

# **Classification of Cassava leaf Diseases Using Deep Neural Networks**



By

**Alina Maryum**

Registration Number: 00000319921

Supervised By

**Dr. Muhammad Usman Akram**

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING (E&ME),

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY (NUST),

ISLAMABAD

September 2021

# **Classification of Cassava leaf Diseases Using Deep Neural Networks**

By

**Alina Maryum**

Registration Number: 00000319921

A thesis report submitted in partial fulfillment of the requirements  
for the degree of MS in Software Engineering

Thesis Supervisor:

**Dr. Muhammad Usman Akram**

Supervisor's Signature:

---

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING (E&ME),

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY (NUST),

ISLAMABAD

September 2021

## **DECLARATION**

I declare that this research work titled, “Cassava Leaf disease Classification Using Deep Neural Networks” is my own work and it has not been submitted for evaluation anywhere else. All the material from other sources used in this report has been appropriately cited.

Student’s Signature:

Alina Maryum-319921

MS-19-CSE

---

## **PLAGIARISM REPORT**

This thesis report has been checked for Plagiarism. Attached is the Turnitin report checked by Supervisor.

Student's Signature:

Alina Maryum

00000319921

---

Signature of Supervisor:

---

## **COPYRIGHT STATEMENT**

The student author owns the copyright of the text of this paper. Full or partial copies (in any form) can only be made according to the author's instructions and stored in the library of NUST institute of Electrical and Mechanical Engineering (CEME). The librarian may be asked for details. This page must be part of any copy of the thesis made. No more copies may be made (in any way) without author's (written) permission.

All intellectual property rights given above belong to NUST institute of Electrical and Mechanical Engineering (CEME) and unless otherwise agreed in advance, they must not be provided to third parties for use, without written permission. CEME will list the terms and conditions of any such agreement.

For further information on the terms of possible disclosure and use, please refer to the library of NUST institute of Electrical and Mechanical Engineering (CEME), Islamabad.

## **ACKNOWLEDGEMENT**

I am grateful to my Creator, Allah Almighty, for guiding me through this research work with each step and every new thought. I am thankful to my parents for their constant support and endless efforts. I am also very grateful to my supervisor for his guidance, support and help that enabled me to achieve the goals of this research work. Finally, I would like to thank everyone who gave me valuable help, directly or indirectly with my dissertation.

.

## TABLE OF CONTENTS

<b>Abstract</b> .....	<b>13</b>
<b>Chapter 1: Introduction</b> .....	<b>14</b>
Cassava Origin .....	14
Benefits of Cassava over other crops .....	16
Disadvantages.....	16
Cassava Diseases.....	17
Remedies .....	18
Goals and Objectives.....	18
Motivation .....	18
Problem .....	19
Report organization .....	19
<b>Chapter 2: Literature Review</b> .....	<b>21</b>
Traditional Methods .....	21
Convolutional Neural Networks.....	23
Research Gaps .....	28
Contribution .....	29
<b>Chapter 3: Materials</b> .....	<b>30</b>
Cassava Image Dataset.....	30
Cassava Bacterial Blight (CBB).....	32
Cassava Brown Streak Disease (CBSD) .....	32
Cassava Green Mottle (CGM2) .....	33
Cassava Mosaic Disease (CMD).....	33
Tools and Languages.....	34
<b>Chapter 4: Methodology</b> .....	<b>35</b>
Proposed Framework.....	35
Image Segmentation .....	36
Color Index Vegetation Extraction .....	37
Image Augmentation (For Segmentation) .....	37
UNet Architecture.....	38

Image Augmentation .....	41
EfficientNet Architecture.....	43
<b>Chapter 5: Experimentation &amp; Results .....</b>	<b>48</b>
Experimentation .....	48
Plant pathology 2021 .....	48
Training .....	48
Hyper Parameters Setting .....	49
Performance Metrics .....	49
Segmentation Results .....	50
Classification Results .....	54
<b>Chapter 6: Conclusion &amp; Future Work.....</b>	<b>71</b>
<b>References .....</b>	<b>73</b>



## LIST OF FIGURES

<b>Figure 1-1-</b> Cassava leaf images .....	14
<b>Figure 1-2-</b> Cassava Production Statistics (FAO STATS 2016 Database) [34] .....	15
<b>Figure 1-3-</b> Sample images of some Cassava Leaf Diseases .....	17
<b>Figure 2-1-</b> Image with ORB interest key points identified [5].....	22
<b>Figure 2-2-</b> System Architecture [7] .....	23
<b>Figure 2-3-</b> Overall accuracy for transfer learning using three machine learning methods [8]....	24
<b>Figure 2-4-</b> Process Illustration of Faster R-CNN for Object Detection [10].....	26
<b>Figure 2-5-</b> Application of distinct block Processing o increase sample size of minority classes [12].....	26
<b>Figure 3-1-</b> Sample Images from each of the 4 diseases and healthy leaf (from left to right).....	31
<b>Figure 3-2-</b> Samples Distribution among 5 classes.....	31
<b>Figure 3-3-</b> Leaves with Cassava Bacterial Blight Disease .....	32
<b>Figure 3-4-</b> Leaves with Cassava Brown Streak Disease .....	32
<b>Figure 3-5-</b> Leaves with Cassava Green Mottle Disease .....	33
<b>Figure 3-6-</b> Leaves with Cassava Mosaic Disease.....	33
<b>Figure 4-1-</b> Flowchart of Proposed Framework.....	36
<b>Figure 4-2-</b> Original Image (Top) and Augmentation results on Healthy Cassava Leaf (bottom)38	
<b>Figure 4-3-</b> UNet Architectural Diagram.....	40
<b>Figure 4-4-</b> Augmentation Results with Original image (left) and transformed images (right) ...	42
<b>Figure 4-5-</b> Samples Distribution among 5 classes in Balanced Cassava 2020 Dataset.....	42
<b>Figure 4-6-</b> EfficientNet-B0 Architectural Diagram.....	46
<b>Figure 5-1-</b> Sample Image (left) and segmented image (right) from Plant Pathology 2021 dataset .....	49
<b>Figure 5-2-</b> Cassava bacterial Blight diseased leaves original images (top) and segmented images (bottom).....	51
<b>Figure 5-3-</b> Segmented Cassava Bacterial Blight Diseased Leaves extracted Using changed CIVE formula .....	51
<b>Figure 5-4-</b> Cassava Brown Streak Diseased Leaves Original Images (top) and Segmented Images (bottom).....	52

<b>Figure 5-5</b> -Cassava Green Mottle Diseased Leaves Original Images (top) and Segmented Images (bottom).....	52
<b>Figure 5-6</b> -Cassava Mosaic Diseased Leaves Original Images (top) and Segmented Images (bottom).....	53
<b>Figure 5-7</b> - Healthy Cassava Leaves Original Images (top) and Segmented Images (bottom)...	53
<b>Figure 5-8</b> -Confusion Matrix for binary classification .....	54
<b>Figure 5-9</b> - A comparison of class-wise accuracies obtained from EfficientNet-B4 on original data and final segmented data .....	61
<b>Figure 5-10</b> - Class-wise Accuracy Score obtained from EfficientNet-B0 on final Segmented Dataset.....	65
<b>Figure 5-11</b> -Samples Distribution among 5 classes (Cassava 2019 Dataset).....	68
<b>Figure 6-1</b> -Poorly Segmented Diseased Leaf Image .....	72

## LIST OF TABLES

<b>Table 1-1-</b> Cassava Diseases and their Causes.....	17
<b>Table 1-2-</b> Under Five Children Statistics Table (National Nutrition Survey 2018) [27] .....	19
<b>Table 2-1-</b> Number of hidden neurons and classification accuracy of healthy and diseased cassava leaves [6].....	22
<b>Table 2-2-</b> Effect of different input image dimensions on CNN model accuracy [9].....	25
<b>Table 2-3-</b> Comparison of different Machine Learning and Deep Learning techniques for Cassava Leaf disease Classification.....	27
<b>Table 3-1-</b> Frequency of Images in each Output Class .....	31
<b>Table 4-1</b> - EfficientNet-B0 Baseline Network [15].....	44
<b>Table 4-2</b> - EfficientNet-B4 Network Scaled from Baseline Network .....	45
<b>Table 4-3</b> - EfficientNet-B6 Network Scaled from Baseline Network .....	45
<b>Table 5-1-</b> Experimentation Details .....	55
<b>Table 5-2-</b> Normalized Confusion Matrix of Original Dataset .....	56
<b>Table 5-3-</b> Normalized Confusion Matrix of Segmented dataset Using Original Formula (eq. 4.1) on Cassava Healthy Leaves Dataset without Augmentation .....	56
<b>Table 5-4-</b> Normalized Confusion Matrix of Segmented dataset Using Original Formula (eq. 4.1) on Cassava Healthy Leaves Dataset with Augmentation .....	57
<b>Table 5-5-</b> Normalized Confusion Matrix of Segmented dataset Using Changed Formula (eq. 4.2) on Cassava Healthy Leaves Dataset with Augmentation .....	58
<b>Table 5-6-</b> Normalized Confusion Matrix of Segmented Dataset Using Original Formula (eq. 4.1) on Combined Dataset (Cassava Healthy Leaves + Plant Pathology Dataset) with Augmentation .....	58
<b>Table 5-7-</b> Normalized Confusion Matrix of Segmented Dataset Using Changed Formula (eq. 4.2) on Combined Dataset (Cassava Healthy Leaves + Plant Pathology Dataset) with Augmentation obtained from EfficientNet-B4 .....	59
<b>Table 5-8-</b> Class-Wise Accuracy Results of all Experiments .....	60
<b>Table 5-9-</b> Overall Validation Accuracy and Loss of all Experiments .....	60
<b>Table 5-10</b> – Class wise result of original and segmented datasets in descending order of increase in accuracy .....	61

<b>Table 5-11-</b> Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B0.....	62
<b>Table 5-12-</b> Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B1.....	62
<b>Table 5-13-</b> Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B2.....	62
<b>Table 5-14-</b> Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B3.....	63
<b>Table 5-15-</b> Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B5.....	63
<b>Table 5-16-</b> Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B6.....	63
<b>Table 5-17-</b> Overall Validation Accuracy and Loss from EfficientNet Models Family.....	64
<b>Table 5-18-</b> Class-Wise Accuracy Results obtained from EfficientNet Models Family .....	64
<b>Table 5-19-</b> 7-Fold Cross Validation Results obtained from EfficientNet-B0 on Cassava 2020 dataset .....	66
<b>Table 5-20-</b> 7-Fold Cross Validation Results obtained from EfficientNet-B0 on Balanced Cassava 2020 dataset .....	66
<b>Table 5-21-</b> Training time and the dataset size for classification using EfficientNet-B0.....	67
<b>Table 5-22-</b> 7-Fold Cross Validation Results obtained from EfficientNet-B0 on Cassava 2019 dataset .....	68
<b>Table 5-23-</b> Comparison of Proposed Work with Previous Work on Cassava 2019 Dataset.....	69
<b>Table 5-24-</b> Class wise comparison of Proposed work with literature using Kaggle 2019 Cassava Disease Dataset .....	70

## ABSTRACT

In recent years, deep learning has gained much popularity over traditional machine learning techniques in terms of accuracy and precision when trained on substantial amount of data. In this research, a state-of-the-art deep learning technique has been employed for classification and prediction of cassava leaf diseases. Being the second largest producer of carbohydrates in the world, cassava plant has become an important source of calories for people in tropical regions, but it is highly susceptible to viral, bacterial, and fungal attacks resulting in stunted plant growth and hence the yield. The dataset that is used in this research is taken from a Kaggle competition containing 21,397 images of cassava plant leaves belonging to 5 classes: Cassava Bacterial Blight, Cassava Brown Streak Disease, Cassava Green Mottle, Cassava Mosaic Disease and Healthy leaf. In this research work, EfficientNet models were trained using transfer learning approach. Further, to remove background noise, Segmentation was performed using U-Net to extract only the leaves from images. Since the dataset was imbalanced, detailed image augmentation was also performed to increase the sample size of minority classes. Our model provided reasonable performance with balanced dataset giving 89.97% accuracy. However, original (imbalanced) dataset results were also comparable to balanced dataset giving mean f1-score of 0.89 and mean accuracy score of 89.73% plus 0.82 standard deviation on segmented dataset trained on EfficientNet model B0 using 7-fold cross validation. For comparison purpose, Kaggle 2019 dataset for cassava disease classification was used, that gave mean accuracy score of  $89.41 \pm 1.62$  using 7-fold cross validation and f1-score of 0.9 leading all state-of-the-art results on same dataset.

# Chapter 1

## INTRODUCTION

Cassava, scientific name “*Manihot esculenta*” also known as manioc, tapioca or yuca is a starchy root crop native to North America, However, it is mostly grown in Africa and some parts of Asia as well. It is cultivated throughout the tropical world for its tuberous roots [30], from which cassava flour, breads, tapioca, a laundry starch, and other beverage are derived [22]. It is the third largest producer of carbohydrates after wheat and corn. However, in terms of carbohydrates production, its production rate is about 40% higher than rice and 25% more than corn [32]. This is because of the plant’s ability to resist severe weather conditions. As a result, it yields more production in low cost becoming major source of income for many poor farmers. However, it is highly vulnerable to many viral, fungal, and bacterial attacks limiting production of crops and hence the yield. According to the Food and Agriculture Organization of the United Nations (FAO), cassava is currently the third most substantial supplier of calories in the tropics, after wheat and corn, making it a source of livelihood for more than 800 million people and hence, it is also considered as a source of income generation in Africa, Asia, and Latin America [34].



**Figure 1-1-** Cassava leaf images

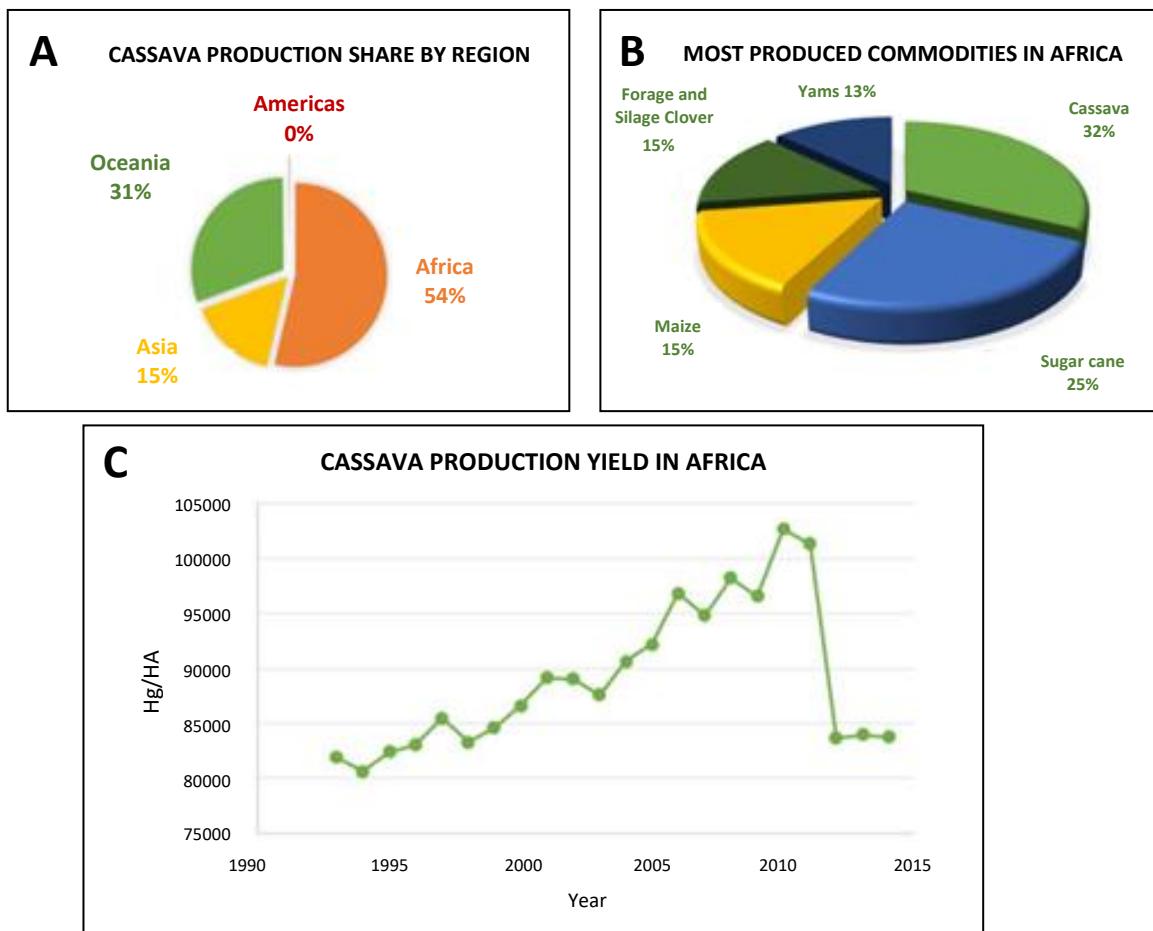
### 1.1. CASSAVA ORIGIN:

Though native to North America, cassava was introduced to Africa by Spanish and Portuguese traders back in 16th century [21]. Since then, cassava has grown to become a staple food for most

African countries. Approximately, half of the world's cassava current production comes from Africa where it is cultivated in around 40 countries [33]. It is the major source of carbohydrates, being a staple food for over 800 million people around the globe.

