Deep Learning Report: Image Classification using CNNs

Submitted in fulfillment of the project component for the course
CS F425 Deep Learning

Birla Institute of Technology and Science, Pilani

Group Members: Anirudh Anand, Dhruv Nair, Dhruv Ravi Krishnan

Date: 17/11/2024

Flower Classification Model Report

Contents

1 Introduction

This report outlines the implementation, training, and evaluation of a flower classification model built using a pre-trained ResNet-18 architecture. The model was fine-tuned on a flower dataset to classify multiple categories. The training utilized *Stochastic Gradient Descent (SGD) with momentum* as the optimizer and *Cross-Entropy Loss* as the loss function.

Additionally, a technical overview of the ResNet-18 architecture is provided, including its advantages and reasons for its selection in this project. A brief overview of YOLOv8 is also included, showcasing its relevance to computer vision tasks like object detection.

2 ResNet-18 Architecture

2.1 Overview

ResNet-18 is a convolutional neural network (CNN) introduced in the seminal paper ”Deep Residual Learning for Image Recognition” by He et al. (2015). The model addresses critical challenges in training deep neural networks, such as vanishing gradients and overfitting, by introducing **residual connections**. These connections allow the network to learn identity mappings, ensuring stable training and better performance.

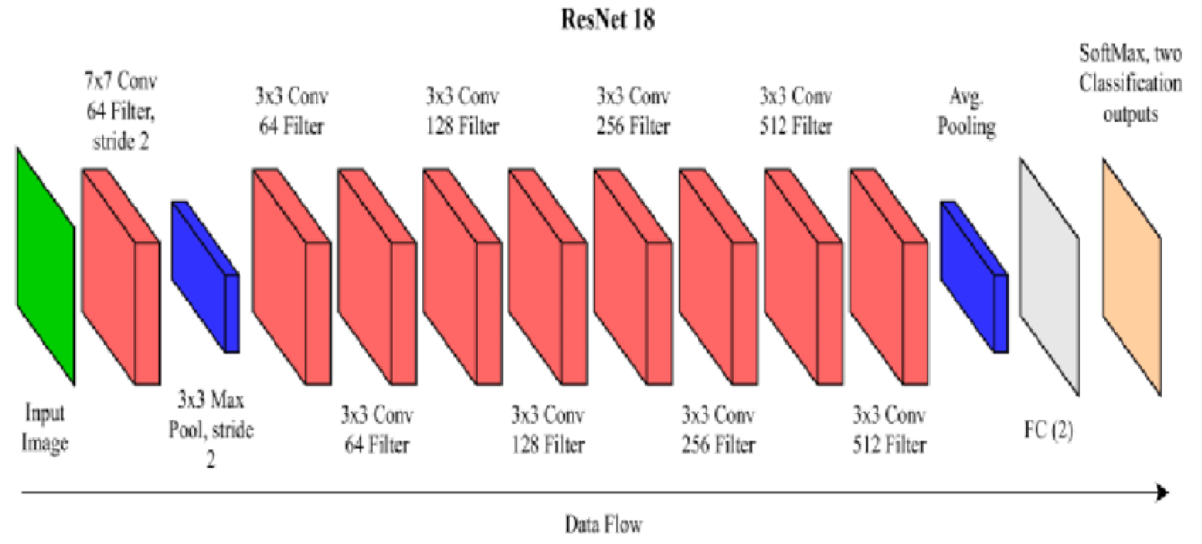

Figure 1: Learning Curve for Training and Validation Accuracy.

2.2 Layer-by-Layer Breakdown

The ResNet-18 architecture consists of the following components:

1. **Input Layer:** Images are resized to 256 × 256 × 3 and normalized.
2. **Initial Convolution and Pooling:**
   - A 7 × 7 convolution with 64 filters, stride 2, followed by Batch Normalization and ReLU activation.
   - A max-pooling layer reduces the spatial dimensions by half, improving computational efficiency.
3. **Residual Blocks:**

- **Block 1:** Two 3 × 3 convolutional layers with 64 filters.
- **Block 2:** Two 3 × 3 convolutional layers with 128 filters (stride 2 for downsampling).
- **Block 3:** Two 3 × 3 convolutional layers with 256 filters.
- **Block 4:** Two 3 × 3 convolutional layers with 512 filters.

4. **Global Average Pooling:** Reduces the feature map to a single value per channel.

5. **Fully Connected Layer:** Outputs class probabilities using a softmax activation function.

