1. Introduction

To achieve Crop Type Identification with the given dataset of 10 types of crops and bare land ,each class comprising 300 images (244x244). By using PyTorch, we hope to implement a training loop and analyze training results.

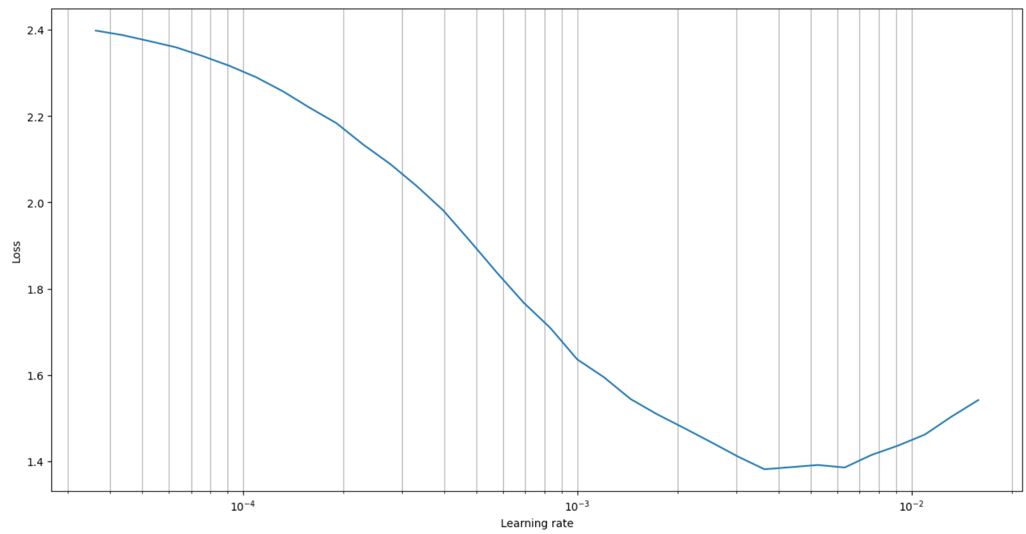
2. Methods

In this project, we have opted for the ResNet50, one of the most commonly used variants of the ResNet model, and employed the Adam optimizer. Additionally, we have set the training ratio to be 0.8.

The initial step involves loading the pre-trained ResNet model. Subsequently, we replace the pre-trained model's linear layer with our own, incorporating a randomly initialized linear layer.

3. Result & Discussion

We commence by initializing an optimizer with a very low learning rate. Subsequently, we leverage the learning rate finder to determine a suitable learning rate for our model.

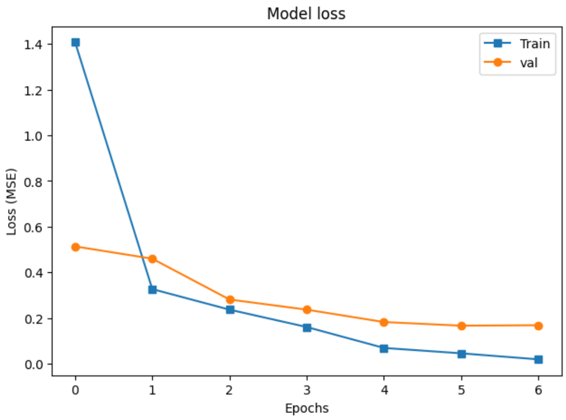


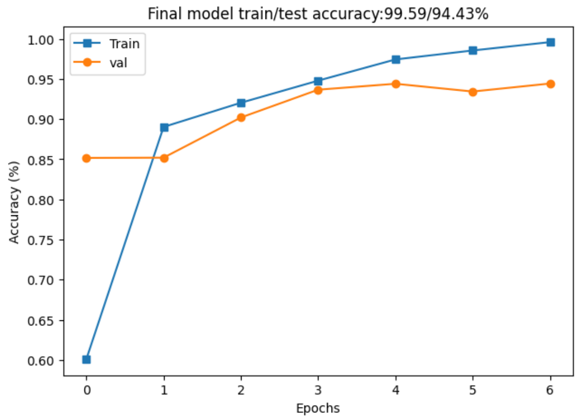
We can see that the loss reaches a minimum at around 2e-3and 3e-3.

