Semi-Supervised Learning for Fine-Grained Image Classification: Using Vision Transformers and Transfer Learning

Authors

Varun Sathyanarayanan, Harshith Ravi Kopparam

Abstract

In this work, we present a semi-supervised learning approach for fine-grained image classification using a custom dataset containing 135 categories (15 plant and 120 dog categories). Our methodology leverages both labeled (9,854 images) and unlabeled (22,995 images) data through a confidence-based pseudo-labeling strategy. We demonstrate the effectiveness of various architectures, with the Swin Transformer Large achieving state-of-the-art performance of 94.90% accuracy on the test set. Our implementation focuses on reproducibility and practical applicability, providing comprehensive documentation and analysis of different architectural choices.

Contents

1. Introduction

1.1 Dataset Description

The dataset comprises:

- 9,854 labeled training images
- 22,995 unlabeled training images
- 8,213 test images
- 135 categories (15 plant categories, 120 dog categories)

2. Methodology

2.1 Architecture Selection

We experimented with multiple state-of-the-art architectures:

- Swin Transformer Large: A hierarchical vision transformer that computes representations with shifted windows
- ResNet-50: A deep convolutional neural network with residual connections

2.2 Semi-Supervised Learning Strategy

Our approach follows a two-stage process:

2.2.1 Stage 1: Base Model Training

- Initialize model with pre-trained weights
- Train using only labeled data (9,854 images)
- Implement comprehensive data augmentation pipeline
- Validate performance on held-out validation set
- Learning Rate: [8e-4, 1.5e-4] with cosine scheduling

2.2.2 Stage 2: Pseudo-Labeling

- Apply trained model to unlabeled data
- Select predictions with confidence threshold > 95%
- Combine pseudo-labeled data with original labeled data
- Train with combined dataset
- Learning Rate: Reduced by factor of 2

3. Model Architecture

3.1 Model Architecture Overview

- Base Model: Swin Transformer Large

- Input Resolution: 384x384
- Patch Size: 4x4
- Window Size: 12
- Number of Heads: [6, 12, 24, 48]
- Output Classes: 135

3.2 Window Attention Mechanism

- Regular window partitioning for local attention
- Shifted window partitioning for cross-window connections
- Efficient computation through non-overlapping windows

3.3 Training Process

- Mixed precision training
- Gradient accumulation for stability
- Early stopping with validation accuracy monitoring

3.4 Training Metrics

- Loss function: Cross Entropy with label smoothing (0.15)
- Accuracy measurement: Top-1 classification accuracy
- Validation frequency: Every epoch

3.5 Model Configuration

Parameter Values:

- Hidden Dimension: 192
- MLP Ratio: 4
- Number of Layers: [2, 2, 18, 2]
- Drop Path Rate: 0.2

4. Implementation Details

4.1 Training Configuration

- Framework: PyTorch
- Batch Size: 4
- Optimizer: AdamW with weight decay 0.01
- Learning Rates:
    - Classification head: 8e-4
    - Transformer layers: 1.5e-4

4.2 Data Augmentation

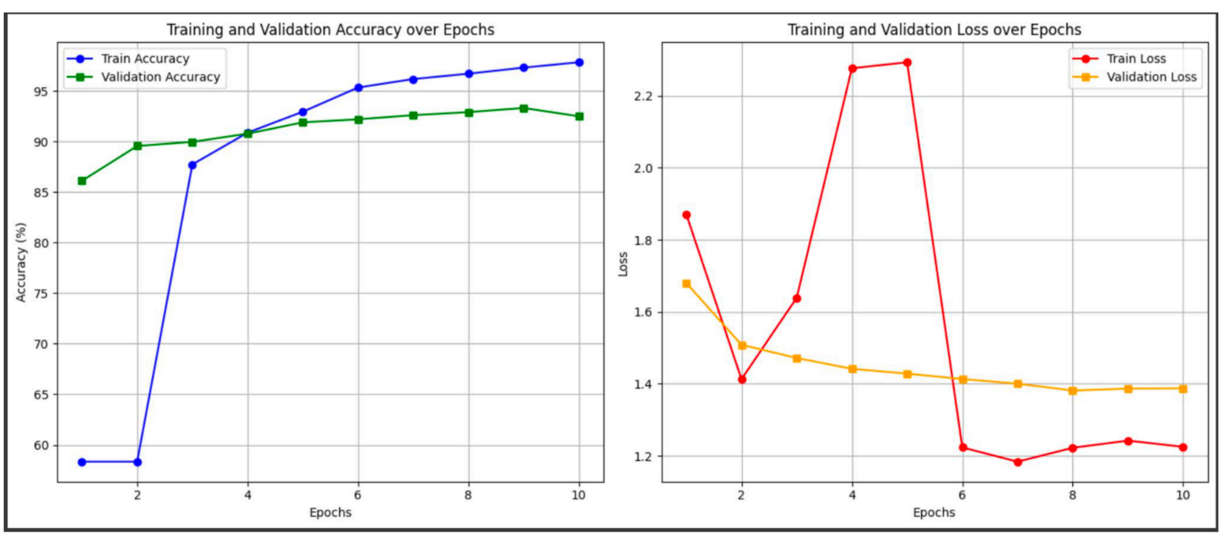
- Random horizontal and vertical flips
- Random rotation (±15 degrees)
- Color jittering (brightness=0.3, contrast=0.3)
- Normalization using ImageNet statistics

