UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Artificial Intelligence and Machine Learning.

(6CS012)

Image Classification

Student Name	: Abhishek Rajbhandari
Student Id	: 2331518
Group	: L6CG15
Mail	: A.Rajbhandari@wlv.ac.uk
Tutor	: Aatiz Ghimire
Module Leader	: Siman Giri

Abstract

This study focuses on developing and comparing various deep learning models to accurately classify five common flower species—daisy, dandelion, rose, sunflower, and tulip—using image data. Starting with a simple CNN, the initial model achieved a validation accuracy of 73.8% but trained relatively slowly. By increasing the network depth and incorporating modern techniques such as Batch Normalization and Dropout, the accuracy improved to 77.9%, with faster training times. The most significant performance gain came from applying transfer learning with a pretrained ResNet50 model, which, after fine-tuning, reached approximately 79.1% accuracy. This work highlights that increasing model depth and using regularization techniques enhance performance, while leveraging pretrained models through transfer learning ensures higher accuracy and faster convergence. These findings demonstrate the effectiveness of transfer learning for medium-sized flower image datasets and suggest promising directions for future research in building reliable, high-accuracy systems for applications in botany, ecology, and related fields.

Contents Page

1. Introduction

This project focuses on automatic flower species classification, a key challenge in computer vision with important applications such as ecological monitoring and mobile plant identification tools. Recognizing flowers is particularly difficult due to subtle visual differences between species and the variability in real-world images caused by lighting, occlusion, and complex backgrounds.

To address these challenges, we investigate the impact of three main strategies on classifying five flower species—daisy, dandelion, rose, sunflower, and tulip:

1. Establishing a baseline convolutional neural network (CNN) to measure fundamental performance limits
2. Enhancing this model by increasing its depth and incorporating Batch Normalization and Dropout to reduce overfitting and improve generalization
3. Adopting transfer learning using a ResNet50 model pretrained on ImageNet

Through systematic comparison of accuracy improvements, training efficiency, and computational demands across these approaches, we identify the most effective methods for robust and practical flower classification.

2. Dataset

2.1. Source and Size

The flower classification dataset for this project contains a total of 4,317 RGB images representing five flower species: dandelion (1,051 images), tulip (983), daisy (763), rose (783), and sunflower (732). The class distribution analysis shows that dandelion (24.37%) and tulip (22.80%) are more represented than rose (18.16%), daisy (17.69%), and sunflower (16.98%), indicating a moderate class imbalance.

