ECE371 Neural Networks and Deep Learning Assignment 1

ZhanYan
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
zhany27@mail2.sysu.edu.cn

Abstract: The purpose of this document is to provide both the basic template of the assignment and submission guidelines. Abstracts should be a single paragraph, between 4–6 sentences long, ideally. Briefly describe the work you have completed and the insights you have gained. The main text should not exceed 4 pages, excluding references.

Keywords:

1 Introduction

Image classification is a fundamental task in computer vision and deep learning. In this assignment, we focus on classifying images of flowers into multiple categories using convolutional neural networks (CNNs). We implemented two models: one based on the PyTorch torchvision's pretrained ResNet-18, and the other using the MMClassification framework. Both approaches achieved over 90%.

validation accuracy, indicating successful training and effective generalization. This report summarizes the implementation details, training process, and comparison between the two models.

2 Related Work

Deep convolutional neural networks have significantly advanced image classification tasks in recent years. Classic models like AlexNet and VGG paved the way, but more recent architectures such as ResNet [**?**] introduced residual connections to address vanishing gradient problems, enabling deeper networks. Transfer learning using pretrained models on large-scale datasets such as ImageNet has also become a popular and effective approach. MMClassification, a part of the OpenMMLab project, provides a modular and extensible framework for training and evaluating classification models, which further simplifies experimentation and reproducibility.

3 Method

We explored two training pipelines:

3.1.PyTorch-based ResNet-18

We loaded a pretrained ResNet-18 model from torchvision.models and replaced the final fully connected layer to match the number of flower categories.

Data augmentation techniques such as random horizontal flips, rotations, resized crops, and normalization were applied to improve generalization.

The model was trained using the cross-entropy loss function, SGD optimizer, and a step learning rate scheduler.

, .

3.2.MMClassification-based Training

We migrated and customized a config file from MMClassification for the same flower dataset.

Modifications were made to adapt the model structure, dataset pipeline, and training schedule.

The training utilized a ResNet-18 backbone with batch normalization, and standard augmentation and optimization settings.

In both cases, training and validation datasets were split in an 8:2 ratio from the original dataset, and training was conducted for 25 epochs with a batch size of 32.

4 Experiments

Both training approaches yielded strong results:

4.1.PyTorch Model Result:

Best validation accuracy: 90.21%

Achieved after careful tuning of data augmentations and learning rate scheduling.

4.2.MMClassification Model Result:

Best validation accuracy: 90.14%

Required migrating configuration files and ensuring correct dataset formatting and preprocessing.

The following observations were made:

Both models achieved similar performance, demonstrating the robustness of the ResNet-18 architecture and the power of transfer learning.

MMClassification provided a more structured and modular training pipeline but required additional setup compared to the native PyTorch version.

Data augmentation played a crucial role in reaching high accuracy without overfitting.

References

None.