Diagnosis of bacterial leaf blight, brown spots and leaf smut rice plant diseases.

A Dissertation Submitted for the Degree of Master of Computer Science

G. R. I. L. Jayasooriya

University of Colombo School of Computing

2020

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: G. R. I. L. Jayasooriya

Registration Number: 18440415

Index Number: 2018/MCS/041

______________________ 30/11/2021

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr. G.R.I. L. Jayasooriya under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by,

Supervisor Name: Dr. Samantha Mathara Arachchi

30.11.2021
____

Signature of the Supervisor & Date

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided me the possibility to complete this research. First of all, I’m very grateful to MCS coordinators and academic staff of University of Colombo School of Computing (UCSC) for providing academic inputs and guidance throughout this invaluable degree program and a special gratitude I give to my research supervisor, senior lecturer Dr. Samantha Mathara Arachchi, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my research especially in writing this dissertation.

I cannot express enough thanks to my family and friends for their immense support rendered throughout the course of this endeavor.Finally, I would like to acknowledge everyone who stood beside me and supported me to complete this dissertation successfully.

ABSTRACT

Rice is the most widely consumed food product and one of the extensively cultivated crops in Sri Lanka. Considering the human population, food is one of the major problems that Sri Lanka might be facing in the near future. Therefore, increasing the crop yield is one of the major needs of the country. When rice crops are infected with diseases, it results in a loss of crops. Therefore it is important to identify the disease in the early stage of infection to prevent the damage that can be done. Disease identification is very difficult without having a clear understanding. With the advancement of new technologies, researchers are interested in identifying paddy diseases through machine learning and image processing techniques to help farmers to identify infectious diseases accurately.

It is a challenge to make observations that are closer to how the human eye is capable of observing the paddy leaf to diagnose the infected disease. In this project, an algorithm was developed to check whether the image contains different changes to the paddy leaf by considering the green color pixels and their variance. OpenCV libraries have been used to develop the algorithm for feature extraction and those features were used as attributes to the LightGBM algorithm to classify the disease images with over 80% accuracy.

Recent Deep Learning model developments have shown that automated image recognition systems with Convolutional Neural Network (CNN) models can be accurate in such problems. Since the Rice leaf disease image dataset is not available, the UCI machine learning repository provided dataset has been used to develop paddy leaf disease, classification models with data augmentation techniques to increase the number of images of the dataset while developing the model.

OpenCV based algorithm has been developed as a python HTTP (Hypertext transfer protocol) service including the ability to identify healthy paddy leaf and leaf validation. A web-based program has been developed using the CNN model with an image processing algorithm to diagnose whether a paddy leaf is infected with bacterial leaf blight, leaf smut, or brown spot using an infected paddy leaf image. Image preprocessing and segmentation techniques have been used to increase the accuracy of the model. This developed system will be useful for people who do not have much knowledge about symptoms of paddy diseases.

LIST OF PUBLICATIONS

i. G.R.I.L.Jayasooriya, Samantha Mathara Arachchi, "LITERATURE REVIEW OF DIAGNOSIS RICE LEAVES DISEASES USING IMAGE PROCESSING", IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P-ISSN 2349-5138, Volume.8, Issue 1, Page No pp.320-333, March 2021, Available at: http://www.ijrar.org/papers/IJRARJFM1392.pdf

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Rice is the principal food of Sri Lanka. Therefore, Paddy cultivation in Sri Lanka is prioritized in the agriculture industry (Gunawardana, 2018). Further, early Ceylon had been globally recognized as one of the cardinal paddy exporters and the traditional and early farmers of the country were self-sufficient in rice production (Gunawardana, 2018; Seneviratne and Jeyanandarajah, 2004). Rice is taken as the main meal by at least half the people in the world, hence it can be considered as the primary source of energy of about half of the world's people. Depending on rice's strain, it can contain decent amounts of fiber, protein, Vitamin B, iron, and manganese (Chaudhari et al., n.d.). This means the nutritional factors of rice are effective against malnutrition.

Traditional varieties were cultivated throughout the country until the early 1960s (Rambukwella and Priyankara, 2016). Since it usually gives low yields, people are accustomed to applying artificial fertilizers to increase yields regardless of the possibility of the plant being attacked by pathogens (Seneviratne and Jeyanandarajah, 2004). Every year farmers face a loss of yield and financial losses due to the pets and the diseases on the rice plants (Seneviratne and Jeyanandarajah, 2004). Many diseases can infect paddy plants (Rambukwella and Priyankara, 2016; Seneviratne and Jeyanandarajah, 2004). Due to infection caused by viruses, fungus, and bacteria, there is a huge loss of rice quality and quantity (Mangla et al., 2019; Sethy et al., 2020). Traditional farmers identified those diseases using their previous experiences. It is important to identify the disease that has infected the paddy field to control the disease spread to other plants. The farmers who do not have sound knowledge about paddy fielding cannot identify diseases efficiently without the help of others.

Considering the above-discussed difficulties, a model with image processing techniques has been proposed to diagnose infected rice diseases. In this project, three diseases have been used to the proposed model. Features that can be identified through naked eye observation, will be used to separate a disease from others. Every disease has shown one or more symptoms base on the disease stage. Humans only can identify a disease when it shows symptoms.

1.1 Motivation

Considering the advancement of technology, the younger generation of farmers uses the newly developed technologies to increase the yield. The traditional methods have been overcome by new and easier technologies. Diagnosis requires a methodology rather than observation with the naked eye.

Numerous diseases of rice, caused by bacteria, fungi, viruses, and nematodes have been recorded in literature. The region in which the rice is grown is the reason for dome diseases. Some are both regional and internationally important. Others arise in local areas. Some diseases reach epidemic proportions and cause serious crop losses while others cause only negligible crop losses. The rice diseases that are of national importance and might cause severe crop losses in Sri Lanka are considered for this study.

Animal infestations, natural disasters (floods, droughts, etc.), bacterial, fungal, and viral infections, and pests are some of the reasons to reduce paddy yield (Institute, n.d.). Such attacks have been recently identified with the assistance of experts with good experience in paddy cultivation. Bacterial blight, Bacterial leaf streak, Blast, Brown spot, False smut Sheath blight are the most common diseases transmitted by microorganisms (Institute, n.d.). Traditional farmers identified diseases in the paddy field using their previous experiences. It is important to identify the disease which has infected to paddy field to apply the correct treatment to control the disease and prevent the other plants from the infection. The farmers who do not have good knowledge about paddy fielding, cannot identify correct diseases easily without the help of an expert.

Every part of the paddy plant is susceptible to pathogenic attacks. Symptoms of each disease have been used to distinguish one disease from another. Alterations in the appearance of the rice plant can be used to differentiate diseases. Therefore, if we devise a methodology to perform the above process without human interaction, the system must be able to distinguish the infected zone by using the appearance of that area.

With the development of the technologies in the world, human intersection to agriculture field has been decreased cause of busyness with their day-to-day activities. But the agriculture field should be given more priority than any other field because food is the third most important thing to animals after water and Oxygen. Hence, smart farming will reduce the complexity of agriculture.

Both diseases and pests are making some discoloration or abominable shape changes when infected. Here, we proposed image processing techniques to separate diseases using an image that is taken from the paddy field. As a startup, we have used three diseases that are infected to paddy leaves and make some changes that can be clearly identified over naked eye observation. The system should be capable of extracting the image features that can be used in the diagnosis and the symptoms of each disease can vary on rice varieties, environment, and climate situations, and disease level.

Under the concept of smart farming, it is important to diagnose plant diseases without human intervention to apply treatments to control the infection. Machine learning and image processing methods were suggested for diagnosis using field images. In the near future, the human generation would be smart enough to control their farms remotely without physical attendance.

It will be a challenge to observe symptoms of paddy plant diseases through an image without naked eye observation. Color changes of the image will be used as the main feature that can be extracted from an image. Images contain not only the paddy plant but also some other objects. When image processing is applied to analyze paddy images to diagnose the disease on a paddy plant, paddy plant boundaries should be distinguished and should be capable to remove other objects from the image.

1.2 Statement of the problem

Paddy plants can be affected by various fungi, pathogens, and viruses and those diseases directly cause a loss of yield. Farmers should have excellent knowledge about paddy disease to identify the infected disease correctly. Since rice is one of the major foods in Sri Lanka, it is essential to provide technical support other than the naked eye observation and prediction for youngsters and people who are lack knowledge about paddy field diseases. Considering the development of computer vision technology, image processing-based disease diagnosing will be useful in smart farming to take observations and predict paddy field status without high human intervention.

1.3 Research Aims and Objectives

The objective of this project is to develop a web-based system for the individual identification of bacterial leaf blight, leaf smut, and brown spots paddy leaf diseases using image processing techniques with using an image of an infected leaf and to diagnose infectious diseases using the uploaded image. Diagnosis is made by image processing methods, which consider changes in the color of the uploaded image and discolored areas and patch sizes, as well as patches and patch locations on the leaf. The system should be able to detect infectious diseases through an image taken from the mobile phone.

1.3.1 Aim

This project aims to provide a platform for paddy farmers to identify whether a paddy field is infected with bacterial leaf blight, leaf smut, or brown spot disease using a paddy leaf image and validate the result using image processing techniques.

1.3.2 Objectives

Objectives of this research are as follows.

- Diagnosis of bacterial leaf blight, brown spot, and leaf smut paddy leaf diseases using image processing.
- Develop a web-based application to predict paddy leaves diseases using image processing.
- Identify infected (damaged) paddy leaf from an image.

1.4 Scope

Both diseases and pests are making some discoloration or abominable shape changes when infected. Here, we proposed an image processing technique to separate diseases through the images that are taken from the paddy field. As a startup, we have used three diseases that are infected to paddy leaves and make some changes that can be clearly identified over naked eye observation. The system should be capable of extracting the image features that can be used to diagnose and the symptoms of each disease can vary on rice varieties, environment and climate situations, and disease level. Domain experts should involve in this stage with the symptoms of each disease that is unique to the relevant area.

Bacterial leaf blight, Leaf smut, and brown spots disease are considered in this research. The database provided by the UCI Machine Learning repository was used for developing and testing purposes in system development. The system will be able to diagnose the above-mentioned diseases through image processing techniques. Color changes, patch shapes, patch size, etc. are considered in the image processing process. The symptoms of each disease are as follows.

- Bacterial Leaf Blight (Figure 1-1)

Bacterial Leaf Blight is a disease caused by bacteria called *Xanthomonas Oryzae* pv. *Oryzae*. This disease mostly occurs in rainy and dry-wet seasons (Ihsan et al., 2020). Bacterial leaf blight is one of the three major rice diseases in Sri Lanka (Nugaliyadde et al., n.d.).

Figure 1-1 : Bacterial leaf blight

- Brown Spot (Figure 1-2)

Brown spot is caused by the fungus *Helmintosporium Oryzae*. The death of young plants and reduce grain quality are the consequences of brown Spot disease (Ihsan et al., 2020).

Figure 1-2: Brown Spot

- Leaf Smut (Figure 1-3)

Leaf Smut caused by the fungus *Entyloma oryzae* is a widely distributed, but somewhat minor disease of rice. Slightly raised and angular black spots on both sides of the leaves. Though it's rare, it also can produce spots on leaf sheaths. The black spots are about 0.5 to 5.0 millimeters long and 0.5 to 1.5 millimeters wide and many spots can be found on a leaf, but the spots remain distinct from each other (Groth and Agcenter, n.d.).

Figure 1-3: Leaf smut

1.5 Structure of the Thesis

The remainder of the research study in terms of chapter structure is unfolded here. The first chapter discussed the purpose along with the introduction of the problem study. Paddy cultivation history and the importance of paddy leaves diseases diagnosis without human intervention have been discussed along with the scope that is addressed in this study.

Chapter two consists of the previous works study about the paddy leaves diseases and different technologies and materials, that were used to diagnose paddy diseases. Moreover, previous works of image preprocessing, feature extraction, and diagnosis methods and materials have been mentioned in this chapter.

The methodology of selected paddy leaves diseases diagnosis has been discussed in chapter three. Two models have been discussed which were developed to diagnose the disease using the LightGBM machine learning algorithm and Convolutional Neural Network model.

The evaluation of the proposed models and their accuracy levels of being successful along with the results of the implementations of the developed web program has been discussed in chapter four. Chapter five is unfolded with the feature view and conclusions of this research.

CHAPTER 2
LITERATURE REVIEW

This chapter unfolded background of the research and have been discussed about previous works and image processing techniques which were used in paddy leaves disease diagnosis. Moreover, Scientific material and methods that were used in this research also discussed under this section.

2.1 A Literature Review

Background of paddy leaves disease diagnosis using image processing along with most common paddy leaves diseases are discussed in this section.

2.1.1 Background

Rice is the principal food of Sri Lanka. Therefore, Paddy cultivation in Sri Lanka is given a prominent place in the agriculture industry (Arjuna Abeysinghe Gunawardana, 2018). Early traditional farmers were self-sufficient in rice production and ancient Ceylon is said to have been among the major paddy exporters in the world (Arjuna Abeysinghe Gunawardana, 2018; Seneviratne and Jeyanandarajah, 2004). Rice is used as the main meal by half the world's population and can be considered as their primary source of energy. Depending on the strain of rice, it can contain decent amounts of fiber, protein, Vitamin B, iron, and manganese (Chaudhary et al., 2012). This means it can play a vital role against malnutrition.

Traditional varieties were cultivated throughout the country until the early 1960s (Rambukwella and Priyankara, 2016). Since it usually gives low yields, people are accustomed to applying artificial fertilizers to increase yields regardless of the possibility of the plant being attacked by pathogens (Seneviratne and Jeyanandarajah, 2004). Every year farmers face a loss of yield and financial losses due to the pets and the diseases on the rice plants (Seneviratne and Jeyanandarajah, 2004). There are a lot of diseases that can infect paddy plants (Rambukwella and Priyankara, 2016; Seneviratne and Jeyanandarajah, 2004). Due to infection caused by pests, viruses, fungus, and bacteria, there is a huge loss of quality and quantity of rice (Mangla et al., 2019; Sethy et al., 2020).

Diseases affect rice plants in two ways, such as seed germination and seedling establishment. This weakens the stability and prevents the normal functioning of the plant parts of different sizes due to damage to the plant by pathogenic agents (Seneviratne and Jeyanandarajah, 2004).

Gain spotting and discoloration, gain sterility, premature senescence of rice crop, and yellowing have been identified as the most reported field problems (Nugaliyadde et al., n.d.).

2.1.2 Paddy Diseases

Paddy diseases are caused by pathogens, insect pests and many other disturbances such as nutrient deficiencies and abnormal environmental conditions. Plant pathogens can be parasitic, bacterial, viral, or nematodes that can damage the plant parts above or underneath the ground. The main paddy leaves diseases have been listed below (Sethy, et al., 2020).

- Leaf blast : Symptoms include small rounded, dark to oval-spotted, diamond-shaped or linear-shaped, elongated diamond-shaped or narrow reddish-gray dead areas in the center, with a narrow reddish-brown border with a gray-white center (Figure 2-1)(Groth and Agcenter, n.d.).

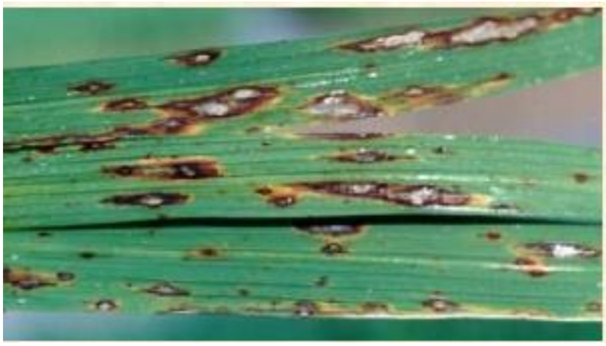

Figure 2-1 : Leaf blast

- Brown spot: Symptoms consist of round to oval dark brown to yellow or golden lesions; As the lesions become larger they are rounded and the central area is neurotic, gray and the border of the lesions is reddish-brown to dark brown (Figure 1-2) (Groth and Hollier, n.d.).
- Sheath blight: Symptoms as shown in Figure 2-2 are white-brown, dark gray-brown or brownish-brown or narrow bands with a wide band of broad bands. The lesion begins at the base of the leaf and spreads from the leaf sheath or infected site and the fungal mycelium can be seen under very wet conditions. On the surface of the leaves can form structures of the fungus called sclerotia. (Groth and Agcenter, n.d.).

Figure 2-2 : Sheath blight

- Leaf scald: The lesion consists of broad bands of deadly gray dead tissue with narrow reddish-brown bands. Chevron band patterns from leaf tips or edges. Sometimes yellow or golden border spots on the edges of the wounds are symptomatic, as shown in Figure 2-3 (Groth and Agcenter, n.d.).

