Cassava Leaf Disease Classification

Jingxiao Tian
University of California San Diego, La Jolla, CA 92093-0238

***Abstract*—The goal of this project is to identify the type of disease present on a Cassava Leaf image. Cassava is Africa's second-largest food source but viral diseases are major sources of poor yields. Existing disease detection methods mainly depend on human experts' eye inspection. Therefore, new cassava disease detection methods are required to help to solve this situation. Our group propose a solution through image classification via conventional network to detect and classify specific leaf disease accurately, efficiently and easy to use.**

II. Introduction

As the second-largest provider of carbohydrates in Africa, cassava is a key food security crop grown by smallholder farmers because it can withstand harsh conditions. At least 80% of household farms in Sub-Saharan Africa grow this starchy root, but viral diseases are major sources of poor yields. With the help of data science and machine learning, farmers can quickly identify diseased plants , potentially saving their crops before they inflict irreparable damage. Existing disease detection methods requires farmers to solicit the help of government-funded agricultural experts to visually inspect and diagnose the plants. This suffers from being labor-intensive, inefficient and costly since it requires expertise.

We propose a solution that using CNN to classify each cassava image into 5 disease categories. The input to our algorithm is an RGB Image. We then use a CNN architecture model to output a probability and using softmax to map it to a categorical variable representing the leaf disease label.

III. Related Work

Recent works have reveal that deep learning approaches, that mainly use convolutional neural networks (CNN), were extremely palmy at image processing problems, supported their capability to extract efficient features from figures [1] [2]. What is more, their result is better with evolutionary computing [3]. Literature has proved that deep learning algorithms have given wide application with researchers for the detection of totally different plant diseases [4] like banana [5], tomato [6], and rice [7]. This can be count to the effectuality of these methods in dealing with classification issues and image segmentation.

For example, Ramcharan [8] used cassava figures to modify a CNN to distinguish diseases. An accumulative accuracy of 93% was reached by the most successful model. Their results show that transfer learning methods give a fast, cheap, and convenient approach for the detection of infection. Another mobile phone CNN model [9] reached an precision of 94%. Sambasivam [10] used predictive ML models applying image augmentation obtained an accuracy of 93% in cassava detection. An image analysis method for the detection of unhealthy leaf in cassava was found by Abdullakasim [11]. They applied an artificial neural network (ANN) to find healthy and unhealthy cassava. Their ANN model properly classified 79.23% of the infected leaves and 89.92% of healthy plants. Deep learning was accustomed to expedite a fast and engaging exploration of information in farming. Their method has the potential to help operators to improve productivity. Ramcharan [8] used deep CNN to find forms of cassava diseases. The general result of their technique is good in the accuracy and confusion matrix, but the disadvantage is that the result was solely based on Support Vector Machine (SVM) and K Nearest Neighbor (KNN). Abayomi [12] employed data augmentation with adjusted color value distribution to increase the training dataset and to modify the CNN to find essential features based on color. Their method is predicated on the Chebyshev orthogonal functions and applies the MobileNetV2 for classification. Radial basis function neural network (RBPNN) for cassava classification was purposed by Capizzi [13]. Different classification approach like Adaptive Artificial Neural Network (AANN) were used by Wozniak [14].

TABLE I
Summary of Related Work

Methods	Results
Fully connected neural network	79.23%
InceptionV3 CNN	93%
Single Shot Multibox model	94%
Custom 15-layer CNN	95%
MobileNetV2 CNN	99.7%
VGG16 CNN	95%

IV. Problem Statement

We formulate the problem as a image classification problem. Dataset $\mathcal{D} := \{(x_i, y_i)\}$ where x_i is the i-th image in the dataset and y_i is the i-th label which is a categorical variable.

