THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Master of Engineering (Agricultural Engineering)
DEGREE

Agricultural Engineering
FIELD

Agricultural Engineering
DEPARTMENT

TITLE: Detection of Brown Leaf Spot Disease in Cassava using Image Analysis

NAME: Mr. Kittipong Powbunthorn

THIS THESIS HAS BEEN ACCEPTED BY

THESIS ADVISOR
(Mr. Wanrat Abdullakasim, Ph.D.)

THESIS CO-ADVISOR
(Mrs. Jintana Unartngam, Ph.D.)

DEPARTMENT HEAD
(Associate Professor Anupun Terdwongworakul, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON _____

DEAN
(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

DETECTION OF BROWN LEAF SPOT DISEASE IN CASSAVA USING IMAGE ANALYSIS

KITTIPONG POWBUNTHORN

A Thesis Submitted in Partial Fulfillment of
the Requirement for the Degree of
Master of Engineering (Agricultural Engineering)
Graduate School, Kasetsart University
2012

Kittipong Powbunthorn 2012: Detection of Brown Leaf Spot Disease in Cassava using Image Analysis. Master of Engineering (Agricultural Engineering), Major Field: Agricultural Engineering, Department of Agricultural Engineering. Thesis Advisor: Mr. Wanrat Abdullakasim, Ph.D. 156 pages.

The objectives of the present study were to develop image analysis techniques for *in situ* detection of the Brown Leaf Spot disease in cassava (*Manihot esculenta* Crantz) caused by *Cercosporidium henningsii* Allesch, as well as for assessing its severity level. Digital images of cassava leaves captured in both controlled and actual field conditions were used in the analyses. Several color indices were used as descriptors to distinguish the infected leaves from the healthy ones. These included red (*r*), green (*g*) and blue (*b*) chromatic coordinates derived from original RGB values, contrast indices $r - g$, $g - b$, $(g - b)/|r - g|$ and $2g - r - b$, and hue (*H*), saturation (*S*) and intensity (*I*) which forms the HSI space. The artificial neural network (ANN) and discriminant analysis (DA) were applied to classify the healthy and the diseased leaves. Experimental results indicated the validity of color indices associated with the ANN and DA classification approach in discriminating the diseased plants from the non-diseased ones. Additional image processing algorithms based on color were further developed in order to locate the positions of brown spot on leaves which allows a higher precision in detecting the disease. The results showed that the algorithm relying on intersection between *H* and *Excess Green* (*ExG*) yielded highest accuracy. Comparison between the number of brown spots detected by image analysis with that obtained from manual counting indicated a good correlation ($R^2 = 0.8325$) and a least RMSE of 10.56.

In the severity level scoring, an image analysis algorithm has been developed. The *H* and *I* values were used in segmentation and feature extraction to calculate the percentage of infection area and to count the number of spots. Comparing the image analysis results with manual scoring performed by raters based on conventional illustrated diagram key demonstrated a satisfactory consistency ($R^2 = 0.8993$).

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

ACKNOWLEDGEMENTS

I hereby express my deep gratitude to Dr. Wanrat Abdullakasim, my principal thesis advisor for his valuable guidance and kindness, without whose encouragement this thesis would not have been completed. I would like to thank Dr. Jintana Unartngam, my co-advisor for her substantial advices regarding plant pathology. I am really grateful to Prof. Dr. Tomohiro Takigawa for providing me a great opportunity to conduct parts of this research at the University of Tsukuba.

I would like to thank Assoc. Prof. Dr. Tofael Ahamed, Mr. Supachai Kulmutiwat, and Mr. Pawin Thanpattranon for their helps during my stay at the University of Tsukuba. I would like to extend my heartfelt thanks to my fellows, Ms. Chutima Kapkaeo, Ms. Phraeophan Rungrueang, and everyone in the Department of Agricultural Engineering and in the Bioproduction and Machinery Laboratory.

With all humbleness, I thank the Graduate School and the Faculty of Engineering at Kamphaeng Saen for providing assistantships supportive to my academic activities. I am indebted to Kasetsart University for providing a scholarship under the Capacity Building for KU students on Internationalization Program, as well as to JASSO scholarship of Japan, which have enabled me to carrying out parts of my research at the University of Tsukuba.

I greatly appreciated Ms. Sonlaya Suksa-Ard and Ms. Benjamas Keawnuch who always devotes themselves for me every time I was in trouble. Finally, I dedicate this thesis to my parents for their loves, compassion, and continual encouragement.

Kittipong Powbunthorn

October 2012

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF ABBREVIATIONS

a	=	Activation
A_D	=	Total disease area
A_L	=	Healthy leaf area
AG	=	Algorithm
ANN	=	Artificial neural network
B	=	Background
BLS	=	Brown leaf spot
B-L	=	Background and leaf
B-L-S	=	Background, leaf and spot
CAD	=	Cassava anthracnose disease
CBB	=	Cassava Bacterial Blight
CMD	=	Cassava mosaic disease
CO_2	=	Carbon dioxide
DA	=	Discriminant analysis
DLS	=	Diffuse Leaf Spot
ExG	=	Exceed green index
ExR	=	Exceed red index
H	=	Hue image
I	=	Intensity image
L	=	Leaf
L-S	=	Leaf and spot
LAI	=	Leaf Area Index
OAE	=	Office of Agricultural Economics
PCA	=	Principal Components Analysis
PI	=	Percentage of inflection
REGR	=	Regression factor scores
ROI	=	Regions of interest
RLS	=	Ring Leaf Spot

LIST OF ABBREVIATIONS (Continued)

RMSE	=	Root Mean Squared Error
S	=	Saturation image
T	=	Tolerance
Std.dev	=	Standard deviation
VIF	=	Variance inflation factor
VR	=	Visual rater

DETECTION OF BROWN LEAF SPOT DISEASE IN CASSAVA USING IMAGE ANALYSIS

INTRODUCTION

Cassava (*Maihot esculenta* Crantz) is of important industrial crop which plays a crucial role to Thailand's economy nowadays. The value of cassava has been markedly increasing as it plays a new role as an energy crop. Cassava starch is a potential raw material for ethanol production while its residues can be utilized as feedstock in gasification process. This has resulted in an increase in the demand of cassava which in turn suggested the need of efficient cassava production systems.

Among field crops, cassava is usually recognized as a crop of having high tolerance to environmental stress as well as to the invasion of pests and diseases by nature. However, a most destructive epidemic of cassava mealy bugs in Thailand during 2007 – 2010 has demonstrated that cassava, nevertheless, is not absolutely free from pests and disease. In addition, the implication of climate change could be another factor that might causes unusual epidemic development of pests and diseases in cassava and hence may not be neglected.

The production of cassava in the future therefore requires and efficient monitoring system which is capable of providing early detection as well as the tracing of spatial and temporal variations of pests and diseases development in fields. Crop protection based on site-specific regime thus may be suitable to serve these requirements. It is advantageous if the cassava plants are monitored at the same time the regular treatments, e.g. weeding, fertilizer spraying, are carried out using a machine vision equipment mounted on a tractor. It is even more beneficial if the crop is continuously observed using automated field observers. Diseased cassava plants are usually observable from visual symptoms and hence suggest the possibility of damaged plants identification using machine vision. This requires an accurate image analysis technique to characterizing atypical appearances of the diseased cassava plant. This study is therefore to be taken place to implement this conceptual approach.

OBJECTIVES

The objective of the present study was to develop an image analysis technique which is capable of detecting visual symptoms of Brown Leaf Spot disease in cassava, which can be divided into three specific objectives as follows:

1. To find out image parameters that potential in reflecting the features of Brown Leaf Spot disease on cassava plants.

2. To develop an optimal image processing algorithm which is capable of detecting infected cassava plants in actual field conditions.

3. To develop an image processing algorithm for assessing the severity level of disease infection.

LITERATURE REVIEW

1. Cassava production in Thailand and economic importance

Cassava is of most important field crop to Thailand's economy in which it plays different functions in a wide range of domestic industries as well as contributing a large portion to the exporting income.

Thailand's uniquely low human consumption sets it apart from other major cassava producers. Thailand's cassava is mainly destined for animal feeds, starch, food components as well as different types of industry. Thailand ranks first among Asian countries and around equal third largest producer after Nigeria and Brazil (Falvey, 2000). Thailand's benefits from this crop are linked to the past four decades' export orientation policies. The three main forms of exports cassava are shredded, pellets, and starch, while the other forms such as fresh roots and manioc waste take minor portions. According to OAE (2009), the export value including its products accounts for 1.8% of agricultural sector GDP. Only 33% of the total cassava production is consumed domestically, i.e., 19% for starch, 8% for animal feeds and 6% for energy. In 2008, Thailand had 1.18 million ha of cassava planted area, yield a total production of 25.16 million tons which equivalent to a yield of 21.26 T/ha. The major region of planting has remained the Northeast followed by the East, while the Central and Northern areas take some smaller portions. The ability of cassava to continue to produce a harvestable yield under conditions of low soil fertility and variable rainfall was complemented by its ease of planting into new areas, flexibility in harvest time, ready marketability, and low requirements for management.

Cassava production received particular attention from Thailand's government during the outbreak of oil price crisis. Since it is of most economically suitable raw material for ethanol production, the cassava has therefore become increasingly important as an energy crop. In addition, cassava starch is currently utilized to produce bio-plastic which has been extensively promoting inside the country as well.

2. Pests and diseases in cassava

2.1 Arthropod pests in cassava

Arthropod damage to cassava is often indirect because most pests are foliage or stem feeders, reducing leaf area, leaf area, leaf life or stem feeders, reducing leaf area, leaf life or photosynthetic rate. Field studies indicate that pests that attack the crop over prolonged period (3-6 months) such as mites, mealybugs, thrips, whiteflies and lacebugs can cause severe root yield reductions. Bellotti (2002) has summarized major pests in cassava and their impacts on productivity. Mites are a universal pest of cassava causing serious yield loss up to 15% in resistant cultivars compared with 73% loss in susceptible cultivars, and 67% of the stem-cutting planting material was damaged. Mealybugs are of important pests causing considerable yield loss as a result of leaf yellowing, curling and cabbage-like malformation of the growing points. High densities lead to leaf necrosis, defoliation, stem distortion and shoot death. Reductions in photosynthetic rate, transpiration and mesophyll efficiency together with moderate increases in water pressure deficit, internal CO_2 and leaf temperature, were also found in infested plants. Whiteflies cause major damage in cassava as direct feeding pests and virus vectors. Whiteflies cause direct damage to cassava by feeding on the phloem of leaves, inducting leaf chlorosis and abscission, which results in considerable reduction in root yield if feeding is prolonged. Other species such as Lepidoptera, hornworms, stemborers burrower bug, lacebugs, and secondary pests are also of problematic pests in cassava that must be taken into consideration.

2.2 Virus, bacterial and fungal diseases in cassava

Calvert and Thresh (2002) has summarized the viruses and virus diseases of cassava. Cassava Mosaic Disease (CMD) is the only virus disease has been reported in Asia and the Pacific regions. There are two types of CMD that divided by symptoms; green mosaic and yellow mosaic. The outbreak of CMD in different areas indicated that CMD can causes cassava yield losses as high as 30–40%. For other

regions, especially in America and Africa, other virus diseases such as Frogskin disease and Brown Streak Disease may be found. Hillocks and Wydra (2002) stated that Cassava Bacterial Blight (CBB) is the most important bacterial disease of cassava. In Africa, it is second to CMD, 75% loss of yield was recorded in Nigeria while 100% loss was reported in Uganda.

For fungal disease, White Leaf Spot (WLS) is a fungal disease favored by cooler conditions. Little is known about the effect of this disease but it generally appears to be negligible, nevertheless, depending on cultivars. Diffuse Leaf Spot (DLS) is more prevalent in the warmer and wetter areas. The disease can causes severe defoliation on the more susceptible cultivars but this tends to occur on plants more than 6–9 months old and yield losses are believed to be slight. Ring Leaf Spot (RLS) is more common in Latin America. The attack of the pathogen to young shoot can cause dieback and death of the plant.

Brown Leaf Spot (BLS) is of worldwide distribution and occurs in most cassava fields in the lower canopy of crops more than 5 months old. Teri *et al*. (1977) reported that the disease appears to be confined to the foliage; the older leaves of the plant are more affected than the younger ones. The spots are found on both sides of the leaf. The symptoms are uniformly brown within a darker border on the lower surface. A brownish black ring was ranging from 0.3 to 0.5 mm in width. A yellowish halo may be found outside this ring on very susceptible varieties under extremely humid conditions. The BLS symptoms are leaf chlorosis and extensive defoliation which in turn results in yield loss up to 20%. Though the damage due to the BLS itself may be less significant, however, the BLS might induce the infection of other diseases. Wydra and Verdier (2002) found a significant positive correlation between the incidence of Cassava Anthracnose Disease (CAD) and the occurrence of BLS disease, as well as implications among the BLS, WLS, and root rots. In addition, the incidence of BLS is more prevalent in humid ecozones while its severity seemed to be increasing with number of surrounding trees and on profusely branching varieties. These clearly suggested that BLS disease should not be neglected particularly in the conditions of Thailand.

3. Assessment of the severity level for cassava BLS disease

Assessment of disease infection degree is crucially important to plant pathologists especially in the decision making to select a correct treatment for a certain epidemic incident. Scoring the disease level is also important for plant breeders in improving the disease tolerance of a plant species. The conventional methods for scoring a plant disease are to visually evaluate the symptoms (e.g. damage area, number and size of spots) of the plant leaves by comparing with an illustrated diagram developed for that specific disease. This is generally performed by well-trained pathologists. For cassava BLS disease, Teri *et al*. (1977) introduced an area diagram key shown in Figure 1 which classifies the infection into 4 levels based on density and distribution of lesions (number of spot and percentage of damage area per leaf). Degree of defoliation or leaf retention have also been used to rate disease severity.

Figure 1 Illustrated area diagram for cassava BLS disease assessment proposed by Teri *et al.*(1977).

Image analysis has been widely applied to assess the severity of plant diseases as an alternative for plant pathologists to score and manage the disease. The method has been considered faster and more precise and accurate than manual assessment. Bock *et al.* (2008) studied the reproducibility of image analysis for measuring the severity of canker symptoms on grapefruit leaves in comparison with visual assessments done by three visual raters for various types of symptoms. The study indicated that image analysis was more precise than visual raters for all types of symptom judging from inter-assessment correlation coefficients. Wijekoon *et al.* (2008) used Scion Image software to measure the severity of several foliar fungal diseases. A good correlation between percent diseased area obtained from Scion Image software and the percent diseased area derived from the diagram key was observed. Bock *et al.* (2009) also reported that using automatic image analysis to assess the severity of foliar citrus canker symptoms showed consistent results with visual raters.

4. Application of machine vision in plant protection and growth monitoring

Machine vision systems based on different image analysis techniques have been applied to various tasks in agriculture. Sermsak and Boonjung (2007) adopted image analysis to estimated Leaf Area Index (LAI) of soybean. Jarimopas and Jaisin (2008) developed a machine vision system for sorting sweet tamarind while Kurtulmaus *et al.* (2011) used image analysis technique to distinguish green citrus fruits in natural canopies in the natural daylight illumination. Furthermore, image processing is potential for non-destructive evaluation of fruits maturity such as pomelo (Aroonyadet, 2002) and durian (Thammabhutra, 2005).

The surveillance operation of weed and disease based on human inspection is very difficult in large fields, especially when precision management is required. Automated monitoring system based on machine vision has been considered a potential approach for detecting weeds and diseases in early stage of plant growth which in turn allows traceability of spatial and temporal progression of those weeds and diseases.

A number of machine vision systems relying on different image processing techniques have been developed for different tasks in agriculture. For weeds detection, Woebbecked *et al.* (1995) reported a success of using color indices for weed identification under various soils, residue, and lighting conditions. Pérez *et al.* (2000) used color and shape analysis techniques for discriminating crop, weeds and soil. Yang *et al.* (2002) developed an image processing component of weed detection and mapping system based on intensity of green color. Hemming and Rath (2001) successfully classified plant and weed using color indices. Aitkenhead *et al.* (2003a) applied image analysis for discriminating carrot seedlings from two species of weeds based on morphological characteristic of leaf shape while Onyango et al. (2003) developed segmentation algorithm for separated plants from weed. Hague *et al.* (2006) reported that the use of an automated crop and weed monitoring in widely spaced cereals showed good agreement with manual assessment. Meyer *et al.* (2008) verified color vegetation indices for automatic threshold using plant-soil-residue images and Burgos-Artizzu *et al.* (2011) discriminated between weeds, crops and soil real-time by machine vision system.

In the diagnosis of crop pests and disease, Sena *et al.* (2003) used computer vision to identify fall armyworm damaged maize plants. Camargo and Smith (2009a) developed a threshold algorithm for disease detection while Camargo and Smith (2009b) used machine vision system to identified diseases on cotton. Likewise, Cui *et al*. (2010) used image processing methods based on multispectral image for quantitatively detecting soybean rust. Weizheng *et al.* (2008) proposed an image processing method for grading plant disease. Numerous researches on image analysis for detecting plant pests and diseases have been carried out. However, application on cassava crop, particularly for BLS disease, has been rarely found. To the best of our knowledge, Aduwo *et al.* (2010) who developed an automated vision system for diagnosing cassava mosaic disease is the only application on cassava.

5. Fundamentals of image analysis

5.1 Basic image processing procedure

An image pixel is defined as a two-dimensional function, $f(x, y)$, where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or grey level of the image at the point. (Gonzalez and Wood, 2010)

$$f(x, y) = i(x, y)r(x, y) \quad (1)$$

where, $0 < i(x, y) < \infty$ and $0 < r(x, y) < 1$

$i(x, y)$ is illumination component and $r(x, y)$ is reflectance component

Fundamental steps in digital image processing consists of five steps as follows:

5.1.1 Image acquisition is the process to capture or transform a picture of a target object into digital format.

5.1.2 Image enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application.

5.1.3 Image segmentation is process of partition an image into its constituent parts or objects.

5.1.4 Image representation and description is the process to interpret each constituent in the image into a form of parameters suitable for further computer processing. A region can be represented either in terms of its external characteristics (its boundary) or its internal characteristics (the pixels comprising the region)

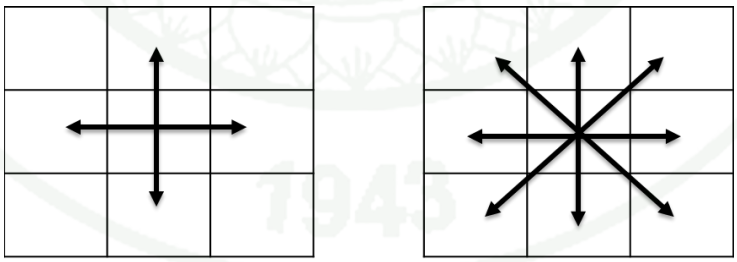
5.1.5 Recognition is the process to assigns a label to an object based on its descriptors.

5.2 Image enhancement

Image enhancement is the process to render an image so as it more visually acceptable or pleasing (Solomon and Breckon, 2011). The noise removal, image sharpening, and image blurring are popular enhancement procedures which can be done by pixel connectivity, opening operation, median and Gaussian filtering techniques, and etc.

5.2.1 Pixel connectivity

Pixel connectivity is used in image enhancement to indicate the region of interest which it is a morphological processing base on the connectivity of the pixels (Motameni *et al*., 2007). An object is considered to be derived from a group of pixels connection. There are two type of connectivity, namely, 4-connected neighborhood and 8-connected neighborhood as shown in Figure 2. In the 4-connected neighborhood, pixels are connected if their edges touch. A pair of adjoining pixels is part of the same object only if they are both on, and are connected along the horizontal or vertical direction. In the 8-connected neighborhood, the connection is three-directional, i.e. horizontal, vertical, and diagonal directions.



(a) (b)

Figure 2 Pixel connectivity, (a) 4-connected neighborhood and (b) 8-connected neighborhood.

5.2.2 Opening operation

Opening operation is morphology processing that smoothens the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. It is used to eliminate noise in binary image. The operation was obtained by erosion operation followed by dilation operation. Thus, the opening of set *A* by structuring element *B*, denoted by $A \circ B$, is defined as:

$$A \circ B = (A \ominus B) \oplus B \tag{2}$$

Thus, the opening *A* by *B* is the erosion of *A* by *B*, followed by a dilation of the result by B (Gonzalez and Woods, 2010).

5.2.3 Median and Gaussian filtering

Both Median and Gaussian filtering are an order in part of image enhancement used to eliminate noise and non-necessary details. The main idea of the Median filter is to run through the image pixel by pixel. The output image is then produced by replacing the median value for every pixel. Thus, Median filtering requires an ordering of pixel connectively. In Gaussian filtering, the algorithm can be described by the following equation:

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right) \tag{3}$$

where, *g(x, y)* is a Gaussian function, *x* and *y* are distance from the origin in the horizontal and vertical axis respectively, σ is the standard deviation of the Gaussian distribution. A Gaussian function with a large value of σ is called low-pass filter. Its commonly forms the first stage of an edge-detection algorithm (Solomon and Breckon, 2011).

5.3 Image segmentation

In image segmentation, color is often used as an index. Typically, the segmentation is performed by determining a threshold value which is capable of differentiating the interesting object from undesirable area (e.g. background) using color indices in RGB or HSI space. The segmented image is called regions of interest (ROI).

5.3.1 Color model and segmentation

Color is an important feature of an image that provides information for humans to recognize an object which can be illuminated under a wide range of conditions. There are many models used to quantify a certain color. The purpose of a color model is to facilitate the specification of colors in some standard. (Gozalez and Woods, 2010).

In RGB color model, each color represents spectral components of red, green, and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube shown in Figure 3. Points along the main diagonal are grey values, from black at the origin (0, 0, 0) to white at point (1, 1, 1). In this model, the data class of the component images determines their range of values. If an RGB image is of class double, the range of values is [0, 1]. Similarly, the range of values is [0, 255] or [0, 65535] for RGB images of class unit 8 or unit 16, respectively.

The HSI space is represented by a vertical intensity axis and the locus of color points that lie on circular planes perpendicular to the axis shown in Figure 4. The hue (H) value is determined by an angle from the reference point. The saturation (S) is the length of the vector (radius) from the origin to the color point and it describes the pureness of color. I represents the amount of intensity.

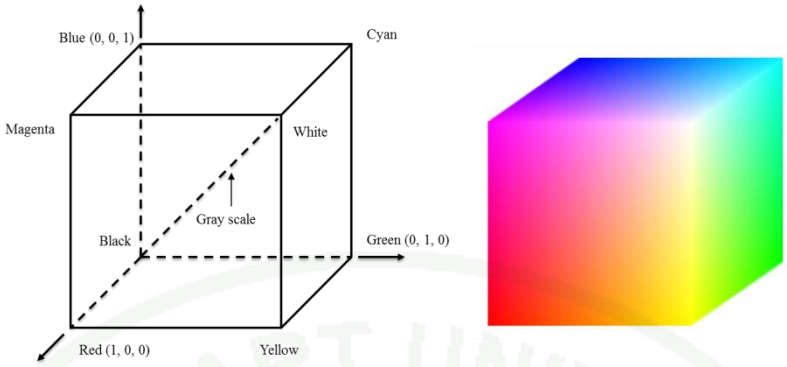


Figure 3 RGB color space.

Source: Gozalez and Woods (2010)

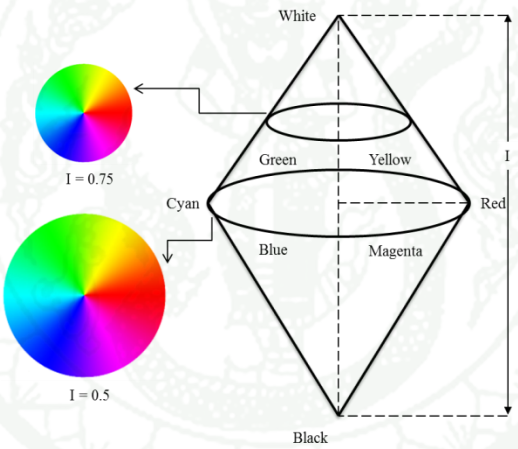


Figure 4 HSI color space.

Source: Gozalez and Woods (2010)

The color segmentation has been applied to various tasks in agriculture, Patil and Bodhe (2011) used color segmentation to identify disease severity of brown spot infection on sugarcane leaves. Sannakki *et al.* (2008) proposed rating method of leaf spot disease based on image processing using color segmentation while Gulhane and Gurijar (2011) studied the feasibility of disease detection on cotton leaves by color recognition with neural network.

5.3.2 Otsu's segmentation method

There are many methods for image segmentation. Of these, Otsu's method is a most popular method to find out a threshold value for segmentation in grey image. It automatically performs thresholding from histogram of the grey scale image and in turn reducing into a binary image. The algorithm assumes that the image to be thresholded contains two classes of pixels or bi-modal histogram (e.g. foreground and background) then calculates the optimum threshold separating those two classes.

6. Artificial neural networks for image classification

In machine vision system, an artificial neural network (ANN) is a method to recognize object in the image after segmentation process. It is nonlinear information processing algorithm which are built from interconnected elementary processing units called neurons. The working of the ANN mimics the biological neurons in which the information is processed in parallel. The information inputs are transmitted to hidden neurons through multiplication with connection weights. These multiplication will be summed and activate by a transfer function to the subsequent layers of neurons. With an appointed algorithm of weight adjustment called learning rule, the ANN would learn the functional relationships underlying the inputs and outputs and get prompt to further perform prediction from unseen information.

The ANN has been applied to various tasks in agriculture including image analysis, Yang *et al.* (2000) used ANN in image recognition and classification of crop and weed base on color features. El-Helly *et al.* (2003) developed image processing system for automatic inspection of leaf batches and identify disease type by using feed-forward neural network with two hidden layers and the standard back propagation rule as the training algorithm and Kiani and Jafari (2012) used shape features of digital image for crop detection and positioning in the field by using discriminant analysis and neural networks.

7. Discriminant analysis

Discriminant analysis (DA) is used to discriminate values of a variable into two or more groups. Given a set of independent variables, DA attempts to find linear combinations of those variables that best separate the groups of cases. These combinations are called discriminant functions expressed in the form of equations.

The objectives of DA include: first, classifying data into groups using a discriminant prediction equation, second, examining dependency among variables and, third, assessing the relative importance of the independent variables. The DA can be implemented according to following procedure:

7.1 Identifying variables that best discriminates among the various groups.

7.2 Use identified variables to develop an equation or a function for computing a new variable or index that will parsimoniously represent the difference between various groups.

7.3 Use the discriminant function to classify future observations into any of the pre-defined groups.

Linear discriminant function equation is generally in the following form:

$$Z = W_1X_1 + W_2X_2 + \ldots + W_nX_n \quad (4)$$

where,

Z = Discriminant score

W_i = Discriminant weight associated with the i^{th} independent variable

X_i = i^{th} independent variable

The DA has been applied to various tasks in agriculture including image analysis. Koger *et al.* (2003) studied the potential of multispectral imagery for weed-infested and weed-free soybean discrimination by using linear discriminant analysis

(LDA). Gulhane and Gurjar (2012) identified and diagnosed cotton disease pattern by using image analysis with LDA technique.

MATERIALS AND METHODS

Materials

1. Digital camera (Canon, IXY55, Japan)
2. Personal computer (Sony, VAIO E-series, Core i3-350M, 2.26 GHz)
3. MATLAB software (The MathWorks Inc., MA, USA)
4. SPSS software
5. Digital light lux meter
6. Illumination box
7. Healthy and BLS-infected cassava plants

Methods

1. Preliminary experiment on classification of healthy and diseased leaves under controlled background condition

The objective of this experiment was to evaluate the effectiveness of some color features in discrimination between healthy and diseased cassava leaves. Parameters derived from RGB values were used in this experiment.

1.1 Image acquisition

The experiment was carried out by capturing color images of healthy and disease cassava leaves with a digital camera at a horizontal by vertical resolution of 1600 by 1200 pixels under natural lighting condition. In this experiment, the background color of an image was controlled using a white paper. This is in order to exclude segmentation process.

1.2 Selection and representation for areas of interest

The original color images were cropped to a block size of 80×80 pixels. This results in six possible categories of coverage namely; background only (B), leaf only (L), background and leaf (B-L), background with spot (B-S), leaf with spot (L-S), and background with leaf and spot (B-L-S). Since the lighting condition may be varied, the background was also included as a reference. Means and standard deviations of red, green and blue values were calculated for each box using Image Processing Toolbox MATLAB®.

1.3 Neural network models for image classification

Two models of feed-forward neural networks were developed to identify disease region using means and standard deviation of RGB as inputs. The Model I had six output neurons representing six coverage categories as shown in Figure 5. The Model II was designed to have two outputs neurons which first output neuron contained diseased region (B-S, L-S, and B-L-S) while the second output neuron contained non-diseased regions (B, L, and B-L) as shown in Figure 6.

For both models, a logistic sigmoid activation function was used in hidden neurons while a linear function was used in the output neurons. The logistic sigmoid function accepts input ranging from -∞ to ∞ and compressed the output within a range of 0 to 1. Activations of the logistic sigmoid function and of the linear function can be described in Eq. 5 and 6 respectively.

$$a = \log sig(n) = \frac{1}{1+e^{-n}} \tag{5}$$

$$a = n = \sum_{i=1}^{n} w_i p_i + b \tag{6}$$

where a is the activation, n represents the net input, w is the weight value, and b is the bias value.

The data set consisted of 10 images for each category, giving a total of 60 images, 36 of which were used in training while the rest 24 images were used for validation. All data was normalized into a range of 0.1–0.9 using Eq. 7.

$$X_{normalized} = 0.1 + \frac{0.8(X_{actual} - X_{\min})}{(X_{\max} - X_{\min})} \tag{7}$$

The target outputs were set to a value of 0.9 in the output neuron that corresponds to its category while the remaining output neurons were assigned a target value of 0.1.

