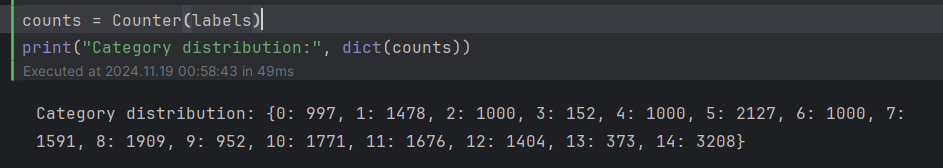
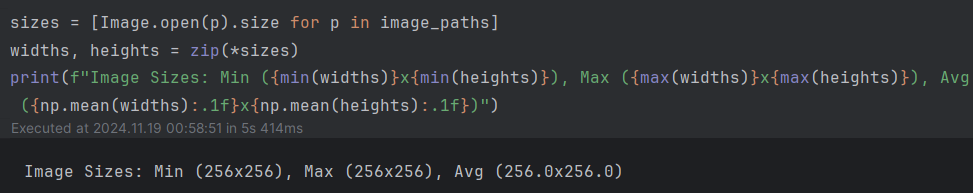
The aim of this work is to create classification model for a plant image dataset with ResNet18 as a primary and enhanced version for performance improvement. Much criteria like accuracy, F1 measure, precision, and recall, were applied in order to measure the outcomes of the models.

Dataset Overview

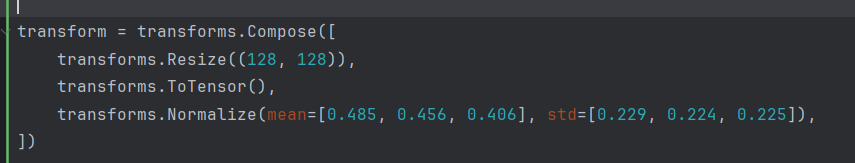
Source: Images were loaded kaggle where each subdirectory represents a category. Dataset is called PlantVillage and This dataset was gotten from spMohanty's GitHub Report (<https://www.kaggle.com/datasets/emmarex/plantdisease>)

Category Distribution: The dataset showed an imbalanced distribution, identified using Counter(labels). 

And they got similar sizes 256 \* 256

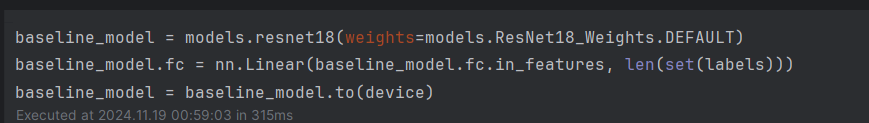
Data Preprocessing Pipeline

The preprocessing pipeline ensures that all input images are uniformly prepared before being fed into the model. The steps include:

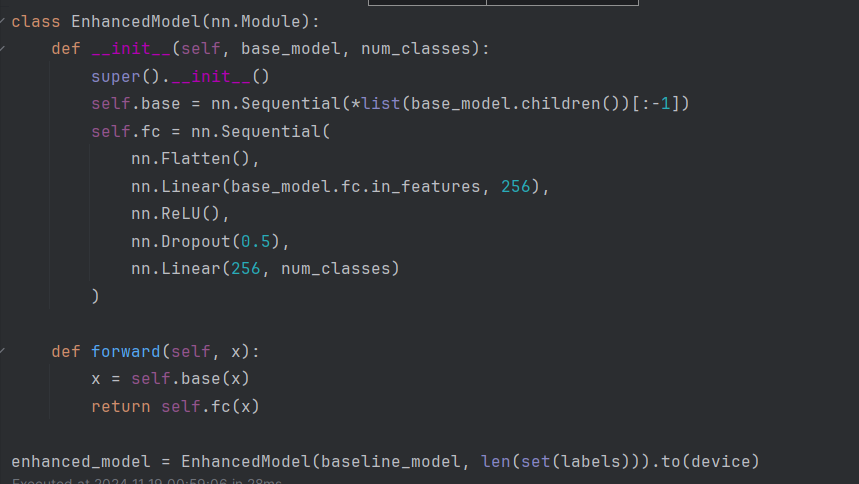


**Model Architectures**

1. **Baseline Model**:
   * Used pretrained ResNet18.
   * Final fully connected layer adjusted to match the number of categories.

