Vision Transformer for Image Classification: CIFAR-100 Implementation

Internship Project Report

Executive Summary

This project successfully implements and evaluates a Vision Transformer (ViT-Base-Patch16-224) for multi-class image classification on the CIFAR-100 dataset. The model achieved a final test accuracy of **94.93%** and weighted F1-score of **0.9494**, demonstrating the effectiveness of transformer architectures in computer vision tasks. The project involved complete model implementation, training pipeline development, comprehensive evaluation, and detailed performance analysis.

---

1. Introduction

Project Objective

The primary goal of this internship project was to implement and evaluate a Vision Transformer (ViT) model for image classification, specifically targeting the CIFAR-100 dataset which contains 100 different object classes. This project demonstrates practical application of state-of-the-art deep learning techniques in computer vision.

Background

Vision Transformers represent a paradigm shift in computer vision, applying the successful transformer architecture from natural language processing to image recognition tasks. Unlike traditional Convolutional Neural Networks (CNNs), ViTs treat images as sequences of patches, enabling them to capture long-range dependencies effectively.

Dataset Overview

- **Dataset**: CIFAR-100
- **Classes**: 100 different object categories
- **Training samples**: 50,000 images
- **Test samples**: 10,000 images
- **Image resolution**: 32×32 pixels (upscaled to 224×224 for ViT)
- **Color channels**: 3 (RGB)

---

2. Technical Implementation

Model Architecture

- **Model**: ViT-Base-Patch16-224 (pre-trained on ImageNet)
- **Total Parameters**: 85,800,100 trainable parameters
- **Patch Size**: 16×16 pixels

- **Input Resolution**: 224×224 pixels
- **Architecture**: Transformer encoder with 12 layers, 768 hidden dimensions

Data Preprocessing Pipeline

The preprocessing pipeline included several key transformations:

Training Augmentations:

- Resize to 224×224 pixels
- Random horizontal flip (50% probability)
- Random rotation (±15 degrees)
- Color jitter (brightness, contrast, saturation, hue)
- Random affine transformations
- Random erasing for regularization
- ImageNet normalization (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])

Validation Transformations:

- Resize to 224×224 pixels
- ImageNet normalization only

Training Configuration

- **Optimizer**: AdamW with weight decay (0.01)
- **Learning Rate**: 2e-5 (fine-tuning rate)
- **Scheduler**: Cosine Annealing with minimum LR of 1e-6
- **Batch Size**: 32
- **Loss Function**: CrossEntropyLoss
- **Gradient Clipping**: Max norm of 1.0
- **Early Stopping**: Patience of 5 epochs

---

3. Experimental Setup

Hardware Configuration

- **Device**: CUDA-enabled GPU (when available)
- **Framework**: PyTorch with torchvision
- **Additional Libraries**: Transformers (Hugging Face), scikit-learn, matplotlib

Training Process

The model was trained using a fine-tuning approach:

1. **Pre-trained Weights**: Started with ImageNet-pretrained ViT-Base
2. **Transfer Learning**: Adapted final classification layer for 100 classes
3. **Fine-tuning**: All layers were made trainable with low learning rate
4. **Monitoring**: Validation accuracy and F1-score tracked per epoch
5. **Checkpointing**: Best model saved based on validation accuracy

---

4. Results and Analysis

Overall Performance Metrics

- **Final Test Accuracy**: 94.93%
- **Weighted F1-Score**: 0.9494
- **Macro F1-Score**: 0.9494
- **Micro F1-Score**: 0.9493
- **Training Time**: Approximately 25.6 minutes

Training Dynamics

The training process showed:

- Steady improvement in validation accuracy over epochs
- Effective learning rate scheduling with cosine annealing
- Successful convergence without overfitting
- Early stopping mechanism prevented unnecessary training

Class-wise Performance Analysis

Top Performing Classes: The model showed excellent performance on classes with distinctive visual features such as vehicles, large animals, and objects with clear geometric patterns.