A good learning rate to choose here would be the middle of the steepest downward curve - which is around 1e-3.

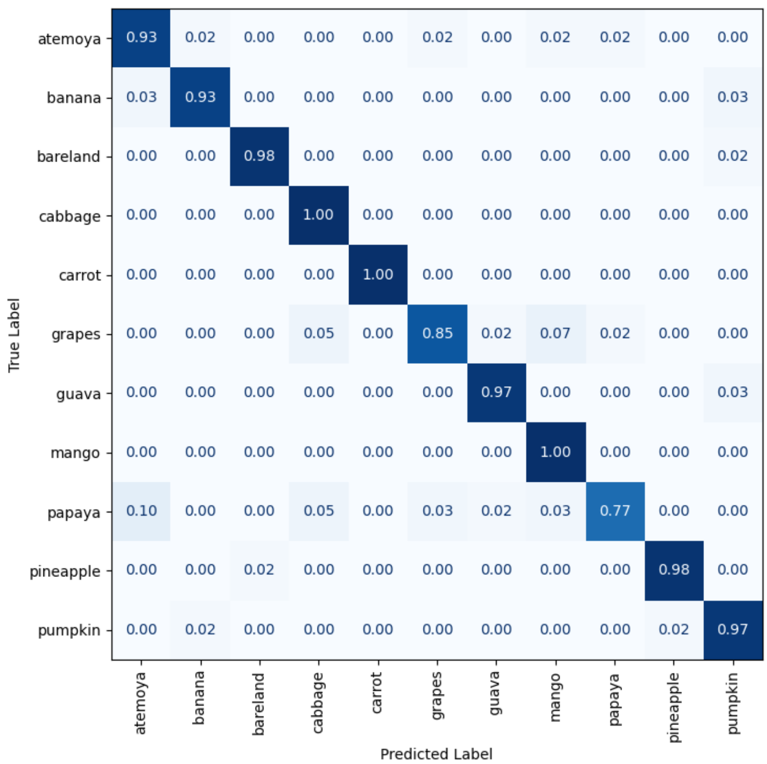
```
FOUND_LR = 1e-3

params = [
          {'params': model.conv1.parameters(), 'lr': FOUND_LR / 10},
          {'params': model.bn1.parameters(), 'lr': FOUND_LR / 10},
          {'params': model.layer1.parameters(), 'lr': FOUND_LR / 8},
          {'params': model.layer2.parameters(), 'lr': FOUND_LR / 6},
          {'params': model.layer3.parameters(), 'lr': FOUND_LR / 4},
          {'params': model.layer4.parameters(), 'lr': FOUND_LR / 2},
          {'params': model.fc.parameters()}
         ]


optimizer = optim.Adam(params, lr = FOUND_LR)
```

We apply discriminative fine-tuning to set the learning rates of our model. The learning rate found by the learning rate finder serves as the maximum rate for the final layer, while the remaining layers adopt lower rates, gradually diminishing towards the input layers.