Figure 2-3 : Leaf scald

- Bacterial leaf blightThe bruises are characterized by elongated lesions at the ends or near the border. A few inches long, it turns white and yellow, and later becomes gray due to the sporophytic fungus; shown in **Error! Reference source not found.** (Groth and Agcenter, n.d.).

- Leaf smut: The small black linear lesions on the leaf blades may be dark golden or light brown, the tips of the leaves dry out and turn gray as the plant matures, the upper sheath lesions being symptomatic; Shown in Figure 1-3 (Groth and Agcenter, n.d.).

2.1.3 Paddy Disease Diagnose Techniques.

Shah et al., (2016) have proposed to prepare an image database using the images which are acquainted from a live farm. Using a digital camera, images have been taken directly in digital form with numerical values so that digital image processing can be applied. Phadikar and Sil, (2008) classified diseased rice plant images using Self Organizing Map (SOM) neural network (Matera, 1998) where train images are taken by extracting features of the infected parts of the leaves. Zooming algorithms (Battiato et al., 2002) and simple computationally efficient techniques have been used to extract images' features.

Mukherjee et al.,(2010) converted the original resized image into a gray image such that the pixels corresponding to the leaf image are the same. Then histogram was used in calculating the change in the pick value. Following steps have been proposed to diagnose the rice blast, Bacterial Leaf Bligh, and Rice tungro using image processing. The proposed steps for diagnosis are as follows.

1. Image Enhancement
2. Image pre-process
3. Image Segmentation
4. Transformation to Histogram
5. Paddy Disease Detection

Orillo et al., (2014) proposed a methodology for monitoring crop Brown Spot, Rice Blast, and Bacterial Leaf diseases using the neural network concept with MATLAB. The neural network is trained through a database of diseased images. In this research, fraction covered by the disease on the leaf, arithmetic mean values for the R, G, and B color components of the disease, the standard deviation of the R, G, and B color component of the disease, and mean value of the H, S, and, V of the disease have been extracted. The number of repetitions of the back-propagation algorithm (Buscema, 1998) is practiced until satisfactory results are obtained.

Mangla et al., (2019) proposed Otsus' method (Chen Yu et al., 2010) and vegetation segmentation for identifying the threshold values in images and use textual analysis to feature extraction and also Support Vector Machine (SVM) algorithm (Noble, 2006) was proposed to classify the images considering the extracted features. Focused primarily on leaf color, has been implemented background noise independent algorithm to isolate disease spots using image processing methods. 'A' component of CIELAB color mode (Ly et al., 2020) has been used for segmentation of disease spots and has been comparing the effects of CIELAB (Ly et al., 2020), HSI (Hue, Saturation, Intensity), and YCbCr color space (Chai and Bouzerdoum, 2000) on the disease identification process. In this algorithm, median filter (Perreault and Hebert, 2007) has been used for image smoothing and the Otsu method (Chen Yu et al., 2010) has been applied to the color component to detect the disease spots on the leaf. The algorithm has been tested on various “Monocot” and “Dicot” family plant leaves with noisy and noiseless backgrounds (Chaudhary et al., 2012).

A good platform has been proposed for Crop Health Advisory Boards to atomize crop health problems and solutions, and present simple and robust image processing methods with a comparison of crop leaf color with leaf color chart (LCC) and mathematical modeling (Singh and Singh, 2015)

2.1.4 Image Pre-Processing

Image pre-processing has been used to remove unwanted noises from collected images. Several ideas of image pre-processing techniques have been discussed.

Image preprocessing can significantly improve the reliability of optical inspection. Several filtering operations that enhance or reduce specific image details can be evaluated more easily or quickly. Users can optimize camera images with just a few clicks. It involves cropping, rotation, normalization, contrast enhancement, filtering, angle correction, and various graphics operations (Mukherjee et al., 2010). Image preprocessing was used to remove noises like dust, dewdrops, insect excrements from digital images, and different types of notice removal filters have been used to remove noises and distortion of some water drops and shadow effects.

Bakar et al., (2018) have used mainly three image pre-processing techniques to transform the original image into a new color space, which is fundamentally like the original image but differs in certain aspects. Images resize, image restoration, and image enhancement have been mentioned as follows: involvement steps in this process.

- Resize

Considering the memory capacity and the computational complexity, original images resized to fixed resolution with the dimension 640 * 480 pixels.

- Noise Restoration

Noise can be caused by motion between camera and object, improper shutter opening, atmosphere disturbance, and miss focusing.

- Image Enhancement

Mukherjee et al., (2010) used Image enhancement for adjusting digital images so that the result was more suitable for display or further analysis. Image enhancement was used to adjust digital images so that the result was more suitable for display or further analysis (Mangla et al., 2019). Ramesh and Vydeki, (2020) have proposed converting RGB images into HSV (Hue, Saturation, Value) because the HSV represents the Hue, saturation, and value part of the images. Considering the ability to reduce the background effect, the images were converted to HSI color space, and the histogram analysis of the intensity channel has been used to obtain an opening that could increase the image contrast (Kamlapurkar, 2016).

RGB images on leaves are converted to HSI color space representations to facilitate color specification in a generally accepted way. After that, green color pixels were identified through specific threshold values and were removed entirely from the image (Arivazhagan et al., 2013).

2.1.5 Image Segmentation

Image segmentation has been used to divide the images into segments for analysis. Images have been converted into another format base on appropriate features.The segmentation of image states represents an image into another meaningful format which is easier to analyze (Mukherjee et al., 2010). Images have been divided into particular regions or objects to analyze and extract the data's useful features (Shah et al., 2016).

Mangla et al., (2019) used Otsu's method (Chen Yu et al., 2010) and adjusted the threshold to mean pixel intensity to find the threshold values. Blob analysis Moeslund, (2012) has been used to shape feature extraction to get the number of objects for labeled regions in a noise-free binary image. It is then combined with the color characteristics extracted from the digitized image and classified using the Support Vector Machine (Noble, 2006).

Image segmentation means partitioning an image into several parts that have the same similarities or same features. Segmentation has been done using various methods like k-means clustering (Vora, 2013), converting RGB image into HIS model, Otsu's method (Chen Yu et al., 2010), etc. (Kamlapurkar, 2016).

K-Means clustering (Vora, 2013) has been used for image segmentation. Moreover, the trial and error method (Prather, 1971) was used to select the K value in the K-means algorithm (Ramesh and Vydeki, 2020). According to Bakar et al., (2018), the image segmentation process consists of background subtraction, feature extraction, and image analysis. Image segmentation is performed by selecting the appropriate threshold range to separate the infected area from the background. The threshold values are chosen from the bottom and top of the image histogram. Considering the non-uniform distribution of the disease zone, Bakar et al., (2018) have identified that one process at the threshold is ineffective. Therefore, a multi-level entry based on pixel-based segmentation was proposed.

Based on a Fuzzy logical grading system in the query image, the disease section and the healthy part are divided (Jhuria et al., 2013). According to Chaudhary et al., (2012), if there is a sharp and deep valley between two peaks in the histogram, the bottom of the valley has been chosen as the threshold. Otherwise, this tactic cannot be used to separate objects from the background. Therefore, the Otsu method (Chen Yu et al., 2010) has automatically selected the most appropriate threshold value.

2.1.6 Feature Extraction

To distinguish between paddy leaf diseases, selected features are extracted from the images. Components are selected based on the image database. The colors and shape of paddy leaves are identified as the main features that can separate one disease from others.

According to Mangla et al., (2019), shape feature extraction and color base segmentation should focus on this stage. Inherent characteristics or features of objects present within the image have been focused on this stage Shah et al., (2016). Color, shape, and texture were the main features extracted from fragmented objects (Mangla et al., 2019). The color element plays a crucial role in classification because it can distinguish one image from another (Mangla et al., 2019; Shah et al., 2016). Color and Shape extraction are defined as follows.

- Color Extraction: extract the color changes from the digitalized image.
- Shape Extraction: identifying the shape of a given rice leaf by taking into account the color difference of the digitized image.
-

An automatic shape recognition algorithm was developed to detect and identify different shapes of any color and non-color images. The algorithm deals with grayscale images, and the contrast-limited adaptive histogram E is applied to increase the contrast, and the enhanced image is converted to a binary image (Rusiñol et al., 2007). According to Narvekar et al., (2007), color image features in the visible light spectrum provide additional image characteristics than traditional grayscale representation. Therefore, the color co-occurrence method (Vadivel et al., 2007), representing both color and texture features as a unique feature, has been used to obtain features from the grape leaves image.The edge is a key expression of local features and image integrity. Edges define the boundary between object and their background. Edge can be mathematically represented by first and second-order differential equations (Shanmugavadivu and Kumar, 2013).

Kamlapurkar, (2016) used Gabor filtration (Nazarkevych et al., 2017) to extract the color and morphological features of the leaf's spots. The mean and variance of the red, green, and blue channels at the location were identified as color features, and the shape of the leaf was assumed to be elliptical. Bakar et al., (2018) chose the Coni Edge detector to identify the boundaries between objects' areas and backgrounds compared to other algorithms in thin lines, continuous edges, and different lighting conditions. Before applying edge detection, Gaussian smooth filters (Basu, 2002) were used to reduce the appearance of high-frequency noise. To distinguish the regional boundary and the image contour, border tracking was used to extract the contours. Green’s theorem was used to calculate the infected area of the disease from the color image.

The gray level co-occurrence matrix (Vadivel et al., 2007) method is a two-dimensional matrix of joint probability between the pair of pixels separated by a distance in each direction. Contrast, homogeneity, correlation, the energy was calculated, and discrete wavelet transform (DWT) (Ghazali et al., 2007) was used to extract the features from the paddy leaf image (Deshmukh, 2015). Ramesh and Vydeki, (2020) calculated statistical features such as mean value, standard deviation, and GLCM (Gray-Level Co-Occurrence Matrix) (Vadivel et al., 2007) for healthy and infected leaves. Statistical features have been used to identify whether they have infected the leaves, taking into account the crop's characteristics and various symptoms.

2.1.7 Image Analysis and Diagnosis

According to Chaudhary et al., (2012), the threshold of RGB images cannot be used to accurately identify the spot of disease from brown infected rice leaves and the threshold on the ‘H’ component of the HSI color model and ‘Cr’ component of YCbCr color model (Basilio et al., 2014) are sometimes identified as disease spots but not in all. However, the threshold on the’ component of the CIELAB (Ly et al., 2020) color model allowed accurate detection of disease spots and is independent of background effects.

The histogram and saturation and intensity components of the red, green, and blue components are calculated in paddy leaf images. Still, the accuracy of the results depends solely on the Hue components. Therefore, decisions were made based exclusively on Hue histogram analysis (Singh and Singh, 2015).

The image is analyzed with the extracted features to determine whether the image has the above three diseases' symptoms. Color histogram and pixel arrangement in paddy leaf images have been used to separate paddy leaf images based on infected disease. According to Bakar et al. (2018), a border tracing algorithm can be used to extract contours from images. The image has been transformed into a histogram before analyzing the features of an image.

2.2 Presentation of Scientific Material

This section unfolde scientific methods and materials which were used to diagnose paddy leaves disease through image processing.

2.2.1 Data Augmentation

The amount of data is major needs in deep learning models. Fewer amounts of training data are a major restriction for developing accurate CNN models. Therefore, the number of images in our dataset is identified as an accessory needed to expand the functional diversity of the CNN base classification. Considering the dataset, image data augmentation has been used to enhance the number of images in the dataset with slight distortion. Using the data augmentation method, five images have been generated from one image in the full dataset with applying 0.2 rotation range, 0.1 width shift and height shift, brightness change in 0.2 to 1.0 range, 0.15 shear range, 0.1 zoom range, 10 channel shifting, enable horizontal flip, and nearest flip mode to each image. These RGB images in the dataset were converted into the gray scale to generate another dataset with grayscale images of bacterial leaf blight, brown spot, and leaf smut diseases using python Open CV libraries.

2.2.2 K – mean

The K-mean algorithm (Equation 2-1) is a partition-based cluster algorithm. It is also one of the ten classical data extraction algorithms. The idea derived from the K-mean algorithm is to cluster the K point in space and cluster objects close to them. The center values of the clusters are updated one by one until the best cluster results are obtained. The K-means algorithm is a general representative of the clustering method based on the prototype function. It takes the distance from the data point to the prototype as the objective function of optimization. The adaptive laws of repetitive function are derived from the method of finding the extreme values of functions. The K-means algorithm takes Euclidean distance as the similarity measure to find the optimal classification of an initial cluster center vector so that the evaluation index is minimum. The error type collection uses the criterion function as a cluster criterion function. Although the K algorithm is efficient, the K value must be given in advance, and it is very difficult to estimate the K value selection. In many cases, it is not known in advance how many sections a given database should be divided into (Zheng et al., 2018).

Equation 2-1: K-mean Algorithm

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

2.2.3 Median filter

The median filter is a non-linear filter commonly used to remove or reduce gray or salt and paper noise that can protect the border of an image during a random noise reduction process. (S. Archana and Sahayadhas, 2018). It is less than or equal to the mean value of the numerical sum of the half-values. The pixels are covered and shortened by the window (Figure 2-4). The central filter moves to activate (Jayanthi et al., n.d.).

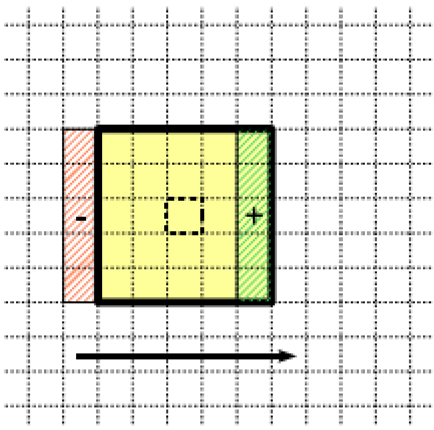

Figure 2-4: Median Filter sliding window

2.2.4 Color Spaces

HSV and RGB color spaces were mainly used in this research. HSV color space describes colors (color or shade) according to their shade (saturation or gray matter) and their brightness value. The HSV color wheel is represented as a cone or cylinder (Figure 2-5). Color is expressed as a number from 0 to 360 degrees representing red (which start at 0), yellow (starting at 60), green (starting at 120), cyan (starting at 180), blue (starting at 240), and magenta (starting at 300). Saturation is the percentage of gray from zero to 100 percent of color. Value (or brightness) works in conjunction with saturation and describes the brightness or intensity of a color from zero percent to 100 percent (Figure 2-6). People use the HSV color space when choosing colors for paints or varnishes because the HSV represents better the way people relate to colors than the RGB color space. HSV color selection starts with choosing one of the existing colors, and most people associate color with setting the value of shade and brightness. Unlike RGB and CMYK 24, which refer to primary colors, HSV is defined in the same way that humans perceive color (Howard, 2017).

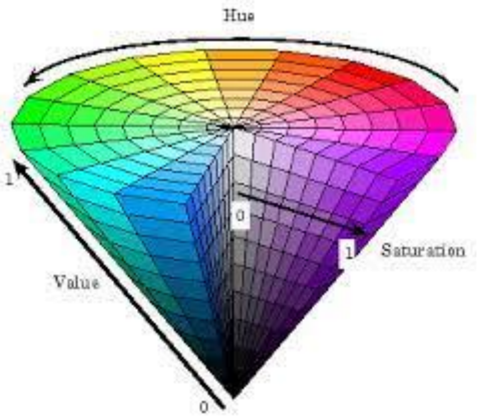

Figure 2-5: Illumination of HSV color space

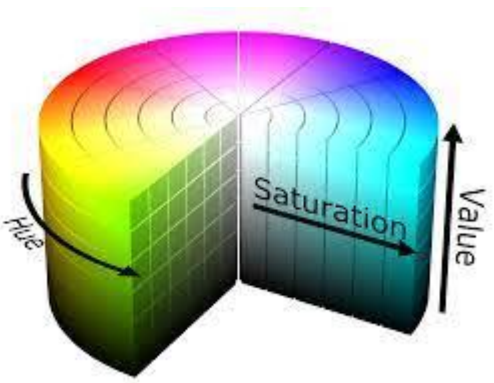

Figure 2-6: HSV color space

2.2.5 LightGBM

Gradient Boosting Decision Tree (GBDT) is a machine learning algorithm that is widely used for its efficiency, accuracy and interpretation. LightGBM is a GBDT algorithm that uses Gradient based one way sampling (GOSS) and Exclusive Feature Bundling (EFB) technology to deal with a large number of data cases and a large number of features, respectively. LightGBM was developed in April 2017 by a team from Microsoft to reduce implementation time. LightGBM can significantly outperform XGBoost and SGB in terms of computational speed and memory with the help of GOSS and EFB (Ke et al., n.d.). The main difference is that decision trees in LightGBM are grown leaf-wise (Figure 2-8), instead of checking all of the previous leaves for each new leaf, and All the attributes are sorted and grouped as bins (Figure 2-7). This implementation is called histogram implementation. LightGBM has several advantages such as better accuracy, faster training speed, and is capable of large-scale handling data, and is GPU learning supported.

