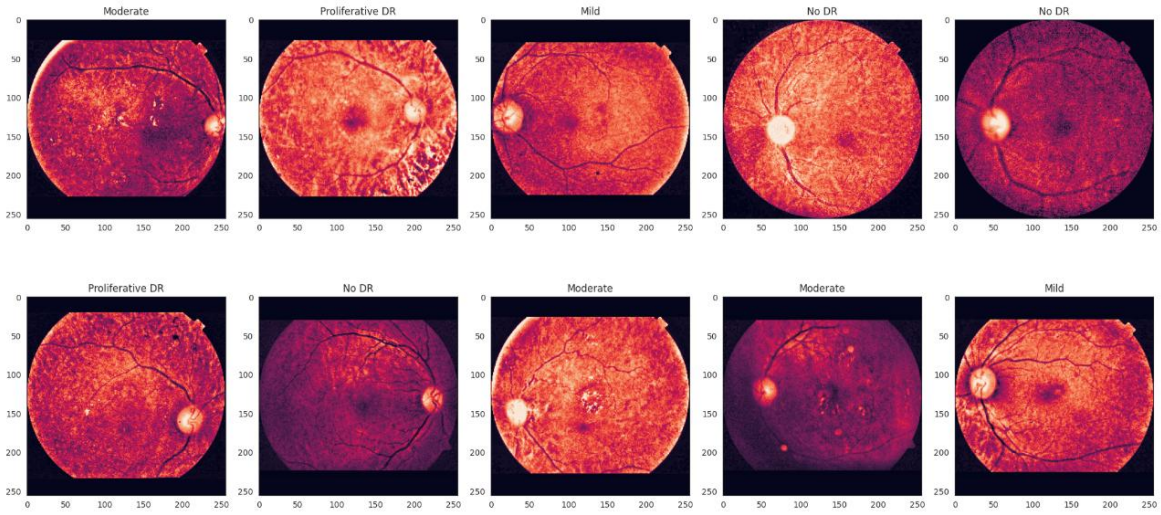

Figure 3.13 Apply Top-hat and Black-hat to the Red channel

Apply Top-hat and Black-hat to the Green channel:

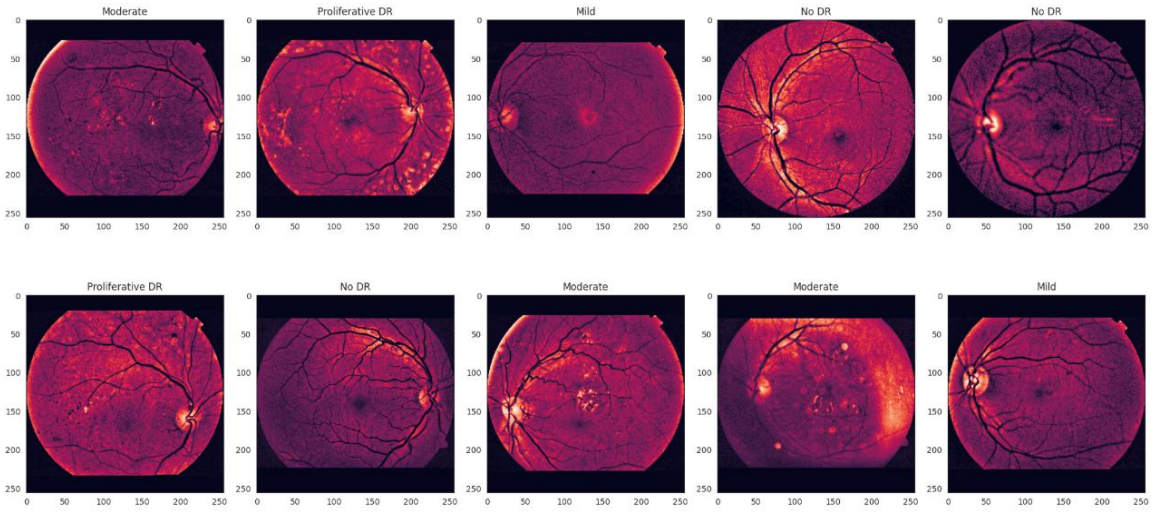

Figure 3.14 Apply Top-hat and Black-hat to the Green channel

Meme:

