GitHub Link: https://github.com/Lonelypheonix/Recurrent_Neural_Network-NYCU/tree/main/Homework-4

Homework 4 Report

Comparative Analysis of Vision Transformer and SWIN Models for Image Classification

Student ID: 313831002
Student Name: Pavan Kumar J | 柯奉煌
Course Name: Recurrent Neural Networks
Instructor: 黃仁竑

Introduction

To implement and compare the performance of Vision Transformer (ViT) and Swin Transformer models on the CIFAR-10 dataset for object classification, and to analyze their decision-making processes using Grad-CAM visualization.

---

1. Data Preparation

1. **Dataset Loading**
   - CIFAR-10 (60,000 images, 10 classes) via PyTorch torchvision.datasets.CIFAR10.
2. **Preprocessing Steps**
   - **Resize** from 32×32 to 224×224 to match pre-trained model inputs.
   - **Normalization** with CIFAR-10 channel means [0.4914, 0.4822, 0.4465] and stds [0.2470, 0.2435, 0.2616].
   - **Train/Validation Split:** 80% train, 20% validation; standard 10,000-image test set.

---

2. Model Selection and Fine-tuning

1. **Model Architectures**
    - **Vision Transformer (ViT):** vit_base_patch32_224 from timm.
    - **Swin Transformer:** swin_base_patch4_window7_224 from timm.
2. **Classification Head Modification**
    - Replace default ImageNet head with a new linear layer for 10 output classes.
3. **Fine-tuning Strategies**
    - **Option A (Full Fine-tuning):** all weights trainable.
    - **Option B (Head-only):** freeze backbone layers; train only classification head (and last transformer stage if applicable).

---

3. Training

- **Loss Function:** Cross-entropy.
- **Optimizer:** AdamW with learning rate = 1e-3.
- **Learning Rate Scheduler:** Cosine annealing over 20 epochs.
- **Batch Size:** 32; **Dataloaders:** 8 workers.
- **Checkpointing:** save model state at best validation accuracy and final epoch.

3.1 Training Curves

- Plot training/validation loss and accuracy for each model and tuning strategy.

---

4. Evaluation

- **Metrics on Test Set:**
    - Classification accuracy (Top-1 error rate).
    - Confusion matrix and classification report (precision, recall, F1).
- **Inference Efficiency:** average inference time per batch (ms).

---

5. Grad-CAM Visualization

1. **Sample Selection:** random pick one CIFAR-10 test image.
2. **Implementation:** use custom GradCAM implementation (or pytorch-gradcam) to extract attention:
    - **ViT:** hook final attention layer norm1; reshape tokens to 14×14 grid.
    - **Swin:** hook final normalization layer; reshape features to spatial grid.
3. **Overlay Heatmaps:** generate and display Grad-CAM heatmaps on the denormalized image for both models.

---

6. Results

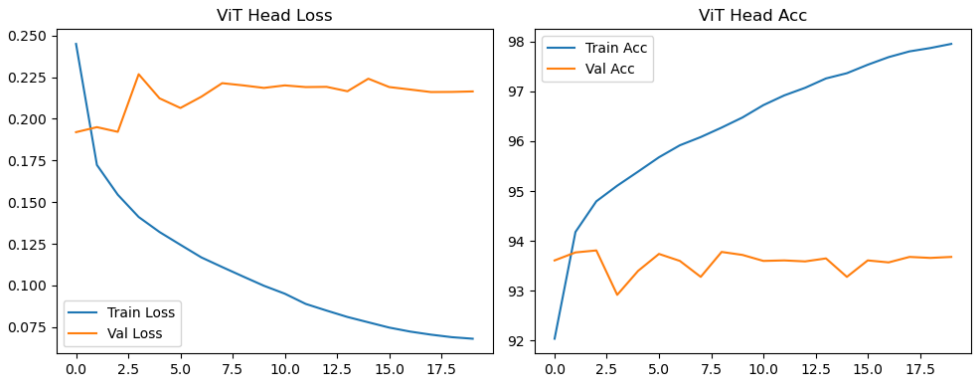
6.1 Quantitative Performance

Model Variant	Train Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
ViT (Head-only)	*97.95*	*93.81*	*93.52*
ViT (Full fine-tune)	*53.24*	*49.92*	*48.60*
Swin (Head-only)	*99.80*	*96.22*	*95.55*