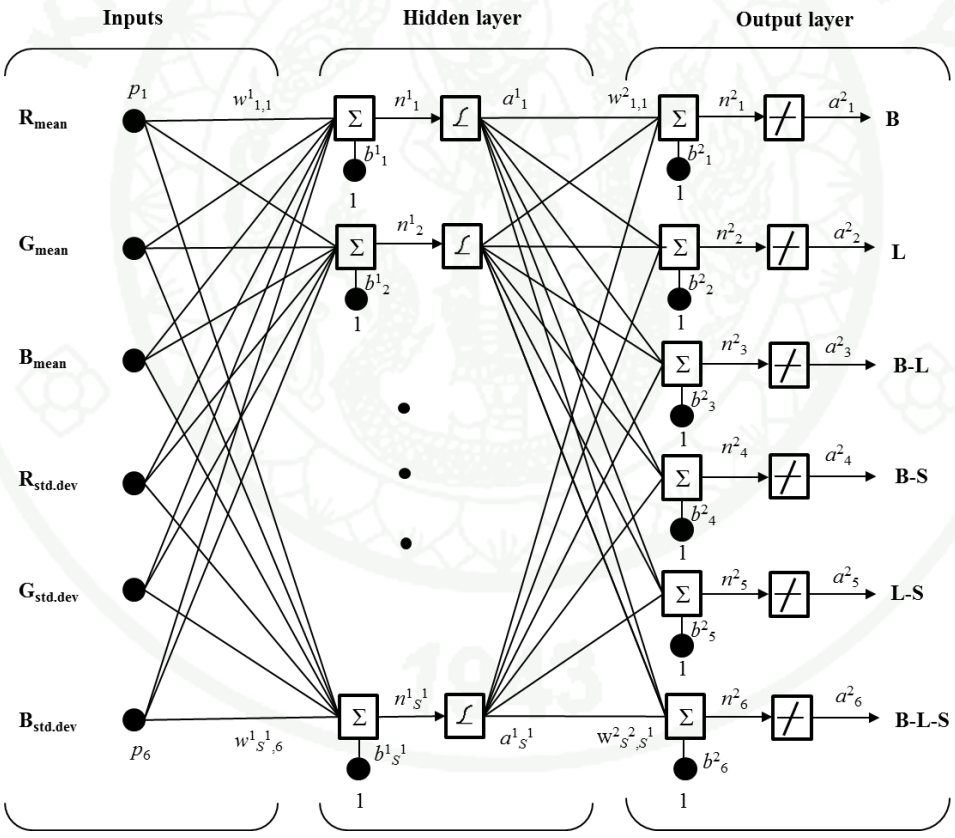


Figure 5 ANN architecture for image classification (Model I).

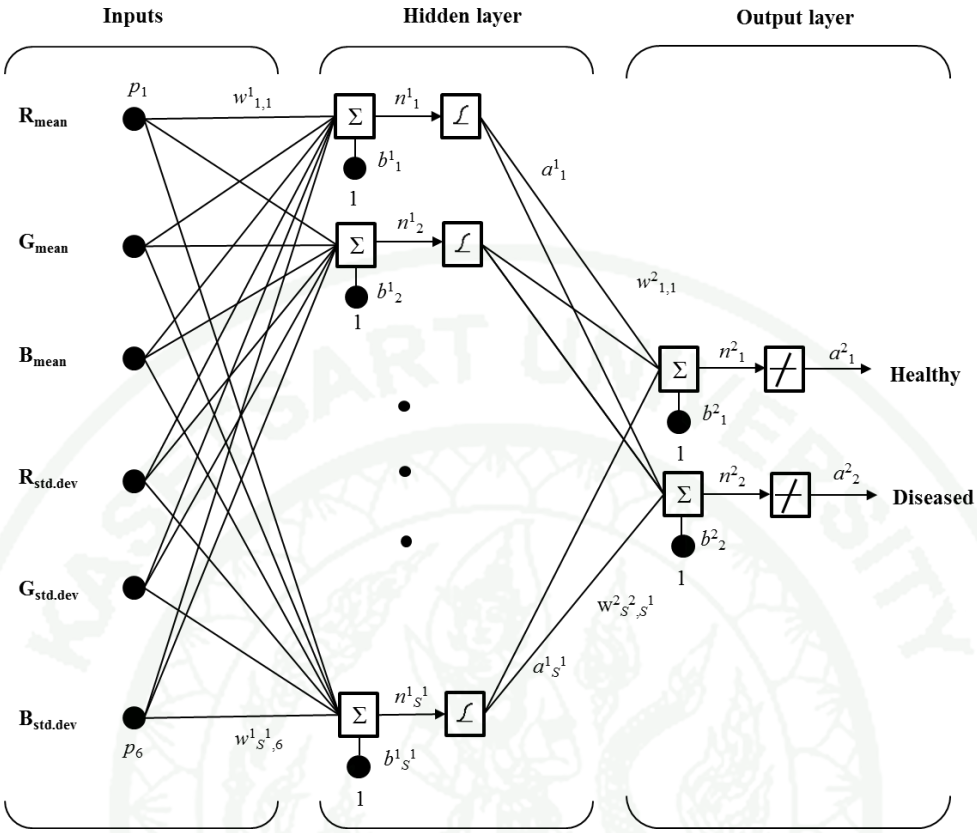


Figure 6 ANN architecture for image classification (Model II).

Training of the ANNs was performed with Levenberg-Marquardt algorithm by setting the goal mean squared error of 1.0×10^{-4} using Neural Network Toolbox for MATLAB®. The trial-and-error method was used to determine the most appropriate number of hidden neurons. In both models, the number of hidden neurons was arbitrarily selected by varying its number for every 5 nodes increment from 10 to 30 nodes. Classification accuracy was observed and evaluated for each ANN model.

2. Development of image analysis techniques for *in situ* detection of diseased cassava plants

The objective of this experiment was to develop and image analysis technique which is capable of recognizing visible symptoms of cassava BLS disease in real field conditions.

2.1 Experimental site and plant materials

Image of cassava leaves were sampled from an experimental field located in Kamphaengsaen Campus of Kasetsart University, Nakhon Pathom, Thailand (Lat 14°2′11″N and Long 99°57′56″E). The variety of cassava plants was Rayong 5 which is of medium cultivar in terms of BLS disease tolerance. The age of the plants at sampling was 6 months at which time their canopy had fully developed. The BLS-infected plants were found scattering naturally throughout the field without systematic inoculation treatment.

2.2 Image aquisition, processing, and classification

Color images of cassava leaves were captured at several randomly chosen positions above the plants using a digital camera. The camera was set to operate in the automatic mode with a resolution of 640×480 pixels under natural sunlight at solar noon. The image set consisted of 80 images of healthy leaves, and 80 images that contain diseased leaves.

The Image Processing Toolbox™ for MATLAB® was used to pre-process and analyze the captured images. The original image were automatically cropped into 48 blocks of 80×80 pixels each. An image therefore presented 6 rows of blocks with 8 blocks on each row. The images which include diseased leaves when cropped into blocks, however, would result in both non-diseased (e.g. healthy, stems) and truely diseased regions (Figure 7). Counting of these blocks gave a total number of 5270 blocks for healthy, and 2410 blocks for diseased leaves. The primary red, green, and blue (RGB) color intensities of each single pixel were then obtained for futher transformation to other indices.

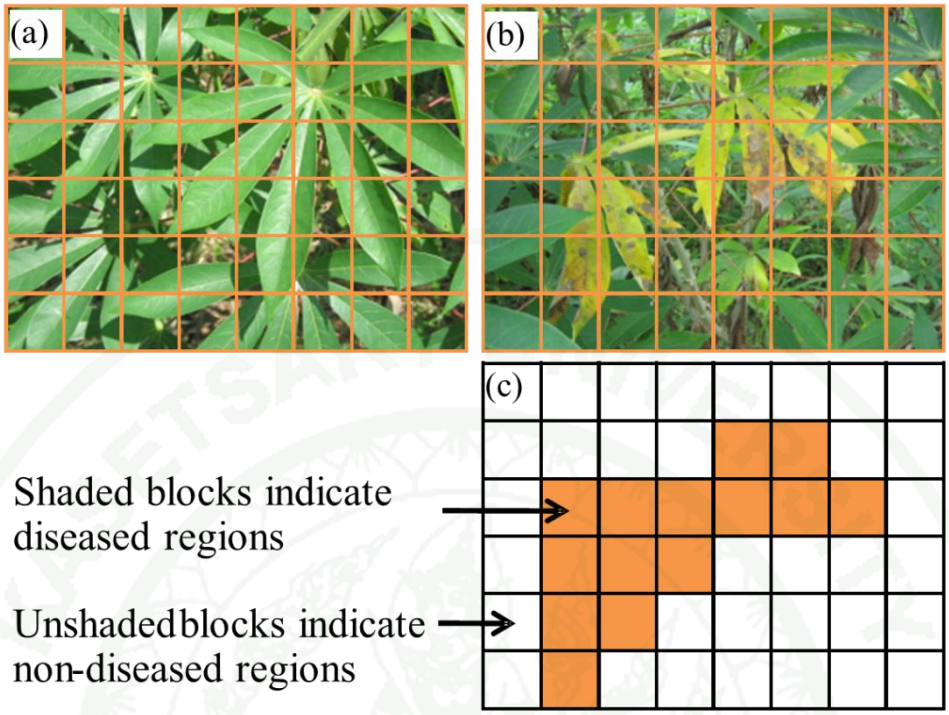


Figure 7 Sample images of: (a) healthy leaves, (b) diseased leaves, and (c) diseased regions mapping.

Different color indices introduced by Woebbecke et al. (1995a) were used in this study. These included chromatic coordinates (*r*, *g*, and *b*) which are defined as:

$$r = \frac{R^*}{R^* + G^* + B^*} \ , \ g = \frac{G^*}{R^* + G^* + B^*} , \ b = \frac{B^*}{R^* + G^* + B^*} \tag{8}$$

where

$$R^* = \frac{R}{R_m}, G^* = \frac{G}{G_m}, \ B^* = \frac{B}{B_m} \tag{9}$$

and R_m, G_m and B_m = 255, are the maximum tonal value for each primary color.

Contrast indices including $r - g$, $g - b$, $(g - b)/|r - g|$ and $2g - r - b$ were obtained to distinguish leaves and plant parts of different color. Preliminary test

indicated the possibility of obtaining a zero in the denominator of the index $(g-b)/|r-g|$, denominator values between –0.01 and +0.01 were set to 0.01.

The hue, saturation, and intensity (HSI) color space was also used in addition to the chromatic coordinates. Modifed hue (*H*), saturation (*S*) and intensity (I) are derived from RGB values as follows:

$$H = \begin{cases} \theta & if\ B \le G \\ 360 - \theta & if\ B > G \end{cases} \tag{10}$$

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right] \tag{11}$$

$$I = \frac{1}{3}(R+G+B) \tag{12}$$

where, is θ an intermediate variable and can be calculated using the following equation

$$\theta = \cos\left\{\frac{\frac{1}{2}\left[(R-G)-(R+B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{0.5}}\right\} \tag{13}$$

Mean value of each color index was calculated across the total 640 pixels (80×80) for each block and used as a representive value for further classification.

2.3 Image classification with neural network and discriminant analysis

In this experiment, ANN and DA techniques were used to recognize and seperate the healthy and diseased leaves. Classification accuracy of both approaches was observed and compared. The ANN model was a fully-connected feed-forward topology consisting of 10 inputs coresponding to ten color indices, and 2 output

neurons representing healthy and diseased leaves (Figure 8). In order to search for the most optimal architecture, the classification accuracy was observed when the number of hidden neurons was varied from 10 to 100 for every increment of 10 neurons.

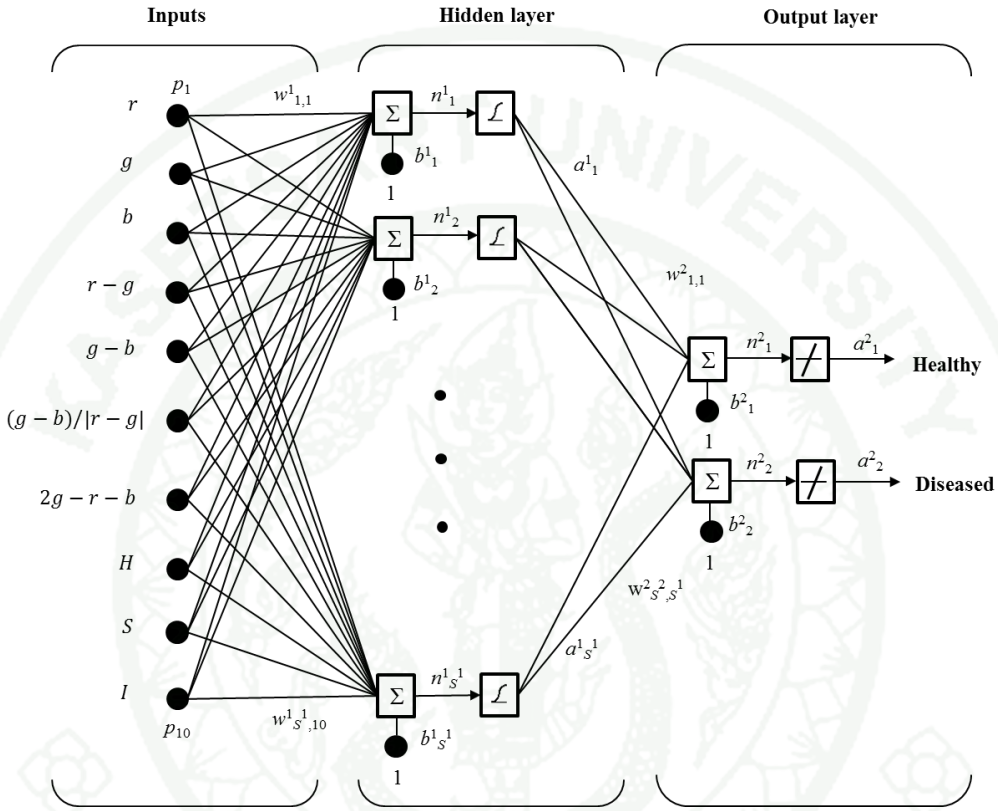


Figure 8 The ANN architecture for classification of healthy and diseased leaves.

A logistic sigmoid activation function was used in hidden neurons while a linear function was used in the output nodes. From the total 5270 samples of healthy plants, 3247 of which were randomly selected for training of the ANN while the rest were used for testing. Likewise, 1553 out of 2410 samples of diseased plants were used for training and the rest were spared for verification. All data was normalized into a range of 0.1–0.9. The target outputs were set to a value of 0.9 for the output node that corresponds to its category while the remaining was assigned to activate a value of 0.1. Training of the ANN was implemented by Levenberg-Marquardt algorithm to the mean squared error of 1.0×10^{-4} using Neural Network Toolbox™ for MATLAB®.

In addition to the ANN approach, DA was also used for image classification. Preparation of the data set for DA was done in the same way as did for the ANN. All color indices were used as independent variables for generating linear equations or, in other words, classification functions. With the DA approach, dependency among indices could be analyzed. The DA was implemented using SPSS software according to following procedure.

First, multicollinearity among color indices were examined by Pearson's Product Moment Correlation, tolerance (T), and Variance Inflation Factor (VIF) which can be calculated by Eq.14, 15, and 16 respectively.

Pearson's Product Moment Correlation

$$r_{xy} = \frac{N\sum XY - \sum X \sum Y}{\sqrt{\left(\left(N\sum X^2 - \left(\sum X\right)^2\right)\left(N\sum Y^2 - \left(\sum Y\right)^2\right)\right)}} \quad (14)$$

Tolerance (T)

$$\mathrm{T} = 1 - R_i^2 \quad (15)$$

Variance Inflation Factor (VIF)

$$\mathrm{VIF} = \frac{1}{1 - R_i^2} \quad (16)$$

where r_{xy} is Pearson's correlation coefficient, N represents the number of data pairing, $\sum X$ and $\sum Y$ are the sum of each color indices, and R_i^2 is the coefficient of determination for each pair of color indices.

In image classification, color indices are expected to be independent against each other since dependency among parameters themselves might affects the

accuracy of classification. However, if multicollinearity was found, subsequent procedure to group similar parameters should be applied. In this study, Factor Analysis method was used. Color indices of similar behavior would be grouped and represented by a new factor component. Image classification by DA method was then carried out based on these factor components.

Similar to the ANN technique, the data was separated into two sets i.e. calibration and validation sets. The calibration data consisted of 3247 healthy and 1553 diseased samples. Once the classification functions have been created, the validation data set which contained a number of samples of 2023 for healthy and 857 were tested by prediction equation from classification function and groups classify assessment.

2.4 Comparison of classification performance

Brier score was used to evaluate the recognition ability of the ANN with different numbers of hidden neurons and to compare the classification performance between the ANN and the DA approaches. Brier score is a unitless index and its range is from 0 to 1. The smaller the score is, the better is the recognition ability of the ANN. The Brier score cnan be calculated from Eq. 17.

$$BR = \frac{1}{N}\sum_{i=1}^{n}(f_t - o_t)^2 \qquad (17)$$

where BR is the Brier score, N is the number of predicting instances, f_t is the probability that was prediction and o_t is the actual outcome of the group classification at instance t (0 if it was not correct classification and 1 if it was correct classification)

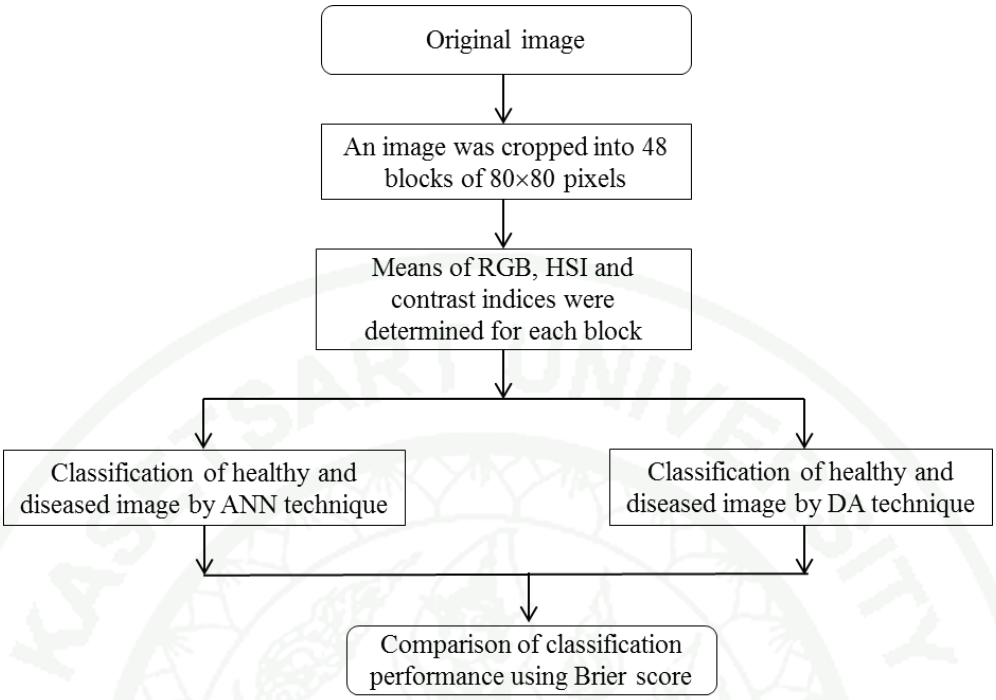


Figure 9 Conceptual scheme of image analysis for *in situ* detection of diseased cassava plants.

3. Development of image analysis algorithms for precision positioning of disease spots on cassava leaves

This experiment aimed at developing an image processing algorithm which is capable of identifying the exact positions of infected areas, i.e. spots, on leaves. The algorithm must therefore be able to discriminate the lesion regions from healthy part of a leaf and other plant parts such as stems, residues, as well as background (e.g. soil). In this experiment, three image processing algorithms (AG) were developed and compared.

3.1 Data preparation

Fifty images of diseased leaves and another fifty images of healthy leaves were chosen from the image set described in Section 2 for this part of study. The

images out of selection were the images on which the disease symptoms appeared unobvious.

3.2 Algorithm I

The first algorithm (AG I) was designed to process am image based on HSI color space. The algorithm was programmed to read the images automatically which allowed the analysis of all image in one time. In the first step, each image was blurred by Gaussian smoothed filter at 5×5 with an approximated discrete value (σ) of 10. The image was then transformed from RGB color space to HSI as described earlier by Eq. 10 – 13. The *H* image of HSI color space was used in segmentation.

After segmentation, the brown spots on an image would standout clearly, however, there might exist some holes on each spot if the damage is severe. The effect of these holes was eliminated by filling the void area of the holes using "imfill" command of the Image Processing Toolbox for MATLAB. In addition, the segmented image often contain noise, the noise reduction process was therefore performed. This was done by opening operation (Gonzalez and Woods, 2010) using disk-shaped structuring element with a radius of five pixels. Second, the size of a spot was specified ranging from 100 to 800 pixels. Thus, an object of size smaller or larger than the range would be recognized as non-spot object. Further, in order to eliminate non-spot objects such as petiole or stem, the shape factor described by aspect ratio between major and minor axis of each spot was examined. The ratio of greater than 3 was considered a non-spot and hence eliminated. After noise reduction, each spot was combined by pixel connectivity in 2D binary image which is an 8-connected object technique. The edge detection was performed on every spot and drawn on original image. The number of spot detected and the number of actual spots were counted and compared.

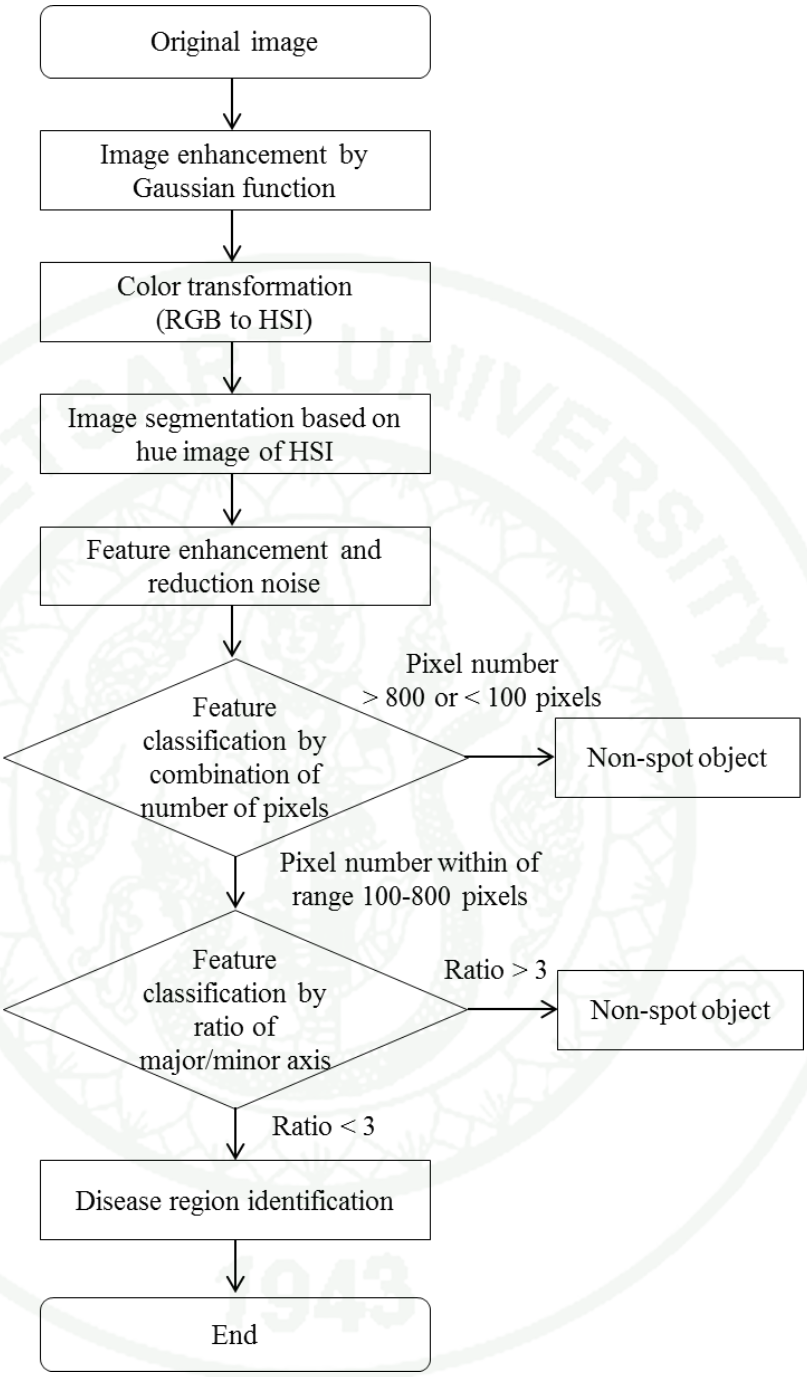


Figure 10 Flowchart of Algorithm I.

3.3 Algorithm II

The second algorithm (AG II) was designed based on the contrast of colors. Two indices, namely, Excess Green (*ExG*) and Excess Red (*ExR)* introduced by Meyer and Neto (2008) were used as parameters for classification. *ExG* and *ExR* can be defined as:

$$EXG = 2g - r - b \tag{18}$$

$$EXR = 1.4r - g \tag{19}$$

The subtraction between of *ExG* and *ExR* was performed for each image. The *ExG – ExR* image was generated and enhanced by 5×5 median filter. Color segmentation was then conducted by following equation:

$$g(x,y) = \begin{cases} 0 if\ f(x,y) > T \\ 1 if\ f(x,y) \leq T \end{cases} \tag{20}$$

where *g(x,y)* is the segmented image, *f(x,y)* is the gray-level histogram at pixel *(x,y)* , and *T* is the threshold value.

Similar to AG I, after segmentation, the brown spots on an image might contain some holes in case of severe damage. These holes must be eliminated by filling the void area using "imfill" command of the Image Processing Toolbox for MATLAB. Likewise, the segmented image which contained noise would be processed to reduce the noise as done for AG I. The spot size was specified within a range between 100 to 800 pixels, and hence an object of size smaller or larger than the range would be recognized as non-spot object. The aspect ratio of greater than 3 which describe a non-spot object was applied similarly to the AG I. Similarly, the edge detection was performed on every spot and drawn on original image. The number of spot detected and the number of actual spots were counted and compared.

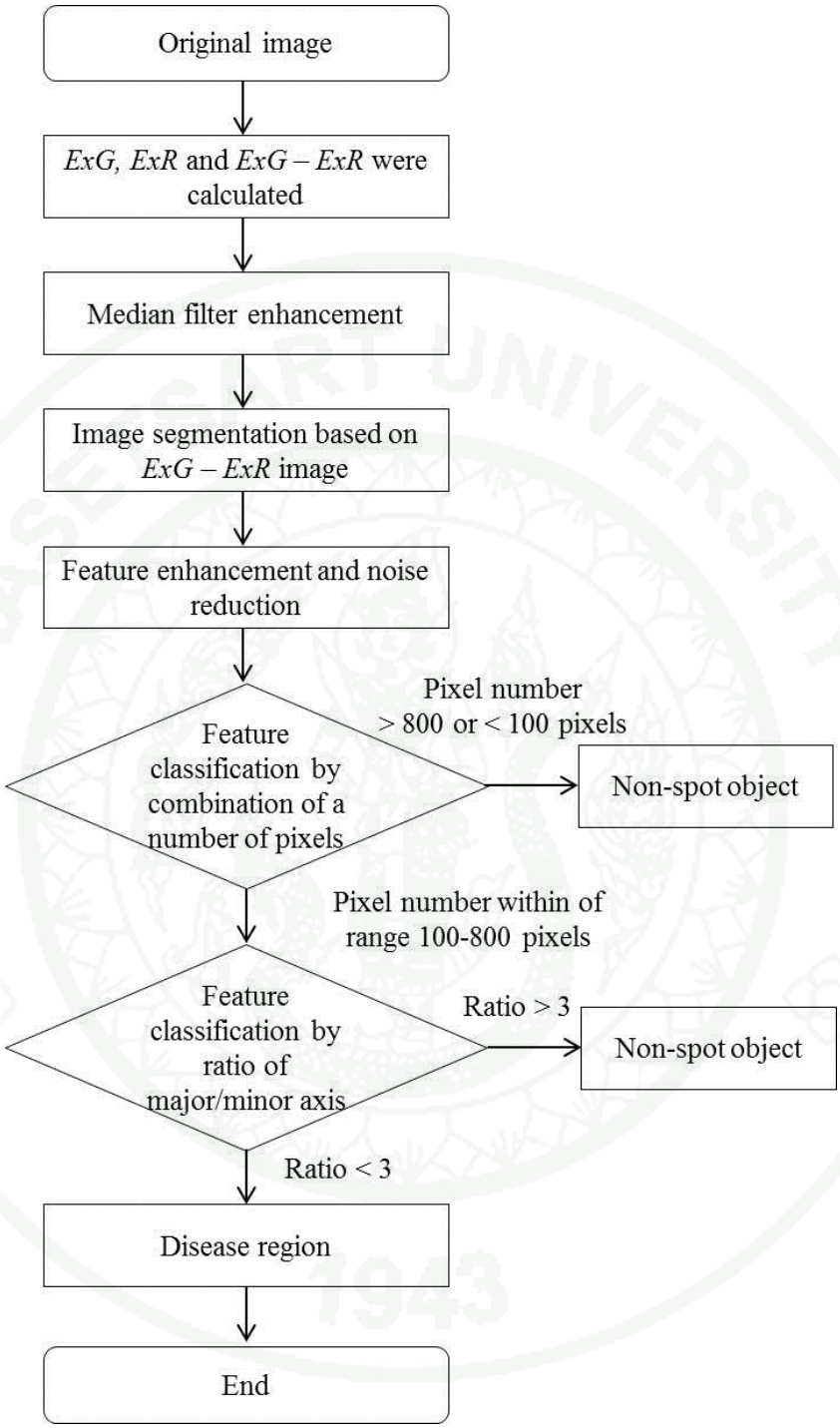


Figure 11 Flowchart of Algorithm II.

