# ECE371 Neural Networks and Deep Learning Assignment 1

**Luo Huaiyu 22308130**
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
luohy67@mail2.sysu.edu.cn

**Abstract:** In this assignment, I implemented an image classification model using a ResNet-18 architecture pre-trained on ImageNet. The flower dataset was divided into training and validation sets with a split of 8-2 and data augmentation techniques were applied to improve generalization. The model was trained using Stochastic Gradient Descent with Momentum (SGDM) and a step learning rate scheduler, achieving a best validation accuracy of 92.98%. Through this experiment, I gained hands-on experience with transfer learning, data preprocessing, and model evaluation in PyTorch. The results demonstrate that SGDM, when paired with an appropriate learning rate schedule and augmentation strategy, can produce a highly accurate and robust classifier for multi-class image classification tasks.

**Keywords:** image classification, ResNet18, SGDM, data augmentation.

## 1 Introduction

Image classification is one of the fundamental tasks in computer vision. The goal of this assignment was to train a neural network to classify flower images into predefined categories. By utilizing transfer learning on a ResNet-18 model pre-trained on ImageNet, and applying data augmentation, the model was optimized to achieve high performance on a validation dataset. The final model reached a validation accuracy of 92.98%. This report details the implementation process, experimental setup, and insights drawn from model performance and behavior across training epochs.

## 2 Related Work

Convolutional Neural Networks (CNNs) have played a pivotal role in advancing image classification tasks over the past decades. One of the earliest milestones was LeNet-5 [1], which demonstrated the effectiveness of CNNs in recognizing handwritten digits. The field underwent a major shift with the introduction of AlexNet [2], which leveraged deep architectures and GPU acceleration to achieve breakthrough performance on the ImageNet dataset. Building on this, VGGNet [3] simplified the architecture by stacking uniform convolutional layers, showing that increased depth can lead to better accuracy. Around the same time, GoogLeNet [4] introduced the Inception module, combining filters of multiple sizes to capture information at different scales.

While these models increased network depth, training very deep networks remained challenging until the emergence of ResNet [5], which employed residual connections to enable the training of ultra-deep networks without suffering from vanishing gradients. Collectively, these works demonstrate a clear trend toward deeper, more efficient architectures with improved feature propagation mechanisms, which continue to influence modern CNN design in image classification tasks.

# 3 Method

## 3.1 Dataset and Augmentation

The flower image dataset was loaded using torchvision.datasets.ImageFolder with an 8/2 train-validation split. Data augmentation methods including random crops, rotations, and horizontal flips were applied to the training set. These augmentations help increase the diversity of the training data, which improves the model's generalization ability and reduces overfitting by simulating variations seen in real-world images.

## 3.2 Model Architecture

The ResNet-18 model was initialized with pre-trained weights, and its final fully connected layer was modified to output the number of flower classes. Adjusting the final layer ensures compatibility with the specific number of target classes in the flower classification task.

## 3.3 Loss Function and Optimization

The loss function was cross-entropy loss, which is standard for multi-class classification problems and measures the discrepancy between predicted and true class distributions.

The optimizer used was SGDM with a learning rate of 0.01, momentum of 0.9, and weight decay for regularization, which is a widely used and stable optimizer for CNNs, where momentum helps accelerate convergence and dampen oscillations. Weight decay prevents overfitting by penalizing large weights.

# 4 Experiments

This section presents the model's performance over 25 training epochs, evaluating accuracy and loss on both training and validation sets. The objective is to analyze learning behavior and generalization ability.

Using the Stochastic Gradient Descent with Momentum (SGDM) optimizer, the model achieved a peak validation accuracy of 92.98% at epoch 23. The following table and Figure summarizes the training and validation accuracy at key epochs (full log available in the output.txt):

Table 1: Training and Validation Accuracy over Epochs

| Epoch | Train Acc | Val Acc |
|---|---|---|
| 0 | 71.55% | 72.11% |
| 5 | 87.93% | 88.25% |
| 7 | 87.93% | 88.25% |
| 10 | 94.21% | 90.53% |
| 15 | 95.00% | 91.23% |
| 23 | 94.38% | **92.98%** |

Early Phase (Epochs 0–5): Rapid improvements in accuracy were observed due to transfer learning from a pretrained model and the application of effective data augmentation strategies.

