SWIN Transformer VS VIT Report

1. Introduction

In recent years, transformer-based architectures have revolutionized the field of computer vision, significantly advancing the performance of image classification tasks. Originally developed for natural language processing, transformers have demonstrated their exceptional capability to capture long-range dependencies and contextual information when adapted to vision problems. Two prominent architectures in this domain are the Vision Transformer (ViT) and the Swin Transformer.

The Vision Transformer (ViT) applies the standard transformer model directly to image patches, treating an image as a sequence of flattened patches akin to tokens in text. This approach benefits from global self-attention mechanisms, allowing the model to capture relationships across the entire image. However, the ViT requires large-scale datasets and considerable computational resources to train effectively.

On the other hand, the Swin Transformer introduces a hierarchical structure and a novel shifted window attention mechanism. This design allows it to efficiently model local and global visual context with reduced computational complexity, making it well-suited for a variety of vision tasks including classification, detection, and segmentation.

This report aims to investigate and compare these two transformer-based architectures Swin Transformer and Vision Transformer on the CIFAR-10 image classification dataset. CIFAR-10 is a widely used benchmark dataset consisting of 60,000 low-resolution images across 10 object categories. Although these images are relatively small (32×32 pixels), resizing to 224×224 is necessary to conform to the input requirements of ViT and Swin models, which were originally trained on larger images.

The primary objectives of this study are:

- To preprocess CIFAR-10 images appropriately for transformer models.
- To adapt and fine-tune pre-trained ViT and Swin Transformer models for CIFAR-10 classification.
- To analyze and compare the training dynamics and performance metrics of both models.
- To interpret model decisions through Grad-CAM visualizations, highlighting important image regions contributing to classification.

By exploring these facets, this study seeks to deepen understanding of transformer-based vision models in handling small-scale datasets and to provide insights into their comparative strengths and limitations.

2. Methodology

This section outlines the step-by-step procedures followed in this study, including data preparation, model setup, and the training and evaluation processes. The goal was to adapt state-of-the-art transformer-based architectures, namely the Vision Transformer (ViT) and Swin Transformer, to the CIFAR-10 image classification task.

2.1 Data Preparation

The CIFAR-10 dataset comprises 60,000 color images, each of size 32×32 pixels, categorized into 10 distinct classes. Given that both ViT and Swin Transformer architectures were originally designed and pretrained on larger images (224×224 pixels), a resizing step was crucial to conform the dataset images to the input dimensions expected by these models.

Two separate transformation pipelines were defined for the training and testing sets:

- **Training Transformations:** The images were resized to 224×224 pixels. To increase the diversity of training data and reduce overfitting, random horizontal flipping was applied as a data augmentation technique. Subsequently, the images were converted to tensors and normalized using mean and standard deviation values of (0.5, 0.5, 0.5) across the RGB channels.

- **Testing Transformations:** For the validation/testing phase, images were also resized to 224×224 pixels and normalized identically, but no augmentation was applied to preserve the integrity of evaluation.

These preprocessing steps ensure the images are compatible with the input size and statistical distribution expected by the pretrained models, while also promoting robust learning during training.

The datasets were loaded using the torchvision.datasets.CIFAR10 class with the respective transformations. DataLoaders were created for both training and test sets with a batch size of 32, facilitating efficient batch-wise processing and shuffling during training.

2.2 Model Setup

Two pretrained transformer architectures were selected:

- **Swin Transformer (Base model):** Known for its hierarchical structure and shifted window self-attention, the Swin Transformer was loaded with pretrained weights. Since the original classification head is designed for 1000 classes (ImageNet), a custom classification head was implemented. This head includes an adaptive average pooling layer followed by a sequence of normalization, fully connected layers, ReLU activation, dropout, and final classification linear layer adjusted to output predictions for the 10 CIFAR-10 classes.

- **Vision Transformer (ViT Base):** The ViT model, pretrained on ImageNet, was also adapted by replacing its original classification head with a single linear layer outputting 10 class scores.

2.3 Training Process

The training pipeline was designed as follows:

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

- **Optimizers:** AdamW optimizers were chosen for both models, with learning rates carefully selected (5e-4 for Swin, 1e-3 for ViT) based on empirical tuning and common practice in fine-tuning transformers.