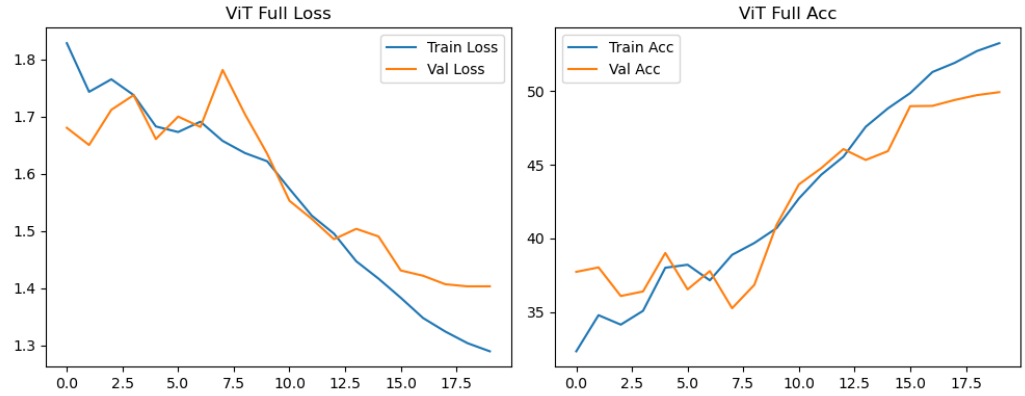
Table 1: Summary of model accuracies.

