AFRICAN MASTERS OF MACHINE INTELLIGENCE

AMMI SENEGAL, Mbour

Name: Mr. Winners	**Project**:
Course: Computer Vision - Cassava Project Report	Date: April 17, 2022

1 Introduction

The project is on a Kaggle-in-Class competition on cassava disease classification. The project's objective was to create a model that classifies the data into five different classes of leaf diseases. The dataset used consists of leaf images of the cassava plant, with 9,436 annotated images and 12,595 unlabeled images of cassava leaves. In our model formulation, we used both labeled and unlabeled data and compared the model results, which are discussed in the report.

2 Exploratory Data Analysis(EDA)

2.1 Dataset Exploration

The data is divided into training, validation, and test data. The training data and validation data are labeled. In addition, we have extra data(extra images) which is unlabeled. The summary of the dataset is as follows;

set	*TotalExamples*
Training	9436
Testing	3774
Extraimages	12595

2.2 Data augumentation and Labeling

We employed various data transformations which includes rotation, flipping, increase of contrast, resizing, and erasing, among others, to learn the invariances efficiently in the image classification. In addition, we used an inbuilt module called ttach(Test Time Augmentation), which helps to perform random modifications to the test images. The transformations helped in making our predictions better. To label the unlabeled data, we used the pseudo labeling technique. First, we trained our model on the training data(labeled data), which we achieved a quite good level of accuracy, then used the model to classify unlabeled data. We obtained the first 2000 classifications associated with the highest confidence score from the new labeled data. Then merged the new labeled data and the original training data and trained again using both resnet101 model with the results given in the model's section. We did a repetition on the procedure choosing different sizes in newly labeled data of 2000,1000, and 500 based on highest confidence score giving various accuracy score as discussed in the model's section. the choice of random sizes of the newly labled data was random but with care to reduce bias introduced to our original trained model.

3 Models

3.1 Backbones

First, we tried convolutional neural network with a structure of *Conv* $\rightarrow$ *ReLU* $\rightarrow$ *MaxPool*$\rightarrow$ *Conv* $\rightarrow$ *ReLU* $\rightarrow$ *MaxPool* $\rightarrow$ *Conv* $\rightarrow$ *ReLU* $\rightarrow$ *Conv*$\rightarrow$ *ReLU* $\rightarrow$ *MaxPool* $\rightarrow$ *Linear*. The accuracy we recorded was poor, so we opted for pre-trained models. We used different derivatives of resnet pre-trained models. The motivation to use the resnet models was inspired by several works of literature we read about image classification, which showed it worked better in previous studies. We tried resnet50, resnet152 and resnext101. Finally we settled on resnext101 which had best results compared to the other models.We tried to improve on it with pseudo labeling. In pseudo labeling, we used the extra images in the dataset where, we first labeled the extra images with the already trained model, and we took k top-scored images with their labels and trained our model again with these data and the initial training data. However, the pseudo labeling did not help improve our score further, but it recorded the highest test accuracy in the validation data.

3.2 Hyperperameter tuning

To improve the accuracy of our models, we tuned various parameters, including the learning rate, batch_size, and the optimizer. In the convolution neural network, we used a stochastic gradient which performed better than the Adam optimizer with an accuracy of 0.35 for 10 epochs. Due to the low accuracy in the convolution model, we settled on the pre-trained models where the Adams optimizer had better results. From the motivation of several works of literature, we settled on Adam's optimizer version, AdamHD, which performed better and had inbuilt automatic tuning of the parameters such as learning rate. In all cases, we maintained the CrossentropyLoss criterion. After several experiments, we had a batch_size=8, input dimension = 550, and 10 epochs worked based on our models, giving results within 5 hours of training, unlike other parameter tunings. In general, we employed early stopping method to obtain our optimal results in the case of resnext101 model where we stopped training at the 8th epoch which gave the best results.

3.3 Model's results and discussion

Model	*No. of Epochs*	*Train loss*	*Test accuracy(%)*	*Public_scores*
Convolution	10	1.131644	35.00	0.33112
Resnet50	10	0.849484	46.00	0.35827
Resnext101	10	0.140204	92.02	0.91589
Resnet152	10	0.214874	90.01	0.88013
Resnext101 with Pseudo labeling(k=500)	10	0.090906	94.40	0.89867
Resnext101 with Pseudo labeling(k=1000)	10	0.050906	93.70	0.88013

where k is the size of highly scored extra images to be added in the training data. The 'resnext101' model gave the best results as evident from thepublic scores in the table above. However, interms of test accuracy, the introduction of pseudo labled data, the test accuracy increased despite the score being low as per the public score. The efficientnet models tried failed to work to complletion due to large number of parameters despite recommendations from literatures.

4 Conclusions

Based on our results, we realized that the optimal results were acquired through the best parameter tuning for the resources at hand. The resnet101 model worked best for us, with ten epochs . Besides, we realized that pseudo labeling is essential in improving the model's accuracy despite giving a lower score than expected. With pseudo labeling, we achieved a test accuracy of 94.40% which was higher than the test accuracy of our highest scoring model without unlabeled data.