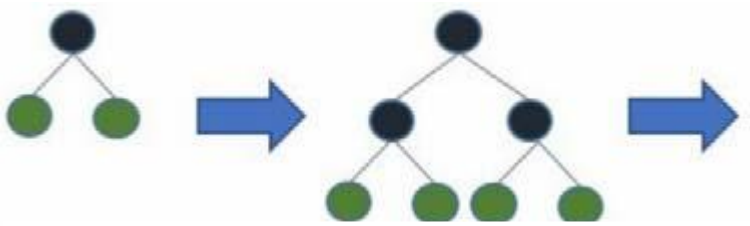

Figure 2-7 : Level wise tree growth XGBoost

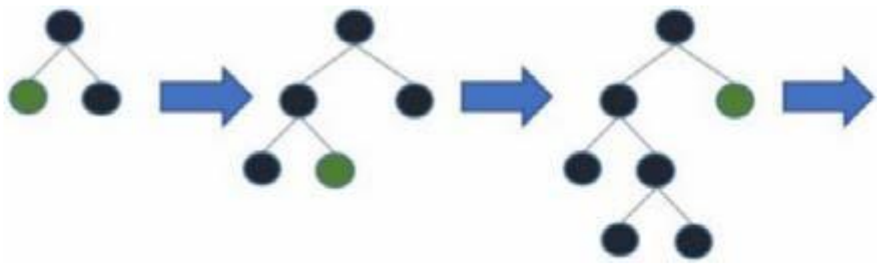

Figure 2-8: Leaf wise tree growth LightGBM

2.2.6 Convolutional Neural Network (CNN)

CNN is a deep learning model with mainly three types of layers na mely input layer, an output layer, and hidden layersIn image classification, the input is an image and the output is a label or class, and the hidden layers can include convolution layers, pooling layers, normalization layers, and dropout layers. Image can be given as input and the size and number of color channels can be denoted. Considering the dataset, data augmentation can be done before passing the images to the CNN model. The main function of the convolutional layer is convolution operation using kernels (filters) to produce an output feature map (Arivazhagan and Ligi, n.d.).

Convolution output can be denoted as Equation 2-2,

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right)$$

Equation 2-2: Convolution output Equation

Where x_j represents the output feature map set, M_j represents the set of input maps, K_{ij} represents the kernel for convolution, b_j represents the bias term.

Max-Pooling Layer

The input is divided into several non-overlapping sections, which maximize the output between the elements of each block and reduce the size while securing important input information. The max-pooling layer is also able to minimize the problem of overfitting. There is no learning process in this layer (Arivazhagan and Ligi, n.d.).

Dropout Layer

The basic idea of the dropout layer is to deactivate or remove a certain probable input element so that each neuron can learn the features that are less dependent on its environment. This process takes place only during the training phase (Arivazhagan and Ligi, n.d.).

Batch Normalization Layer

This layer has the ability to increases the training speed and reduces the sensitivity of neural network initialization. The activation of each channel is normalized by subtracting the mini-batch mean and dividing it by the mini-batch standard deviation (Arivazhagan and Ligi, n.d.).

Output Layer

The output layer consists of the softmax (Equation 2-3) layer following by the classification layer. The softmax layer outputs a probability distribution based on which, the network model classifies an instance as a class that has the maximum probability value (Arivazhagan and Ligi, n.d.).

$$P(c_r \,|x, \theta) = \frac{P(x, \theta|c_r)\, P(c_r)}{\sum_{j=1}^{k} P(x, \theta|c_j)\, P(c_j)}$$

Equation 2-3: Softmax function

Where, $P(c_r \,|x, \theta) \leq 1$ and $\sum_{j=1}^{k} P\big(c_j \,\big|x, \theta\big) = 1$ and $P(x, \theta|c_r)$ is the conditional probability of an instance given class r and is the class prior probability.

CHAPTER 3
METHODOLOGY

This chapter express the steps of the model which were developed to diagnose paddy leaves diseases. When we developed a system to diagnose paddy leaves diseases, identification of paddy leaf can be named as the most primary task to achieve before diagnois the infected disese base on image contained features. After leaf pixels are identified from the image, background pixels are eliminated and green color pixels that rely inside paddy leaf pixel boundary, are eliminated to isolate discoloration areas, that can be identified as disease symptoms and those discolored pixel characteristics are used to diagnose paddy leaf diseases. Bacterial Leaf Blight, Leaf Smut, and Brown Spot are the diseases that we are going to diagnosis using this system, and color of discoloration pixels are used to check whether there are similarities with symptoms to clarify leaf infected with an above-described disease (Figure 3-1).

There are two models developed using the LightGBM tree base machine learning algorithm and Convolutional Neural Network (CNN) model. CNN model was trained using image inuts and inputs for the LightGBM algorithm were thirteen attributes extracted from discoloration pixels using python OpenCV libraries. The algorithm was developed to isolate disease-infected areas by accessing given image pixels' colors and several features have been identified with referring to the pixels which were remained after removing the background and green pixels. Those features were used as attributes to the LightGBM model.

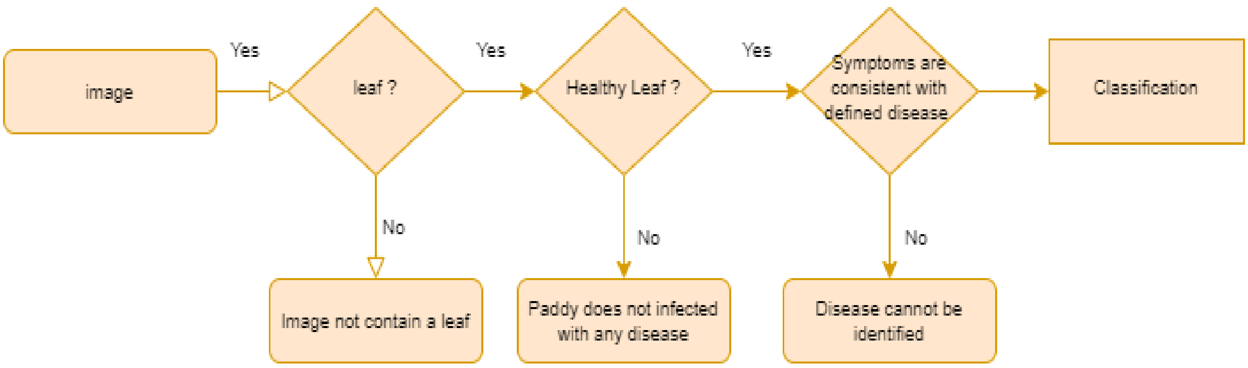

Figure 3-1 : Flow of the model

3.1 Leaf Validation.

Classification algorithms consider the similarities of the features of the given image with defined classes to check their features. In our case, when we insert any image which does not contain a paddy leaf will also be classified as bacterial leaf blight, brown spot, or leaf smut after being compared with the most matching features of these classes. Therefore, it is important to check whether there are paddy leaves contained in the image which are uploaded to the website. It is a hard task to check whether there is a paddy leaf in the image, therefore, a simple mathematical method was used to verify whether there are enough green color contained in the image to have a paddy leaf and how close they are located in the image. Image background should contain non-green color background to use this method.

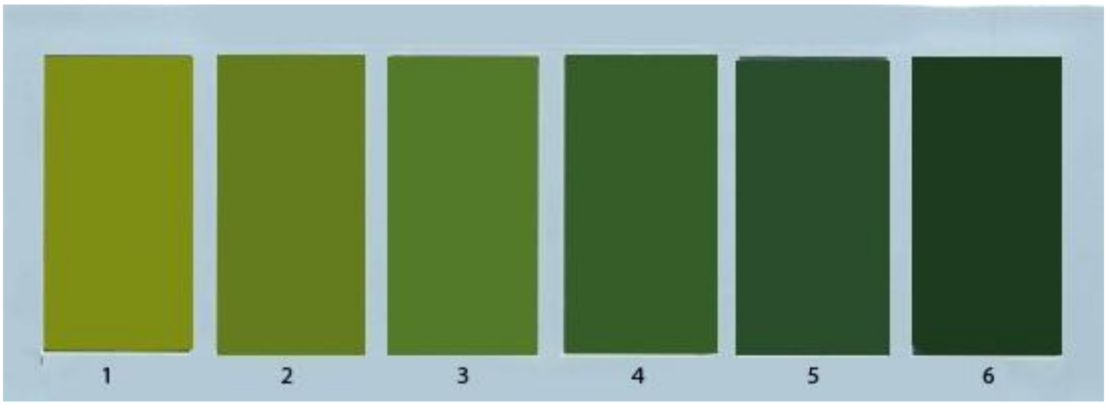

Figure 3-2: Leaf Color Chart

The Python OpenCV library has been used for the image validation part of this project. As a very first step, all images should be the same size. Therefore the uploaded image is resized into 150 width and 150 heights to contain a fixed number of pixels in the image. Moreover, the background of the image is eliminated to isolate leaf pixels. Green color pixels have been used to achieve this task. RGB color space wasn't supported for color detection based on the RGB values. HSV (Hue, Value, Saturate) color space was identified as the most suitable color space to convert numeric values to known colors. OpenCV reads the image as a BGR color space, hence after resizing the uploaded image, the image is converted into other three-color spaces respectively RGB, RGBA and HSV. In RGBA, A for the alpha channel and this parameter can be used to eliminate pixels from the image. HSV color space image is used to determine whether pixel color is in the green color range (Figure 3-2). Lower HSV values for green color were identified as (25,52,72) and higher values were identified as (126,255,255). That pixel in this range is identified from HSV image and then apply bitwise operation with RGB image to remove green color pixels from the image and highest and lowest green color pixels are identified in each column and pixels which are in outside of lower and upper green color pixel in each column are identified as background pixels and then those pixels are eliminated from the image (Figure 3-3).

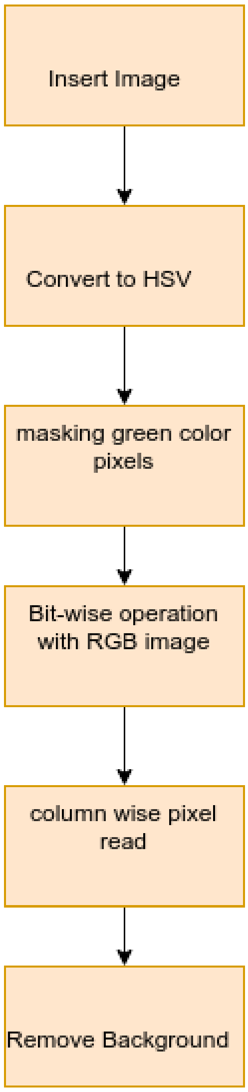

Figure 3-3 : Steps of background pixel elimination

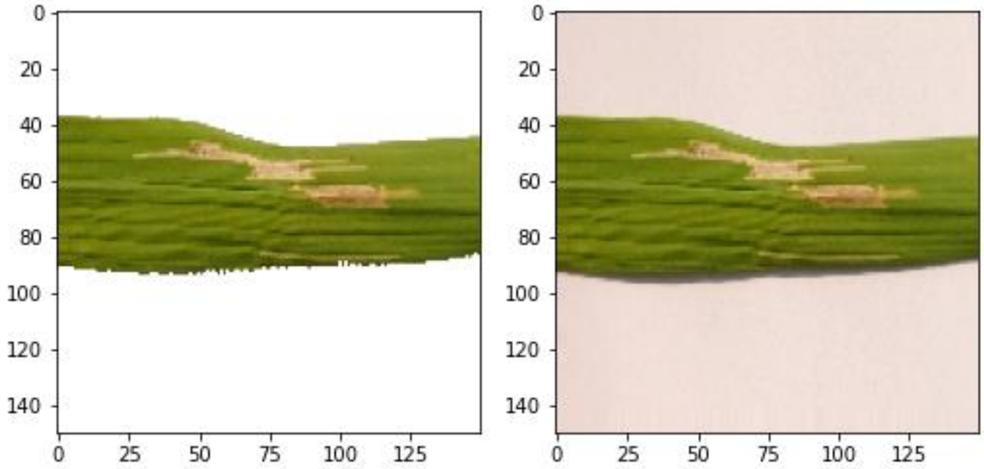

Figure 3-4: Paddy leaf image after and before removing the background

After eliminating background pixels from the image (Figure 3-4), the number of green color pixels is counted, and calculate standard divination of green color pixels in the x and y-axis separately. With considering the number of green color pixels in the image and their standard divination, the system defines whether the image consists of enough pixels using the number of pixels and how nearly all pixels are located to each other using standard deviation (Table 3-1).

Table 3-1 : Leaf validation conditions

Number of pixels	*Standard Deviation*
less than 5000	less than 20
less than 8000 and greater than or equal 5000	less than 30
Greater than 8000	less than 50

3.2 Disease Spots Isolation

Leaf border pixels and green color pixels in the leaf have been identified in the leaf validation part. After validating an image containing a paddy leaf is done, discoloration areas should be identified to detect if there are any disconsolation areas to have a disease in the paddy leaf. Green color pixels are eliminated from the image as the same as background pixels. After removing green color pixels and background pixels, images only contain discoloration areas that can be used for paddy disease diagnosis. the remaining number of pixels with their colors can be used to identify whether the leaf is healthy. According to our naked eye and open CV-based calculation observations, leaf discoloration areas should have at least one of brown, yellow, black, and orange color pixels. These characteristics are used to check whether the paddy leaf is infected with one of the defined diseases. After removing background pixels from the image, leaf border pixels (Figure 3-6) have been extracted before applying the K-mean algorithm (Figure 3-5). Python OpenCV library is used to predict the remaining pixel color from the image. Image pixel colors can have various types of noises in infected area’s pixels and some unmatched color pixels. Therefore, K-mean algorithms with k = 5 are applied in RGB color space to segment the color of the pixel concerning the nearest pixel's colors (Figure 3-7). After applying the K-mean algorithm to image pixels, individual pixel colors are identified based on HSV color space hue values because of RGB color space did not provide good performance with pixel color detection. Therefore, images convert into HSV color space and predict pixel color using lower and higher value ranges of each color. The number of brown, yellow, orange, red, and black color pixels is calculated from the remaining pixels and check whether there are enough pixels with these colors to have a disease that we described (Figure 3-8). If the remaining pixels don’t have enough pixels or above color, we can identify that paddy leaf infected with some-else paddy diseases that cannot be identified by our model. Otherwise, uploaded images into the system, classified as one of bacterial leaf blight, Leaf Smut, or Brown Spot infected leaf image using Constitutional Neural Network (CNN) that have been developed and tested using our dataset.