Not only its roots are beneficial, but its leaves are also an important source of proteins, minerals, and vitamins [24]. Cassava has many health benefits as well. It can be used to treat mild fever. Cassava roots and leaves, when used together, increases the ability to relieve fever. It also improves health by boosting immunity and regulating digestion. The root of the plant is rich in vitamin C [23], while the leaves contain compounds beneficial to the skin and metabolism [31]. Its leaves are also edible and can contain up to 25% protein.

Figure 1-2 shows the cassava production statistics as given by Food and Organization 2016 [34].



**Figure 1-2-Cassava Production Statistics (FAO STATS 2016 Database) [34]**

Based on regions, Africa has the largest cassava production share (more than half) whereas Asia has only 15% share in total production. Among other commodities in Africa, cassava has the largest share of about 32%. Its production yield in Africa is also given in Figure 1-2.

## **1.2. BENEFITS OF CASSAVA OVER OTHER CROPS:**

Cassava has many advantages over other crops. A few are given below:

1. It produces more carbohydrate production than other crops including rice, maize, and wheat etc.
2. It can withstand dry, eroded soils, pests, and severe weather conditions.
3. This starchy vegetable can give high yield and can be kept in the ground for large period as a “reserve food” when other crops are limited [24].
4. Both cassava leaves and roots have nutritional benefits that help keep the developing world fed [37].
5. Cassava is important for both small-scale farmers and larger-scale plantations due to its low requirement for nutrients, ability to tolerate dry conditions and easy low-cost propagation. [10]
6. Most cassava is produced by smallholder farmers living in marginal and fragile environments, and particularly on eroded, acid, and infertile soils. This ability to grow on poor soils, where most other crops would fail, has made cassava an important food security crop [36].

Because of all the above-mentioned benefits, Cassava is sometimes referred to as the “drought, war and famine crop” of the developing world.

## **1.3. DISADVANTAGES:**

Besides many advantages, cassava plant has some disadvantages as well. If taken in raw form or consumed in large amount, cassava has devastating effects on health. This is because raw cassava contains chemicals called cyanogenic glycosides, which can release cyanide in the body when consumed [23]. When eaten frequently, these increase the risk of cyanide poisoning, which may impair thyroid and nerve function. It is associated with paralysis and organ damage and can be even fatal.



#### 1.4. CASSAVA DISEASES:

Despite of its advantages over other crops, its production can be limited because of many viral, bacterial, and fungal attacks on the plant. Most of the diseases caused by such attacks are very devastating, and widespread, limiting production and hence the yield [35]. Other disease includes Cassava Bacterial Blight (CBB), Cassava Brown Streak Disease (CBSD), Cassava Green Mottle (CGM), mite, and pests' diseases etc. [8]. Such diseases produce a variety of symptoms that include mosaic, mottling, deformed and warped leaflets, and an overall reduction in size of leaves and plants. Among those diseases, Cassava Mosaic Disease is the most widespread and the leaves affected by this produce few or no tubers depending on the severity of the disease and the age of the plant at the time of infection. Some of the diseases and their cause is given in Table 1-1. Sample images of some diseases are also shown in figure 1-3.

**Table 1-1-** Cassava Diseases and their Causes

<b>Fungal Diseases</b>	<b>Bacterial Diseases</b>	<b>Viral Diseases</b>
Black root and stem rot	Bacterial Stem rot	Cassava Mosaic
Brown Leaf Spot	Bacterial Stem Gall	Brown Streak disease
Blight leaf spot	Bacterial Blight	Green Mottle
Cassava Ash	Bacterial Wilt	Vein Mosaic

Precautionary steps have been taken to control the spread of these diseases, but the two diseases Cassava Mosaic Disease and Cassava Brown Streak Disease have become the largest constraints to cassava production and food security [7] resulting in losses of over \$1 billion every year.



**Figure 1-3-** Sample images of some Cassava Leaf Diseases

### **1.5. REMEDIES:**

To detect such diseases, presently farmers ask for agricultural experts from their government to visually inspect and diagnose the plants in person. But unfortunately, this process is very labor-intensive, slow, and inefficient to use [5]. If not timely detected and treated, farmers could be losing a large portion of their harvest. So, such outbreaks can cost farmers a lot. One of the remedies followed by farmers is to burn the infected plants to prevent further spread of the disease leading to affecting a country's economy greatly [29].

### **1.6. GOALS AND OBJECTIVES:**




This research focuses on automating the process of detecting and classifying cassava diseases by applying transfer learning approach to the dataset to make some efficient and accurate prediction of the disease that will help the farmers quickly and accurately classify whether a given plant is suffering from some disease or not so that such diseases can be timely detected and treated before they cause severe damage.

### **1.7. MOTIVATION:**

Among the developing countries, Pakistan has been reported to be greatly suffering from child malnutrition [26] because of lack of macronutrients consumption including proteins, fats, and carbohydrates. Such malnutrition causes headache, fatigue, weakness, nausea and vitamin and mineral deficiencies leading to stunted and underweight growth. According to National Nutrition Survey 2018 [27], 4 out of every 10 children under 5 years are stunted and 2 out of every 10 under 5 years are wasted. The same statistics in terms of percentage is given in Table 1-2.

The table shows that it is the stunted growth that has majorly effected children under five years. This situation is even worse in Sindh where about 50% of children are stunted and the number is increasing [25]. Stunting is mainly caused by poor nutrition intake, lack of access to diverse food and clean water. Being the third largest producer of carbohydrates, cassava production and its utilization in the form of carbohydrates and proteins is of vital importance for Pakistan. Presently in Pakistan, cassava plant is grown on coastland from Sindh to Baluchistan at a small scale [28].

**Table 1-2**-Under Five Children Statistics Table (National Nutrition Survey 2018) [27]

	<b>The Three Faces of Malnutrition</b>	<b>Percentage of Affected Children</b>
	Stunting (Too short for their age)	40%
	Wasting (Too thin for their height)	17.5%
	Overweight	4.8%

Because of its ability to withstand harsh weather conditions and more yield production in low cost, it can be cultivated by local poor farmers that will have the following benefits:

1. Enhancement in productivity
2. Aid in poverty reduction
3. Make up for the lack of carbohydrates deficiency overcoming the above-mentioned health issues.

### **1.8. PROBLEM:**

As already mentioned, cassava plant is highly vulnerable to viral, bacterial, and fungal attacks resulting in many diseases that have very devastating effects on a country's economy. One of the remedies mostly followed by farmers is to burn the infected plant to prevent further spread of the disease. So, there is a need to early detect these diseases before they cause huge loss. For that one of the solutions is to use image processing and deep learning techniques for early prediction and better classification of cassava leaf diseases caused by viral and bacterial attacks.

### **1.9. REPORT ORGANIZATION:**

The report is organized as follows:

- **Chapter 1** gives the introduction about proposed topic, aims, objectives and motivation.

- **Chapter 2** presents the review of state of the art in the context of cassava leaf disease classification techniques.
- **Chapter 3** gives the materials, including dataset and tools used for implementation.
- **Chapter 4** discusses the proposed deep learning methodology for reliable classification of leaf diseases.
- **Chapter 5** discussed the experimentation including the setup used for implementation, results obtained and their discussion
- **Chapter 6** concludes the topic by suggesting some future work that is not under the scope of this research but can be implemented in future.

## **Chapter 2**

### **LITERATURE REVIEW**

In this section, a thorough but critical attempt is made to review existing state of the art and to identify their shortcomings. All the literature discussed below is focused on various machine learning and deep learning techniques and their advantages and drawbacks related to cassava leaf disease prediction. The work done in available literature can be divided into two categories based on the methodology used by authors.

#### **1. Traditional Machine Learning Methods**

Decision Tree, Random Forest, K-Nearest Neighbors, Logistic regression, Support Vector Machine, Ensemble etc.

#### **2. Deep Learning based Methods**

Custom Convolutional Neural Networks, Transfer learning models like MobileNet, Plain Residual Neural Networks, Inception etc.

#### **2.1. TRADITIONAL METHODS:**

Many authors [5], [6], [7] have utilized various image processing and machine learning methods for classification purpose. First the image is preprocessed by applying various techniques to remove background noise, handling low contrast and low-resolution images, removing irregular and poorly captured images etc. The main drawback of all these traditional methods is that we need to extract informative and high-level features prior to giving input to the model for classification purpose. Extraction of such features requires extra effort for choosing the best feature extraction, selection, and reduction techniques.

In [5], authors have extracted two types of features from images: 50 Chromatic features by applying HSV color transformation and 120 ORB (Oriented fast and Rotated BRIEF) features. After passing the features to three different classifiers (Linear Support Vector Classifier, k-nearest

neighbors, and Extra trees), they were able to achieve an accuracy of 80% by passing chromatic features to linear SVC and 100% for k-NN with ORB features. Image with identified ORB key points taken from original paper is given in Figure 2-1.



**Figure 2-1**-Image with ORB interest key points identified [5]

10 chromatic, color indices and Hue, Saturation and Intensity features were extracted to perform Hierarchical binary classification for detection of Brown leaf spot (BLS) in cassava plant and were fed to Artificial Neural Network in [6] that was trained with varying number of hidden neurons from 1 to 100 with an increment of 10 each time.

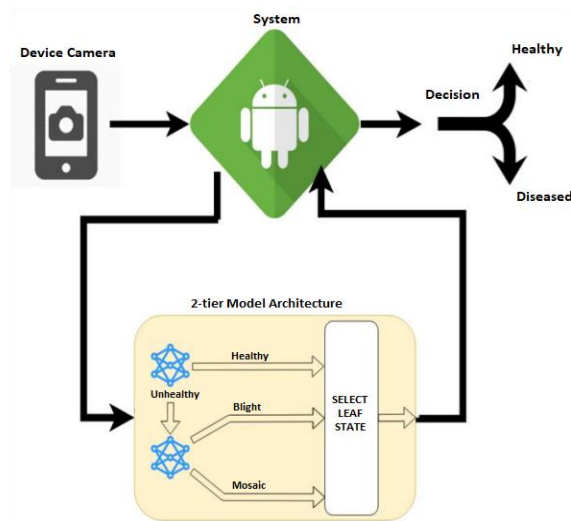
**Table 2-1**-Number of hidden neurons and classification accuracy of healthy and diseased cassava leaves [6]

Number of hidden neurons	Success classification (%)	
	Healthy plants	Diseased plants
10	87.94	77.13
20	89.92	75.73
30	86.70	79.23
40	88.68	77.48
50	87.49	77.36
60	88.04	78.06
70	89.77	74.21
80	87.35	78.18
90	87.59	77.25
100	87.05	77.36

Highest accuracy of 89.92% for health diagnosis was achieved with 20 neurons in hidden layer and 79.23% for BLS was achieved with 30 neurons in hidden layer. Table 2-1 shows the

classification accuracy for both healthy and diseased leaf detection for various number of hidden neurons.

Bag of features technique has been utilized in [7] to extract 1 million features from cassava leaf images including images of healthy leaves as well as leaves with CMD and CBBD. The dimensionality of extracted features was then reduced to 500 predictors that were then passed to 6 classifiers (Discriminant analysis, Logistic Regression, Support Vector machines, Ensemble classifier, Decision Tree and k-NN) to train 18,000 images. Hierarchical classification was performed by first predicting the leaf as healthy or diseased and then the type of disease as either Mosaic or blight. They achieved an accuracy of 83.9% for health diagnosis and 61.6% for disease type detection with cubic SVM and Coarse Gaussian SVM respectively whereas ROC value was 0.85 for healthy leaf and 0.66 for disease detection. System architecture is given in Figure 2-2.



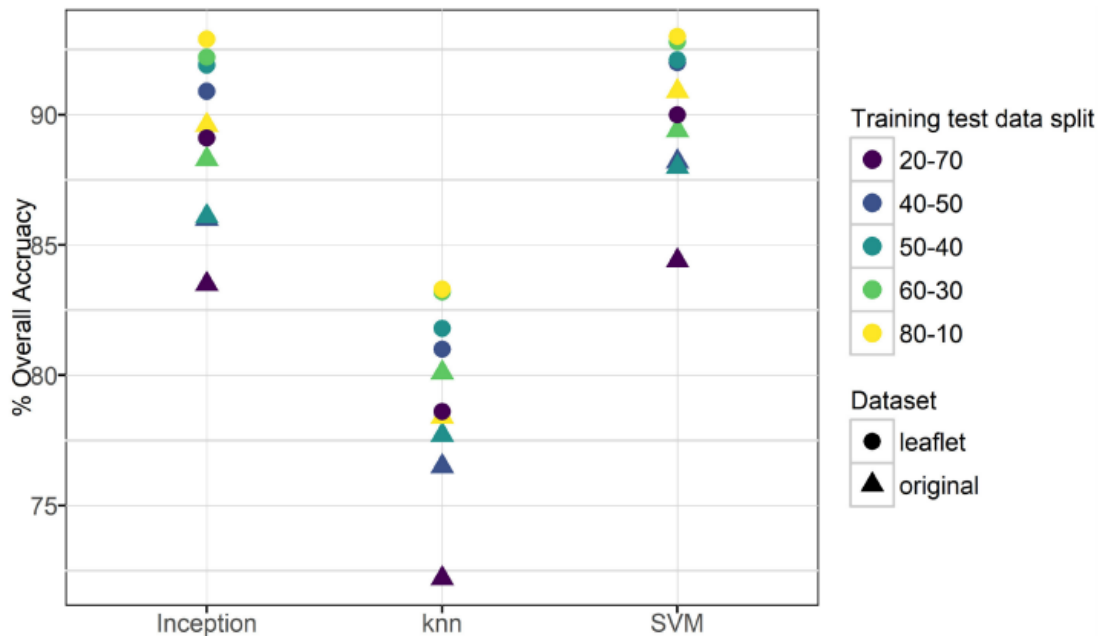
**Figure 2-2-**System Architecture [7]

## 2.2. CONVOLUTIONAL NEURAL NETWORKS:

Deep learning is a rising trend in data science community for image classification and is a major step ahead of traditional machine learning methods in that it tries to learn high level features from the image data on its own. It involves the use of Convolutional neural networks (CNN) in which a kernel is convolved (mapped) over the whole image pixels to extract highly important features. But it has an overhead that it is computationally expensive than traditional methods as it takes a lot of time to train and for efficient training, the size of training dataset much be large to capture

all the varieties of data. To overcome this issue, transfer learning was utilized that applies pre-trained models on a new classification problem. Studies have shown that the application of transfer learning not only saves a lot of time, but it also improves model performance.