3.4 Algorithm III

According to the preliminary test upon developing AG I and AG II, it was observed that the *H* image was highly potential to standout the brown spots, whereas the *ExG* image was superior in detecting the leaves from undesirable plant parts background. The third algorithm (AG III) was therefore developed by taking these advantages of *H* and *ExG* images. For more specifically, the intersection between, $H \cap ExG$ would best locate the positions of disease spots while other objects are eliminated.

Generation of *H* image was done in the same way of AG I. In the part of leaves region extraction, the *ExG* value was determined and further processed by blur enhancement using Gaussian smoothed filter at 5×5 with an approximated discrete value, σ, of 10. Otsu's method (Otsu, 1979) which is a segmentation method based on histogram of the tonal image was used. This enabled to threshold an extract the leaves region on an image. The "imfill" command was then applied again to fill the holes inside the leaves boundary. The *H* and *ExG* images were then intersected. The aspect ratio of greater than 3 which describe a non-spot object was applied similarly to the AG I and II. With this concept the brown spots on leaves could be segmented. In the same way, the edge detection was performed on every spot and drawn on original image. The number of spot detected and the number of actual spots were counted and compared.

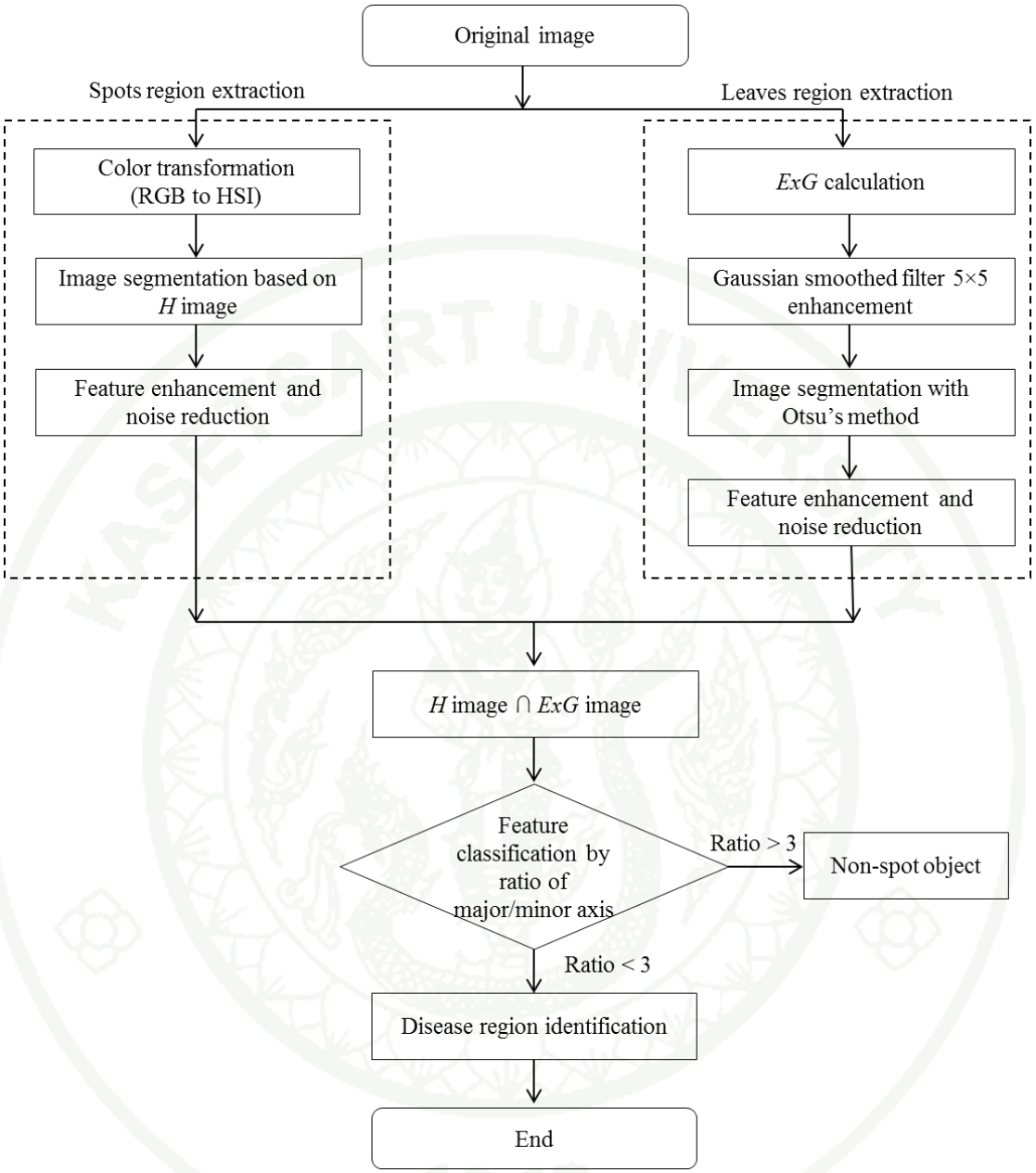


Figure 12 Flowchart of Algorithm III.

3.5 Evaluation of classification performance and comparison between algorithms

Statistical tests were conducted to evaluate the classification performance of the three algorithms, implemented with SPSS software. The paired sample correlations between actual number of spots and detected number of spots given by developed algorithms were observed. The Root Mean Squared Error (*RMSE*) determined by Eq. (21) was also observed.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(N_{actual} - N_{image}\right)^2}{n-1}} \quad (21)$$

where, N_{actual} is the actual number of spots obtained by manual counting, N_{image} is the number of spot detected by the image analysis, and n is the number of sample image.

4. Development of image processing algorithm for the assessment of diseases infection severity

The objective of this part was to develop an image analysis procedure for identifying the damage level of the BLS disease. In this experiment, an image analysis technique based on *H* color index was developed. The study also aimed to compare the assessment result with the use of conventional method which relies on illustrated diagram key. Comparison of the severity assessment result between manual scoring and the image analysis was carried out based on percentage of infection area and number of spots.

4.1 Cassava leaf samples

The cassava leaves were collected from experimental field located in Kasetsart University at Kamphaengsaen Campus. The variety of cassava plants was Rayong 5 and the age of the plants at sampling was 6 months. The BLS-infected

plants were found scattering naturally throughout the field without systematic inoculation treatment.

4.2 Assessment by conventional diagram key

Teri (1977) has proposed a diagram key for assessing the damage due to BLS disease in cassava which categorized the infection into four levels as illustrated in Figure 13. Seven indipendent raters were assigned to score the severity level using Teri's diagram. Consistency of scoring among different raters were observed and to be compared with the results obtainable from image analysis.

(a) (b) (c) (d)

Figure 13 Teri's diagram key for cassava brown leaf spot disease assessment: (a) 5%, (b) 10%, (c) 15% and (d) 20%.

4.3 Assessment by image analysis

The image processing developed consisted of three main parts, i.e., image acquisition, image enhancement, and image segmentation associated with feature extraction.

4.3.1 Image acquisition

Image of each cassava leaf was captured using a digital camera (Canon, IXY55, Japan) at a resolution of 1600 by 1200 pixels. The image acquisition

was done in an illumination control box which comprised of four 18W day lights D65 (Color temperature 6500K) light bulbs installed at the top corners of the box, oriented 45° to the camera centerline (Figure 14). Measurement of light intensity inside the box with a light lux meter indicated an intensity level of 3428 lux. The inner side of the illumination box was lined with black canvas and a white paper was used as the object background. The camera was positioned on the top of the illumination box at a distance of 50 cm above and perpendicular to the object base. The focal length of the camera was set to 8 mm with a constant f-stop of 3.5. The images were acquired in JPEG format with a total number of 48 images.

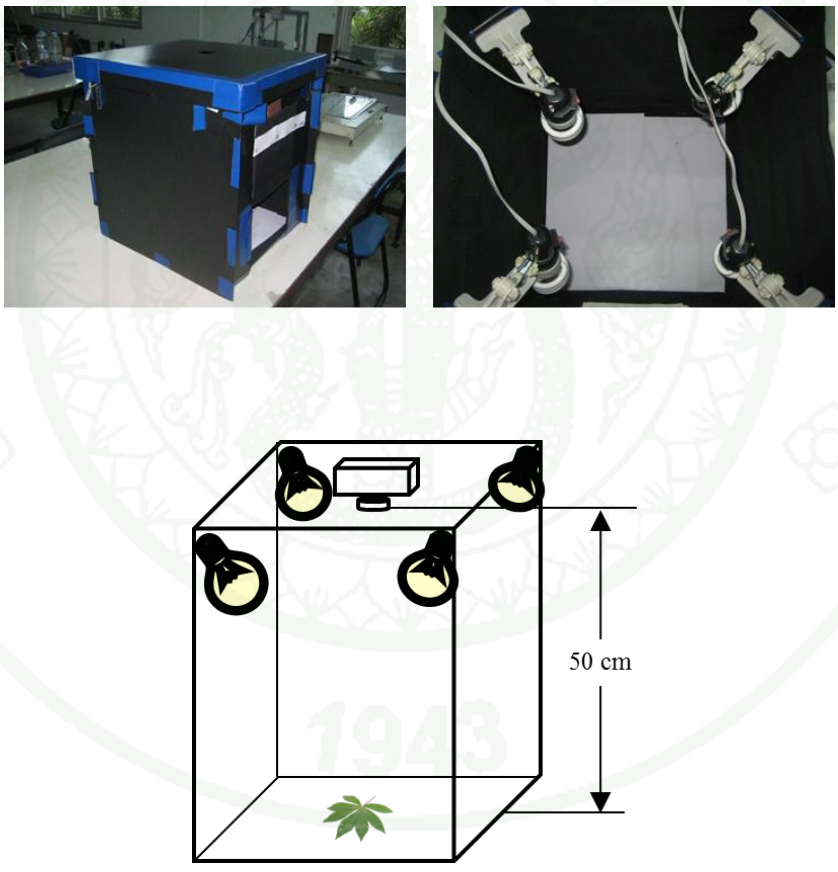


Figure 14 Illumination box and camera setup.

4.3.2 Image enhancement

The Image Processing Toolbox MATLAB® was used to enhance all images. The original images were resized to 640 pixels horizontally by 480 pixels vertically. The RGB images were converted to HSI color representation as it has been widely adopted for color segmentation rather than the use of RGB values because the HSI color is more tolerant to light intensity variation.

4.3.3 Image segmentation and feature extraction

Image segmentation and feature extraction was performed by searching an optimum threshold that would able to differentiate the spots from leaves and background. The segmentation procedure was separated into two parts.

First, the *I* image was segmented using Otsu's method which based on the analysis of the histogram of the tonal image. The leaf region was extracted and calculated for total leaf area (A_L) by pixels combination.

Second, the histogram of *H* image was analyzed to segment the spots feature. This could be done by following the Eq. 20 and the related procedure described previously. The noise distributing in the image was removed by Opening operations. An object in 2D binary image with 8 connected pixels was combined and identified as a spot. The disease area (A_D) was calculated from pixels combination and used as a parameter to indicate the percentage of infection (*PI*) according to the equation:

$$PI = \left(\frac{A_D}{A_L} \right) \times 100 \tag{21}$$

Given by *PI* is the percentage of inflection, A_D is the disease area, and A_L is total leaf area

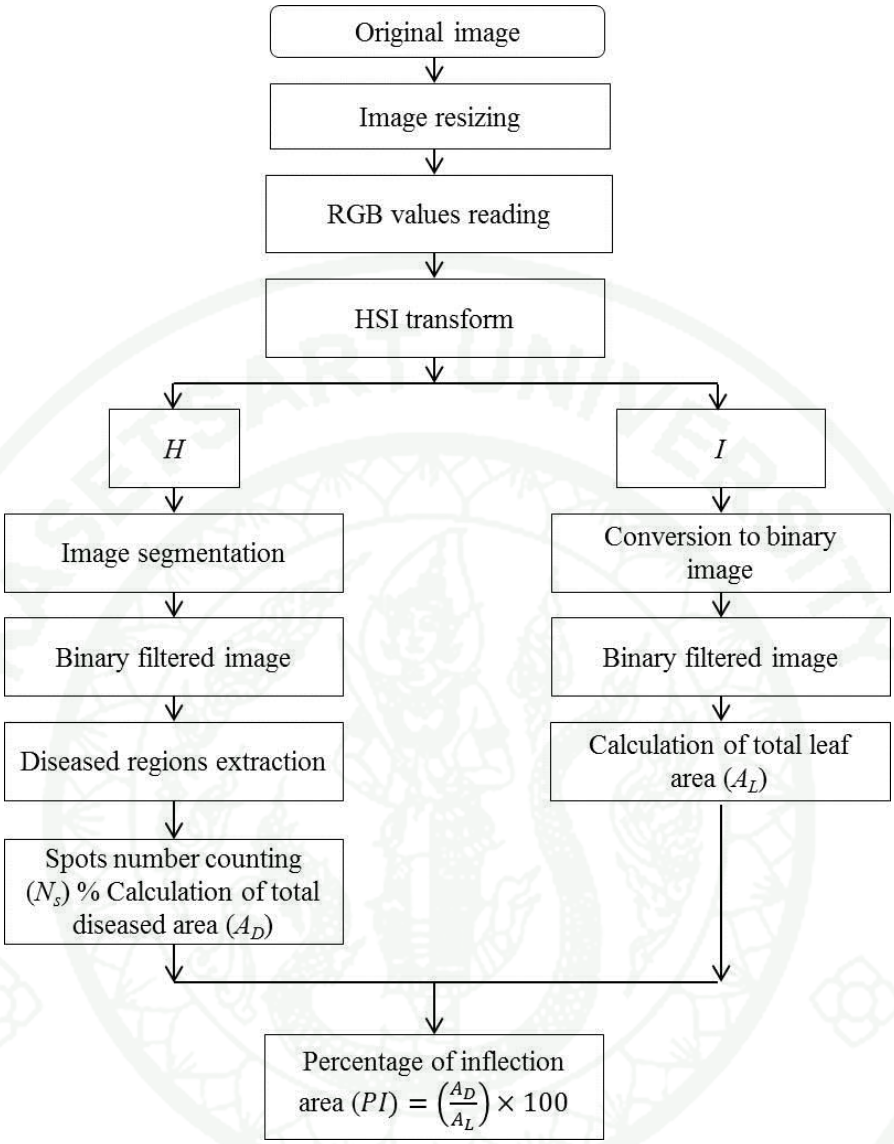


Figure 15 Flowchart of image processing algorithm for assessing the severity of BLS disease infection.

RESULTS AND DISCUSSION

1. Preliminary experiment on classification of healthy and diseased leaves under controlled background condition

The preliminary experiments have been conducted in order to observe the feasibility of using color indices for classification between an image that contains the brown spots and the non-diseased one. Six possible categories of the cassava leaves image after cropping to a block size of 80×80 pixels were, background only (B), leaf only (L), background and leaf (B-L), background with spot (B-S), leaf with spot (L-S), and background with leaf and spot (B-L-S) as illustrated in Figure 16.

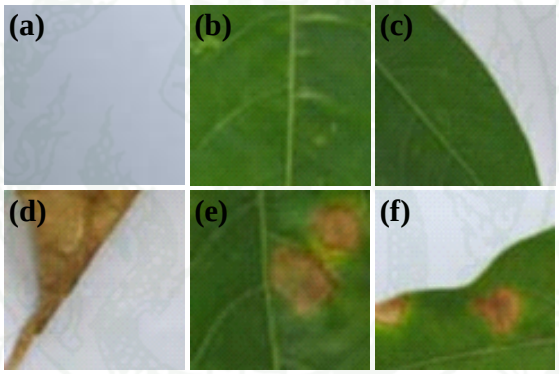


Figure 16 Examples of cassava leaf images cropped to a size of 80×80 pixels: (a) B, (b) L, (c) B-L, (d) B-S, (e) L-S, and (f) B-L-S.

The range of means and standard deviations of RGB color in each category are summarized in Table 1. The RGB value of background (B) reached nearly their maximum of 255. This was reasonable because of a white paper was used as background. Comparison of *R* value between L and L-S showed an obvious difference while the *B* value was likely to be a good separator between the case of healthy leaf (L) and some cases of disease (B-S and B-L-S).

Table 1 Ranges of mean and standard deviation of RGB values for each category.

Category		Red	Green	Blue
B	Mean	173.8-222.2	183.1-223.7	188.5-226.2
	Std.dev	1.45-8.59	1.21-8.56	0.79-10.54
L	Mean	45.5-77.5	72.3-122.7	27.7-48.3
	Std.dev	3.49-10.80	3.47-11.56	3.29-31.61
B-L	Mean	81.3-154.8	105.2-162.8	57.3-153.2
	Std.dev	40.38-79.07	32.64-68.43	49.95-95.32
B-S	Mean	165.6-197.8	146.9-192.4	112.0-194.3
	Std.dev	11.24-35.49	16.45-48.77	10.63-74.01
L-S	Mean	60.9-110.4	74.0-117.4	26.2-37.0
	Std.dev	16.88-35.81	8.75-15.26	6.78-28.29
B-L-S	Mean	90.7-158.1	66.6-138.7	66.6-138.7
	Std.dev	46.40-73.04	29.63-64.52	63.84-98.16

The two ANN models have been trained using different architectures. Several trials showed that networks could adequately learn the relationships with one hidden layer of 15 neurons. Actual activations of the networks did not reach their extreme target values of 0.1 or 0.9, but it was obviously classifiable if a threshold value of 0.5 is applied. Validation tests demonstrated that the Model II network gave a correct classification rate of 91.67%, higher than 87.5% of the Model I architecture (Table 2).

Table 2 Effect of number of hidden neurons and classification accuracy.

Number of hidden neurons	Success classification (%)	
	Model I	Model II
10	75.00	70.83
15	87.50	91.67
20	87.50	66.67
25	79.17	91.67
30	83.33	87.50

According to the results, the highest classification accuracy of 91.67% was achieved from the ANN Model II with the number of hidden neurons of both 15 and 25, nevertheless, the ANN with 15 neurons was considered the most preferable architecture because it have been adequate to learn functional relationships embedded in the samples as well as have been able to classify the patterns while keeping its simplicity and to prevent over-fitting of the network. In conclusion, this part of study has demonstrated the possibility of using color features associated with the ANN-based classification technique for BLS disease detection in cassava.

2. Development of image analysis techniques for *in situ* detection of diseased cassava plants

The color indices of healthy and diseased leaves images have been obtained. Since the indices *rgb* and *HSI* theoretically range from 0 to 1, the ranges of these indices given in Table 3 thus demonstrated that the data used in this study was adequately extensive which confirmed the generality of the technique. The modified hue fell into a range of approximately 0.1 to 0.4. These respectively correspond to the hue angle between 36 and 144° which are the sectors of yellow to green colors.

Comparison between index values of healthy and diseased leaves is shown in Figure 17. The green coordinate was more or less the same. The red coordinate of disease was greater than that of healthy leaves while the coordinate blue resulted in

the opposite. These are reasonable because the spots appear in brown color. The $r - g$ index of both healthy and diseased leaves had values that were often negative. When the indices were combined as a ratio index of $(g - b)/|r - g|$, better contrast between the healthy and diseased leaves was obtained, but with increased standard deviation. Such result is very similar to the findings of Woebbecke *et al*. (1995a) though in their study the index was used to distinguish between plant and background.

Table 3 Ranges of color indices

Indices	Ranges of index values	
	Healthy leaves	Diseased leaves
r	0.1044 – 0.7790	0.1325 – 0.9184
g	0.1419 – 0.9359	0.1808 – 0.9771
b	0.0617 – 0.7358	0.0261 – 0.6790
$r - g$	–0.3036 – 0.0539	–0.2793 – 0.0842
$g - b$	0.0282 – 0.4880	0.0409 – 0.5425
$(g - b)/\|r - g\|$	0.1579 – 10.5600	0.5315 – 26.7671
$2g - r - b$	0.0187 – 0.7548	0.0548 – 0.7469
H	0.1049 – 0.4382	0.1117 – 0.3993
S	0.0673 – 0.8432	0.1081 – 0.9439
I	0.1168-0.7882	0.1168-0.8370

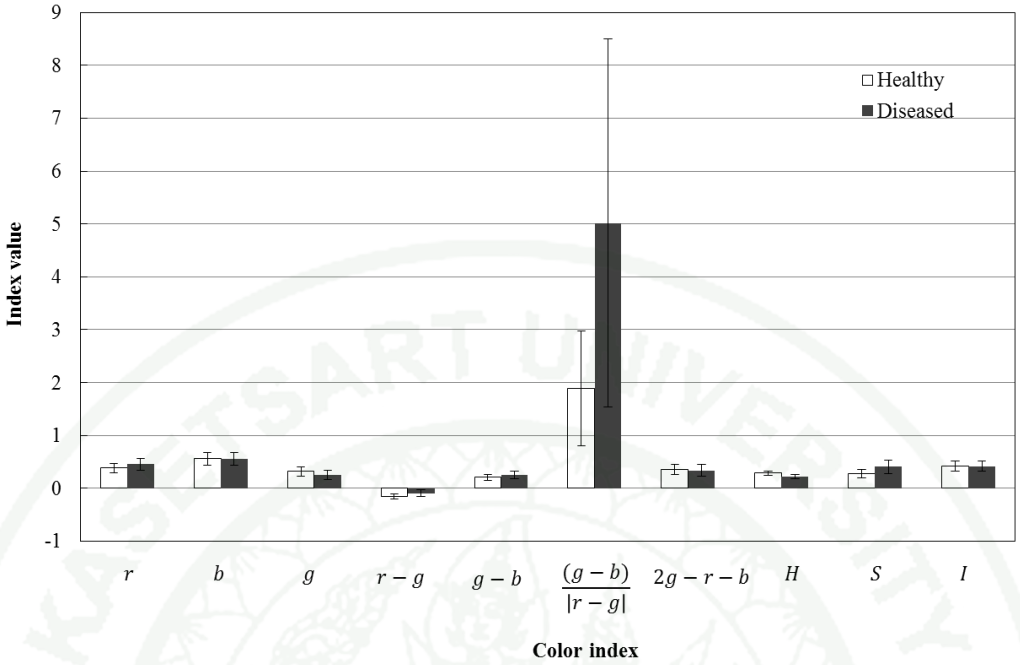


Figure 17 Comparison of index values of healthy and diseased leaves. Error bars represent standard deviation from the mean.

The present result thus suggested that the ratio index is still applicable to distinguish the healthy and diseased leaves even though in which case its contrast may be smaller than that between the plant and background. Modified hue of diseased leaves was lower than healthy leaves, indicating the approach of color towards yellow region. The modified hue and saturation showed some promise for separation of diseased leaves. The intensity was nearly the same and hence may not be a good parameter.

2.1 Artificial neural network classification

The ANN have been trained using different different numbers of hidden neurons. Actual activations of the neural network did not reach their extreme target values neither the lowest value of 0.1 nor the highest value of 0.9. However, the predictions were obviously classifiable if a threshold value of 0.5 was applied. The highest recognition accuracy of healthy leaves was 89.92% with 20 hidden neurons, while the highest possible accuracy for the diseased leaves was 79.23% with 30 hidden neurons (Table 4). It is notable that a high classification accuracy for a class

may be attained but with a repay of obtaining lower accuracy for the other class. Several trials showed that the ANN could adequately learned the relationships with one hidden layer of only 10 neurons. Increasing the number of hidden neurons seems to have no effect on the classification accuracy. Using hidden neurons more than 20 nodes was therefore considered unnecessary. Evaluation of total classification performance based on Brier score confirmed that the ANN with 20 hidden neurons was the best architecture for the circumstances of this study.

The present study showed a feasibility to detect the visible symptoms of BLS-infected cassava plants under field conditions by means of the machine vision. Considering from the porportion of training and test data set (62.5% for training and 37.5% for testing), the ANN was satisfactorily performed.

Table 4 Implication of number of hidden neurons and success classification rate.

Number of hidden neurons	Success classification (%)		Brier score
	Healthy plants	Diseased plants	
10	87.94	77.13	0.1528
20	89.92	75.73	0.1431
30	86.70	79.23	0.1552
40	88.68	77.48	0.1465
50	87.49	77.36	0.1552
60	88.04	78.06	0.1493
70	89.77	74.21	0.1486
80	87.35	78.18	0.1538
90	87.59	77.25	0.1549
100	87.05	77.36	0.1583

2.2 Discriminant analysis classification

The multicollinearity of color indices has been examined by Pearson's Product Moment Correlation (Table 5). The results showed that all color indices were significant in classification ($P<0.05$), but there existed a high dependency among indices. In addtion, the tolerance and VIF values have been calculated as shown in Table 6. The most of color indices (except $g - b$), the tolerance value was not close to 1 and the VIF value was greater than 10, implying that all color indices, excluding only $g - b$, have some linear correlations among themselves (Vanichbuncha, 2010). The factor analysis has been therefore carried out to represent those color indices with new parameters which are independent against each other.

Table 5 Results of Pearson's Correlation test.

		r	g	b	$r-g$	$g-b$	$\frac{(g-b)}{\lvert r-g \rvert}$	$2g-r-b$	H	S	I	group
	Pearson Correlation	1	.798**	.525**	.510**	-.086**	.420**	-.362**	-.317**	.139**	.838**	-.316**
r	Sig. (2-tailed)		.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
	Pearson Correlation	.798**	1	.722**	-.074**	-.032**	-.046**	.024*	.226**	-.066**	.915**	.013
g	Sig. (2-tailed)	.000		.000	.000	.005	.000	.037	.000	.000	.000	.256
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
	Pearson Correlation	.525**	.722**	1	-.034**	-.665**	-.373**	-.409**	.452**	-.675**	.802**	.309**
b	Sig. (2-tailed)	.000	.000		.003	.000	.000	.000	.000	.000	.000	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
	Pearson Correlation	.510**	-.074**	-.034**	1	-.284**	.679**	-.785**	-.795**	.162**	.136**	-.468**
$r-g$	Sig. (2-tailed)	.000	.000	.003		.000	.000	.000	.000	.000	.000	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
	Pearson Correlation	-.086**	-.032**	-.665**	-.284**	1	.277**	.817**	-.212**	.867**	-.263**	-.299**
$g-b$	Sig. (2-tailed)	.000	.005	.000	.000		.000	.000	.000	.000	.000	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680

Table 5 (Continuted)

		r	g	b	$r-g$	$g-b$	$\frac{(g-b)}{\lvert r-g\rvert}$	$2g-r-b$	H	S	I	group
$\frac{(g-b)}{\lvert r-g\rvert}$	Pearson Correlation	.420**	-.046**	-.373**	.679**	.277**	1	-.230**	-.727**	.594**	.005	-.559**
	Sig. (2-tailed)	.000	.000	.000	.000	.000		.000	.000	.000	.673	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
$2g-r-b$	Pearson Correlation	-.362**	.024*	-.409**	-.785**	.817**	-.230**	1	.341**	.463**	-.251**	.088**
	Sig. (2-tailed)	.000	.037	.000	.000	.000	.000		.000	.000	.000	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
H	Pearson Correlation	-.317**	.226**	.452**	-.795**	-.212**	-.727**	.341**	1	-.549**	.131**	.564**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000		.000	.000	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
S	Pearson Correlation	.139**	-.066**	-.675**	.162**	.867**	.594**	.463**	-.549**	1	-.210**	-.526**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000		.000	.000
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680
I	Pearson Correlation	.838**	.915**	.802**	.136**	-.263**	.005	-.251**	.131**	-.210**	1	-.003
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.673	.000	.000	.000		.802
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680

Table 5 (Continued)

		r	g	b	$r-g$	$g-b$	$\frac{(g-b)}{\lvert r-g \rvert}$	$2g-r-b$	H	S	I	group
group	Pearson Correlation	-.316**	.013	.309**	-.468**	-.299**	-.559**	.088**	.564**	-.526**	-.003	1
	Sig. (2-tailed)	.000	.256	.000	.000	.000	.000	.000	.000	.000	.802	
	N	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680	7680

* Correlation is significant at 0.05 level (2-tailed).

** Correlation is significant at 0.01 level (2-tailed).

Table 6 Result of collinearity test by tolerance and VIF statistics.

Parameter	Unstandardized coefficients		Standardized coefficients	t	Sig.	Collinearity statistics	
	B	Std. Error	Beta			Tolerance	VIF
(Constant)	0.602	0.297		2.023	0.043		
r	-2.496	0.266	-0.548	-9.375	0.000	0.021	46.799
g	0.713	0.206	0.173	3.462	0.001	0.029	34.314
b	1.928	0.224	0.449	8.604	0.000	0.027	37.317
$r-g$	0.173	0.325	0.045	0.533	0.594	0.010	97.523
$g-b$	-	-	-	-	0.000	-	0.000
$\frac{(g-b)}{\lvert r-g \rvert}$	0.205	0.105	0.034	1.960	0.050	0.236	4.233
$2g-r-b$	1.206	0.348	0.274	3.462	0.001	0.012	85.969
H	-0.243	0.093	-0.062	-2.605	0.009	0.129	7.754
S	-1.523	0.151	-0.338	-10.057	0.000	0.065	15.449
I	-0.277	0.135	-0.063	-2.050	0.040	0.078	12.891

Factor analysis has been performed using Principal Components Analysis (PCA). The PCA reduced parameters of color indices by forming up three new factors which orthogonal to each other. Factor 1 is the first linear combination which stores as much as possible the details of color indices. Factor 2 is the second linear combination that is orthogonal to Factor 1 which store additional information of color indices different from Factor 1. Factor 3 would keep the rest of relationships and that does not correlate with Factor1 and Factor 2.

The results shown in Table 7 suggested that there were three principal components (PCs) suitable for DA technique because only these three PCs had an eigenvalue greater than 1.0 (Figure 18). These three principal components have covered the variance of color indices respectively of 39.634, 33.159, and 22.465%, giving an overall variance of 95.2581%. This has been sufficient for DA classification.

Table 7 Total variance explained by principal component analysis.

