ECE371 Neural Networks and Deep Learning Assignment 1

Hanwen Liu (22322041)
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
liuhw56@mail2.sysu.edu.cn

Abstract: This report presents the implementation and analysis of a deep learning model for flower image classification. Using transfer learning, a pre-trained ResNet-18 model was fine-tuned to classify images from five flower categories. Data augmentation techniques were employed to enhance model generalization. The model achieved a validation accuracy of 93.68% after training with a learning rate schedule. Analysis shows that the first learning rate decay contributed significantly to performance, while subsequent reductions yielded minimal improvements. This implementation demonstrates the effectiveness of transfer learning for image classification tasks with limited datasets.

Keywords: Image Classification, Transfer Learning, ResNet-18, Data Augmentation

1 Introduction

Image classification remains a fundamental challenge in computer vision with numerous practical applications. In this assignment, the task of classifying flower images into five distinct categories using deep learning techniques is addressed. The flower dataset contains images of daisies, dandelions, roses, sunflowers, and tulips, presenting various challenges including intra-class variations in shape, color, and orientation.

Deep Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image classification tasks. However, training these networks from scratch requires substantial computational resources and large datasets. To overcome these limitations, transfer learning is employed, leveraging a pre-trained ResNet-18 model that has been trained on the large-scale ImageNet dataset. This approach allows for benefiting from the feature extraction capabilities already learned by the model while adapting it to the specific classification task.

The objectives of this assignment are to implement data augmentation techniques, modify the pre-trained model for flower classification, implement appropriate loss functions and optimization strategies, and evaluate model performance.

2 Related Work

Image classification has evolved significantly with the development of deep learning techniques. The breakthrough work by Krizhevsky et al. [1] introduced AlexNet, which significantly outperformed traditional methods on the ImageNet competition, marking the beginning of the deep learning era in computer vision.

The VGG networks by Simonyan and Zisserman [2] demonstrated that increasing network depth with very small convolutional filters consistently improved performance. To address degradation problems in deeper networks, He et al. [3] introduced Residual Networks (ResNet), which incorporate skip connections to facilitate gradient flow. The ResNet architecture has become one of the most widely used backbone networks for various computer vision tasks.

Transfer learning has emerged as a powerful technique for applying deep learning to domains with limited labeled data. Yosinski et al. [4] demonstrated that features learned in early layers of CNNs trained on large datasets contain general features that can be transferred to other tasks. Kornblith et al. [5] found that models performing better on ImageNet generally transfer better to other tasks.

For flower classification specifically, Nilsback and Zisserman [6] created the Oxford 102 Flowers dataset and proposed methods for fine-grained classification. Hu et al. [7] employed squeeze-and-excitation networks for improved feature representation. Data augmentation techniques, as explored by Shorten and Khoshgoftaar [8], have proven particularly valuable for improving generalization when training data is limited.

The approach in this work builds upon these advancements by utilizing a pre-trained ResNet-18 model with transfer learning, implementing multiple data augmentation techniques, and optimizing the training process through an appropriate learning rate schedule.

3 Method

3.1 Dataset Preparation

The flower dataset contains 5 categories: daisy, dandelion, rose, sunflower, and tulip, with approximately 500-600 images per class. The dataset was randomly split into training (80%) and validation (20%) sets, maintaining the class distribution.

3.2 Data Augmentation

To enhance model generalization and reduce overfitting, five data augmentation techniques were implemented:

1. Random Resized Crop: Randomly crops and resizes to 224×224 pixels, introducing variations in position and scale.
2. Random Horizontal Flip: Horizontally flips images with 50% probability.
3. Random Rotation: Rotates images within ±15 degrees.
4. Random Vertical Flip: Vertically flips images with 20% probability.
5. Color Jitter: Randomly adjusts brightness, contrast, saturation, and hue.

Additionally, normalization was applied using the mean and standard deviation values from the ImageNet dataset ([0.485, 0.456, 0.406] and [0.229, 0.224, 0.225]).

