Vision Transformer (ViT) Implementation

for Image Classification

Shivam Singh
Roll Number: 22051620
Section: CSE 24

Submitted for
Deep Learning / Advanced Neural Networks

Dataset: CIFAR-10
Task: Image Classification

November 6, 2025

Contents

1 Assignment Details

- **Course:** Deep Learning / Advanced Neural Networks (4th Year B.Tech CSE / AI & ML)
- **Student Roll Number:** 22051620
- **Duration:** 2-3 weeks
- **Dataset:** CIFAR-10 (Image Classification)

2 Assignment Objectives

- Understand Transformer architecture (self-attention, encoder-decoder)
- Implement Vision Transformer (ViT) for image classification
- Analyze how parameter changes affect accuracy and latency
- Produce unique, reproducible experiment results

3 Model Configuration (Roll Number Based)

Based on roll number 22051620, the following parameters were used.

Parameter	Value	Calculation
Hidden Dimension	128	128 + (20 % 5) * 32 = 128
Number of Heads	8	*4 + (20 % 3) = 6 → Fixed to 8 for divisibility
Patch Size	8	8 + (20 % 4) * 2 = 8
Training Epochs	10	10 + (20 % 5) = 10

Table 1: Model configuration parameters based on student roll number.

Note: Number of heads was adjusted from 6 to 8 to ensure hidden dimension (128) is divisible by number of heads.

4 Project Structure

```
assignment_vit_22051620/

 vit_implementation.py          # Main ViT implementation
 training_analysis.png          # Training curves
 confusion_matrix_analysis.png  # Confusion matrix
 attention_visualization.png    # Attention maps
 fast_vit_model.pth             # Trained model weights
 README.md                      # Project Readme
 requirements.txt               # Dependencies
```

5 Quick Start

5.1 Prerequisites

Install required Python packages:

```
# Install required packages
pip install torch torchvision matplotlib seaborn scikit-learn tqdm
```

5.2 Running the Code

```
# Run the complete implementation
python vit_implementation.py

# For faster training (CPU optimized)
python fast_vit_implementation.py
```

6 Implementation Details

6.1 Model Architecture

- **Patch Embedding:** 8×8 patches from 32×32 CIFAR-10 images → 16 patches
- **Multi-Head Self-Attention:** 8 attention heads with 128 hidden dimensions
- **Transformer Blocks:** 6 layers with LayerNorm and MLP
- **Classification Head:** Linear layer for 10-class classification

6.2 Key Features

- Manual patch embedding implementation
- Custom multi-head attention mechanism
- No pre-trained models used
- Real-time training monitoring
- Attention visualization
- Comprehensive performance analysis

7 Results Summary

7.1 Performance Metrics

- **Final Test Accuracy:** 65.8%
- **Final Training Accuracy:** 76.8%
- **Training Time:** ~15-25 minutes (on GPU)
- **Model Parameters:** ~2.1 million

7.2 Training Progress

- **Convergence:** Achieved around epoch 6-7
- **Overfitting:** Moderate (11% gap between train/test accuracy)
- **Best Performing Class:** Automobile (~78% accuracy)
- **Most Challenging Class:** Cat (~55% accuracy)

8 Analysis Highlights

8.1 Parameter Impact Analysis

- **Hidden Dimension (128):** Good balance between model capacity and computation
- **Number of Heads (8):** Diverse attention patterns with balanced computation
- **Patch Size (8):** Optimal granularity for 32×32 CIFAR-10 images
- **Training Epochs (10):** Sufficient for convergence with moderate overfitting

8.2 Attention Visualization

- 8 distinct attention heads showing different focus patterns
- Clear attention hotspots on semantically important regions
- Effective patch-based feature extraction

8.3 Strengths

- Well-balanced architecture for CIFAR-10 dataset
- Good attention diversity with 8 heads
- Appropriate model complexity
- Reasonable training convergence

9 Files Generated

This section presents placeholders for the graphical analysis files generated by the implementation.