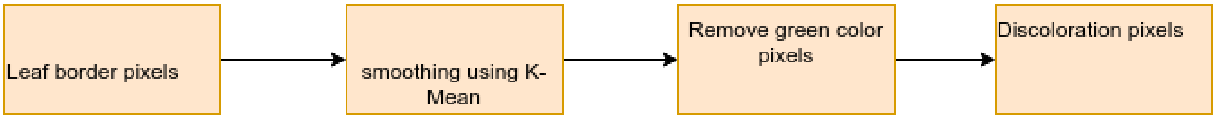

Figure 3-5: Discoloration areas detection

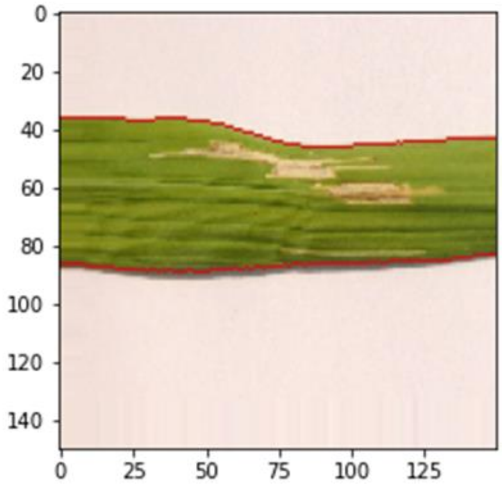

Figure 3-6: Paddy leaf border selection

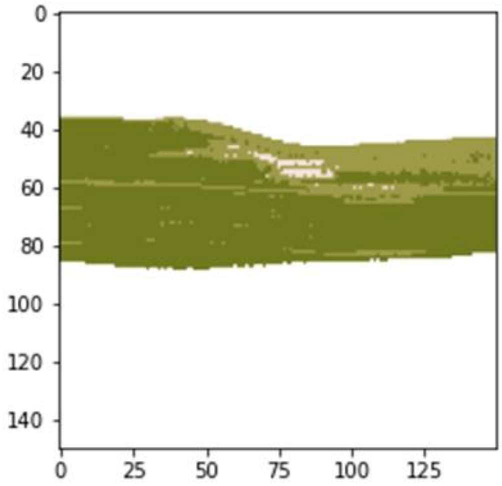

Figure 3-7: Paddy leaf after applying K-mean

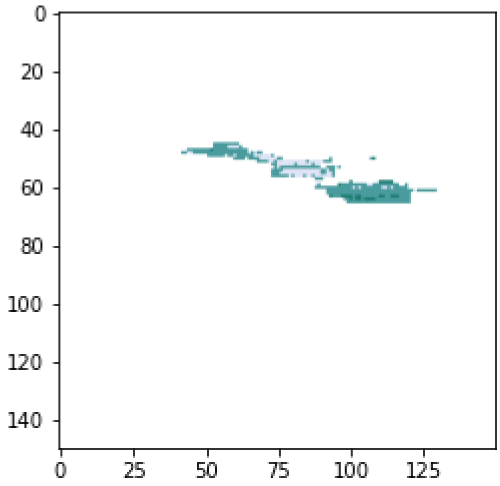

Figure 3-8: Infected areas pixels

3.3 Model 01 - Convolutional Neural Network

Datasets have been manually augmented and those datasets were used to develop a Convolutional Neural Network with three hidden layers. Input images were resized into 150 x 150 size and 3 x 3 max-pooling layers and Batch Normalization layers are used to feature extraction purposes.. After identifying a leaf as an infected leaf, Convolutional Neural Network (CNN) model was trained to classify bacterial leaf blight, leaf smut, and brown spot diseases using three convolutional 2D layers. Dataset (Table 3-2) that has been enhanced using image augmentation techniques ere used to train a CNN model (Table 3-3).

Table 3-2 : Dataset for CNN model

Disease	*Number of train images*	*Number of test images*
Bacterial Leaf Blight	100	50
Brown Spot	100	50
Leaf Smut	100	50

Table 3-3 : CNN model summary

Layers	*Function*	*Filter/Pool*	*No. Filters*	*Output*	*No. parameters*
Input	-	-	-	150 x 150	
Convo 1	Convolution	3x3	16	16x148x148	448
Batch Normalization	Batch_Normalization	-	-	16x148x148	64
Pooling1	Max pooling	3x3	-	16x49x49	0
Convo2	Convolution	3x3	32	32x47x47	4640
Batch Normalization	Batch_Normalization	-	-	32x47x47	128
Pooling2	Max pooling	3x3	-	32x15x15	0
Convo3	Convolution	3x3	64	64x13x13	18496
Batch Normalization	Batch_Normalization	-	-	64x13x13	256
Pooling3	Max pooling	3x3	-	64x4x4	0
Flattern	Flattern			1024	0
Dense1				128	131200
Dense1				3	387
Output	Softmax				

3.4 Model 02 - Classification using LightGBM

With the exception of the CNN model, machine learning based classification model was developed using image's features that were extracted using Python OpenCV. The median filter was used to reduce noise in paddy leaf images and after applying the median filter, images'width and height were resized into 150,150-pixel size. Considering the number of green color pixels of the image, background and paddy leaf boundaries define and remove background pixels from the image. After removing green color pixels and background pixels from the image, there are only discoloration pixels (Figure 3-9) that can be identified as infected areas. Considering the row-wise pixel lengths of remain pixels, large patch was identified (Figure 3-10). Moreover, thirteen features have been extracted from paddy leaf images shown in Table 3-4:

Table 3-4 : Attributes list for LightGBM

#	*Attribute*	*Non – Null Count*	*Data Type*
1	Shape	294 non-null	int64
2	Size	294 non-null	int64
3	Width	294 non-null	int64
4	Height	294 non-null	int64
5	No. of Orange Pixels	294 non-null	int64
6	No. of Brown Pixels	294 non-null	int64
7	No. of Black Pixels	294 non-null	int64
8	No. of Yellow Pixels	294 non-null	int64
9	No. of Gray Pixels	294 non-null	int64
10	No. of Red Pixels	294 non-null	int64
11	No. of Other Pixels	294 non-null	int64
12	Distance	294 non-null	int64
13	Line Count	294 non-null	int64

The colors of remain pixels were calculated using HSV color space and some brown, yellow, black, red, orange, gray, and other color pixels were calculated for each image. Image pixels have been accessed through X-axis and the first and last pixel of each line were identified with considering the neighbor pixel colors. Considering each continuous pixel in lines, the longest continuous pixel line was identified as the width that belongs to the biggest patch in the leaf. Moreover, a perpendicular line that goes through the center of the longest width has been used to measure the height of the patch. The magnitude of width and height are used to categorize patch size into five categories.

Shapes of the identification of the infected areas are a hard task to achieve because there were no exact shapes in the images. Therefore, width and height with their ratio have been used to define the shape of the biggest patch.

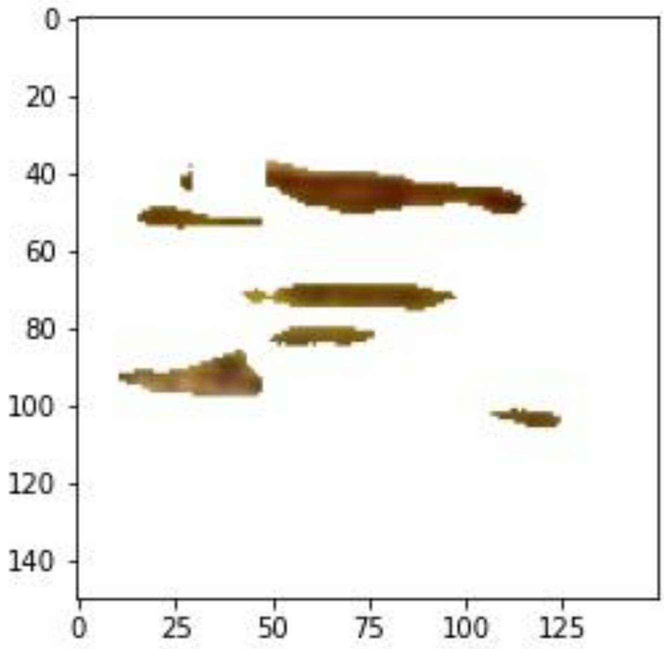

Figure 3-9 : Discoloration pixel extraction

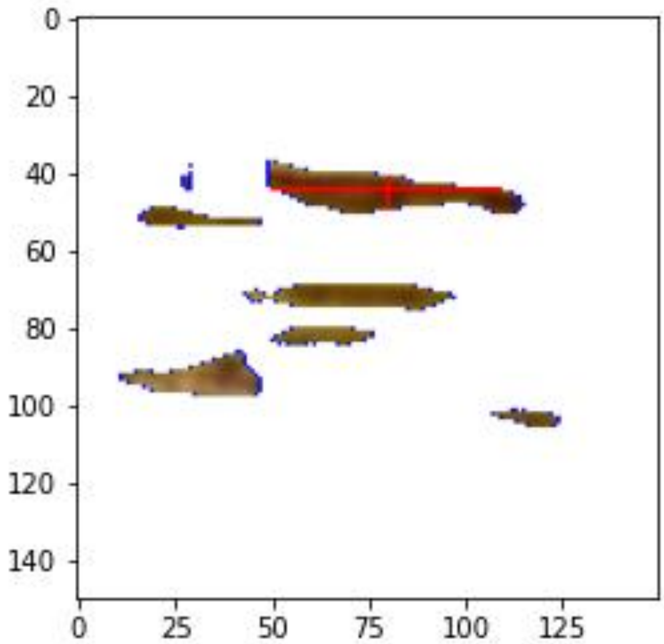

Figure 3-10: Feature Extraction from the image

CHAPTER 4
EVALUATION AND RESULTS

In this research, we were hoping to find a way to diagnose rice leaves disease using naked eye observation attributes. As a first step, we have verified there is a leaf in the given image using the number of green color pixels with their Standard Deviations. Considering the inablity to detect pixel colors using RGB color space, images were converted to HSV color space for color prediction. Discoloration areas of the image can be isolated after removing background and green color pixels from the image (Figure 3-9).

Thirteen attributes were computed from paddy leaf images related to three common paddy leave disease. Set of computed attributes were able to provide an accuracy of about sixty precent from Naïve Bayes and Support Vector Machine (SVM) classifications, but the LightGBM decision three algorithm provided an eighty one precent of classification accuracy.

4.1 LightGBM Model Results

The first model that was developed using the LightGBM algorithm has shown about 81% accuracy with the given dataset. 70% of the dataset was used to train the model and remain 30% was used as the test dataset. LightBGM model precision and recall values are shown in Table 4-1 for three iterations. Considering the attributes which were extracted from the dataset, the most important attributes against their importance levels are shown in Figure 4-1.

Table 4-1 : Confusion Matrix of LightGBM Model

#	*Precision*	*Recall*	*F1-score*	*Support*
0	0.85	0.92	0.88	25
1	0.77	0.71	0.74	28
2	0.81	0.81	0.81	36

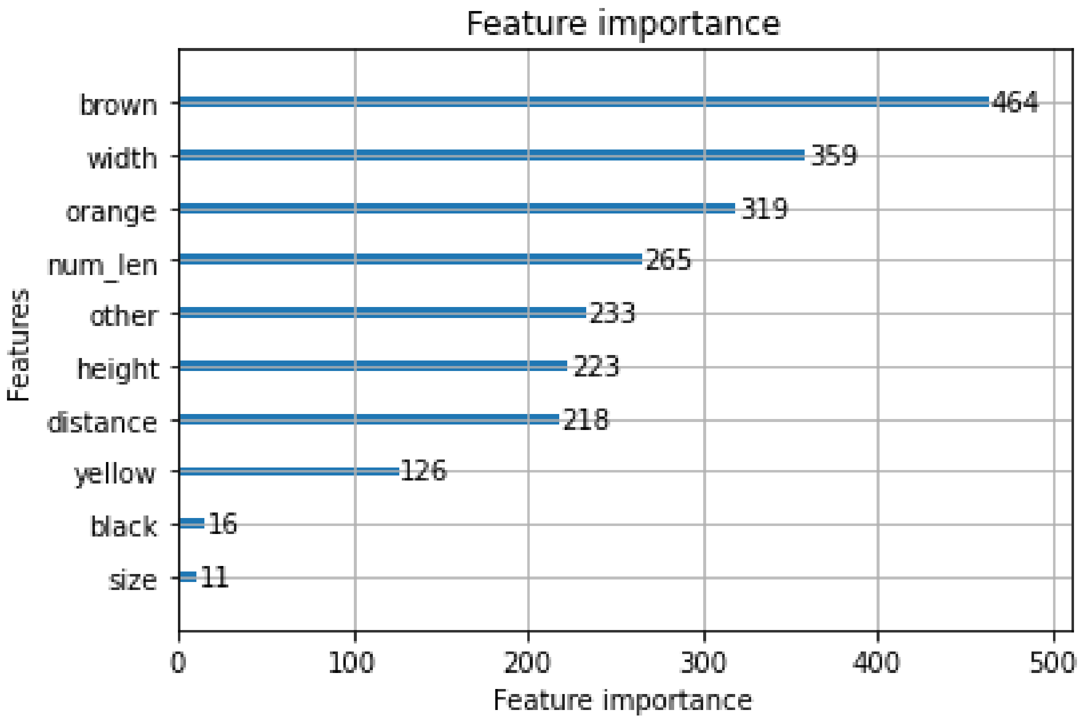

Figure 4-1: Feature Importance

Average impact of each class in the LightBGM model is represented Figure 4-2. According to the this diagram, number of red and gray color pixels do not provide enough support to the classification model. Attribhutes ‘width’ and ‘brown’ have provided better average impact to the model. Backterial leaf blight, brown spot, and leaf smut are reprecented by class0,class1,and class2 respectively.

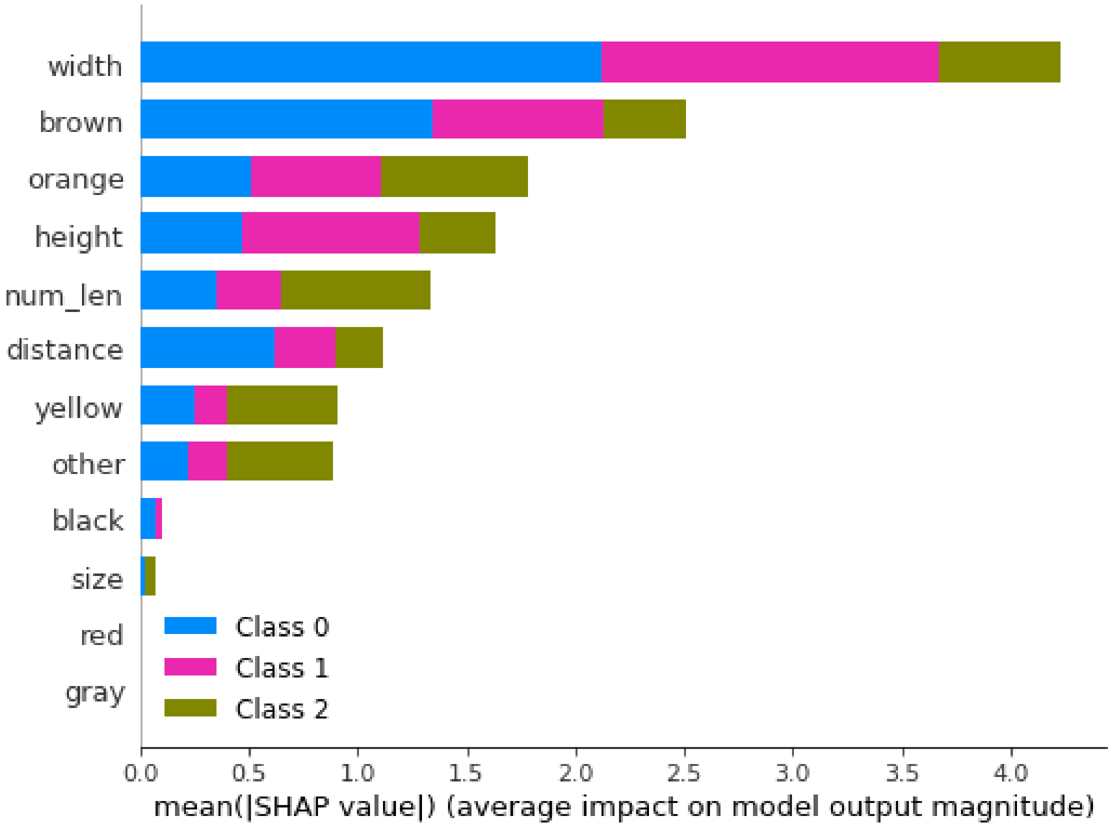

Figure 4-2: Average impact on model output magnitude

4.2 CNN Model Results

The experiments were conducted on an Intel® Core(TM) i5 – 7300U CPU2.60GHz with 8 GB RAM and HD Graphics 620 GPU memory. The model is implemented in Python with packages Keras and TensorFlow under OS Ubuntu 16.04.

The image was validated as a leaf contain image using the number of green color pixels with their standard deviation and image processing techniques with K-mean algorithm were applied to extract features from the image and a CNN model was used for diagnosis infected disease. According to the number of Epochs, the accuracy and loss of the trained model are as for Figure 4-3.