Component	Initial eigenvalues			Extraction sums of squared loadings		
	Total	% of variance	Cumulative %	Total	% of Variance	Cumulative %
1	3.963	39.634	39.634	3.963	39.634	39.634
2	3.316	33.159	72.793	3.316	33.159	72.793
3	2.247	22.465	95.258	2.247	22.465	95.258
4	.264	2.636	97.894	-	-	-
5	0.092	0.924	98.819	-	-	-
6	0.056	0.557	99.376	-	-	-
7	0.035	0.346	99.722	-	-	-
8	0.018	0.175	99.898	-	-	-
9	0.010	0.102	100.000	-	-	-
10	0.000	0.000	100.000	-	-	-

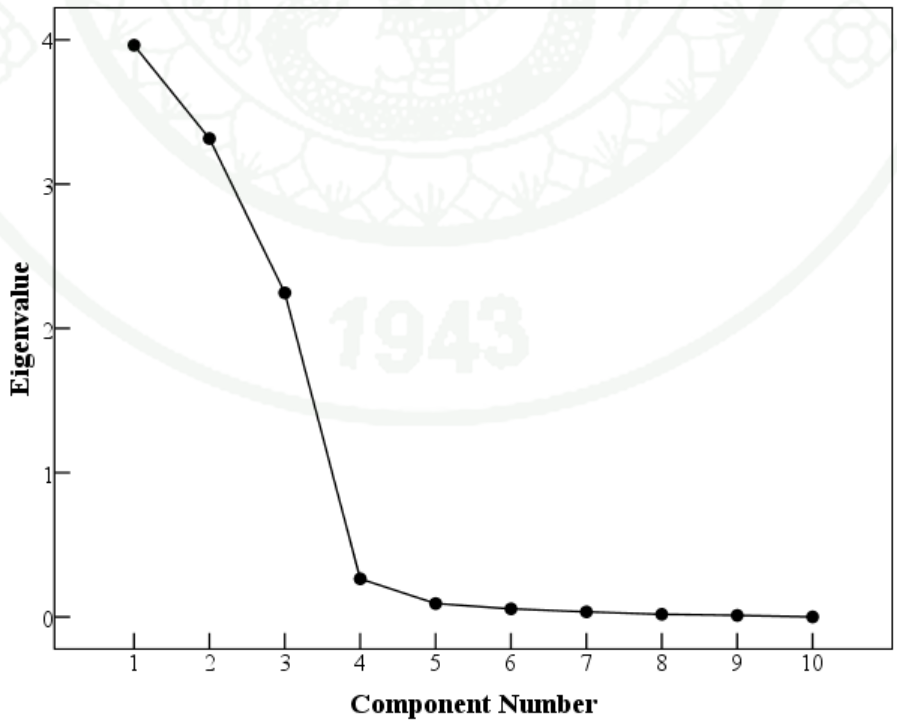


Figure 18 Eigenvalues of each component

Table 8 Rotated component matrix

Color index	Principal component (PC)		
	PC1	PC2	PC3
b	0.969	-	-
I	0.834	0.200	0.473
$g - b$	-0.725	-	0.672
g	0.700	-	0.696
S	-0.658	0.512	0.508
$r - g$	-	0.892	-0.391
H	0.290	-0.882	-
$(g-b)/\lvert r-g\rvert$	-0.231	0.872	-
r	0.621	0.638	0.436
$2g - r - b$	-0.571	-0.468	0.670

As a consequence of PCA, the original color indices were rearranged into either PC1, PC2 or PC3 depending on loading value. If the loading value of a color index was close to -1 or +1, the index then belongs to that corresponding PC. On the other hand, the index with loading value close to zero would be grouped into other PC. In the other words, a color index would be categorized into a PC that having the highest loading value. As shown in Table 8 and Figure 19, the PC1 consisted of five parameters; b, I, $g - b$, g and S. The PC2 covered four parameters; $r - g$, H, $(g-b)/\lvert r-g\rvert$ and r, while merely $2g - r - b$ constituted PC 3.

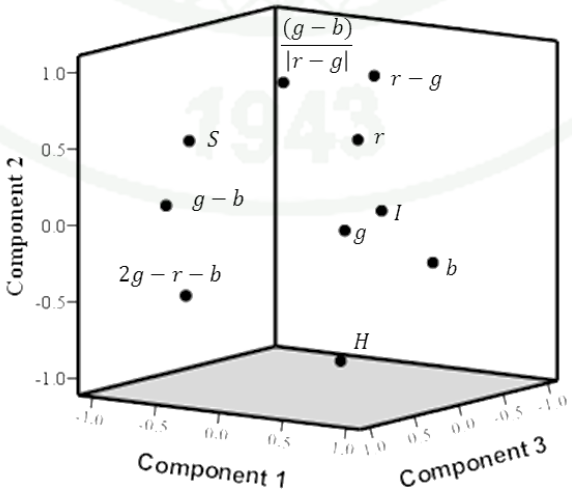


Figure 19 Distribution of principal components

The component score coefficient of each color index for each PC are given in Table 9

Table 9 Component score coefficient matrix

Color index	Principal component (PC)		
	PC1	PC2	PC3
r	0.157	0.192	0.194
g	0.177	0.027	0.310
b	0.245	-0.054	0.040
$r-g$	0.043	0.269	-0.174
$g-b$	-0.183	0.032	0.299
$(g-b)/\lvert r-g\rvert$	-0.058	0.263	0.026
$2g-r-b$	-0.144	-0.141	0.298
H	0.073	-0.266	0.070
S	-0.166	0.154	0.226
I	0.210	0.060	0.211

Using the coefficients given in the Table 9, three principal functions may be derived as follows:

$$PC1 = 0.157r + 0.177g + 0.245b + 0.043(r-g) - 0.183(g-b) - 0.058\frac{(g-b)}{|r-g|} - 0.144\,(2g-r-b) + 0.073H - 0.166S + 0.21I \quad (22)$$

$$PC2 = 0.192r + 0.027g - 0.054b + 0.269(r-g) + 0.032(g-b) + 0.263\frac{(g-b)}{|r-g|} - 0.141\,(2g-r-b) - 0.266H + 0.154S + 0.060I \quad (23)$$

$$PC3 = 0.194r + 0.310g + 0.040b - 0.174(r-g) + 0.299(g-b) + 0.026\frac{(g-b)}{|r-g|} + 0.298\,(2g-r-b) + 0.07H + 0.226S + 0.211I \quad (24)$$

These three PCs were used to classify the healthy and diseased cassava leaves by DA. Two prediction equations (Eq. 25 and 26) for healthy and diseased images could be developed using the coefficients given in Table 10. The classification could be implemented by substituting PC1, PC2 and PC3 in both equations. The higher regression factor scores (REGR) indicates that the image belong to which group.

Table 10 Coefficients of classification functions

PC	Classification groups	
	Healthy	Diseased
1	0.317	-0.451
2	-0.780	1.488
3	-0.155	0.38
Constant	-0.901	-1.409

Healthy group:

$$HG = 0.317\ (PC1) - 0.780\ (PC2) - 1.155\ (PC3) - 0.901 \quad (25)$$

Diseased group:

$$DG = -0.451\ (PC1) + 1.488\ (PC2) + 0.38\ (PC3) - 1.409 \quad (26)$$

In Figure 20, the histogram shows the distribution of discriminant scores of the classification function for each group. Although the score frequency on axes of healthy and disease groups showed some overlapping, however, a significant mean difference was observed for all of the predictors. This result suggested that the function was highly potential to discriminate the healthy and the diseased groups.

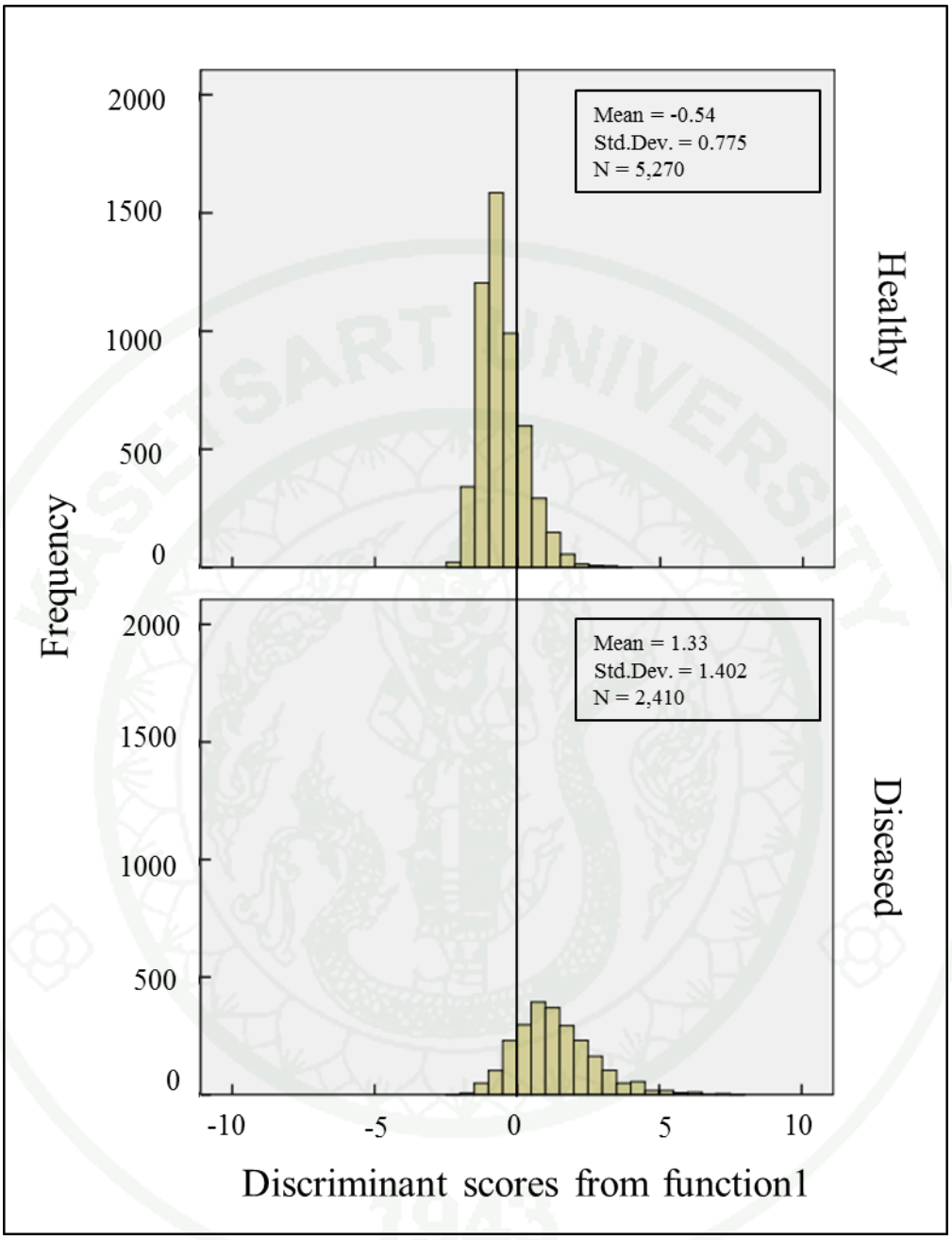


Figure 20 Histograms of discriminant scores distribution for healthy and diseased groups.

The 3247 images of healthy and 1153 images of diseased leaves have been used to create the prediction equations. As shown in Table 11, the classification accuracy for calibration set was 83.7%. In detail, the accuracy of calibration set for healthy group was 89.0% while the accuracy for the diseased groups was 72.6%. This suggested that, for calibration data set, the discriminant analysis could recognized the healthy leaves better than the diseased leaves. When the classification functions were applied a separated validation data set, the result the average pecentage of correct classification was 83.3% (Table 12). Similarly, the DA could recognized the healthy leaves better than the diseased leaves. However, the recognition of diseased leaves in the validation image set could reach 79.4%, even higher than that in the calibration image set (72.6%). The discriminant analysis was therefore, to some extent, a capable method to classify the BLS disease in cassava.

Table 11 Classification accuracy of calibraiton image set.

Actual groups	Accuracy of classification		All images	Average percentage of correction
	Healthy	Disease		
Healthy	2889 (89.0%)	358 (11.0%)	3247	83.7%
Disease	426 (27.4%)	1127 (72.6%)	1553	

Table 12 Classification accuracy of validation image set.

Actual groups	Accuracy of classification		All images	Average percentage of correction
	Healthy	Disease		
Healthy	1714 (84.7%)	309 (15.3%)	2023	83.3%
Disease	172 (20.1%)	685 (79.9%)	857	

Comparison of ANN and DA classification indicated that the ANN could classify the group of healthy leaves better than DA, whereas the accuracy of DA for classifying diseased leaves was higher than ANN. The comprison of the overall goodness between the ANN and DA was therefore performed based on Brier score. As shown in Table 13, the Brier score indicated that ANN had higher pontential in classification than DA. Furthermore, the ANN may be superior than the DA because it is not necessary neither to convert the primary color indices into principal components nor to test if there is multicollinearity among variables. The ANN is therefore a promising approach for further development of a machine vision system for BLS disease detection.

Table 13 Comparison of classification performance between ANN and DA.

Classification method	Group		Brier score
	Healthy	Diseased	
ANN with 20 hidden neurons	89.92%	75.73%	0.1431
DA	84.70%	79.90%	0.1654

Both ANN and DA classification, nevertheless, suggested some possibility of misidentification of the diseased plants. This is likely to have been occurred due to many reasons. First, the study attempted to analyze the images without segmentation of leaves from background prior to detect the spots on the leaves. Maintaining of the background could therefore caused misleading color representation of actual healthy leaves and background which can be any from stems, soil, weeds, or residues. Second, the method did not taken into consideration the stage of disease infection or, in the other words, the degree of leaves damage. Obseravation indicated that the algorithm tended to detect yellowish leaves rather than really detecting the brown spots on leaves. Therfore, the classification would be correct only if the plants has already reached a high level damage. The early stages of infection at which the brown spots have appeared whereas the leaves are still green, can result in incorrect identification.

Misclassification of the diseased plants may also be attributed to the illuminating condition in the cassava fields. Although the more robust color model such as HSI was used, however, variation of natural lighting was unavoidable. This can also depends on camera position relative to the cassava canopy and shading due to occlusion among the leaves. In order to improve the recognition ability, these issues must be taken in account associated with a more elaborate representation and interpretation of details in the images.

3. Development of image analysis algorithms for precision positioning of disease spots on cassava leaves

3.1 Histogram analysis of selected color indices for spot segmentation

An initial comparison of color indices was performed and analyzed based on the difference of histogram. Color indices derived from HSI and RGB color models were used. In the identification of disease regions in the initial step, the histograms of *ExR*, *ExG*, *ExG – ExR*, and *H* were processed and observed their features (Figure 21). The histogram of *ExG* image showed bi-modal distribution (Figure 21b) in which one peak represented the leaves object the other peak indicated the background (e.g. soil, stems, and weeds). The *ExG* image was thereby used to calculate for an appropriate threshold value by Otsu's methods for leaves pixels extraction.

The characteristic of *ExG – ExR* image was almost similar to the *ExG* image for the area of spots and leaves, but the *ExG – ExR* image showed better contrast and hence a promising index for the image segmentation (Figure 21c). The *ExG – ExR* has showed a great deal of gray-level details, thus, the spots on cassava leaves could be standout and the *ExG – ExR* index was more suitable for spot extraction comparing with the *ExG* or the *ExR* images. The *H* image, as shown in Figure 21(d), gave a clear appearance of spots positions and hence can be utilized to extract the spots as well.

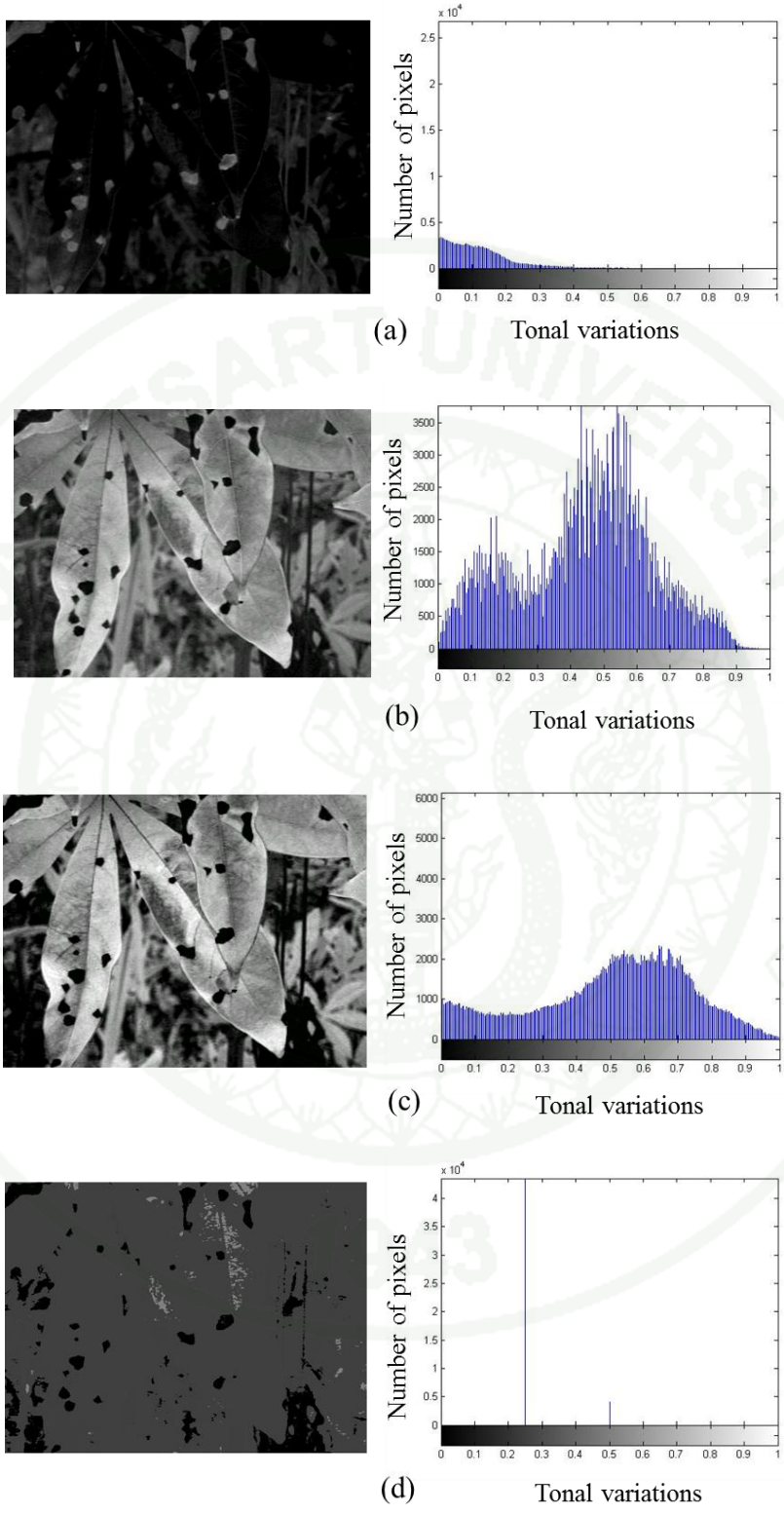
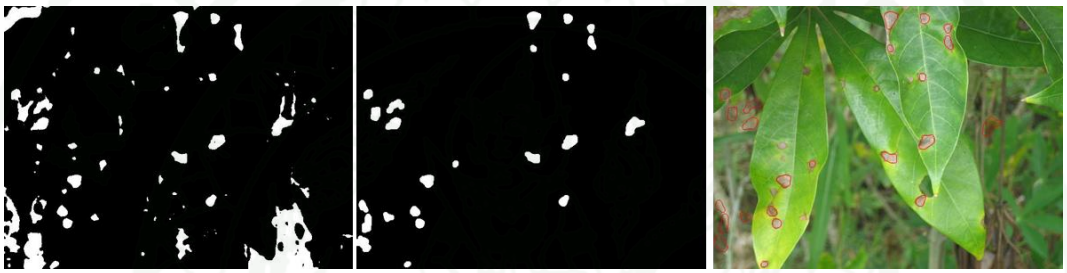


Figure 21 Histogram analysis of selected color indices for spots segmentation: (a) *ExR*, (b) *ExG*, (c) *ExG – ExR* and (d) *H*.

In the spots segmentation, both *ExG – ExR* and *H* images were used. The threshold value for segmentation was determined manually by selecting the peaks in the histogram that best discriminate the infected regions from the healthy leaves. Selection of threshold value on *H* image could be done slightly easier than done on the *ExG – ExR* image as the *H* image revealed sharper lines. On the *H* histogram, there were three lines appeared at grey levels of around 0, 0.25 and 0.5, giving boundaries for spots, background and noise, respectively. The results of image segmentation by each of the three algorithms are shown in Figure 22–24.



(a) (b) (c)

Figure 22 Image segmentation and feature extraction by AG I: (a) Segmented image, (b) Disease regions extracted after noise reduction, and (c) Marking with red lines on the spots detected.

(a) **(b)** **(c)**

Figure 23 Image segmentation and feature extraction by AG II: (a) Segmented image, (b) Disease regions extracted after noise reduction, and (c) Marking with red lines on the spots detected.

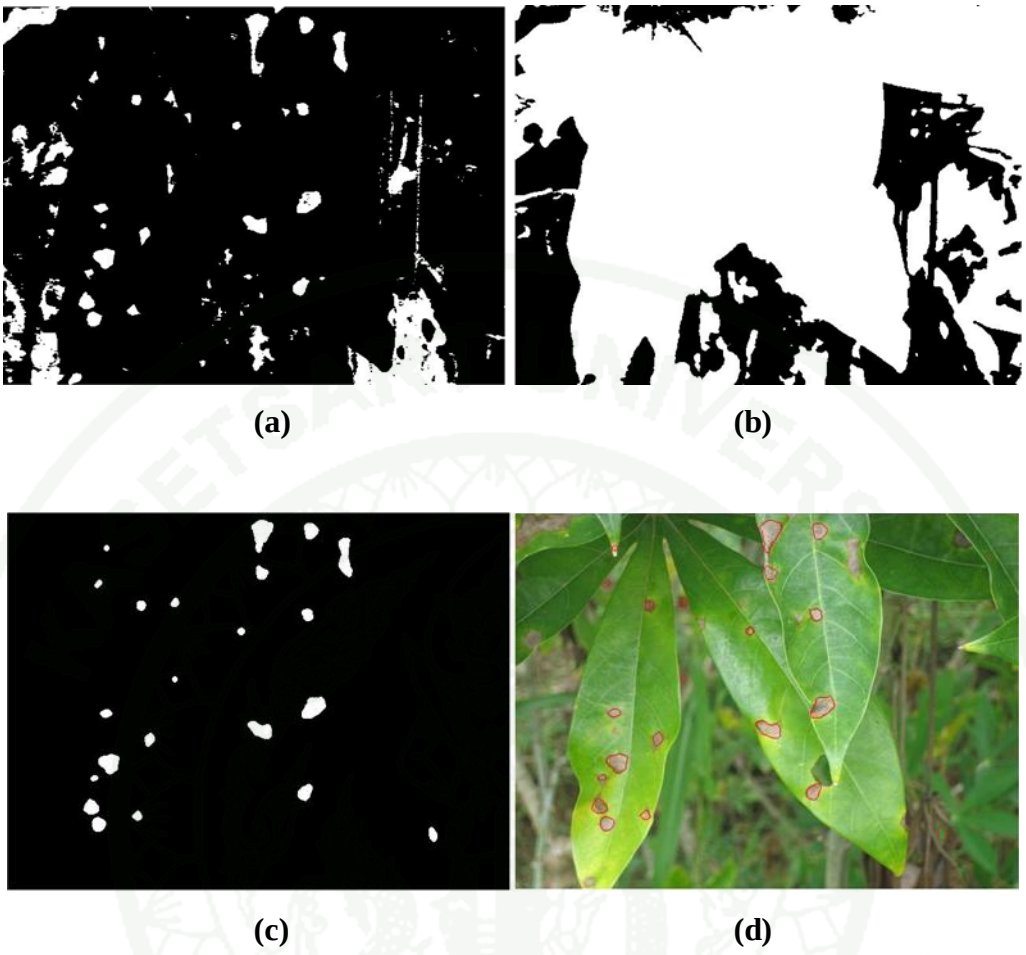


Figure 24 Image segmentation and feature extraction by AG III:
(a) Spots segmentation, (b) Leaves segmentation (c) Disease regions after intersection between (a) and (b), and after noise reduction, and (d) Marking with red lines on the spots detected.

3.2 Evaluation of spot detection performance

Accuracy of classification for healthy leaves performed by each algorithm is shown in Table 14. If the healthy leaves image is correctly recognized, the number of spot detected should be zero. The AG I and AG II, however, reported some spots detected which is obviously incorrect and hence very low accuracy i.e. 22% for AG I and 0% for AG II. In other words, the AG II was not able to classify any healthy leaves, indicating that the AG II may not be a suitable algorithm for BLS disease detection at all. The accuracy of AG III was slightly higher than AG I, and the

standard deviation of AG I was also lower than the AG III. In addition, considering from the number of misled spot detection of less than 3 which much better than the other two algorithms, suggesting that the AG III may be the most preferable algorithm for this specific application.

Table 14 Accuracy and misclassification of healthy leaves detection

Algorithm	Accuracy	Misclassification	
		Mean ± Std.dev	Number of misclassified spot
I	22%	4.46 ± 4.726	0-20
II	0%	10.96 ± 7.537	2-39
III	36%	1.12 ± 1.023	0-3

The performance of diseased regions detection was evaluated by Paired Sample T-test across all 50 samples of diseased leaves images. The results indicated all of the algorithms gave significant correlation ($P<0.05$) when compared with the actual number of spots counted manually while the AG III gave highest correlation with a coefficient of determination, R^2 of 0.744 (Table 15).

Table 15 Performance of diseased regions detection evaluated by Paired Sample T-test

Algorithm	Number of sample	R^2	Significant level	Mean number of spot	Std. dev
AG I	50	0.529*	7.75×10^{-5}	12.040	5.194
AG II	50	0.743*	6.30×10^{-10}	16.840	9.274
AG III	50	0.744*	6.06×10^{-10}	13.780	6.716

*Significant at 0.05 level.

Table 16 Overall performance of algorithms evaluated by Paired Sample T-test and RMSE

Algorithm	Number of sample	R^2	Significant level	RMSE
AG I	100	0.4658	5.28×10^{-15}	13.10
AG II	100	0.3216	7.69×10^{-10}	12.77
AG III	100	0.8325*	8.42×10^{-40}	10.56

*Significant at 0.05 level

Table 16 shows the overall performance of the three algorithms when applied to the entire image set including both healthy and diseased leaves. All algorithm could recognized respective images, giving significant correlations ($P<0.05$) when compared with actual number of spots counted manually. The AG III provided best correlation at a coefficient of determination, R^2 of 0.8325. Evaluation based on RMSE has consistently demonstrated that the AG III was the most accurate algorithm with a least RMSE value of 10.56.

Regression analysis to compare the number of brown spots detected by image analysis with the actual number obtained by visual counting for each algorithm is shown in Figure 25. It is notable that all algorithms tend to underestimate the number of spots. This error, however, is likely systematic, suggesting that an adjustment could be done to achieve better accuracy upon actual implementation.

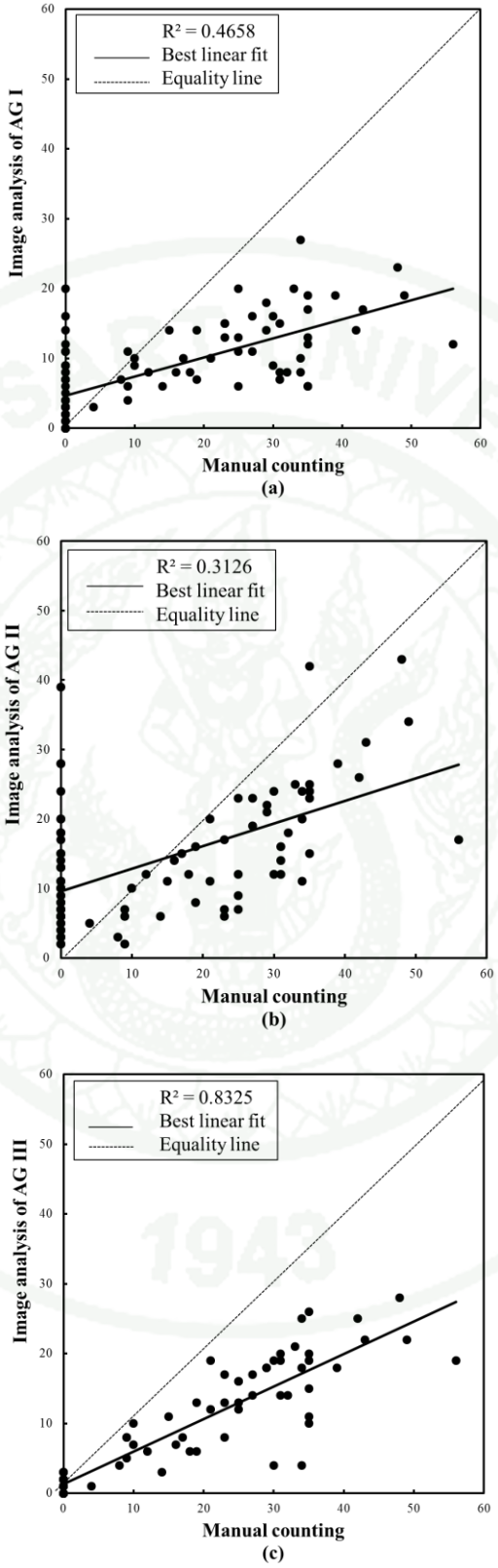


Figure 25 Regression analysis to evaluate the disease detection performance of image processing algorithms: (a) AG I, (b) AG II, and (c) AG III.

Table 17 Correction coefficients for AG I

	Unstandardized Coefficients		Standardized Coefficients	t	Significant level
	B	Std.Error	Beta		
Constant	-0.858	1.902		-0.451	0.653
AG I	1.703	0.184	0.683	9.245	0.000*

Significant at 0.05 level.

