ECE371 Neural Networks and Deep Learning Assignment 1

Qingyuan Liu, 22308116
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
liuqy96@mail2.sysu.edu.cn

Abstract: To address the inherent challenges of feature extraction difficulties and insufficient accuracy associated with traditional methods in flower image classification, a solution based on the ResNet18 deep learning model was employed. This model leverages ImageNet pre-trained weights through transfer learning and enhances its generalization capability via data augmentation techniques, including random cropping, flipping, rotation, and color jittering. Experiments were conducted on a dataset comprising five common flower categories. The experimental results indicate that after 25 training epochs, the model achieved a final validation accuracy of 90.2%, with a peak accuracy of 91.92%. Furthermore, the synchronized trends of the training and validation curves demonstrated no significant overfitting.

Keywords: Image classification, Deep learning, ResNet18, Transfer Learning

1 Introduction

Classification tasks are one of the core problems in the field of computer vision. Traditional machine learning methods, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees, while successful in some simple classification problems, often exhibit significant limitations when faced with large-scale image data. These methods rely on manual feature extraction, which necessitates extensive domain knowledge and expert experience. However, image features are often complex and diverse, making it challenging for traditional approaches to comprehensively capture high-level semantic information. Consequently, when processing large-scale, highly variable image data, traditional methods frequently encounter issues such as overfitting and difficulties in feature selection, leading to suboptimal classification accuracy. This is especially pertinent to flower classification tasks, where the wide variety of species and significant intra-class variations due to differing shooting angles, lighting conditions, and backgrounds further compound the complexity and challenges.

To overcome the shortcomings of traditional methods in feature extraction, feature selection, computational efficiency, and handling high-dimensional complex data, and to improve the accuracy of flower classification tasks, this study employs deep learning techniques to construct an efficient flower image classification model. By leveraging the advantages of Convolutional Neural Networks (CNNs)[1] in automatic feature extraction and learning, we selected a deep learning model based on the ResNet18 architecture. This model automatically learns effective feature representations from raw images, thereby circumventing the limitations of traditional methods that rely on manually designed features. In the data preprocessing stage, we introduced various data augmentation techniques, including random cropping, random horizontal flipping, rotation, color jittering, and affine transformations, to increase data diversity and enhance the model's generalization ability. In our experiments, these data augmentation methods were applied to a dataset containing multiple common flower species, and a transfer learning strategy was combined to fine-tune the ResNet18 model. The core objective of this research is to validate the effectiveness and robustness of the employed deep learning framework in complex flower classification tasks through systematic experimental design and performance evaluation. This aims to provide a viable technological pathway for the automated and intelligent identification of flowers.

2 Related Work

2.1 Convolutional Neural Networks for Classification Tasks

Deep learning, particularly convolutional neural networks (CNNs), has demonstrated exceptional performance in image classification tasks. Unlike traditional machine learning methods that rely on handcrafted feature extraction, deep learning approaches automatically learn features from raw data through multi-layer neural networks and perform classification using optimized network architectures. Early CNN models, such as AlexNet [2], laid the foundation for the application of deep learning in image classification tasks. By stacking multiple convolutional layers, AlexNet significantly improved the accuracy of image classification. With advancements in computational power and the expansion of dataset scales, deeper network architectures, such as VGG [3] and GoogLeNet [4], were subsequently proposed. These networks further enhanced classification performance by increasing network depth and introducing novel network modules, enabling the models to capture finer details and achieve superior results across various image classification tasks. However, as network depth increased, challenges in training deep networks became more prominent, particularly issues related to vanishing gradients and training instability. To address these challenges, ResNet [5] introduced residual connections, incorporating skip connections that allow the network to directly learn the residual between input and output rather than the complete mapping.

2.2 Methods for Improving Training Performance

In image classification tasks, a series of effective strategies are widely employed to enhance training performance. Transfer learning is utilized to leverage general visual features pre-trained on large-scale datasets such as ImageNet, enabling faster convergence and improved performance baselines for new tasks. Data augmentation is applied by introducing random geometric and photometric transformations to training images, thereby artificially increasing the diversity of the training set and enhancing model robustness. At the optimization level, the stochastic gradient descent (SGD) optimizer, combined with learning rate scheduling—such as stepwise learning rate decay—is used to achieve rapid and stable convergence. Additionally, hardware acceleration with GPUs is employed, significantly reducing training time through parallel computing capabilities. The integrated application of these techniques forms the cornerstone for improving the training efficacy of deep learning models.

3 Method

3.1 Model architecture

The ResNet-18 residual network, as depicted in Figure 1, is adopted as the foundational backbone network. Input images processed by the ResNet-18 model are standardized to a resolution of 224×224 pixels. The network is composed of a series of convolutional layers, pooling layers, and residual modules, specifically comprising an initial 7×7 convolutional layer (64 filters, stride 2), followed by a 3×3 max-pooling layer (stride 2). Subsequently, four groups of residual blocks (each containing 2, 2, 2, and 2 residual modules, respectively) are included, with each residual block consisting of 3×3 convolutional layers (filter counts of 64, 128, 256, and 512, respectively) and employing skip connections to mitigate the vanishing gradient problem. The final output is generated through a fully connected layer (fc) with a softmax activation function for classification. To leverage prior knowledge from large-scale datasets and accelerate model convergence, pre-trained ResNet-18 weights from the ImageNet dataset are loaded, implementing a transfer learning strategy. For the flower classification task in this study, the pre-trained ResNet-18 model is adapted by retaining its