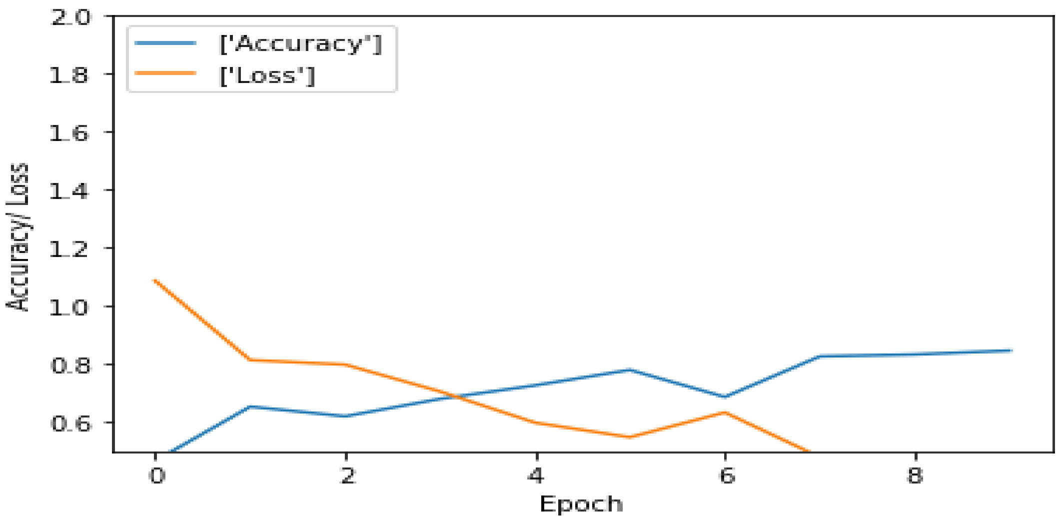

Figure 4-3: CNN model accuracy and loss

Trained CNN models have been exported as a .json file and results of bacterial leaf blight (Figure 4-4), brown spot (Figure 4-5), and leaf smut (Figure 4-6) have shown below with leaf validation result (Figure 4-7).

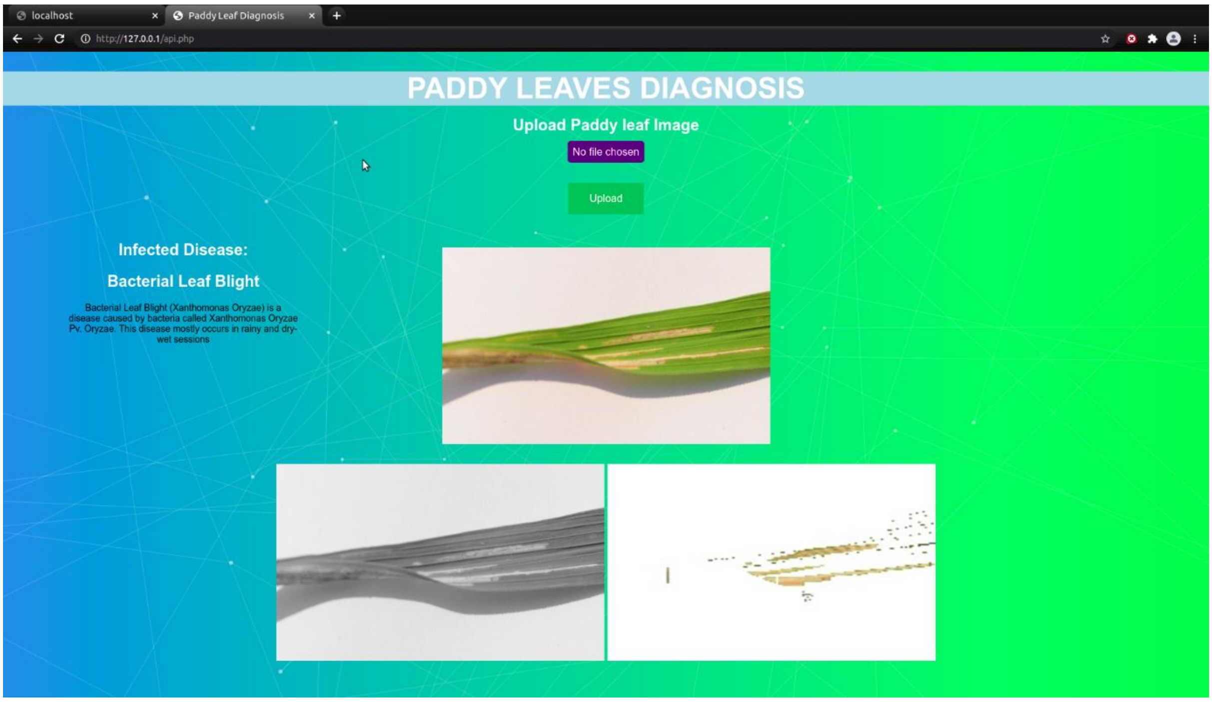

Figure 4-4 : Diagnose Bacterial leaf

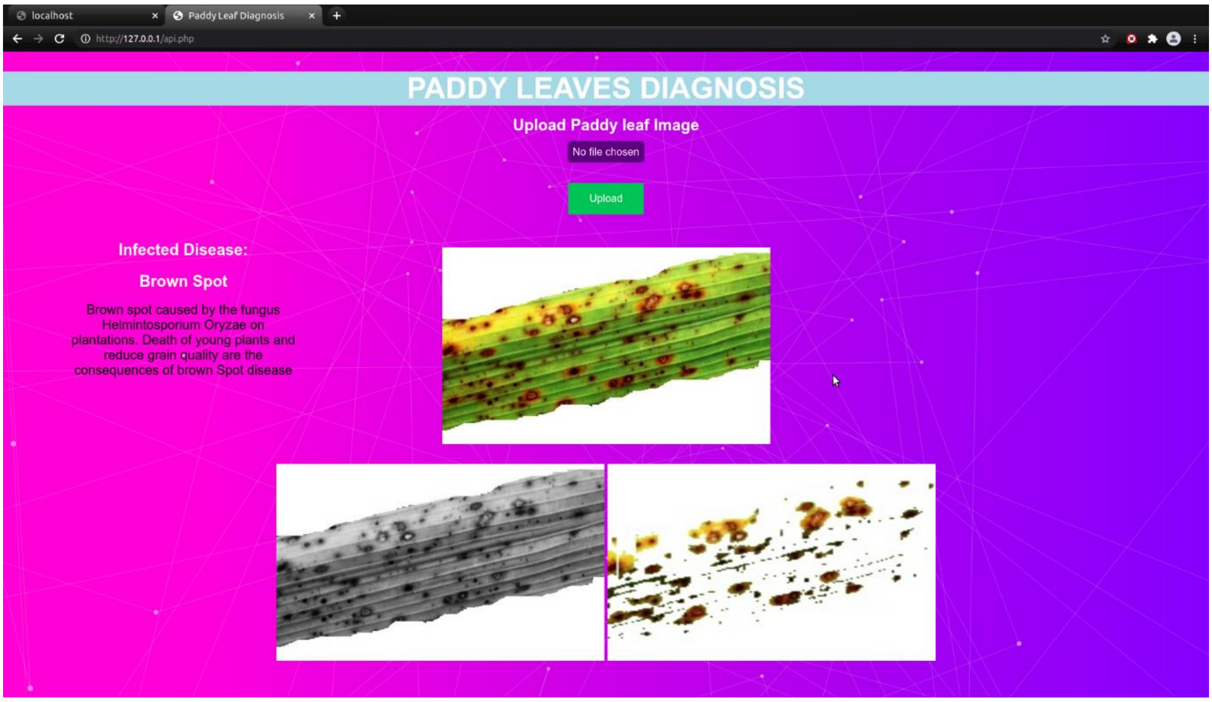

Figure 4-5: Diagnose Brown Spot

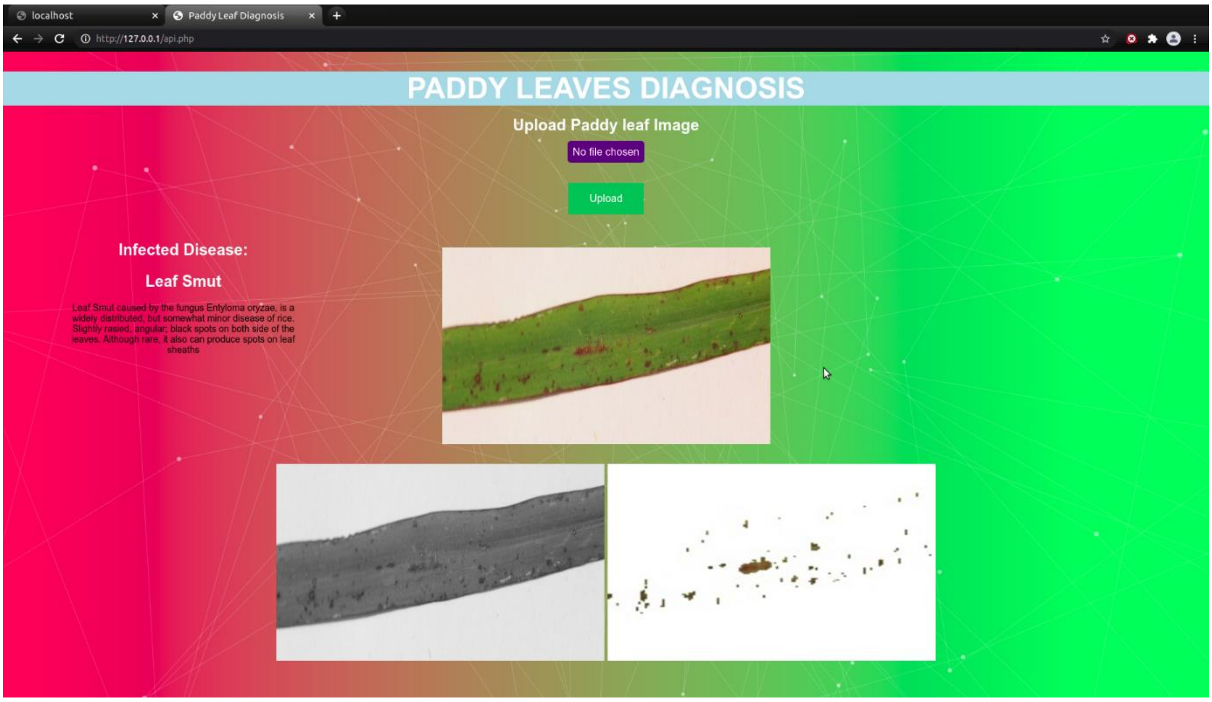

Figure 4-6: Diagnose Leaf Smut

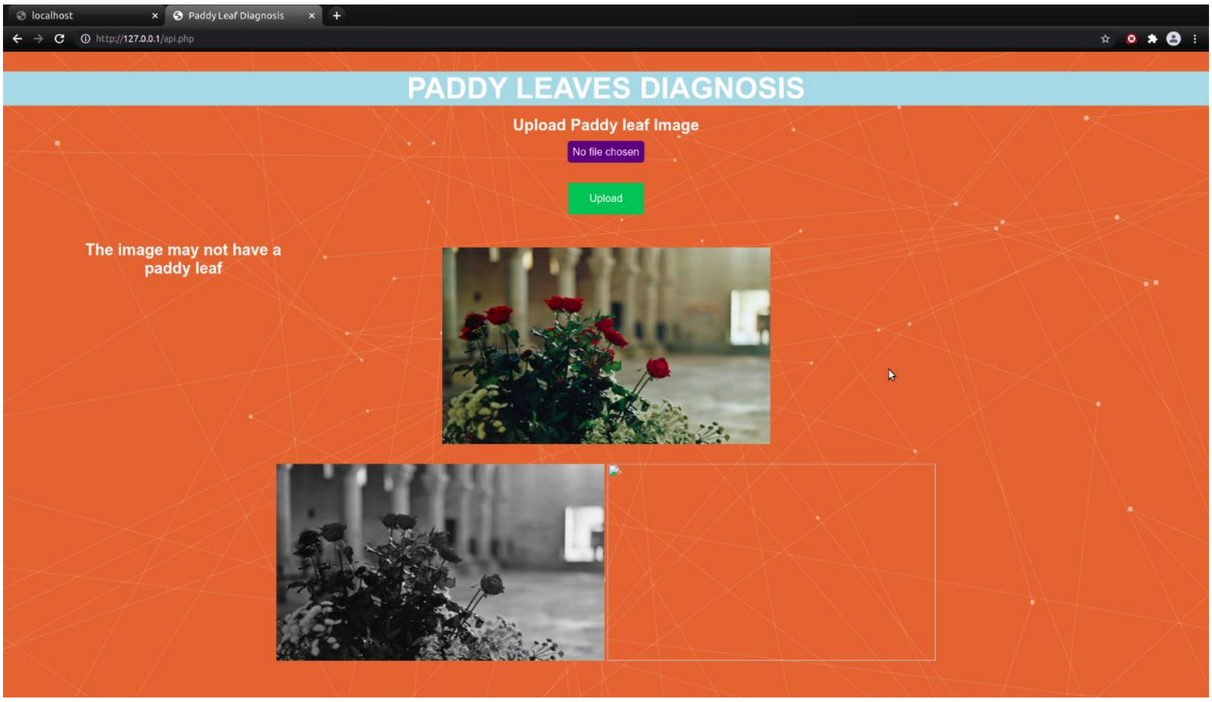

Figure 4-7: Paddy leaf Validation

4.3 Discussion

CNN models can extract features from images across the Maxpooling layer, but we cannot see what features are used for taxonomic images. Given the complexity of the CNN layers, we cannot control the selection of features in the CNN model. Therefore, any image is identified as one of the defined paddy diseases regardless of the symptoms. Therefore, the Light GBM Decision Tree Algorithm was used to classify the described paddy leaf diseases by considering the symptoms related to the described diseases. Infected areas were properly isolated using pixel color values, and the properties for the lightGBM mode were extracted using the color of the pixels along with their locations. This mode provides better control over the selection of features, and a better feature extraction algorithm increases the accuracy of the model.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The main reason for the loss of yield in the agriculture field is the extensive spread of diseases. Identification and detection of the diseases are noticed mostly when the disease has escalated to a severe stage. This results in the loss of time, money and causes major losses in yields. In this project, a web application has been developed using an image processing algorithm with the CNN Hybridge model to provide a classification of the paddy leaf. CNN model has provided 84% accuracy according to the dataset image set and validation of the paddy leaf has been done considering the number of pixels in the image and their standard deviation together to check whether there is a leaf in the image. Algorithms to validate the paddy leaf from the image will be developed using leaf border pixels distribution with relevant positions. Here we only extracted discoloration pixels' color from the image as a feature. Color feature alone couldn't be used to identify paddy disease, therefore discoloration areas' shapes, size, and color distribution over a single patch will be identified to provide a platform to diagnosis paddy leaf disease using their eye-catching features. Those mentioned attributes have been extracted from images that belong to Bacterial leaf blight, brown spot, and Leaf-smut, and those attributes with LightGBM decision tree algorithm were provided over 80% accuracy for paddy leaf diseases diagnosing model for the above-mentioned diseases.

In this research, two models were developed to diagnosis bacterial leaf blight, brown spot, and leaf smut paddy leaves diseases using CNN and LightGBM models. Accuracy of the LightGBM mode depend on extracted attributes and attributes are extracted through a defined algorithm. As a future works, we are hoping to find an accurate algorithm to extract shapes and color features from given images and model will be created to classify other main paddy leaved diseases using image processing. Image quality is a key factor that directly affects the diagnostic process. Leaf validation and feature extraction for low quality images do not work properly. Image background also plays a major role in feature extraction. Background with green pixels will mislead leaf validation and feature extraction. Therefore, we cannot monitor paddy diseases from the paddy field with this algorithm.

The LightGBM-based model has the potential to add more parameters to rice symptoms. The accuracy of the feature extraction algorithm largely depends on the diagnostic process. Therefore, the above algorithm can be improved to extract other eye-catching features from the image. The shape and size of each area along with their geometric location will be used to develop an algorithm to extract the most effective features from the diagnosis. Here we consider only three major infectious diseases of paddy. In the future, we hope to develop an expert system that will enable farmers to interact with domain excerpts through a machine learning algorithm for accurate predictions with a live camera to provide a better service for their fields to receive proper treatment.

Sri Lanka is an agricultural country with sufficient resources to feed every animal in Sri Lanka. We have a proud history in the field of agriculture. In order to develop Sri Lanka as a self-sufficient country, our knowledge must be integrated with human skills, Machine learning and computer vision technologies to provide a better platform for integrating domain expertise.