Table 18 Correction coefficients for AG II

	Unstandardized Coefficients		Standardized Coefficients	t	Significant
	B	Std.Error	Beta		
Constant	-0.565	2.394		-0.236	0.814
AG II	0.990	0.145	0.567	6.816	0.000*

Significant at 0.05 level.

Table 19 Correction coefficients for AG III

	Unstandardized Coefficients		Standardized Coefficients	t	Significant
	B	Std.Error	Beta		
Constant	-0.104	0.0879		-0.119	0.906
AG III	1.784	0.081	0.912	22.067	0.000*

Significant at 0.05 level.

The correction coefficients for AG I, II, and III are given in Table 17, 18, and 19 respectively. The regression equation for respective algorithms can be thereby derived as expressed in Eq. 28–30.

$$Y_{AG1} = 1.703X_{AG1} - 0.858 \quad (28)$$

$$Y_{AG2} = 0.990X_{AG2} - 0.565 \quad (29)$$

$$Y_{AG3} = 1.784X_{AG3} - 0.104 \quad (30)$$

where, Y_{AG1}, Y_{AG2} and Y_{AG3} are the corrected number of spots given by AG I, II and III respectively, and X_{AG1}, X_{AG2} and X_{AG3} are the original number of spots given by AG I, II and III before correction.

Using the above equations, an adjustment to correct of number of spots given by the developed image analysis algorithms can be successfully done.

This part of study has demonstrated that it is highly possible to locate the positions of brown spots on cassava leaves using image analysis if appropriate color indices associated with a systematic processing algorithm is used. As for this study, the use of *ExG* index has provided a good segmentation of leaves from background while the *H* index could well standout the brown spots on leaves. The intersection operation of *ExG* and *H* images, i.e. the AG III, thus, enable the positioning of the diseased spots on leaves at a satisfactory extent of accuracy (highest R^2 of 0.8325 and smallest RMSE of 10.56).

4. Development of image processing algorithm for the assessment of diseases infection severity

4.1 Image segmentation

The pixels of infected region have been extracted from the healthy portion based on the difference of *H* value. A key to the success in segmentation has been the determination of a proper threshold value. In this experiment, threshold value was manually selected based on the peaks appeared in the histogram. The result of image segmentation is illustrated in Figure 26.

(a) (b) (c)

Figure 26 Segmented and feature extracted images: (a) Total leaf area, (b) Diseased regions with noise, and (c) Disease regions after noise reduction.

4.2 Verification of Teri's diagram with image analysis

Teri's diagram has been analyzed by images processing. The results indicated that the number of spots and spots positions were correct (Figure 27). However, the percentage of infection area was found considerably different especially at the severity levels of 1–3 (Table 20). The values obtained from image analysis were used as a new criterion for further classification of the severity levels.

Table 20 Verification of Teri's diagram with image analysis

Infection level	Teri's values		Image analysis	
	Infected area	Number of spots	Infected area	Number of spots
1	5%	14	0.87%	14
2	10%	53	3.94%	53
3	15%	65	9.87%	65
4	20%	74	18.71%	74

4.3 Assessment of BLS disease inflection severity

4.3.1 Comparison of assessment methods

Comparison between the percentage of infected leaves samples obtained from image analysis and Teri's diagram resulted in Table 21. The results of disease level assessed by three out of seven raters (rater 3, 4, and 7) were more or less consistent with the results of image analysis. Most of the total 48 samples, assessment by image analysis found that the percentage of infection (*PI*) was less than 4%, for only one sample, the *PI* was 8.20%. The estimation of infection degree by both methods was found coincide only for the level 1, whereas the rest was not well consistent. In general, image analysis tended to give lower estimation of PI than the use of Teri's diagram For example, the image analysis may classified disease severity into level 2 while a human rater scored it as level 3. Moreover, there were some discrepancies among raters depending on individual experience and decision of each rater. Therefore, the use of image analysis has suggested some advantage overwhelming manual scoring based on an illustrated area diagram.

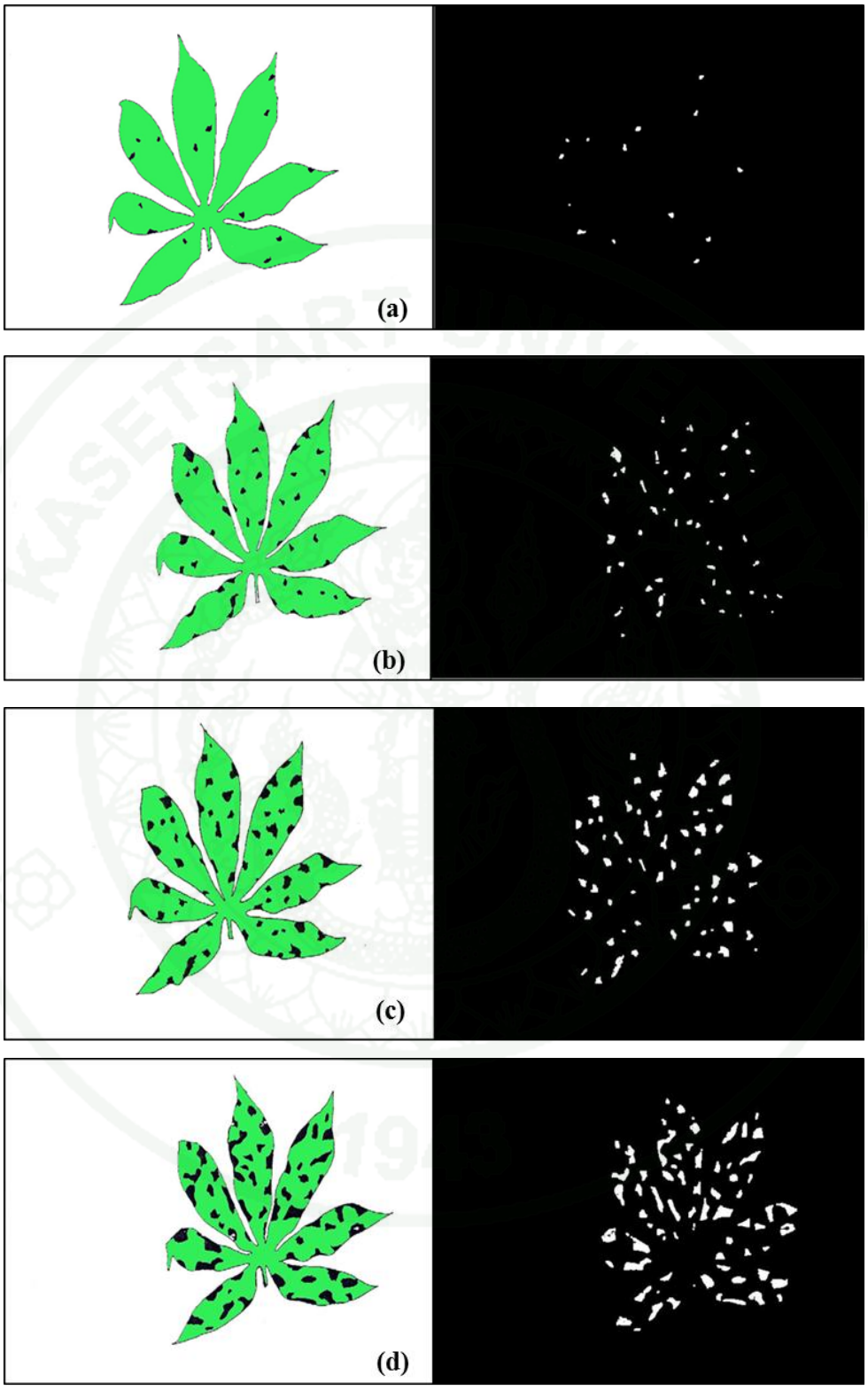


Figure 27 Verification of Teri's diagram with image analysis at severity levels of: (a) 5%, (b) 10%, (c) 15%, and (d) 20%.

Table 21 Comparison of disease level assessment results using image analysis and manual scoring based on Teri's diagram**.**

Disease level	Number of samples							
	Image analysis	Manual scoring						
		Rater 1	Rater 2	Rater 3	Rater 4	Rater 5	Rater 6	Rater 7
1	47	27	33	41	38	26	32	40
2	1	16	14	7	10	20	15	8
3	0	5	1	0	0	2	1	0
4	0	0	0	0	0	0	0	0
Total	48	48	48	48	48	48	48	48

Figure 28 shows the mean of disease levels averaged from of seven assessments (seven raters) for each sample throughout the entire 48 samples. It is noticeable that the mean of seven assessments on most samples was not coincide with the result of image analysis, especially for levels 2 and 3, there was a great variation among raters in the assessment which might cause a substantial error. It seems quite difficult to visually distinguish different patterns of the disease symptom at levels 2 and 3 because the size and distribution characteristics of the brown spots are very similar. Thus, experience in plant pathology and personal decision of a rater would be the most influential factors in the disease level assessment.

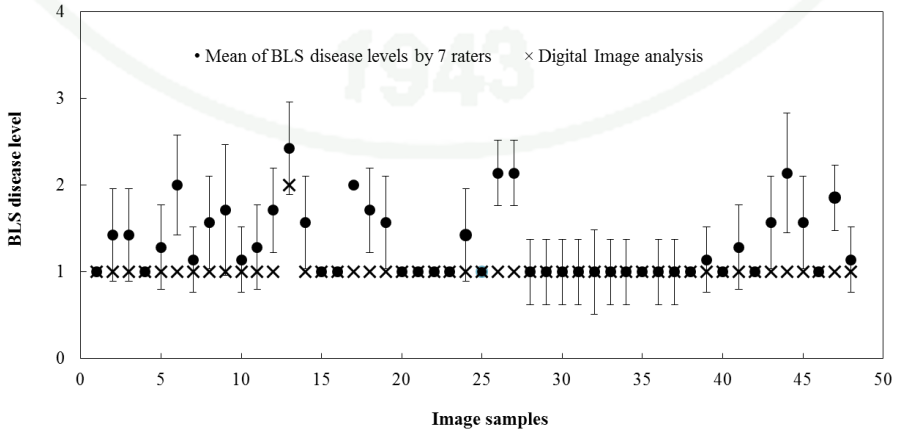


Figure 28 Results of disease level assessment by image analysis and manual scoring based on Teri's diagram.

4.3.2 Number of spot counts

Counting the number of disease spot on each cassava leaf samples found that there the number of spots ranged from 2 to 46. Figure 29 shows a strong correlation between the number of spots obtained by manual counting and that given by image analysis ($R^2 = 0.89$). Different results might be found for some case which may be attributed to the inconsistency of the raters.

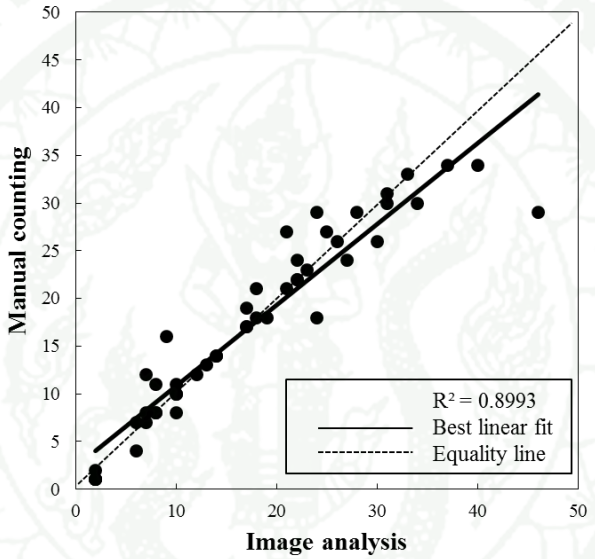


Figure 29 Comparison between the number of spot counts given by image analysis and by manual counting.

4.3.3 Influence of severity level on the accuracy of image analysis

The result showed some error when the number of spots less than 10. This suggested that the assessment may not be satisfactorily accurate at early stage of disease infection because the size and the number of spot are small. If the spots are small, the erosion and dilation processes might result in an elimination of those spots because they are very similar to the noise. For the leaf sample with more than 10 disease spots, the accuracy of image analysis could be as high as 89.30% and even higher at 91.37% when the number spots was more than 30 spots (Table 22).

Table 22 Effect of number of spots on accuracy of spots detection

Number of spot	Accuracy of spots detection
1 – 10	62.50%
11 – 20	89.30%
21 – 30	88.62%
31 – 40	91.37%
Average	82.94%

Besides the severity level, the assessment accuracy might be affected by incompleteness of image pixels. Sometime, many spots occur on an adjacent position and connect to each other which in turn form a large circular shape. This type of pixels can cause misleading interpretation when using the proposed image analysis procedure. On the other hand, sometime a single spot may contains a hollow inside and appear as though it is a multiple spots as shown in Figure 30. This type of pixels can cause overestimation of spot count as well.

Figure 30 Source of error in spots counting by image analysis: a single spots containing hollows.

A digital image analysis technique has been developed to identify the severity level of BLS disease infection on a cassava leaf. This technique was capable to estimate the percentage of disease infection quantified by damage area and number of brown spots. Preliminary verification of conventional diagram key indicated a

shortcoming in that it overestimates the percentage of infection area. A new percentage of inflection of area diagram key was therefore investigated and used as a new criterion for disease assessment. Comparing the results given by human raters further demonstrated many rooms for the occurrence of variance. An image analysis technique therefore suggested some advantages which help the plant pathologists as well as plant breeders to perform an accurate scoring of the cassava brown leaf spot disease.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the experimental results obtained from the present study, the following conclusions may be drawn:

1. Chromatic and color contrast derived from RGB and HSI color space can be potential parameters for image processing to detect visual symptoms of brown leaf spot disease in cassava.

2. It is possible to detect the cassava brown leaf spot disease in real field conditions with image analysis technique if appropriate color indices are used associated with an appropriate classification algorithm. In this study, some color indices have been used associated with artificial neural network and discriminant analysis classification. The disease has been successfully recognized at a highest possible accuracy of 79.90% which was achievable with discriminant analysis technique. The artificial neural network classification, however, yielded better overall performance when it is necessary to recognize both healthy and diseased plants.

3. Another image analysis technique has been developed to identify the positions of disease spots on a cassava leaf. The proposed algorithm operates by an intersection between *Hue* and *Excess Green* images. The algorithm has been satisfactorily located the positions of disease at an accuracy, by R^2, of 0.8325 and a root mean squared error of 10.56.

4. A different image analysis technique has been developed to assess severity level of brown leaf spot disease infection on a cassava leaf. This technique was capable to estimate the percentage of disease infection quantified by number of brown spots at an R^2 of 0.8993 comparing with the results given by human raters who scored the disease based on conventional diagram key.

Recommendations

1. The present study has developed different image analysis techniques by focusing primarily on color features and has been successful to a certain extent. In order to achieve higher accuracy, incorporation of other parameters such as texture, shape, fractal dimensions and lacunarity may be of interest to consider.

2. The experiments carried out in the study, however, relied basically on a constant illuminating condition. Influence of different lighting conditions, particularly in real fields, should therefore further investigated. In addition, effects of camera orientation and position should be also clarified.

3. The present study adopted two classification techniques, i.e. backpropagation neural network and discriminant analysis. Other classification techniques such as self organizing feature map, or support vector machine may be further tested and taking into comparison.

4. This study has demonstrated some shortcoming of conventional diagram key for cassava brown leaf spot disease assessment. It would be advantages to establish a new criterion as well as further develop a reliable equipment for this specific application.

LITERATURE CITED

Aduwo, J.R., E. Mwebaze, and J.A. Quinn, 2010. **Automated vision-based diagnosis of cassava mosaic disease**. Workshop on Data Mining in Agricuture (DMA 2010), Berlin.

Aitkenhead, M.J., I.A. Dalgetty, C.E. Mullins, A.J.S. McDonald and N.J.C. Strachan. 2003. Weed and crop discrimination using image analysis and artificial intelligence methods. **Computers and Electronics in Agriculture** 39: 157-171.

Aroonyadet, N. 2002. **Fruit Maturation Analysis Using Image Processing Technique in Kao Numpung Pummelo.** M.E. Thesis, Kasetsart University.

Bellotti, A.C. 2002. Arthropod Pest. *In*: Hillocks, R.J., J.M. Thresh, and A.C. Bellotti, (Eds), **Cassava: Biology, Production and Utilization.** CAB International, UK.

Bock, C.H., P.E. Parker, A. Z. Cook and T.R. Gottwald. 2008. Visual Rating and The Use of image analysis for assessing different symptoms of circus canker on grapefruit leaves. **Plant Dis.** 92: 530-541.

Bock, C.H., P.E. Parker and T.R. Gottwald. 2009. automated image analysis of the severity of foliar citrus canker symptoms. **Plant Dis.** 93: 660-665.

Burgos-Artizzu, X. P., A. Riberio, M. Guijarro and G. Pajares. 2011. Real-time image processing for crop/weed discrimination in maize fields. **Computers and Electronics in Agriculture** 75: 337-346.

Calvert, L.A. and J.M. Thresh. 2002. The Viruses and Virus Diseases of Cassava. *In*: Hillocks, R.J., J.M. Thresh, and A.C. **Bellotti, (Eds), Cassava: Biology, Production and Utilization.** CAB International, UK.

Camargo, A. and J.S. Smith. 2009a. An image-processing based algorithm to automatically identify plant disease visual symptom. **Biosystems Engineering** 102: 9-21.

Camargo, A. and J.S. Smith. 2009b. Image pattern classification for the identification of disease causing agents in plants. **Biosysterms Engineering** 66: 121-125.

Clifford, H. K., D. R. Shaw, C. E. Watson and K. N. Reddy. 2003. Detecting late-season weed infestations in soybean (*Glycine max*). **Weed Science Society of America** 17(4): 696-704. 2003.

Cui, D., Q. Zhang, M. Li, G. L. Hartman, Y. Zhao 2010. Image processing methods for quantitatively detecting soybean rust from multispectral images. **Biosystems Engineering** 107(3): 186-193.

El-Helly, M., A.A. Rafea and S. EI-Gammal. 2003. An integrated image processing system for leaf disease detection and diagnosis, pp. 1180-1195. *In* **Proceeding of Indian International Conference on Artificial Intelligence.** 18-20 December 2003, India.

Falvey, L. 2000. **Thai Agriculture: Golden cradle of millennia.** Kasetsart University Press, Bangkok, Thailand.

Gonzalez, R.C. and R.E. Woods. 2010. **Digital Image Processing.** Pearson, USA.

Gonzalez, R.C., R.E. Woods. and S.L. Eddins. 2004. **Digital Image Processing Using MATLAB®.** Pearson, USA .

Gulhane, V. A. and A. A. Gurjar. 2011. Detection of diseases on cotton leaves and its possible diagnosis. **International Journal of Image processing** 5(5): 2011.

Gulhane, V. A., and A. A. Gurijar. 2012. Disease classification on cotton leaves by advance digital image processing approach, pp. 9-12. *In* **IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2012) .** March 2012, Foundation of Computer Science, New York, USA.

Hague, T., N.D. Tillett. H. Wheeler. 2006. Automated crop and weed monitoring in widely spaced cereals. **Precision Agric.** 7: 21-32.

Hemming, R.J., and K. Wydra. 2002. Bacterial, fungal and nematode diseases. *In*: Hillocks, R.J., J.M. Thresh, and A.C. Bellotti, (Eds), Cassava: **Biology, Production and Utilization.** CAB International, UK.

Hillocks, R.J., and K. Wydra. 2002. Bacterial, fungal and nematode disease. *In*: Hillocks, R.J., J.M. Thresh, and A.C. Bellotti, (Eds), **Cassava: Biology, Production and Utilization.** CAB International, UK.

Jarimopas, B. and N. Jaisin. 2008. An experimental machine vision system for sorting sweet tamarind. **Journal of Food Engineering** 89: 291-297.

Kampanich, W. 2003. **Investigation on Screening Methods for Cassava Resistant Varieties to Brown Leaf Spot Disease (*Cercospora henningsii* Allescher).** M.S. Thesis, Kasetsart University.

Kiani, S. and A. Jafari. 2012. Crop detection and positioning in the field using discriminant analysis and neural networks based on shape features. **J.Agr. Sci. Tech.** 14: 755-765.

Kurtulmus, F., W. S. Lee, and A. Vardar. 2011. Green citrus detection using 'eigenfruit', color and circular gabor texture features under natural outdoor conditions. **Computers and Electronics in Agriculture** 78: 140-149.

Meyer, G. E. and J. C. Neto. 2008. Verification of color vegetation indices for automated crop imaging applications. **Computers and Electronics in Agriculture** 63: 282-293.

Motameni, H., M. Norouzi, M. Jahandar and A. Hatami. 2007. Labeling method in steganography, pp. 349-354. *In* **Proceedings of World Academy of Science, Engineering and Technology.** 24 October 2007, France.

Otsu, N. 1979. A threshold selection method from gray-level histograms. **IEEE Trans. Sys., Man., Cyber.** 9(1): 62-63.

Onyango, C.M. and J.A. Marchant. 2003. Segmentation of row crop plants from weeds using colour and morphology. **Computers and Electronics in Agriculture** 39: 141-155.

Patil, S., and S.K. Bodhe. 2011. Leaf disease severity measurement using image processing. **International Journal of Engineering and Technology** 3(5): 297-301.

Pérez, A.J., F. López, J.V. Benlloch and S. Christensen. 2000. Colour and shape analysis technique for weed detection in cereal fields. **Computers and Electronics in Agriculture** 25: 197-212.

Sannakki, S.S., V. S. Rajpurohit, V.B. Nargund, A. Kumar R and P. S. Yallur. 2011. **Leaf disease grading by machine vision and fuzzy logic.** J.Comp.Tech.Appl. 2(5): 1709-1716.

Sena Jr, D.G., F.A.C. Pinto, D.M. Queriroz and P.A. Viana. 2003. Fall armyworm damaged maize plant identification using digital images. **Biosystems Engineering** 85(4): 449-454.

Sermsak, R. and H. Boonjung. 2007. Appication of digital image for leaf area index estimation of soybean. **Suranaree J. Sci. Technol.** 14(2): 163-172.

Solomon, C. and T. Breckon. 2011. **Fundamentals of Digital Image Processing.** Wiley-Backwell, Singapore.

Teri, J.M., H.D. Thurston and J.C. Lozano. 1977. Effect of brown leaf spot and Cercospora leaf blight on cassava production. **Tropical Agriculture** 57 (3): 239-243.

The Math Works. 2010. **Image Processing Toolbox™ User's Guide.** The Math Works, Natick. (Mimeographed)

Thammabhutra, A. 2005. **Analysis of Maturity of "Monthong" Durian Using Image Processing of Fruit Stem.** M.E. Thesis, Kasetsart University.

Vanichbancha, K. 2010. **Advance statistical analysis by SPSS for windows.** Chulalongkorn University, Bangkok. in Thai.

Weizheng, S., Yachun, W., Zhanliang, C., and Hongda, W. 2008. Grading Method of Leaf Spot Disease Based on Image Processing, pp. 491-494. *In* **Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 06**. 12- 14 December 2008, CSSE. IEEE Computer Society, Washington DC., USA.

Wijekoon, C.P., P.H. Goodwin, T. Hsiang. 2008. Quantifying fungal infection of plant leaves by digital image analysis using Scion Image software. **Journal of Microbiological Methods** 74: 94-101.

Woebbecke, D.M., G.E. Meyer, K. Von Bargen and D.A. Mortensen. 1995. Color indices for weed identification under various soil, residue, and lighting conditions. **Transactions of the ASAE** 38 (1): 259-269.

Wydra, K., and V. Verdier. 2002. Occurrence of cassava diseases in relation to environmental, agronomic and plant characteristics. **Agriculture Ecosystems and Environmental** 93: 211-226.

Yang, C.-C., S.O. Prasher and J.A. Landry. 2000. Application of artificial neural networks in image recognition and classification of crop and weeds. **Canadian Agricultural Engineering** 42: 147-152.

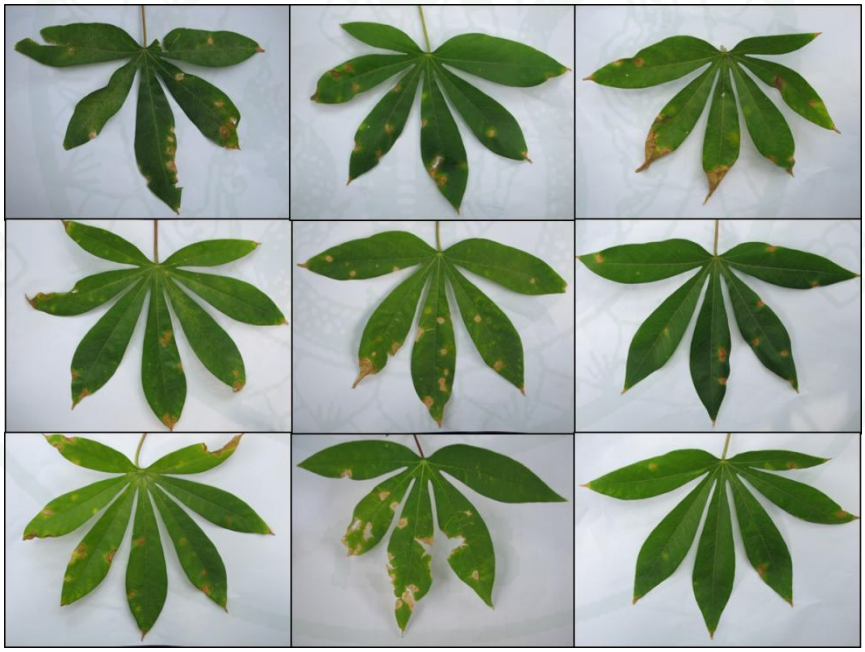
Yang, C.-C., S.O. Prasher and J.A. Landry. 2002. Weed recognition in corn fields using Back-Propagation neural network Models. **Canadian Biosystems Engineering** 44, 7.15-7.22.

APPENDICES

Appendix A

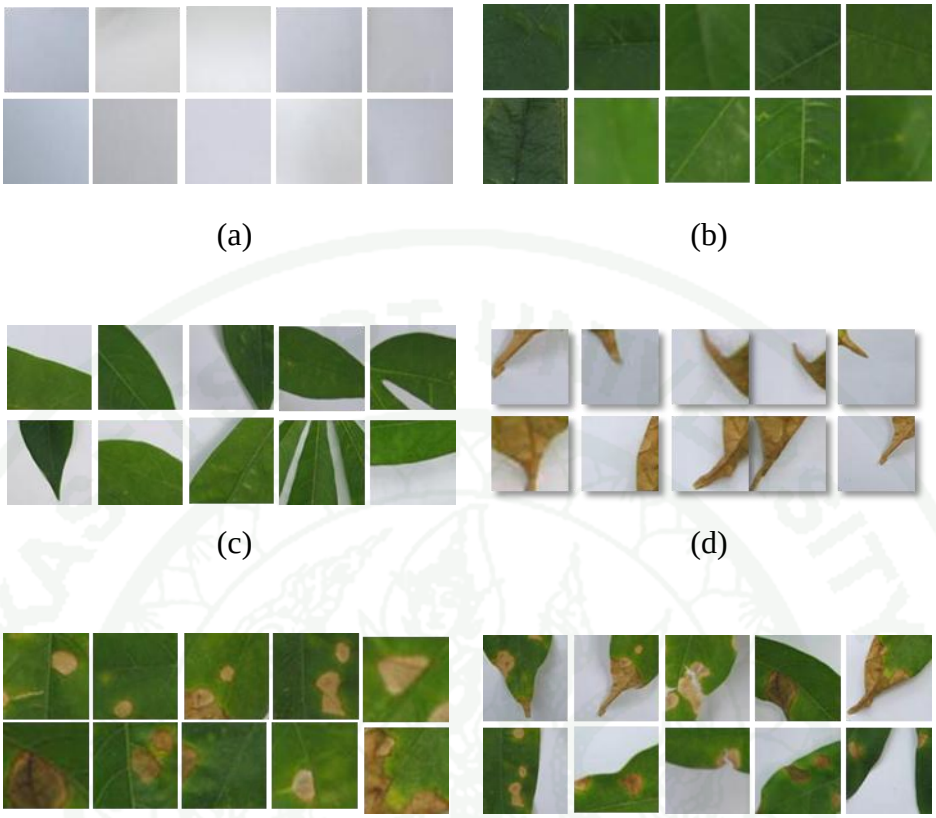
Images of cassava leaves used in experiments

(a)



(b)

Appendix Figure A1 Images of cassava leaf for preliminary experiments: (a) healthy leaves, and (b) disease leaves.



(a) (b)

(c) (d)

(e) (f)

Appendix Figure A2 Cropped image for preliminary experiments: (a) Background only, (b) Leaf only, (c) Background and leaf, (d) Background and spot, (e) Leaf and spot, and (f) Background, leaf and spot.

Appendix Figure A3 Images of healthy cassava leaves.