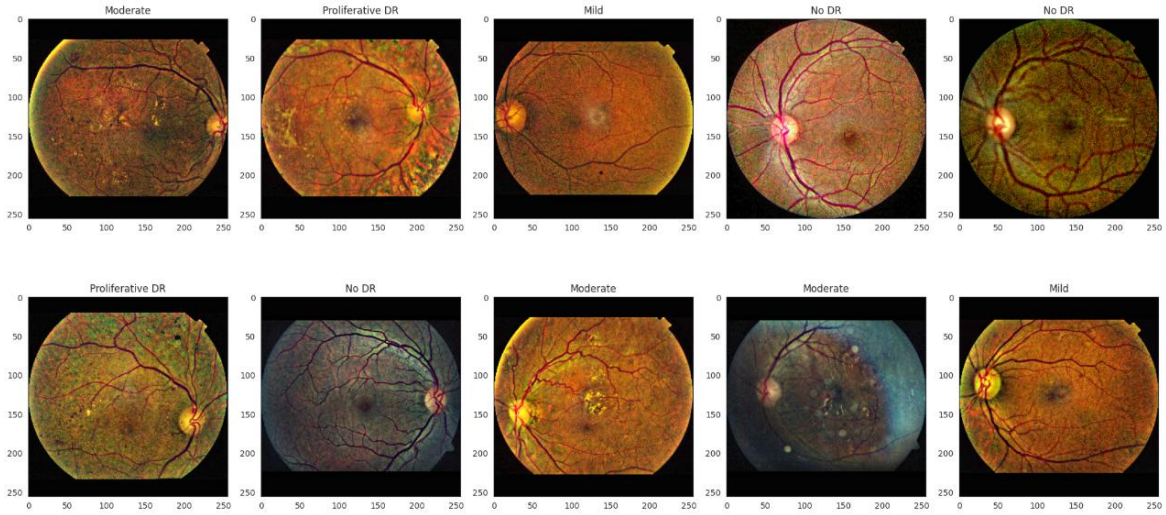

Figure 3.15 Apply Top-hat and Black-hat to the Color channel

3.3.4 Data augmentation

Data augmentation is used to increase the size of the training dataset, improve the generalization ability and accuracy of the model, and help the model recognize retinal lesions more effectively under various conditions.

Split: Data augmentation is only applied to the training dataset. Therefore, the data was split into training (80%) and validation (20%) sets before being passed through the augmentation process.

Purpose of Data Augmentation:

- Increase Dataset Size: When data is small, data augmentation helps increase the number of training samples, allowing the model to learn better.
- Reduce Overfitting: By training on multiple variations of the same image, the model learns more general features instead of memorizing the specific details of the training set. This improves the model's ability to perform well on unseen data.
- Improve Accuracy: When the model is trained on a more diverse dataset, it becomes better at recognizing objects under various conditions (e.g., different rotations, flips, brightness changes).
- Help the Model Learn Invariant Features: For example, if you apply image rotation, the model learns to recognize the object regardless of its orientation.

```
train_trasforms= T.Compose([
    T.RandomHorizontalFlip(),
    T.RandomVerticalFlip(),
    T.RandomRotation(degrees=60),
    T.ColorJitter(brightness=0.10),
    T.ConvertImageDtype(torch.float32),
    T.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])
```

Transformations Used in Data Augmentation for Images:

- Randomly flips the image horizontally with a default probability of 0.5.
- Randomly flips the image vertically with a default probability of 0.5.
- Randomly rotates the image by an angle between -60 and +60 degrees.
- Randomly changes the brightness of the image within a range of ±10%.

So:

- RandomHorizontalFlip, RandomVerticalFlip: Flipping images helps the model learn to recognize lesions regardless of their orientation in the retinal image.

- RandomRotation: Rotating images helps the model learn to recognize lesions at different angles.
- ColorJitter: Changing the brightness helps the model become less sensitive to variations in lighting conditions during image capture.

3.4 Model

3.4.1 A Swin Transformer V2 image classification model

```
MODEL_SWINV2_SMALL = "swinv2_small_window16_256"

MODEL_SWINV2_SMALL_SAVE=  "/kaggle/working/swinv2.Csv"

CHECKPOINT_MODEL_SWINV2_SMALL_DIR =
"/kaggle/working/model_swinv2/"

os.makedirs(CHECKPOINT_MODEL_SWINV2_SMALL_DIR, exist_ok=True)

set_debug_apis(False)
```

A Swin Transformer V2 image classification model. Pretrained on ImageNet-1k.

Model Details:

- Model Type: Image classification / feature backbone
- Model Stats:
    - Params (M): 49.7
    - GMACs: 12.8
    - Activations (M): 66.3
    - Image size: 256 x 256

3.4.2 A FastViT image classification model

```
MODEL_FastVit = "fastvit_s12"

MODEL_FastVit_SAVE=  "/kaggle/working/fastvit_s12.Csv"

CHECKPOINT_MODEL_FastVit_DIR = "/kaggle/working/model_fastvit/"

os.makedirs(CHECKPOINT_MODEL_FastVit_DIR, exist_ok=True)

set_debug_apis(False)
```

A FastViT image classification model. Trained on ImageNet-1k.