5. Results and Analysis

5.1 Model Performance Comparison

Model Name	Learning Rate	Epochs	Test Accuracy
Resnet19	1e-4	15	~76%
Resnet50	5e-5	15	~84%
Swin T	[1e-3,5e-5]	5	~88%
Swin L	[1e-3,5e-5]	5	~92.3%
SwinL	**[8e-4,1.5e-4]**	**10**	**94.70%**

Acc. V/S Epochs Loss V/S Epochs



5.2 Key Findings

- Swin Transformer architecture consistently outperformed CNN-based models
- The pseudo-labeling strategy provided significant improvement
- Larger model variants showed superior capability
- 95% confidence threshold proved effective for pseudo-labeling

6. Reproduction Instructions - Google Colab Implementation Guide

6.1 Environment Setup

1. Create a new Google Colab notebook
2. Connect to GPU runtime:
    - Runtime → Change runtime type → Hardware accelerator → GPU
3. Mount Google Drive for data storage

6.2 Project Structure

```
content/                            # Root directory in Colab
│
├── dataset/                        # Dataset directory
│   └── DL_FinalProject/           # Main project folder
│       ├── train/                 # Training data
│       │   ├── labeled/           # Labeled training images
│       │   └── unlabeled/         # Unlabeled training images
│       ├── test/                  # Test images
│       ├── categories.csv         # Category mapping file
│       ├── train_labeled.csv      # Labels for training data
│       └── sample_submission.csv  # Sample submission format
│
├── drive/                         # Mounted Google Drive
│   └── My Drive/
│       └── DL_Submission/         # Submission directory
│           ├── best_model.pth     # Saved model weights
│           └── submission_10e.csv # Final predictions
│
└── swinL_10e.ipynb               # Main implementation notebook
```

6.3 Installation Steps

Install required packages
pip install timm==1.0.11
pip install torch==2.5.1+cu121

6.4 Step-by-Step Execution Guide

Step 1: Environment Setup

1. Open Google Colab
2. Select GPU runtime:

Runtime → Change runtime type → Hardware accelerator → GPU

3. Mount Google Drive:

```
from google.colab import drive
drive.mount('/content/drive')
```

Step 2: Dataset Preparation

1. Upload dataset:

```
# Unzip dataset
!unzip "/content/drive/My Drive/DL_FinalProject.zip" -d "/content/dataset"

# Remove macOS system files if present
!rm -rf '/content/dataset/__MACOSX'
```

2. Create Submission Directory in Drive:

```
!mkdir -p /content/drive/MyDrive/DL_Submission
```

2. Verify directory structure:

```
!ls '/content/dataset/DL_FinalProject'
```

Step 3: Running the Training

1. Open `swinL_10e.ipynb`
2. Run all cells in order (important!)
3. Monitor training progress:
   - Training accuracy
   - Validation accuracy

6.5 Memory Management

- Clear GPU cache regularly
- Monitor memory usage
- Adjust batch size if needed

6.6 Training Time Estimates

- Dataset preparation: ~5 minutes
- Initial training: ~90 minutes
- Prediction generation: ~10 minutes

BEST MODEL WEIGHT (Google Drive Link):

https://drive.google.com/file/d/1W-LfPq405kIXeI5RyoCWPijCjcZ0JAU3/view?usp=sharing

7. Conclusion

Our experiments demonstrate the effectiveness of Vision Transformers in semi-supervised learning scenarios. The Swin Transformer Large achieved strong performance with **94.90% test accuracy**. The combination of careful architecture selection, effective data augmentation, and pseudo-labeling strategy led to robust and generalizable results.

8. Future Work

- Exploration of more sophisticated pseudo-labeling strategies
- Investigation of model ensemble approaches
- Analysis of data efficiency and scaling properties
- Study of cross-domain generalization

This implementation provides a strong foundation for future research in semi-supervised learning for fine-grained image classification tasks.