Flower Image Classification with Transfer Learning

A deep learning project that classifies flower images into 14 categories using transfer learning with PyTorch.

Project Overview

This project applies deep learning and transfer learning techniques to classify images of flowers into 14 distinct categories.

The goal was to develop a high-performing image classifier that generalizes well to unseen data using pre-trained CNN models, PyTorch, and efficient training strategies.

Dataset

The dataset was sourced from Kaggle, and includes:

- 14 flower classes (e.g., rose, tulip, dandelion, iris, sunflower, etc.)
- 10,913 training images, 2729 validation images and 98 manually reserved test images
- Images were uniformly resized to 224×224 for consistency with pretrained model input requirements.

Class distributions were visualized and balanced using stratified sampling to ensure fair model evaluation.

Project Approach

This project followed a structured machine learning pipeline:

1

Data Preprocessing

- Resize to 224×224
- Augment training data with flips, rotations, and color jittering
- Normalize using ImageNet mean and standard deviation

2

Train-Validation Split

- 80/20 split using StratifiedShuffleSplit for class balance
- PyTorch Subset was used to create separate datasets for each split

3

Modeling Strategy

- Transfer learning with frozen base layers and custom classifiers
- Three models compared:
  - EfficientNet-B0
  - ResNet34
  - DenseNet121
- Only selected layers were unfrozen for fine-tuning to improve performance and reduce overfitting.

4

Training Setup

- Used CrossEntropyLoss and Adam optimizer
- Tracked performance with loss/accuracy plots per epoch
- Saved trained models using custom save_checkpoint() utility

Results

VALIDATION PERFORMANCE

Across all three architectures, DenseNet121 yielded the best validation results:

DenseNet121

- Final Validation Accuracy: 95.46%
- Train Accuracy: 98.08%
- Smooth convergence with minimal overfitting

ResNet34

- Final Validation Accuracy: 93.92%
- Train Accuracy: 96.11%

EfficientNet-B0

- Final Validation Accuracy: 89.85%
- Train Accuracy: 86.14%

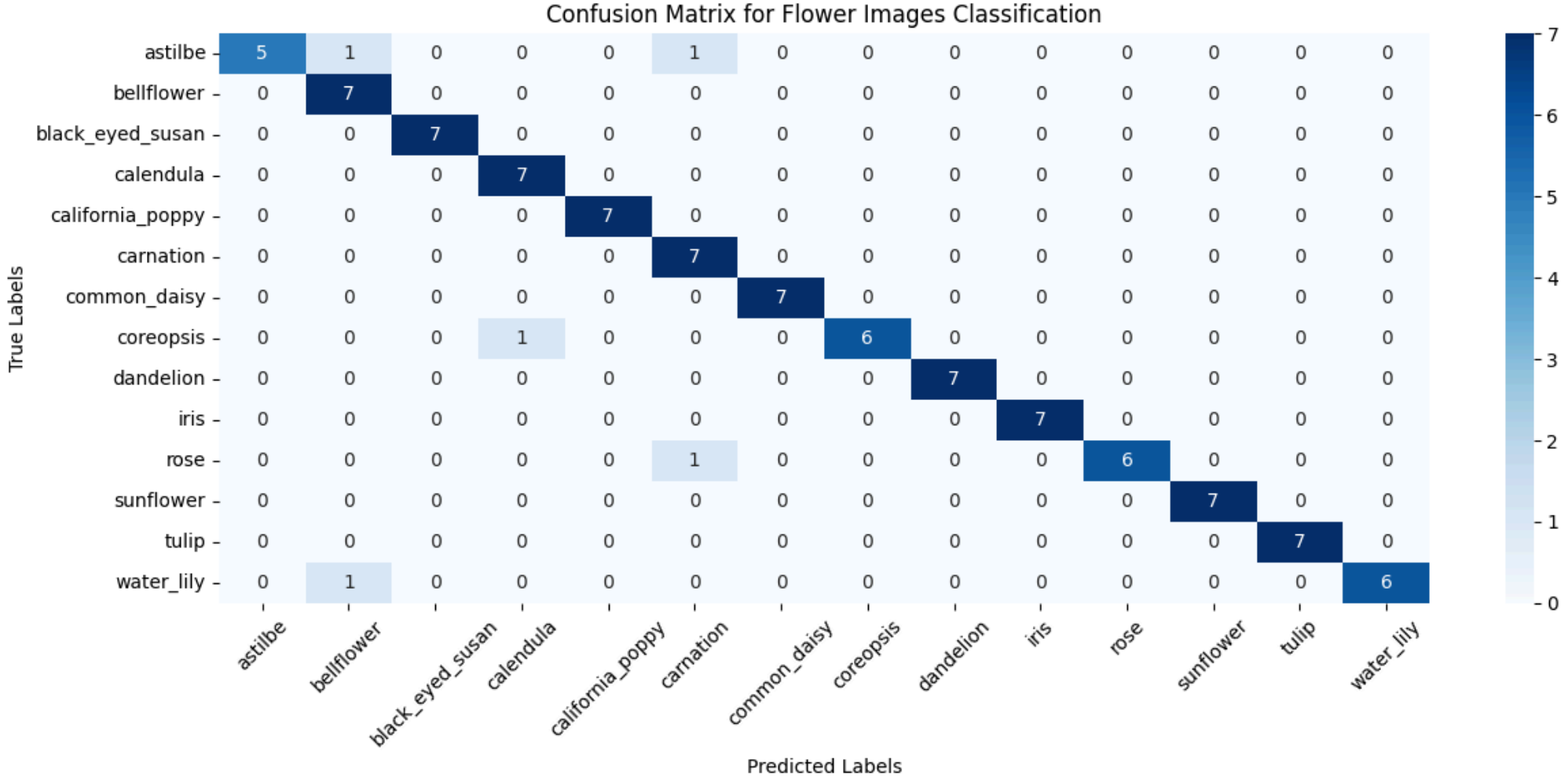
Final Model Evaluation

The best-performing DenseNet121 model was evaluated on a completely unseen test set of 98 flower images (7 images per class).