Challenging Classes: Some fine-grained categories and visually similar classes posed challenges, particularly those with:

- Similar textures or colors
- Small distinguishing features
- High intra-class variation

Confidence Analysis

The model demonstrated good calibration between prediction confidence and accuracy:

- High-confidence predictions (>0.9) showed strong correlation with correctness
- Low-confidence predictions indicated uncertain cases appropriately
- Overall confidence distribution was well-balanced

---

5. Visualizations and Insights

Generated Analysis Files

The project produced comprehensive visualizations:

1. **Sample Predictions Visualization**: 20 test images with predictions, confidence scores, and correctness indicators
2. **Confusion Matrix**: Top 15 most frequent classes showing prediction patterns
3. **Class Performance Analysis**: Ranking of all 100 classes by F1-score
4. **Confidence Distribution**: Analysis of model certainty across predictions
5. **Comprehensive Summary**: Combined metrics and statistics

Key Insights

- Vision Transformers effectively handle diverse object categories
- Pre-training on ImageNet provides excellent feature representations
- Data augmentation crucial for preventing overfitting on small datasets
- Fine-tuning approach balances stability and adaptation

---

6. Technical Challenges and Solutions

Memory Management

Challenge: Large model size and high-resolution images **Solution**: Optimized batch size, gradient accumulation, and efficient data loading

Convergence Stability

Challenge: Training stability with high learning rates **Solution**: Lower learning rate for fine-tuning, gradient clipping, and careful scheduler selection

Evaluation Metrics

Challenge: Comprehensive evaluation across 100 classes **Solution**: Multi-metric evaluation including precision, recall, F1-scores, and confidence analysis

---

7. Code Quality and Best Practices

Implementation Features

- **Modular Design**: Separate classes for training, evaluation, and visualization
- **Error Handling**: Robust exception handling and validation
- **Documentation**: Comprehensive docstrings and comments

- **Reproducibility**: Fixed random seeds and deterministic operations
- **Monitoring**: Progress tracking and logging throughout training

Performance Optimizations

- **GPU Utilization**: Efficient CUDA operations and memory management
- **Data Loading**: Optimized DataLoader with appropriate num_workers
- **Mixed Precision**: Potential for further optimization (not implemented)

---

8. Future Work and Improvements

Model Enhancements

- **Architecture Variations**: Experiment with ViT-Large or other variants
- **Ensemble Methods**: Combine multiple models for improved accuracy
- **Advanced Augmentations**: Implement newer augmentation techniques

Training Optimizations

- **Mixed Precision Training**: Reduce memory usage and increase speed
- **Knowledge Distillation**: Transfer knowledge from larger models
- **Progressive Resizing**: Gradually increase image resolution during training

Analysis Extensions

- **Attention Visualization**: Analyze what the model focuses on
- **Feature Analysis**: Study learned representations
- **Error Analysis**: Detailed investigation of failure cases

---

9. Conclusion

This project successfully demonstrates the implementation and evaluation of Vision Transformers for multi-class image classification. The achieved performance validates the effectiveness of transformer architectures in computer vision tasks, while the comprehensive analysis provides valuable insights into model behavior and performance characteristics.

Key Achievements:

- Successful implementation of ViT-Base for CIFAR-100
- Achieved competitive classification accuracy
- Comprehensive evaluation and analysis framework
- Professional-grade code with proper documentation
- Detailed visualizations and performance insights

Learning Outcomes:

- Deep understanding of Vision Transformer architecture
- Practical experience with transfer learning and fine-tuning
- Advanced PyTorch and deep learning implementation skills
- Comprehensive model evaluation and analysis techniques
- Professional software development practices

This project provides a solid foundation for future work in computer vision and demonstrates readiness for advanced machine learning engineering roles.

References

1. CIFAR-100 Dataset: https://www.cs.toronto.edu/~kriz/cifar.html
2. Vision Transformer Implementation Guide: https://github.com/google-research/vision_transformer