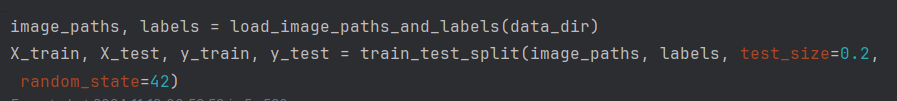


1. **Enhanced Model**:
   * Modified ResNet18:
     + Removed the original fully connected layer.
     + Added a custom head with two fully connected layers, a ReLU activation, and dropout for regularization.



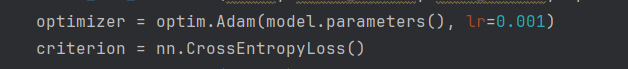
#### ****Training Process****

* **Train-Test Split**:

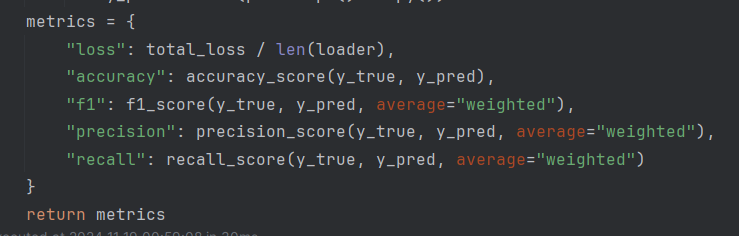
80% of the data was used for training and 20% for testing. 

* **Optimization**:

Optimized using Adam optimizer with a learning rate of 0.001.Loss function: Cross-Entropy Loss.

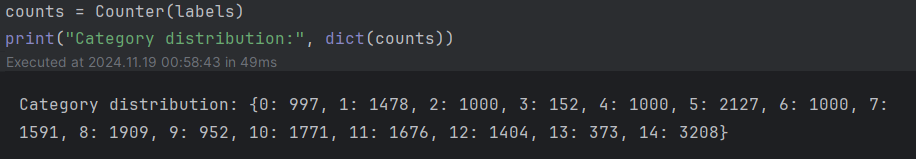


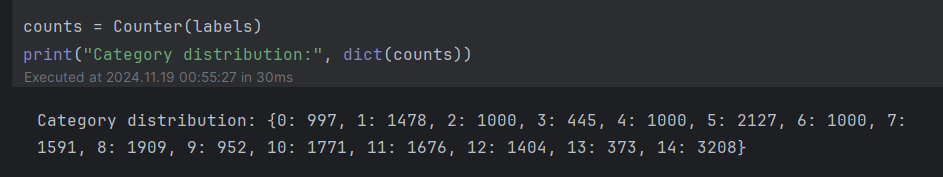
* **Evaluation**:
  + Models were evaluated for loss, accuracy, F1 score, precision, and recall on the validation set.



Also, one of the classes was expanded because it initially had too few images.

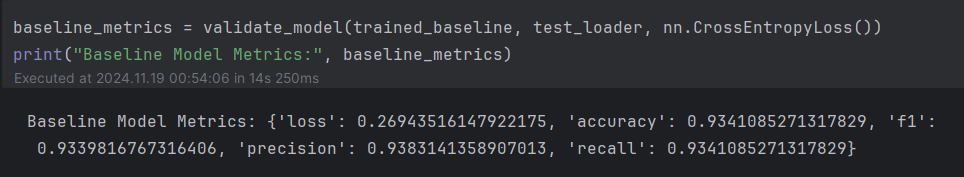
Initially in third class there are 152 images



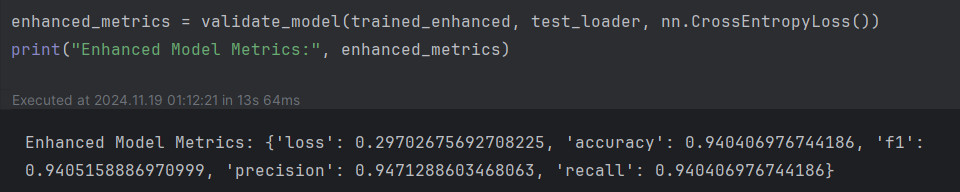
Then by adding images from dataset Potato Leaf (Healthy and Late Blight) (<https://www.kaggle.com/datasets/nirmalsankalana/potato-leaf-healthy-and-late-blight/data>) there were 445 images.