- **Training Loop:** For each epoch, the models were set to training mode. For every batch, the images and labels were transferred to the device GPU. Labels were reshaped as necessary to ensure compatibility with the loss function. A forward pass computed the model's outputs, followed by the calculation of loss and backpropagation to update weights. Running loss and accuracy were tracked for each epoch.
- **Evaluation:** After each training epoch, the models were evaluated on the test set in evaluation mode without gradient updates. Loss and accuracy metrics were computed to monitor generalization performance.

The entire process was repeated for 50 epochs, allowing sufficient time for convergence and performance assessment.

2.4 Visualization and Interpretability

To gain insights into the decision-making process of these transformer models, Grad-CAM (Gradient-weighted Class Activation Mapping) was employed. Grad-CAM helps visualize the regions of an input image that most strongly influence the model's predictions, thus enhancing interpretability.

Given the architectural differences between ViT and Swin models, separate Grad-CAM procedures were defined:

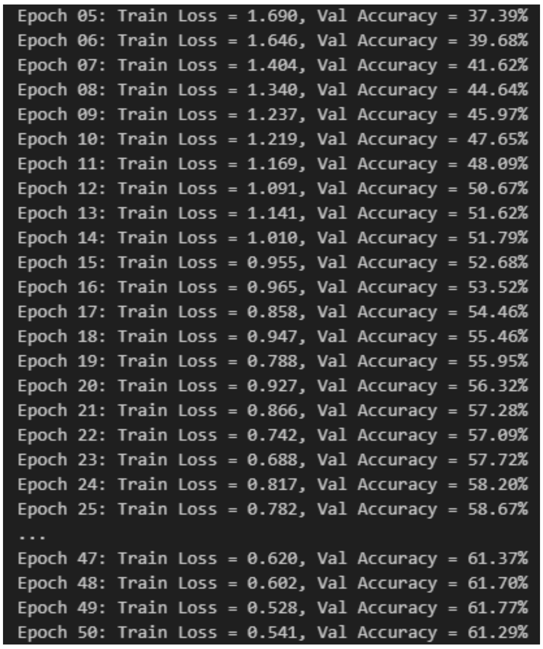
- In the Swin Transformer, Grad-CAM was applied to the attention blocks or normalization layers of the final transformer layers, and the second with custom reshaping of feature maps to spatial dimensions.
- In the Vision Transformer, since the input is processed as a sequence of patches, the CLS token was excluded during reshaping to visualize the spatial attention map properly.

These visualizations were generated for sample test images to compare how each model focuses on different image regions for classification.

2. Experiment Result

In this section we discussing about our experiment result of finetuning VIT and swin transformer base model using CIFAR10 dataset. The result conduct the analisys and the train loss vs validation accuracy and also the gradient capture with gradientCam

2.1. Result of finetuning VIT base model



```
Epoch 05: Train Loss = 1.690, Val Accuracy = 37.39%
Epoch 06: Train Loss = 1.646, Val Accuracy = 39.68%
Epoch 07: Train Loss = 1.404, Val Accuracy = 41.62%
Epoch 08: Train Loss = 1.340, Val Accuracy = 44.64%
Epoch 09: Train Loss = 1.237, Val Accuracy = 45.97%
Epoch 10: Train Loss = 1.219, Val Accuracy = 47.65%
Epoch 11: Train Loss = 1.169, Val Accuracy = 48.09%
Epoch 12: Train Loss = 1.091, Val Accuracy = 50.67%
Epoch 13: Train Loss = 1.141, Val Accuracy = 51.62%
Epoch 14: Train Loss = 1.010, Val Accuracy = 51.79%
Epoch 15: Train Loss = 0.955, Val Accuracy = 52.68%
Epoch 16: Train Loss = 0.965, Val Accuracy = 53.52%
Epoch 17: Train Loss = 0.858, Val Accuracy = 54.46%
Epoch 18: Train Loss = 0.947, Val Accuracy = 55.46%
Epoch 19: Train Loss = 0.788, Val Accuracy = 55.95%
Epoch 20: Train Loss = 0.927, Val Accuracy = 56.32%
Epoch 21: Train Loss = 0.866, Val Accuracy = 57.28%
Epoch 22: Train Loss = 0.742, Val Accuracy = 57.09%
Epoch 23: Train Loss = 0.688, Val Accuracy = 57.72%
Epoch 24: Train Loss = 0.817, Val Accuracy = 58.20%
Epoch 25: Train Loss = 0.782, Val Accuracy = 58.67%
...
Epoch 47: Train Loss = 0.620, Val Accuracy = 61.37%
Epoch 48: Train Loss = 0.602, Val Accuracy = 61.70%
Epoch 49: Train Loss = 0.528, Val Accuracy = 61.77%
Epoch 50: Train Loss = 0.541, Val Accuracy = 61.29%
```