In [8], authors have applied 3 architectures of inception-v3: 1) with original softmax layer 2) k-NN and 3) SVM on cassava leaf images belonging to three diseases (Cassava Bacterial Blight, Cassava Brown Streak Disease and Cassava Mosaic Disease), 2 pest damages (Red Mite Damage and Green Mite Damage) and healthy leaves. They have used a range of splitting data to be used for training and testing from 80:10 to 20:70 and found that traditional methods can be applied to deep learning methods to improve performance. They were able to achieve a very good accuracy results of 91% on original dataset and 93 % on leaflet dataset that was obtained by cropping the images in original dataset thereby increasing the sample size. However, this method of cropping has a drawback that not all the leaflets in an image will have the disease. So, it will result in inaccurate labels if not properly checked. Figure 2-3 shows the classification accuracy results for different training testing splits of all the 3 architectures used.



**Figure 2-3**-Overall accuracy for transfer learning using three machine learning methods [8]

To aid with low contrast and poor resolution, CLAHE (Contrast Limited Adaptive Histogram Equalization) was used in [9] along with SMOTE (Synthetic Minority Oversampling Technique),



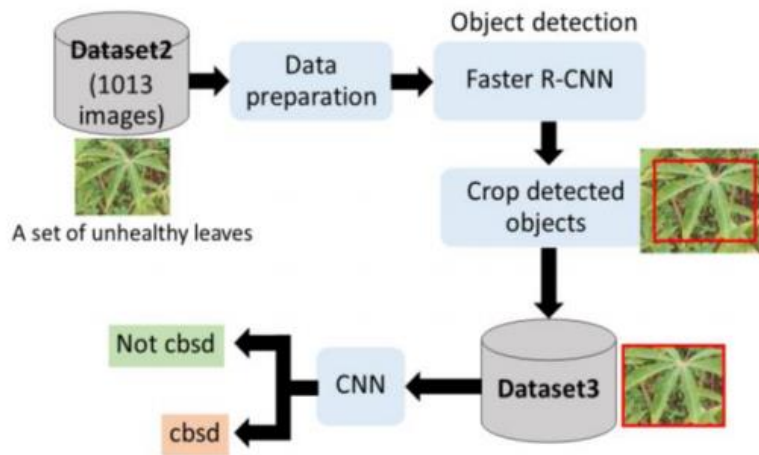
class-weights and augmentation techniques to increase the size of training data with the assumption that more training data results in more accurate results. They have trained a Custom Convolutional Neural Network with 3 convolutional layers and 4 fully connected for 124 Epochs with different input resolutions and achieved the highest accuracy of 88% on imbalanced and 93% on balanced dataset against resolution 448x448x3 concluding that higher resolution improves performance and this is intuitive as high-resolution image will have more information and more high-level features can be extracted from that. But this method is very time consuming as they have not applied transfer learning and building from scratch is computationally very expensive. Also, the model is trained for many epochs where each epoch took 3600s. The results are shown in Table 2-2.

**Table 2-2**-Effect of different input image dimensions on CNN model accuracy [9]

Input Shape	Accuracy (%)	Loss (%)	Epochs	Time per Epoch
(128,128,3)	76.9	26	124	699 s
(224, 224,3)	80	18	124	1513 s
(256, 256, 3)	89.0	10	124	2065 s
(448, 448, 3)	93.0	0.06	124	3600 s

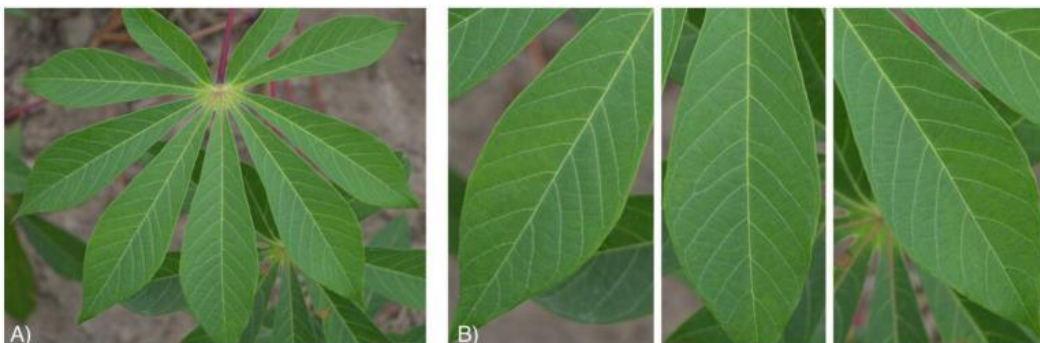
Similarly, in [10], another Convolutional Neural Network was built from scratch with 9 convolutional layers and 2 max pooling layers to classify cassava leaves as either healthy or diseased (leaves with Cassava Brown Streak). Authors have used Faster R-CNN that is a pre trained algorithm for object detection, that detects object of interest from image by removing background details and achieved an accuracy as well as f-score of 0.96. However, there input image resolution was low and sample size was also very small. Faster R-CNN for object detection used in on the original paper is shown in Figure 2-4.

MobileNet-V2 is applied on cassava dataset containing 2317 images and was trained for 40 epochs to classify 4 disease and one healthy leaf in [11]. By using a 70:20:10 data splitting for training, validation and testing respectively, authors have achieved an accuracy of 74.5% for training data and 67.3% for validation data.



**Figure 2-4-**Process Illustration of Faster R-CNN for Object Detection [10]

Further, in [12], Distinct Block Processing technique, shown in Figure 2-5 has been employed by authors to increased size of each class (4 diseases and 1 healthy leaf) as their dataset was imbalanced with 5,656 images. The resulting dataset contained 13,500 images but with low resolution of 30x30x3. Plain Convolutional Neural Network (PCNN) and Deep Residual Neural Network (DRNN) were trained for 80 epochs giving 52.87% and 46.24% on imbalanced dataset and 87.5% and 96.75% on balanced dataset, respectively. However, this block processing technique has the same shortcoming as that was in [8]. Different blocks obtained from the same image might not have the disease.



**Figure 2-5-**Application of distinct block Processing to increase sample size of minority classes [12]

Comparison of all the above-mentioned techniques used in literature is given in tabular form in Table 2-3.

**Table 2-3-**Comparison of different Machine Learning and Deep Learning techniques for Cassava Leaf disease Classification

Paper	Year	Method(s)	Features	Database	Accuracy	Classification
[5]	2016	Linear SVC, k-NN, Extra Trees	Color and ORB features	Own Database	80% with linear SVC	Multi class with 5 classes
[6]	2011	ANN	Chromatic coordinates and HSI Features	Own database	Healthy: 89% Disease: 79 %	Hierarchical Binary (Healthy & BLS)
[7]	2019	SVM	Bag of Features + PCA	Own database	Healthy: 83% Disease: 61.6%	Hierarchical (Healthy, Mosaic, BLS)
[8]	2017	Inception-v3 with softmax, SVM and k-NN	Cropping to obtain leaflet dataset	Own database	Original: 91% Leaflet: 93%	Multi class with 6 classes
[9]	2021	Custom CNN with 3 conv layers and 4 fully connected layers	CLAHE, SMOTE and Augmentation	Kaggle database	Imbalanced: 88% Balanced: 93% with 448x448 resolution	Multi class with 5 classes
[10]	2020	Custom CNN with 9 conv layers 2 max-pooling layers	Faster-RCNN for Object Detection and cropping	Own database	96 %	Hierarchical Binary (Healthy and CBSD)

---

		and 1 dense and flatten later				
[11]	2021	MobileNet-V2	-	Kaggle database	67.3%	Multi class with 5 classes
[12]	2021	P-CNN and DRNN	Distinct Bock Processing	Kaggle database	Imbalanced: 46.24% Balanced: 96.75%	Multi class with 5 classes

---

### 2.3. RESEARCH GAPS:

In most of the work, binary classification is performed [6], [7], [10] and the size of dataset is small [6], [10], [11], [12] but to yield solid and promising results with deep learning, the dataset size must be large enough to capture all the possible conditions found in practice irrespective of transfer learning and augmentation techniques. Studies have shown that having too few samples in the input dataset results in high error rates most of the times [4].

Although in literature, deep learning techniques have been applied to classify and predict the diseases, there are some drawbacks as given below:

- Block Processing Technique [12] or cropping [8], where authors perform leaf wise analysis does not represent real time scenario as it is very time consuming and additional effort is required to do this tedious task.
- Generally, authors have performed just binary classification [6], [7], and [10] to determine whether the leaf is healthy or diseased.
- In most of the cases, dataset size was very small [6], [8], [11] and [12] that may lead to less generalized solutions.
- For natural and more cluttered leaf images, background noise should be removed as irrelevant and unnecessary information may create difficulty in improving the overall performance. So, such background noise needs to be removed during pre-processing of dataset.

#### **2.4. CONTRIBUTION:**

Based on the extensive review of literature and to overcome all the above-mentioned shortcomings and drawbacks, the aim of this research is to use a dataset containing substantial number of sample images with high resolution to capture high level features and to apply transfer learning to classify cassava leaf images into 4 diseases including Cassava Mosaic Disease, Cassava Bacterial Blight, Cassava Brown Streak Disease and Cassava green Mottle and one healthy leaf. For Transfer learning, we have used EfficientNet because it can not only increase the performance as compared to other benchmark transfer learning models e.g., VGG, ResNet, DenseNet etc. but also reduces the number of parameters required for training. Further to remove background noise, U-Net architecture was used to perform semantic segmentation as it focuses more on precisely locating the desired object and is much faster as compared to other segmentation techniques. Unlike other deep learning models, it does not need a large dataset for training purpose [19]. In addition to segmentation, augmentation was also performed on the dataset to increase the sample size of minority classes as studies have shown that augmentation helps in generalizing the model and results in improved performance [42].

## Chapter 3

### MATERIALS

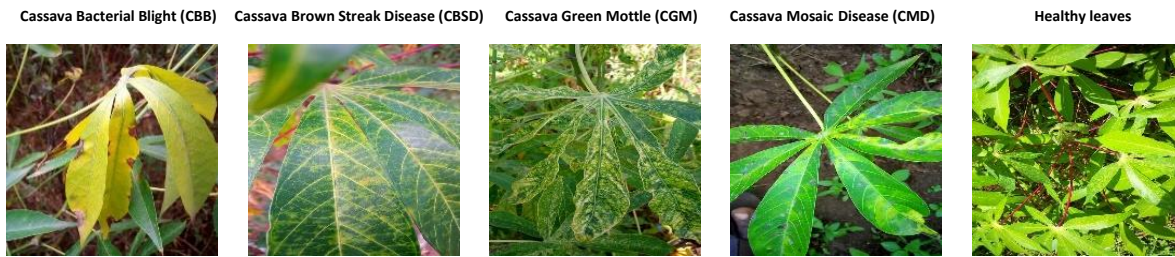
#### 3.1. CASSAVA IMAGE DATASET

The dataset for cassava leaf disease classification is taken from Kaggle competition [29] that consists of 21,367 labeled images collected during a regular survey in Uganda. These images were captured by farmers and were further processed by experts at the National Crops Resources Research Institute (NaCRRI) in collaboration with the AI lab at Makerere University, Kampala. There were five output categories in the dataset; four being the diseases and one for healthy leaf and their label to disease mapping is given below.

1. 0: "Cassava Bacterial Blight (CBB)"
2. 1: "Cassava Brown Streak Disease (CBSD)"
3. 2: "Cassava Green Mottle (CGM)"
4. 3: "Cassava Mosaic Disease (CMD)"
5. 4: "Healthy"

However, the dataset was highly imbalanced with 1,087 images for CBB, 2,189 images for CBSD, 2,386 for CGM, 13,158 for CMD and 2,577 images belonging to healthy leaf class. All the images were resized to 512x512 resolution using TensorFlow as it provides a fast and easy way to perform image classification. Sample images for each of the 4 diseases and healthy leaf is shown in Figure 3-1. Figure 3-2 shows sample distribution among 5 classes. The same has been shown in tabular form in Table 3-1.

A brief description of the above mentioned 4 diseases is given below:

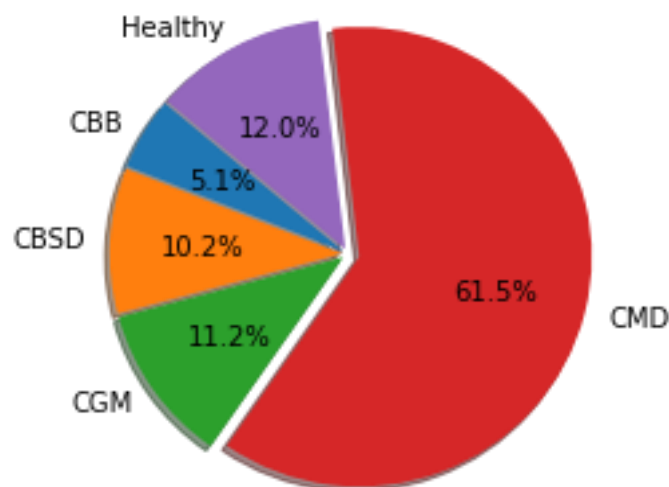


**Figure 3-1**-Sample Images from each of the 4 diseases and healthy leaf (from left to right)

**Table 3-1**-Frequency of Images in each Output Class

DISEASE	Label	Frequency
Cassava Bacterial Blight	0	1,087
Cassava Brown Streak Disease	1	2,189
Cassava Green Mottle	2	2,386
Cassava Mosaic Disease	3	13,158
Healthy Leaf	4	2,577
<b>Total Images</b>		<b>21,397</b>

Samples division among 5 classes in Cassava Leaf Disease Dataset 2020



**Figure 3-2**-Samples Distribution among 5 classes

### 3.1.1. CASSAVA BACTERIAL BLIGHT (CBB):

This disease is one of the devastating diseases that affects cassava production worldwide and is caused by bacteria. This disease is mostly found on cassava plants that are grown in moist places. Its main symptoms include black leaf spots, wilting, and blight, drying and shedding of leaves. However, these symptoms vary depending upon the location, season and severity of disease. Sample images of CBB are given in Figure 3-3.



**Figure 3-3**-Leaves with Cassava Bacterial Blight Disease

### 3.1.2. CASSAVA BROWN STREAK DISEASE (CBSD):

CBSD is caused by two virus species; Cassava Brown Streak Virus and Cassava Brown Streak Uganda Virus that are transmitted by a vector called whitefly. Its chlorotic symptoms include yellow patches in leaves that sometimes enlarges based on the severity of disease resulting in chlorosis. Its symptoms are also shown on roots that consist of dark brown necrotic areas resulting in root size reduction. Figure 3-4 shows images of CBSD leaves.



**Figure 3-4**-Leaves with Cassava Brown Streak Disease



### 3.1.3. CASSAVA GREEN MOTTLE (CGM):

This disease is a viral infection caused by cassava green mottle virus. Its main symptoms include yellow spots on leaves from small dots to irregular patches depending on the severity of disease, twisted and distorted leaf margins and mosaics. Most of the time, cassava plants affected by green mottle become severely stunted. CGM diseased leaves images are given in Figure 3-5.



**Figure 3-5**-Leaves with Cassava Green Mottle Disease

### 3.1.4. CASSAVA MOSAIC DISEASE (CMD):

This is the most severe and widespread disease that limits cassava production greatly. Its foliar symptoms include mottling, mosaic, and twisted leaves. Leaf size is also affected that can be easily seen on leaves. The disease disturbs the green color of leaves that changes to small yellow and white patches affecting the photosynthesis process and thereby reducing plant growth and hence the yield. Sample images of CMD leaves are given in Figure 3-6.



