AFRICAN MASTERS OF MACHINE INTELLIGENCE (AMMI), AIMS - Senegal
KAGGLE COMPETITION REPORT AMMI 2023
CASSAVA LEAVES DISEASE

Dieu-Donne FANGNON and Verlon Roel MBINGUI /AMMI-2023(Bourbaki's Team)

Introduction

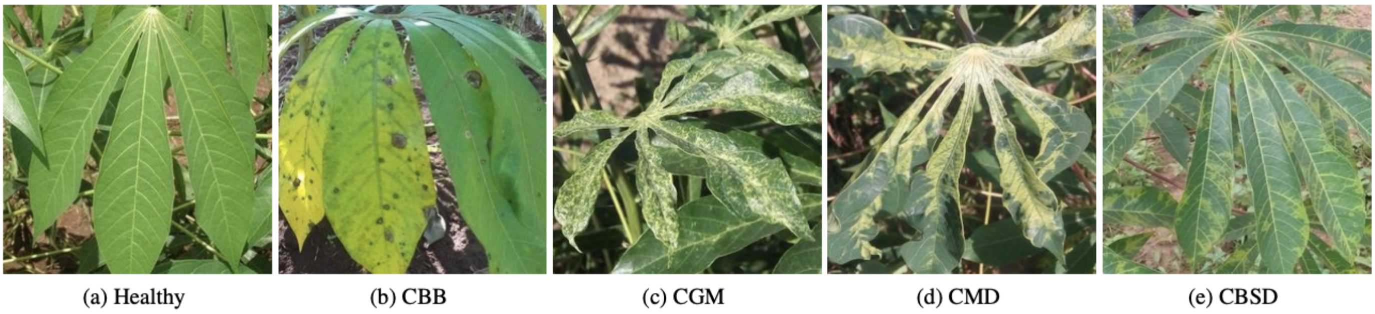
As the 2nd largest provider of carbohydrates in Africa, cassava is a key food security crop grown by small-holder farmers because it can withstand harsh conditions. At least 80% of small-holder farmer households in Sub-Saharan Africa grow cassava and viral diseases are major sources of poor yields.

In this competition, we introduce a dataset of 5 fine-grained cassava leaf disease categories with 9,436 labeled images collected during a regular survey in Uganda, mostly crowdsourced from farmers taking images of their gardens, and annotated by experts at the National Crops Resources Research Institute (NaCRRI) in collaboration with the AI lab in Makarere University, Kampala.

The Cassava disease leaf Data : For more details see here

There are 12,595 unlabeled images of cassava plant leaves in our dataset, which consists of 9,436 identified images. The annotations are divided into 5 classes: healthy plant leaves (316/211 train/test examples), diseased plant leaves (representing the 4 diseases CMD (2658/1773 train/test photos), CBSD (1443/963 train/test images), CBB (466/311 train/test images), and CGM (773/516 train/test images). Figure 1.1 shows examples of each of the four categories of illnesses' typical leaf images.

Visualization of the classes



(a) Healthy (b) CBB (c) CGM (d) CMD (e) CBSD

Figure 1: Visualization of classes. See here

1 Model Used and Hyperparameters

We employed a pretrained model called **EfficientNet-B0**(for more details see here)which is a convolutional neural network that is trained on more than a million images from the ImageNet database for our final result, which helped us place where we do in the competition.

1.1 Architecture:

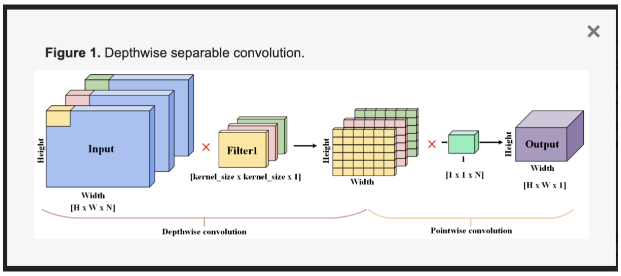


Figure 1. Depthwise separable convolution.

Figure 2: EfficientNet (Credits : see here)

1.2 Hyperparameters Used for the competition

For the competition, our best score (public score) was obtained by using the AdamHD optimizer ($lr = 1e-4, \quad hypergrad\_lr = 1e-9$), as criterion the Cross-Entropy Loss, we used Classification model wrapping for Test- Time Data Augmentation. The batch size=8, Number of epochs=30. We used data augmentation for the model:

(i) For training: Random Rotation (30), Resize (550), Random Crop (500), Random Horizontal Flip (0.3), Random Vertical Flip (0.3), Random Erasing(0.1), Normalization

(ii) For inference: Resize (550), Center Crop (500), Normalization

2 Evolution on the scores and tried models

In Table 1 we present a sample of 16 submissions that we made during the competition.

Models	Public Score	Description, Epochs, optimizer, batch-size, transform
CNNs	**0.4596**	Pretrained, 10 epochs, SGD(lr=0.001), batch size :32, resize (224,224
ResNet50	**0.84105**	Pretrained, 10 epochs, SGD(lr=0.001), batch size :32, resize (224,224)
ResNet18	**0.85761**	Pretrained, 25 epochs, SGD(lr=0.001), resize (224,224)
ResNext101_32 × 8d	**0.87947**	Pretrained, 35 epochs, SGD(lr=0.001), batch size :32, resize (224,224)
EfficientNet-B0	**0.90728**	Pretrained, 20 epochs, AdamHD(lr=0.0001),batch size :8
EfficientNet-B0	**0.9086**	Pretrained, 30 epochs, AdamHD(lr=0.0001),batch size :8

Table 1: Evolution on the scores and tried models

Conclusion and Observations

The six models that are being presented in Table 1 are a sample of 16 submissions that we made during the completion on the platform for the competition. By the way we tried several models and we noticed that, to get better accuracy, one should take into account different technics to fight the overfitting but also the underfitting. Fixing hyperparameters at ”good” scales, working on the data processing (splitting and transforming data: data augmentation), choosing good optimizer with its optimum parameters.

Using pretrained models is one of the key ideas from what we observed. So we use pretrained model and make sure to optimize it for best solution. The challenging part, when deciding to work with a pretrained model was to get the sweetest model and to find interesting parameters for it.

Finally, we use again pretrained model to make sure that the model is optimized, from the training and testing. As one can see in the above table, a model can perform well on a small data set (40% in our case), but wouldn’t do on the large. Training longer can work but not in all the cases. So, focusing on data processing and optimal hyperparameters is a good option to emphasize for the actual problem. Our best public score (0.9086) was gotten by using **EfficientNet-B0** with a training time of 30 epochs, batch size = 8, learning rate=1e-4, AdamHD optimizer.