Figure 1. Train loss and validation accuracy per epoch

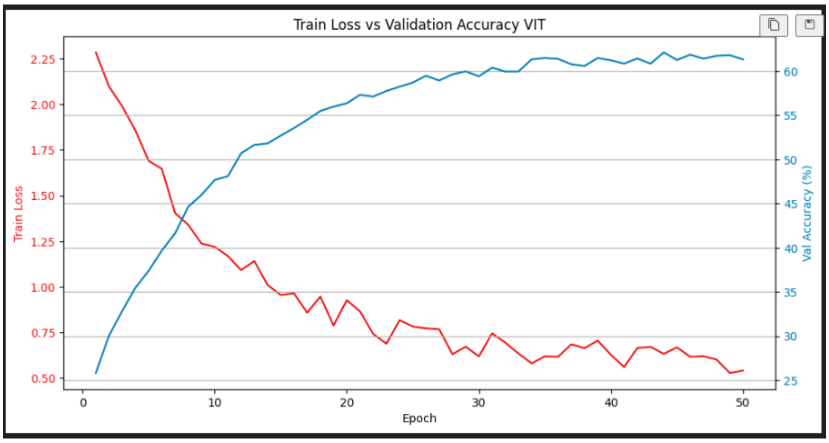


Figure 2. Train Loss and validation accuracy per epoch Graph

The training loss and validation per epoch of our experiment show by figure 1 and figure 2. The Vision Transformer (ViT) model achieved a training loss of 0.541, Conduct 50 epoch and the final validation accuracy of 61.29% on the CIFAR-10 dataset. This result indicates that the model successfully learned meaningful features from the data through fine-tuning. The consistent downward trend in training loss, along with the upward trend in validation accuracy, reflects effective convergence without significant signs of overfitting.