Model Details:

- Model Type: Image classification / feature backbone
- Model Stats:
    - Params (M): 9.5
    - GMACs: 1.8
    - Activations (M): 13.7
    - Image size: 256 x 256

3.5 Evaluation

3.5.1 A Swin Transformer V2 image classification model Evaluation

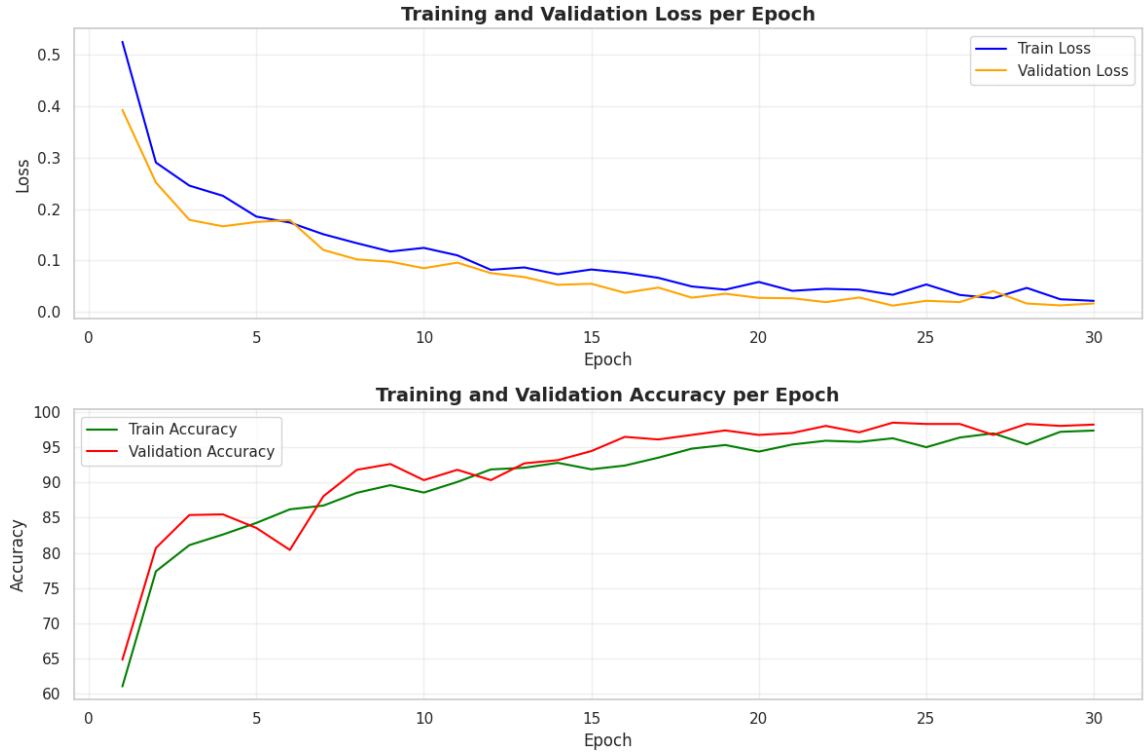

Figure 3.16 Training and Validation Loss and Accuracy per Epoch

The image depicts two graphs illustrating the performance of a machine learning model over 30 training epochs:

Training and Validation Loss per Epoch

- Both training loss and validation loss decrease over the epochs, indicating that the model is learning and improving.

- Training loss is consistently lower than validation loss, which is typical behavior.
- The gap between training and validation loss starts to narrow after around the 15th epoch, suggesting the model is starting to overfit less.
- From epoch 25 to 30, the validation loss fluctuates slightly but shows a slight upward trend. This might indicate the beginning of overfitting.

Training and Validation Accuracy per Epoch

- Both training accuracy and validation accuracy increase over the epochs, signifying that the model is learning and improving its predictions.
- Training accuracy is consistently higher than validation accuracy.
- Both accuracies reach high levels (above 95%) after around 20 epochs.
- Validation accuracy shows some minor fluctuations between epochs 25 and 30 but remains high overall.

Epoch 29 as a Good Stopping Point: Based on the graphs and these final metrics, epoch 29 appears to be a good point to stop training. The model has achieved high accuracy on both training and validation sets, and further training might lead to overfitting, as hinted by the slight upward trend of the validation loss near the end of the first graph.

```
Train Accuracy: 98.35%
Validation Accuracy: 98.16%
```

Figure 3.17 Accuracy for Model Swin Transformer V2

High Performance: The model demonstrates excellent performance, with both training and validation accuracies exceeding 98%. This suggests the model has learned the underlying patterns in the data effectively.

Minimal Overfitting (Likely): The small difference (0.19%) between training and validation accuracy suggests that overfitting is minimal at epoch 29. This is consistent with the validation accuracy plateauing in the second graph. Also the validation loss is still relatively low in the graph 3.16.

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	336
1	0.96	0.99	0.97	204
2	0.99	0.93	0.96	212
3	0.93	0.99	0.96	94
4	0.99	1.00	0.99	242
accuracy			0.98	1088
macro avg	0.97	0.98	0.98	1088
weighted avg	0.98	0.98	0.98	1088

Figure 3.18 Classification Report for Model Swin Transformer V2

Excellent Performance: This classification report indicates that the model is performing exceptionally well. The precision, recall, and F1-score are very high (close to 1) for almost all classes.