Key results:

- Test Accuracy: 94.90%
- Precision, Recall, F1-score (Macro Avg): ~95%
- Confusion Matrix: Strong diagonal dominance and minimal misclassification

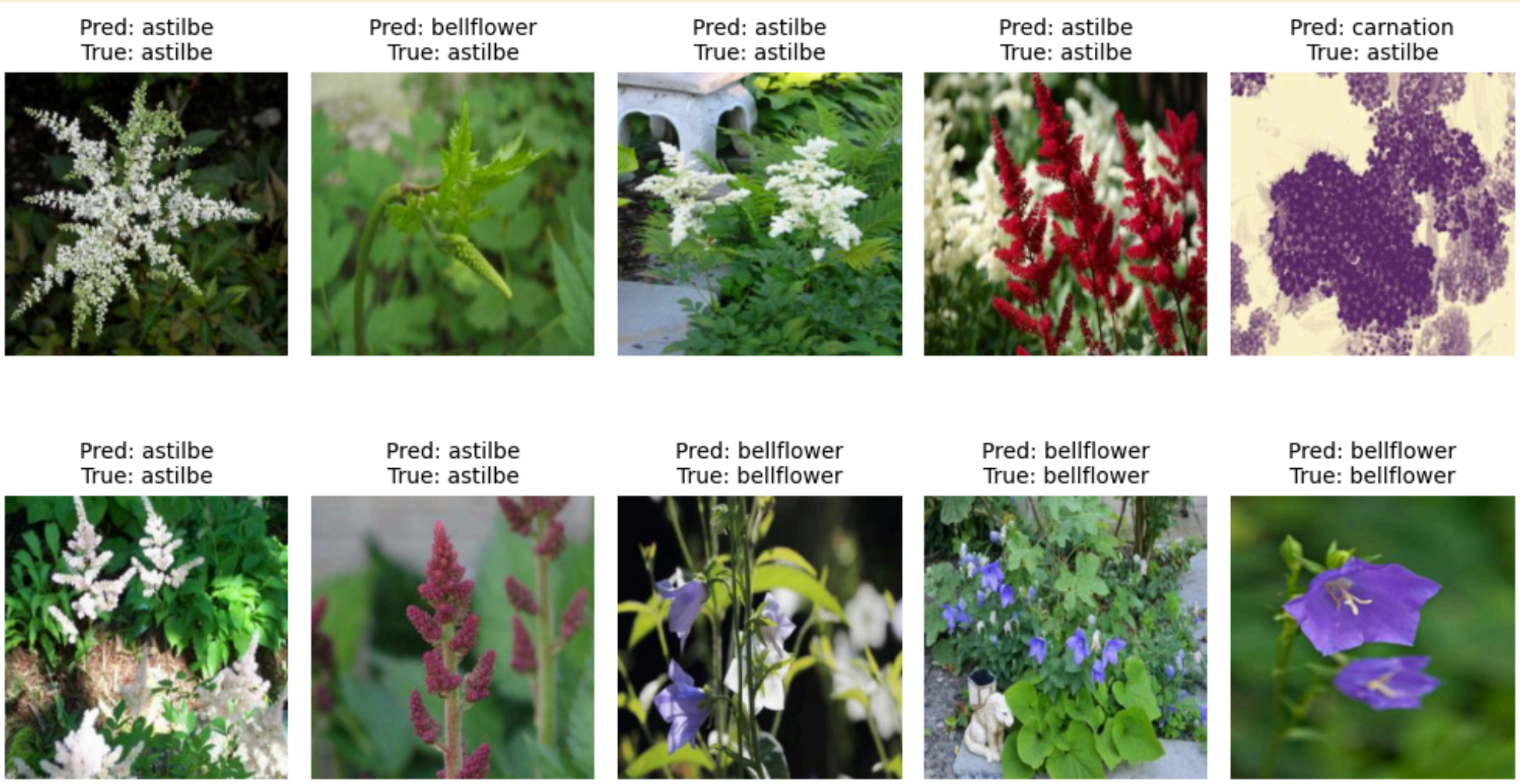
The evaluation script included both classification_report and confusion_matrix, ensuring reliable performance insight.



Sample Predictions

To visually evaluate the model's performance, a few sample test images are displayed below alongside their predicted and actual class labels.

These examples provide a quick, intuitive sense of how well the model generalizes to unseen data.



Key Takeaways

This project demonstrates the effectiveness of transfer learning for image classification on small-to-medium datasets. By freezing most layers and fine-tuning only the top layers, we achieved strong generalization while reducing training time. DenseNet121, in particular, offered a robust balance of speed and accuracy.

Tools & Tech

- Python, PyTorch, Torchvision
- Google Colab (with GPU)
- Matplotlib, Seaborn, scikit-learn
- Dataset from Kaggle

Next Steps

- Deploying model with Streamlit for interactive demo
- Adding Grad-CAM visualizations for model interpretability
- Potentially training on larger flower datasets or fine-tuning all layers for further performance gains