2.2. Gradient cam analisys of finetuning VIT base model

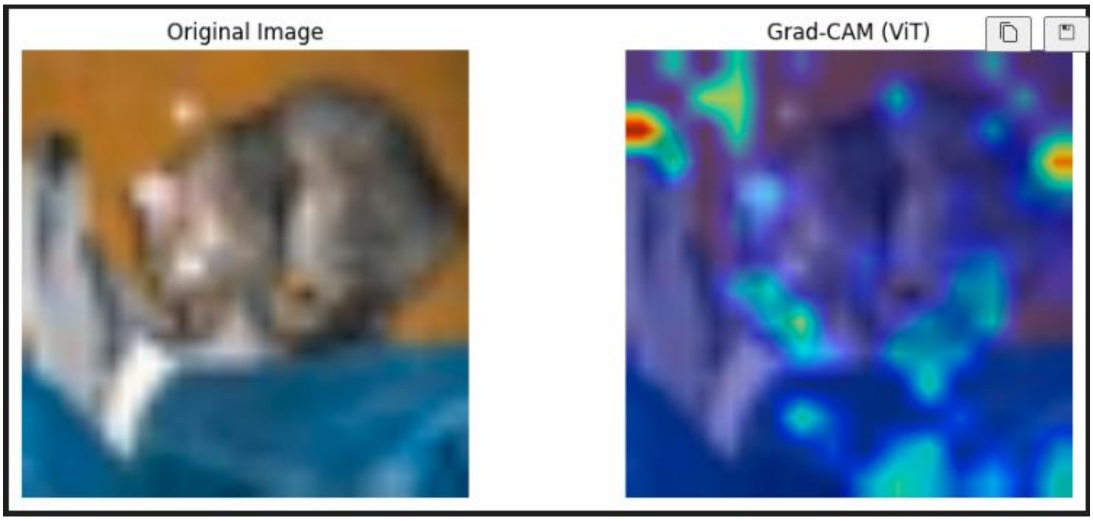


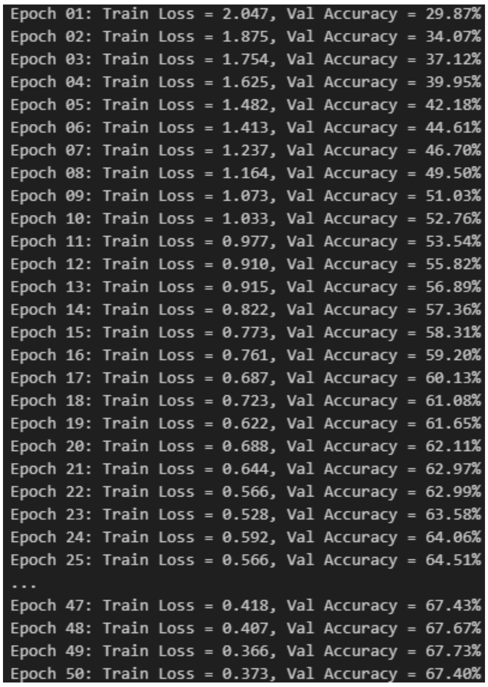
Figure 3 GradCam result of VIT base model

The visualization result using Grad-CAM on figure 3 demonstrates how the fine-tuned Vision Transformer (ViT) model attends to specific regions of the input image when making predictions on CIFAR-10. On the left is the original input image, and on the right is the Grad-CAM heatmap overlaid on the image. The highlighted regions in the Grad-CAM visualization indicate the areas the model focused on most to classify the object.

In this experiment, the Grad-CAM method was applied by extracting the last encoder block's normalization layer (vit_model.blocks[-1].norm1) as the target for visualization. The transformer output was reshaped using a custom reshape_transform function to convert the token sequence back into a spatial feature map suitable for visualization. The model successfully localized key parts of the image, as shown by the concentration of attention in semantically relevant areas, such as the head and upper body of the animal in the image.

This result confirms that the ViT model not only performed well quantitatively (as shown by validation accuracy improvements) but also qualitatively demonstrated strong interpretability. The attention map shows that the model learns to focus on discriminative features necessary for accurate classification, validating the effectiveness of the fine-tuning process on CIFAR-10.

2.2. Result of finetuning Swin Transformer base model



```
Epoch 01: Train Loss = 2.047, Val Accuracy = 29.87%
Epoch 02: Train Loss = 1.875, Val Accuracy = 34.07%
Epoch 03: Train Loss = 1.754, Val Accuracy = 37.12%
Epoch 04: Train Loss = 1.625, Val Accuracy = 39.95%
Epoch 05: Train Loss = 1.482, Val Accuracy = 42.18%
Epoch 06: Train Loss = 1.413, Val Accuracy = 44.61%
Epoch 07: Train Loss = 1.237, Val Accuracy = 46.70%
Epoch 08: Train Loss = 1.164, Val Accuracy = 49.50%
Epoch 09: Train Loss = 1.073, Val Accuracy = 51.03%
Epoch 10: Train Loss = 1.033, Val Accuracy = 52.76%
Epoch 11: Train Loss = 0.977, Val Accuracy = 53.54%
Epoch 12: Train Loss = 0.910, Val Accuracy = 55.82%
Epoch 13: Train Loss = 0.915, Val Accuracy = 56.89%
Epoch 14: Train Loss = 0.822, Val Accuracy = 57.36%
Epoch 15: Train Loss = 0.773, Val Accuracy = 58.31%
Epoch 16: Train Loss = 0.761, Val Accuracy = 59.20%
Epoch 17: Train Loss = 0.687, Val Accuracy = 60.13%
Epoch 18: Train Loss = 0.723, Val Accuracy = 61.08%
Epoch 19: Train Loss = 0.622, Val Accuracy = 61.65%
Epoch 20: Train Loss = 0.688, Val Accuracy = 62.11%
Epoch 21: Train Loss = 0.644, Val Accuracy = 62.97%
Epoch 22: Train Loss = 0.566, Val Accuracy = 62.99%
Epoch 23: Train Loss = 0.528, Val Accuracy = 63.58%
Epoch 24: Train Loss = 0.592, Val Accuracy = 64.06%
Epoch 25: Train Loss = 0.566, Val Accuracy = 64.51%
...
Epoch 47: Train Loss = 0.418, Val Accuracy = 67.43%
Epoch 48: Train Loss = 0.407, Val Accuracy = 67.67%
Epoch 49: Train Loss = 0.366, Val Accuracy = 67.73%
Epoch 50: Train Loss = 0.373, Val Accuracy = 67.40%
```