Class Imbalance: The "support" column shows some class imbalance. Class 0 has the most instances (336), while class 3 has the fewest (94).

Class 3 Performance: Class 3 has slightly lower precision (0.93) compared to other classes, meaning that there were a few more false positives for this class. However, its recall is high (0.99).

Class 2 Performance: Class 2 has a slightly lower recall (0.93), but its precision is good

High Overall Accuracy: The overall accuracy of 0.98 is very impressive, confirming the model's strong predictive capability

Averages: The macro and weighted averages are both 0.98 for all three metrics (precision, recall, and F1-score), indicating consistently high performance across all classes.

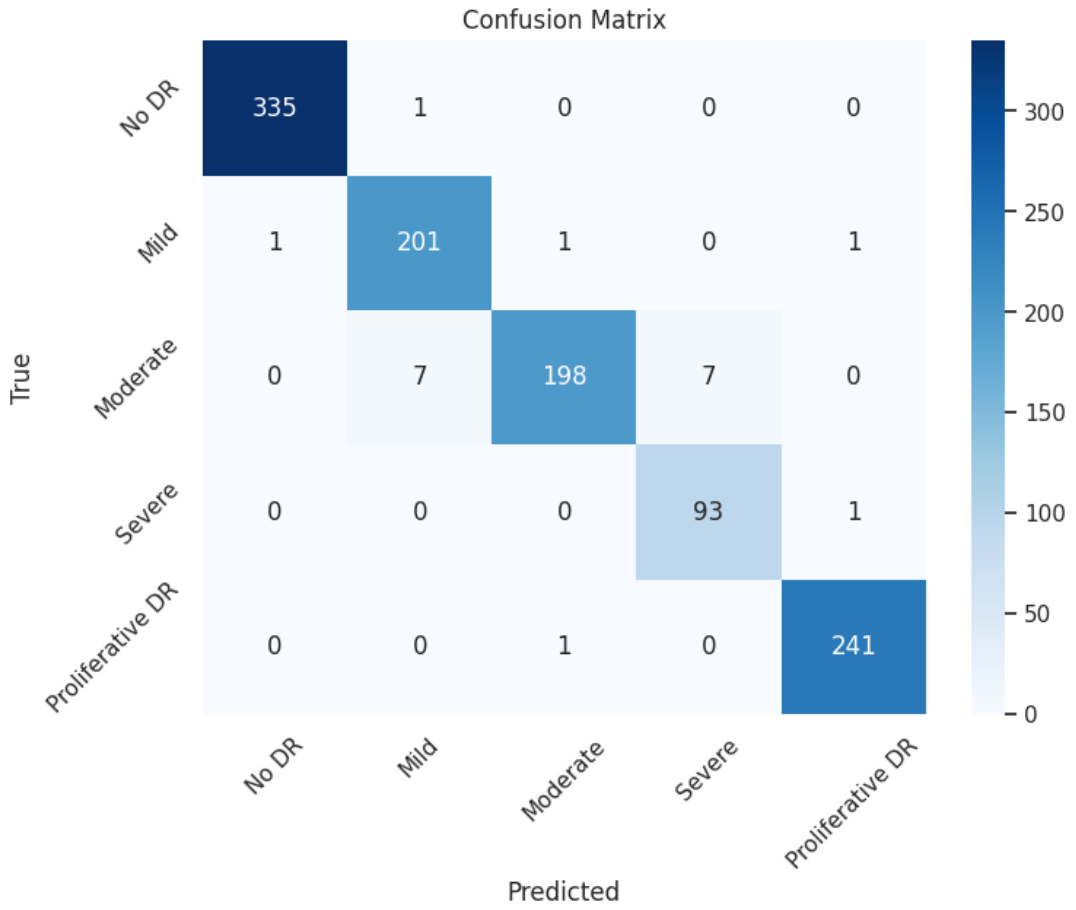

Figure 3.19 Model Swin Transformer V2 Confusion Matrix

Analysis of this Confusion Matrix:

High Accuracy: The majority of the instances lie along the diagonal, indicating high accuracy. The model is doing a good job of correctly classifying most instances.

Specific Observations:

- No DR: Out of 336 actual No DR cases, 335 were correctly classified, with only 1 being misclassified as Mild.
- Mild: Out of 204 actual Mild cases, 201 were correctly classified. The misclassifications were 1 as No DR, 1 as Moderate, and 1 as Proliferative DR.
- Moderate: Out of 212 actual Moderate cases, 198 were correctly classified. The misclassifications were 7 as Mild, 7 as Severe. This is where most errors are.

- Severe: Out of 94 actual Severe cases, 93 were correctly classified, with only 1 misclassified as Proliferative DR.
- Proliferative DR: Out of 242 actual Proliferative DR cases, 241 were correctly classified, with only 1 misclassified as Moderate.

