Cassava Image Classification Capstone Report

Introduction:

Cassava, a woody shrub native to South America, has become the third most important source of calories in the tropics, behind rice and corn.[1] It is especially important in Africa in agro-ecological zones where other crops such as cereals cannot thrive due to hot and humid or sub-humid environs. More than 800 million people depend on cassava as their primary food staple, frequently in areas where food insecurity and mal- or undernutrition is rampant. The root of cassava is a source of dietary starch, and the leaves are an important source of proteins, minerals, and vitamins. Cassava is not only a major source of nutrition but also income for many African farmers, and recent research into its potential use as a biofuel may increase its value as an income source.[2]

Unfortunately cassava yields are lower than average in some of the poorest regions that depend on it the most, such as central, eastern, and southern Africa. These losses are largely due to cassava-related diseases such as cassava mosaic disease (CMD), cassava brown streak disease (CBSD), cassava bacterial blight (CBB), and cassava green mottle (CGM). While these diseases have different etiologies they can have similar devastating effects on plants by causing leaf wilting, reduced plant size, tuber rot, or decreased starch production. They can cause yield losses of 10% to 70% depending on the disease and context, with CBSD causing losses of up to 100% in susceptible varieties.[3]

Leaves of a healthy cassava plant. Credit: FAO/G. Napolitano

The leaves of a plant infected with CMV. Credit: FAO/G. Napolitano

Figure 1: Healthy and diseased cassava plants

Mitigation approaches require both long- and short-term strategies. Long-term research into developing resistant cultivars is challenging, since some resistant breeds are actually just tolerant to the diseases, meaning they can still host them with no symptoms and inoculate

[1] https://www.apsnet.org/edcenter/apsnetfeatures/Pages/cassava.aspx

[2] https://www.bangkokpost.com/business/2042767/e20-set-to-be-leading-fuel-by-july-2021

[3] https://plantvillage.psu.edu/topics/cassava-manioc/infos/diseases_and_pests_description_uses_propagation

neighboring plants. Additionally new strains of viruses and bacteria can emerge to threaten previously resistant crops. Short-term strategies primarily rely on cultural practices and quarantine procedures as the first line of defense.[4] Intercropping with maize or melon can significantly reduce spread, possibly due to the increased distance between plants. Cassava is propagated through vegetative cuttings which can cause spread, as can cassava debris from falling leaves and soil contamination. Worse, even apparently symptom-free stems may contain viral or bacterial particles if they are in proximity to sick plants. It is therefore very important that farmers only propagate cuttings from *completely disease-free* fields, and that any plants that begin to show symptoms of disease are immediately extracted and disposed of, possibly through burning.[5] This approach can be challenging for farmers since some disease symptoms may be difficult to identify, especially if the farmer does not have extensive experience with the disease.[6] Currently many farmers must solicit help from government-funded agricultural experts to come visually inspect their crops and advise them, which is costly and labor-intensive.

Objective and Data Source:

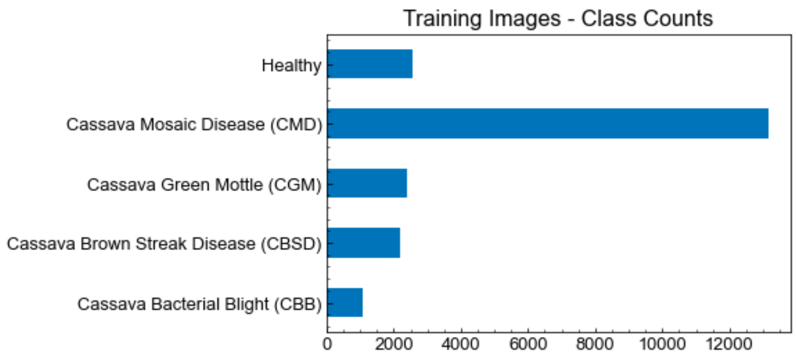
The goal of this project was to develop a model capable of classifying an image of a cassava plant into one of four disease categories or as healthy. The model was trained using 21,367 images crowdsourced from farmers in Uganda and annotated by experts at the National Crops Resources Research Institute (NaCRRI) in collaboration with the AI lab at Makerere University, Kampala.[7] It is worth noting that since the data were crowdsourced from farmers the images are realistic to what is most useful to farmers, given that many may only have access to mobile, low-quality cameras with low bandwidth. The goal was to develop a model that could give farmers quick, accurate, inexpensive and actionable feedback on the status of their crops so that they can take appropriate actions to mitigate any losses. As such the main model metric is overall classification accuracy for the five classes, with special attention to differentiating the healthy class from the diseased classes.