APPENDICES

• APPENDIX A : DATA SET

The dataset which was used in LightGBM model have been attached below. Those dataset have been extracted using an algorithm along with python OpenCV libraries.

id	*shape*	*size*	*width*	*height*	*orange*	*brown*	*black*	*yellow*	*gray*	*red*	*other*	*distance*	*num_len*	*class*
1	2	2	64	8	140	389	0	1	0	0	8	454	15	0
2	2	2	76	11	1896	162	0	0	0	0	0	700	45	0
3	3	4	34	6	44	21	0	2	0	0	232	221	19	0
4	3	3	48	11	414	49	0	0	0	0	1	391	19	0
5	3	3	42	10	394	70	0	0	0	0	3	413	19	0
6	4	4	25	8	557	65	0	55	0	0	3	580	39	0
7	3	4	27	5	505	69	0	1	0	0	1	530	35	0
8	3	3	54	10	706	2	0	0	0	0	0	598	31	0
9	3	4	25	6	107	146	0	1184	0	0	5	768	44	0
10	4	4	26	7	517	54	0	0	0	0	1	503	34	0
11	3	2	61	15	820	169	0	27	0	0	5	901	51	0
12	3	4	39	9	180	278	0	579	0	0	1	548	32	0
13	5	4	35	23	563	32	0	0	0	0	3	521	27	0
14	3	4	37	7	214	270	0	0	0	0	0	390	20	0
15	3	4	36	7	1098	167	0	0	6	0	36	1132	97	0
16	4	3	54	18	2334	3	0	13	0	0	2	2290	137	0
17	2	4	31	5	154	0	0	49	0	0	3	146	10	0
18	1	2	74	8	625	313	0	176	0	0	31	1010	66	0
19	1	3	60	7	599	103	0	0	0	0	1	624	25	0
20	3	4	29	5	181	262	0	1361	0	0	1	423	23	0
21	2	3	44	7	425	163	0	0	0	0	0	454	21	0
22	3	3	45	11	35	21	0	384	0	0	1	447	50	0
23	1	5	19	2	54	13	0	22	0	0	42	76	9	0

24	4	4	36	10	1048	222	0	120	0	0	0	1326	63	0
25	1	3	53	6	224	166	38	1183	0	0	248	1685	110	0
26	3	4	37	7	377	61	0	0	0	0	0	368	21	0
27	3	4	36	8	1106	147	0	1	0	0	0	1019	92	0
28	0	3	53	5	415	5	0	0	0	0	3	355	15	0
29	1	4	39	4	201	32	0	0	0	0	0	184	10	0
30	0	4	30	2	127	0	0	124	0	0	15	198	14	0
31	0	4	23	2	48	10	0	62	0	0	49	114	11	0
32	4	4	33	12	503	8	0	0	0	0	0	319	18	0
33	4	4	32	8	323	143	0	56	0	0	2	468	28	0
34	3	3	57	12	798	78	0	277	0	0	51	776	41	0
35	2	4	35	5	211	176	0	5	0	0	5	339	17	0
36	2	4	32	5	197	155	0	50	0	0	20	372	21	0
37	4	5	12	3	124	13	0	2	0	0	0	73	10	0
38	2	2	77	10	35	21	0	672	0	0	728	1164	81	0
39	4	2	65	17	1015	174	0	0	0	0	4	1064	58	0
40	1	3	42	5	174	77	0	24	0	0	0	213	12	0
41	1	3	53	6	375	13	0	0	0	0	0	322	15	0
42	1	2	70	7	596	206	0	3	0	0	13	715	36	0
43	0	4	29	2	189	136	0	54	0	0	1	395	33	0
44	1	2	63	7	825	6	0	0	0	0	0	606	25	0
45	3	2	64	12	868	54	0	0	0	0	3	826	39	0
46	0	3	53	5	413	36	0	0	0	0	0	376	17	0
47	1	4	39	4	376	138	0	0	0	0	0	446	31	0
48	1	3	42	5	172	70	0	0	0	0	0	190	9	0
49	1	4	28	3	294	144	0	0	0	0	1	326	19	0
50	4	4	37	14	630	114	0	0	0	0	0	692	32	0
51	1	4	40	4	1110	117	0	0	0	0	0	1162	54	0

52	2	1	94	12	2052	78	0	0	0	0	0	938	53	0
53	2	1	87	13	1571	262	0	0	0	0	0	1611	84	0
54	4	4	25	9	127	169	0	0	0	0	1	227	15	0
55	2	3	52	7	326	272	0	1	0	0	0	517	22	0
56	3	3	48	8	743	90	0	0	0	0	0	608	31	0
57	4	4	22	9	212	138	0	0	0	0	7	286	25	0
58	3	3	45	8	1160	180	0	0	0	0	3	1100	95	0
59	2	3	55	8	268	184	0	1084	0	0	18	1348	85	0
60	0	3	57	5	435	1	0	0	0	0	0	304	10	0
61	3	2	64	12	916	240	0	22	0	0	0	1059	60	0
62	3	1	92	20	43	21	0	3915	0	0	1027	4012	204	0
63	3	4	39	9	953	220	0	116	0	0	0	1230	55	0
64	4	4	26	8	216	121	0	0	0	0	0	276	21	0
65	2	4	38	6	272	173	0	14	0	0	5	409	21	0
66	0	1	95	7	599	288	0	1416	0	0	1	2008	96	0
67	3	4	33	8	250	4	0	0	0	0	1	202	12	0
68	4	4	31	9	259	164	4	741	0	0	2	1044	87	0
69	2	2	62	8	135	357	0	0	0	0	0	416	14	0
70	3	4	32	7	1000	202	0	43	0	0	8	988	93	0
71	3	5	17	3	144	0	0	42	0	0	1	135	13	0
72	3	3	53	10	612	8	0	21	0	0	0	501	24	0
73	5	5	18	13	103	167	0	761	0	0	4	561	47	0
74	5	4	24	15	667	188	0	873	0	0	5	619	46	0
75	0	4	34	2	117	0	0	115	0	0	14	190	16	0
76	4	3	44	15	737	200	0	332	0	0	3	1167	63	0
77	4	5	11	4	149	14	0	1	0	0	0	115	15	0
78	4	3	41	17	774	214	0	509	0	0	0	1386	75	0
79	3	3	50	10	620	8	0	10	0	0	0	511	26	0

80	3	5	17	4	96	15	0	0	0	0	3	70	7	0
81	4	4	31	8	296	20	0	0	0	0	146	178	10	0
82	1	3	52	6	272	187	0	1413	0	0	46	1588	94	0
83	2	3	42	6	141	59	0	909	0	0	4	677	46	0
84	1	3	59	6	609	4	0	0	0	0	2	524	25	0
85	1	3	41	5	433	127	0	0	0	0	0	408	29	0
86	4	5	11	5	147	9	0	1	0	0	5	96	12	0
87	1	0	144	15	1707	35	0	0	0	0	9	1547	43	0
88	0	4	32	3	293	179	0	503	0	0	2	848	78	0
89	0	4	33	3	470	24	0	0	0	0	0	436	24	0
90	2	4	28	4	492	108	0	0	0	0	2	542	35	0
91	0	0	129	9	689	376	0	16	0	0	466	1333	70	0
92	3	4	36	6	303	113	0	0	0	0	2	369	19	0
93	3	3	56	12	737	209	0	0	0	0	0	853	37	0
94	4	4	29	8	392	14	0	0	0	0	411	174	10	0
95	3	4	34	7	303	153	0	0	0	0	0	400	21	0
96	3	5	14	3	140	0	0	0	0	0	2	103	12	0
97	0	3	51	5	411	24	0	180	0	0	23	558	32	0
98	3	4	35	7	1074	218	0	36	0	0	4	1162	112	0
99	5	5	11	10	149	218	54	1	0	0	5	296	42	1
100	4	4	27	9	115	143	0	0	0	0	0	190	11	1
101	5	4	28	19	773	576	0	18	0	0	4	1280	87	1
102	5	5	14	9	50	76	0	0	0	0	0	94	8	1
103	5	5	10	10	61	54	0	0	0	0	1	81	10	1
104	5	5	9	7	38	46	0	0	0	0	4	32	4	1
105	5	4	23	21	526	376	0	4	0	0	13	781	59	1
106	5	5	8	6	51	31	0	0	0	0	4	37	5	1
107	2	4	23	3	211	209	0	0	0	0	4	313	31	1

108	5	5	9	6	20	73	0	0	0	0	0	51	7	1
109	5	5	9	7	28	64	0	0	0	0	0	41	6	1
110	5	5	14	12	150	180	0	0	0	0	4	273	38	1
111	5	3	42	26	928	661	0	3	3	3	4	1742	138	1
112	4	4	40	11	996	5	0	8	0	0	0	824	58	1
113	5	5	15	9	148	207	0	0	0	0	2	264	37	1
114	5	5	7	7	19	27	0	6	0	0	1	26	4	1
115	5	5	10	9	40	64	0	0	0	0	2	50	7	1
116	5	5	10	6	30	68	0	0	0	0	0	61	8	1
117	5	3	47	32	566	947	747	60	27	2	20	4192	207	1
118	4	5	19	8	23	95	0	0	0	0	33	112	8	1
119	5	5	13	9	49	75	0	3	0	0	0	92	8	1
120	5	5	10	7	28	64	0	0	0	0	1	50	8	1
121	5	4	35	34	641	533	0	0	0	0	0	1091	49	1
122	5	5	8	6	45	32	0	0	0	0	4	29	4	1
123	5	5	7	6	20	32	0	0	0	0	0	24	4	1
124	5	5	14	9	50	74	0	89	0	0	0	144	16	1
125	4	5	11	3	221	32	0	0	0	0	4	83	11	1
126	5	5	13	9	43	73	0	0	0	0	1	72	6	1
127	4	5	10	3	40	37	0	532	8	0	4	257	37	1
128	4	4	28	8	271	422	1	0	0	0	4	543	54	1
129	4	5	11	3	38	20	0	610	1	0	1	185	19	1
130	5	5	19	16	716	87	0	4	0	0	4	607	69	1
131	4	4	22	9	119	197	0	1	0	0	2	252	21	1
132	4	4	30	11	150	118	0	0	0	0	3	205	12	1
133	5	4	32	17	791	193	0	0	0	0	0	575	28	1
134	5	5	10	8	40	63	0	0	0	0	0	51	7	1
135	3	3	59	10	1957	1986	52	8	8	5	5	4625	293	1

136	5	5	17	11	749	114	0	0	0	0	4	731	79	1
137	4	4	40	13	1738	348	0	2	2	0	406	1731	157	1
138	5	3	49	29	518	964	1738	90	23	3	29	4210	260	1
139	4	4	22	9	109	173	0	0	0	0	5	223	19	1
140	5	5	10	9	47	58	0	0	0	0	1	73	9	1
141	5	5	15	12	593	172	0	93	0	0	5	680	84	1
142	2	3	47	7	855	430	0	4	0	0	1	1032	59	1
143	4	5	17	6	231	378	0	0	0	0	1	578	63	1
144	5	5	9	9	34	53	0	1	0	0	1	59	7	1
145	5	5	9	6	31	75	0	0	0	0	0	72	10	1
146	4	4	22	10	108	126	0	0	0	0	0	175	11	1
147	5	4	32	21	852	512	0	0	0	0	2	1226	82	1
148	5	5	10	9	154	194	37	5	0	0	4	274	39	1
149	5	5	17	11	91	187	0	80	0	0	1	221	21	1
150	5	4	28	18	1281	1219	36	30	0	0	339	2719	255	1
151	5	4	23	20	535	391	0	0	0	0	0	773	63	1
152	4	4	23	11	117	175	0	0	0	0	4	231	21	1
153	5	4	24	16	1033	1105	71	34	0	0	575	2625	234	1
154	5	4	26	14	732	759	1	2	0	0	5	1262	115	1
155	5	5	16	10	525	68	0	0	0	0	3	420	39	1
156	2	3	49	8	958	543	0	0	0	0	0	1214	70	1
157	4	4	24	10	116	139	0	0	0	0	0	191	12	1
158	4	4	24	9	263	416	0	0	0	0	63	517	52	1
159	3	5	11	2	46	45	0	275	0	0	19	98	13	1
160	5	5	10	7	36	56	0	0	0	0	1	39	6	1
161	5	5	10	9	39	57	0	0	0	0	0	61	7	1
162	4	3	44	11	873	889	3	0	0	0	235	1779	139	1
163	5	5	7	4	75	38	0	0	0	0	0	44	7	1

164	5	3	41	26	958	873	0	0	0	3	1	1823	136	1
165	4	4	23	8	496	556	10	8	0	0	5	1112	101	1
166	5	5	8	6	38	34	0	0	0	0	4	29	4	1
167	5	5	19	13	75	195	0	0	0	0	5	211	17	1
168	5	5	8	7	22	39	0	0	0	0	0	35	5	1
169	5	5	8	6	24	58	0	0	0	0	0	41	6	1
170	4	4	24	9	261	467	8	1	0	0	83	595	59	1
171	3	5	10	2	29	21	0	541	0	0	1	161	19	1
172	5	5	7	4	86	38	0	0	0	0	0	37	6	1
173	5	5	7	6	33	31	0	0	0	0	3	12	2	1
174	5	4	25	15	1009	1054	61	38	0	0	445	2415	225	1
175	5	3	47	29	580	1027	975	50	22	5	25	4468	262	1
176	5	5	9	8	43	52	0	6	7	0	15	70	9	1
177	5	5	7	5	48	63	0	2	0	0	9	25	4	1
178	5	5	15	11	536	106	0	15	0	0	5	525	63	1
179	5	5	8	7	29	38	0	1	0	0	0	43	6	1
180	5	5	10	8	46	58	0	0	0	0	1	37	5	1
181	5	5	17	11	765	120	0	0	0	0	4	749	78	1
182	5	5	9	8	132	155	0	0	0	0	1	221	33	1
183	5	4	29	17	1142	1143	35	29	0	0	307	2430	224	1
184	5	5	10	7	29	67	0	342	0	0	4	88	14	1
185	3	3	57	12	1944	1926	45	5	10	6	10	4466	273	1
186	4	5	17	6	281	29	0	0	0	0	0	223	24	1
187	1	4	25	3	224	230	0	0	0	0	0	340	31	1
188	5	5	11	10	152	188	0	0	0	0	4	275	37	1
189	5	4	34	33	592	486	0	0	0	0	0	997	46	1
190	4	4	24	9	257	443	1	1	0	0	76	555	55	1
191	5	5	10	8	45	57	0	0	0	0	1	62	7	1

192	5	5	9	8	32	59	0	0	0	0	0	44	6	1
193	5	4	24	15	1169	116	0	0	0	0	3	1074	106	1
194	4	4	27	8	233	410	13	0	0	0	4	525	52	1
195	3	3	45	11	1689	283	0	5	0	0	128	2006	171	1
196	3	3	54	11	1748	1934	63	9	12	7	5	4538	296	1
197	4	5	6	2	40	19	0	0	0	0	7	6	1	2
198	3	5	10	2	240	66	0	0	0	0	5	149	22	2
199	0	2	71	4	236	265	0	8	2	55	34	1362	77	2
200	4	5	9	4	249	89	0	0	0	0	5	150	23	2
201	4	5	12	4	132	95	0	1	0	0	23	55	7	2
202	4	4	39	19	628	1476	0	1	0	0	13	1739	85	2
203	4	5	10	3	311	151	0	0	0	0	4	343	52	2
204	4	3	49	22	226	50	0	1140	0	0	42	330	44	2
205	3	5	14	3	93	15	0	0	0	0	0	74	8	2
206	4	5	13	4	116	69	0	0	0	0	19	62	8	2
207	5	5	17	9	231	83	0	0	0	0	5	232	28	2
208	4	5	7	2	54	25	0	0	0	0	7	13	2	2
209	3	4	24	5	79	69	42	1	0	0	14	109	10	2
210	4	5	20	5	123	67	0	0	0	0	0	103	9	2
211	5	5	8	7	250	51	0	0	0	0	2	143	22	2
212	4	4	24	9	663	58	0	0	0	0	3	426	65	2
213	5	4	39	20	380	1070	0	0	9	0	135	1591	74	2
214	6	5	0	0	16	4	0	0	0	0	0	0	0	2
215	4	5	19	7	148	156	0	0	0	0	3	151	14	2
216	4	4	23	11	115	358	54	113	0	0	2	1458	185	2
217	4	4	28	13	644	221	0	0	0	0	0	714	75	2
218	4	5	11	5	256	64	0	1	0	0	6	211	30	2
219	5	5	19	11	138	96	0	0	0	0	92	245	31	2

220	1	5	17	2	212	77	0	0	0	0	0	140	20	2
221	5	5	9	5	282	47	0	3	0	0	2	171	26	2
222	4	5	7	2	136	24	0	0	0	0	7	51	8	2
223	4	5	20	9	168	179	0	0	0	0	3	175	17	2
224	5	5	20	10	594	92	0	0	0	0	3	547	57	2
225	4	5	18	5	186	213	0	0	0	0	5	338	38	2
226	4	5	12	4	63	62	14	0	0	0	1	34	4	2
227	4	5	18	7	149	138	0	0	0	0	4	130	13	2
228	5	5	20	16	108	130	0	0	0	0	22	191	19	2
229	3	5	17	3	114	18	0	0	0	0	0	93	9	2
230	5	5	20	11	95	92	0	0	0	0	98	239	31	2
231	2	2	66	9	610	959	0	2	0	0	3	1502	55	2
232	5	5	17	9	159	161	0	0	0	0	3	175	17	2
233	4	4	30	14	113	107	0	0	0	0	180	392	36	2
234	4	5	6	2	80	22	0	0	0	0	5	6	1	2
235	5	5	10	6	30	34	0	0	0	0	0	44	5	2
236	4	5	10	4	360	80	0	637	0	0	89	442	64	2
237	5	5	17	11	516	125	0	0	0	0	2	516	54	2
238	3	5	9	2	331	180	0	0	0	0	3	365	53	2
239	1	3	52	6	40	102	66	36	0	0	806	806	72	2
240	3	5	15	3	98	14	0	0	0	0	0	80	8	2
241	3	4	28	5	0	0	52	13	0	0	1	790	85	2
242	1	4	25	3	163	191	2	15	0	0	187	466	49	2
243	6	5	0	0	9	3	0	0	0	0	0	0	0	2
244	5	5	18	15	169	418	6	28	0	0	219	688	85	2
245	4	4	27	10	148	92	0	0	0	0	81	10	1	2
246	3	1	96	17	568	1092	2897	26	55	0	379	6538	338	2
247	5	4	32	17	1232	257	0	0	0	0	0	1145	130	2