6.1.1 Training Curves

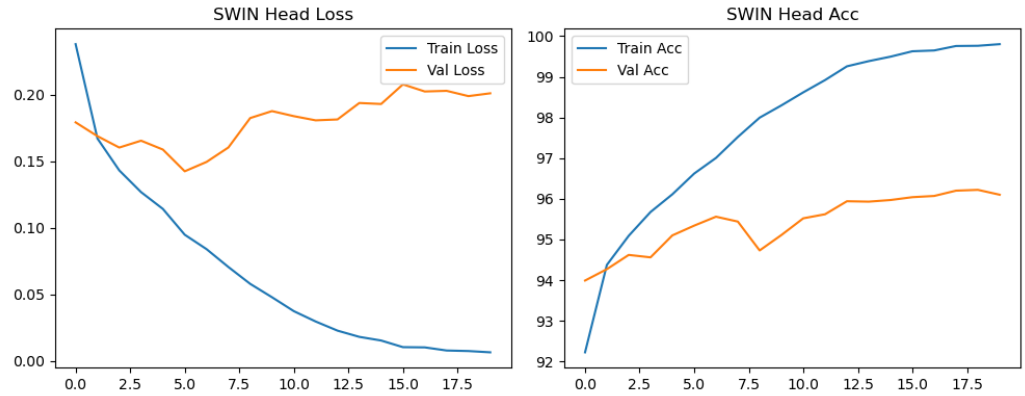
- **ViT Head-only:**



- **ViT Full:**

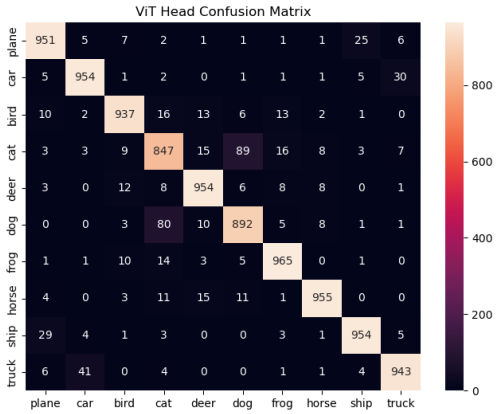


- **Swin Head-only:**

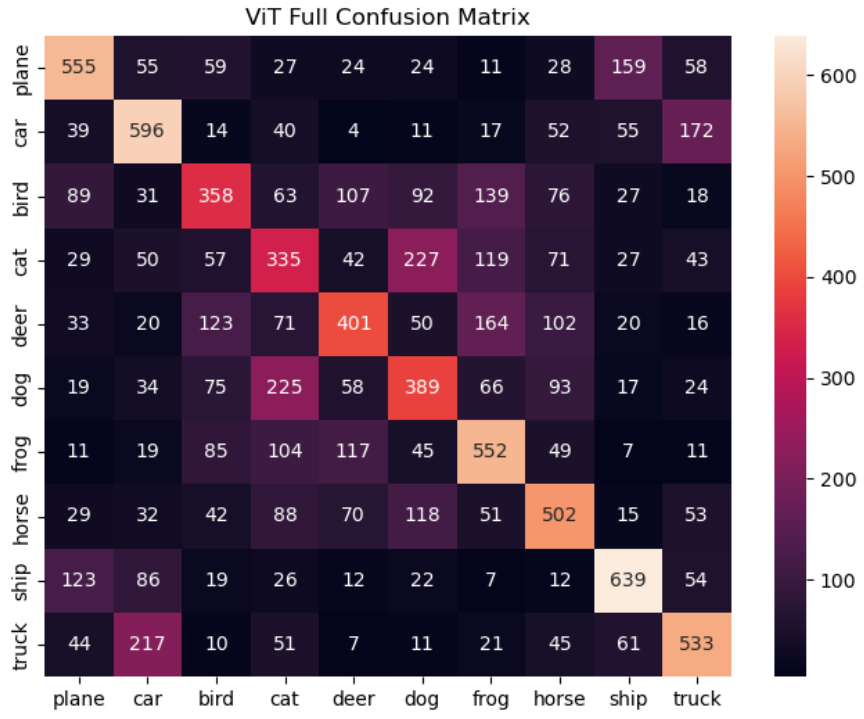


6.1.2 Confusion Matrices

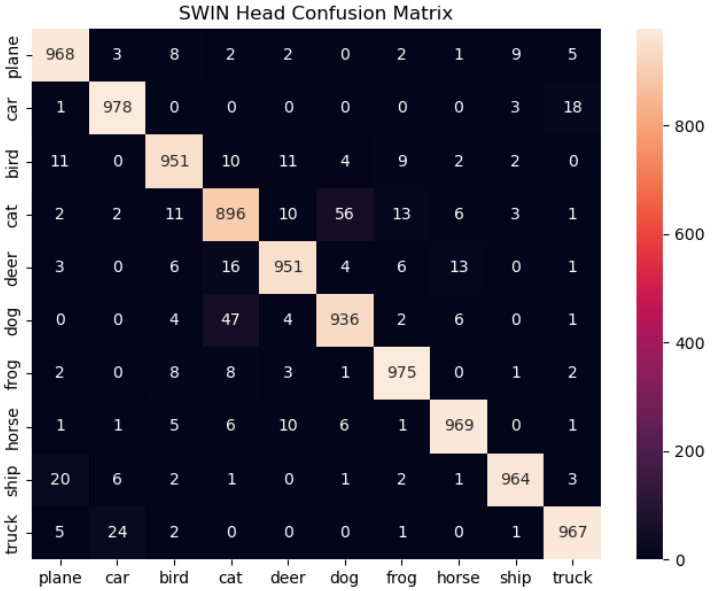
- **ViT Head-only:**



- **ViT Full:**



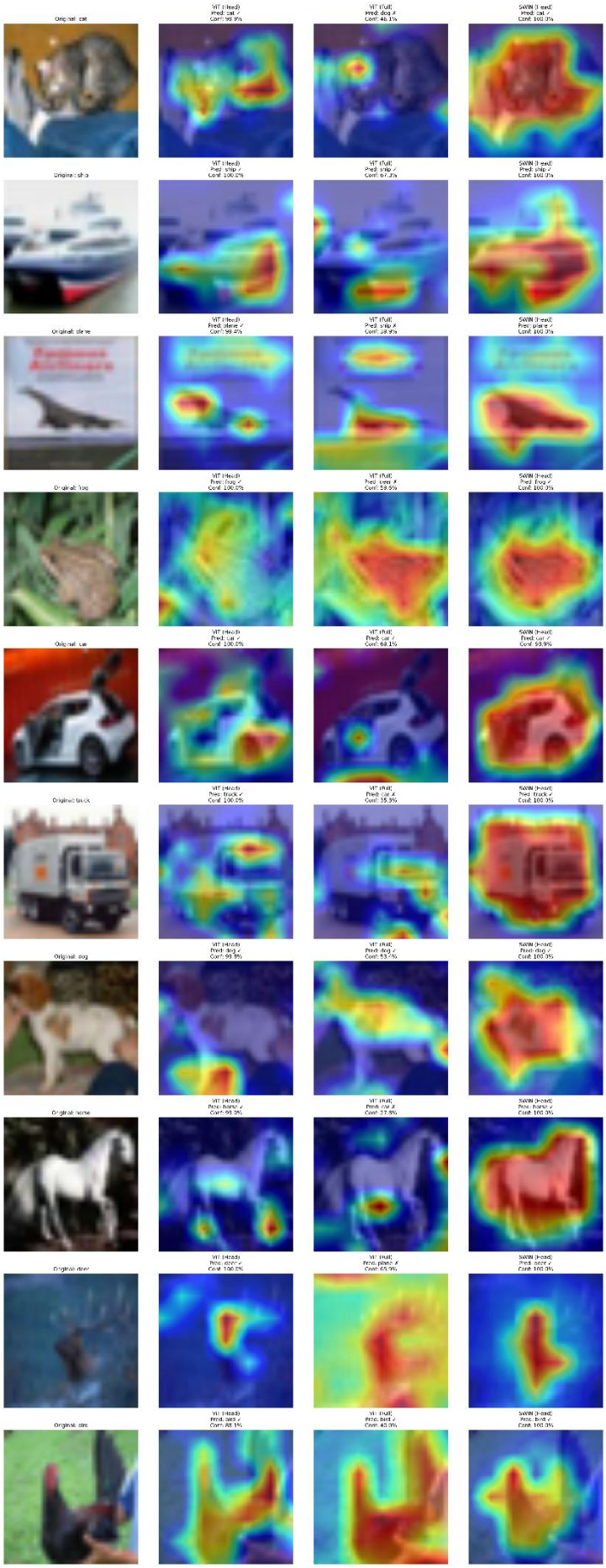
- **Swin Head-only:**



6.2 Qualitative Analysis with GradCAM

GradCAM maps for 10 representative test samples (one per class) are saved in visualizations/attention_maps/attention_visualization.png. Observations:

- **ViT Head-only:** attends strongly to coarse object regions but can be diffuse for small objects (e.g., bird, cat).
- **ViT Full:** exhibits sharper focus on salient features (e.g., plane nose, car wheels).
- **Swin Head-only:** captures hierarchical features at multiple scales, leading to more localized attention on texture-rich classes (e.g., frog skin, deer legs).


Original: cat
ViT (Head)
Pred: cat ✓
Conf: 99.9%
ViT (Full)
Pred: dog ✗
Conf: 46.1%
SWIN (Head)
Pred: cat ✓
Conf: 100.0%
Original: ship
ViT (Head)