**Figure 3-6**-Leaves with Cassava Mosaic Disease

### 3.2. TOOLS AND LANGUAGES:

The tool used in this research for implementing proposed algorithm is Google Collaboratory with the following specifications:

- GPU: Google Compute Engine Backend (NVIDIA Tesla k80 accelerator)
- RAM: 25 GB Available
- DISK SPACE: 147.72 GB Available

Because GPU is much faster in processing image data, it is preferred over CPU that takes almost 10x more time to perform the same task. Google co-lab is used because it provides free GPU Support and RAM up to 12 GB. Also, the same virtual machine can be used continuously for at most 12 hours. Since the size of dataset that is used in this research was very large i.e., there were about 21K leaf images that requires more RAM than 12 GM, so instead of co-lab, co-lab pro was used as it provides faster access to GPU, longer runtime, and more Storage space (almost double i.e., 25GB RAM). Also, the virtual machine can be used for training up to max 24 hours.

Further, co-lab also supports Python that is used as a scripting language in this research. Python machine learning and deep learning packages can be imported into the co-lab notebook by using just a single line of code instead of downloading and installing them separately thereby saving a lot of time. The packages used include the following:

- Keras with TensorFlow backend
- Numpy and pandas
- Matplotlib for plotting
- csv for reading csv file containing images names and their corresponding labels.
- cv2 for reading, writing, and displaying images.
- glob etc.

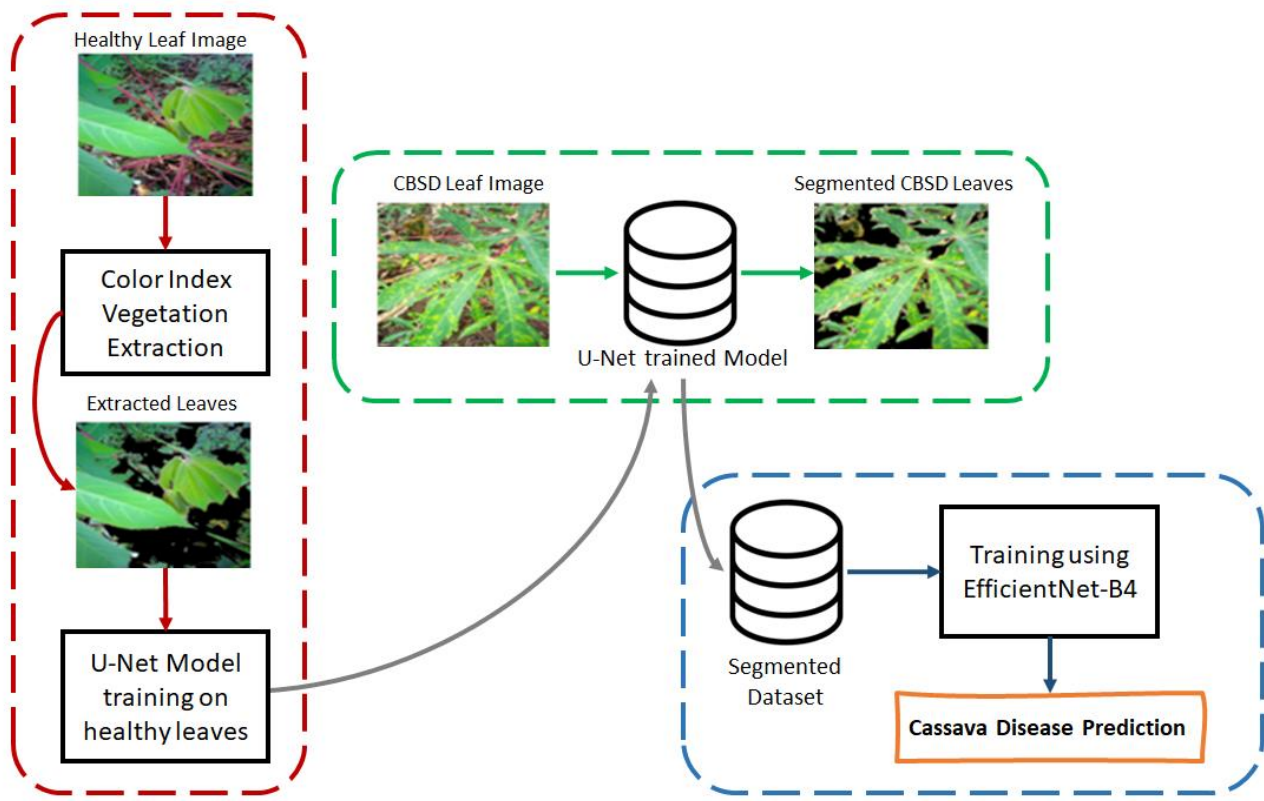
## **Chapter 4**

### **METHODOLOGY**

This research focuses on the use of U-Net, a deep learning architecture to perform precise leaves segmentation. A novel and efficient transfer learning model, EfficientNet was trained on the original as well as segmented dataset to classify leaf diseases quickly and accurately. Since the resolution of images was 512x512x3, so EfficientNet models were used to extract the most important features. First the input images from under sampled classes were increased by performing detailed image augmentation. It increased the size of dataset from 21,397 to 54,902 images. For evaluation purpose, first the dataset was divided into two sets for training and validation in the ratio 85:15 and then 7-fold cross validation was used to be assured about model's performance.

#### **4.1. PROPOSED FRAMEWORK:**

Figure 4-1 shows the flow of proposed framework. First, all the healthy leaves were separated from cassava leaves dataset. Then using Color Index Vegetation Extraction, vegetation parts from healthy leaves were segmented that gives the ground truth for corresponding healthy leaf image. Once all the healthy leaves and their corresponding ground truths were obtained, augmentation was performed on original images with the assumption that the trained model would correctly segment diseased leaves as well. The data was given to UNet model for training to perform semantic segmentation and this process is illustrated by red dotted rectangle in figure 4-1. In the second step, the trained model was used to segment other diseased leaves from cassava leaves dataset (used in this research) shown by green dotted rectangle in figure 4-1. And in the last step, the whole segmented dataset (original as well as balanced) was given input to EfficientNet model to perform multi class classification for cassava leaf disease prediction, given by blue dotted rectangle in figure 4-1. Each of these activities are further explained in detail in below sections.



**Figure 4-1**-Flowchart of Proposed Framework

#### 4.1.1. IMAGE SEGMENTATION:

Since the images in the dataset contained background noise like sky, soil, hands etc., so there was a need to segment only the leaves from images. Segmentation not only reduces the complexity of an image but also helps to analyze the image in a simpler way.

In this research work, we have used U-Net architecture to train the images for segmentation purpose. The model consists of two paths: contracting (down-sampling) path that is like convolutional network and an expanding (up-sampling) path to construct a mask of the same size as that of input image. U-Net model is much better than other segmentation techniques such as Mask R-CNN, SegNet etc. in that it is much faster and works well with small datasets as well. To segment images using U-Net, we need to train images along with their corresponding ground truths. For that, Color Index for Vegetation Extraction (CIVE) was used.

#### 4.1.1.1. COLOR INDEX VEGETATION EXTRACTION:

Color Index for Vegetation Extraction (CIVE) [18] extracts vegetation part from image and is given by the following equation:

$$Z = -0.441 \times R + 0.811 \times G - 0.385 \times B - 18.787 \quad (4.1)$$

Where  $Z$  in the CIVE band and  $R$ ,  $G$ , and  $B$  are the intensity values of each pixel. Since this formula only extracts vegetation part from background, augmentation was applied on images before training with the assumption that it will correctly segment diseased leaves as well. Based on the availability of segmented images, Plant pathology 2021 dataset from Kaggle was also combined with the healthy images to increase the diversity of training data. When trained for 20 Epochs and validated on diseased cassava leaves, the model was able to correctly segment most of the diseased leaf images except for Cassava Bacterial Blight disease, since this disease has brown patches on leaves that were not learned by the model. So, to segment such leaves, a slight change was made to eq. (4.1), giving more weights to red and green colors to extract the leaves along with brown patches of disease on them, given by eq. (4.2).

$$Z' = 0.441 \times R + 0.611 \times G - 1.885 \times B - 48.787 \quad (4.2)$$

Where  $Z'$  is the changed CIVE band for extraction of leave with brown patches.

As a result, two segmented datasets were obtained, one from U-Net trained model containing images of CBSD, CGM, CMD and healthy leaves and second from changed CIVE formula containing images of CBB leaves. Both the datasets were then combined to obtain a single segmented dataset containing the segmented images belonging to 5 classes.

#### 4.1.1.2. IMAGE AUGMENTATION (FOR SEGMENTATION):

To accurately segment diseased leaves from background, the healthy leaf dataset was subjected to augmentation with the assumption that the trained model will be able to detect diseased leaves as well. The augmentation transforms were applied sometimes randomly, and they include the following:

- Course Dropout – remove some areas that can be larger in size than simple dropout.
- Gaussian Blur – blur the input image using a random-sized kernel
- Sigmoid Contrast – adjust the image contrast level

These three transforms were selected because they represent real time setting. Dropout is added to introduce some noise in the image. Blurring is added to remove noise and lower high-level details as not all features are sharp, and contrast is used to enhance the features present in image. Augmentation results of the above-mentioned transforms along with the original image is shown in Figure 4-2.



**Figure 4-2**-Original Image (Top) and Augmentation results on Healthy Cassava Leaf (bottom)

#### 4.1.1.3. UNET ARCHITECTURE:

UNet architecture [19] was proposed by Olaf Ronneberger, in 2015. It is a symmetric model, and its architectural shape resembles the alphabet U, hence the name UNet was given. This architecture was first used as a semantic segmentation model for bio medical images, however, with the advent of time, it has been used in many other fields as well including agriculture. The model consists of a total of 23 convolutional layers and is divided into 2 paths:

1. Contracting Path also known as Encoder

## 2. Symmetric Expanding Path also known as Decoder

The first path that is contracting path is the same as normal convolutional network where convolution is performed using kernel followed by max pooling layers. This path down-samples the image by doubling the number of channels whereas height and width of image gets halved after each stage. However, the second path is just the opposite of normal convolution as there the height and width of image are doubled, and channels are halved to get the output image with the same high resolution as that of the input image because normal contracting path lowers the resolution of image. In the Expanding path, up-sampling is performed using transposed convolution, also known as deconvolution or fractionally strided convolution. Transposed convolution is mostly used instead of other ways of up-sampling like interpolation because of its ability to learn parameters on its own.

For transposed convolution, we need to define convolution matrix of size  $m^2 \times n^2$  where  $m$  is the input image size and  $n$  output image size. The square is taken since the matrix when multiplied with the flattened input vector will give a high-resolution output vector that will be rearranged to get desired output. Convolution matrix is obtained by rearranging weights from kernels. The low-resolution input image is flattened to get a vector of size  $m^2 \times 1$ . The transposed convolution matrix  $n^2 \times m^2$  when multiplied with the vector  $m^2 \times 1$  will give a vector of size  $n^2 \times 1$  that when rearranged will give desired up-sampled output image of size  $n \times n$ .

Architectural diagram of UNet along with transposed convolution is shown in Figure 4-3. In the original paper, a convolution layer of 3x3 was used with valid padding. However, this doesn't produce the same resolution image as that of input. Hence in the implementation same padding is used and the same has been shown in architectural diagram in Figure 4-3. Since, the image size of the dataset is 512 x 512, so the same size is used in describing UNet architecture therefore sizes may vary from original paper, however basic concept is same.

The network architecture can be divided into 5 stages based on the number of convolutional layers. In the first stage, a normal convolution is applied 2 times with 3x3 filter and same padding to extract high level information from the image, followed by a 2x2 max pooling layer that reduces the parameters in the network. The same process is repeated for all 5 stages, where each stage has

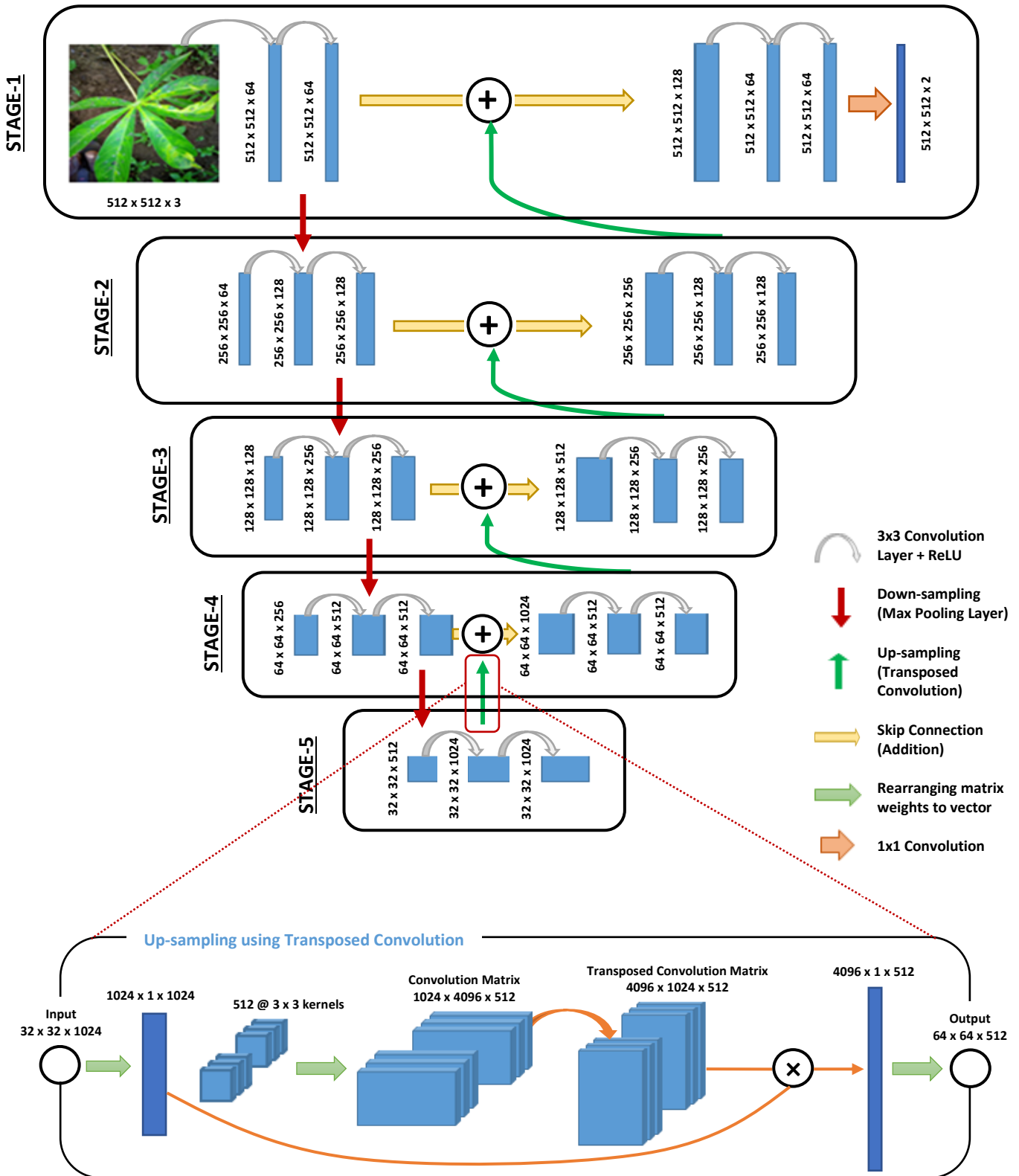


Figure 4-3-UNet Architectural Diagram



double number channels whereas resolution gets halved. After that, expanding path begins, where image is up-sampled back to its original resolution that has only the extracted mask from original image. During each up-sampled stage, a skip connection is also added by concatenating the transposed convolution with the corresponding feature maps from the same stage in contracting path. Concatenation is followed by the same 2 normal 3x3 convolutional layers with same padding as were added in contracting path. This is done to recover the fine-grained features captured during contracting path and precisely locate the segmented image. A ReLU activation function is applied after each convolution operation.