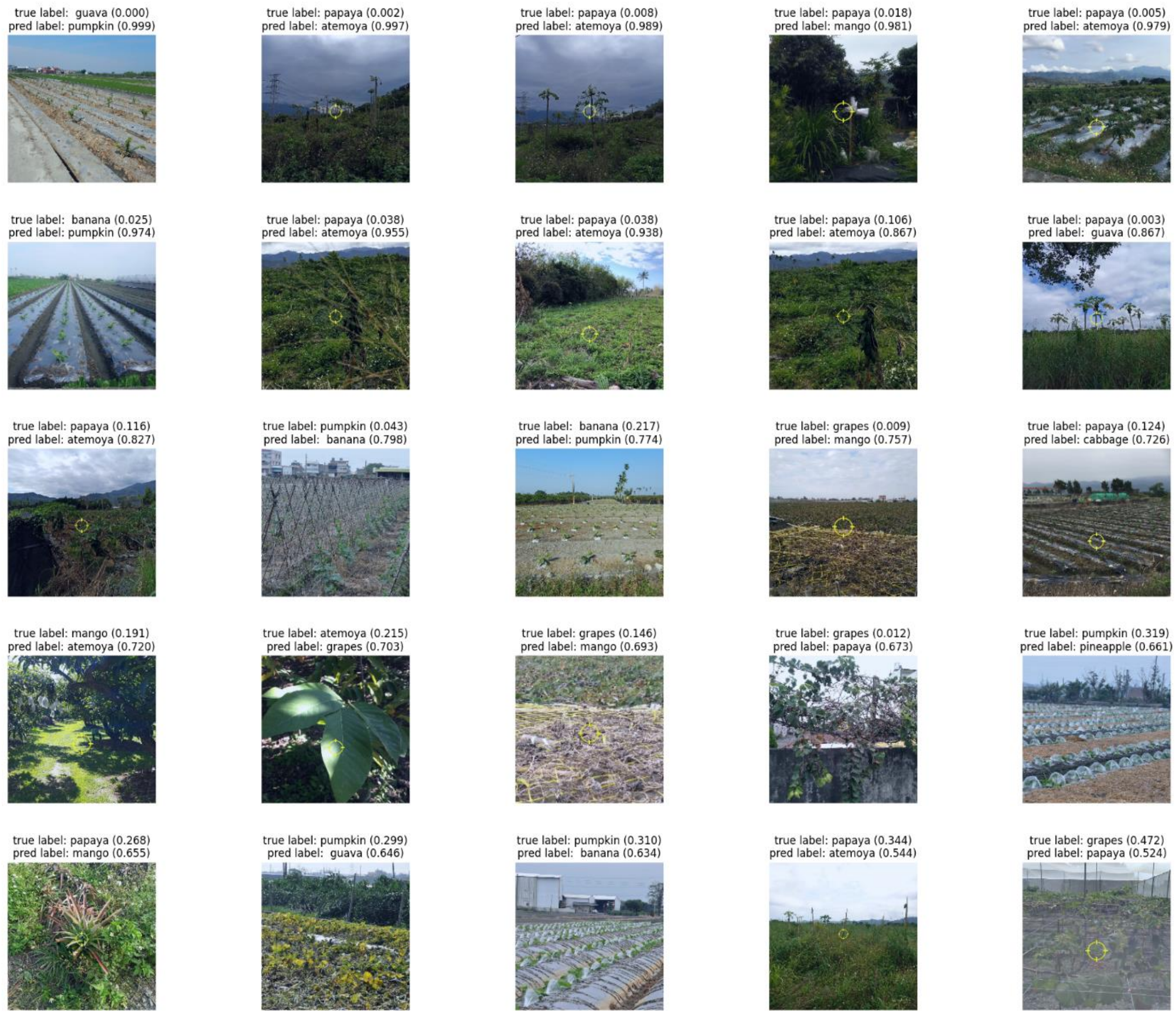




Following that, we visualize the performance across 7 epochs, culminating in a final model with training and validation accuracies of **{99.59% / 94.43%}.**



We found that our model performed poorly specifically on the classes corresponding to "papaya" , "grapes" and "atemoya." ,after plot the confusion matrix.
Next, we can extract all correct predictions, filter them, and subsequently sort the incorrect predictions based on their confidence level in the incorrect prediction. This allows us to visualize the most inaccurately predicted images, displaying both the predicted and actual classes.



Upon evaluating our model's outcomes, a notable observation is that a predominant number of misclassifications involve images initially labeled as "Papaya" but predicted as "Atemoya."

Due to a lack of comprehensive understanding of the underlying theorem within the optimizer, we encountered challenges in effectively leveraging the technique and adjusting the parameters optimally. Consequently, looking ahead, we aspire to deepen our knowledge of both the model and optimizer, aiming to enhance our learning models through more informed optimization strategies.

4. Conclusion

In conclusion, we achieved a validation accuracy of 94.43% using the Adam optimizer. Moving forward, our aspiration is to delve deeper into the intricacies of various optimizers and learning models. By gaining a more profound understanding of their functioning, we aim to adeptly adjust parameters, leading to more efficient and improved results in our future endeavors.