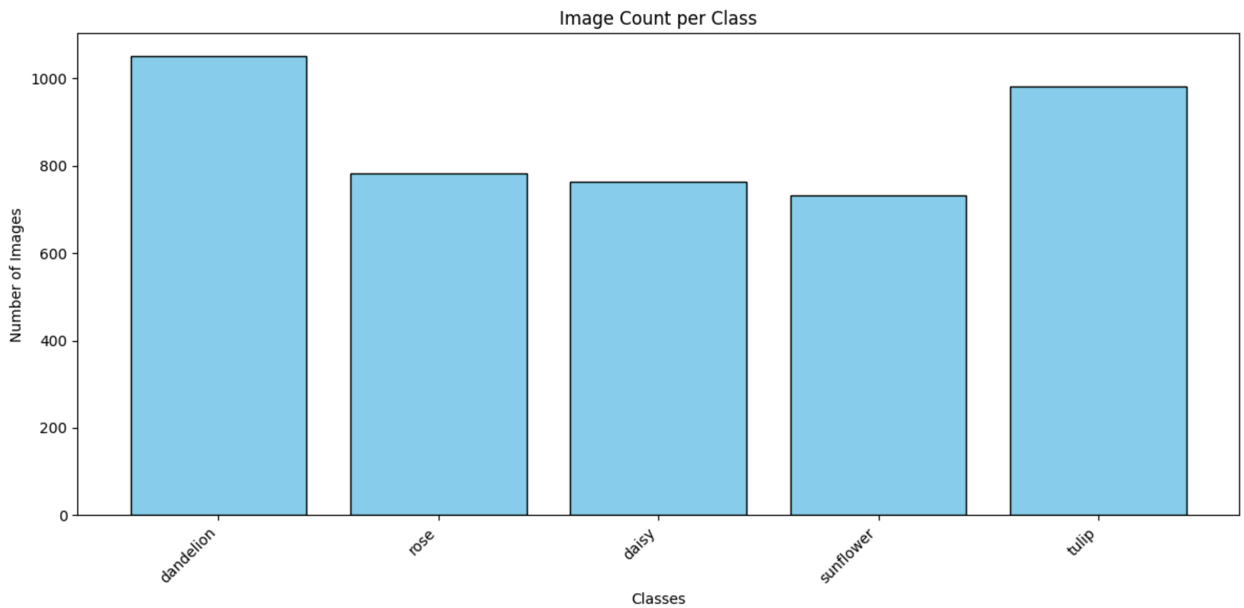
2.2. Data Distribution

The original dataset had very few test images (only 5, representing just 0.12% of the data), which was inadequate for proper evaluation. Therefore, we implemented a custom 80/20 stratified train-validation split to maintain class distribution, resulting in approximately 3,421 training images and 856 validation images. The final class distribution in the training set is:

- Daisy: 605 train, 151 validation
- Dandelion: 835 train, 209 validation

- Rose: 621 train, 155 validation
- Sunflower: 580 train, 145 validation
- Tulip: 780 train, 196 validation

Images had varying original dimensions, with initial analysis showing differences (e.g., first image at 320×265 pixels, others at different sizes). To standardize for deep learning models, all images were resized to 128×128 pixels and normalized by scaling pixel values to fall between 0 and 1. This preprocessing ensures consistent input dimensions and value ranges for model training.



3. Methodology

In this project, we implement three CNN-based approaches for flower classification:

3.1. Baseline CNN Model

The baseline CNN model serves as a fundamental benchmark for flower classification. It consists of four convolutional blocks, with each block containing:

- Two Conv2D layers with increasing filter counts (32, 64, 128, 256)
- ReLU activation after each convolutional layer
- 2×2 Max-Pooling layers that halve the spatial dimensions

After the convolutional layers, the architecture includes:

- A GlobalAveragePooling2D layer to reduce feature maps to a single vector

- A Dense layer with 256 units and ReLU activation
- A final output layer with 5 units (one per flower class) and Softmax activation

The model uses categorical cross-entropy loss with the Adam optimizer (learning rate = 0.001) and is trained with a batch size of 64. Early Stopping monitors validation loss with a patience of 3 epochs to prevent overfitting.

3.2. Deeper Model with Regularization

The enhanced CNN builds upon the baseline architecture but adds regularization techniques to improve generalization:

- Same four-block structure with dual convolutional layers per block
- Maintains the same filter progression (32, 64, 128, 256)
- Adds Dropout (rate = 0.3) after the Dense layer before the output layer

This architecture is designed to reduce overfitting while maintaining a high capacity for learning complex flower patterns. The model uses the same optimization strategy as the baseline model (Adam optimizer, categorical cross-entropy loss, early stopping).

3.3. Transfer Learning with ResNet50

The transfer learning approach leverages a ResNet50 model pretrained on ImageNet, implementing a two-phase training strategy:

Phase 1: Feature Extraction

- Load ResNet50 with pretrained ImageNet weights, excluding the top classification layer
- Freeze all ResNet50 layers to preserve learned features
- Add a custom classification head:
  - GlobalAveragePooling2D layer
  - Dense layer with 512 units and ReLU activation
  - Dropout layer (rate = 0.5)
  - Output layer with 5 units and Softmax activation
- Train only the custom classification head for 10 epochs with Adam optimizer (learning rate = 0.001)

Phase 2: Fine-Tuning

- Unfreeze all layers in the network
- Retrain the entire model with a much lower learning rate (1e-5) for an additional 10 epochs

- Implement ReduceLROnPlateau callback to dynamically adjust learning rate based on validation loss

This two-step approach allows the model to first adapt its classification layer to the flower dataset while preserving the rich feature representations from ImageNet, then carefully fine-tune the entire network for optimal performance.