Class	Error Rate
No DR	0.003
Mild	0.0147
Moderate	0.066
Severe	0.0106
Proliferative DR	0.0041

Table 3.4 Model Swin Transformer V2 Error Rates

3.5.2 A FastViT image classification model Evaluation

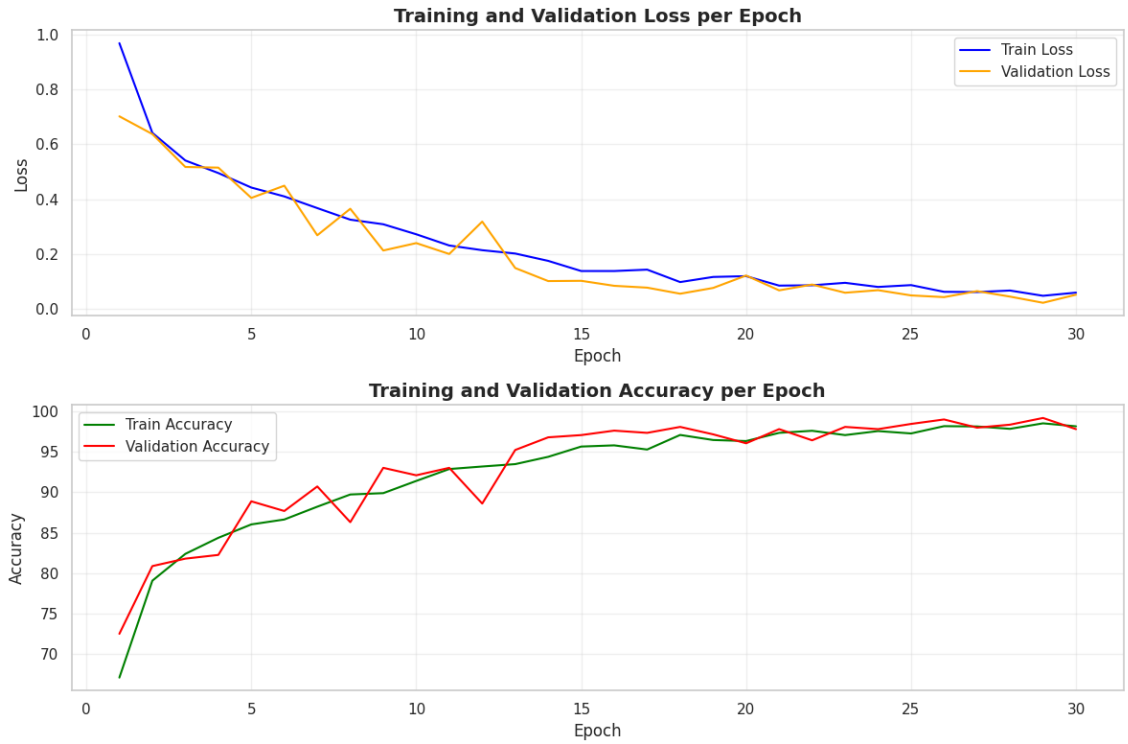

Figure 3.20 Training and Validation Loss and Accuracy per Epoch

The image depicts two graphs illustrating the performance of a machine learning model over 30 training epochs:

Training and Validation Loss per Epoch

- Rapid Initial Decrease in Loss: Both training and validation loss decrease rapidly in the first few epochs, indicating the model is quickly learning from the training data.
- Higher Initial Loss: Compared to the first model, this model starts with a much higher initial training loss (around 1.0) and validation loss (around 0.7).
- Validation Loss Fluctuations: The validation loss shows more fluctuations than in the first model, especially between epochs 5 and 15. This suggests the model might be struggling a bit more to generalize to unseen data during this period.
- Convergence: Both training and validation loss converge to low values (close to 0) by the end of the training process, indicating that the model has learned to fit the data well.
- Potential Overfitting: While both losses are low, the training loss is consistently below the validation loss, and there are a few points where the gap widens slightly.
- The validation loss also shows a slight upward trend at the very end of the graph. This could be a mild sign of overfitting, but it's not very pronounced.

Training and Validation Accuracy per Epoch

- Rapid Initial Increase in Accuracy: Both training and validation accuracy increase rapidly in the first few epochs, mirroring the rapid decrease in loss.
- High Accuracy: The model achieves high accuracy on both the training and validation sets, with both reaching above 95% towards the end of training.
- Validation Accuracy Fluctuations: The validation accuracy (red line) shows some fluctuations, particularly between epochs 5 and 15, which corresponds to the fluctuations in validation loss in the top graph.
- Slightly Lower Validation Accuracy: The validation accuracy is generally a bit lower than the training accuracy, which is expected.

- Plateau: Both training and validation accuracies plateau towards the end of training, suggesting that further training might not yield significant improvements.