Appendix Figure A3 (Continued)

Appendix Figure A3 (Continued)

Appendix Figure A3 (Continued)

Appendix Figure A3 (Continued)

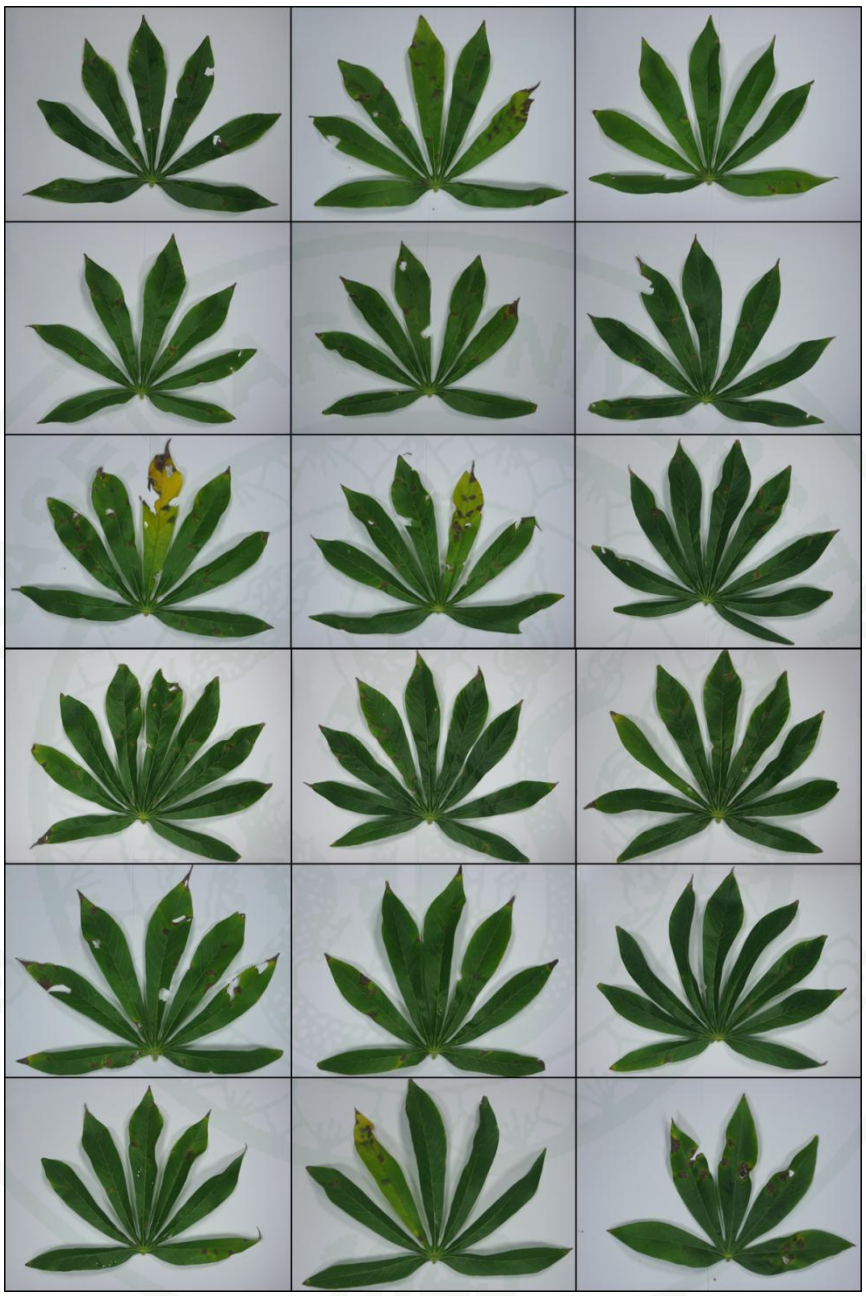
Appendix Figure A4 Images of diseased cassava leaves.

Appendix Figure A4 (Continued)

Appendix Figure A4 (Continued)

Appendix Figure A4 (Continued)

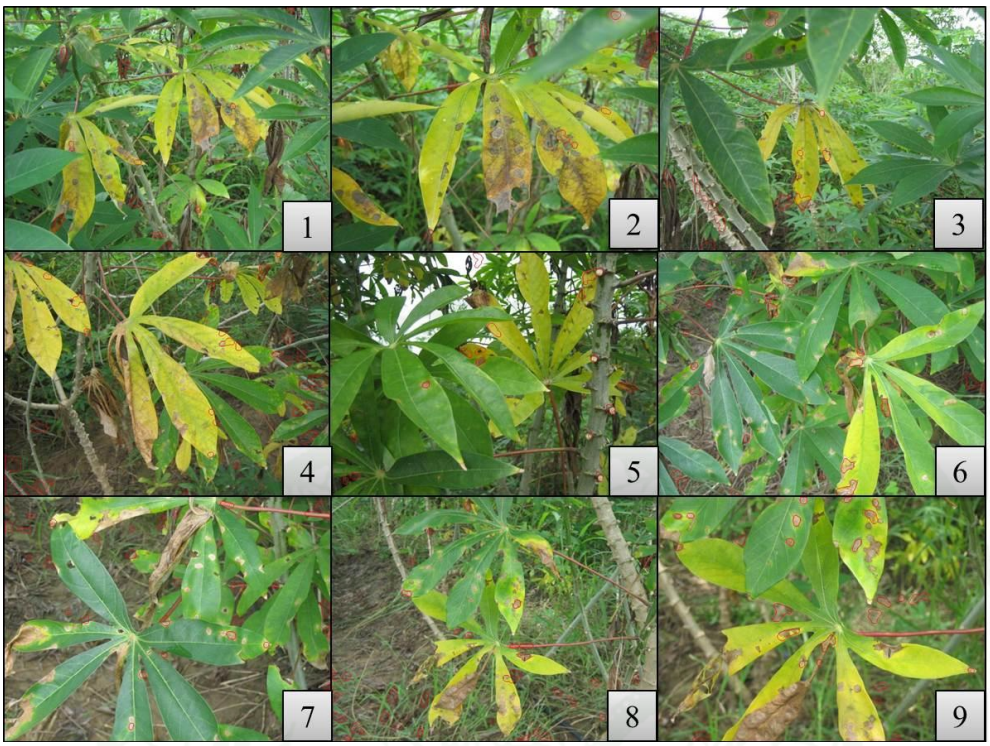
Appendix Figure A4 (Continued)



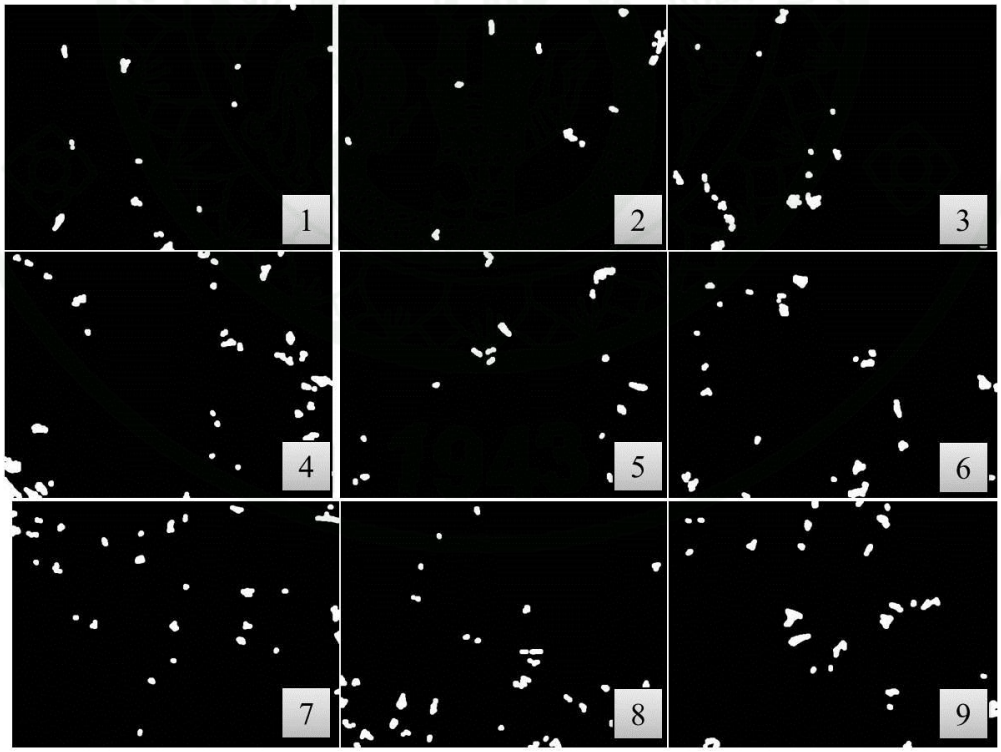
Appendix Figure A5 Images of cassava leaf for disease level assessment.

Appendix Figure A5 (Continued)

Appendix Figure A5 (Continued)

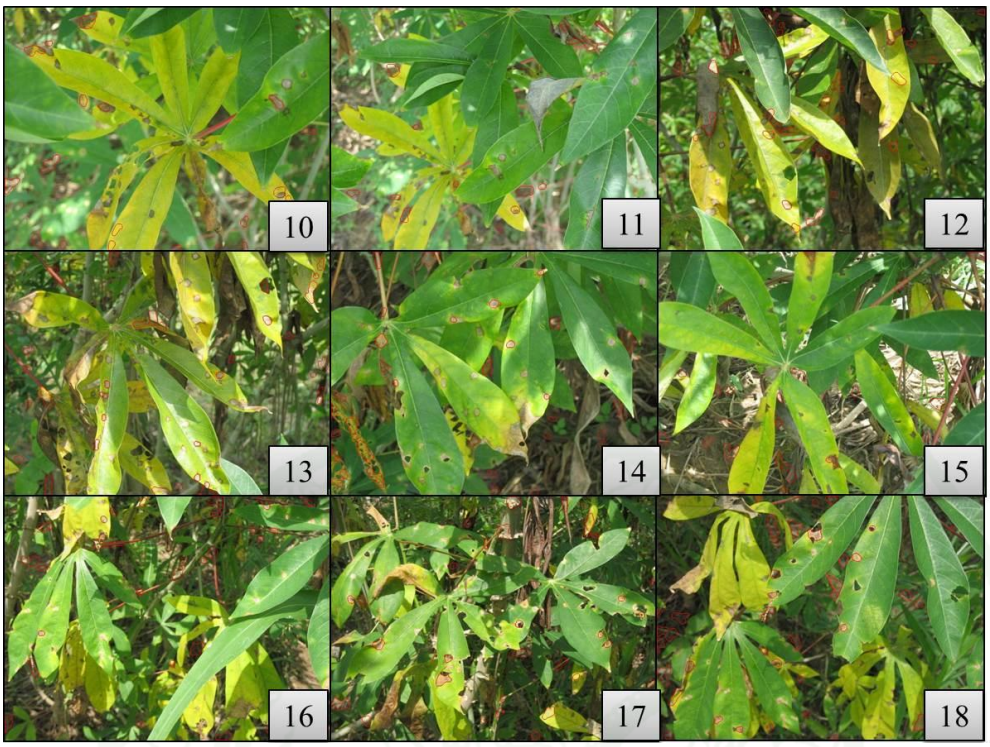


(a)

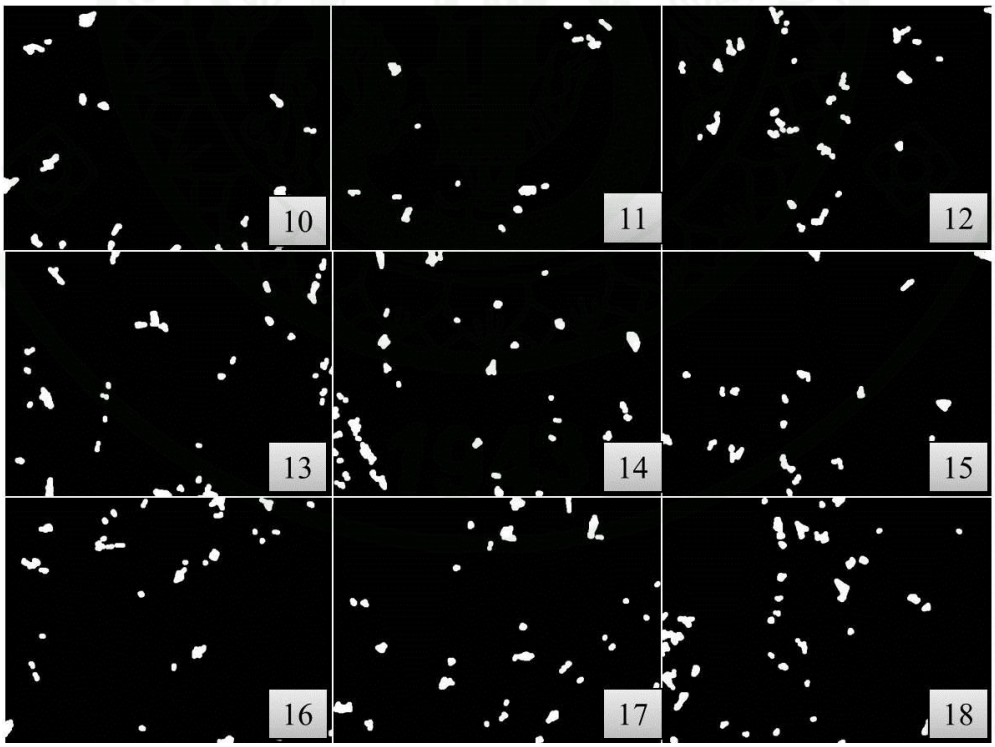


(b)

Appendix Figure A6 Results of spot detection: (a) Original image, and (b) Binary image given by AG I.

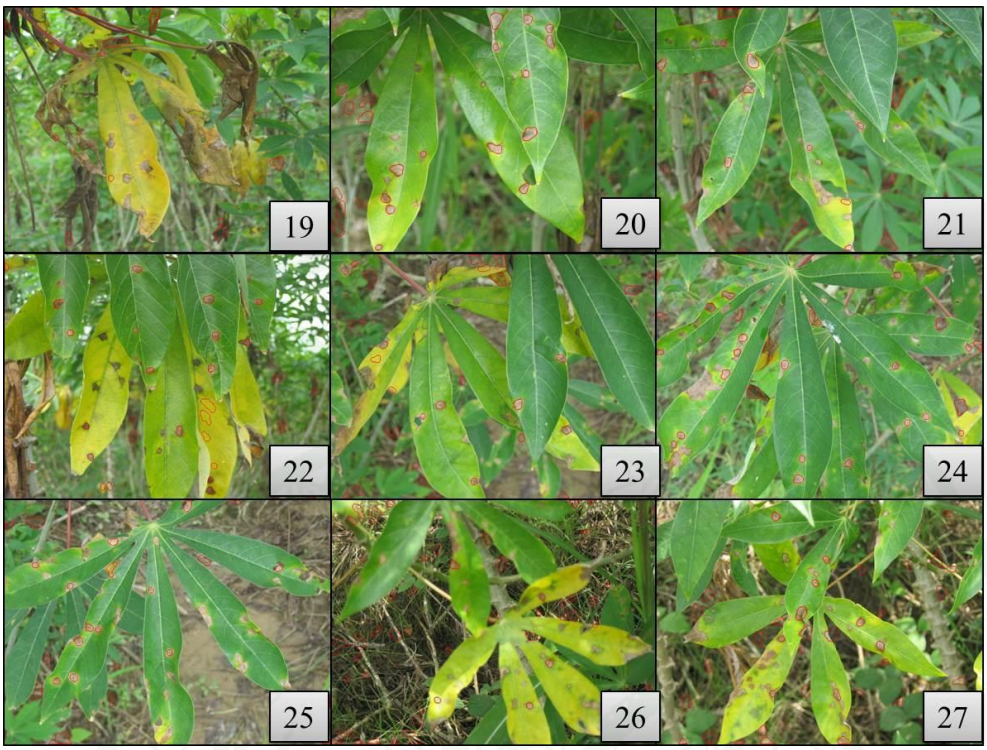


(a)

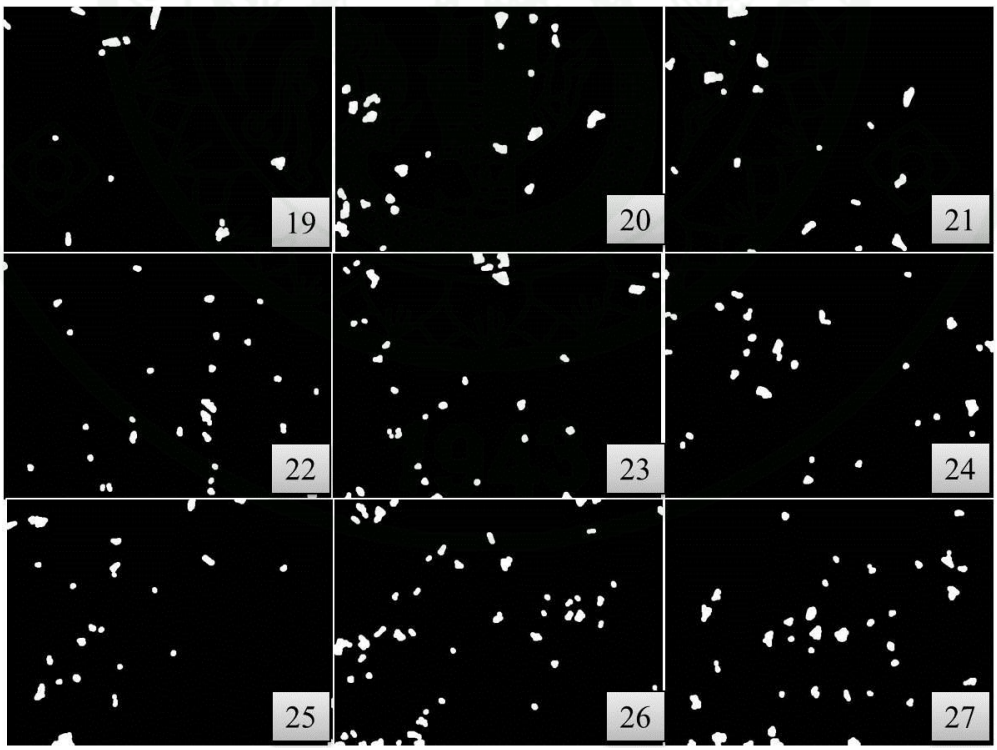


(b)

Appendix Figure A6 (Continuted)

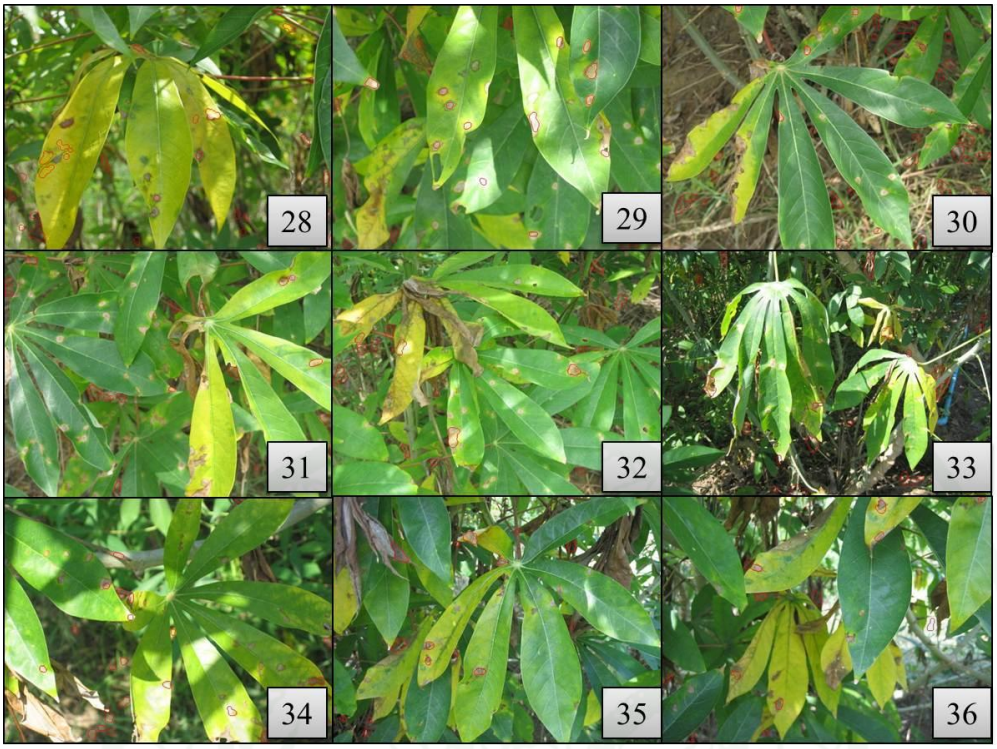


(a)

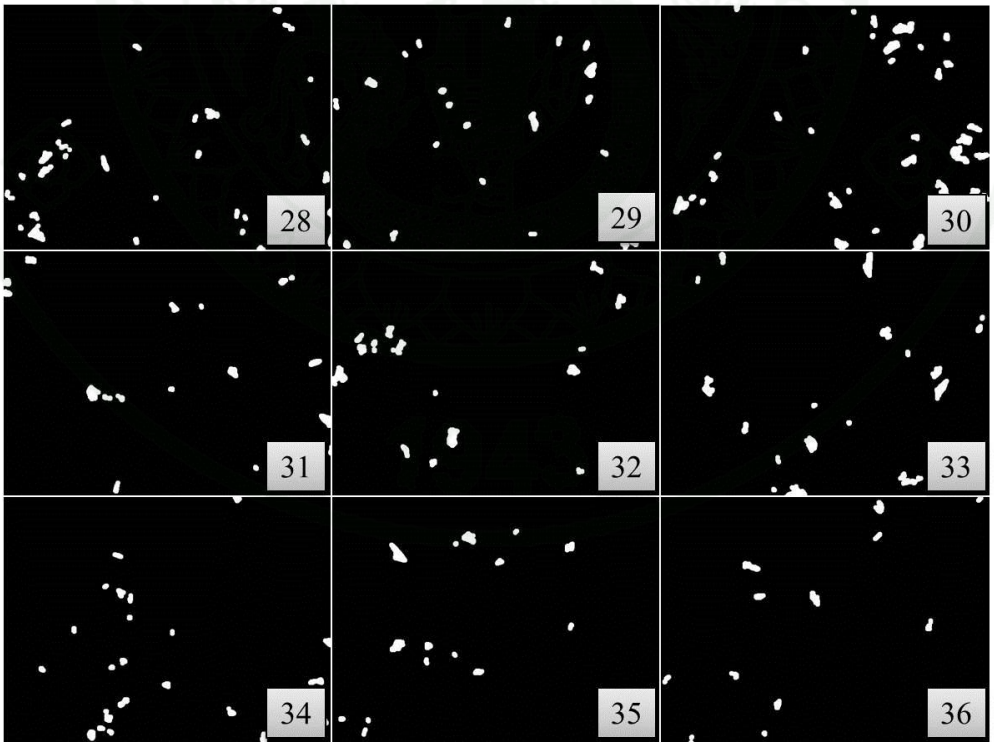


(b)

Appendix Figure A6 (Continuted)

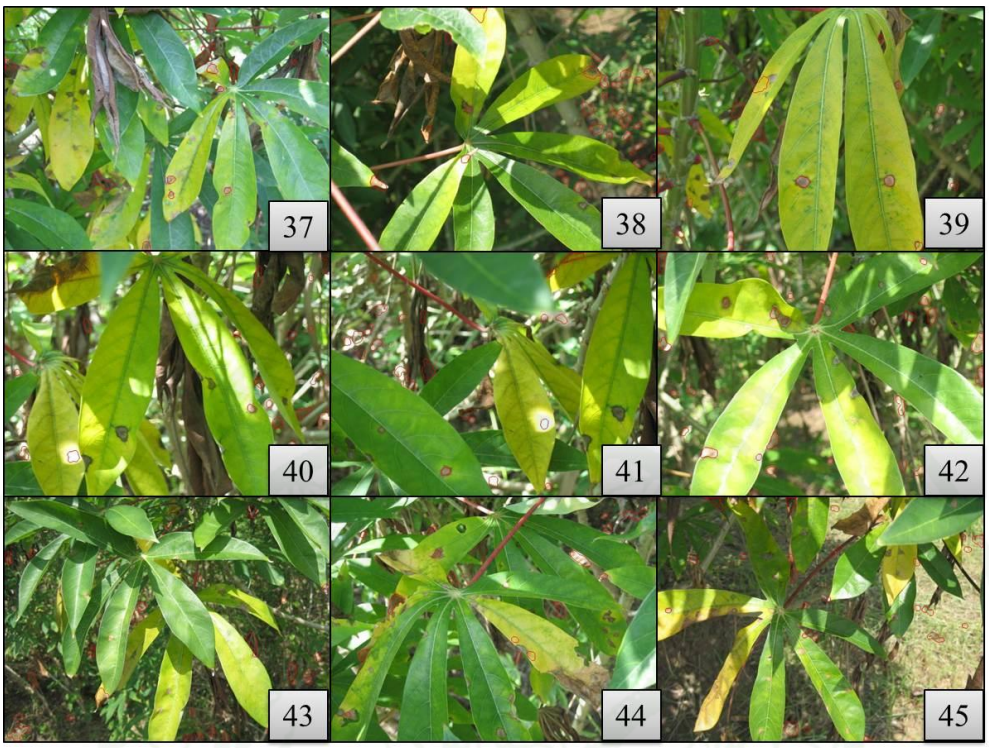


(a)

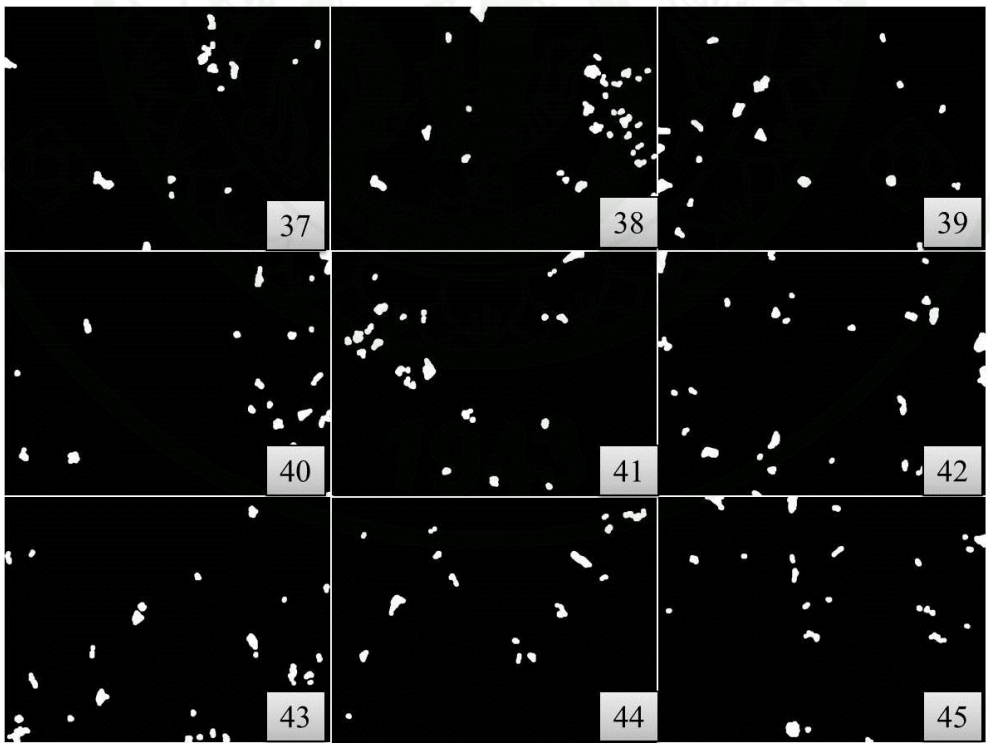


(b)

Appendix Figure A6 (Continuted)

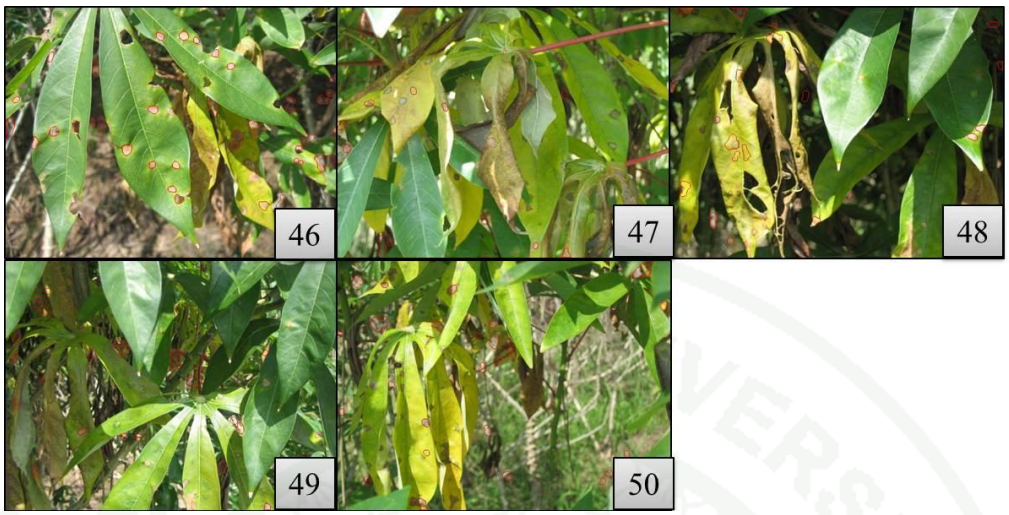


(a)

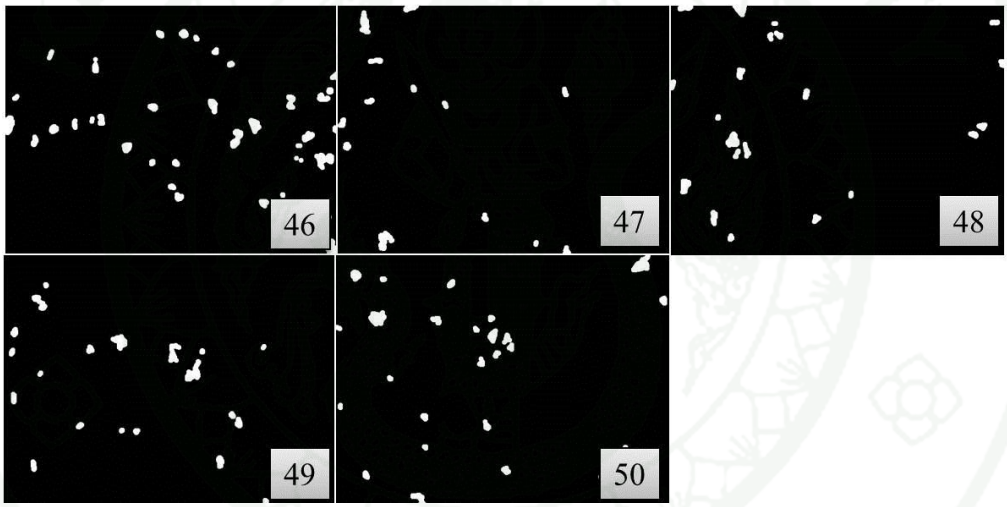


(b)