---

[4] https://www.intechopen.com/books/cassava/cassava-bacterial-blight-a-devastating-disease-of-cassava
[5] https://www.pestnet.org/fact_sheets/cassava_green_mottle_068.htm
[6] https://www.cabi.org/isc/datasheet/17107
[7] https://www.kaggle.com/c/cassava-leaf-disease-classification/overview

Image Preprocessing

The full data set of 21,367 images were found to be quite imbalanced with most classes having ~2000 images, a large overrepresentation of the CMD class (~13,000 images), and a slight underrepresentation of the CBB class (~1000 images). A balanced training generator using class sizes of 2000 images with appropriate up- and down-sampling of the classes was initially used for model training before training on the full unbalanced data set in later epochs.



There were two types of augmentation used during training: basic augmentation and cutmix preprocessing. Basic augmentation methods were selected based on the data variability that would be expected from pictures taken by farmers in the field. These included:

1. rotation - cameras can be held at any angle relative to the plant
2. brightness - pictures can be taken in shadow or bright direct sunlight
3. zoom range - there is no standard distance from which these images would be taken

Modest image distortions were implemented to serve as a regularizing effect. Since image distortions are randomly selected every time a batch is generated this means a given image may have slightly different alterations each epoch, providing an effectively new or partially new training set each iteration. Shearing, height and width shift, channel shift, and flipping are all methods for slightly altering the images such that the output still contains meaningful features that one would expect to encounter in different test images. Some example images with basic augmentation are shown in Figure 2.

Figure 2: Training images with basic augmentation.

Figure 3: Training image with cutmix augmentation

Cutmix[8] preprocessing was performed in the later training stages. Cutmix randomly crops training images together and reclassifies the resulting image as an appropriately weighted combination of the two individual image classes together. This method has shown to be effective in training networks to generalize better and have better object localization abilities in the same manner as regional dropout, but makes more efficient use of training pixels. An example of a cutmix image is shown on the left.

For both basic and cutmix augmentation the resulting images were rescaled so that pixel values lie in a 0 to 1 range and a training / validation split of 85:15 was used.

[8] Yun, S. *et al*, arXiv:1905.04899 **[cs.CV]**

Model Architecture

Preliminary training on balanced, non-cutmix training data was performed on several model architectures including ResNet50, ResNet152v2, Xception, and EfficientNet-B4. Of these EfficientNet-B4 performed the best and was selected for further training.

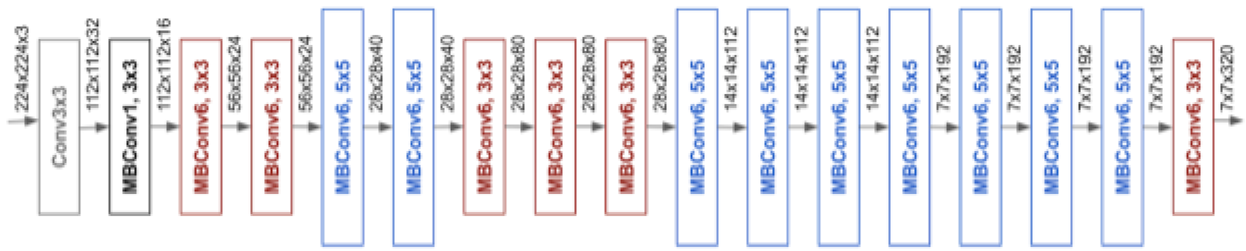


Figure 4: EfficientNet-B0 Architecture

The EfficientNet architectures were developed to optimize the scaling of network depth, width, and resolution within fixed resource constraints. Tan and Le[9] were able to show that within the constraint of using 2^N more computational resources performance can be optimized by scaling depth by α^N, width by β^N, and image size by γ^N, where α, β, and γ are constants determined by a small grid search. This result is supported by the intuitive notion that these scaling parameters are related, since higher-resolution images should require more channels to capture detailed patterns and more layers to increase the receptive field since features will contain more pixels. This scaling method was then applied to a baseline network, EfficientNet-B0 (Figure 4), that was optimized for both accuracy and to minimize floating point operations per second, resulting in the EfficientNet-BX series of architectures. These architectures were found to compare quite favorably to other architectures using Imagenet top-1 accuracy as a metric, as shown in Figure 5.

