ECE371 Neural Networks and Deep Learning Assignment 1

LiuZongqi 22308124
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
`liuzq53@mail2.sysu.edu.cn`

Abstract: This project implements an image classification pipeline based on the ResNet-18 convolutional neural network, using transfer learning on a flower dataset consisting of five classes. The model was trained with PyTorch using data augmentation and SGD optimization. The final best validation accuracy reached 93.33%, with a total training time of around 19 minutes. This demonstrates the effectiveness of ResNet18 for small-scale classification tasks with well-augmented data. We also experimented with learning rate scheduling and found that performance improved steadily during fine-tuning.

Keywords: Image Classification, ResNet, Transfer Learning

1 Introduction

Image classification is a fundamental task in computer vision, aiming to assign category labels to input images. In this assignment, we adopt a transfer learning approach based on the ResNet-18 model to classify five types of flowers. By leveraging a pre-trained model and applying data augmentation, we are able to train an accurate classifier with limited training data. Our model achieved a validation accuracy of 93.33%, showing that even lightweight architectures like ResNet18 are powerful when combined with proper training strategies. This report describes our implementation, training pipeline, and experimental results in detail.

2 Related Work

Deep convolutional neural networks have been widely adopted for image classification tasks since the success of AlexNet [1]. Later architectures, such as VGG [2] and ResNet [3], introduced more sophisticated designs to improve depth and training stability. In particular, ResNet introduced residual connections that mitigate vanishing gradient problems, enabling the training of deeper networks.

Transfer learning has also emerged as a practical strategy, where models pre-trained on large datasets like ImageNet [4] are fine-tuned on smaller target datasets. This reduces training time and improves generalization, especially when data is limited. PyTorch's torchvision library provides such models with easy fine-tuning capabilities, as shown in our implementation.

3 Method

We adopt a ResNet-18 model pre-trained on ImageNet, replacing the final fully connected layer with a new one adapted to our five-class flower dataset. The data is preprocessed using several augmentation techniques, including:

- `RandomResizedCrop(224)`: crops images to 224×224 with a random scale and aspect ratio.

- `RandomHorizontalFlip()`, `RandomRotation(10)`, `ColorJitter()`, and `RandomAffine()` to simulate real-world variance.

Normalization is applied using ImageNet's channel-wise mean and standard deviation. We use the SGD optimizer with a momentum of 0.9 and an initial learning rate of 0.001. The learning rate is adjusted using a `StepLR` scheduler with a step size of 7 epochs and a gamma of 0.1.

The training logic is implemented using PyTorch's `train()` and `eval()` modes. CrossEntropyLoss is used as the objective function:

$$\mathcal{L}(x, y) = -\sum_{i=1}^{C} y_i \log \hat{y}_i$$

where C is the number of classes, y_i the true label, and $\hat{y}_i$ the predicted softmax probability.

The model is trained for 15 epochs, and the version with the highest validation accuracy is saved during training.

4 Experiments

The model is trained and evaluated on a five-class flower dataset. We split the dataset into 80% for training and 20% for validation using `random_split`. The training process logs accuracy and loss at each epoch.

Below is a summary of the training log:

- Initial accuracy: 66.9% (Epoch 0)
- Accuracy improved steadily to 93.3% at Epoch 11
- Learning rate decay occurred at Epoch 7 and 14
- Training time: approximately 19 minutes 35 seconds

Final Performance:

- Best validation accuracy: **93.33%**
- Final training accuracy: **92.93%**
- Final validation loss: 0.2581

The scheduler effectively reduced the learning rate, helping the model fine-tune in the later epochs. Minor fluctuations in validation loss suggest possible overfitting, which could be addressed by techniques such as early stopping or weight decay in future work.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*.