A. *Cassava Leaf Disease Dataset*

Cassava Leaf Disease Dataset can be found on Kaggle [15]. This dataset consists of 21,367 labeled RGB images collected during a regular survey in Uganda. Most images were photoed by farmers from their own gardens, and annotated by expert at the National Crops Resources Research Institute (NaCRRI). This is in a format that most realistically represents what farmers would need to diagnose in real life.

There are 5 class categories with different number of samples:

- 0: Cassava Bacterial Blight (*1087 samples*)
- 1: Cassava Brown Streak Disease (*2189 samples*)
- 2: Cassava Green Mottle (*2386 samples*)
- 3: Cassava Mosaic Disease (*13158 samples*)
- 4: Healthy (*2577 samples*)

From Figure 1, we saw that class 3 has more samples in the dataset than any of other classes. It approximately more than half of the size of the dataset. Therefore, we have a imbalanced dataset. Technique that used to deal with the imbalanced dataset will be discussed in Technical Approach. We split the data into training and validation dataset with ratio $8:2$ and in stratify fashion.

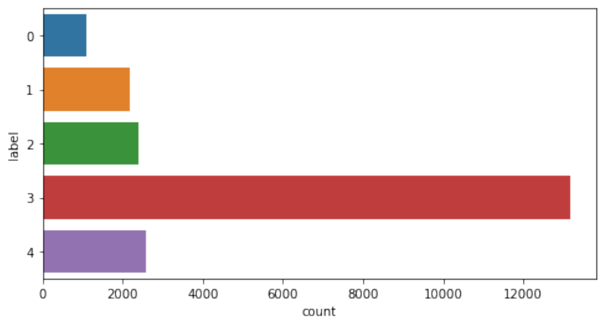

Fig. 1. *Class Distribution*

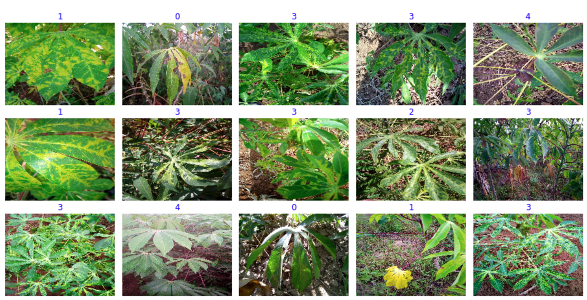

Fig. 2. *Image Dataset*

Each image in the original dataset show in fig 2 has shape (H, W, C) where $H = 800$ is the height, $W = 600$ is the width and $C = 3$ denotes 3-color channel(R, G, B).

V. Technical Approach

A. Dealing with imbalanced datasets

Our dataset is extremely imbalanced, and it is often the case in real world. This leads model trained to have high tendency to be biased toward the class 3 as shown in Figure 1.

To solve this problem, a widely adopted technique is called resampling. It consists of removing samples from the majority class (under-sampling) or adding more examples from the minority class (over-sampling) illustrated in Figure 3 [16].

Despite the advantage of obtaining balancing classes, these techniques have their disadvantages well. Over-sampling, naive approach, is to duplicate random records from the minority class, which can cause overfitting. On the other hand,

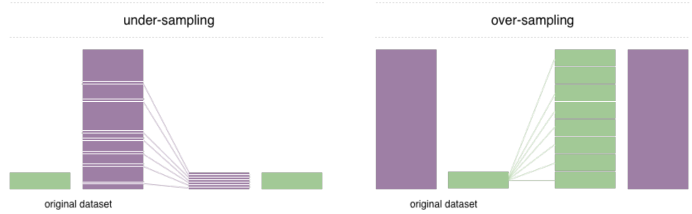

Fig. 3. *Under-sampling and Over-sampling*

under-sampling naively involves removing random records from the majority class, which can cause loss of information.