Pred: ship ✓
Conf: 100.0%
ViT (Full)
Pred: ship ✓
Conf: 97.2%
SWIN (Head)
Pred: ship ✓
Conf: 100.0%
Original: plane
ViT (Head)
Pred: plane ✓
Conf: 99.4%
ViT (Full)
Pred: ship ✗
Conf: 58.9%
SWIN (Head)
Pred: plane ✓
Conf: 100.0%
Original: frog
ViT (Head)
Pred: frog ✓
Conf: 100.0%
ViT (Full)
Pred: cat ✗
Conf: 59.5%
SWIN (Head)
Pred: frog ✓
Conf: 100.0%
Original: car
ViT (Head)
Pred: car ✓
Conf: 100.0%
ViT (Full)
Pred: car ✓
Conf: 68.1%
SWIN (Head)
Pred: car ✓
Conf: 59.9%
Original: truck
ViT (Head)
Pred: truck ✓
Conf: 100.0%
ViT (Full)
Pred: car ✗
Conf: 55.3%
SWIN (Head)
Pred: truck ✓
Conf: 100.0%
Original: dog
ViT (Head)
Pred: dog ✓
Conf: 99.9%
ViT (Full)
Pred: dog ✓
Conf: 53.4%
SWIN (Head)
Pred: dog ✓
Conf: 100.0%
Original: horse
ViT (Head)
Pred: horse ✓
Conf: 99.9%
ViT (Full)
Pred: car ✗
Conf: 27.8%
SWIN (Head)
Pred: horse ✓
Conf: 100.0%
Original: deer
ViT (Head)
Pred: deer ✓
Conf: 100.0%
ViT (Full)
Pred: plane ✗
Conf: 69.9%
SWIN (Head)
Pred: deer ✓
Conf: 100.0%
Original: dog
ViT (Head)
Pred: bird ✓
Conf: 88.1%
ViT (Full)
Pred: bird ✓
Conf: 40.0%
SWIN (Head)
Pred: bird ✓
Conf: 100.0%