Epoch 24 as a Potential Stopping Point: Given the high and very similar training and validation accuracies, and the plateauing observed in the graphs, epoch 24 could be considered a good stopping point for training Model.

```
Train Accuracy: 99.03%
Validation Accuracy: 98.99%
```

Figure 3.21 Accuracy for Model FastViT

Extremely High Accuracy: These accuracy values are exceptionally high, indicating that Model performs extremely well at epoch 24.

Minimal Overfitting: The difference between training and validation accuracy is very small (0.04%), suggesting minimal overfitting at this epoch. This is a very positive sign.

Consistent with Graphs: These values align with the trends observed in the training/validation graphs. By epoch 24, both training and validation accuracy had plateaued at high levels, and the gap between them was small.

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	380
1	0.99	0.99	0.99	219
2	0.99	0.97	0.98	188
3	0.94	1.00	0.97	63
4	0.99	1.00	0.99	238
accuracy			0.99	1088
macro avg	0.98	0.99	0.99	1088
weighted avg	0.99	0.99	0.99	1088

Figure 3.22 Classification Report for Model FastViT

Excellent Performance: Model FastViT demonstrates exceptional performance across all classes, with very high precision, recall, and F1-scores.

Class 0: Perfect scores (1.00) for precision, recall, and F1-score. The model perfectly identifies this class.

Class 1: Very high scores (0.99) for all metrics.

Class 2: Slightly lower recall (0.97) compared to other classes, but still very good. This aligns with our previous observations that Moderate is the most challenging class. Precision is high at 0.99.

Class 3: High precision (0.94) and perfect recall (1.00).

Class 4: Near-perfect scores (0.99 and 1.00).

Overall Metrics: Accuracy, macro average, and weighted average are all 0.99, indicating near-perfect overall performance.

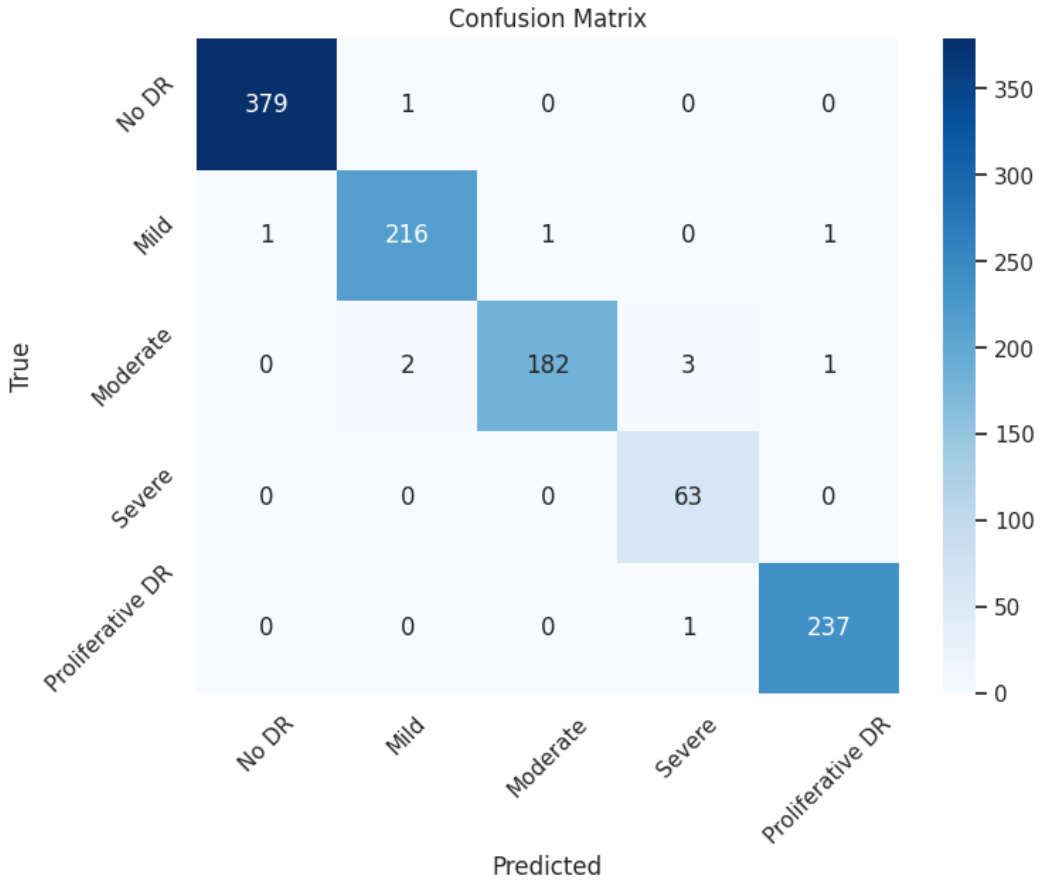

Figure 3.23 Model FastViT Confusion Matrix

Analysis of this Confusion Matrix:

High Accuracy: The vast majority of instances are on the diagonal, confirming the model's high accuracy.

