Cassava Leaf Disease Classification

Matthew Armstrong

Department of Computer Science

University of Saskatchewan

Saskatoon, Saskatchewan, Canada

mma118@usask.ca

ABSTRACT

The Cassava Leaf Disease Classification task is a competition, provided by Kaggle. The intent of this challenge is for users to provide machine learning techniques and approaches to the dataset, in order to achieve accurate classifications of large sets of data. The dataset provided is a collection of images of Cassava leaf images, both healthy and afflicted with a variety of common leaf diseases. This report will cover the entire process of inspecting the dataset and processing image data into data formats that can be used by a neural network. A CNN architecture called EfficientNetB3 was used during the evaluation phase, with a categorical accuracy of 0.6342 being achieved. Being an improvement over more simplistic learning models like logistic regression, these results further prove that deep learning solutions can be successful applied on image classification tasks. This presents an exciting opportunity for deep learning, as improvements on these results have positive implications on the viability of practical applications in the future.

1. INTRODUCTION

The successful harvest of the Cassava plant is vital for the farmers who grow it and millions of people who rely on it as a nutrient-dense food source. Cassava roots and leaves also serve as a vital basis for a multitude of consumer products like flour, alcohol, sweeteners, textiles and bio-degradable materials. Cassava producers, especially in Africa, lose up to 50%[1] of their crop yields each year to plant leaf diseases. The only method of disease detection currently in widespread use in the region is manual inspection. This presents a significant opportunity for computer learning. A learning model that could detect and classify these leaf diseases accurately could save producers a significant amount of time and enable them to improve yields through earlier outbreak detection. The dataset for this task is provided is a collection of images of Cassava leaves, captured during a national survey of Cassava plant farmers in Uganda. The proposed task is a classification task, as the goal for an applied learning model is to match input data with the correct output label (integer from 0-4), representing one of 5 Cassava Leaf Disease classifications. These classifications are as

[1] Estimate Figure from the *Agricultural Research Council*

follows: “0” for a leaf with Cassava Bacterial Blight (CBB), “1” for a leaf with Cassava Brown Streak Disease (CBSD), “2” for a leaf with Cassava Green Mottle (CGM), “3” for a leaf with Cassava Mosaic Disease (CMD), “4” for a healthy leaf. This dataset presents an intriguing challenge from a computer science perspective, as it provides a platform for the exploration of various dataset augmentation techniques. The images in this dataset contain a significant amount of variance in resolution, background/foreground objects, lighting conditions and other image noise. This variance means that even images sharing a label might look very different, making it difficult for a model to make accurate connections between an image and its correct classification. As such, we can use this competition to demonstrate how exposing a model to a larger, more varied (augmented) dataset can help reduce generalization error in the classification of poorly captured images.

2. METHODS

2.1. The Dataset

The dataset used in evaluation was annotated by experts at the National Crops Resources Research Institute (NaCRRI). It is a collection of 21,367 images, separated into folders representing the class they belong to. Kaggle has also provided a .csv file that associates an image id with a corresponding label. An example image from each class is included below to illustrate the significant presence of image noise, due to the handheld capture methods used by survey participants.

Figure 1: **Cassava Leaf with Cassava Bacterial Blight. The leaf is captured out of focus, with light saturation from the camera’s flash**

Figure 2: Cassava Leaf with Cassava Brown Streak Disease. Shadows and the presence of multiple leaves make classification more challenging

Figure 3: Cassava Leaf with Cassava Green Mottle. The leaf is again out of focus, while also being cropped slightly out of frame

Figure 4: Cassava Leaf with Cassava Mosaic Disease. Resolution of this picture is lower, so the image appears grainy and discolored

Figure 5: Healthy Cassava Leaf. Out of focus subject, multiple leaves in the background, and the inclusion of a rare leaf coloration all present an added challenge for classification

Another potential obstacle presented by this dataset is its distribution of class images. The pandas and matplotlib Python libraries were utilized to create a visualization of the significant class imbalance:

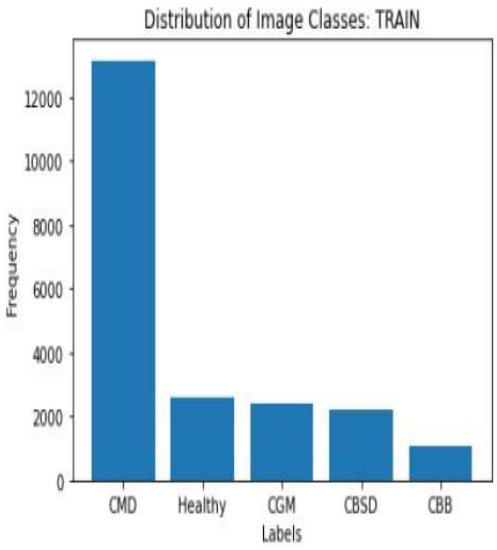

Figure 6: Class Distributions from the Cassava Leaf Disease Classification dataset

