ECE371 Neural Networks and Deep Learning Assignment 1

Jianan Wu 22308182
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
wujn37@mail2.sysu.edu.cn

Abstract: This report investigates a provided dataset containing images of five flower categories (daisy, dandelion, rose, sunflower, tulip). The dataset, comprising 2,848 images, is split into training and validation sets in an 8:2 ratio and formatted according to ImageNet standards. Five common data augmentation techniques are applied to the training images: random cropping and scaling, random horizontal flipping, color jittering, random rotation, and random translation. The network model employs ResNet18 from the PyTorch framework, with the final fully connected layer modified to output five classes. Training utilizes cross-entropy loss and an SGD optimizer with an initial learning rate of 0.001, coupled with a StepLR scheduler that decays the learning rate by a factor of 0.1 every 7 epochs. The experiment trains the model for 25 epochs, recording learning rates, training/validation losses, and accuracies. Results show a maximum validation accuracy of 79.30%, with a total training time of 1 minute and 56 seconds. Overall, the training process remains stable but exhibits slight overfitting tendencies.

Keywords: Flower classification; Convolutional Neural Network (CNN); Residual Network (ResNet); Data augmentation; PyTorch

1 Introduction

Image classification is a critical task in computer vision. In recent years, deep learning methods have significantly advanced visual recognition performance, with classical models like AlexNet achieving historic breakthroughs in the ImageNet competition.Krizhevsky et al. [1] Flower image classification, as a fine-grained recognition problem, poses considerable challenges due to the visual similarities among different species. This study utilizes a dataset provided by the instructor, containing 2,848 images across five flower categories (daisy, dandelion, rose, sunflower, tulip). The goal is to train a ResNet18-based classification model for flower recognition and analyze its training dynamics and performance.

2 Related Work

Deep Convolutional Neural Networks (CNNs) have become the mainstream approach for image classification. Models such as AlexNet and VGGSimonyan and Zisserman [2] continuously improved accuracy by increasing network depth and parameter counts. He et al. proposed Residual Networks (ResNets), which address the degradation problem in deep networks through residual learning frameworks. He et al. [3] ResNet achieves exceptional performance on ImageNet and maintains efficient convergence even with extremely deep architectures. ResNet18, a shallower variant in the ResNet family, is widely adopted as a baseline for small-to-medium classification tasks due to its low computational cost.

Data augmentation is a common and effective technique to enhance model generalization. Early studies demonstrated that label-preserving transformations—such as random horizontal flipping and cropping—significantly improve model performance. Shorten and Khoshgoftaar's survey summarizes various augmentation methods, including geometric and color transformations, highlighting their importance in boosting classification accuracy. Thus, this experiment employs multiple augmentation strategies to diversify training data and mitigate overfitting.

3 Method

Dataset and Preprocessing: The flower dataset consists of 2,848 images across five categories, split into training and validation sets in an 8:2 ratio. All images are formatted to ImageNet standards for compatibility with PyTorch's `ImageFolder` loader. The following data augmentation techniques are applied to the training set:

- Random cropping and scaling
- Random horizontal flipping
- Color jittering
- Random rotation
- Random translation

These geometric and color transformations enhance model robustness to variations in object pose and lighting conditions.

Model Architecture and Training: The ResNet18 model from PyTorch's `torchvision` library is adopted, with its final fully connected layer adjusted to output five classes. Training employs cross-entropy loss and an SGD optimizer with an initial learning rate of 0.001 (momentum 0.9). A StepLR scheduler reduces the learning rate by a factor of 0.1 every 7 epochs to enable finer parameter tuning. Training runs for 25 epochs, with learning rates, training/validation losses, and accuracies recorded at each epoch. All experiments are conducted on the AutoDL cloud platform, completing in 1 minute and 56 seconds.