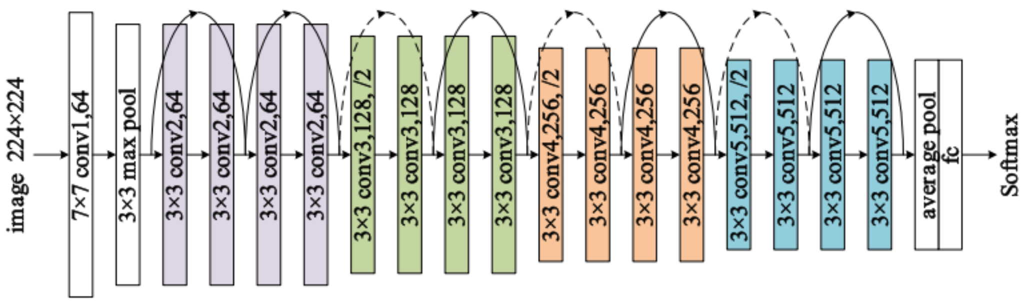

Figure 1: ResNet18 architecture Network

convolutional feature extraction components and replacing the original fully connected classification layer to accommodate the five-class dataset.

3.2 Data Preprocessing

Prior to model training, a series of systematic preprocessing and data augmentation operations are applied to the image data to enhance the model's generalization ability and robustness. Initially, images are randomly resized, and a 224×224 pixel region is cropped to meet the model's input requirements while improving adaptability to variations in object scale and position. Subsequently, geometric transformations are introduced, including random horizontal flipping with a certain probability, random rotation within a ±30-degree range, and random affine transformations incorporating small-angle rotations (±10 degrees) and minor translations (±10

3.3 Optimizer and Loss Function

The stochastic gradient descent (SGD) algorithm is selected as the optimizer to update the model's network parameters. An initial learning rate of 0.001 is set, and a momentum factor of 0.9 is incorporated to accelerate convergence and mitigate the risk of local optima. To dynamically adjust the learning rate during training, a step-wise learning rate decay strategy is adopted, whereby the current learning rate is multiplied by a decay factor of 0.1 every 7 training epochs.

For this multi-class image classification task, the cross-entropy loss function is employed as the training objective. This loss function effectively measures the discrepancy between the model's predicted class probability distribution and the true class labels, serving as a standard metric for guiding the learning of classification models.

4 Experiments

4.1 Experimental Parameter Settings

The flower dataset used in the experiments comprises five categories: daisy, dandelion, tulip, sunflower, and rose. The dataset is split into training and validation sets at an 80:20 ratio. During data loading, training data are organized into batches of size 32 and randomly shuffled at the beginning of each training epoch. Data loading efficiency is enhanced by employing four parallel worker processes. A total of 25 training epochs are conducted, with experiments implemented using the PyTorch deep learning framework on a server equipped with an RTX3060 GPU. This section should include data tables, figures/charts, and analytical discussion. You are expected to integrate classroom knowledge with your experimental findings for comprehensive discussion, including critical analysis of any identified issues. This section will largely determine the overall execution quality and corresponding grading of the report.

4.2 Experimental Results

Table 1 presents the peak training and validation accuracies achieved during model training, with the peak training accuracy reaching 91.83

Table 1: Best achieved training and validation accuracies (alternative format).

	Training Accuracy (%)	Validation Accuracy (%)
Best Value	91.83	91.93

Figure 2 illustrates the results of the entire training process, with the left subplot depicting training loss and validation loss , and the right subplot showing training accuracy and validation accuracy . At an initial learning rate of 0.001 (Epochs 0–6), the training loss decreases rapidly from 0.960 to 0.345, and the validation loss drops from 0.550 to 0.281; concurrently, the training accuracy rises from 0.644 to 0.877, and the validation accuracy increases from 0.804 to 0.898. As the learning rate is progressively reduced to 0.0001, 0.00001, and 0.000001 at Epochs 7, 14, and 21, both loss and accuracy continue to improve and stabilize. At the conclusion of training (Epoch 24), the training loss reaches 0.303, the validation loss is 0.263, the training accuracy is 0.895, and the validation accuracy is 0.902, with the highest validation accuracy of approximately 0.919 achieved at Epoch 22. The training and validation curves exhibit synchronized trends with minimal discrepancies, indicating strong model generalization and no evident overfitting. The step-wise learning rate schedule effectively facilitates model convergence.

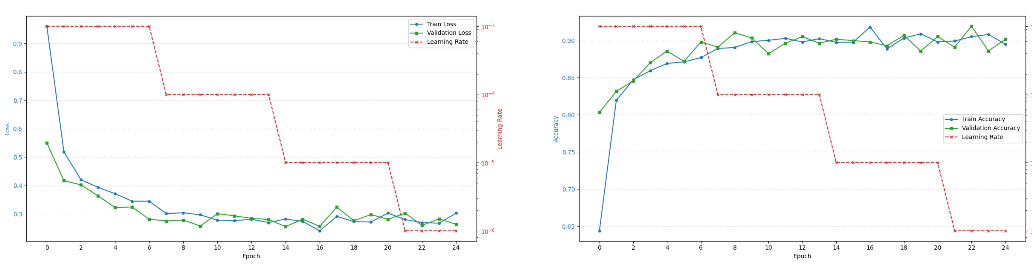

Figure 2: Model training performance: training and validation loss (left) and accuracy (right) plotted against epochs.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi:10.1109/5.726791.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

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

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. doi:10.1109/CVPR.2015.7298594.

[5] 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. doi:10.1109/CVPR.2016.90.