Therefore, a mixture of resampling techniques and data augmentation are combined to form a balanced dataset without overfitting and loss of information illustrated in Figure 4 [16]

- Re-balance the class distributions when sampling from the imbalanced dataset
- Estimate the sampling weights
- Avoid creating a new balanced dataset
- Mitigate overfitting when it is used in conjunction with data augmentation techniques

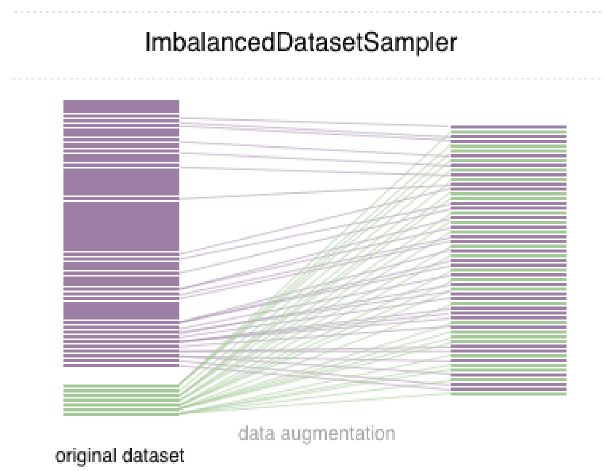

Fig. 4. *Combination of Resampling and Data Augmentation*

B. Image Prepossessing and Data Augmentation

Image augmentation is a process of creating new training examples from the existing ones with lightly change in the original image. We applied similar Image augmentation using Albumentations [17] to both training and validation dataset. Results of images after augmentation are demonstrated in Figure 5

- Resize the image to $(300, 300, 3)$ and center crop it to $(256, 256)$.
  We reduce the size of our data without losing too much information
- Affine Transform (Horizontal/Vertical Flip, ShiftScaleRotate).
  Training a model with available and transformed images improves the performance and generalization.

- HSV Saturation and Enhancement
  Allow to extract some useful information in image photograph under extreme light condition.
- Standardization with Zero Mean and Standard Deviation one

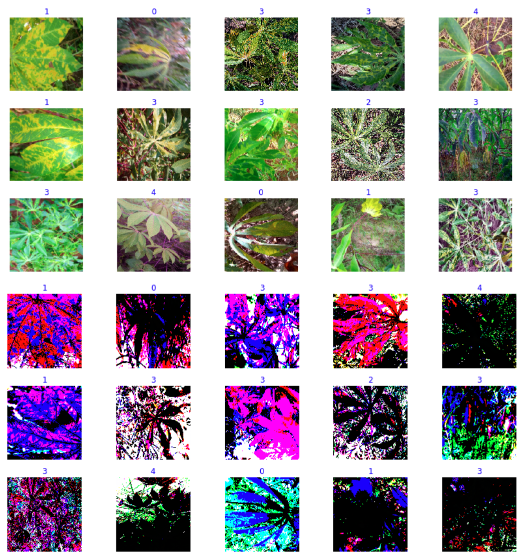

Fig. 5. *Image Augmentation and Normalization*

C. Feature Extraction

One common feature extraction technique is to feed the image to a pre-trained conventional neural network, and use the representation for that particular image in the intermediate layers of the neural network. We use ResNet for visual feature extraction.

D. ResNet Model

Deep Residual Network(ResNet) [18] makes it possible to train up to hundreds of layers and still achieves good performance. According to the universal approximation theorem, a neural network with a single layer is sufficient to approximate any function. However, the layer might be huge and the network is prone to overfitting the data. Therefore, there is a common trend that our network architecture needs to go deeper. However, increasing network depth simply via stacking layers are hard to train because of the vanishing gradient problem.

The core idea of ResNet is introducing a so-called "identity shortcut connection", or identity mapping that skips one or more layers, as shown in the figure 9 [18] below:

However, sometimes x and $\mathcal{F}(x)$ do not have the same dimension. A convolution operation will introduced to shrinks the spatial resolution of the image.

$$y = \mathcal{F}(x, \{\mathcal{W}_i\}) + \mathcal{W}_s + x \tag{1}$$

where $\mathcal{W}_s$ can be implemented with 1x1 convolutions.