Figure 4. Train loss and Validation per epoch of swin transformer

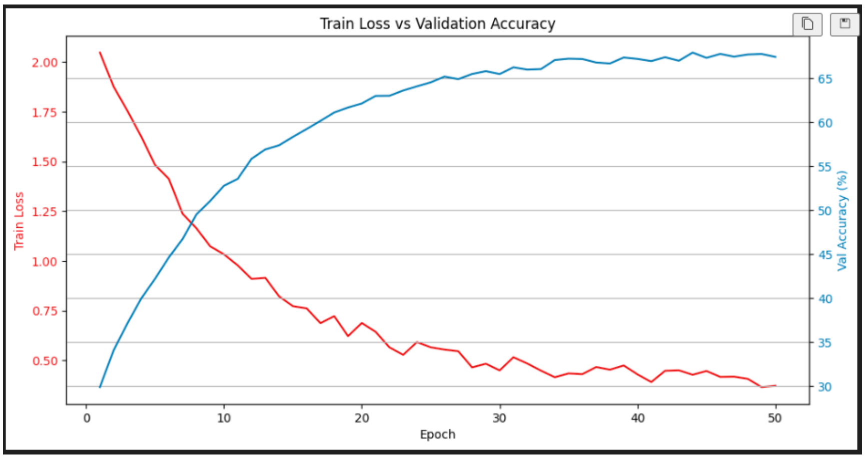


Figure 5. Train Loss vs Validation accuracy per epoch of swin transformer finetune cifar10

The training loss and validation per epoch of our fine tuning swin transformer experiment show by figure 4 and figure 5. In the final phase of training, the Swin Transformer model reached a training loss of 0.373 and achieved a validation accuracy of 67.40% on the CIFAR-10 dataset. This result reflects a strong learning outcome, where the model was able to effectively generalize to unseen data. The training loss consistently decreased throughout the 50 epochs, while validation accuracy showed a steady upward trend, ultimately plateauing around 67% in the last few epochs. This indicates convergence and model stability without overfitting.