Appendix Figure A6 (Continuted)

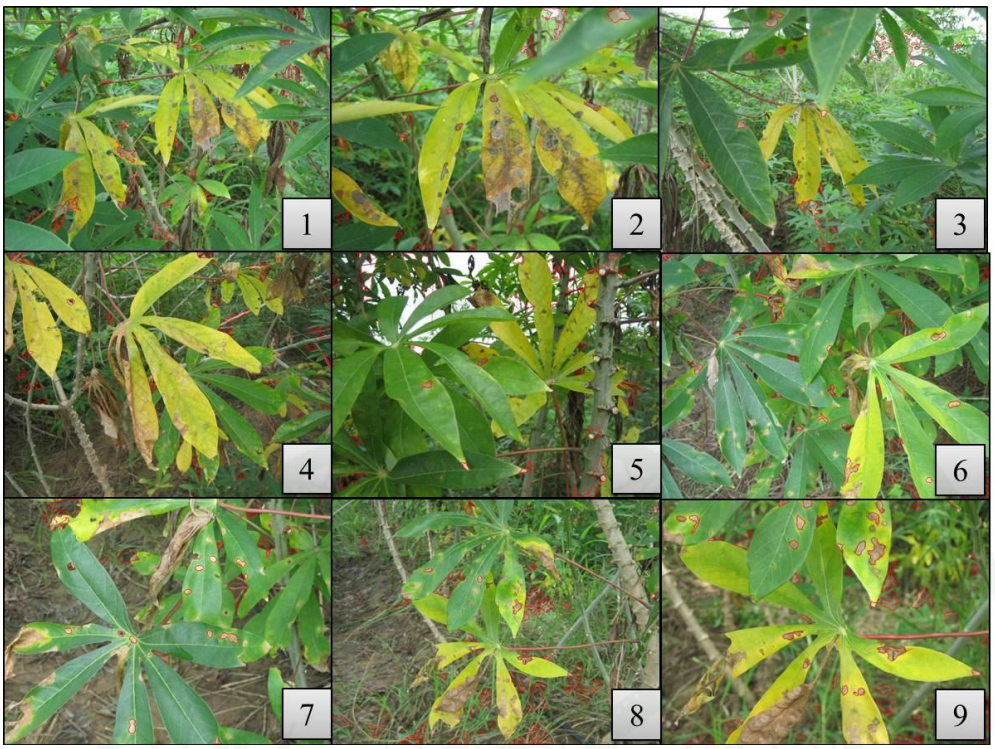


(a)

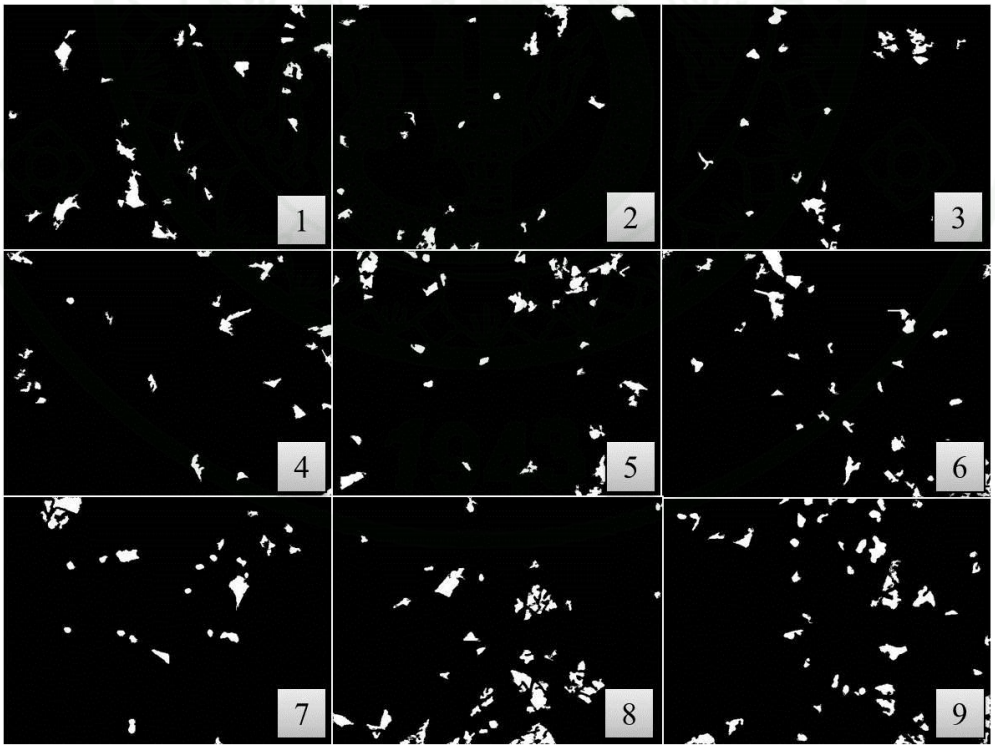


(b)

Appendix Figure A6 (Continuted)

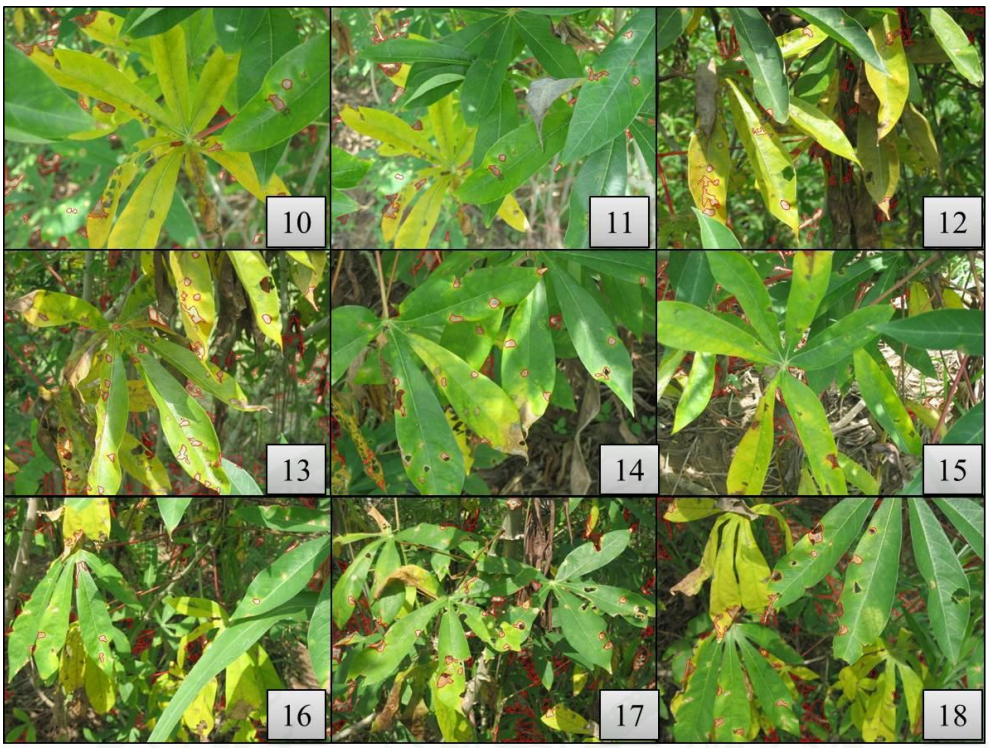


(a)

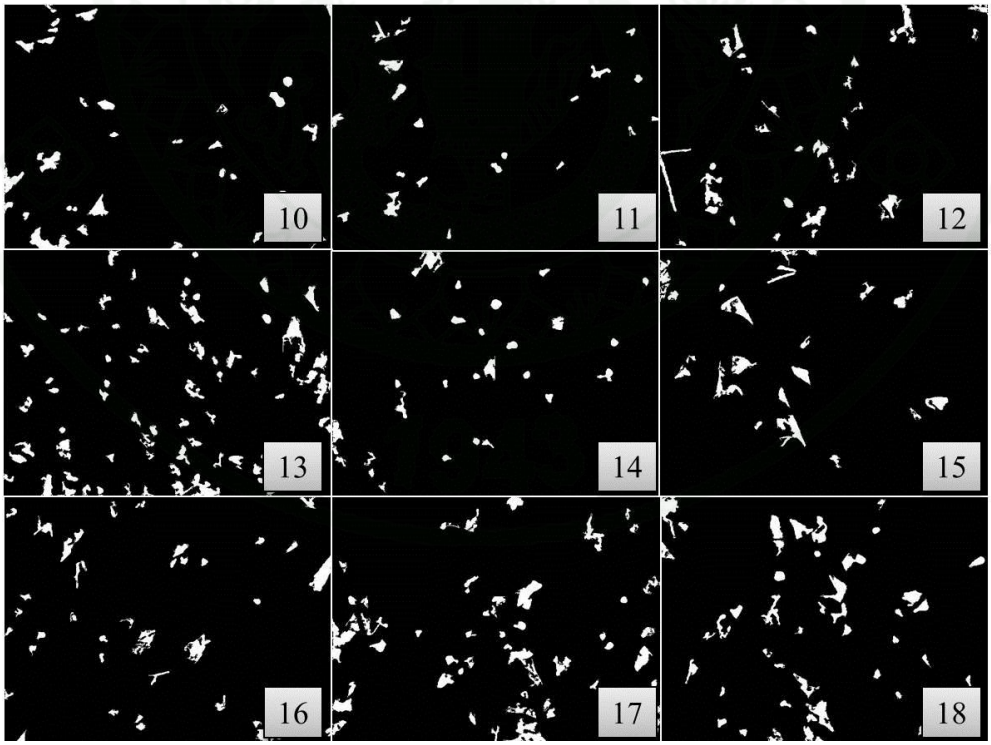


(b)

Appendix Figure A7 Results of spot detection: (a) Original image, and (b) Binary image given by AG II.

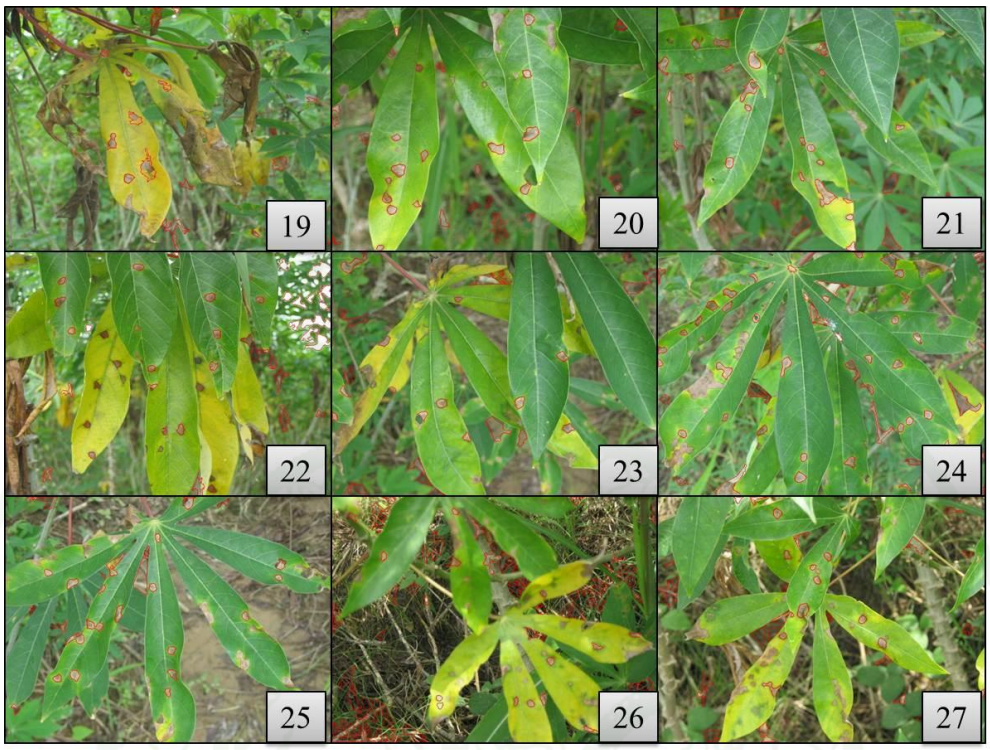


(a)

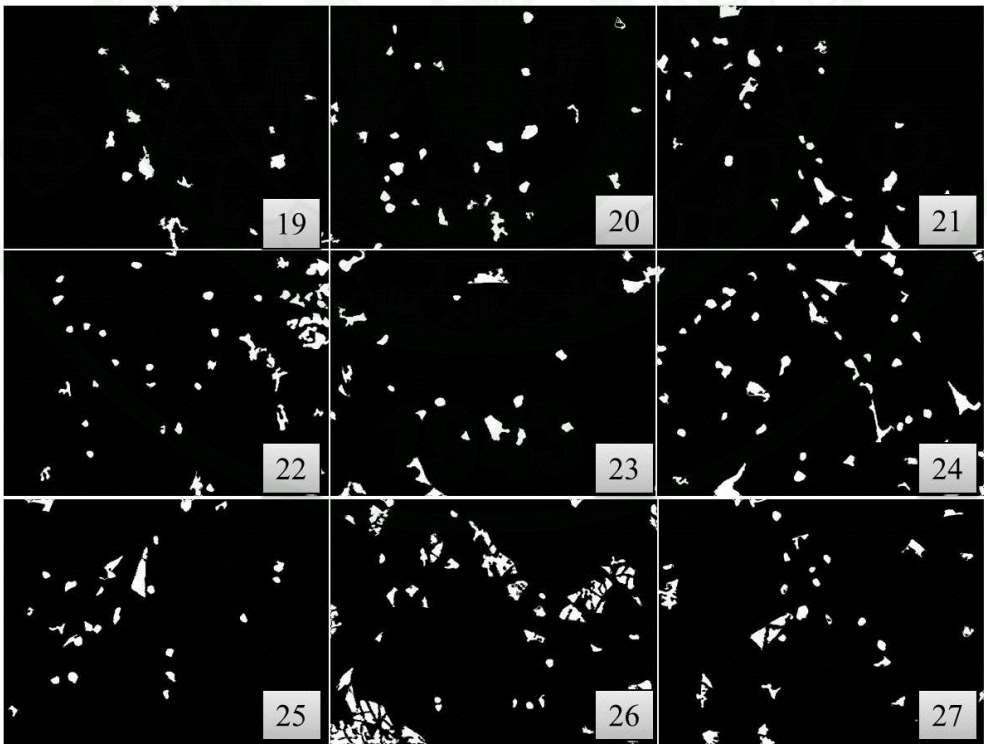


(b)

Appendix Figure A7 (Continuted)

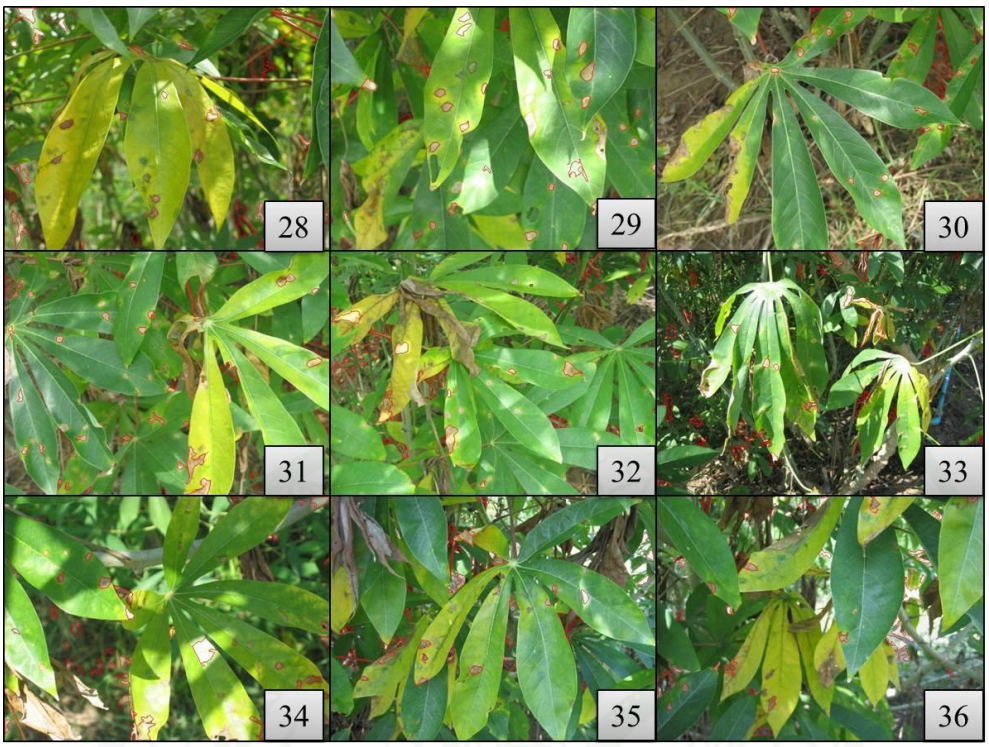


(a)

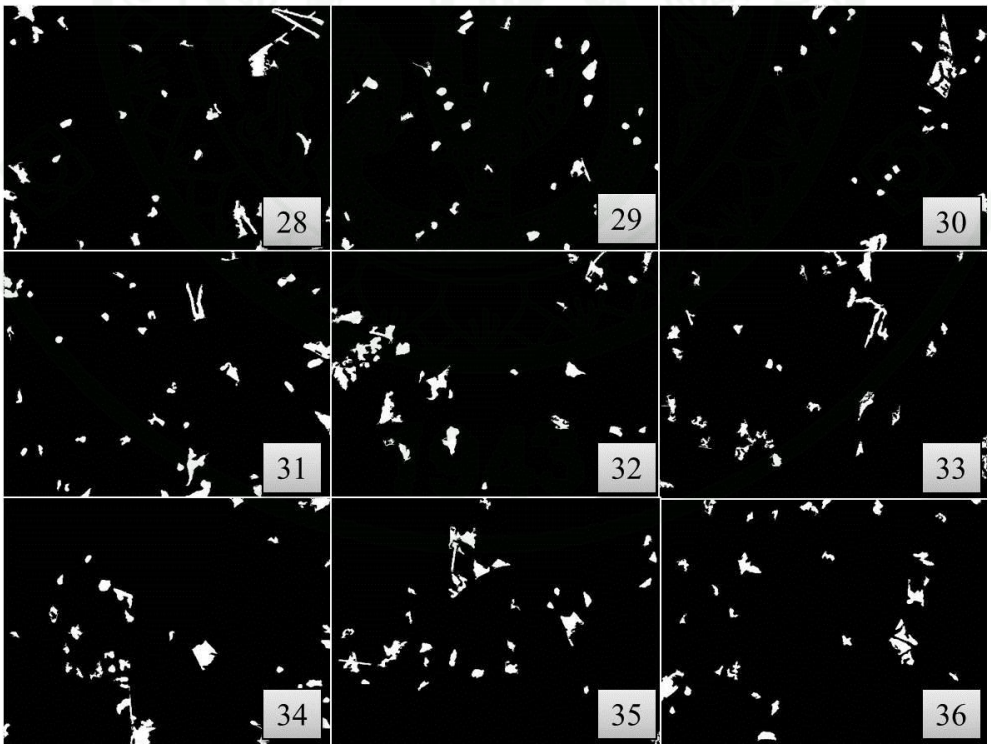


(b)

Appendix Figure A7 (Continuted)



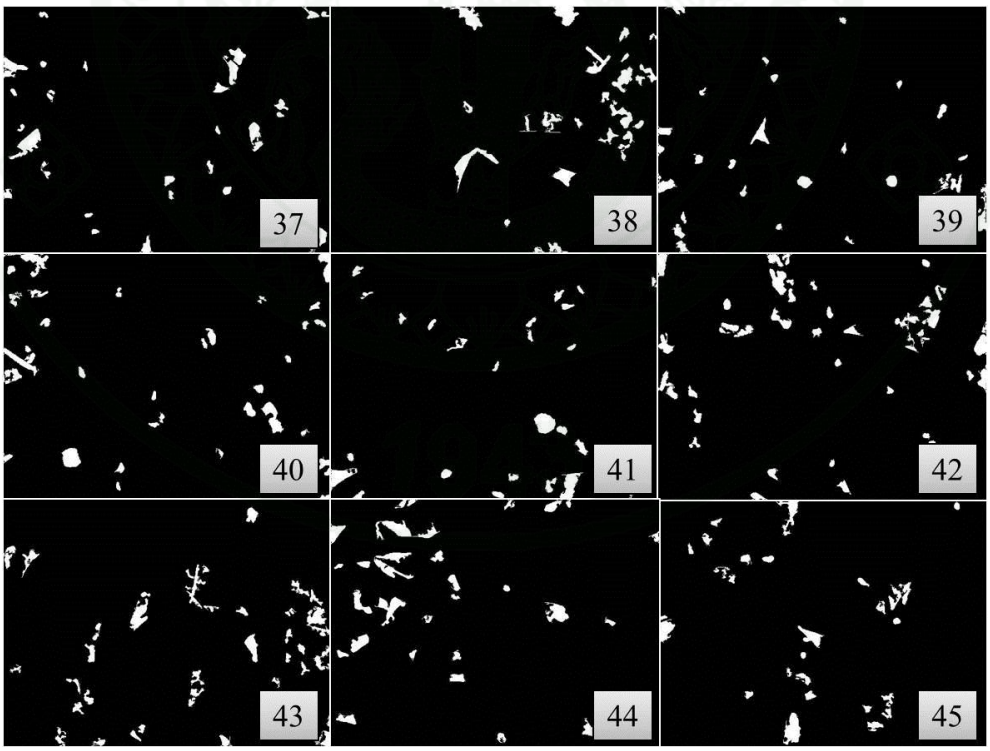
(a)



(b)

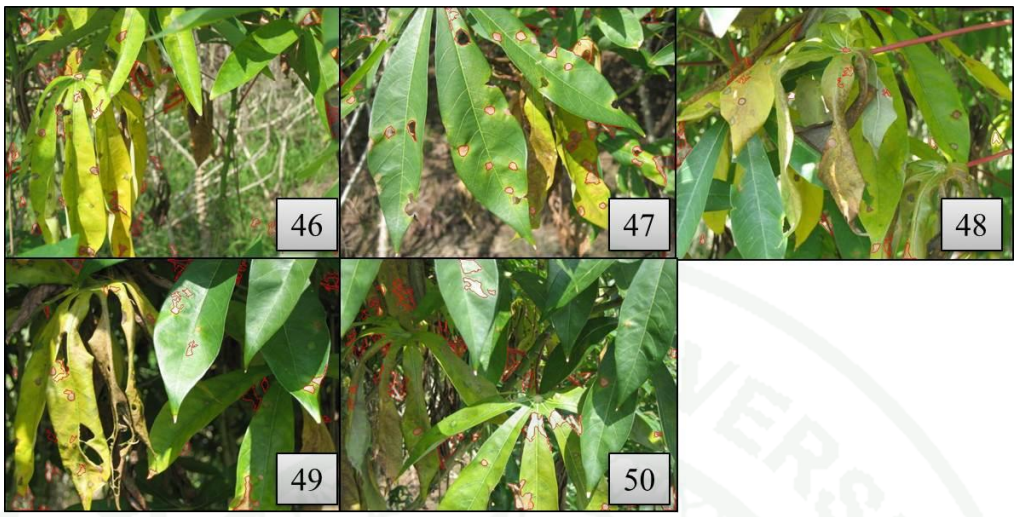
Appendix Figure A7 (Continuted)

(a)

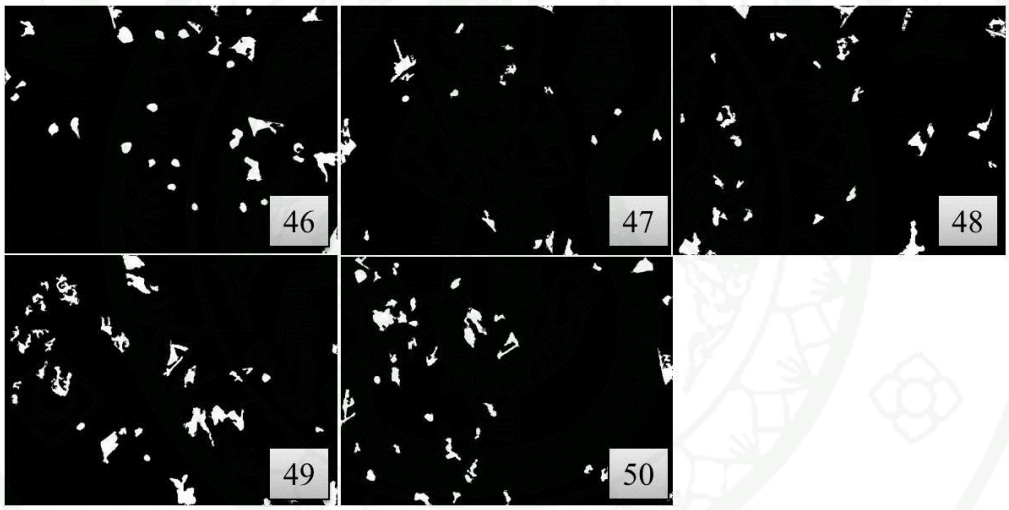


(b)

Appendix Figure A7 (Continuted)

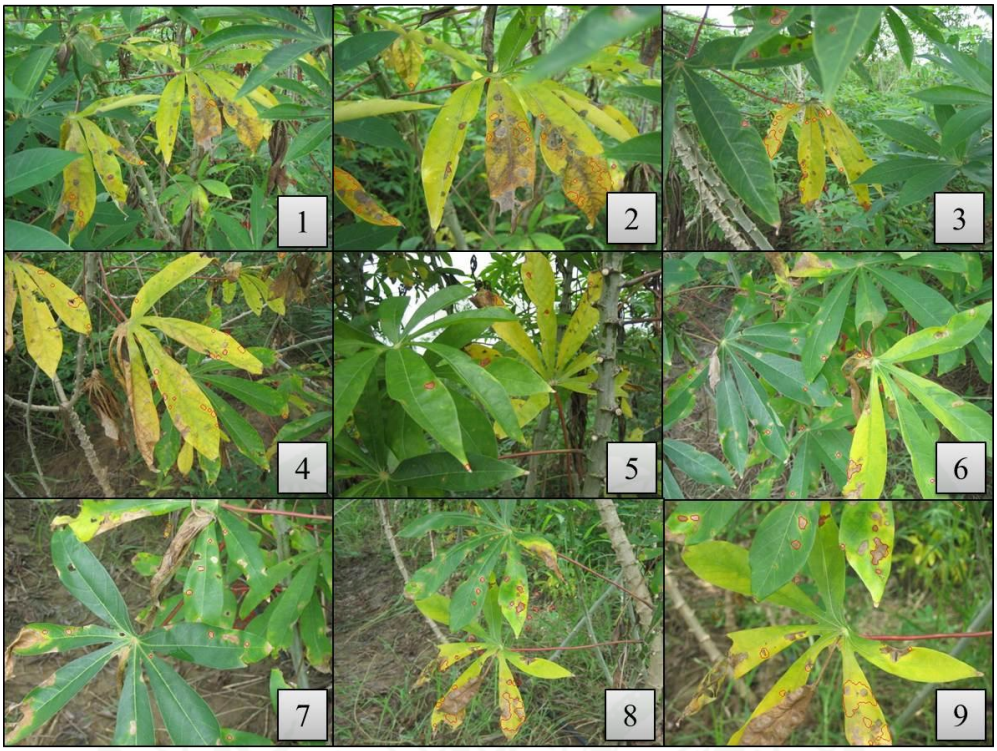


(a)

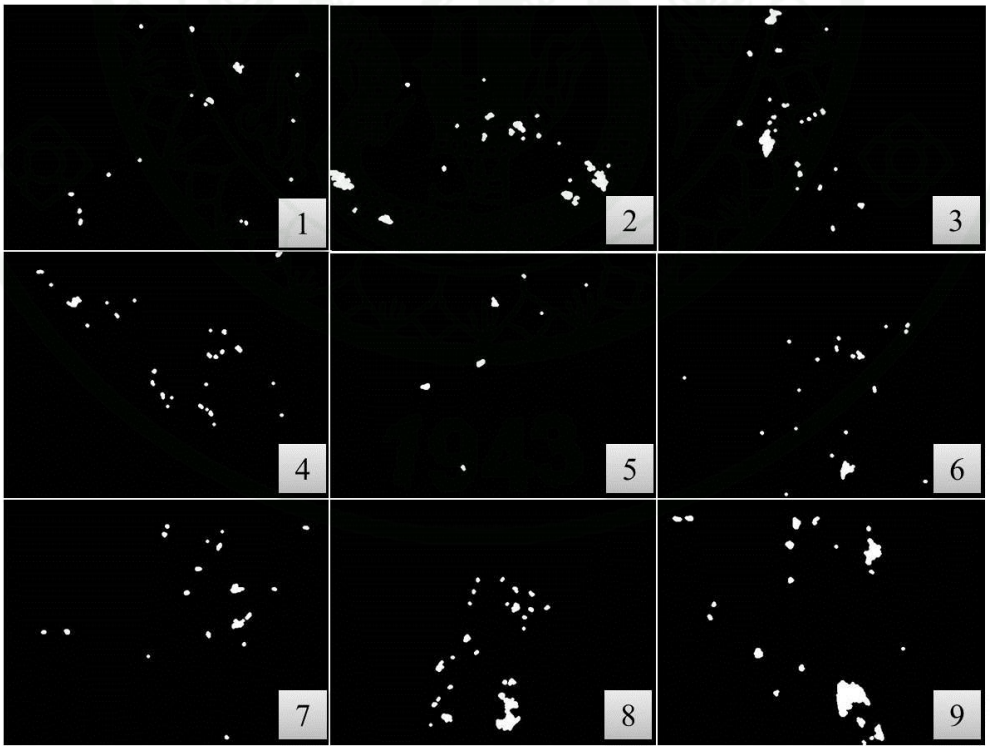


(b)