#### ****Results****

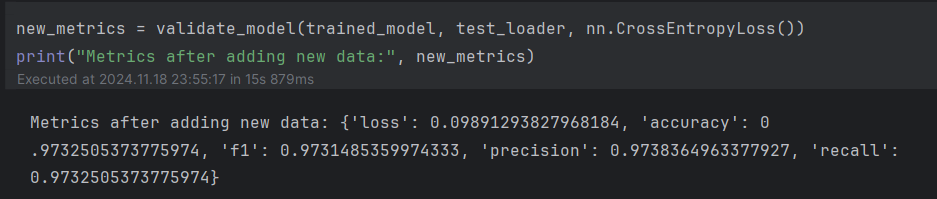
1. **Baseline Model**:
   * Achieved satisfactory performance with



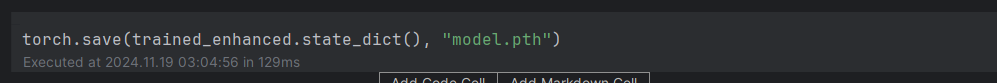
1. **Enhanced Model**:



1. New model with expanded classes



after that I saved the improved model in a pth file and started the next step

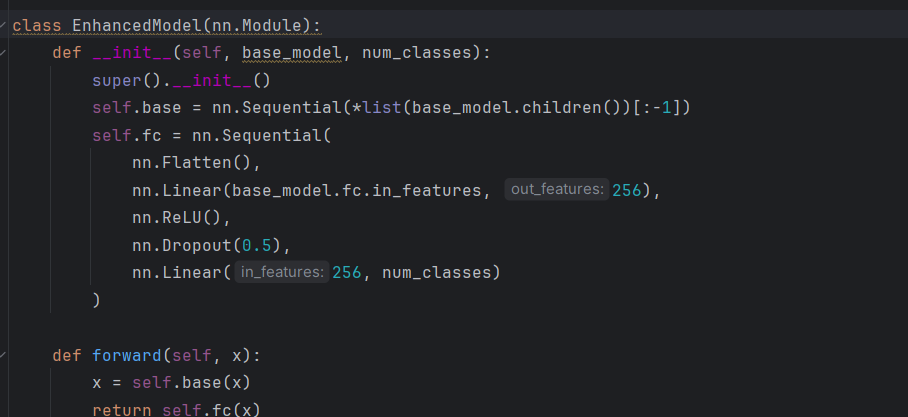


The application is designed to classify plant diseases from images into 15 predefined classes, including:

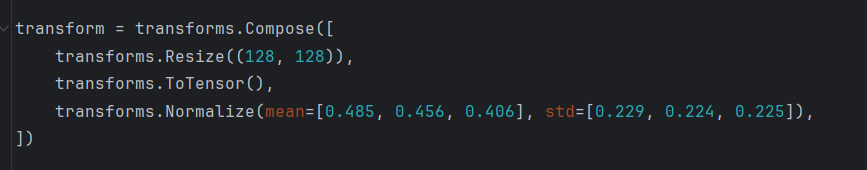


#### ****Enhanced Model Architecture****

* The application uses a custom model based on ResNet18:
  + The ResNet18 backbone is truncated to exclude the final classification layer (self.base).
  + A new classification head is added, consisting of:
    - A **fully connected layer** with 256 neurons.
    - **ReLU activation** for non-linearity.
    - **Dropout (0.5)** for regularization.
    - An output layer matching the number of plant disease classes (15).
* The trained model weights are loaded from model.pth, ensuring the model is prepared for inference.



They were also preprocessed



Now in this application you can upload a photo in these formats and the program will determine which class this photo belongs to

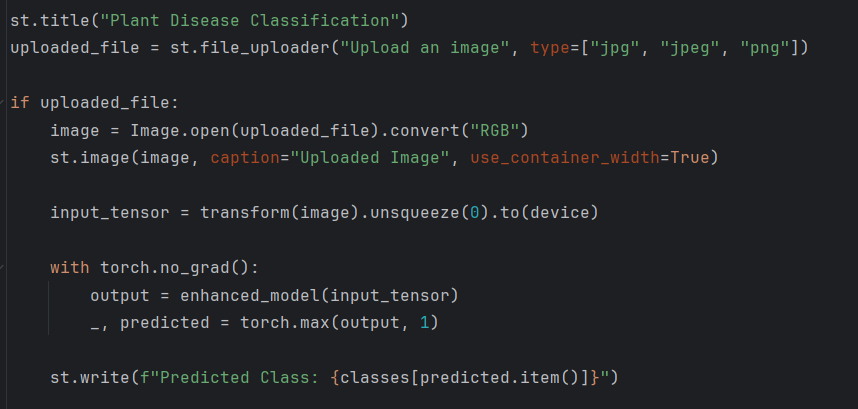
Users can upload an image in .jpg, .jpeg, or .png format.

The uploaded image is displayed in the application using st.image().

