Transfer Learning for Mushroom Classification

Introduction

This project explores the application of transfer learning using TensorFlow to classify mushroom images into nine categories: Agaricus, Amanita, Boletus, Cortinarius, Entoloma, Hygrocybe, Lactarius, Russula, and Suillus.

Dataset

The dataset comprises images of mushrooms categorized into the classes. The training set contains a balanced number of images per class, while the test set includes unseen images to evaluate model performance.

Methodology

Data Preprocessing

- Image Resizing: All images were resized to 224x224 pixels to match the input size expected by the pre-trained models.
- Normalization: Pixel values were normalized to the [0, 1] range.

Data Augmentation

To enhance the model's generalization capabilities, the following augmentations were applied:

- Random rotations
- Horizontal and vertical flips
- Zoom operations
- Width and height shifts

Model Architecture

- Base Model: MobileNetV2 pre-trained on ImageNet was used as the base.

- Custom Layers:
  - Global Average Pooling
  - Dense layer with 512 units and ReLU activation
  - Batch Normalization
  - Dropout layer with a rate of 0.5
  - Dense layer with 256 units and ReLU activation
  - Batch Normalization
  - Dropout layer with a rate of 0.5
  - Output Dense layer with 9 units and softmax activation

Training Strategy

1. **Phase 1: Initial Full Fine-Tuning:**
    a. The pre-trained MobileNetV2 base and custom top layers were trained together for 30 epochs using Adam (default learning rate).
    b. **Reasoning:** The base model was intentionally set as trainable (base_model.trainable = True) before training commenced, overriding any prior layer-specific freezing. This allowed all layers to adapt simultaneously to the mushroom dataset from the beginning.
2. **Phase 2: Refinement with Lower Learning Rate:**
    a. Training continued for 20 epochs with the entire model still trainable.
    b. The model was recompiled with Adam using a lower initial learning rate (1e-4) and a ReduceLROnPlateau scheduler was introduced.
    c. **Reasoning:** This phase aimed to refine the learned weights using smaller adjustments, guided by the scheduler monitoring validation loss, to enhance convergence and potentially improve generalization.

Optimization

- **Loss Function:** Categorical Crossentropy, suitable for multi-class classification.
- **Optimizer:** Adam, initially with its default learning rate (Phase 1), then with a lower starting rate of 1e-4 (Phase 2) for finer adjustments.
- **Learning Rate Scheduler:** ReduceLROnPlateau activated in Phase 2. It monitored val_loss and halved the learning rate if no improvement occurred for 3 epochs (patience=3), aiding convergence when learning plateaued.

Results

- Training Accuracy: 97.17%
- Validation Accuracy: 77.37%
- Test Accuracy: 100%

The model demonstrated excellent performance on the test dataset, indicating effective learning and generalization.

Conclusion

The application of transfer learning using MobileNetV2 significantly improved the classification accuracy for mushroom images. Data augmentation and fine-tuning strategies played a crucial role in enhancing model performance.

References

- TensorFlow Transfer Learning Tutorial
- Tensorflow Keras