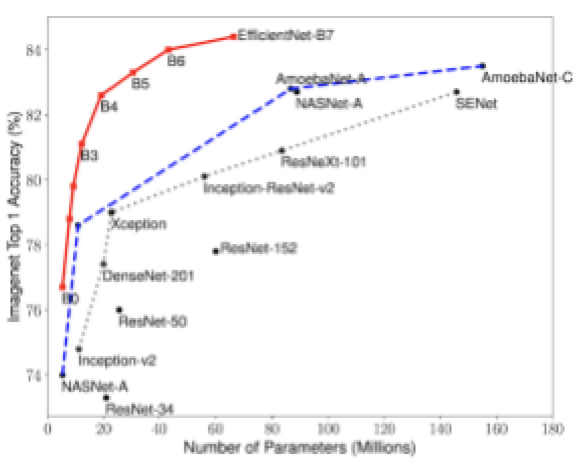


Figure 5: EfficientNet performance.

The base of the EfficientNet-B4 architecture was used along with a two-dimensional global average pooling layer, a dropout layer with an initial dropout rate of 0.2, and a dense output layer with softmax activation for the five possible classes.

[9] arXiv:1905.11946 **[cs.LG]**

Model Training

The model was initially trained for 30 epochs using a balanced training set with a class size of 2000 images without cutmix augmentation. During this phase of training the first two blocks (first 87 layers) of the model were frozen. The Adamax optimizer was used with an initial learning rate of 0.001. A callback was instituted to automatically reduce the learning rate by a factor of 0.3 should the validation loss not decrease for three epochs. As shown in Figure 6 the model accuracy and loss improved significantly during these first epochs.

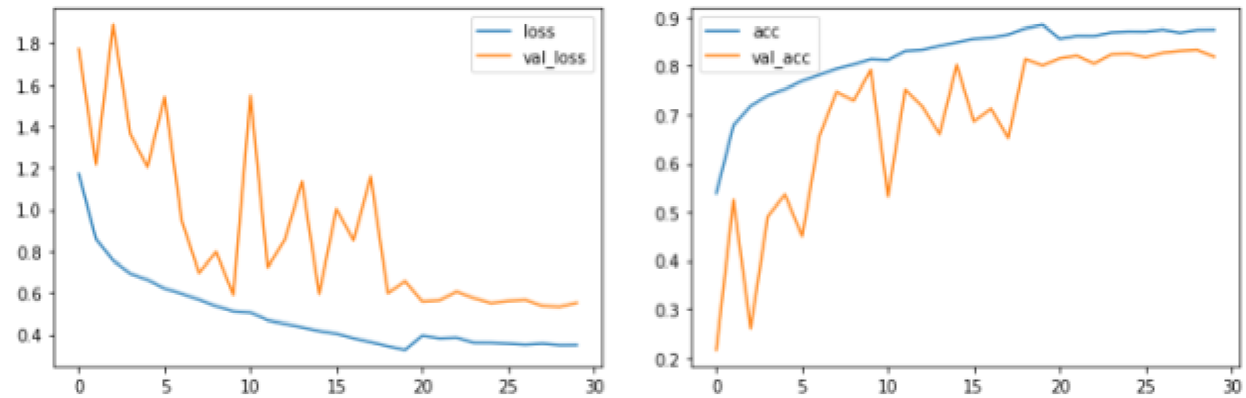


Figure 6: Training and validation loss and accuracy for the first 30 epochs.