248	5	5	8	4	169	23	0	0	0	0	2	47	7	2
249	3	4	26	6	265	425	2	10	0	0	1	551	55	2
250	5	4	29	18	1254	356	0	0	0	0	0	1220	120	2
251	5	5	11	8	34	53	0	0	0	0	0	58	6	2
252	5	5	18	11	0	0	17	11	0	0	178	151	12	2
253	6	5	0	0	11	5	0	0	0	0	0	0	0	2
254	4	4	35	15	683	255	0	0	0	0	0	773	77	2
255	4	5	9	3	79	47	3	0	0	0	3	30	4	2
256	5	5	10	7	35	37	0	0	0	0	0	46	5	2
257	3	1	96	18	656	1167	1870	21	58	0	71	6003	319	2
258	5	3	44	22	336	1085	0	7	0	0	115	1445	53	2
259	0	1	85	6	534	795	0	29	0	0	3	1552	69	2
260	4	5	19	6	154	142	0	0	0	0	4	178	16	2
261	4	4	29	13	726	217	0	0	0	0	0	557	63	2
262	2	1	82	11	331	336	0	9	4	44	28	1325	69	2
263	3	5	13	3	334	65	0	753	0	0	73	434	60	2
264	6	5	0	0	20	4	0	0	0	0	0	0	0	2
265	4	4	21	6	147	80	0	0	0	0	0	123	10	2
266	4	5	7	3	57	36	0	0	0	0	7	25	4	2
267	4	5	17	6	241	95	0	0	0	0	4	218	28	2
268	5	4	30	16	1392	414	0	0	0	0	0	933	113	2
269	3	5	13	3	94	16	0	0	0	0	0	75	8	2
270	4	5	19	5	118	80	0	0	0	0	0	110	10	2
271	3	1	88	18	605	1109	2386	33	39	0	363	6313	345	2
272	4	5	19	8	0	0	2	7	0	0	153	121	8	2
273	3	5	18	4	252	97	0	0	0	0	1	204	25	2
274	4	3	58	20	281	54	0	1344	0	0	50	390	52	2
275	5	4	37	20	629	1525	0	0	0	0	9	2057	105	2

276	4	5	14	5	164	174	0	1	0	0	8	278	33	2
277	4	4	39	13	935	249	0	0	0	0	0	901	111	2
278	4	5	19	6	178	127	0	0	0	0	0	156	14	2
279	4	5	12	3	332	94	0	128	0	0	5	352	51	2
280	5	4	27	16	1284	356	0	0	0	0	0	789	93	2
281	5	4	32	21	1000	239	0	0	0	0	5	947	102	2
282	5	5	9	6	253	88	0	0	0	0	4	148	22	2
283	5	5	9	6	292	102	0	1	0	0	4	167	25	2
284	5	4	21	14	148	107	0	0	0	0	90	336	39	2
285	4	5	15	5	348	107	0	2	0	0	58	415	60	2
286	5	5	10	7	31	42	0	0	0	0	0	48	5	2
287	5	5	9	6	300	137	0	0	0	0	4	304	46	2
288	4	5	11	5	338	151	0	0	0	0	3	346	52	2
289	4	5	8	3	269	64	0	277	0	0	38	296	46	2
290	5	5	12	6	30	42	0	0	0	0	0	44	4	2
291	5	5	13	8	44	60	0	0	0	0	0	72	7	2
292	2	2	61	9	532	864	0	2	0	0	3	1399	54	2
293	5	5	10	7	242	77	0	0	0	0	3	249	37	2
294	4	5	7	2	19	8	0	0	0	0	7	7	1	2

• APPENDIX B : CNN MODEL

```
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.layers import BatchNormalization
import matplotlib.pyplot as plt
import numpy as np

# Initialising the CNN
activation = 'sigmoid'
# activation = 'relu'
classifier = Sequential()

# Step 1 - Convolution
classifier.add(Conv2D(32, (3, 3), input_shape = ( 150, 150, 3), activation = activation))
classifier.add(BatchNormalization())
```

```
# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (3, 3)))

# Adding a second convolutional layer
classifier.add(Conv2D(64, (3, 3), activation = activation))
classifier.add(BatchNormalization())
classifier.add(MaxPooling2D(pool_size = (3, 3)))

classifier.add(Conv2D(128, (3, 3), activation = activation))
classifier.add(BatchNormalization())
classifier.add(MaxPooling2D(pool_size = (3, 3)))

# Step 3 - Flattening
classifier.add(Flatten())

# Step 4 - Full connection
classifier.add(Dense(128, activation=activation, kernel_initializer='he_uniform'))
classifier.add(Dense(units = 3, activation = 'softmax'))

# optimizer = 'rmsprop'
# optimizer = 'adam',
# Compiling the CNN
# loss = "sparse_categorical_crossentropy"
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

############################################
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = 
True)
test_datagen = ImageDataGenerator(rescale = 1./255)

# train_path = '/home/nadeeshan/Desktop/share/MSC/python/cnn3/'
# test_path = '/home/nadeeshan/Desktop/share/MSC/python/cnn3'

train_path = '/home/nadeeshan/Desktop/share/MSC/python/cnn1/train/'
test_path = '/home/nadeeshan/Desktop/share/MSC/python/cnn1/test/'

# train_path = '/home/nadeeshan/Desktop/share/MSC/python/extracted_data/train/'
# test_path = '/home/nadeeshan/Desktop/share/MSC/python/extracted_data/test/'

training_set = train_datagen.flow_from_directory(train_path,
                                                 target_size = (150, 150),
                                                 batch_size = 15,
                                                 class_mode = 'categorical')
test_set = test_datagen.flow_from_directory(test_path,
                                            target_size = (150, 150),
                                            batch_size = 15,
                                            class_mode = 'categorical')


history = classifier.fit_generator(training_set,steps_per_epoch=10,epochs=10,verbose=1,validation_data = 
test_set,validation_steps = 500)
plt.plot(history.history['accuracy'], label=['Accuracy'])
plt.plot(history.history['loss'],label =['Loss'])
plt.xlabel('Epoch')
plt.ylabel('Accuracy/ Loss')
plt.ylim([0.5, 2])
plt.legend(loc='upper left')
plt.show()

# tfjs.converters.save_keras_model(classifier, 'CNNN.json')
```

```
classifier.save('CNN_1.json')
print(classifier.summary())
```

- ## APPENDIX C : PADDY DISEASE DIAGNOSIS API

```
from flask import Flask,request,jsonify
import pickle
import numpy as np
import sys
import json
from flask_cors import CORS
import cv2
import sys
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
import os
from os import listdir
from numpy import asarray
from numpy import save
from flask import jsonify
import ntpath

app = Flask(__name__)
CORS(app)

@app.route('/')
def index():
    return "Weolcome to the Paddy leaves Diagnosis"

def path_leaf(path):
    head, tail = ntpath.split(path)
    return tail or ntpath.basename(head)

def image2gray(url):
    dest = '/var/www/html/images_gray/'
    image = cv2.cvtColor(cv2.imread(url), cv2.COLOR_BGR2GRAY)
    cv2.imwrite(str(dest+path_leaf(url)),image)

def shape_count(x_array,y_array,pixel_colors,disease_shapes):

    if(len(x_array) > 2):

        max_x = np.max(x_array)
        min_x = np.min(x_array)
        max_y = np.max(y_array)
        min_y = np.min(y_array)

        mean_x = int(np.mean(x_array))
        mean_y = int(np.mean(x_array))

        x_dif = max_x - min_x
        y_dif = max_y - min_y

        if(x_dif > y_dif):
            ratio = x_dif/y_dif
        else:
            ratio = y_dif/x_dif


        unique_y = len(set(y_array))
        unique_x = len(set(x_array))
```

```
        if(unique_x > unique_y):
            ratio1 = unique_x/unique_y
        else:
            ratio1 = unique_y/unique_x

        if(ratio< 1.5 and ratio1 <1.5):
            disease_shapes.append("Circle")
        elif(ratio <= 2 and ratio1 <=2):
            disease_shapes.append("Oval")
        elif(ratio <= 3 and ratio1 <=3):
            disease_shapes.append("Strip_short")
        else:
            disease_shapes.append("Strip_long")

def shape_ditection(img,x_disease,y_disease,img_old):
    i = 0
    y_array = [] #to save sub part of array
    x_array = []
    img_old2hsv  = cv2.cvtColor(img_old, cv2.COLOR_RGB2HSV)

    disease_shapes = [] # to store return shapes from shape_count
    shapes_colors = [] #store colors of each shape

    pixel_colors = {'black':0,'brown':0,'yellow':0,'orange':0,'gray':0,'white':0,'red1':0,'red2':0}

    for i in range(len(y_disease)-1):

        if((y_disease[i+1] - y_disease[i])>2):
            if(len(x_array) > 4):
                shape_count(x_array,y_array,pixel_colors,disease_shapes) # fill data to disease_shapes array
                shape_color = ''
                for k, v in pixel_colors.items():
                    if(v != 0):
                        shape_color+=k
                shapes_colors.append(shape_color)
            y_array.clear() #tnew array
            x_array.clear()
            # pixel_colors = {'black':0,'brown':0,'yellow':0,'orange':0,'gray':0,'white':0,'red1':0,'red2':0}

        else:
            x_array.append(x_disease[i])
            y_array.append(y_disease[i])
            # print(img_old[x_disease[i],y_disease[i]])
            img_color = img_old2hsv[x_disease[i],y_disease[i]]
            if(img_color[0]>=0 and img_color[0]<=180 and img_color[1]>=0 and img_color[1]<=255 and
img_color[2]>=0 and img_color[2]<=40):
                # print( img_old[x_disease[i],y_disease[i]],'black')
                pixel_colors['black'] = pixel_colors['black']+1
            elif(img_color[0]>=10 and img_color[0]<=20 and img_color[1]>=100 and img_color[1]<=255 and
img_color[2]>=20 and img_color[2]<=200):
                # print( img_old[x_disease[i],y_disease[i]],'brown')
                pixel_colors['brown'] = pixel_colors['brown']+1
            elif(img_color[0]>=25 and img_color[0]<=35 and img_color[1]>=50 and img_color[1]<=255 and
img_color[2]>=70 and img_color[2]<=255):
                # print( img_old[x_disease[i],y_disease[i]],'Yellow')
                pixel_colors['yellow'] = pixel_colors['yellow']+1
            elif(img_color[0]>=10 and img_color[0]<=24 and img_color[1]>=50 and img_color[1]<=255 and
img_color[2]>=70 and img_color[2]<=255):
                # print( img_old[x_disease[i],y_disease[i]],'Orange')
                pixel_colors['orange'] = pixel_colors['orange']+1
            elif(img_color[0]>=0 and img_color[0]<=180 and img_color[1]>=0 and img_color[1]<=18 and
img_color[2]>=40 and img_color[2]<=230):
```

```
            # print( img_old[x_disease[i],y_disease[i]],'Gray')
            pixel_colors['gray'] = pixel_colors['gray']+1
        elif(img_color[0]>=0 and img_color[0]<=180 and img_color[1]>=0 and img_color[1]<=18 and
img_color[2]>=231 and img_color[2]<=255):
            # print( img_old[x_disease[i],y_disease[i]],'White')
            pixel_colors['white'] = pixel_colors['white']+1
        elif(img_color[0]>=159 and img_color[0]<=180 and img_color[1]>=50 and img_color[1]<=255 and
img_color[2]>=70 and img_color[2]<=255):
            # print( img_old[x_disease[i],y_disease[i]],'Red1')
            pixel_colors['red1'] = pixel_colors['red1']+1
        elif(img_color[0]>=0 and img_color[0]<=9 and img_color[1]>=50 and img_color[1]<=255 and
img_color[2]>=70 and img_color[2]<=255):
            # print( img_old[x_disease[i],y_disease[i]],'Red2')
            pixel_colors['red2'] = pixel_colors['red2']+1
    print("-------------------------------------------")
    print(pixel_colors)


def disease_ares(img_hsv,x_upper,x_lower,y_upper,img_old,url):
    i = 0
    x_disease = []
    y_disease = []
    dest = '/var/www/html/images_e/'
    try:
        for y in y_upper:
            if(x_lower[i]>x_upper[i]):
                lower = x_upper[i]
                upper = x_lower[i]
            else:
                upper = x_upper[i]
                lower = x_lower[i]
            for x in range(lower,upper):
                if(np.all(img_hsv[x,y]) == np.all([0,0,0])):
                    img_hsv[x,y] = [255,255,255]
                    x_disease.append(x)
                    y_disease.append(y)
                else:
                    img_old[x,y] = [255,255,255,0]
            i+=1
    except:
        print("index out of range", x_disease, y_disease)

    img = cv2.cvtColor(img_old, cv2.COLOR_RGBA2BGR)
    img = cv2.resize(img,(150,150))
    cv2.imwrite(str(dest+path_leaf(url)),img)
    shape_ditection(img_hsv,x_disease,y_disease,img_old)


def leaf_pixels(url):

    img = cv2.imread(url)
    img = cv2.resize(img,(150,150))
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img_rgba = cv2.cvtColor(img, cv2.COLOR_BGR2RGBA)
    img_hsv = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2HSV)

    low_green = np.array([25,52,72])
    high_green = np.array([126,255,255])

    mask = cv2.inRange(img_hsv, low_green, high_green)
    image = cv2.bitwise_and(img_rgb,img_rgb,mask=mask)

    dimensions = image.shape
```

```
    w = dimensions[0]
    h = dimensions[1]
    x_axis = []
    y_axis = []
    x_upper =[]
    x_lower =[]
    y_upper =[]
    y_lower =[]

    for y in range(h):
        upper_x = 0
        lower_x = 0
        check = 0
        for x in range(int(w)):
            if(np.all(image[x,y]) != np.all([0,0,0])):
                x_axis.append(x)
                y_axis.append(y)

                if(lower_x == 0):
                    x_lower.append(x)
                    y_lower.append(y)
                    lower_x = 1
                    image[x,y] =[255,255,255]
                    check = 1
                elif(upper_x < x):
                    upper_x = x
            elif(check == 0):
                img_rgba[x,y] = [255,255,255,0]

        x_upper.append(upper_x)
        y_upper.append(y)
        image[upper_x,y] =[255,255,255]

        for x in range(upper_x,int(w)): # remove lower boundry pixels
            img_rgba[x,y] = [255,255,255,0]

    if(len(x_axis) < 3000):
        return '0'
    else:
        x_std = np.std(x_axis)
        y_std = np.std(y_axis)

        if(len(x_axis) < 5000 and x_std > 20 and y_std>20):
            return '0'
        elif(len(x_axis) < 8000 and x_std > 30 and y_std>30):
            return '0'
        elif(x_std <50 and y_std<50):
            disease_ares(image,x_upper,x_lower,y_upper,img_rgba,url) # pass leaf upper and lower coordinates as
array
        else:
            return '0'

@app.route("/classify",methods=['GET'])
def imageClassify (url):
    test_image = image.load_img(url, target_size = (150, 150))
    test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis = 0)

    classifier = load_model('CNN2.json')

    result = classifier.predict(test_image)

#           print(result)
```

```
    index = result.argmax()
    if index == 0:
        prediction = 'Bacterial Leaf Blight'
    elif index == 1:
        prediction = 'Brown Spot'
    else:
        prediction = 'Leaf Smut'

    return prediction

@app.route("/classify_gray",methods=['GET'])
def imageClassify_gray (url):
        test_image = image.load_img(url,target_size = (150, 150))
        test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis = 0)

        classifier = load_model('CNN_gray.json')

        result = classifier.predict(test_image)

    #   print(result)

        index = result.argmax()
        if index == 0:
                prediction = 'Bacterial Leaf Blight'
        elif index == 1:
                prediction = 'Brown Spot'
        else:
                prediction = 'Leaf Smut'

        return prediction

@app.route("/classify_e",methods=['GET'])
def imageClassify_e (url):
        test_image = image.load_img(url,target_size = (200, 200))
        test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis = 0)

        classifier = load_model('CNN_E.json')

        result = classifier.predict(test_image)

    #   print(result)

        index = result.argmax()
        if index == 0:
                prediction = 'Bacterial Leaf Blight'
        elif index == 1:
                prediction = 'Brown Spot'
        else:
                prediction = 'Leaf Smut'

        return prediction

@app.route("/disease",methods=['GET'])
def main1():
    path = request.args.get('url')
    predict = leaf_pixels(path)
    image2gray(path)
    if(predict == '0'):
        return '0'
    else:
```

```
        predicted = imageClassify(path)
        return predicted

@app.route("/disease_gray",methods=['GET'])
def main2():
    path = request.args.get('url')
    predicted = imageClassify_gray(path)
    return predicted

@app.route("/disease_e",methods=['GET'])
def main3():
    path = request.args.get('url')
    predicted = imageClassify_e(path)
    return predicted

if __name__ == "__main__":
    print("Started http://localhost:5000/")
    app.run(debug=True,host="0.0.0.0")
```