4 Experiments

In this experiment, we used the provided flower dataset containing five categories of flower images (2,848 images total), split into training and validation sets at an 8:2 ratio. The model was ResNet18, trained for 25 epochs on an image classification task. The total training time was approximately 1 minute and 56 seconds, with a best validation accuracy of 79.30

Table 1: Training Process Metrics

Epoch	LR	Train Loss	Train Acc	Val Loss	Val Acc
0	0.001	0.5311	0.8007	0.5440	0.7895
1	0.001	0.5119	0.7998	0.5450	0.7912
2	0.001	0.5144	0.8011	0.5547	0.7825
3	0.001	0.5048	0.7968	0.5433	0.7895
4	0.001	0.4897	0.7989	0.5482	0.7789
5	0.001	0.4859	0.8011	0.5426	0.7842
6	0.001	0.4924	0.7981	0.5558	0.7737
7	0.0001	0.4688	0.8051	0.5558	0.7912
8	0.0001	0.4664	0.8055	0.5498	0.7912
9	0.0001	0.4665	0.8055	0.5597	0.7930
10	0.0001	0.4672	0.8077	0.5531	0.7895
11	0.0001	0.4625	0.8051	0.5382	0.7877
12	0.0001	0.4662	0.8055	0.5612	0.7895
13	0.0001	0.4555	0.8038	0.5576	0.7825
14	1e-05	0.4602	0.8064	0.5530	0.7842
15	1e-05	0.4654	0.8020	0.5543	0.7877
16	1e-05	0.4574	0.8068	0.5588	0.7860
17	1e-05	0.4564	0.8064	0.5460	0.7860
18	1e-05	0.4553	0.8060	0.5563	0.7807
19	1e-05	0.4541	0.8047	0.5440	0.7877
20	1e-05	0.4606	0.8064	0.5414	0.7877
21	1e-06	0.4569	0.8033	0.5600	0.7895
22	1e-06	0.4579	0.8073	0.5486	0.7912
23	1e-06	0.4557	0.8068	0.5474	0.7842
24	1e-06	0.4583	0.8060	0.5397	0.7895

The learning rate was scheduled to decrease at epochs 7, 14, and 21 (from 0.001 to 0.000001). After each adjustment, the training loss stabilized, and validation performance remained consistent, indicating that this scheduling strategy positively contributed to convergence and helped control fluctuations. Slight overfitting was observed toward the later stages, as validation accuracy plateaued below training accuracy.

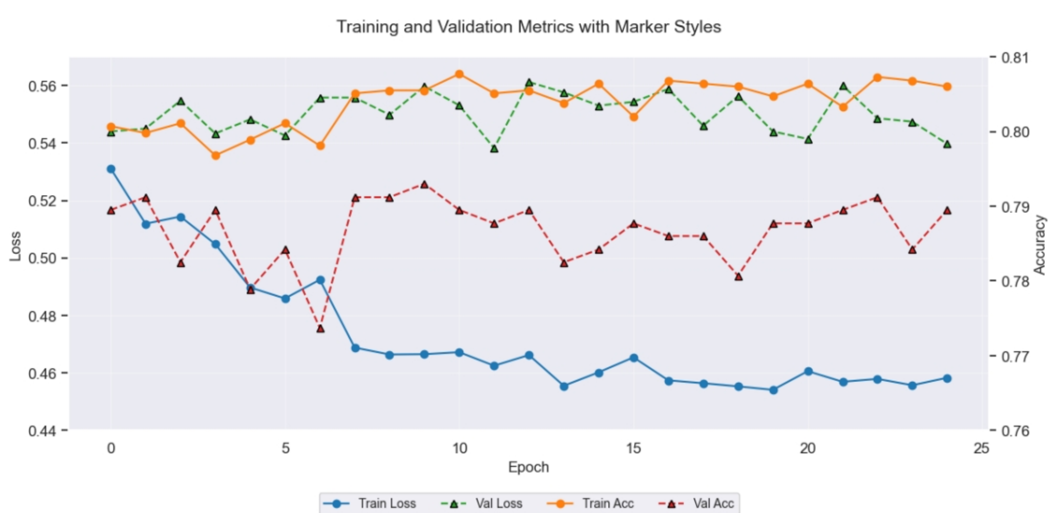

Figure 1: Loss and Accuracy Curve

For future work, regularization methods such as Dropout could be introduced to reduce overfitting. Exploring deeper models (e.g., ResNet50) and more flexible learning rate schedules, as well as extending training epochs, may further optimize performance.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[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.