Specific Observations:

- No DR: 379 correctly classified, 1 misclassified as Mild.
- Mild: 216 correctly classified, 1 misclassified as No DR, 1 as Moderate, and 1 as Proliferative DR.
- Moderate: 182 correctly classified, 2 misclassified as Mild, 3 as Severe, and 1 as Proliferative DR. This is where most of the errors are, as we've seen before.
- Severe: 63 correctly classified, with no misclassifications.
- Proliferative DR: 237 correctly classified, 1 misclassified as Severe.

Class	Error Rate
No DR	0.0026
Mild	0.0137
Moderate	0.0319
Severe	0.0
Proliferative DR	0.0042

Table 3.5 Model FastViT Error Rates

3.6 Results and Discussion

3.6.1 Individual Model Evaluations

Model Swin Transformer V2 is a good model for the task of classifying diabetic retinopathy severity. It achieved high accuracy, showed minimal overfitting, and learned effectively. However, the model could be further improved to better classify the Moderate class.

Model FastViT is an excellent model for the task of classifying diabetic retinopathy severity. It achieved very high accuracy, showed almost no overfitting, and learned very effectively. In particular, Model FastViT overcame the weakness of Model

Swin Transformer V2 in classifying the Moderate class, achieving significant improvement in this area.

3.6.2 Comparison Swin Transformer V2 and FastViT Model

Comparison Accuracy

Metric	Swin Transformer V2	FastViT
Train Accuracy	98.35%	99.03%
Validation Accuracy	98.16%	98.99%

Table 3.6 Comparison Accuracy

Evaluation:

Model FastViT is better: Model FastViT has higher train accuracy and validation accuracy compared to Model Swin Transformer V2.

Less overfitting: Model FastViT shows significantly less evidence of overfitting compared to Model Swin Transformer V2.

Comparison Classification Report

Metric	Swin Transformer V2	FastViT
Accuracy	0.98	0.99
Train Accuracy	98.35% (Epoch 29)	99.03% (Epoch 24)
Validation Accuracy	98.16% (Epoch 29)	98.99% (Epoch 24)
Precision (Weighted Avg)	0.98	0.99
Recall (Weighted Avg)	0.98	0.99
F1-score (Weighted Avg)	0.98	0.99
Macro Avg F1-score	0.98	0.99
Recall Moderate (Class 2)	0.93	0.97

Precision Severe (Class 3)	0.93	0.94
Error Rate Moderate	0.066	0.0319
Error Rate Severe	0.0106	0.0
Overfitting	Slight (difference of ~0.19% between train and validation accuracy)	Almost None (difference of ~0.04%)
Training Loss	Decreased gradually, stable	Decreased rapidly, some minor fluctuations
Validation Loss	Decreased gradually, slight upward trend near the final epoch	Decreased gradually, minor fluctuations, tended to plateau
Training Accuracy	Increased gradually, stable	Increased rapidly, some minor fluctuations
Validation Accuracy	Increased gradually, stable	Increased rapidly, minor fluctuations, tended to plateau
Techniques Likely Used	Potentially didn't use or didn't optimally use Focal Loss and Class Weighting	Effectively used Focal Loss and Class Weighting

Table 3.7 Comparison Classification Report

Conclusion:

- Model FastViT outperforms Model Swin Transformer V2 in all aspects.
- Model FastViT achieved higher accuracy, showed almost no overfitting, and performed significantly better in classifying the Moderate class (the most challenging class).
- The use of Focal Loss and Class Weighting is likely the key factor that enabled Model FastViT to achieve better results.

Comparison to Model Error Rates:

Here's a comparison table:

Class	Swin Transformer V2	FastViT
No DR	0.003	0.0026
Mild	0.0147	0.0137
Moderate	0.066	0.0319
Severe	0.0106	0.0
Proliferative DR	0.0041	0.0042

Table 3.8 Comparison to Model Error Rates

3.6.3 Results

Model FastViT outperforms Model Swin Transformer V2 in all aspects:

- Higher accuracy.
- Less overfitting.
- Significantly better classification of the Moderate class.
- Lower error rates across most classes.

CHAPPTER IV: CONCLUSIONS AND RECOMMENDATIONS

4.1 Revised Conclusions

- **Superior Performance of FastViT**: FastViT demonstrated superior performance compared to the Swin Transformer in classifying the severity of diabetic retinopathy from fundus images. This is evidenced by its higher accuracy (99.03% train, 98.99% validation), better handling of the challenging "Moderate" class, and minimal overfitting.
- **Effectiveness of Image Preprocessing**: The novel image preprocessing approach, combining CLAHE with Top-hat and Black-hat morphological operations, likely contributed significantly to the high performance of both models. This approach effectively highlighted pathological features, making it easier for the models to learn discriminative features.
- **Effectiveness of Focal Loss and Class Weighting**: The application of Focal Loss and class weighting in the FastViT model successfully addressed the issue of class imbalance and improved the model's ability to learn from and classify minority classes, especially the critical Moderate severity level.
- **FastViT's Suitability for Medical Image Analysis**: The results indicate that the FastViT architecture is highly suitable for medical image analysis tasks, particularly for diabetic retinopathy severity classification.
- **Swin Transformer as a Strong Alternative**: Although outperformed by FastViT, the Swin Transformer still demonstrated good performance. Its lower accuracy could potentially be improved with further optimization or different training strategies.

