ECE371 Neural Networks and Deep Learning Assignment 1

ShiHaifeng22308157
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
shihf5@mail2.sysu.edu.cn

May 14, 2025

Abstract

This report presents the process of implementing a flower classification model through transfer learning using the pre - trained ResNet18. By modifying the last fully - connected layer, the model can classify five flower categories. Meanwhile, data augmentation techniques and hyper - parameter adjustment methods are employed to enhance the generalization ability of the model. After 25 training epochs, the best accuracy rate achieved by the model on the validation set is 90.53%.

Transfer Learning; Residual Network; Data Augmentation

1 Introduction

The goal of this assignment is to develop a powerful flower classification model using a pre - trained neural network. We choose ResNet18 as the backbone network and, through transfer learning, mainly including data pre - processing, data augmentation strategies, and hyper - parameter optimization steps, make it adapt to the target task.

2 Related Work

ResNet18 is an 18 - layer deep convolutional neural network with residual blocks and skip - connection structures. It has obvious advantages in image classification applications: it can efficiently extract features and automatically learn complex features for accurate classification; it has good generalization ability and performs well on different datasets; it is relatively lightweight and computationally fast, making it suitable for resource - constrained environments. Practical applications include being directly used in image classification tasks, such as training and testing on various standard datasets; it can be used for transfer learning, where pre - trained models are fine - tuned to adapt to new tasks; it can also be applied to object detection, semantic segmentation, industrial inspection, medical image analysis, and other fields.

3 Method

3.1 Dataset and Preprocessing

The flower dataset contains 3,670 images of 5 categories, including 2,936 training images and 734 validation images. To alleviate the over - fitting problem, a variety of data augmentation techniques are adopted, such as randomly cropping and resizing to 224×224, randomly flipping horizontally or vertically, color jittering (with brightness, contrast, and saturation varying by ±20%), and randomly rotating by ±20°. Moreover, the images are normalized using the statistical data of ImageNet (mean = [0.485, 0.456, 0.406], standard deviation = [0.229, 0.224, 0.225]) and then converted into tensors.

3.2 Model Architecture and Training Configuration

We use ResNet18 as the backbone network. We only replace the last fully - connected layer and keep all the convolutional layers. The backbone network outputs 512 - dimensional features through adaptive average pooling, and these features are input into a new linear layer "fc = nn.Linear(512, 5)" to map to the five flower categories.

Use cross - entropy loss as the loss function. The optimizer is stochastic gradient descent (SGD), and its parameters are set as follows: the learning rate is 0.0008, the momentum is 0.92, and the weight decay is 0.00015. The StepLR learning rate scheduler is adopted, and the learning rate is decayed by 10% every 7 epochs. The model is trained on NVIDIA GPU for 15 epochs with a batch size of 32.

4 Experiments

4.1 Evaluation Metrics

The model operation results include the learning rate, as well as the loss and accuracy of the training set and the validation set.

4.2 Key Results

The training of this deep learning model took 26 minutes and 25 seconds and consisted of 24 epochs. The initial learning rate was 0.001, which decreased to 0.0001 at the 7th epoch, to 0.00001 at the 14th epoch, and to 0.000001 at the 21st epoch. The training loss decreased from 0.7909 in the 0th epoch to 0.2047 in the 24th epoch, and the training accuracy increased from 0.7103 to 0.9254. The validation loss and accuracy fluctuated during the training process. The lowest validation loss was 0.4018 in the 7th epoch, and the highest validation accuracy was 0.9053 in the 15th epoch. The best validation accuracy was 0.905263.