2.3 Advantages of ResNet-18

- **Residual Learning:** Residual connections mitigate the degradation problem in deep networks by enabling the learning of identity mappings.

- **Scalability:** The architecture can be extended to deeper versions like ResNet-50 or ResNet-101 without significantly affecting performance.

- **Transfer Learning:** Pre-trained models on large datasets (e.g., ImageNet) allow fine-tuning for domain-specific tasks, reducing training time and computational resources.

3 Optimizer and Loss Function Selection

3.1 Stochastic Gradient Descent with Momentum

Stochastic Gradient Descent (SGD) is widely used for optimization in deep learning due to its computational efficiency and simplicity. We chose SGD with momentum for the following reasons:

- **Momentum:** Adds a fraction of the previous update to the current gradient, helping the optimizer navigate through local minima and saddle points effectively.

- **Stable Convergence:** Momentum smooths the oscillations along the optimization path, especially in scenarios with high-dimensional datasets.

- **Hyperparameter Control:** The learning rate and momentum coefficient (0.9 in this case) were fine-tuned to balance convergence speed and accuracy.

3.2 Cross-Entropy Loss

Cross-Entropy Loss was chosen as it is well-suited for multi-class classification problems. It quantifies the distance between the predicted probability distribution and the true label distribution:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}),$$

where y_{ij} is the true label and $\hat{y}_{ij}$ is the predicted probability for class j. This function:

- **Encourages Correct Predictions:** Penalizes incorrect classifications proportionally to the confidence of the incorrect prediction.

- **Smooth Optimization:** Provides a differentiable, convex function, ensuring stability in gradient-based methods.

4 Training Details

4.1 Dataset and Preprocessing

The dataset consisted of images of flowers across multiple classes. To prevent overfitting and improve generalization, the following preprocessing steps were applied:

- **Data Augmentation:** Techniques such as random rotations, horizontal flips, and scaling.
- **Normalization:** Image pixel values were scaled to a range of [0, 1] and normalized using the mean and standard deviation of the ImageNet dataset.

4.1.1 Impact of Wavelet Transforms on Pre-trained Models

At first we attempted to use the wavelet transform before passing it to the model, but we found that the accuracy took a significant decrease with it. We were only able to achieve a validation accuracy of **80%**. Pre-trained models like ResNet18 are trained on ImageNet, which consists of raw RGB images. These models learn to extract hierarchical features directly from pixel intensities, edges, textures, and patterns in the spatial domain. Wavelet transforms convert images from the spatial domain to a combined spatial-frequency domain, altering the pixel-level structures the model is used to interpreting. This disrupts the alignment between the learned weights and the input data.

- Loss of Information Wavelet transforms often break an image into frequency components, such as low and high frequencies. This process can inadvertently discard important spatial details, especially if only a subset of the transformed data (e.g., low-frequency components) is used as input to the model. High-frequency information, such as edges or fine details, which might be critical for classification, could be lost or distorted.
- Redundancy in Features Models like ResNet already incorporate mechanisms (e.g., convolutional filters) to extract relevant spatial and frequency features. Preprocessing with a wavelet transform might introduce redundant or noisy features, confusing the model's existing feature extraction capabilities.
- Change in Input Distribution The pre-trained model assumes a specific input distribution based on its training set. Wavelet-transformed images have a different distribution due to their altered frequency content, which can lead to a significant domain shift, reducing performance.
- Over-processing Adding wavelet transformations could be an unnecessary step if the raw images already contain sufficient information for classification. Over-processing can complicate the data unnecessarily and lead to the loss of key patterns the model depends on.

4.2 Training Configuration

- **Batch Size:** 16
- **Learning Rate:** 0.001, reduced dynamically based on validation performance.
- **Epochs:** 10, chosen to balance training time and overfitting.
- **Momentum for SGD**: 0.9