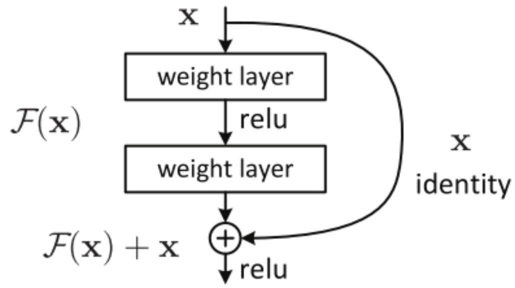

Fig. 6. *ResNet Block*

E. Transfer Learning and Fine Tuning

In practice, very few people train an entire CNN from scratch (with random initialization), because it is relatively rare to have a dataset of sufficient size. Instead, it is common to pretrain a ConvNet on a very large dataset and then use it either as an initialization, or a fixed feature extractor. We will applied transfer learning via Pytorch [19] pretrain ResNet model as initialization and training it with our Cassava datast.

VI. Results

A. ResNet Model Results

According to above introduction about ResNet model, three different types of ResNet model have been used in our project, ResNet18, ResNet34, ResNet50. The results from ResNet18 and ResNet50 were shown in the figure 9 below:

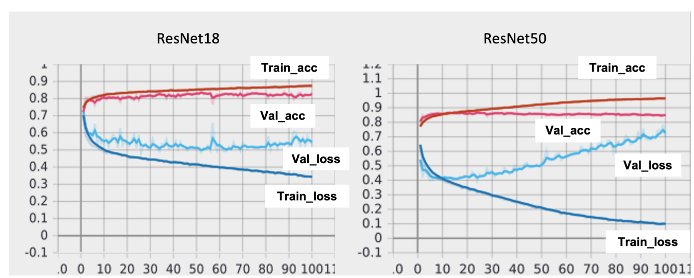

Fig. 7. *left:ResNet18 accuracy; right: ResNet50 accuracy*

After 100 epoch, both of results showed overfitting effect, such that the validation accuracy of ResNet18 after 100 epoch, which reached to constant after 60 epochs, was slightly above 80%, but training accuracy kept increasing as epoch number increasing, which could reach up to 90%. More evident overffiting effect was shown from ResNet50. The training accuracy could reach up to near 100%, but the validation accuracy was only around 85%.

B. ResNet Model with Dropout

Based on above results, above 80% validation accuracy wad obtained but with overfitting. Hence, add one dropout layer before the ResNet Model was applied. We added 50% dropout layer for ResNet50 and 30% dropout layer for ResNet34. The results were shown in the figure 9 below:

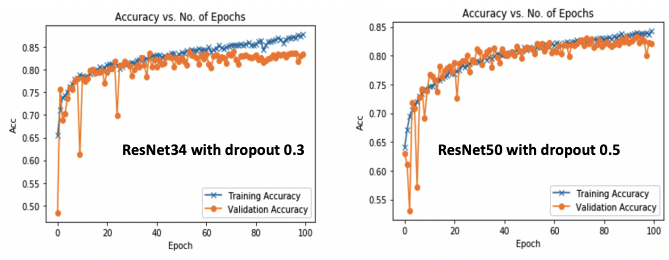

Fig. 8. *left:ResNet34 accuracy; right: ResNet50 accuracy*

With 50% dropout for ResNet50, the overfitting has been modified after 100 epochs, but the validation accuracy hasn't been increased, still around 80-85%. We tried ResNet34 with 30% droupout layer, but overfitting was still there after 60 epochs. The final validation accuracy is only slightly above 80%.