#### **4.1.2. IMAGE AUGMENTATION:**

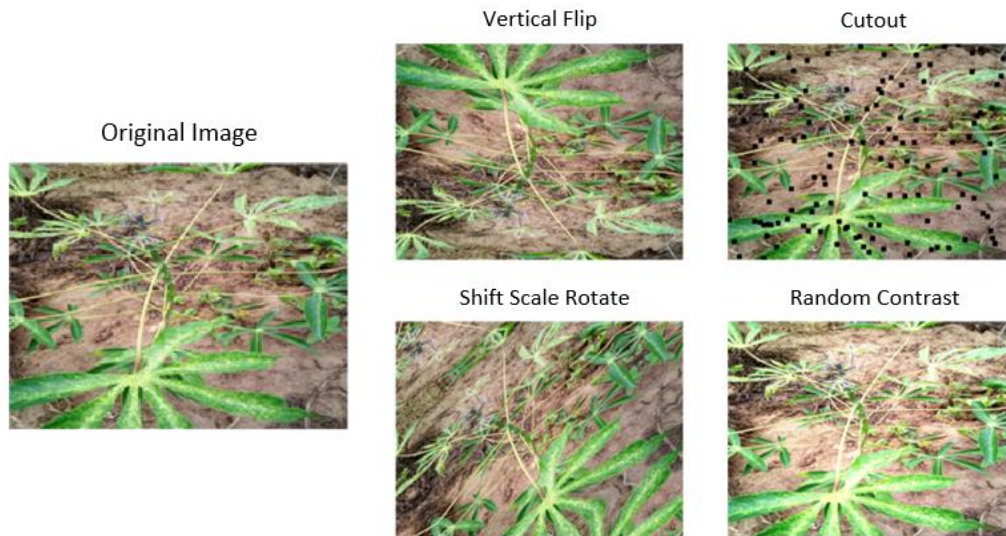
To address the class imbalance issue, the cassava leaf dataset obtained from Kaggle was increased by applying different transforms on the images. For images in the minority classes, the following transforms were applied on images:

- Cutout – Removes some portions from the image to introduce noise
- Random Contrast - Randomly changes the contrast of input image
- Random Brightness - Randomly changes the brightness level of input image
- Shift Scale Rotate – Collectively performs all the three operations on the input image
- Gauss Noise - Apply Gaussian Noise to the input image
- Vertical Flip – Flips the input image vertically around the y-axis
- Random Rotate – randomly rotates the input image
- Horizontal Flip – Flips the input image horizontally around the x-axis

The main purpose of performing augmentation was to avoid overfitting the model by making it generalized and robust to improve the performance of deep neural networks and to handle class imbalance issue as well. All these transforms were only applied to training dataset. Validation dataset had only the original images. Augmentation results of some of the above-mentioned transforms along with the original image is shown in Figure 4-4.

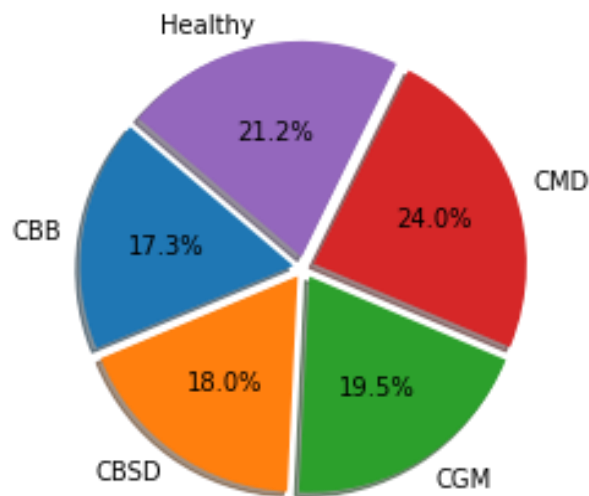
After detailed augmentation on all labels except CMD, the size of dataset increased from 21,397 images to 54,902 images with almost equal distribution of all 5 output labels as shown in Figure 4-5 The distribution of labels in the balanced dataset was as follows: 9,520 images for CBB, 9,881

images for CBSD, 10,682 images for CGM and 11,661 images for Healthy leaves. The difference in distribution of labels is because augmentation was performed on only training data and not on validation data.



**Figure 4-4-**Augmentation Results with Original image (left) and transformed images (right)

**Samples division among 5 classes in Balanced Cassava Leaf Disease Dataset 2020**



**Figure 4-5-** Samples Distribution among 5 classes in Balanced Cassava 2020 Dataset

### 4.1.3. EFFICIENTNET ARCHITECTURE:

Efficient-Net was developed by Google in 2019 and as the name indicates, it gives promising performance not only in terms of accuracy but in terms of efficiency as well. When trained on the ImageNet classification problem, it gave 84.3% top-1 accuracy with 66M parameters leading the state-of-the-art transfer learning models including ResNet, GoogleNet, MobileNet etc. It is a family of 8 models from B0-B7 covering different resolutions of input images as well as varying number of depth and width parameters [15].

Studies have shown that scaling ConvNets gives better accuracies with minimum parameters. The most common way of scaling is by image depth i.e., by increasing the number of layers followed by image resolution and this is intuitive, as we increase the image resolution, more layers and more channels will be required to capture high level and fine-grained patterns from the input image. But scaling can only be done up to a certain limit after that model performance tends to decrease. For example, Resnet-101 and Resnet-1000 have almost similar accuracy irrespective of the fact that Resnet-1000 has much more layers than Resnet-101. Similarly, an extremely wide but shallow network can have difficulty in capturing the required patterns from input data. So, to cater for such issues, EfficientNet was proposed that is a compound scaling method and uses a compound coefficient  $\phi$  that uniformly scale all the three dimensions – network width (No. of channels), depth (No. of layers), and resolution (height x width) in a constant ratio given by eq. (4.3).

$$\text{depth} : d = \alpha^\phi$$

$$\text{width} : w = \beta^\phi$$

$$\text{resolution} : r = \gamma^\phi$$

$$\text{subject to } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \text{ and } \alpha, \beta, \gamma \geq 1 \quad (4.3)$$

Where  $\alpha$ ,  $\beta$  and  $\gamma$  are the depth, width, and resolution scaling coefficients and  $\phi$  is a compound coefficient that controls how many more resources are available for model scaling. Based on the resource constraint  $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ , if we want to increase depth by  $\alpha^\phi$ , width by  $\beta^\phi$  and image resolution by  $\gamma^\phi$ , we will require  $2^\phi$  time more computational resources.

The optimal values of  $\phi$ ,  $\alpha$ ,  $\beta$  and  $\gamma$  were found by performing a small grid search and it was simple and easy as compared to arbitrarily selecting the values for three dimensions that requires manual

tuning. First, the values of  $\alpha$ ,  $\beta$  and  $\gamma$  were found by keeping the resource constraint same ( $\varphi = 1$ ) and then, baseline scaling coefficient,  $\varphi$  was scaled with different values to get the family of models from B0-B7. Table 4-1 shown the baseline Network (EfficientNet B0).

**Table 4-1** - EfficientNet-B0 Baseline Network [15]

Stage	Operator	Resolution	# Of Channels	# Of Layers
1	Conv 3x3	224x224	32	1
2	MBConv1, k3x3	112x112	16	1
3	MBConv6, k3x3	112x112	24	2
4	MBConv6, k5x5	56x56	40	2
5	MBConv6, k3x3	28x28	80	3
6	MBConv6, k5x5	14x14	112	3
7	MBConv6, k5x5	14x14	192	4
8	MBConv6, k3x3	7x7	320	1
9	Conv 1x1 & Pooling & FC	7x7	1280	1

The main building block in EfficientNet model is mobile inverted bottleneck MBConv that is similar to MobileNetV2 [16] but with added Squeeze-and-Excitation blocks. As MBConv is an inverted bottleneck block, it means the network is first expanded and then narrowed to get the same features as from normal convolution but with significant reduction in parameters to achieve highly efficient performance. In general, there are three layers.

1. **Expansion Layer** is a 1x1 convolution that expands the number of channels by an expansion factor  $t$  such that the output will have  $t \times k$  channels where  $k$  is the input channel size. For example, if we have 16 channels in input to expansion layer and  $t = 6$ , there will be  $16 \times 6 = 96$  output channels that will become input to the next layer.
2. **Depth-wise Convolutional Layer** performs filtering by applying a single convolutional filter with same number of channels as there are in input image producing  $t \times k$  output channels.
3. **Projection Layer** is also a 1x1 convolution but with a linear activation function to avoid any further information loss. This layer performs point to point mapping and decreases the number of channels to the desired ones.

In EfficientNet Architecture, Squeeze-and-Excitation block [17] is added before pointwise convolution to give more attention/weight to important channels over others that are basically our feature maps. A dropout layer and shortcut connection are also added between layers in one stage where stride=1 and input and output have same number of channels.

**Table 4-2** - EfficientNet-B4 Network Scaled from Baseline Network

Stage	Operator	Resolution	# Of Channels	# Of Layers
1	Conv 3x3	380x380	48	1
2	MBCConv1, k3x3	190x190	24	2
3	MBCConv6, k3x3	95x95	32	4
4	MBCConv6, k5x5	48x48	56	4
5	MBCConv6, k3x3	24x24	112	6
6	MBCConv6, k5x5	24x24	160	6
7	MBCConv6, k5x5	12x12	272	8
8	MBCConv6, k3x3	12x12	448	2
9	Conv 1x1 & Pooling & FC	12x12	1792	1

**Table 4-3** - EfficientNet-B6 Network Scaled from Baseline Network

Stage	Operator	Resolution	# Of Channels	# Of Layers
1	Conv 3x3	528x528	56	1
2	MBCConv1, k3x3	264x264	32	3
3	MBCConv6, k3x3	132x132	40	6
4	MBCConv6, k5x5	66x66	72	6
5	MBCConv6, k3x3	33x33	144	8
6	MBCConv6, k5x5	33x33	200	8
7	MBCConv6, k5x5	17x17	344	11
8	MBCConv6, k3x3	17x17	576	3
9	Conv 1x1 & Pooling & FC	17x17	2304	1

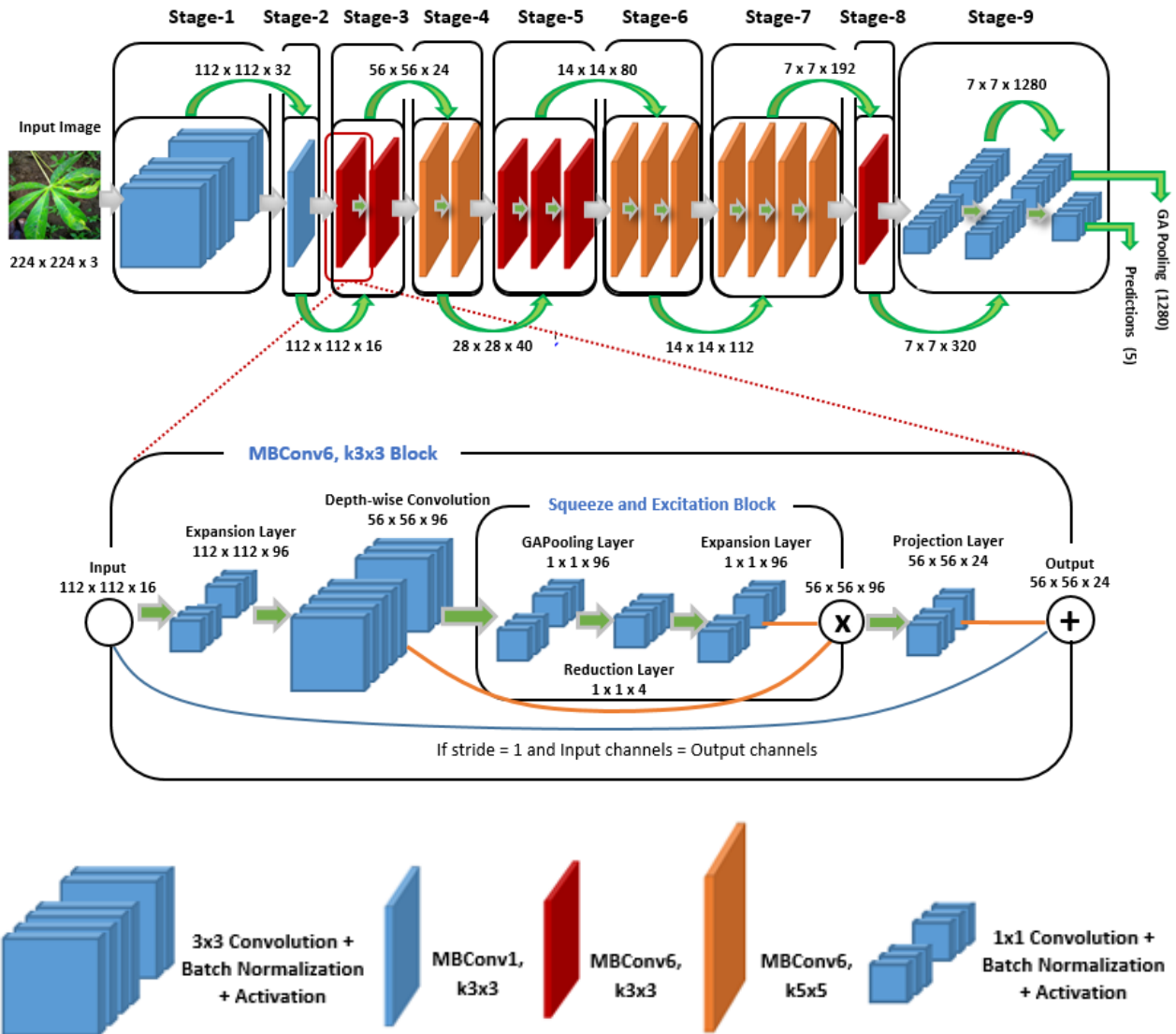


Figure 4-6-EfficientNet-B0 Architectural Diagram

Furthermore, instead of using Rectified Linear Unit (ReLU) as the activation function, a novel *Swish* activation function is used in EfficientNet that is given by:

$$f(x) = x \cdot \sigma(x) \quad (4.4)$$

Where  $x$  is the input value and  $\sigma$  is the sigmoid function that is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4.5)$$

There are a total of 18 layers in EfficientNet-B0 with input image size 224x224. The optimal values of scaling coefficients found for B0 were 1.0, 1.0, 224 and 0.2 for width, depth resolution and dropout respectively, and is given in Table 4-1. Since, our dataset has input size 512x512, we have first used B4 model of EfficientNet having scaling coefficients 1.4, 1.8, 380 and 0.4 for width, depth, resolution, and dropout, respectively. Depth coefficient value suggests that there should be a total of 32 layers in the model. However, in the model there are total 34 layers. This is because some of these values were handpicked based on highest accuracy. The scaling coefficients of EfficientNet-B6 are 1.8, 2.6, 528 and 0.5 for width, depth resolution and dropout respectively. Depth coefficient value suggests that there are 47 layers in the model and the maximum allowed input image size is 528x528. Table 4-2 and table 4-3 shows the network of EfficientNet model B4 and B6 respectively derived from baseline Network using the coefficient values mentioned before. The architectural diagram of EfficientNet-B0 is shown in Figure 4-6.

# Chapter 5

## EXPERIMENTATION & RESULTS

### 5.1. EXPERIMENTATION:

The model was implemented in Google Colaboratory utilizing Keras with TensorFlow on Google Compute Engine (GPU) Backend having 25 GB RAM and 147.4 GB Disk Space. The code was written in Python 3 with many packages including numpy, pandas, csv, scikit learn etc.