Label	Count
3	13158
4	2577
2	2386
1	2189
0	1087

Figure 7: Numerical values of each class size

Images with the label '3', indicating the presence of Cassava Mosaic Disease, comprise over 60% of the overall dataset. An applied learning model might struggle to correctly classify images belonging to the other 4 classes, simply due to the comparative lack of exposure to their training examples. Augmentation of this dataset to rebalance this dataset is therefore very important. The second significant challenge posed by the input data is the similarity between images of different classes. Supervised Contrastive Learning translates multi-dimensional image into 2 linear dimensions to create a visualization of the similarity between class data. An embedding of Cassava Leaf Disease Classification is included here:

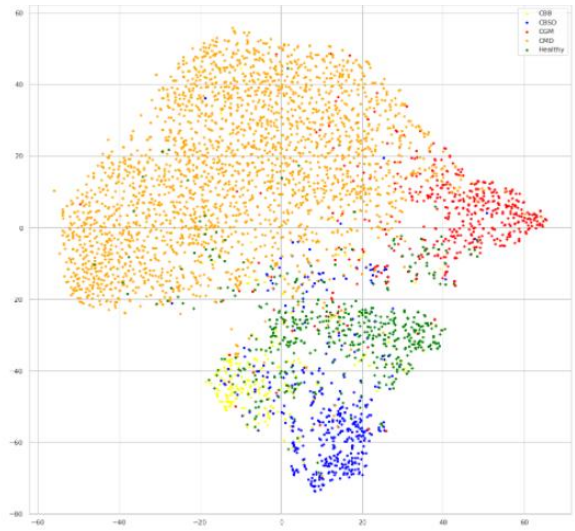

Figure 8: Embedding visualization of the model trained with Supervised Contrastive Learning. Data legend: CBB = yellow, CBSD = blue, CGM = red, CMD = orange, Healthy = green

The 2D translations, and the overall lack of separation between data from different classes demonstrates the significant overlap between data from different classes. Classification of images with a large number of similarities will be difficult for a model with an insufficient number of trainable parameters, which must be considered during evaluation. The final hurdle presented by the dataset is the existence of a variety of image resolutions. There are over 15 unique image sizes present, with the most common size being 500x500 (width, height) pixels. Preprocessing image data will inevitably mean that some input features will be lost in the reshaping process.

2.2. Preprocessing

In order to be processed by a CNN or similar learning model, images must be standardized to a specified input size for the data transformations to take place. In the initial planning phase of the evaluation, a target size of 512x512 was identified as an ideal tensor shape for the input data. In practice however, memory constraints prevented the use of this as an input shape, so a value of 300x300 was chosen instead. Using Keras'[2] ImageDataGenerator, image data is processed and converted to a tensor of size 300x300x3. Several augmentations were also applied to offset the class imbalance and reduce overall generalization error. Only 80% of the class '3' images were selected for use in the training/validation sets. Since this was a real-world dataset, an assumption was made during preprocessing that the overall class distributions corresponded with real-world frequency of each disease. However, during initial evaluation attempts, models would often struggle to correctly compensate for the oversaturation of these class '3' samples. A more muted class imbalance was introduced as an attempt to combat this. A width and height shift of a factor of 0.2 were applied to each image. Shearing was also used, with a shear value of approximately 1.26 radians. Zooming was another augmentation technique used during preprocessing, with a 20% zoom being applied on all input data. Finally, vertical and horizontal translations were applied randomly throughout the data set, with a frequency of 25%. Below are the exact settings used during preprocessing:

```
datagen = ImageDataGenerator(
                    width_shift_range = 0.2,
                    height_shift_range = 0.2,
                    shear_range = 0.2,
                    zoom_range = 0.2,
                    horizontal_flip = True,
                    vertical_flip = True,
                    fill_mode = 'nearest',
                    rescale = 1.0/255.0)
```

Figure 9: ImageDataGenerator settings

After augmentations were applied, input was separated into training and validation sets. A

[2] Keras is a machine learning library used in the evaluation portion of this classification task. It is provided by TensorFlow.

ratio of 80/20 was used, leaving over 17000 image tensors in the training set, with roughly 6000 image tensors left for testing purposes. Once this step was complete, the input data was ready for the next stage of evaluation

