Transfer Learning with MobileNetV2 & ResNet50

Introduction

Exploring transfer learning using MobileNetV2 and ResNet50, two convolutional neural networks (CNNs) with different design goals. MobileNetV2 being lightweight for mobile and embedded vision applications, and ResNet50 offering deeper architecture through residual connections. The experiment focuses on fine-tuning both models for flower classification using the TF-Flowers dataset, which consists of 3,670 labeled images spanning five classes: daisy, dandelion, roses, sunflowers, and tulips. The goal is to leverage the representational power of pre-trained models and adapt them to the flower classification task with minimal training time.

Methodology

Three main stages were followed: data preparation, model setup, and training stage.

Data Setup: The dataset was loaded and prepared with the following configurations: the input images were resized to 224×224. The data was split into 75% for training, 15% for testing, and 10% for validation. All images were converted to NumPy arrays, normalized, and one-hot encoded. Class names were dynamically extracted from the dataset metadata.

Model Setup: The model architecture was built using a customizable OOP based system configured with ImageNet weights. The input shape was set to (224, 224, 3), and all backbone layers were frozen during the initial training phase. A custom head was added, consisting of global average pooling, configurable dense hidden layers, optional dropout and batch normalization, and a final output layer with softmax activation for five classes.

Training: Training occurred in two phases. Phase 1 involved training only the classification head while keeping the pre-trained backbone frozen, using the SGD optimizer with a learning rate of 0.01 and a momentum of 0.9. Metrics such as accuracy, recall, precision, and F1-score were tracked over typically three epochs. In Phase 2, fine-tuning was performed by selectively unfreezing parts of the backbone, either by blocks or individual layers (e.g., unfreezing the top 56 layers), with the top layers always unfrozen to adapt to the dataset (e.g the top 2 layers).

Results

After fine-tuning, the MobileNetV2 architecture achieved approximately 78% training accuracy and 70% testing accuracy. Performance was relatively unstable, with fluctuations in loss and inconsistent precision and recall across classes. While some classes showed reasonable F1-scores, the overall model struggled with generalization, with most errors occurring between visually similar classes such as daisy and dandelion. In comparison, the ResNet50 architecture achieved approximately 90% accuracy on both training and testing datasets. However, applying a block-wise unfreezing strategy introduced instability in training and led to a drop in test accuracy to 88%, along with signs of overfitting. A layer-wise unfreezing strategy, on the other hand, resulted in stable training and preserved the 90% test accuracy without overfitting.

Model	Strategy	Train Acc	Test Acc	Stability	Overfitting	Notes
MobileNet	Finetuning	~78%	~70%	Unstable	Yes	Inconsistency
ResNet50	Finetuning	~**90**%	~**90**%	Stable	No	Good baseline

Discussion

MobileNetV2 showed moderate adaptability to the TF-Flowers dataset. Transfer learning, particularly the reuse of ImageNet features, helped speed up training and slightly reduced overfitting, but overall performance remained unstable and generalization was limited. Fine-tuning by layers offered marginally better results than block-based tuning, though the gains were minimal. While the modular pipeline supported flexible experimentation, MobileNetV2 may not be ideal for this fine-grained task without further tuning or architectural adjustments. In contrast, ResNet50 demonstrated stronger adaptability, with transfer learning leading to faster convergence, higher accuracy, and more stable performance. Layer-wise fine-tuning consistently reached around 90% test accuracy without overfitting, whereas block-wise tuning introduced instability and a slight drop in performance.

Conclusion

Transfer learning with MobileNetV2 on the TF-Flowers dataset achieved around 70% test accuracy, showing only modest effectiveness. While using a frozen pre-trained backbone provided a solid starting point, selective fine-tuning led to only slight improvements and overall unstable performance. Lightweight models like MobileNetV2 can be efficient, but in this case, struggled to generalize well on a fine-grained dataset with limited data. Although suitable for experimentation and fast iteration, this setup requires further optimization before being considered for edge deployment or production use.