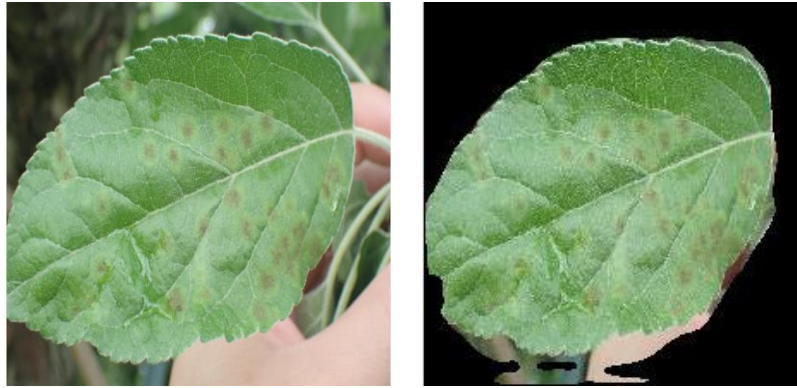
#### 5.1.1. PLANT PATHOLOGY 2021 [38]:

Plant pathology 2021 challenge is available on Kaggle and has about 23k foliar images of apple diseases. The challenge was to detect the right category of foliar leaf images. 18K images were from training dataset, that were used as an input to UNet model in this research. Segmented images of those 18K training images are also available on Kaggle [39]. Sample image along with segmented one is shown in Figure 5-1.

#### 5.1.2. TRAINING:

In addition to the train/test image files, the dataset used in this research has 16 training TensorFlow Record files having 21,397 images for training and 1 Testing TFRecord containing 1 image for testing. TFRecord is a component of TensorFlow that stores the data in binary format either as list of bytes or list of float or list of integers. Keeping the resource constraints in mind, we have utilized TFRecords for disease classification purpose because storing data in binary file format takes up less time to copy, less disk space and can be read efficiently from disk. Furthermore, when there is a large dataset, it becomes an overhead to read all the data at one time. TFRecords solves this issue by only reading a batch of data from memory that is required thereby efficiently and effectively utilizing the resources.





**Figure 5-1**-Sample Image (left) and segmented image (right) from Plant Pathology 2021 dataset

First, healthy leaf images were separated from rest of the dataset and their corresponding mask were generated using eq. (4.1) to train UNet model. In addition, Plant pathology 2021 dataset as described above was also added to increase the diversity of dataset. The model was trained for 20 Epochs with default parameters. The rest of the cassava segmented images were predicted based on the trained model. All the segmented images were saved in TFRecords format in the same order as that of original images.

Once all the segmented images were saved, 85% of them were used for training and 15% were separated for validation to measure model performance then was then followed by 7-fold cross validation.

### **5.1.3. HYPER-PARAMETERS SETTING:**

We first used EfficientnetB4 model having 17.5M parameters with 9K trainable parameters to train the model for segmentation and classification and once final segmented dataset was obtained, it was given input to all EfficientNet models to reach at final conclusion. Categorical cross-entropy was used as loss function along with Adaptive moment (Adam) optimizer with default learning rate (0.001). Using a batch size of 128, the model was trained for 50 Epochs.

### **5.1.4. PERFORMANCE METRICS:**

Two performance metrics were used to evaluate the trained model: accuracy and f-score. When the output labels are skewed, accuracy becomes an unreliable performance measure. To resolve this issue, precision and recall are used [9] where precision means exactness i.e. the percentage of positive class labels that the classifier predicted as positive were actually positive whereas recall

means completeness i.e. number of positive class labels that are predicted positive. So, f1 score was used that is the harmonic mean between precision and recall. These metrics are given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

Where:

- TP (True Positive) is the number of truly predicted samples of positive class
- TN (True Negative) is the number of truly predicted samples of negative class
- FP (False Positive) means the number of negative class samples that are predicted positive
- FN (False Negative) means the number of positive class samples that are predicted negative

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5.4)$$

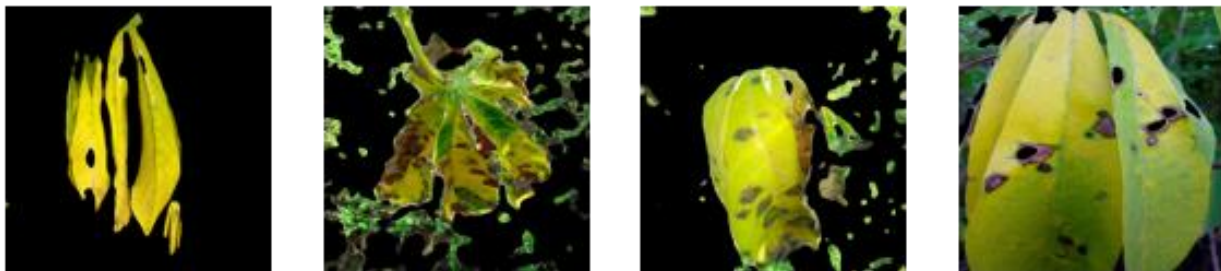
## 5.2. SEGMENTATION RESULTS:

Leaves were segmented from background using color index for vegetation extraction given by eq. (4.1). UNet model was trained on healthy images and their extracted masks for 20 Epochs accompanied with augmentation with the assumption that it will be able to detect diseases leaves from background as well. When tested on Cassava Bacterial Blight diseased leaves, the model could not accurately segment the diseased leaves. In most of the cases, the diseased part was also masked and eliminated with the background as shown in Figure 5-2 part (b) and part (d) whereas as in other cases, the model could not do proper segmentation and is clear from part (c) where the trained model has not done any kind of segmentation on the input image. There were very few images as part (a) that were accurately segmented.



**Figure 5-2-**Cassava bacterial Blight diseased leaves original images (top) and segmented images (bottom)

To cater for this issue, a slight change was made to CIVE formula and is given by eq. (4.2). The masked images were directly extracted using formula (4.2). And the resultant images are shown in Figure 5-3. It is clear from the figure that by using the changed formula, CBB suffering leaves were more accurately segmented along with the diseased part.



**Figure 5-3-**Segmented Cassava Bacterial Blight Diseased Leaves extracted Using changed CIVE formula

For segmenting rest of the diseases, including brown streak, mosaic and green mottle, Plant pathology dataset was also combined with healthy leaves before training. Since it gave much better results than just predicting based on healthy leaves with some augmentation as described in the previous chapter. Sample images of CBSD are shown in Figure 5-4 along with the segmented images.





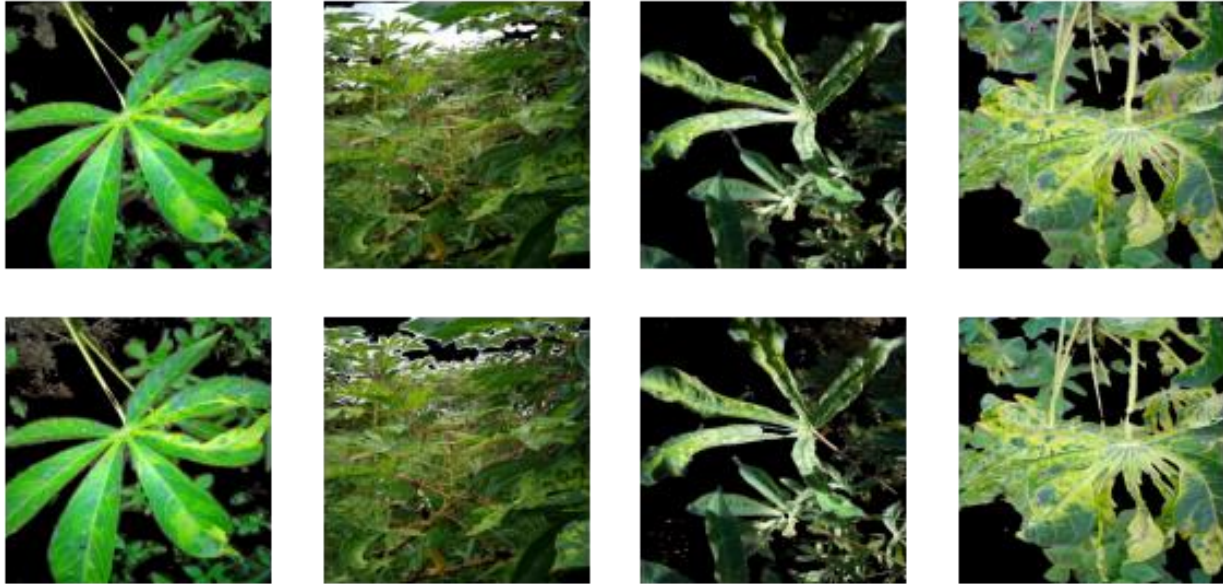
**Figure 5-4**-Cassava Brown Streak Diseased Leaves Original Images (top) and Segmented Images (bottom)

Sample images of CGM are also shown in Figure 5-5 along with the segmented images.



**Figure 5-5**-Cassava Green Mottle Diseased Leaves Original Images (top) and Segmented Images (bottom)

Original mosaic images and their corresponding segmented images using combined dataset are shown in Figure 5-6.



**Figure 5-6** -Cassava Mosaic Diseased Leaves Original Images (top) and Segmented Images (bottom)

Moving on to healthy leaves, their segmented images were extracted in the same way as CBB diseased leaves were segmented but with original CIVE formula since healthy leaves contain only the vegetation part, so they were accurately segmented using eq. (4.1). Based on these images and their corresponding masked image, UNet model was trained to predict the rest of diseased leaves. Healthy leaves images and their corresponding segmented images are shown in Figure 5-7.







**Figure 5-7**- Healthy Cassava Leaves Original Images (top) and Segmented Images (bottom)

### 5.3. CLASSIFICATION RESULTS:

Confusion Matrix is the most commonly used technique for measuring the performance of a classification model. So, it is used in this research as it presents the results in a tabular form that is very easy to understand. It is a matrix made up of rows and columns. The rows in the table give the actual labels whereas columns give the labels that are predicted by trained model.

For example, for a binary classification problem having only 2 class labels, the confusion matrix can be made as follows:

		Predicted Labels	
		POSITIVE	NEGATIVE
Actual Labels	POSITIVE	 True Positive	 False Negative
	NEGATIVE	 False Positive	 True Negative

**Figure 5-8**-Confusion Matrix for binary classification

Where True Positive mean the number of truly predicted samples of positive class, True Negative mean the number of truly predicted samples of negative class, False Positive means the number of negative class samples that are predicted positive, and False Negative means the number of positive class samples that are predicted negative. False negative is also known as type-II error and False Positive is known as Type-I error. The aim of a classification problem is to increase true positives and true negatives and reduce false positives and false negatives and most specifically false positives in the context of health-related classification problems, thereby increasing classification accuracy and hence the performance.



The same confusion matrix can be used to evaluate model performance of multi class classification problem. In the case, where there are more than 2 classes, class-wise accuracy is the best metric to evaluate performance. Accuracy, precision, recall and f-score can be calculated from confusion matrix using eq. (5.1), (5.2), (5.3) and (5.4), respectively.

Six different experiments were conducted to reach a conclusion. Starting from original dataset, followed by segmentation and then augmentation using original CIVE formula (eq. 4.1) and changed one (eq. 4.2). The following table more clearly and precisely depicts the experimentation parameters.

**Table 5-1**-Experimentation Details

Experiment	Dataset for UNet Training	Segmentation		Augmentation
		Original Equation	Changed Equation	
1	-		×	×
2	Cassava Healthy leaves	✓	-	×
3	Cassava Healthy leaves	✓	-	✓
4		-	✓	✓
5	Plant Pathology +	✓	-	✓
6	Healthy Leaves	-	✓	✓

The class-wise normalized results of confusion matrix for original and segmented datasets trained on EfficientNet-B4 on all above-mentioned 6 experiments are given in tabular form Table 5-2 – Table 5-7. For classification purpose, among the family of EfficientNet models, B4 model was first used but with high resolution of 512 x 512. Table 5-2 shows that the model has correctly classified 96% of CMD on leaves with original dataset, followed by Healthy, CBD, CGM and CBB giving 63%, 59%, 57% and 47% respectively.

**Table 5-2-** Normalized Confusion Matrix of Original Dataset

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.47	0.16	0.03	0.11	0.23
	CBSD	0.06	0.59	0.04	0.16	0.15
	CGM	0.01	0.04	0.57	<b>0.28</b>	0.1
	CMD	0.01	0.01	0.01	<b>0.96</b>	0.01
	Healthy	0.08	0.11	0.07	0.11	0.63

Table 5-3 shows the normalized confusion matrix of segmentation without augmentation using original CIVE formula (eq. 1) on cassava healthy leaves dataset. The table indicates that the model has correctly classified 95% of CMD on leaves, followed by Healthy, CBSD, CGM and CBB giving 66%, 56%, 56% and 52% respectively. The overall accuracy scores as given by table 5-9 shows that the results of segmentation are almost the same on original and segmented dataset. This can be argued as we have not done any kind of augmentation to mimic the diseases. So, most of the diseased leaves are predicted either as healthy leaves or CMD. Healthy leaves prediction is because segmentation has also masked the diseased part in leaves leaving only the vegetation part, so the model got confused in predicting such diseased leaves (CBB, CBSD and CGM) as healthy leaves.

**Table 5-3-** Normalized Confusion Matrix of Segmented dataset Using Original Formula (eq. 4.1) on Cassava Healthy Leaves Dataset without Augmentation

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.52	0.11	0.01	0.09	<b>0.27</b>
	CBSD	0.06	0.56	0.02	0.16	0.2
	CGM	0.01	0.03	0.56	<b>0.27</b>	0.13
	CMD	0.0	0.01	0.01	<b>0.95</b>	0.02
	Healthy	0.11	0.09	0.03	0.11	0.66



The overall accuracy score of segmentation with augmentation was a little bit higher (81.46 %) than segmentation without augmentation (81.39 %). So, for the next experiments augmented dataset was used. However, the model has wrongly predicted 29% of CBB as Healthy whereas only 19% of CBSD were predicted as healthy leaves. This was because CBB leaves were not properly segmented. Leaves suffering from CBB show brown patches on them while other diseases are mostly characterized by yellow patches. So changed CIVE formula (eq. 4.2) was used to segment CBB leaves and original formula (eq. 4.1) with augmentation was used to segment rest of the leaves.

**Table 5-4-** Normalized Confusion Matrix of Segmented dataset Using Original Formula (eq. 4.1) on Cassava Healthy Leaves Dataset with Augmentation

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.46	0.16	0.01	0.08	<b>0.29</b>
	CBSD	0.05	0.6	0.02	0.14	0.19
	CGM	0.01	0.05	0.55	0.26	0.13
	CMD	0.01	0.01	0.01	<b>0.96</b>	0.01
	Healthy	0.1	0.1	0.04	0.11	0.65

Table 5-5 shows the normalized confusion matrix of segmented dataset using changed CIVE formula (eq. 4.2) on cassava healthy leaves dataset with augmentation and the results were surprising. The overall accuracy score jumped from 81% to 88% with highest class-wise accuracy achieved was 90% for CMD, followed by Healthy, CBB, CBSD and CGM with accuracy scores of 90%, 88%, 74% and 61% respectively. Now, the model has correctly predicted most of the diseased leaves and wrongly predicted 31% CGM as CMD.

**Table 5-5-** Normalized Confusion Matrix of Segmented dataset Using Changed Formula (eq. 4.2) on Cassava Healthy Leaves Dataset with Augmentation

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.88	0.0	0.01	0.01	0.1
	CBSD	0.02	0.74	0.02	0.19	0.03
	CGM	0.01	0.07	0.61	<b>0.31</b>	0.01
	CMD	0.0	0.01	0.01	<b>0.96</b>	0.02
	Healthy	0.04	0.02	0.01	0.02	0.9

To make further improvement for reliable classification, plant pathology dataset 2021 was also added to healthy leaves dataset for UNet training to perform semantic segmentation of leaves using original as well as changed CIVE formula. Table 5-6 shows the normalized confusion matrix of segmented dataset using original CIVE formula (eq. 4.1) on combine dataset (cassava healthy leaves + plant pathology 2021 dataset) with augmentation. The results show that adding more images for training model for segmentation increased the accuracy a little from 81.46% to 81.65%. However, it is still not a good accuracy score because a better score was achieved from previous experiment.