C. Results from Other Models

Besides ResNet model, we tried two more different models, MobileNetV2&MnasNet [20] [21]. The results from two models were shown in the figure 9 below:

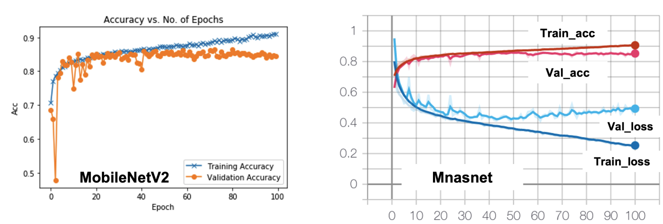

Fig. 9. *left:MobileNetV2 accuracy; right: MnasNet accuracy*

By analyzing above results, there were still overfitting effect in both models. The final validation accuracy from MobileNetV2 after 100 epochs is 84.4%, and 85.25% from MnasNet after 100 epochs. It looks like very similar if we only compared training and validation accuracy from both models.

VII. Conclusion

In this project, we trained several different models to classify cassava image into 5 disease categories using data set from Kaggle. ResNet18&50 were first trained with overfitting effect. By adding dropout layer before ResNet model with 30&50% were used later. The overfitting effect has been improved with slightly sacrificed accuracy. MobileNetV2 and MnasNet models were trained without dropout layer, both of them gave around 85% validation accuracy with slightly overfitting effect after 100 epochs. According to all of our results, around 85% validation accuracy was the best results. Due to extreme imbalanced training data set, overfitting effect was inevitable although resampling process had been applied. In general, 85% accuracy should be adequate based on current training cost and data set.

References

[1] D. Połap and M. Woźniak, "Bacteria shape classification by the use of region covariance and convolutional neural network," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–7.

[2] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.

[3] A. Darwish, A. E. Hassanien, and S. Das, "A survey of swarm and evolutionary computing approaches for deep learning," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1767–1812, 2020.

[4] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, p. 1419, 2016.

[5] J. Amara, B. Bouaziz, and A. Algergawy, "A deep learning-based approach for banana leaf diseases classification," *Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband*, 2017.

[6] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.

[7] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.

[8] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers in plant science*, vol. 8, p. 1852, 2017.

[9] A. Ramcharan, P. McCloskey, K. Baranowski, N. Mbilinyi, L. Mrisho, M. Ndalahwa, J. Legg, and D. P. Hughes, "A mobile-based deep learning model for cassava disease diagnosis," *Frontiers in plant science*, vol. 10, p. 272, 2019.

[10] G. Sambasivam and G. D. Opiyo, "A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks," *Egyptian Informatics Journal*, vol. 22, no. 1, pp. 27–34, 2021.

[11] W. ABDULLAKASIM, K. POWBUNTHORN, J. UNARTNGAM, and T. TAKIGAWA, "An images analysis technique for recognition of brown leaf spot disease in cassava," *Tarım Makinaları Bilimi Dergisi*, vol. 7, no. 2, pp. 165–169, 2011.

[12] O. Abayomi-Alli, R. Damaševičius, R. Misra, and R. Maskeliūnas, "Cassava disease recognition from low quality images using enhanced data augmentation model and deep learning," *Expert systems. Figshare. DOI*, vol. 10, p. m9, 2021.

[13] G. Capizzi, G. LO SCIUTO, C. Napoli, E. Tramontana, and M. WOŹNIAK, "A novel neural networks-based texture image processing algorithm for orange defects classification." *International Journal of Computer Science & Applications*, vol. 13, no. 2, 2016.

[14] M. Woźniak and D. Połap, "Adaptive neuro-heuristic hybrid model for fruit peel defects detection," *Neural Networks*, vol. 98, pp. 16–33, 2018.

[15] M. U. A. Lab, "Cassava leaf disease classification," *Artificial Intelligence and Data Science*, 2021.

[16] Ming, "imbalanced-dataset-sampler," https://github.com/ufoym/imbalanced-dataset-sampler, 2019.

[17] E. K. V. I. I. A. Buslaev, A. Parinov and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *ArXiv e-prints*, 2018.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[21] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.