4.2 Revised Recommendations

- **Focus on FastViT for Deployment**: Given its superior performance, FastViT should be the primary focus for further development and potential deployment in a clinical setting.
- **Further Optimization of FastViT**: Explore more extensive hyperparameter optimization for FastViT to potentially further improve its performance. Areas to focus on could include learning rate scheduling, optimizer choice, and data augmentation strategies.

- **Investigate FastViT Variants**: Research and experiment with different variants of the FastViT architecture to potentially find even more performant configurations.
- **Further Validation**: While the results are promising, further validation on larger and more diverse datasets is recommended to confirm the generalizability of the FastViT model.
- **Clinical Deployment Considerations**: Before clinical deployment, address issues related to model interpretability and explainability. Understanding the model's decision-making process is crucial for building trust among clinicians.
- **Experiment with Different Preprocessing**: Experiment with variations of the preprocessing pipeline to determine the optimal combination of techniques for different datasets and imaging modalities.
- **Re-evaluate Swin Transformer**: Investigate the reasons behind the relatively lower performance of the Swin Transformer. It might require different hyperparameter settings or training strategies to reach its full potential.
- **Speed and Efficiency**: Since FastVit is designed to be faster, quantify the speed advantage that FastVit has over the Swin Transformer in a production setting.

REFERENCES

[1] Hugging Face, “Cơ chế hoạt động của Transformer?”, https://huggingface.co/learn/nlp-course/vi/chapter1/4?fw=pt.

[2] Hugging Face, “swinv2_small_window16_256.ms_in1k”, https://huggingface.co/timm/swinv2_small_window16_256.ms_in1k.

[3] Hugging Face, “swinv2-small-patch4-window16-256”, https://huggingface.co/microsoft/swinv2-small-patch4-window16-256.

[4] Hugging Face, “fastvit_s12.apple_dist_in1k”, https://huggingface.co/timm/fastvit_s12.apple_dist_in1k.

[5] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, Anurag Ranjan (2023), “FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization”, Hugging Face.

[6] Le Hoang Hiep (2023), “Nghiên cứu ứng dụng học sâu hỗ trợ phát hiện bệnh võng mạc đái tháo đường từ ảnh võng mạc”, Luận văn thạc sĩ máy tính, Học viện khoa học và công nghệ, Viện hàn lâm khoa học và công nghệ Việt Nam, Hà nội.

[7] To Duc Thang (2021), “Imbalanced Multiclass Datasets”, https://viblo.asia/p/imbalanced-multiclass-datasets-Do754dmQ5M6.

[8] Allan Kouidri (2023), “Understanding ResNet: A Milestone in Deep Learning and Image Recognition”, https://www.ikomia.ai/blog/mastering-resnet-deep-learning-image-recognition#how-resnet-works.

[9] A.Dihin, Rasha & Alshemmary, Ebtesam & Al-Jawher, Waleed (2023), “Diabetic Retinopathy Classification Using Swin Transformer with Multi Wavelet”, Journal of Kufa for Mathematics and Computer, 10(2), pp.167-172.

[10] A.Dihin, Rasha & Alshemmary, Ebtesam & Al-Jawher, Waleed (2023), “Automated Binary Classification of Diabetic Retinopathy by SWIN Transformer”, Journal of Al-Qadisiyah for Computer Science and Mathematics, 15(1).

[11] Geeksforgeeks (2023), “Top Hat and Black Hat Transform using Python-OpenCV”, https://www.geeksforgeeks.org/top-hat-and-black-hat-transform-using-python-opencv/.

[12] Hermawan, Hendar & Whardana, Adithya (2024), “Hemorrhage Segmentation on Retinal Images for Early Detection of Diabetic Retinopathy”, JEECS (Journal of Electrical Engineering and Computer Sciences), 9 (2), pp.117-128.

[13] Hou, Yanli (2014), “Automatic Segmentation of Retinal Blood Vessels Based on Improved Multiscale Line Detection”, Journal of Computing Science and Engineering, 8(2), pp.119-128

[14] Li, Zhenwei & Han, Yanqi & Yang, Xiaoli (2023), “Multi-Fundus Diseases Classification Using Retinal Optical Coherence Tomography Images with Swin Transformer V2”, Journal of Imaging, 9(10), pp.203.

[15] Vasu, Pavan & Gabriel, James & Zhu, Jeff & Tuzel, Oncel & Ranjan, Anurag (2023), “FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization”.