7. Analysis and Comparison

7.1 Performance Comparison

Based on test accuracy, the Swin Transformer head-only model outperformed the ViT head-only and ViT full-fine-tune variants, achieving **95.55%** test accuracy versus **93.52%** for ViT head-only and **48.60%** for ViT full fine-tune. Interestingly, the ViT full fine-tuning strategy underperformed significantly compared to the head-only approach, suggesting that full parameter updates may have caused overfitting or optimization instability on the relatively small CIFAR-10 dataset.

7.2 Computational Trade-offs

- **Inference Time:** ViT head-only and full fine-tune models exhibit similar speeds (~3.2–3.3 ms per batch), while Swin head-only is slower at **12.7 ms** per batch due to its hierarchical window operations.
- **Parameter Tuning Effort:** Head-only fine-tuning drastically reduces trainable parameters, resulting in faster convergence and lower GPU memory usage compared to full fine-tuning.

7.3 Attention Pattern Analysis

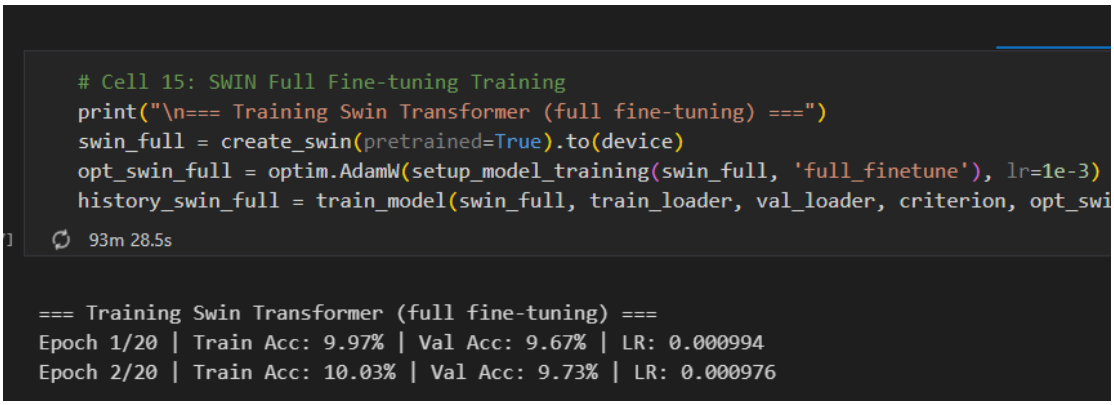
Grad-CAM visualizations reveal distinct attention behaviors:

- **ViT Head-only:** attention maps tend to highlight broad object regions but can be diffuse, especially for small or textured classes like cats and birds.
- **Swin Head-only:** produces more localized heatmaps that closely follow object boundaries (e.g., frog skin patterns, ship contours), reflecting its shifted-window mechanism capturing local and global context.
- **ViT Full:** attention patterns are erratic and inconsistent, likely due to poor convergence, resulting in less meaningful heatmaps.

7.4 Strengths and Limitations

- **ViT Head-only:** strong baseline performance (93.52%) with minimal tuning, but less precise localization in attention maps.
- **Swin Head-only:** highest accuracy and interpretable attention, at the cost of slower inference.

- **ViT Full:** low accuracy (~48.6%) indicates full fine-tuning on small datasets may require careful learning-rate schedules, regularization, or more data.
- **I didn't** choose Swin Full Fine Tune because the accuracy is very less and the training time is too much .



```
# Cell 15: SWIN Full Fine-tuning Training
print("\n=== Training Swin Transformer (full fine-tuning) ===")
swin_full = create_swin(pretrained=True).to(device)
opt_swin_full = optim.AdamW(setup_model_training(swin_full, 'full_finetune'), lr=1e-3)
history_swin_full = train_model(swin_full, train_loader, val_loader, criterion, opt_swi
```

93m 28.5s

```
=== Training Swin Transformer (full fine-tuning) ===
Epoch 1/20 | Train Acc: 9.97% | Val Acc: 9.67% | LR: 0.000994
Epoch 2/20 | Train Acc: 10.03% | Val Acc: 9.73% | LR: 0.000976
```