2.3. The Model

Convolutional Neural Networks are effectively employed as a computationally efficient method of achieving a high degree of classification accuracy in image classification tasks. For evaluation of the Cassava Leaf Disease Classification dataset, the EfficientNet CNN architecture provides industry-leading results when used as a base model during the training process. EfficientNetB3 uses a compound coefficient to gradually scale the number of trainable parameters in a more principled and structured way. This reduces the number of overall parameters needed to achieve state-of-art classification accuracy. EfficientNet is primarily constructed using convolutional and depth-wise convolutional layers, with global average pooling, batch normalization and skip connection being used for supplemental regularization. A visualization of the entirety of the model is provided by Vardal, Argwal:

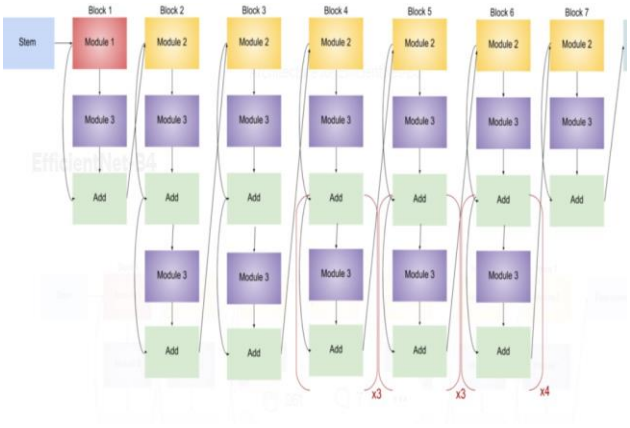

Figure 10: Complete architecture of EfficientNetB3

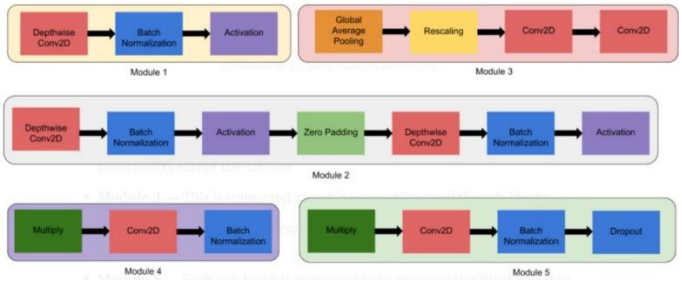

Figure 11: Module blueprints for EfficientNet models

Keras provides an instance of the EfficientNetB3 architecture that can be employed for use on the preprocessed Cassava Leaf Disease Classification training and validation sets. Weights for the EfficientNet model were initialized with using the glorot uniform algorithm, with Rectified Linear activation functions applying nonlinear data transformations to receive meaningful data translation between layers. To achieve a classification, 2 fully connected output layers are needed, with a SoftMax activation function being used to convert inputs into a label prediction. Once these layers are added, the model is ready for the fitting process.

```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
efficientnetb3 (Functional)  (None, 10, 10, 1536)      10783535
_________________________________________________________________
global_average_pooling2d_8 ( (None, 1536)              0
_________________________________________________________________
flatten_8 (Flatten)          (None, 1536)              0
_________________________________________________________________
dense_24 (Dense)             (None, 256)               393472
_________________________________________________________________
dropout_16 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_25 (Dense)             (None, 32)                8224
_________________________________________________________________
dropout_17 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_26 (Dense)             (None, 5)                 165
=================================================================
Total params: 11,185,396
Trainable params: 11,098,093
Non-trainable params: 87,303
_________________________________________________________________
```

Figure 12: Evaluation model with the total number of trainable parameters

2.4. Metrics

The metric used in the Cassava Leaf Disease Classification Competition is categorical accuracy. Correct predictions are divided by total predictions to give an accuracy score to a

model. An additional metric that can be used in evaluation is 'Top K' categorical accuracy. Instead of determining success in a binary fashion (correct/incorrect), top k categorical accuracy measures success by evaluating if the correct label is with the specified ('k') range in the model's prediction hierarchy for any given test input. Categorical accuracy is the only measure used for reporting results on the Kaggle leaderboards, so the results reported in subsequent sections of this report are values that represent the model's categorical accuracy.

2.5 Hyperparameter Optimization

During the process of fitting the model, hyperparameter optimization was preformed to improve categorical accuracy. Due to the length of the training cycles (upwards of 12 hours) and Kaggle's' GPU use quotas, limited experimentation was possible. The number of epochs used during evaluation was 10. The Adam optimizer, which uses a variation of Stochastic Gradient Descent that captures the moment of each step to improve the speed of convergence was used to train the model. A learning rate of 1E-3 was used as an input to the optimizer during the training process. The batch size that was used during evaluation was 32. Given the significant time needed to complete each epoch, the batch size would be the first parameter I would consider adjusting in further optimizations. A larger batch size would require less steps during each epoch for the model to fit its parameters. This would speed up overall training times and allow for optimizations of the other hyperparameters to be tested.