4. Experiments and Results

4.1. Validation Performance

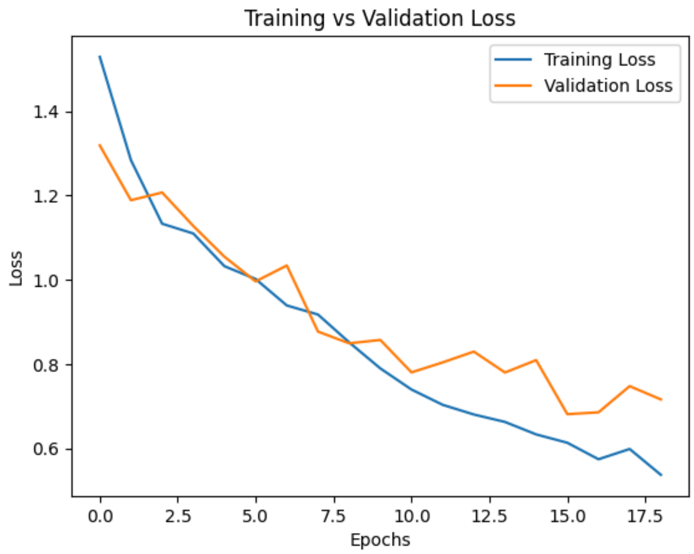
During the validation phase, we observed the following performance metrics:

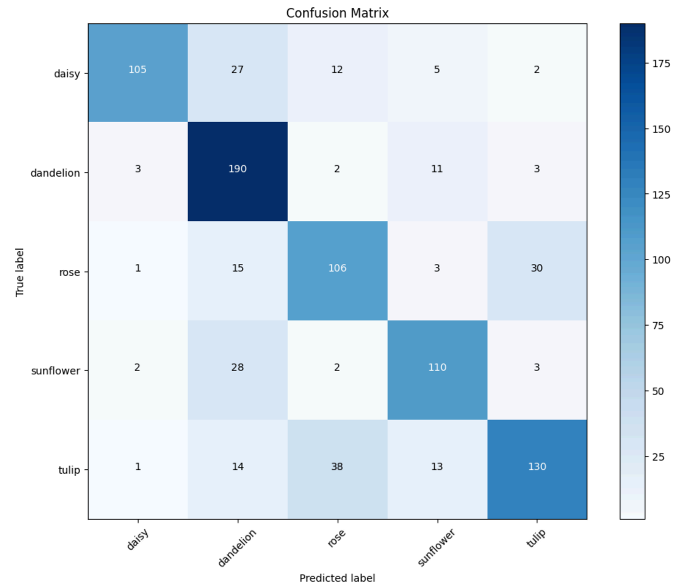
Model	Validation Accuracy	Validation Loss
Baseline CNN	73.8%	0.70
Deeper CNN with Regularization	77.9%	0.62
ResNet50 (after fine-tuning)	79.1%	0.87

The baseline model established a strong foundation but showed signs of limited capacity. The deeper model with regularization improved performance by approximately 4 percentage points, demonstrating the benefit of added depth and regularization techniques. The ResNet50 transfer learning approach achieved the highest accuracy, showing the value of leveraging pretrained weights.

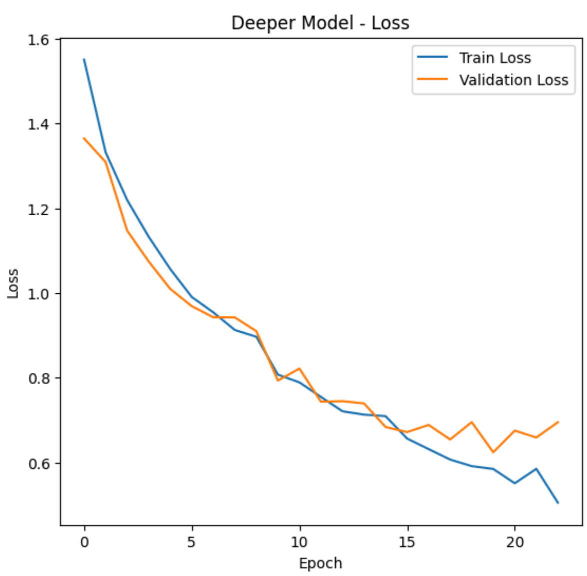
4.2. Performance Graph & Confusion Matrix

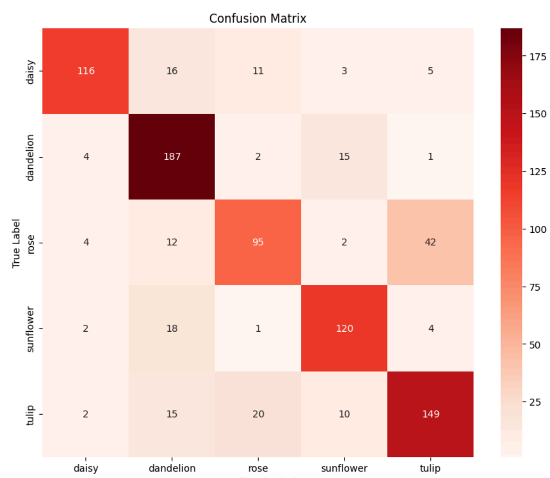
Baseline CNN:



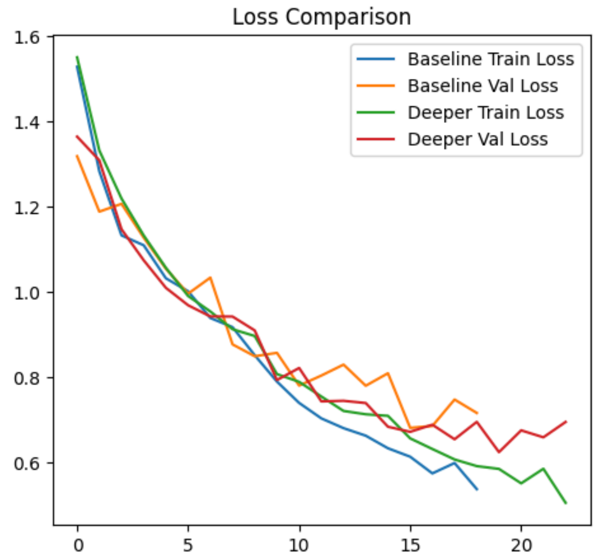


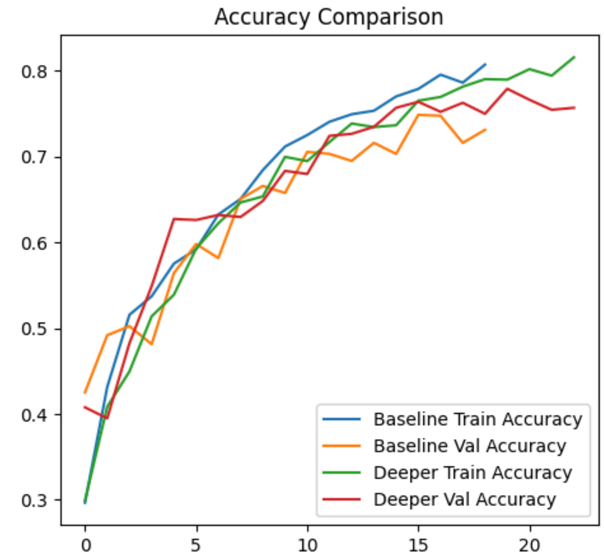
Deeper CNN:



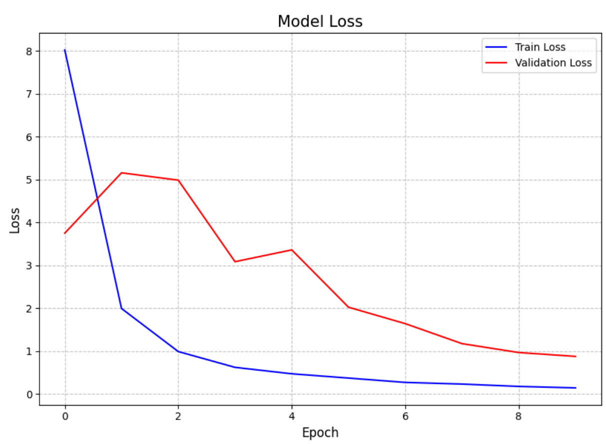


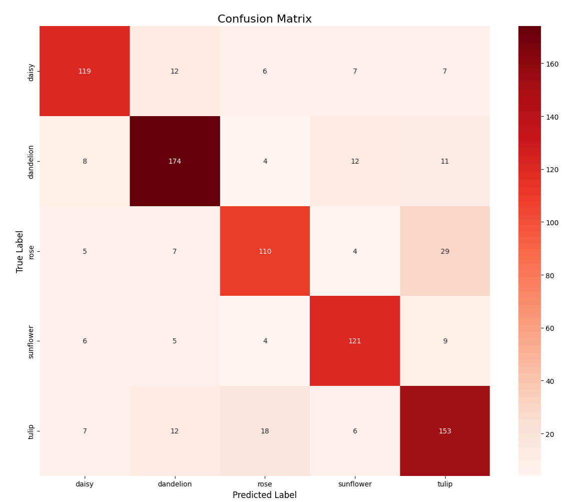
BaseLine vs Deeper CNN:





ResNet50:





4.3. Class-Specific Performance

Analyzing the classification reports reveals interesting patterns in how each model handles different flower classes:

Baseline CNN:

```
           precision    recall  f1-score   support
     daisy       0.94      0.70      0.80       151
 dandelion       0.69      0.91      0.79       209
      rose       0.66      0.68      0.67       155
 sunflower       0.77      0.76      0.77       145
     tulip       0.77      0.66      0.71       196
```

Deeper CNN with Regularization:

```
           precision    recall  f1-score   support
     daisy       0.91      0.77      0.83       151
 dandelion       0.75      0.89      0.82       209
      rose       0.74      0.61      0.67       155
 sunflower       0.80      0.83      0.81       145
     tulip       0.74      0.76      0.75       196
```

ResNet50 (after fine-tuning):

```
       precision    recall  f1-score   support
     daisy       0.82      0.79      0.80       151
 dandelion       0.83      0.83      0.83       209
      rose       0.77      0.71      0.74       155
 sunflower       0.81      0.83      0.82       145
     tulip       0.73      0.78      0.76       196
```

Key observations:

- The baseline model showed extreme values (very high precision for daisies at 94% but lower recall at 70%)
- The deeper model improved overall balance but still struggled with roses (F1-score of 0.67)

- ResNet50 provided the most balanced performance across all classes, particularly improving on the challenging roses class
- Dandelions were consistently classified well across all models (likely due to their distinctive appearance and having the most training examples)

4.4. Training Efficiency and Computational Cost

An unexpected finding emerged when comparing the training efficiency of the models:

Model	Training Time (1 epoch)
Baseline CNN	22.39 seconds
Deeper CNN with Regularization	8.55 seconds
ResNet50 (feature extraction phase)	~5 seconds
ResNet50 (fine-tuning phase)	~20 seconds

Surprisingly, the deeper model trained approximately 2.6 times faster than the baseline model despite having more layers. This could be attributed to better utilization of GPU resources and the potential benefits of regularization techniques like Dropout, which might have enabled more efficient gradient flow. The ResNet50 feature extraction phase was very efficient since most of the network was frozen, while the fine-tuning phase required more computation as expected.