**Table 5-6-** Normalized Confusion Matrix of Segmented Dataset Using Original Formula (eq. 4.1) on Combined Dataset (Cassava Healthy Leaves + Plant Pathology Dataset) with Augmentation

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.57	0.13	0.03	0.07	0.2
	CBSD	0.07	0.57	0.02	0.14	0.2
	CGM	0.01	0.04	0.56	<b>0.26</b>	0.13
	CMD	0.01	0.01	0.01	<b>0.96</b>	0.01
	Healthy	0.14	0.1	0.04	0.1	0.62

So, in the final experiment, combined dataset was used but with changed CIVE formula (eq. 4.2) for CBB segmentation and the results are given in Table 5-7.

The results indicate that when segmented dataset was used with changed CIVE equation and combined dataset, the model was able to correctly classify 97% of CMD on leaves, giving us the highest accuracy among other classes, followed by Healthy, CBB, CBSD and CGM giving 91%, 86%, 70% and 62% respectively giving the highest overall validation accuracy of 88.94% compared to all the previous experiments.

**Table 5-7-** Normalized Confusion Matrix of Segmented Dataset Using Changed Formula (eq. 4.2) on Combined Dataset (Cassava Healthy Leaves + Plant Pathology Dataset) with Augmentation obtained from EfficientNet-B4

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.86	0.01	0.01	0.01	0.11
	CBSD	0.03	0.7	0.04	0.22	0.02
	CGM	0.01	0.07	0.62	<b>0.3</b>	0.01
	CMD	0.0	0.01	0.02	<b>0.97</b>	0.0
	Healthy	0.05	0.02	0.01	0.01	0.91

In almost all the experiments, the highest percentage of wrong predictions made by the model was the prediction of CGM leaves as CMD leaves. This can be argued as both the diseases are characterized by yellow patches, so, model got confused in predicting the disease correctly. In addition, our dataset was also biased towards CMD having about 60% images. As a result, model wrongly predicted CGM as CMD.

A comparison of the class wise accuracy results achieved from all 6 experiments is given in Table 5-8. Whereas Table 5-9 gives the overall accuracy (recall), loss, precision, and f-score of all the experiments. It can be seen that accuracy and f-score are the same for all experiments. Comparison shows that the best classification results with minimum loss is achieved by using combined dataset accompanied with augmentation for UNet training to segment CBSD, CGM, and CMD leaves and changed CIVE formula to segment CBB leaves.

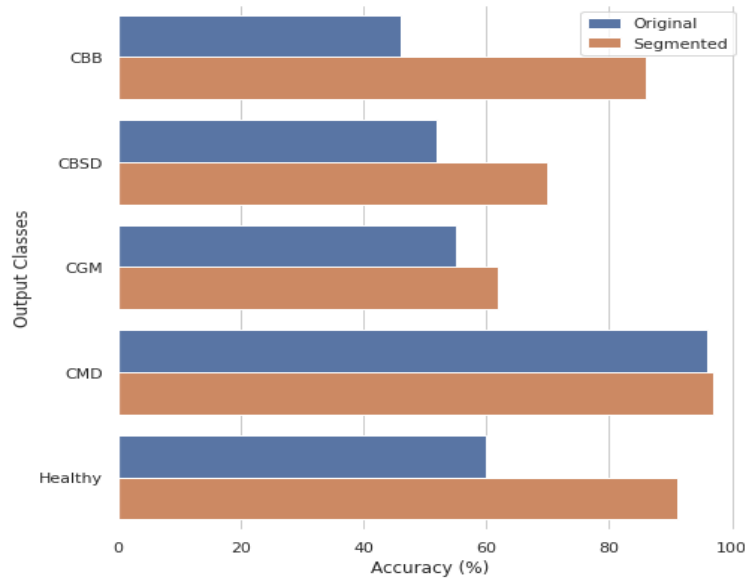
**Table 5-8-** Class-Wise Accuracy Results of all Experiments

OUTPUT CLASSES	Class-wise Accuracy Score (%) of Experiment No.					
	1	2	3	4	5	6
Cassava Bacterial Blight	46.96	52.3	46.12	87.47	56.99	85.98
Cassava Brown Streak Disease	59.28	55.58	59.6	74.41	56.63	70.00
Cassava Green Mottle	56.86	55.87	55.5	60.64	55.5	61.69
Cassava Mosaic Disease	95.77	95.48	95.71	96.19	96.19	97.05
Healthy Leaf	62.76	66.46	64.8	90.39	62.13	90.75
<b>Overall Accuracy</b>	<b>81.43</b>	<b>81.39</b>	<b>81.46</b>	<b>88.83</b>	<b>81.65</b>	<b>88.94</b>

**Table 5-9-** Overall Validation Accuracy and Loss of all Experiments

OUTPUT CLASSES	Overall Accuracy Score (%) and Loss of Experiment No.					
	1	2	3	4	5	6
<b>Validation Accuracy</b>	81.43	81.39	81.46	88.83	81.65	<b>88.94</b>
<b>Validation Loss</b>	0.52	0.53	0.54	0.32	0.53	<b>0.31</b>
<b>Precision</b>	0.81	0.81	0.81	0.89	0.82	0.89
<b>F-Score</b>	0.81	0.81	0.81	0.88	0.81	<b>0.88</b>

Comparing the class-wise results of both original and final segmented dataset, the accuracy rate was increased by 39% for CBB suggesting that most of the CBB leaves were segmented correctly, followed by 28% for healthy leaves, 11% for CBSD, 5% for CGM and 2% for CMD. The same has been shown in Table 5-10. Figure 5-9 shows class-wise increase in accuracy of all 5 output classes.



**Figure 5-9** - A comparison of class-wise accuracies obtained from EfficientNet-B4 on original data and final segmented data

**Table 5-10** – Class wise result of original and segmented datasets in descending order of increase in accuracy

Output classes	Original Accuracy (%)	Segmented Accuracy (%)	Increase in Accuracy (%)
<b>Cassava Bacterial Blight</b>	46.96	85.98	<b>39.02</b>
<b>Healthy Leaf</b>	62.76	90.75	27.99
<b>Cassava Brown Streak Disease</b>	59.28	70.00	10.72
<b>Cassava Green Mottle</b>	56.86	61.69	4.83
<b>Cassava Mosaic Disease</b>	95.77	97.05	1.28
<b>Overall Accuracy</b>	<b>81.43</b>	<b>88.94</b>	<b>7.51</b>

Once final segmented dataset was obtained, it was given as input to all the models of EfficientNet family for training from B0-B6 using the same image size of 512 x 512 x 3 without resizing the image to lower dimensions. The normalized confusion matrices obtained from all the 6 models on final segmented dataset are given in Table 5-11 – Table 5-16.

**Table 5-11-** Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B0

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.89	0.01	0.0	0.02	0.08
	CBSD	0.02	0.71	0.03	0.24	0.0
	CGM	0.0	0.1	0.58	<b>0.31</b>	0.01
	CMD	0.0	0.01	0.02	<b>0.97</b>	0.0
	Healthy	0.04	0.0	0.0	0.01	0.95

**Table 5-12-** Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B1

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.90	0.0	0.0	0.01	0.09
	CBSD	0.02	0.73	0.03	0.22	0.0
	CGM	0.0	0.09	0.54	<b>0.37</b>	0.0
	CMD	0.0	0.02	0.01	<b>0.97</b>	0.0
	Healthy	0.01	0.0	0.0	0.0	0.95

**Table 5-13-** Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B2

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.86	0.02	0.02	0.02	0.08
	CBSD	0.02	0.69	0.02	0.26	0.01
	CGM	0.0	0.1	0.51	<b>0.39</b>	0.0
	CMD	0.0	0.01	0.01	<b>0.98</b>	0.0
	Healthy	0.04	0.0	0.01	0.02	0.93

**Table 5-14-** Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B3

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.82	0.01	0.01	0.02	0.13
	CBSD	0.01	0.72	0.03	0.22	0.03
	CGM	0.0	0.08	0.54	<b>0.37</b>	0.01
	CMD	0.0	0.02	0.01	<b>0.97</b>	0.0
	Healthy	0.03	0.01	0.01	0.01	0.94

**Table 5-15-** Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B5

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.87	0.01	0.0	0.01	0.11
	CBSD	0.02	0.77	0.03	0.16	0.02
	CGM	0.01	0.11	0.54	<b>0.33</b>	0.01
	CMD	0.0	0.02	0.02	<b>0.96</b>	0.0
	Healthy	0.04	0.0	0.0	0.01	0.95

**Table 5-16-** Normalized Confusion Matrix of Final Segmented Dataset obtained from EfficientNet-B6

		Predicted Labels				
		CBB	CBSD	CGM	CMD	Healthy
Actual Labels	CBB	0.85	0.01	0.01	0.01	0.12
	CBSD	0.03	0.70	0.04	0.21	0.02
	CGM	0.0	0.11	0.55	<b>0.33</b>	0.01
	CMD	0.0	0.02	0.02	<b>0.96</b>	0.0
	Healthy	0.03	0.01	0.0	0.01	0.95

Overall validation accuracy, loss, precision, and f-score for all the above mentioned 7 models is given in Table 5-17. Again, f-score and accuracy are same. Table 5-18 shows the class-wise accuracy results of 5 output classes for 7 models on final segmented dataset to reach a conclusion.

**Table 5-17-** Overall Validation Accuracy and Loss from EfficientNet Models Family

OUTPUT CLASSES	Overall Accuracy Score (%) and Loss of EfficientNet						
	B0	B1	B2	B3	B4	B5	B6
Validation Accuracy	<b>89.31</b>	89.13	88.34	88.71	88.94	88.79	88.23
Validation Loss	0.29	0.29	0.31	0.31	0.31	0.32	0.32
Precision	0.89	0.89	0.88	0.88	0.89	0.88	0.88
F-Score	<b>0.89</b>	0.89	0.88	0.88	0.88	0.88	0.88

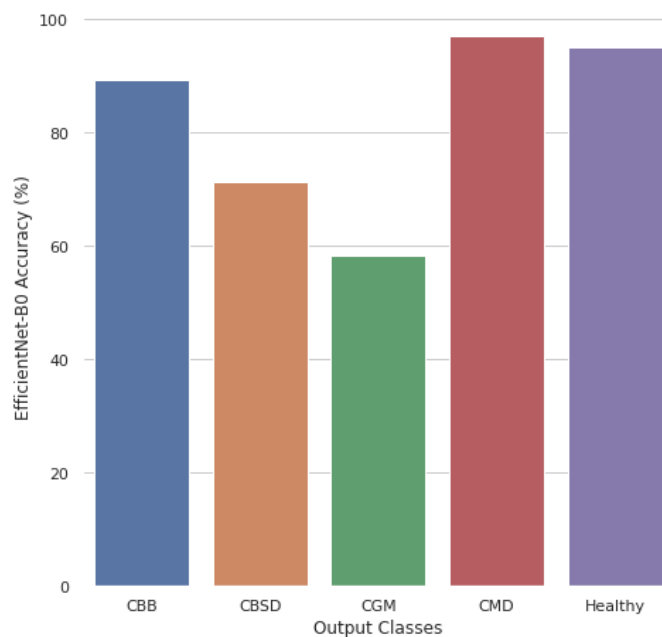
**Table 5-18-** Class-Wise Accuracy Results obtained from EfficientNet Models Family

OUTPUT CLASSES	Class-wise Accuracy Score (%) of EfficientNet						
	B0	B1	B2	B3	B4	B5	B6
Cassava Bacterial Blight	89.12	89.77	86.16	82.01	85.98	86.82	84.7
Cassava Brown Streak Disease	71.35	73.00	68.71	71.71	70.00	76.83	70.42
Cassava Green Mottle	58.2	54.23	51.01	54.13	61.69	54.13	55.13
Cassava Mosaic Disease	96.98	96.98	97.71	97.34	97.05	96.14	96.38
Healthy Leaf	95.02	95.29	93.33	94.4	90.75	94.92	95.11
Overall Accuracy	<b>89.31</b>	<b>89.13</b>	<b>88.34</b>	<b>88.71</b>	<b>88.94</b>	<b>88.79</b>	<b>88.23</b>

Based on the results shown in Table 5-17 and Table 5-18, where accuracies and losses obtained from all models are given, it is clear that EfficientNet-B0 outperformed all the other models giving an accuracy score of 89.31% with minimum loss (0.29). Comparing the confusion matrices of all models, the lowest percentage of wrong predictions (predicting CGM as CMD leaves) made by model is 30% by B4 followed by 31% by B0. Apart from CGM, B0 has correctly predicted all other classes comparing to B4.



The model has correctly predicted 97% of CMD, followed by Healthy leaves, CBB, CBSD and CGM with accuracy scores of 95%, 89%, 71% and 58%. CGM has the lowest accuracy score and most of the CGM leaves are predicted as CMD and as explained earlier, this can be because both the diseases are characterized by yellow patches, so, model got confused in predicting the disease correctly and the dataset was also heavily biased towards CMD having about 61% images given by Figure 3-2. As a result, model wrongly predicted CGM as CMD. Figure 5-10 shows the final class-wise results obtained from EfficientNet-B0 on final segmented dataset.



**Figure 5-10-** Class-wise Accuracy Score obtained from EfficientNet-B0 on final Segmented Dataset

Since the results obtained from one-time data splitting (in any ratio) can be biased or they may be a best case among all the samples, so cross validation was used to be confident about our algorithm's performance. A 7-Fold cross validation was performed to determine if the model performs well on all the folds or not. The accuracy results obtained from all the folds are given in Table 5-19 along with the mean accuracy and standard deviation.

The standard deviation value less than 1 for accuracy score from Table 5-19 indicates that the results of each fold are mostly similar with a very little variance. It can be seen from the same table that the highest score achieved is 90.55% with minimum loss of 0.26 and lowest score is 88.39% with loss of 0.3 giving an average accuracy score of 89.73% plus a standard deviation of 0.82.

**Table 5-19-** 7-Fold Cross Validation Results obtained from EfficientNet-B0 on Cassava 2020 dataset

<b>Fold</b>	<b>Accuracy (%)</b>	<b>Loss</b>	<b>F-Score</b>
<b>1</b>	88.39	0.3046	0.88
<b>2</b>	90.21	0.2779	0.9
<b>3</b>	88.83	0.3010	0.88
<b>4</b>	89.84	0.2814	0.89
<b>5</b>	90.47	0.2815	0.9
<b>6</b>	89.79	0.2866	0.89
<b>7</b>	90.55	0.2698	0.9
<b>Mean ± S.D.</b>	89.73 ± 0.82	0.29 ± 0.01	0.89 ± 0.01

As the dataset was imbalanced, so detailed augmentation was applied on minority classes to increase the diversity of data. Segmented balanced dataset when trained on EfficientNet-B0 for 50 epochs gave an overall increase in accuracy by about 0.25% compared to the original dataset, making it clear that augmentation increases the performance of a deep learning model. The accuracy results for balanced dataset obtained from all the folds are given in Table 5-20 along with the mean accuracy and standard deviation.

**Table 5-20-** 7-Fold Cross Validation Results obtained from EfficientNet-B0 on Balanced Cassava 2020 dataset

<b>Fold</b>	<b>Accuracy (%)</b>	<b>Loss</b>	<b>F-Score</b>
<b>1</b>	88.42	0.3089	0.88
<b>2</b>	90.32	0.2762	0.9
<b>3</b>	90.04	0.2812	0.9
<b>4</b>	89.95	0.2802	0.89
<b>5</b>	90.21	0.2878	0.9
<b>6</b>	90.32	0.2782	0.9
<b>7</b>	90.55	0.2704	0.9
<b>Mean ± S.D.</b>	89.97 ± 0.71	0.28 ± 0.01	0.9 ± 0.01

Comparing the results of all folds of both original and balanced dataset, it can be seen that accuracy increased in almost all the folds when balanced dataset was used giving a means accuracy score of 89.97% with a loss of 0.28. The standard deviation is also less as compared to original data i.e. 0.71. This low value indicates that the results of each fold are mostly similar with a very little variance. The increase in accuracy with balanced dataset using augmentation suggests that validation accuracy increases with the increase in dataset size as given in [4] opposed to the critic that large sized and heavy augmentation results in overfitting. It in fact, increases the generalizing capability of a deep learning model [42]. However, the training time was also increased as the size of dataset was almost 2.5 times more than the original data and is given in Table 5-21.