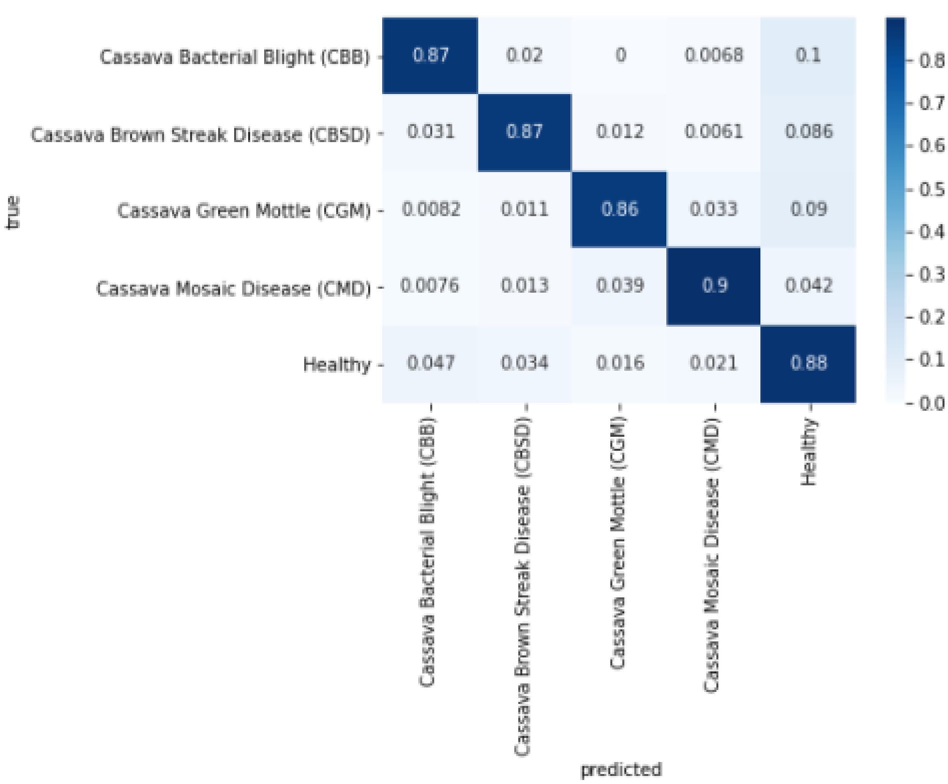
The early layers were then unfrozen and the model was trained on balanced cutmix training data, and then the full unbalanced training set. The dropout rate of the final layer was decreased to 0.1 during this training stage. Unfortunately due to inconsistent access on Google Colab's free GPU there is no record of the training history for these stages. For example code please refer to the provided notebooks. The final, compiled model h5py file is too large to host on github, but it can be [downloaded from google drive here](#).

Model Evaluation

The final optimized model was evaluated on the validation set. The results are shown in the table below, and the confusion matrix is shown in Figure 7. Overall the model performs fairly well with a validation accuracy of 89%. The model was also submitted to a true blind test set on Kaggle where it achieved 87.8% accuracy, demonstrating that the validation set is a fair representation for model performance. The model's greatest weakness is the precision of the healthy class. Nearly one third of images predicted to be healthy are actually diseased in some way. In deployment this could lead to serious consequences, e.g. a farmer propagating a diseased plant and spreading even more blight amongst her crops.

	Precision	**Recall**	**F1-Score**	**Support**
Cassava Bacterial Blight	0.74	0.87	0.8	148
Cassava Brown Streak Disease	0.86	0.87	0.86	326
Cassava Green Mottle	0.78	0.86	0.82	365
Cassava Mosaic Disease	0.99	0.9	0.94	1984
Healthy	0.68	0.88	0.77	386
Accuracy			0.89	3209
Macro Average	0.81	0.88	0.84	3209
Weighted Average	0.9	0.89	0.89	3209

Figure 7: Confusion Matrix.



Applications and Future Work:

Future work could involve improving the model performance with respect to the healthy class in particular. It could be beneficial to reclassify the images as either being healthy, diseased (CBB, CBSD, CMD), or infected with pests (CGM). By decreasing the classes to three it is possible the resulting model would do a better job of distinguishing between these three groups. These three classes are suggested because the response that a farmer would take for any plant within a class is identical, and is different between classes. In other words, cassava plants with CBB and CMD are treated basically the same regardless of the underlying disease.

In conclusion, the model developed here could have very useful applications as a way for farmers to determine if their cassava crop is infected. To demonstrate the potential application a flask application has been created where a user can upload an image and receive the model prediction and probability.

Figure 8: Website screenshots.