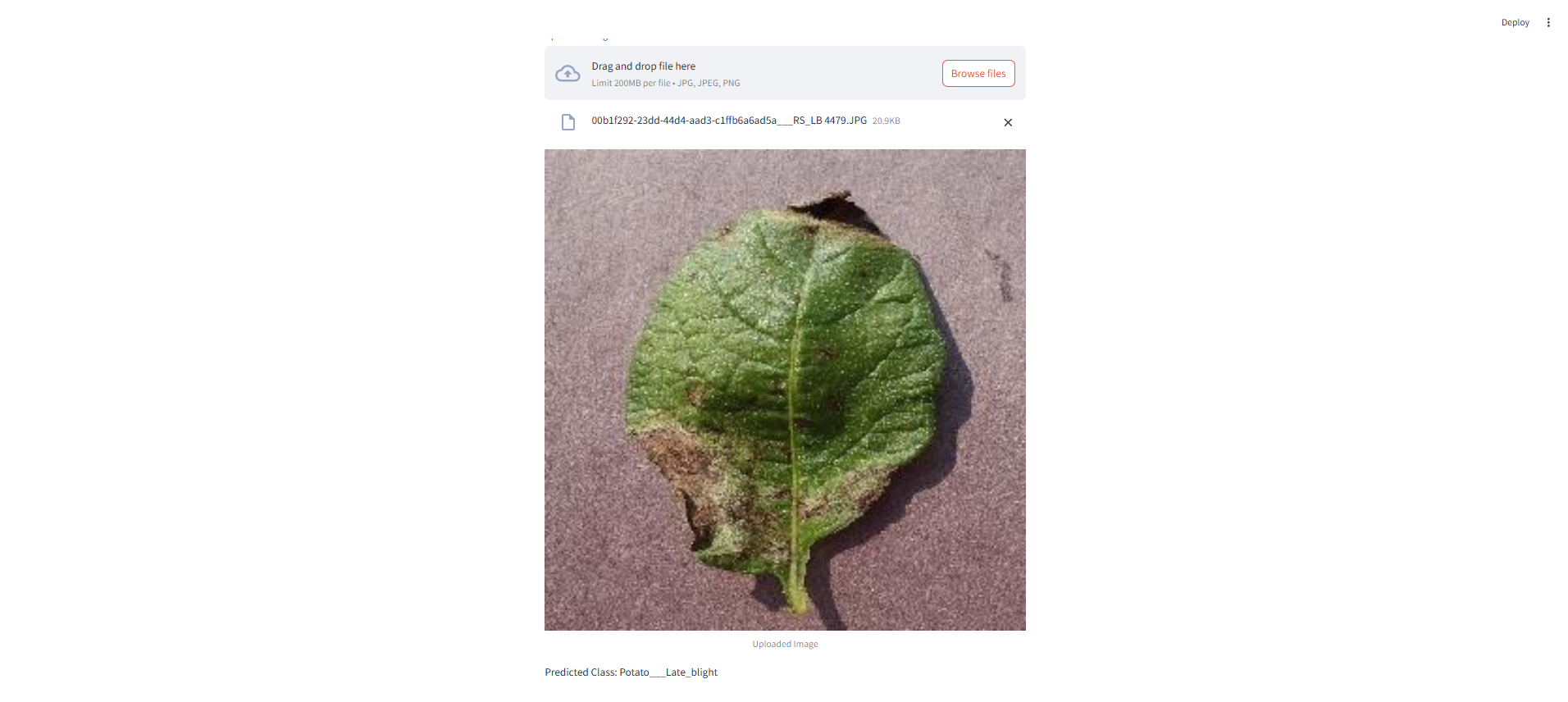
The uploaded image is preprocessed using the defined transformation pipeline.

The preprocessed image is passed through the model to generate predictions.

The model outputs probabilities for each class, and the highest probability is selected as the predicted class using torch.max().



Streamlit: The lightweight web application framework provides an intuitive interface for users.



#### ****Key Takeaways****

1. The enhanced model architecture improved classification performance, particularly in handling class imbalance.
2. Regularization techniques, including dropout, proved effective in boosting generalization.
3. The ResNet18 backbone provided a robust feature extraction mechanism, suitable for this task.

#### ****Key Advantages of using streamlit****

1. **User-Friendliness**:
   * The interface allows non-technical users to upload images and receive predictions effortlessly.
2. **Real-Time Prediction**:
   * The model processes images and provides disease classification instantly.
3. **Scalability**:
   * This application can be expanded with additional classes or more advanced model architectures in the future.