**Table 5-21-** Training time and the dataset size for classification using EfficientNet-B0

<b>Dataset</b>	<b>Dataset size</b>	<b>Time per Epoch (s)</b>	<b>Accuracy (%)</b>
<b>Original</b>	21,397	213	89.73 ± 0.82
<b>Balanced</b>	54,902	568	89.97 ± 0.71

The results obtained indicate that augmentation technique along with segmentation provides fast and easiest way to improve the performance of a model by removing irrelevant features such as background noise. Further, the application of transfer learning also provides a convenient way to produce quality results in limited resources irrespective of dataset size.

Since no other article has used Kaggle 2020 dataset for cassava leaf disease classification, for comparison purpose, the same methodology was used to train Kaggle 2019 dataset for cassava disease classification [40][41]. Cassava 2019 dataset has the same 5 output labels for CBB, CBSD, CGM, CMD and Healthy Leaves but their distribution in the dataset is different with CBSD and CMD having the highest percentage and is given by Figure 5-11. Also, the dataset size is small as compared to 2020 dataset containing 5,656 training images belonging to 5 output classes.

Same segmentation technique was used to segment CBB leaves using eq (4.2) and rest of the leaves using UNet trained model on cassava healthy leaves and Plant Pathology dataset. Using all the default parameters, efficientNet-B0 was trained for 100 epochs. For evaluation purpose, 7-fold cross validation was used. The result obtained from cross validation are given in Table 5-22 along

with overall accuracy and loss. Comparison of all the articles that have used 2019 dataset for classification is given in Table 5-23.

Samples division among 5 classes in Cassava Disease Dataset 2019

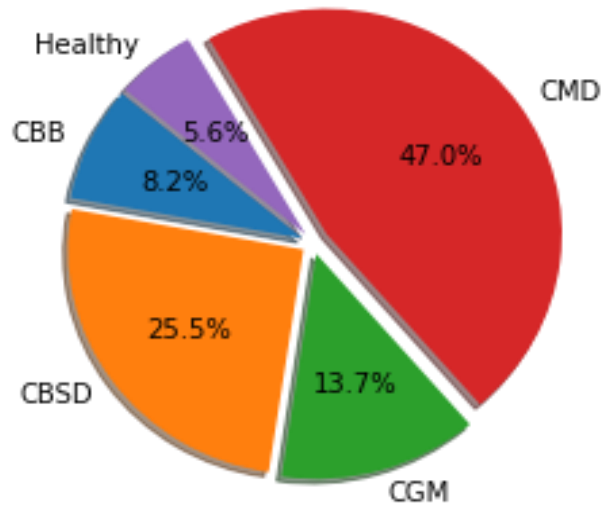


Figure 5-11-Samples Distribution among 5 classes (Cassava 2019 Dataset)

Table 5-22- 7-Fold Cross Validation Results obtained from EfficientNet-B0 on Cassava 2019 dataset

Fold	Accuracy (%)	Loss
1	87.99	0.3401
2	89.98	0.3238
3	86.51	0.3566
4	91.21	0.2850
5	89.23	0.3358
6	90.72	0.2696
7	90.22	0.3137
<b>Mean ± S.D.</b>	<b>89.41 ± 1.65</b>	<b>0.31 ± 0.03</b>

Table 5-23 shows the model used in the literature along with the preprocessing technique used (if any), training epochs to train the model, input image resolution, data splitting used to evaluate the model and the accuracy score value.

Based on the results from Table 5-23, it is clear that our proposed work gave highest score as compared to previous work on the same dataset in less resources i.e., 100 epochs where each epoch took only 1 minute, giving 89.41% validation accuracy with a standard deviation of 1.65 using 7-fold cross validation.

**Table 5-23-** Comparison of Proposed Work with Previous Work on Cassava 2019 Dataset

Paper	Model	No. of Epochs	Time per Epoch (s)	Input Image Resolution	Data Splitting	Accuracy (%)
[9]	Custom CNN (CLAHE)	124	3600	448 x 448 x 3	3-Fold CV	88
[11]	MobileNet-V2	40	-	224 x 224 x 3	70:30	67.5
[12]	PCNN					52.87
	DRNN (Gamma Correction)	80	-	224 x 224 x 3	-	46.24
<b>Proposed Work</b>	EfficientNet-B0 (Segmentation)	100	54	512 x 512 x 3	7-Fold CV	<b>89.41 ± 1.65</b>

Although, in [9] and [12] authors have used various techniques to increase the size of dataset for handling class imbalance issue and they have achieved a very high accuracy score concluding that accuracy increases with the increase in dataset size, we have used only imbalanced dataset results from these articles for comparison purpose. As, 2019 dataset was also imbalanced, so accuracy alone cannot be used to reach at reliable results. For this purpose, precision, recall and f-score were also measured and compared, and the results are given in Table 5-24.

Referring to f-score values from Table 5-24, Bacterial Blight, Green Mottle, and healthy leaves have a much better score giving 0.95, 0.82 and 0.96 respectively compared to previous work. However, the rest of classes i.e., Brown streak Disease and Mosaic disease results are also better

then [12] but are comparable to [9] with a difference of 0.02 for CBSD and 0.04 for CMD giving f-score of 0.90 and 0.93, respectively.

**Table 5-24-** Class wise comparison of Proposed work with literature using Kaggle 2019 Cassava Disease Dataset

Paper	Evaluation Measures	CBB	CBSD	CGM	CMD	Healthy	Weighted Average
[1]	Precision	0.81	0.92	0.8	0.97	0.7	0.9
	Recall	0.83	0.91	0.72	0.96	0.67	0.88
	F-Score	0.82	<b>0.92</b>	0.76	<b>0.97</b>	0.69	0.89
[2]	Precision	0.27	0.5	0.25	0.55	0.33	0.46
	Recall	0.03	0.48	0.03	0.86	0.02	0.52
	F-Score	0.05	0.49	0.05	0.67	0.04	0.49
Proposed Work	Precision	0.93	0.89	0.87	0.9	0.98	0.9
	Recall	0.95	0.90	0.72	0.94	0.94	0.9
	F-Score	<b>0.94</b>	0.89	<b>0.79</b>	0.92	<b>0.96</b>	<b>0.9</b>

The comparison of state-of-the-art techniques to classify cassava leaves with this research work shows that our proposed work outperformed all the previous work giving 89.41% accuracy and f-score value of 0.9. Among other classes, the lowest accuracy and f-score is for CGM because its symptoms are like CMD in appearance and the dataset was also biased towards CMD shown in figure 5-11. The rest of the classes have a very good recall and f-score indicating that segmentation module (UNet model along with the modified equation) has accurately segmented the diseased as well as healthy leaves. Based on the above results and keeping in mind that there is still some noise in the form of few poorly segmented images for 2 classes: CBSD and mostly CGM (with lowest f-score among other labels), it can be argued that by using segmentation data, our model was able to successfully classify most of the leaves correctly.

## Chapter 6

### CONCLUSION & FUTURE WORK

This work presented the use of a novel deep learning architecture, EfficientNet to classify cassava leaf disease images quickly and accurately. Cassava plant is one of the most widely grown food crop not only because it is cost-efficient, producing more yield in low budget but it can also withstand harsh environmental conditions. However, it is highly vulnerable to many viral and bacterial attacks resulting in various diseases. Some of those diseases are widespread and have severe devastating effects if not timely detected and treated. In this research, cassava leaf disease classification dataset, taken from Kaggle competition was used that contained about 21K images. As our input image has 512x512 resolution, so instead of resizing the images to some lower resolution, we used EfficientNet models for classification purpose to avoid any information loss and to capture more fine-grained patterns from images. Further, to improve performance, we have used U-Net, a semantic segmentation model, to precisely segment the leaves from images. Our results showed reasonable performance giving a mean accuracy rate of 89.73% on original dataset and 89.97% on balanced dataset with EfficientNet-B0 using 7-fold cross validation. For comparison purpose, Kaggle 2019 dataset for cassava disease classification was used, that gave mean accuracy score of  $89.41 \pm 1.62$  using 7-fold cross validation and f1-score of 0.9 leading all previous work on same dataset. Based on the obtained results, it can be concluded that:

- Segmentation technique provides fast and easiest way to improve the performance of a model by removing irrelevant and unnecessary details from background.
- Combined method of segmentation using UNet along with the modified Color Index Vegetation Extraction equation gave promising scores leading all state-of-the-art results, concluding that original CIVE equation can be modified accordingly to accurately segment vegetation as well as diseased part in leaves.

- Augmentation can be used along with CIVE to extract diseased leaves from background. For that, UNet architecture is best to use as it precisely locates the desired object of interest from background and gets perfectly trained even on very few samples.
- Further, augmentation can not only handle class imbalance issue by increasing the sample size of minority classes, but it also aids in generalizing the model that results in improved performance.
- For classification purpose, transfer learning provides fast, effortless, and convenient way to produce quality results in limited resources.

Despite we have used combined method along with the modified CIVE equation and augmentation to segment healthy as well as diseased leaf images, there is still a room to improve segmentation module so that it can handle cluttered background in a more efficient manner. Figure 6-1 shows poorly segmented images where some of the leaf parts are also masked along with the background during segmentation because of some natural image capturing conditions. To cater for this issue, object localization technique can be explored further to reduce unwanted background region without excluding any part of leaf (either diseased or healthy).



**Figure 6-1**-Poorly Segmented Diseased Leaf Image



## REFERENCES

1. M. A. Jasim and J. M. Al-Tuwaijari, "Plant leaf diseases detection and classification using image processing and deep learning Techniques," Proc. 2020 Int. Conf. Comput. Sci. Softw. Eng. CSASE 2020, pp. 259– 265, 2020.
2. E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Comput. Electron. Agric., vol. 161, no. March 2018, pp. 272–279, 2019.
3. E. J. McCallum, R. B. Anjanappa, and W. Gruissem, "Tackling agriculturally relevant diseases in the staple crop cassava (*Manihot esculenta*)," Curr. Opin. Plant Biol., vol. 38, pp. 50–58, 2017.
4. J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," Comput. Electron. Agric., vol. 153, no. March, pp. 46– 53, 2018.
5. G. Owomugisha and E. Mwebaze, "Machine learning for plant disease incidence and severity measurements from leaf images," Proc. - 2016 15th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2016, pp. 158–163, 2017.
6. W. Abdullakasim, K. Powbunthorn, and J. Unartngam, "An image Analysis technique for recognition of brown leaf spot disease in cassava," Tarım Makinaları Bilim. Derg., vol. 7, no. 2, pp. 165–169, 2011.
7. O. Emuoyibofarhe, J. O. Emuoyibofarhe, S. Adebayo, A. Ayandiji, O. Demeji and O. James, "Detection and classification of cassava diseases using machine learning," in International Journal of computer science and Software Engineering., vol. 8, issue 7, pp. 166–176, 2019.
8. A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," Front. Plant Sci., vol. 8, no. October, pp. 1–7, 2017.

- 
9. 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," *Egypt. Informatics J.*, vol. 22, no. 1, pp. 27–34, 2021.
  10. I. Sangbamrung, P. Praneetpholkrang, and S. Kanjanawattana, "A novel automatic method for cassava disease classification using deep learning," *J. Adv. Inf. Technol.*, vol. 11, no. 4, pp. 241–248, 2020.
  11. H. R. Ayu, A. Surtono, and D. K. Apriyanto, "Deep learning for detection cassava leaf disease," *J. Phys. Conf. Ser.*, vol. 1751, no. 1, 2021.
  12. D. O. Oyewola, E. G. Dada, S. Misra, and R. Damaševičius, "Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing," *PeerJ Comput. Sci.*, vol. 7, p. e352, 2021.
  13. S. S. Keh, "Semi-supervised noisy student pre-training on efficientnet architectures for plant pathology classification," *arXiv*, 2020.
  14. Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol. Inform.*, vol. 61, no. June 2020, p. 101182, 2021.
  15. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.
  16. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018.
  17. I. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-andExcitation Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 2020.
  18. T. Kataoka, T. Kaneko, H. Okamoto, and S. Hata, "Crop growth estimation system using machine vision," *IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM*, vol. 2, pp. 1079–1083, 2003.
  19. O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351, pp. 234–241, Nov 2015
  20. Akeret, Joel and Chang, Chihway and Lucchi, Aurelien and Refregier, Alexandre, "Radio frequency interference mitigation using deep convolutional neural networks," in *journal of Astronomy and Computing*, Vol 18, pp. 35–39, 2017.

- 
21. F. Guira, K. Some, D. Kabore, H. S. Lingani, Y. Traore and A. Savadogo, “Origins, production, and utilization of cassava in Burkina Faso, a contribution of a neglected crop to household food security,” in *Food Sci Nutr.*, pp. 415–423, May 2017.
  22. [https://www.cassavastarchmachine.com/news/Industry\\_News/list\\_of\\_products\\_made\\_from\\_cassava\\_251.html](https://www.cassavastarchmachine.com/news/Industry_News/list_of_products_made_from_cassava_251.html)
  23. <https://www.webmd.com/diet/health-benefits-of-cassava-flour>
  24. <https://www.verywellfit.com/cassava-nutrition-facts-calories-carbs-and-health-benefits-4177847>
  25. <https://www.dawn.com/news/1586121>
  26. M. S. Mirza and S. T. Ahmed, “Malnutrition Evidence and Solutions for Pakistan,” in *ACTA Scientific Paediatrics*, Vol. 2, Issue 9, Sept. 2019.
  27. <https://www.unicef.org/pakistan/media/1951/file/Final%20Key%20Findings%20Report%202019.pdf>
  28. <http://www.parc.gov.pk/index.php/en/59-research-institutes?start=9>
  29. Cassava Leaf Disease Classification. <https://www.kaggle.com/c/cassava-leaf-disease-classification/>
  30. *Plant Protection in Tropical Root and Tuber Crops*, 2015
  31. Heuzé V., Tran G., 2016. Cassava leaves and foliage. Feedipedia, a programme by INRAE, CIRAD, AFZ and FAO.
  32. J. A. Montagnac, C. R. Davis, S. A. Tanumihardjo, “Nutritional Value of Cassava for Use as a Staple Food and Recent Advances for Improvement,” June 2009
  33. *THE CASSAVA TRANSFORMATION IN AFRICA* by Felix I. Nweke, Michigan State University, USA.
  34. A. Adekunle, P. Osazuwa and V. Raghavan, “Socio-economic determinants of agricultural mechanisation in Africa: A research note based on cassava cultivation mechanization,” in *Technological Forecasting & Social Change*, Nov. 2016.
  35. Alabi, O. J., Kumar, P. L., and Naidu, R. A. 2011. Cassava mosaic disease: A curse to food security in Sub-Saharan Africa. Online. APSnet Features. doi:10.1094/APSnetFeature-2011.
  36. Burns, A.; Gleadow, R.; Cliff, J.; Zacarias, A.; Cavagnaro, T. Cassava: The Drought, War and Famine Crop in a Changing World. *Sustainability*, pp. 3572-3607, Nov. 2010.

37. Save and Grow Cassava. A Guide to Sustainable Production Intensification.  
<http://www.fao.org/3/i3278e/i3278e.pdf>
38. Plant Pathology 2021-FGVC8, <https://www.kaggle.com/c/plant-pathology-2021-fgvc8>
39. Leaf-segmented-224-pt2021, <https://www.kaggle.com/rustyelectron/leafsegmented224pt2021>
40. Cassava Disease Competition 2019, <https://www.kaggle.com/c/cassava-disease/>
41. Mwebaze, Ernest & Gebru, Timnit & Frome, Andrea & Nsumba, Solomon & Tusubira, Jeremy, “iCassava-2019Fine-Grained Visual Categorization Challenge,” pp. 4321-4326, Dec 2019.
42. Romero, Nelson, M. Gutoski, L. T. Hattori and H. S. Lopes. “The Effect of Data Augmentation on the Performance of Convolutional Neural Networks.” in *ChemBioChem*, 2018.