3.3 Model Architecture

The ResNet-18 architecture was adopted for this task. For transfer learning, a ResNet-18 model pre-trained on ImageNet was utilized. To adapt this model to the specific flower classification task, the final fully connected layer was modified:

```
num_features = model.fc.in_features
num_classes = len(class_names)  # 5 flower categories
model.fc = nn.Linear(num_features, num_classes)
```

This modification replaces the original 1000-class output layer with a 5-class output layer corresponding to the flower categories.

3.4 Loss Function and Optimization

For this multi-class classification task, Cross-Entropy Loss was implemented. For optimization, Stochastic Gradient Descent (SGD) with momentum and weight decay was employed:

```
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.001,
                      momentum=0.9, weight_decay=1e-4)
```

To improve training dynamics, a step learning rate scheduler was implemented that reduces the learning rate by a factor of 0.1 every 7 epochs:

```
scheduler = lr_scheduler.StepLR(optimizer, step_size=7, gamma=0.1)
```

This approach allows the model to converge to a better local minimum by initially making large updates and then refining the weights with smaller updates as training progresses.

4 Experiments

4.1 Experimental Setup

The model was trained for 25 epochs using a batch size of 32 on a NVIDIA GeForce RTX 3060 Laptop GPU. The model was evaluated after each epoch on the validation set, and the model with the highest validation accuracy was saved.

4.2 Results and Analysis

Table 1 shows the training and validation accuracy across key epochs. The model achieved a final validation accuracy of 93.68%, with the best-performing model occurring at epoch 7.

Table 1: Performance metrics at key training stages

Stage	Epoch	Learning Rate	Train Acc.	Val. Acc.
Initial	0	0.001	67.21%	84.56%
Before 1st LR decay	6	0.001	89.03%	91.75%
Best model	7	0.0001	90.47%	93.68%
Before 2nd LR decay	13	0.0001	92.01%	91.23%
Final	24	0.000001	91.44%	91.23%

Several interesting patterns were observed during training:

Early Performance: The model achieved relatively high validation accuracy (84.56%) after just the first epoch, demonstrating the effectiveness of transfer learning. This suggests that the pre-trained ResNet-18 already contained relevant features for flower classification.

Impact of Learning Rate Schedule: The learning rate reduction at epoch 7 (from 0.001 to 0.0001) led to an immediate improvement in validation accuracy, from 91.75% to 93.68%. However, subsequent learning rate reductions (at epochs 14 and 21) did not yield further improvements. This indicates that the initial training phase with a higher learning rate effectively located a promising region in the parameter space, while the first reduction enabled fine-tuning within that region.

Generalization Gap: Throughout training, the validation accuracy remained close to the training accuracy, particularly in early epochs. This suggests that the validation set might have contained slightly easier examples or that the data augmentation applied during training made the training task artificially more difficult. The absence of a significant generalization gap also indicates that overfitting was not a major concern, likely due to effective data augmentation and transfer learning.

Validation Performance Plateau: After epoch 7, validation performance fluctuated but did not consistently improve. This suggests that either the model had reached its capacity for this dataset with the current architecture, or longer training with a more gradual learning rate schedule might be beneficial.

4.3 Discussion and Potential Improvements

While the model achieved strong performance, several improvements could potentially enhance results further:

Fine-tuning Strategy: In this implementation, all model parameters were updated during training. An alternative approach would be to freeze the early convolutional layers and only update the later layers or introduce a differential learning rate approach.

Advanced Architectures: Although ResNet-18 provided good results, more advanced architectures like EfficientNet or Vision Transformers could potentially yield better performance, especially with appropriate transfer learning strategies.

Addressing Class Imbalance: The flower dataset has slight variations in the number of images per class. Implementing techniques like weighted sampling or focal loss could help improve performance for underrepresented classes.

In conclusion, this implementation demonstrates the effectiveness of transfer learning combined with data augmentation for flower classification. The approach of fine-tuning a pretrained ResNet-18 model achieved high validation accuracy (93.68%), showing that advanced deep learning techniques can be successfully applied to specialized image classification tasks with limited training data.

References

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

[2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations, 2015.

[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. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems, volume 27, 2014.

[5] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2661–2671, 2019.

[6] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pages 722–729. IEEE, 2008.

[7] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7132–7141, 2018.

[8] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of Big Data, 6(1):1–48, 2019.