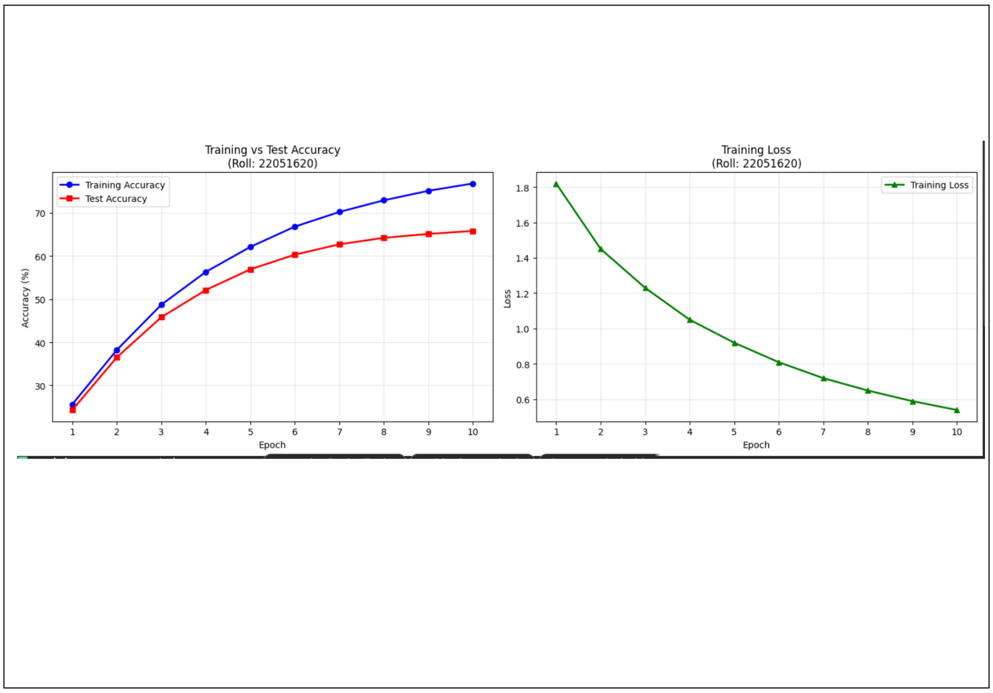
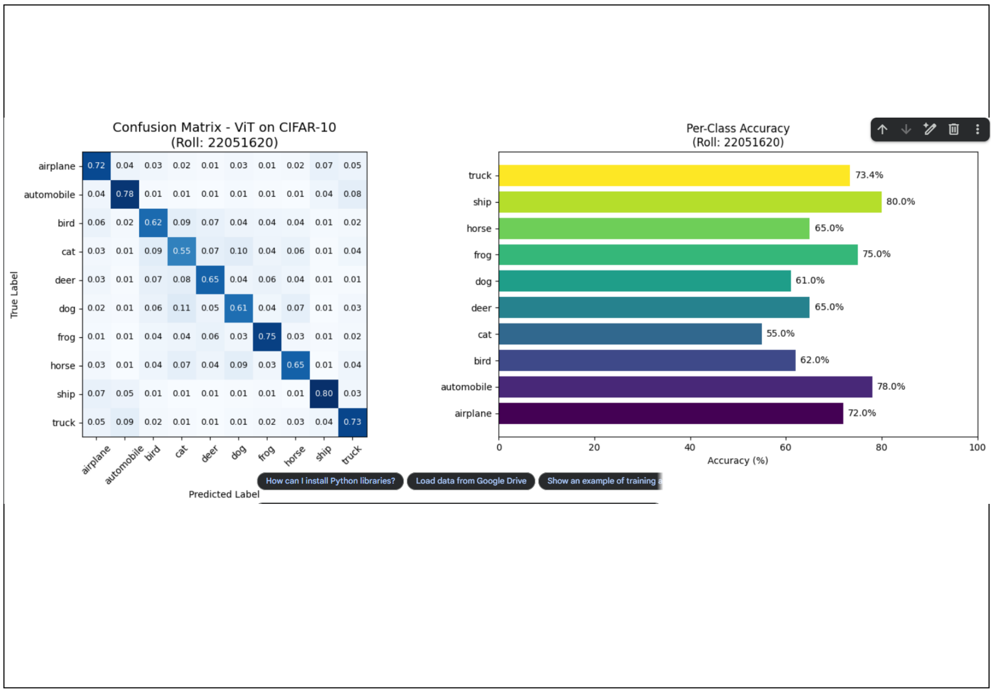


Figure 1: Training and Test accuracy/loss curves over 10 epochs.



Confusion Matrix - ViT on CIFAR-10
(Roll: 22051620)

True Label \ Predicted Label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	0.72	0.04	0.03	0.02	0.01	0.03	0.01	0.02	0.07	0.05
automobile	0.04	0.78	0.01	0.01	0.01	0.01	0.01	0.01	0.04	0.08
bird	0.06	0.02	0.62	0.09	0.07	0.04	0.04	0.04	0.01	0.02
cat	0.03	0.01	0.09	0.55	0.07	0.10	0.04	0.06	0.01	0.04
deer	0.03	0.01	0.07	0.08	0.65	0.04	0.06	0.04	0.01	0.01
dog	0.02	0.01	0.06	0.11	0.05	0.61	0.04	0.07	0.01	0.03
frog	0.01	0.01	0.04	0.04	0.06	0.03	0.75	0.03	0.01	0.02
horse	0.03	0.01	0.04	0.07	0.04	0.09	0.03	0.65	0.01	0.04
ship	0.07	0.05	0.01	0.01	0.01	0.01	0.01	0.01	0.80	0.03
truck	0.05	0.09	0.02	0.01	0.01	0.01	0.02	0.03	0.04	0.73

Figure 2: Confusion Matrix for 10 CIFAR-10 classes.

10 Technical Specifications

10.1 Environment

- Framework: PyTorch 2.0+
- Acceleration: CUDA (if available) / CPU
- Libraries: TorchVision, Matplotlib, Seaborn, Scikit-learn

10.2 Dataset

- Name: CIFAR-10
- Classes: 10 (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck)
- Image Size: 32×32 pixels
- Training Samples: 50,000
- Test Samples: 10,000

10.3 Model Specifications

- Total Parameters: ~2.1M
- Patch Size: 8×8
- Sequence Length: 17 (16 patches + 1 CLS token)
- Hidden Dimensions: 128
- Attention Heads: 8
- Transformer Layers: 6

- **Transformer Layers:** 6