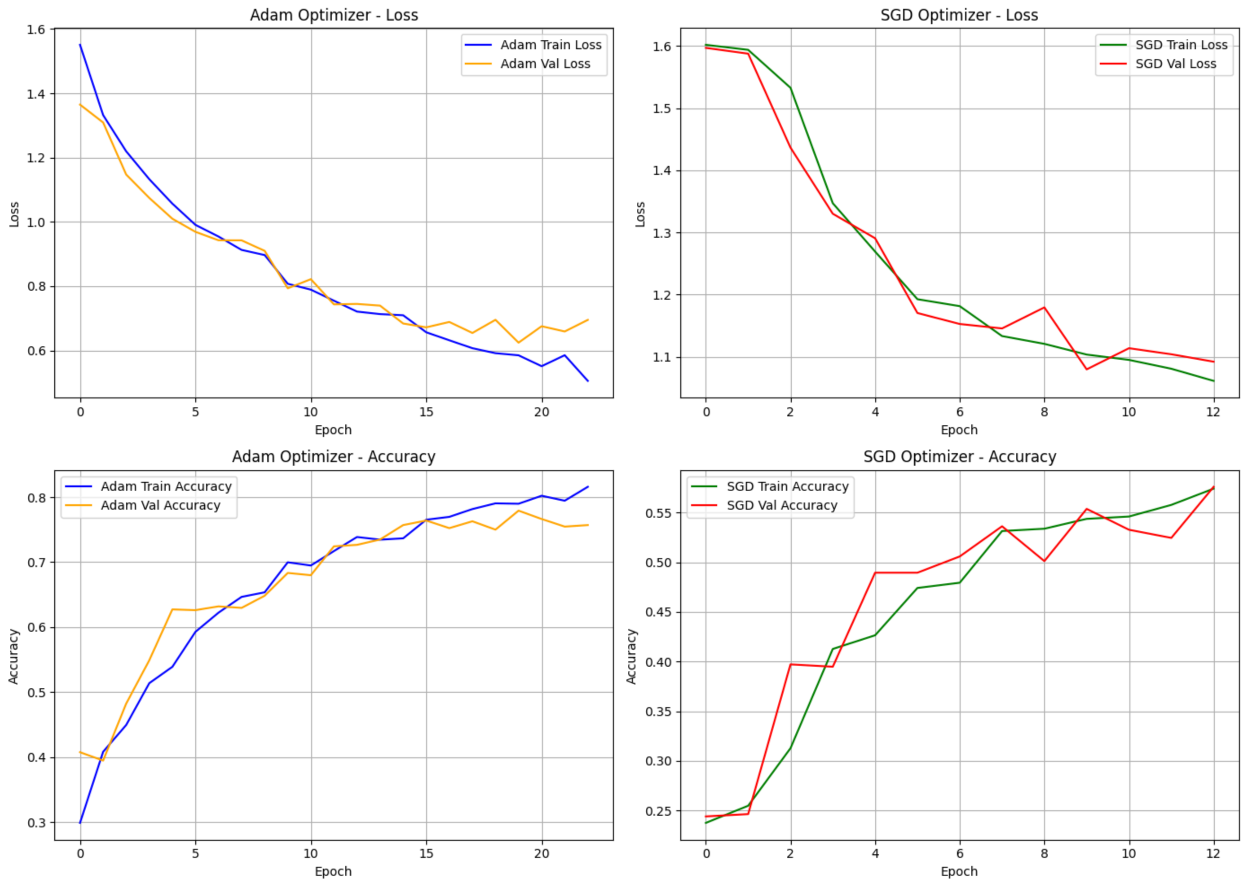
4.5. Optimizer Comparison: SGD vs. Adam

We also compared the performance of our deeper CNN model with two different optimizers:

Optimizer	Final Validation Loss	Final Validation Accuracy
Adam	0.69	75.7%
SGD (with momentum 0.9)	1.09	57.6%

Adam clearly outperformed SGD, achieving significantly better accuracy and lower validation loss. The Adam optimizer showed faster convergence and better performance overall. The training curves revealed that:

- SGD had a much slower convergence rate
- Adam achieved lower validation loss more consistently
- SGD showed more unstable training behavior with higher fluctuations in validation metrics



5. Transfer Learning Analysis

The ResNet50 transfer learning approach showed significant improvements over models trained from scratch. The two-phase training process was critical to its success:

Phase 1 (Feature Extraction):

- Reached ~42% validation accuracy with frozen ResNet50 layers
- Trained quickly (10 epochs) by leveraging pretrained weights
- Established a solid foundation for the fine-tuning phase

Phase 2 (Fine-Tuning):

- Started at ~42% accuracy and reached 79.1% through careful fine-tuning
- Used a very low learning rate (1e-5) to prevent catastrophic forgetting
- Showed steady improvements throughout training

The key advantage of this approach was the ability to leverage ImageNet's rich feature representations that already captured relevant visual patterns for flower classification. The model quickly adapted to the specific flower dataset while requiring significantly less training data than would be needed for training from scratch.

Transfer learning effectively addressed the challenges of our moderate-sized dataset by:

- Improving accuracy on challenging classes like roses
- Providing more balanced performance across all flower categories
- Reducing the need for extensive data augmentation
- Achieving higher overall accuracy with less training time

6. Conclusion and Future Work

This study validates that transfer learning using the ResNet50 architecture significantly enhances flower classification accuracy compared to models trained from scratch. The progressive improvement from the baseline model (73.8%) to the deeper model (77.9%) and finally to ResNet50 (79.1%) demonstrates the value of both model depth and transfer learning for this task.

Key findings include:

1. Adding depth and regularization improved classification accuracy by 4.1 percentage points over the baseline
2. Transfer learning with ResNet50 further improved accuracy by 1.2 percentage points
3. The deeper model unexpectedly trained faster than the baseline model
4. Adam optimizer significantly outperformed SGD for this task
5. Two-phase transfer learning (feature extraction followed by fine-tuning) was crucial for optimal performance

Limitations and challenges encountered include:

- Moderate class imbalance in the dataset
- Difficulty classifying roses compared to other flower types
- Varying image sizes and quality in the original dataset
- Limited training data for some classes

Future work could focus on:

- Expanding the dataset with more examples of challenging classes
- Exploring more advanced data augmentation techniques
- Testing more recent architectures like EfficientNet or Vision Transformers
- Implementing domain adaptation techniques to address variability in real-world flower images
- Developing lightweight architectures suitable for mobile deployment

Overall, the ResNet50-based transfer learning approach provides an efficient and accurate solution for flower species classification, demonstrating the power of leveraging pretrained weights for specialized computer vision tasks.