Paddy Disease Classification using Transformer and CNN

Introduction

This report details the implementation of solutions for the Paddy Disease Classification task using two models: a Vision Transformer (ViT) and a Convolutional Neural Network (CNN). The objective is to classify paddy leaf images into one of ten classes (nine diseases and one normal). The implementation is designed to achieve high accuracy, leverage state-of-the-art models, and ensure robust and scalable code.

Implementation Details

Step 1: Importing Libraries

The project starts by importing necessary libraries such as os, torch, torchvision, timm, pandas, numpy, and others. These libraries are essential for data loading, preprocessing, model creation, training, and evaluation.

Step 2: Configuration

A Config class is defined to store various configuration parameters such as data paths, device, number of classes, image size, batch size, epochs, learning rate, and weight decay.

Step 3: Dataset Class

A custom PaddyDataset class is implemented to handle loading of images and their corresponding labels. This class also supports data augmentation for training data.

Step 4: Data Transformations

Data transformations are defined using torchvision.transforms. These include random resizing, cropping, horizontal flip, rotation, and normalization for training data, and resizing and normalization for validation and test data.

Step 5: Model Definitions

Vision Transformer (ViT)

A PaddyModel class is defined to create a Vision Transformer model using the timm library. The model is initialized with pretrained weights and adapted for the classification task with the specified number of classes.

Convolutional Neural Network (CNN)

A PaddyCNNModel class is defined to create a CNN model. The model consists of multiple convolutional layers, batch normalization, ReLU activations, max pooling, and fully connected layers to adapt for the classification task.

Step 6: Training and Validation Functions

Functions for training and validation are implemented. These functions handle the forward pass, loss computation, backpropagation, and accuracy calculation for both the ViT and CNN models.

Step 7: Training Loop

The training loop iterates over the specified number of epochs, performing training and validation in each epoch. The best model based on validation accuracy is saved for both the ViT and CNN models.

Step 8: Testing and Submission

The best models are used to make predictions on the test dataset. The predictions are saved in a CSV file in the required format for submission.

Outcome of Experiments

Vision Transformer (ViT)

The Vision Transformer (ViT) model was trained for 25 epochs on the Paddy Disease Classification dataset. The following outcomes were observed:

- **Training Accuracy:** The training accuracy improved consistently over the epochs, indicating that the model was learning effectively from the training data.
- **Validation Accuracy:** The validation accuracy also showed improvement, demonstrating good generalization to unseen data.
- **Best Validation Accuracy:** The best validation accuracy achieved was 96.49%
- **Test Predictions:** The model was able to make predictions on the test dataset, and the submission file was created in the required format.

Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) model was trained for 25 epochs on the Paddy Disease Classification dataset. The following outcomes were observed:

- **Training Accuracy:** The training accuracy improved consistently over the epochs, indicating that the model was learning effectively from the training data.

- **Validation Accuracy:** The validation accuracy also showed improvement, demonstrating good generalization to unseen data.
- **Best Validation Accuracy:** The best validation accuracy achieved was 97.05%.
- **Test Predictions:** The model was able to make predictions on the test dataset, and the submission file was created in the required format.

Conclusion

The implementation of both the Vision Transformer (ViT) and Convolutional Neural Network (CNN) models for Paddy Disease Classification was successful. Both models achieved high accuracy and demonstrated good generalization. I will recommend the CNN model for further experimentation due to its lower computation cost during predictions as well as faster training time. The code is designed to be robust, scalable, and easy to modify for further experimentation. The results are promising and can be further improved with additional hyperparameter tuning including learning rate, batch size, etc and data augmentation techniques.