Middle Phase (Epochs 6–12): The learning rate was reduced from 0.01 to 0.001 after epoch 6 by a step scheduler, allowing the model to shift from coarse tuning to fine-tuning. While training accuracy continued to increase, validation accuracy experienced minor fluctuations, suggesting the onset of slight overfitting.

Late Phase (Epochs 13–24): With the learning rate further decaying to 0.0001 and 0.00001, the model entered a stable convergence phase. Validation accuracy remained steady between 91–93%, and validation loss generally decreased, indicating improved generalization.

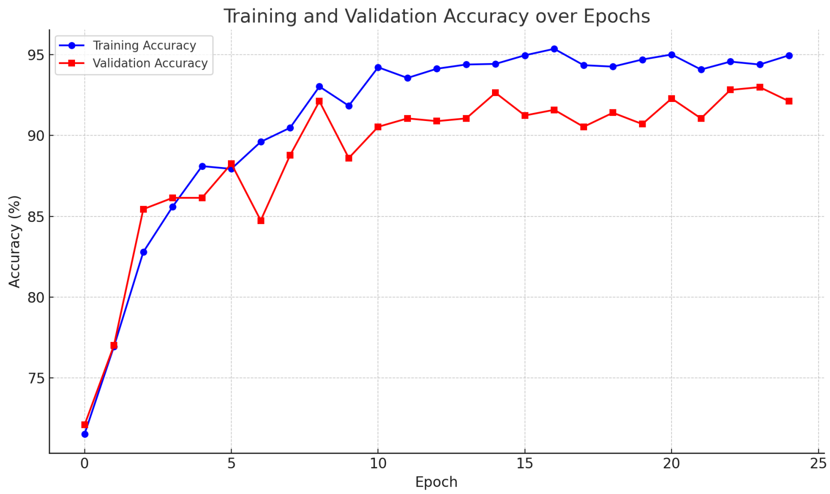


Figure 1: Training and Validation Accuracy over Epochs

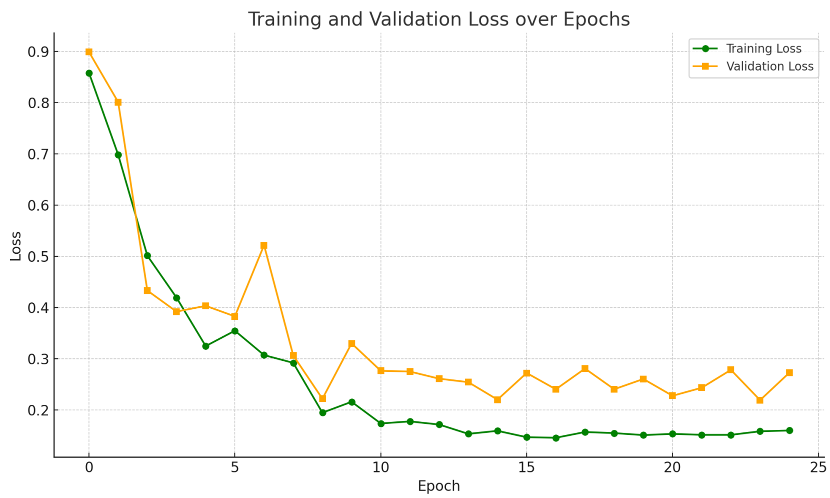


Figure 2: Training and Validation Loss over Epochs

In addition to SGDM, the Adam optimizer was also tested. While Adam typically enables faster convergence in early epochs due to its adaptive learning rates, it underperformed in this task: Although training remained stable, validation accuracy never exceeded 90%, and the best validation accuracy with Adam was around 88–89%. Adam tended to converge to local optima more quickly, resulting in poorer generalization compared to SGDM. Based on these observations, SGDM was the better choice for this classification task in terms of achieving higher validation accuracy and better generalization.

Occasional increases in validation loss during certain epochs suggest that the model's robustness could be further improved on specific samples. While training accuracy approached saturation, the gap to validation accuracy suggests remaining room for improvement. Future enhancements may include experimenting with deeper architectures, or incorporating advanced augmentation techniques.

## 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.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, volume 25, pages 1097–1105, 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.

[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.