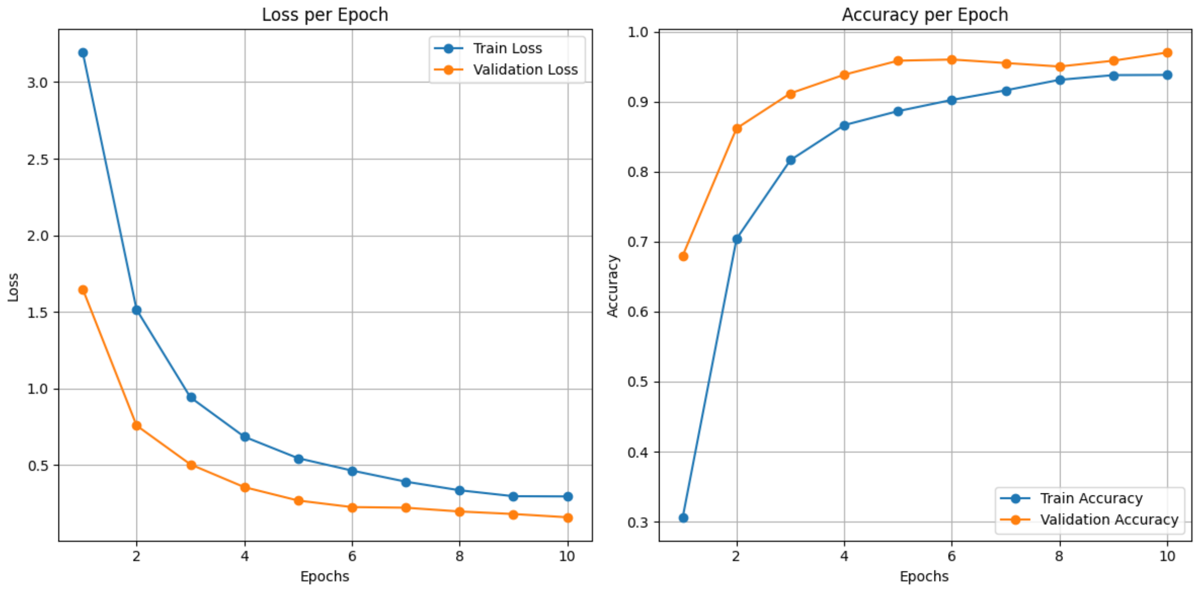

Figure 2: Learning Curve for Training and Validation Accuracy.

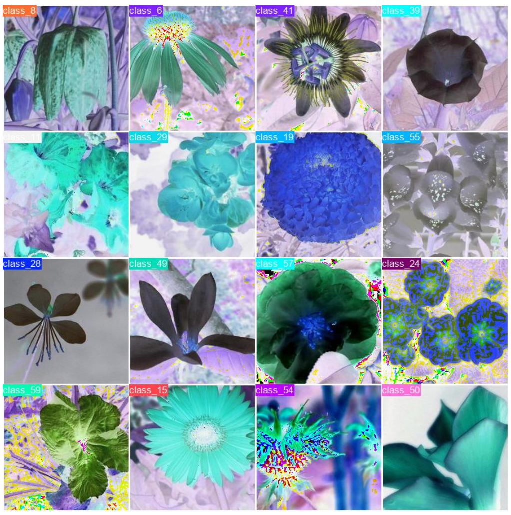

Figure 3: Features extracted.

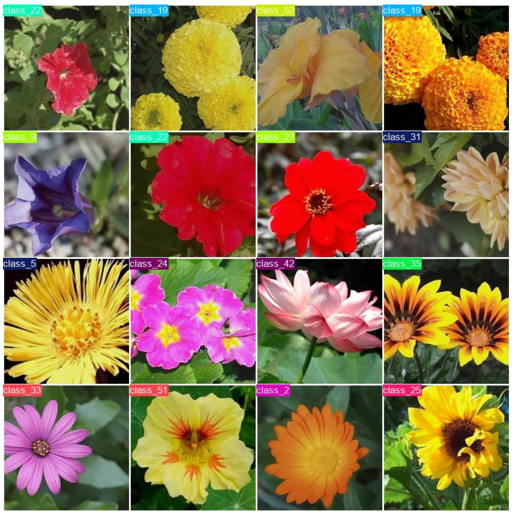

Figure 4: Final prediction.

4.3 Results

The model achieved an accuracy of **97%** on the test set, indicating robust performance. The learning curve (Figure 2) demonstrates steady convergence.

5 YOLOv8: A Brief Overview

YOLOv8 is an advanced model for object detection, segmentation, and keypoint detection. It is built on the foundational YOLO architecture and introduces several improvements:

- **Dynamic Network Design:** Adjusts computational effort based on object size and complexity.
- **Enhanced Backbone:** Utilizes CSPDarknet53, optimized for feature extraction.

- **Application Scope:** Suitable for tasks requiring real-time inference, such as autonomous vehicles and surveillance.

We attempted to use YOLOv8 as well and got an accuracy of **96%** which is slightly lower than ResNet.

