ECE371 Neural Networks and Deep Learning Assignment 1

22308159
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
22308159@mail2.sysu.edu.cn

May 14, 2025

Abstract

In this assignment, we implemented a flower image classification model using PyTorch and ResNet50 with transfer learning. The goal was to fine-tune a pre-trained model on a custom flower dataset and achieve high classification accuracy. The training was carried out over 25 epochs with learning rate scheduling, and the best validation accuracy reached 80.53%. This report documents the model design, training strategies, performance evaluation, and experiment analysis.

Image Classification, Transfer Learning, PyTorch, ResNet50

1 Introduction

Image classification is one of the most fundamental tasks in computer vision. In this assignment, we aim to fine-tune a pre-trained ResNet50 model on a flower dataset using PyTorch. The objective is to train a model that can accurately classify images of different flower species. The final trained model is saved and evaluated based on its performance on the validation dataset.

2 Related Work

Image classification has been extensively studied in deep learning research. Convolutional Neural Networks (CNNs) like AlexNet krizhevsky2012imagenet, VGGNet simonyan2014very, and ResNet he2016deep have significantly improved classification performance. Transfer learning, where pre-trained models on large datasets such as ImageNet are fine-tuned on smaller datasets, has become a common approach for improving performance and reducing training time. In our work, we utilize a pre-trained ResNet50 model to leverage this technique for flower classification.

3 Method

We used a ResNet50 model from PyTorch's ‘torchvision.models‘, initialized with pretrained ImageNet weights. The last fully connected layer was modified to match the number of flower classes in the dataset. The dataset was preprocessed with resizing, center cropping, normalization, and data augmentation techniques (such as random horizontal flipping).

We trained the model using the Adam optimizer and cross-entropy loss function. The learning rate started at 0.001 and was decreased stepwise using ‘torch.optim.lr$_s cheduler.StepLR‘.Trainingwasperformedfor25epochs.$

4 Experiments

4.1 Training Settings

- Model: ResNet50 (pretrained)
- Optimizer: Adam
- Learning Rate: Initial 0.001, decayed by factor of 0.1 at milestones
- Epochs: 25
- Dataset: Custom flower dataset with training and validation split

4.2 Training Logs

The best model was obtained at epoch 0, with a validation accuracy of **80.53%**. Throughout the training, the model demonstrated stable performance. The learning rate scheduler successfully reduced the rate over time, preventing overfitting.

4.3 Result Summary

Epoch	**Train Acc**	**Val Acc**
0	77.17%	**80.53%**
10	79.76%	80.53%
20	79.76%	80.53%
24	79.76%	80.53%

Table 1: Training and Validation Accuracy (Selected Epochs)

4.4 Discussion

Although the training accuracy remained stable around 79.76%, the validation accuracy did not improve beyond 80.53%. This suggests the model might have quickly reached a performance plateau on this dataset, possibly due to limited data or class imbalance. Further improvements could involve using more data augmentation, deeper architectures, or fine-tuning more layers in the model.