- ## APPENDIX D : PHP CODE

```
<?php

function image2gray($path)

{

        $im = imagecreatefrompng($path);

        if($im && imagefilter($im, IMG_FILTER_GRAYSCALE))

        {

            echo 'Grayscale conversion successful';

          //Saving and replacing the original png file with the grayscale png

            imagepng($im, 'colorful.png');

        }

        else

        {

          //generates error message if process is unsuccessfull

            echo 'Conversion failed.';

        }

        //clearing the memory buffer

        imagedestroy($im);

}

?>
```

```
<html>

<head>

<title>Paddy Leaf Diagnosis</title>

<link rel="preconnect" href="https://fonts.gstatic.com">

<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@300;400;500&display=swap"
rel="stylesheet">

  <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-
scale=1.0">

</head>

<body>

<section class="canvas-wrap">

    <div class="canvas-content">

        <div class="header"><h1>PADDY LEAVES DIAGNOSIS</h1></div>

              <form method="POST" action="" enctype="multipart/form-data">

              <div class="upload-btn-wrapper">

                  <p><h1>Upload Paddy leaf Image</h1><br>

                            <input type ="file"  name = "UploadFileName" class="inputfile" id="actual-btn" hidden>

                            <label for="actual-btn">No file chosen</label>

                            </p>

                            <br />

              </div>

                  <input type = "submit" class="button" name = "Submit" value = "Upload">

              </form>



              <?php

                            $img_path= "";

                            if(isset($_FILES['UploadFileName'])){

                                  $errors= array();

                                  $file_name = $_FILES['UploadFileName']['name'];

                                  $file_size =$_FILES['UploadFileName']['size'];

                                  $file_tmp =$_FILES['UploadFileName']['tmp_name'];
```

```
$file_type=$_FILES['UploadFileName']['type'];

$file_ext=strtolower(end(explode('.',$_FILES['UploadFileName']['name'])));

$extensions= array("jpeg","jpg","png");


if(in_array($file_ext,$extensions)=== false){

        $errors[]="extension not allowed, please choose a JPEG or PNG file.";

}


if($file_size > 2097152){

        $errors[]='File size must be excately 2 MB';

}


if(empty($errors)==true){

        move_uploaded_file($file_tmp,"images/".$file_name);

       $img_path1 = "/var/www/html/images/".$file_name;

       $img_path2 = "/var/www/html/images_gray/".$file_name;

       $img_path3 = "/var/www/html/images_gray/".$file_name;




?>

<div class="result">

<?php

$descBlb="<span style='color:black;font-size: 14px;'>Bacterial Leaf Blight (Xanthomonas Oryzae) is a disease caused by bacteria called Xanthomonas Oryzae Pv. Oryzae. This disease mostly occurs in rainy and dry-wet sessions</span>";

$descBs ="<span style='color:black;font-size: 20px;'>Brown spot caused by the fungus Helmintosporium Oryzae on plantations. Death of young plants and reduce grain quality are the consequences of brown Spot disease</span>";

$descLs ="<span style='color:black;font-size: 14px;'>Leaf Smut caused by the fungus Entyloma oryzae, is a widely distributed, but somewhat minor disease of rice. Slightly rasied, angular; black spots on both side of the leaves. Although rare, it also can produce spots on leaf sheaths</span>";


       $url = "http://localhost:5000/disease?url=".$img_path1;

       $client = curl_init($url);
```

```
curl_setopt($client,CURLOPT_RETURNTRANSFER,true);
$response1 = curl_exec($client);
//$result = json_decode($response);

if($response1 =='0')
        echo ("<h2>The image may not have a paddy leaf</h2>");
else
{
$url = "http://localhost:5000/disease_gray?url=".$img_path2;
$client = curl_init($url);

curl_setopt($client,CURLOPT_RETURNTRANSFER,true);
$response2 = curl_exec($client);

if($response1 == $response2)
{
        if($response1 == 'Bacterial Leaf Blight')
                echo ("<h2>Infected Disease: </h2><h1>Bacterial Leaf
Blight</h1><br>".$descBlb);
        elseif($response1 == 'Brown Spot')
                echo ("<h2>Infected Disease: </h2><h1>Brown
Spot</h1><br>".$descBs);
        elseif($response1 == 'Leaf Smut')
                echo ("<h2>Infected Disease: </h2><h1>Leaf
Smut</h1><br>".$descLs);
}

//echo ("<h2>Infected Disease: </h2><h3>".$response2."</h3>");
}

?>
```

```
            </div>

                        <div id="original" class="img"><img src="<?php echo "images/".$file_name; ?>"
width="500" height="300"></div>

                        <div id="gray" class="img"><img src="<?php echo "images_gray/".$file_name; ?>"
width="500" height="300">

            <img src="<?php if($response1 !='0') echo "images_e/".$file_name; ?>" width="500"
height="300"></div>




    </div>

    <div id="canvas" class="gradient"></div>


</section>

<script src="js/three.min.js"></script>

<script src="js/projector.js"></script>

<script src="js/canvas-renderer.js"></script>

<script src="js/3d-lines-animation.js"></script>

<script src="http://cdnjs.cloudflare.com/ajax/libs/jquery/2.0.2/jquery.min.js"></script>

<script src="js/color.js"></script>

<?php

                  }

                }

"http://localhost:5000/disease?url=/home/nadeeshan/Desktop/share/MSC/python/cnn1/test/bs/bs_0_4897.png";


                ?>


</body>

</html>
```

REFERENCES

Arivazhagan, S., Shebiah, R.N., Ananthi, S., Varthini, S.V., 2013. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features 15, 7.

Arjuna Abeysinghe Gunawardana, 2018. Agriculture sector performance in the Sri Lankan Economy: A systematic review and a Meta data analysis from year 2012 to 2016. https://doi.org/10.13140/RG.2.2.34177.92003

Bakar, M.N.A., Abdullah, A.H., Rahim, N.A., Yazid, H., Misman, S.N., Masnan, M.J., 2018. Rice Leaf Blast Disease Detection Using Multi- Level Colour Image Thresholding 10, 6.

Basilio, J.A.M., Torres, G.A., Pérez, S., Medina, L.K.T., Meana, H.M.P., 2014. Explicit Image Detection using YCbCr Space Color Model as Skin Detection 7.

Basu, M., 2002. Gaussian-based edge-detection methods-a survey. IEEE Trans. Syst., Man, Cybern. C 32, 252–260. https://doi.org/10.1109/TSMCC.2002.804448

Battiato, S., Gallo, G., Stanco, F., 2002. A locally adaptive zooming algorithm for digital images. Image and Vision Computing 20, 805–812. https://doi.org/10.1016/S0262-8856(02)00089-6

Buscema, M., 1998. Back Propagation Neural Networks 38.

Chai, D., Bouzerdoum, A., 2000. A Bayesian approach to skin color classification in YCbCr color space, in: 2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No.00CH37119). Presented at the 2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium, IEEE, Kuala Lumpur, Malaysia, pp. 421–424. https://doi.org/10.1109/TENCON.2000.888774

Chaudhari, P.R., Tamrakar, N., Singh, L., Tandon, A., Sharma, D., n.d. Rice nutritional and medicinal properties: A review article 7.

Chen Yu, Chen Dian-ren, Li Yang, Chen Lei, 2010. Otsu's thresholding method based on gray level-gradient two-dimensional histogram, in: 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010). Presented at the 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), IEEE, Wuhan, China, pp. 282–285. https://doi.org/10.1109/CAR.2010.5456687

Deshmukh, R., 2015. Detection of Paddy Leaf Diseases. International Journal of Computer Applications 3.

Ghazali, K.H., Mansor, M.F., Mustafa, Mohd.M., Hussain, A., 2007. Feature Extraction Technique using Discrete Wavelet Transform for Image Classification, in: 2007 5th Student Conference on Research and Development. Presented at the 2007 5th Student Conference on Research and Development, IEEE, Selangor, Malaysia, pp. 1–4. https://doi.org/10.1109/SCORED.2007.4451366

Groth, D., Agcenter, L.S.U., n.d. Rice Disease Identification Photo Link.

Gunawardana, A.A., 2018. Agriculture sector performance in the Sri Lankan Economy: A systematic review and a Meta data analysis from year 2012 to 2016. https://doi.org/10.13140/RG.2.2.34177.92003

Hassan, R., Ema, R.R., 2017. Color Image Segmentation using Automated K-Means Clustering with RGB and HSV Color Spaces 10.

Ihsan, I., Hidayat, E.W., Rahmatulloh, A., 2020. Identification of Bacterial Leaf Blight and Brown Spot Disease In Rice Plants With Image Processing Approach. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika 5, 59. https://doi.org/10.26555/jiteki.v5i2.14136

Jhuria, M., Kumar, A., Borse, R., 2013. Image processing for smart farming: Detection of disease and fruit grading, in: 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013). Presented at the 2013 IEEE Second International Conference on Image Information Processing (ICIIP), IEEE, Shimla, India, pp. 521–526. https://doi.org/10.1109/ICIIP.2013.6707647

Kamlapurkar, S.R., 2016. Detection of Plant Leaf Disease Using Image Processing Approach 6, 4.

Khirade, S., Patil, A.B., 2015. [No title found], in: 2015 International Conference on Computing Communication Control and Automation. Presented at the 2015 International Conference on Computing, Communication, Control and Automation (ICCUBEA), IEEE, Pune.

Ly, B.C.K., Dyer, E.B., Feig, J.L., Chien, A.L., Del Bino, S., 2020. Research Techniques Made Simple: Cutaneous Colorimetry: A Reliable Technique for Objective Skin Color Measurement. Journal of Investigative Dermatology 140, 3-12.e1. https://doi.org/10.1016/j.jid.2019.11.003

Maheswari, S., Korah, R., 2016. Review on image segmentation based on color space and its hybrid, in: 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT). Presented at the 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), IEEE, Kumaracoil, India, pp. 639–641. https://doi.org/10.1109/ICCICCT.2016.7988028

Mangla, D.N., Raj, P.B., Hegde, S.G., 2019. Paddy Leaf Disease Detection Using Image Processing and Machine Learning 7, 4.

Matera, F., 1998. Self-Organizing Maps 17.

Moeslund, T.B., 2012. BLOB Analysis, in: Introduction to Video and Image Processing, Undergraduate Topics in Computer Science. Springer London, London, pp. 103–115. https://doi.org/10.1007/978-1-4471-2503-7_7

Mukherjee, M., Pal, T., Samanta, D., 2010. DAMAGED PADDY LEAF DETECTION USING IMAGE PROCESSING 4.

Narvekar, P., Kumbhar, M.M., Patil, S.N., 2007. Grape Leaf Diseases Detection & Analysis using SGDM Matrix Method 2, 8.

Nazarkevych, M., Oliarnyk, R., Dmytruk, S., 2017. An images filtration using the Ateb-Gabor method, in: 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). Presented at the 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), IEEE, Lviv, pp. 208–211. https://doi.org/10.1109/STC-CSIT.2017.8098770

Noble, W.S., 2006. What is a support vector machine? Nat Biotechnol 24, 1565–1567. https://doi.org/10.1038/nbt1206-1565

Nugaliyadde, L., Dissanayake, N., Mitrasena, J., n.d. Advance in pest and disease management of rice in sri lanka : A review.

Orillo, J.W., Dela Cruz, J., Agapito, L., Satimbre, P.J., Valenzuela, I., 2014. Identification of diseases in rice plant (oryza sativa) using back propagation Artificial Neural Network, in: 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). Presented at the 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), IEEE, Palawan, Philippines, pp. 1–6. https://doi.org/10.1109/HNICEM.2014.7016248

Padmavathi, K., Thangadurai, K., 2016. Implementation of RGB and Grayscale Images in Plant Leaves Disease Detection – Comparative Study. Indian Journal of Science and Technology 9. https://doi.org/10.17485/ijst/2016/v9i6/77739

Perreault, S., Hebert, P., 2007. Median Filtering in Constant Time. IEEE Trans. on Image Process. 16, 2389–2394. https://doi.org/10.1109/TIP.2007.902329

Phadikar, S., Sil, J., 2008. Rice disease identification using pattern recognition techniques, in: 2008 11th International Conference on Computer and Information Technology. Presented at the 2008 11th International Conference on Computer and Information Technology (ICCIT), IEEE, Khulna, Bangladesh, pp. 420–423. https://doi.org/10.1109/ICCITECHN.2008.4803079

Prather, D.C., 1971. Trial-and-Error versus Errorless Learning: Training, Transfer, and Stress. The American Journal of Psychology 84, 377. https://doi.org/10.2307/1420469

Rambukwella, R., Priyankara, E.A.C., 2016. Production and marketing of traditional rice varieties in selected districts in Sri Lanka: present status and future prospects, Research report. Hector Kobbekaduwa Agrarian Research and Training Institute, Colombo, Sri Lanka.

Ramesh, S., Vydeki, D., 2020. Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm. Information Processing in Agriculture 7, 249–260. https://doi.org/10.1016/j.inpa.2019.09.002

Rusiñol, M., Dosch, P., Lladós, J., 2007. Boundary Shape Recognition Using Accumulated Length and Angle Information, in: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (Eds.), Pattern Recognition and Image Analysis, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 210–217. https://doi.org/10.1007/978-3-540-72849-8_27

Seneviratne, S.N. de S., Jeyanandarajah, P., 2004. RICE DISEASES -PROBLEM AND PROGRESS. Tropical Agricultural Research and Extension.

Sethy, P.K., Barpanda, N.K., Rath, A.K., Behera, S.K., 2020. Image Processing Techniques for Diagnosing Rice Plant Disease: A Survey. Procedia Computer Science 167, 516–530. https://doi.org/10.1016/j.procs.2020.03.308

Sghair, M.E., Jovanovic, R., Tuba, M., 2017. An Algorithm for Plant Diseases Detection Based on Color Features 2, 6.

Shah, J.P., Prajapati, H.B., Dabhi, V.K., 2016. A survey on detection and classification of rice plant diseases, in: 2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC). Presented at the 2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), IEEE, Bangalore, India, pp. 1–8. https://doi.org/10.1109/ICCTAC.2016.7567333

Shanmugavadivu, P., Kumar, A., 2013. Boundary Detection of Objects in Digital Images Using Bit-Planes and Threshold Modified Canny Method, in: Prasath, R., Kathirvalavakumar, T. (Eds.), Mining Intelligence and Knowledge Exploration, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 192–200. https://doi.org/10.1007/978-3-319-03844-5_20

Singh, A., Singh, M.L., 2015. Automated color prediction of paddy crop leaf using image processing, in: 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR). Presented at the 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), IEEE, Chennai, India, pp. 24–32. https://doi.org/10.1109/TIAR.2015.7358526

Vadivel, A., Sural, S., Majumdar, A.K., 2007. An Integrated Color and Intensity Co-occurrence Matrix. Pattern Recognition Letters 28, 974–983. https://doi.org/10.1016/j.patrec.2007.01.004

Vora, P., 2013. A Survey on K-mean Clustering and Particle Swarm Optimization 1, 3.

Jayanthi, G., Archana, K.S., Saritha, A., n.d. Analysis of Automatic Rice Disease Classification Using Image Processing Techniques 8, 6.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., n.d. LightGBM: A Highly Efficient Gradient Boosting Decision Tree 9.

S. Archana, K., Sahayadhas, A., 2018. Comparison of various filters for noise removal in paddy leaf images. IJET 7, 372. https://doi.org/10.14419/ijet.v7i2.21.12444