2.2. Gradient cam analisys of finetuning Swin Transformer base model

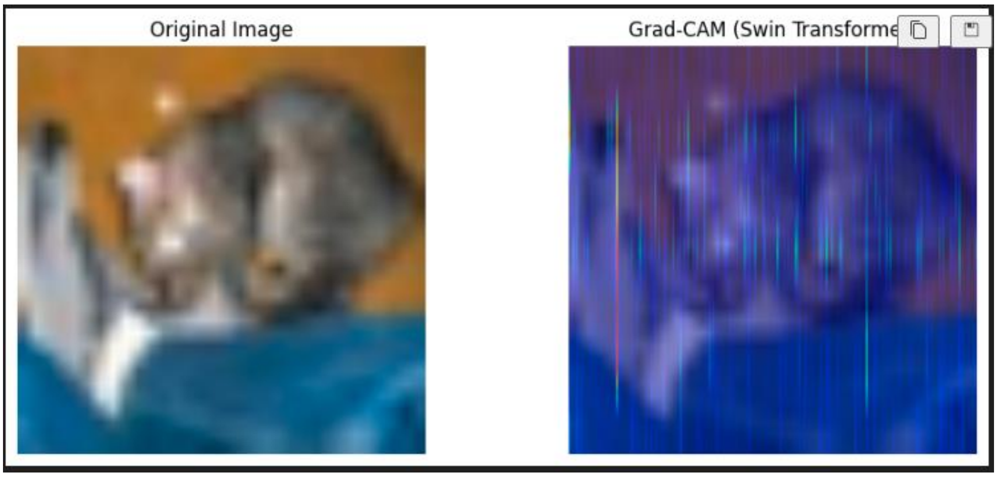


Figure 6. Gradient Cam of swin transformer on the final stage (normalization part)

The Grad-CAM result for the Swin Transformer, shown on the right side of the figure 6, displays a distinctive striped and diffused attention pattern compared to the Vision Transformer (ViT). This behavior is a direct consequence of the architectural differences between Swin and ViT models. Unlike ViT, which uses global self-attention across the entire image patch tokens, the Swin Transformer operates using shifted window, a hierarchical mechanism that computes self-attention locally within non-overlapping windows and then shifts them to enable cross-window connections. As a result, the attention maps produced by the Swin Transformer tend to be more localized and grid-like, and may not always highlight clear object boundaries. The vertical stripe artifacts observed in the Grad-CAM visualization likely reflect the window-based partitioning and the positional structure of Swin's attention mechanism.

Because Swin Transformer has a more spatially structured inductive bias (like CNNs), it sometimes distributes attention more evenly across the entire image, especially if the fine-tuned model has not fully converged to focus on the most discriminative features.

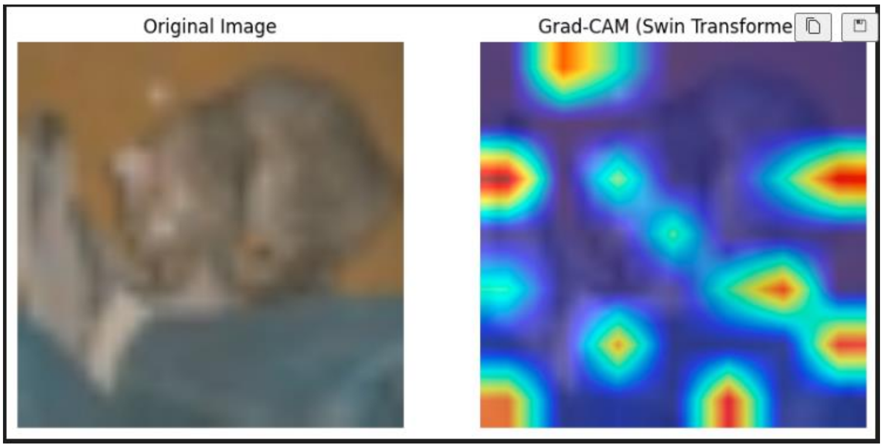


Figure 7. Gradcam of swin transformer in final stage (Attention part) with reshaping

In the other hand, the Grad-CAM result produced using the norm1 layer from the last block of the Swin Transformer differs significantly in interpretability and clarity compared to Grad-CAM results obtained from attention layers (like attn) with a proper spatial reshape_transform.

When using norm1 (as in your latest code snippet), the output features are still in token sequence form, not yet reshaped into a spatial structure. Since no reshaping or spatial remapping is applied, Grad-CAM operates over a flat sequence of tokens rather than a 2D spatial grid. As a result, the heatmap lacks a precise understanding of *where* in the image the model is focusing, and it often appears blurry, noisy, or vertically striped. These artifacts are typical when spatial correspondence between the attention tokens and image pixels is not fully reconstructed.

Contrast with that, when using the attn layer (as in your earlier example) along with a custom reshape_transform function, the token embeddings are explicitly reshaped back into their original 2D structure (e.g., 7×7 or 14×14 grid). This spatial mapping enables Grad-CAM to generate meaningful, localized heatmaps that show exactly which regions of the image the model considers important. The resulting visualization appears with distinct blobs or hotspots, highlighting object-relevant areas more accurately.