3. RESULTS

3.1 Baseline Comparison

Logistic Regression algorithms can be used as an alternative to deep learning in various image classification tasks. Logistic regression is a linear set of operations used to estimate the relationship between a target value and a predicted value. This allows for comparisons between an input value, and how likely each pixel corresponds with a pixel of any given class. A logistic regression model uses Sigmoid activation functions and MSE as its loss function. Given the similarities between images of different classes in the Cassava Leaf Disease Classification dataset, a logistic regression model struggles to capture the nuanced complexity of the input data as well as deep learning models with millions of trainable parameters. Using a logistic regression model from the SKLearn machine learning library, a categorical accuracy of 0.4712 was achieved on the validation set.

3.2 EfficientNetB3 Results

The results after 10 epochs using the EfficientNetB3 base model and the Keras Dense output layers was a categorical accuracy of 0.6342 on the validation set. This was achieved after the 7th epoch, with overfitting taking place at around this point.

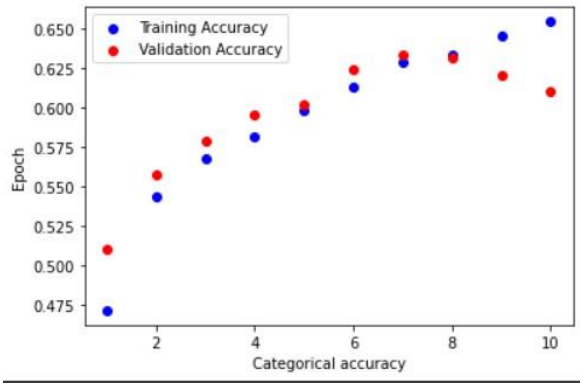

Figure 13: Training/Validation accuracy after each epoch.

4. DISCUSSION

While the model improved on the baseline, a categorical accuracy of 0.6342 on the validation set is significantly lower than I had hoped. Further optimization of the model can be

achieved with the adjustment of hyperparameters, the addition of new convolutional layers, additional dataset augmentation etc. This model and the training parameters used in this report achieve a higher level of accuracy than would be possible by simply predicting every input to be from class '3'. This was initially a concern due to the outsized presence of samples from that class, indicating that the model and the augmentations were successful in creating enough variance in the data for the model to make a range of label predictions. With the alleviation of some time constraints, further adjustments can be made on the model before a final submission is made to the competition page on Kaggle. Through these proposed adjustments, I hope to see improvements in the categorical accuracy, with a goal score of 0.75 seeming reasonably achievable.

5. CONCLUSION

The security of the Cassava plant is vital for the people who rely on it for food and it's many other of its unique applications. With the threat of crop loss due to disease ever present, new solutions are constantly being explored. Computer Vision systems present an exciting new alternative to traditional disease detection methods. In the hopes of exploring new approaches, this dataset was provided to data scientists across the globe. This report, while limited in scope, proves that through the use of techniques like data augmentation, CNNs and other similar learning models can be successfully applied to Cassava Leaf Disease detection tasks. This provides an optimistic proof that the practical widespread application of these systems is possible in near future.

6. REFERENCES

Ajitesh, Kumar. "Linear vs Logistic Regression: Differences, Examples". Data Analytics. https://vitalflux.com/linear-vs-logistic-regression-differences-examples/. Accessed March 30th, 2022

Argwal, Vardan. "Complete Architectural Details of all EfficientNet Models". Towards Data Science. https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet models-5fd5b736142. Accessed March 30th

Eunus, Salman I. Cassava Leaf Disease Classification 2 (Notebook). 2021, https://www.kaggle.com/salmaneunus/cassava-leaf-disease-classification-2/notebook. Accessed February 17th, 2022.

Mingxing Tan, Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". Proceedings of the 36th International Conference on Machine Learning. 28 May 2019, http://proceedings.mlr.press/v97/tan19a.html. Accessed February 17th, 2022.

Oliveira, Dimitre. Cassava Leaf - Supervised Contrastive Learning (Notebook). 2021, https://www.kaggle.com/dimitreoliveira/cassava-leaf-supervised-contrastive-learning/notebook. Accessed March 15th, 2022

Ramcharan, Amanda et al. "Deep Learning for Image-Based Cassava Disease Detection." Frontiers in plant science vol. 8 1852. 27 Oct. 2017, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5663696/. Accessed February 17th, 2022.

Takimoglu, Ayesegul. "What is Data Augmentation? Techniques & Examples in 2022." AI Multiple. https://research.aimultiple.com/data-augmentation/. Accessed March 14th, 2022.

Unknown Author. "Cassava". Agricultural Research Council. 2014, https://www.arc.agric.za/arc iic/Pages/Cassava.aspx. Accessed February 17th, 2022