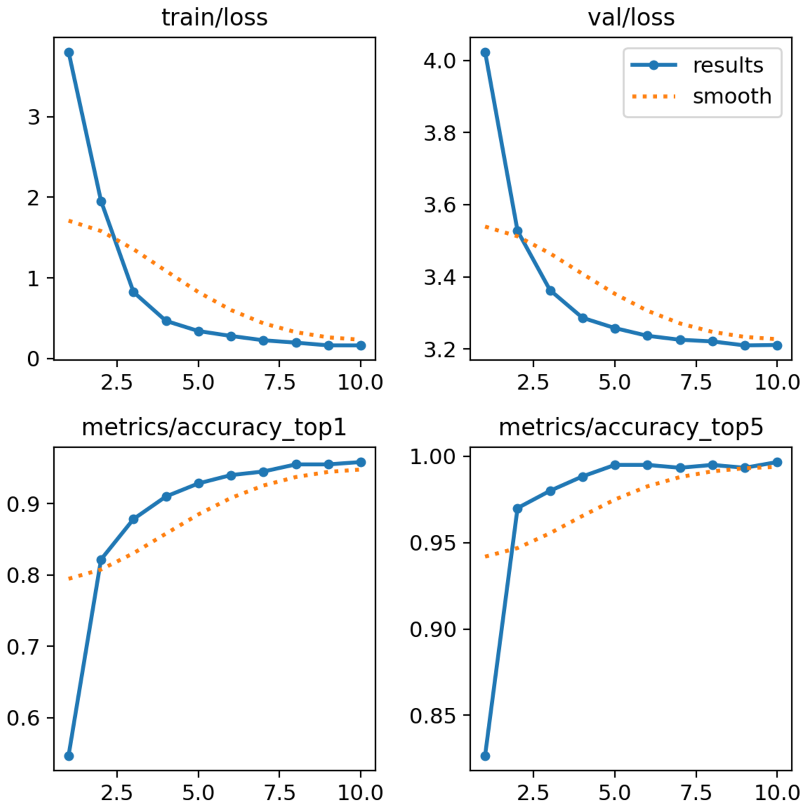

Figure 5: Learning Curve for Training and Validation Accuracy for yolo.

6 Model Performance Comparison: CNN vs MLP

In this section, we compare the performance of a Convolutional Neural Network (CNN) and a Multilayer Perceptron (MLP) on the image classification task. The CNN model achieved significantly higher accuracy compared to the MLP, as shown in the following plots.

The CNN model achieved a best validation accuracy of 97.0%, while the MLP achieved a best test accuracy of 37%. These results clearly demonstrate the advantage of CNNs for image classification tasks, especially with larger and more complex datasets.

6.1 CNN vs MLP: Pros and Cons

Below are the key advantages and disadvantages of using a Convolutional Neural Network (CNN) over a Multilayer Perceptron (MLP) for image classification:

6.1.1 Advantages of CNN over MLP

- **Better at Capturing Spatial Hierarchies:** CNNs are specifically designed to capture spatial dependencies (like patterns or objects) in images. They learn local patterns and combine them into higher-level features, making them ideal for image classification tasks.

- **Parameter Sharing:** CNNs use shared weights (filters), reducing the number of parameters and improving efficiency.
- **Translation Invariance:** CNNs are robust to translations of objects in an image. This means the model can still recognize objects even if they move slightly within the image.
- **Automatic Feature Extraction:** Unlike MLPs, which require manually designed features, CNNs automatically learn hierarchical features like edges, textures, and complex shapes that are crucial for image classification.

6.1.2 Disadvantages of CNN over MLP

- **More Computationally Expensive:** CNNs are generally more computationally intensive and require more time for training, especially with large networks or large datasets.
- **Requires Larger Datasets:** CNNs need larger datasets to perform well, as they have many parameters to train. For small datasets, MLPs may sometimes perform better due to fewer parameters.
- **Complexity:** CNNs are more complex to implement and often require specialized hardware (such as GPUs) to train efficiently.

7 Conclusion

This project demonstrates the efficacy of ResNet-18 for flower classification, achieving high accuracy with minimal training overhead. By leveraging pre-trained models and robust optimization techniques, we achieved reasonably good performance. In this image classification task, the CNN model outperforms the MLP by a significant margin. The CNN's ability to learn spatial hierarchies, local patterns, and features in images gives it a clear advantage over the MLP, which struggles to capture these spatial relationships.