Appendix Figure A7 (Continuted)

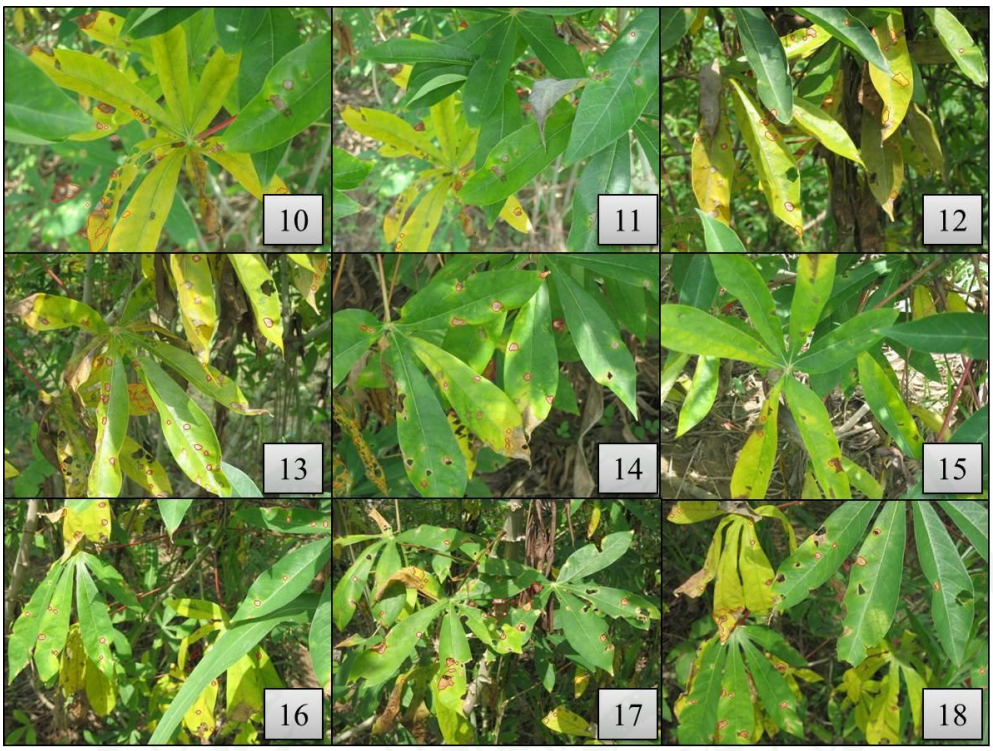


(a)

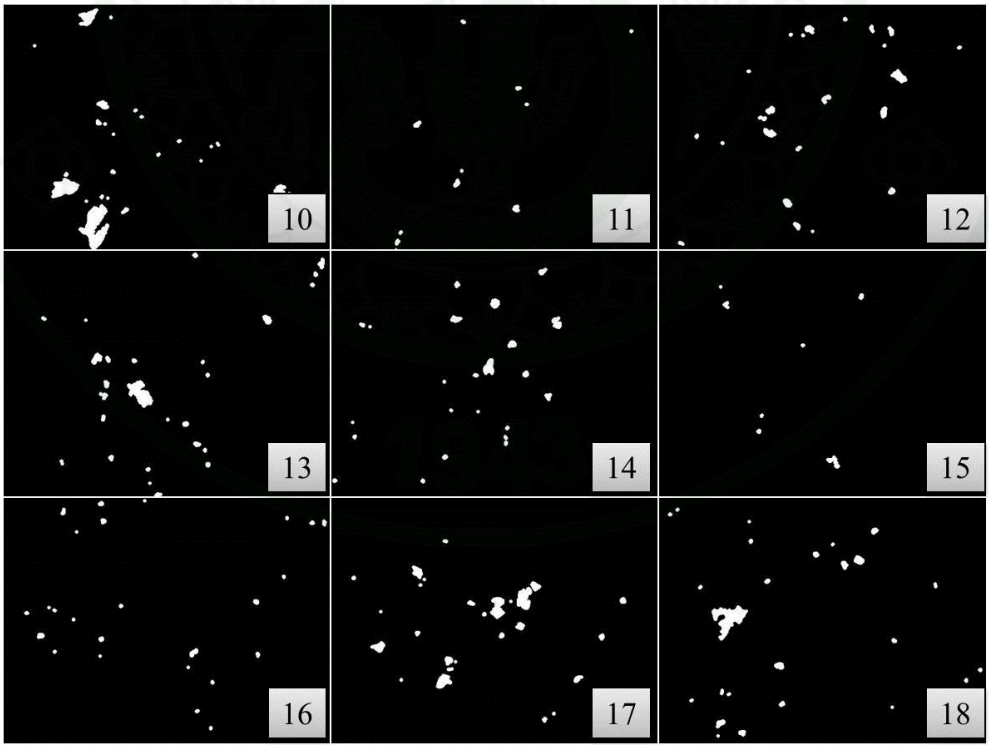


(b)

Appendix Figure A8 Results of spot detection: (a) Original image, and (b) Binary image given by AG III.

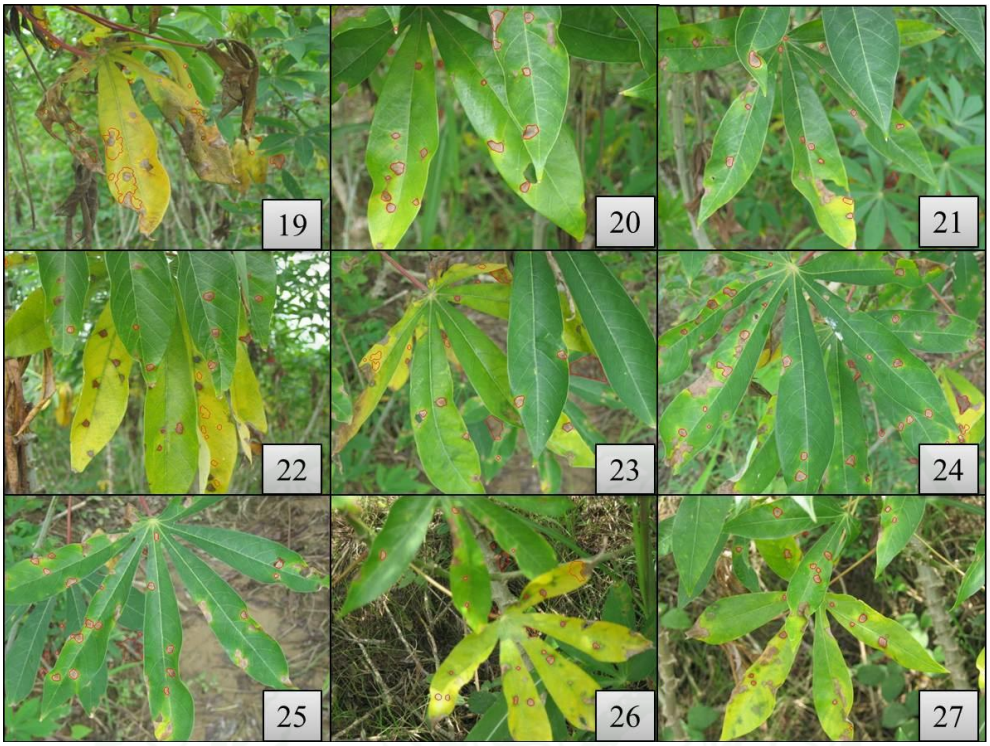


(a)

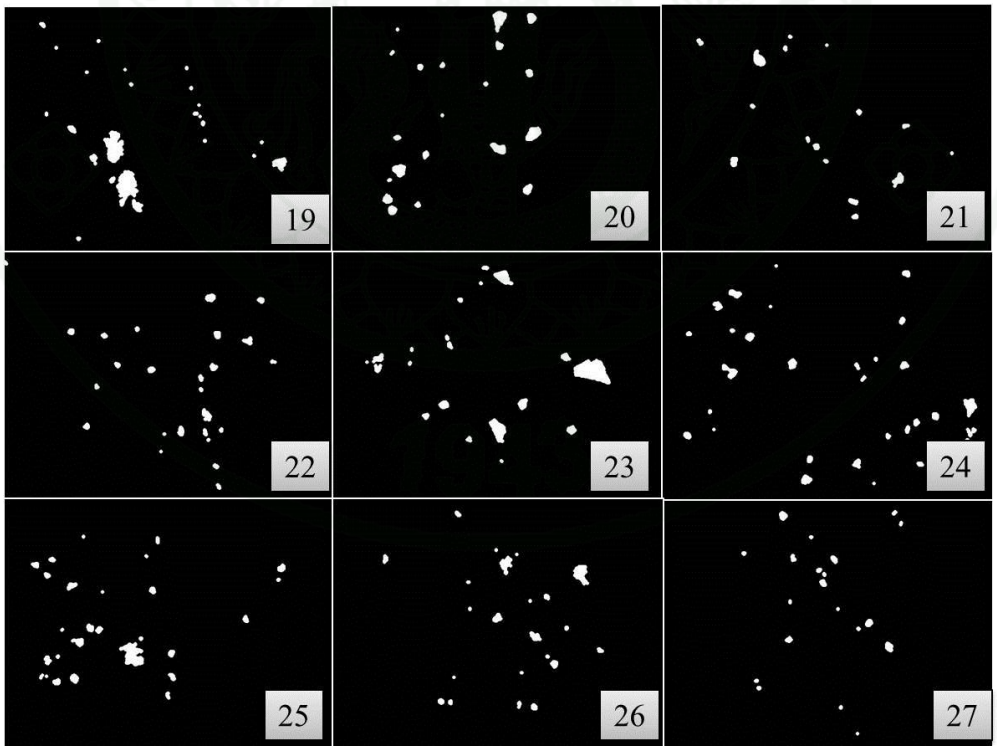


(b)

Appendix Figure A8 (Continuted)

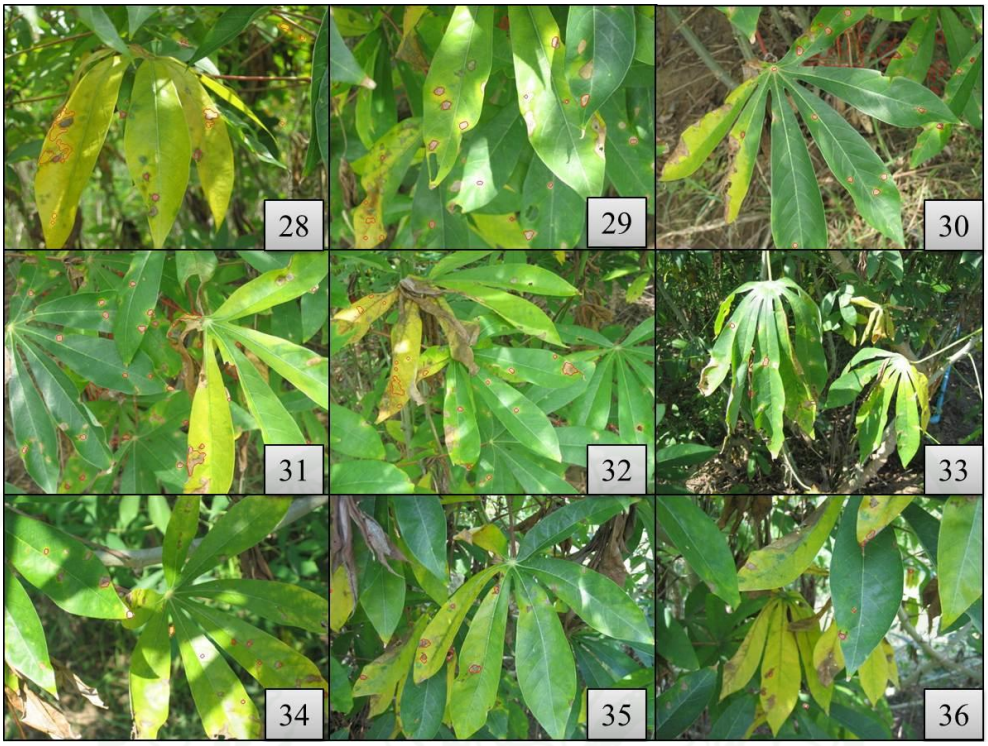


(a)

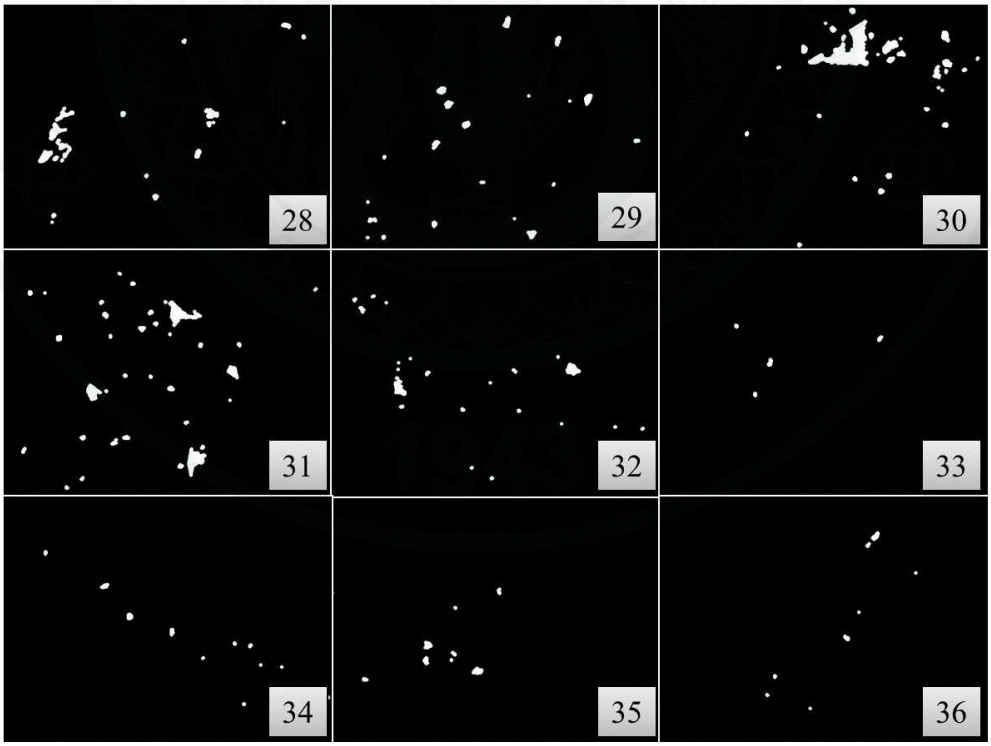


(b)

Appendix Figure A8 (Continuted)

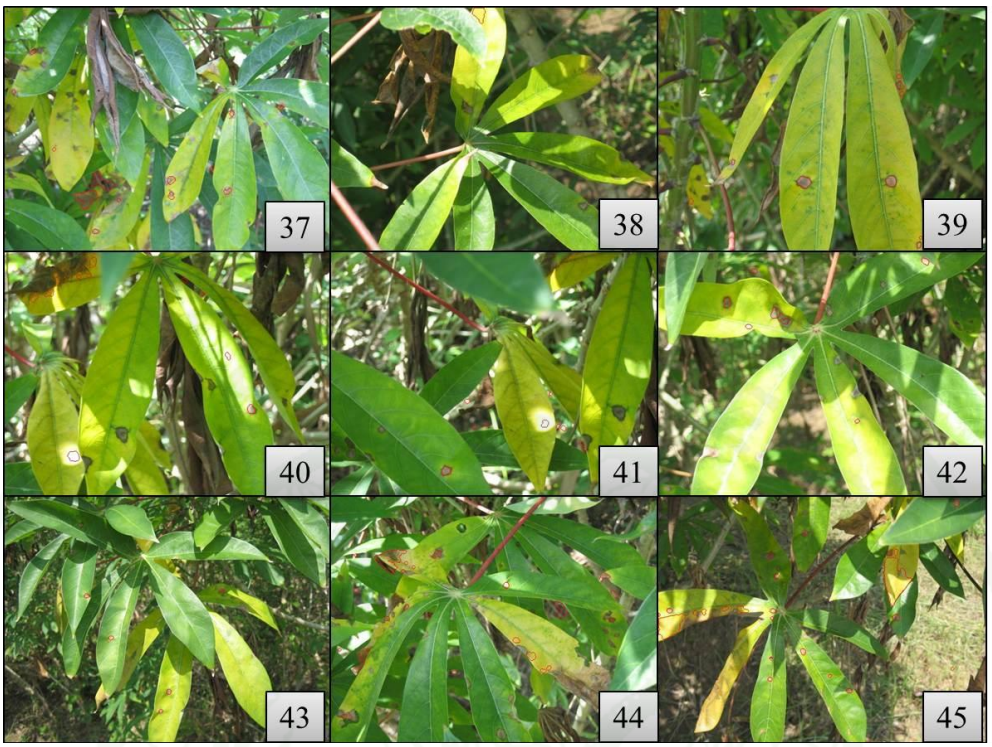


(a)

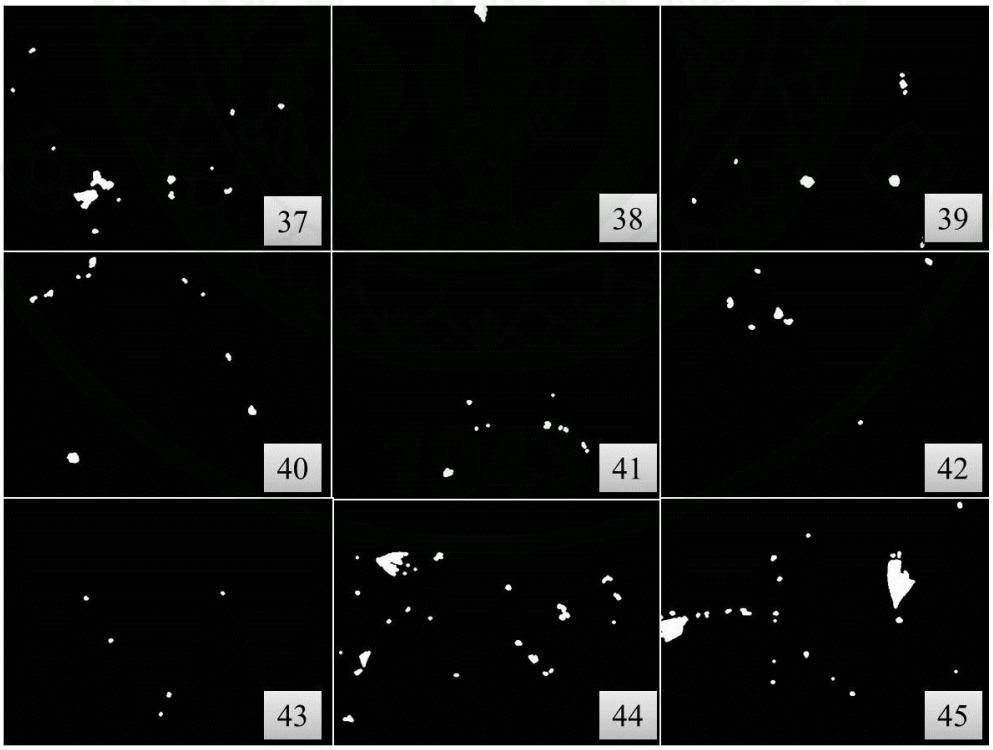


(b)

Appendix Figure A8 (Continuted)

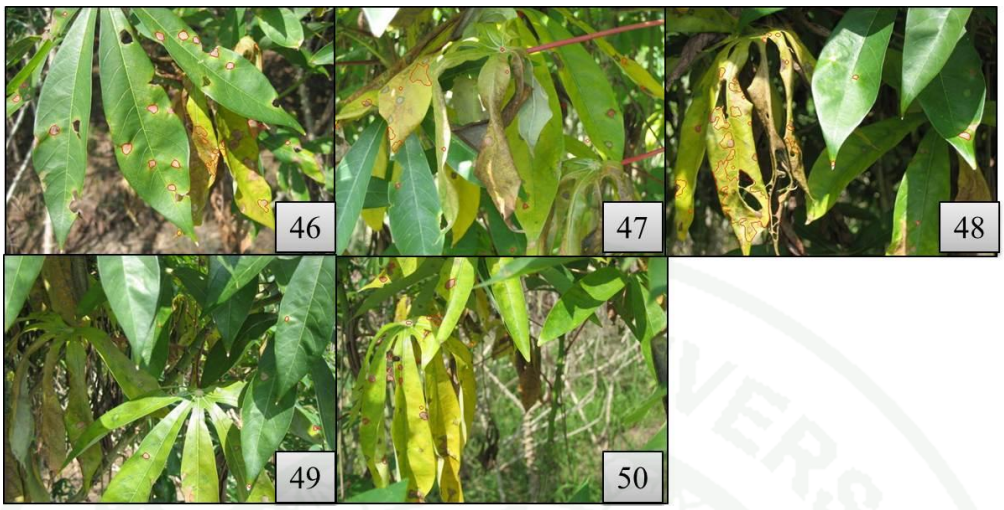


(a)

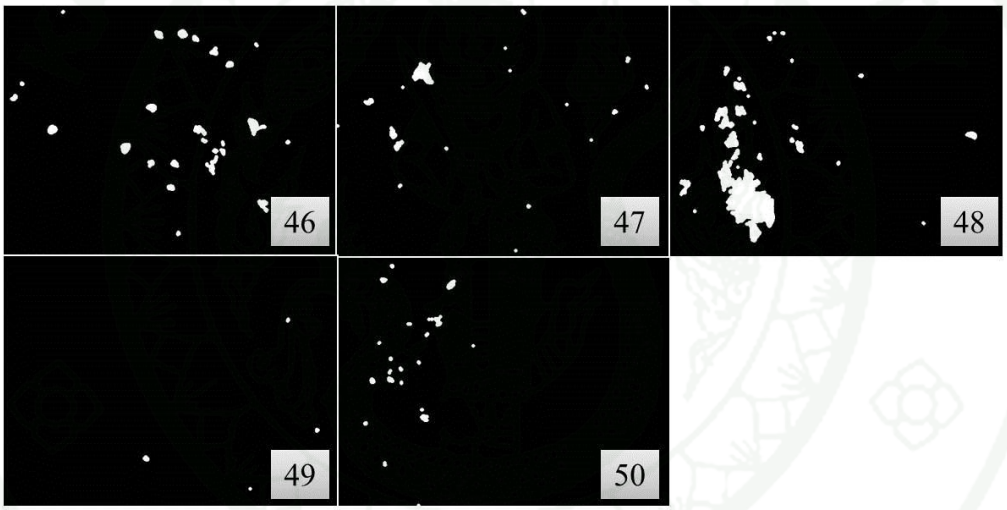


(b)

Appendix Figure A8 (Continuted)



(a)



(b)

Appendix Figure A8 (Continuted)

Appendix B

Source codes for programming with MATLAB

1. Source code for preliminary experiment on classification of healthy and diseased leaves under controlled background condition

1.1 Quantification of RGB values

```
A = imread (‘cropped image.jpg’);
redb1 = A (:,:,1);
greenb1= A (:,:,2);
blueb1 = A (:,:,3);
xlswrite ('redb1.xls',redb1);
xlswrite ('greenb1.xls',greenb1);
xlswrite ('blueb1.xls',blueb1);
```

1.2 Computation for means and standard deviations of RGB

```
%Red calculation
Total_redb1=[0];
Mean_redb1=[0];
for i=1:80
   for j=1:80
      Total_redb1(1,1)= Total_redb1(1,1)+ redb1(i,j);
   end
end
Mean_redb1 = Total_redb1(1,1)/6400;
Total1_redb1=[0];
for i =1:80
   for j = 1:80
   Total1_redb1(1,1) = Total1_redb1+(redb1(i,j)-Mean_redb1)^2;
   end
end
Std_redb1 = (Total1_redb1(1,1)/6400)^0.5;
%Green calculation
```

```
Total_greenb1=[0];
Mean_greenb1=[0];
for i=1:80
   for j=1:80
      Total_greenb1(1,1)= Total_greenb1(1,1)+ greenb1(i,j);
   end
end
Mean_greenb1 = Total_greenb1(1,1)/6400;
Total1_greenb1=[0];
for i =1:80
   for j = 1:80
   Total1_greenb1(1,1) = Total1_greenb1+(greenb1(i,j)-Mean_greenb1)^2;
   end
end
Std_greenb1 = (Total1_greenb1(1,1)/6400)^0.5;

%Blue calculation
Total_blueb1=[0];
Mean_blueb1=[0];
for i=1:80
   for j=1:80
      Total_blueb1(1,1)= Total_blueb1(1,1)+ blueb1(i,j);
   end
end
Mean_blueb1 = Total_blueb1(1,1)/6400;
Total1_blueb1=[0];
for i =1:80
   for j = 1:80
Total1_blueb1(1,1) = Total1_blueb1+(blueb1(i,j)-Mean_blueb1)^2;
   end
end
Std_blueb1 = (Total1_blueb1(1,1)/6400)^0.5;
```

2. Source code of image analysis techniques for *in situ* detection of diseased cassava plants

1.1 Image cropping

```
N01=imread('N01.jpg');          % N01 = original image
%crop image
c1=imcrop(N01,[1 1 79 79]);          c2=imcrop(N01,[81 1 79 79]);
c3=imcrop(N01,[161 1 79 79]);        c4=imcrop(N01,[241 1 79 79]);
c5=imcrop(N01,[321 1 79 79]);        c6=imcrop(N01,[401 1 79 79]);
c7=imcrop(N01,[481 1 79 79]);        c8=imcrop(N01,[561 1 79 79]);
c9=imcrop(N01,[1 81 79 79]);         c10=imcrop(N01,[81 81 79 79]);
c11=imcrop(N01,[161 81 79 79]);      c12=imcrop(N01,[241 81 79 79]);
c13=imcrop(N01,[321 81 79 79]);      c14=imcrop(N01,[401 81 79 79]);
c15=imcrop(N01,[481 81 79 79]);      c16=imcrop(N01,[561 81 79 79]);
c17=imcrop(N01,[1 161 79 79]);       c18=imcrop(N01,[81 161 79 79]);
c19=imcrop(N01,[161 161 79 79]);     c20=imcrop(N01,[241 161 79 79]);
c21=imcrop(N01,[321 161 79 79]);     c22=imcrop(N01,[401 161 79 79]);
c23=imcrop(N01,[481 161 79 79]);     c24=imcrop(N01,[561 161 79 79]);
c25=imcrop(N01,[1 241 79 79]);       c26=imcrop(N01,[81 241 79 79]);
c27=imcrop(N01,[161 241 79 79]);     c28=imcrop(N01,[241 241 79 79]);
c29=imcrop(N01,[321 241 79 79]);     c30=imcrop(N01,[401 241 79 79]);
c31=imcrop(N01,[481 241 79 79]);     c32=imcrop(N01,[561 241 79 79]);
c33=imcrop(N01,[1 321 79 79]);       c34=imcrop(N01,[81 321 79 79]);
c35=imcrop(N01,[161 321 79 79]);     c36=imcrop(N01,[241 321 79 79]);
c37=imcrop(N01,[321 321 79 79]);     c38=imcrop(N01,[401 321 79 79]);
c39=imcrop(N01,[481 321 79 79]);     c40=imcrop(N01,[561 321 79 79]);
c41=imcrop(N01,[1 401 79 79]);       c42=imcrop(N01,[81 401 79 79]);
c43=imcrop(N01,[161 401 79 79]);     c44=imcrop(N01,[241 401 79 79]);
c45=imcrop(N01,[321 401 79 79]);     c46=imcrop(N01,[401 401 79 79]);
c47=imcrop(N01,[481 401 79 79]);     c48=imcrop(N01,[561 401 79 79]);
```

1.2 Transformation of RGB colors to HSI scale

```
%extract the individual component images
rgb=imdouble(c1);
r1=rgb(:,:,1);
g1=rgb(:,:,2);
b1=rgb(:,:,3);
% Implement the conversion equations
num1 = 0.5*((r1-g1)+(r1-b1));
den1 = sqrt((r1-g1).^2+(r1-b1).*(g1-b1));
theta1=acos(num1./(den1+eps));
H1=theta1;
H1(b1>g1)=2*pi-H1(b1>g1);
H1=H1/(2*pi);
num1=min(min(r1,g1),b1);
den1=r1+g1+b1;
den1(den1 == 0)= eps;
S1 = 1-3.*num1./den1;
H1(S1==0)=0;
I1=(r1+g1+b1)/3;
% Combine all three results into an hsi image
hsi1 = cat(3, H1, S1, I1);
xlswrite('red1.xls',r1,'A1');
xlswrite('green1.xls',g1, 'B1');
xlswrite('blue1.xls',b1, 'C1');
xlswrite('hue1.xls',H1,'D1');
xlswrite('saturation1.xls',S,'E1');
xlswrite('intensity1.xls',I1,'F1');
```

1.3 Computation for mean of each color index

```
Clc
clear
rc1=xlsread('red1.xls');
gc1=xlsread('green1.xls');
bc1=xlsread('blue1.xls');
%ru, gu, and bu were the chromatic coordinates.
        ru1=r1./(r1+g1+b1); gu1=g1./(r1+g1+b1); bu1=b1./(r1+g1+b1);
%color indices: (w=ru-gu, x=gu-bu, y=(gu-bu)/(ru-gu) and z=2gu-ru-bu)
w1=ru1-gu1; x1=gu1-bu1; z1=(2*gu1)-ru1-bu1;
for i = 1 : 80
   for j = 1 : 80
if abs(w1(i, j))>=0.01
   y1(i, j)=x1(i, j)/abs(w1(i, j));
   else y1(i, j)=x1(i, j)/0.01;
end
   end
end
%mean w1, x1, y1, z1
Total_w1=[0];
Mean_w1=[0];
for i=1:80
   for j=1:80
      Total_w1(1,1)= Total_w1(1,1)+ w1(i, j);
   end
end
Mean_w1 = Total_w1(1,1)/6400;
 Total_x1=[0];
Mean_x1=[0];
for i=1:80
   for j=1:80
```

```
        Total_x1(1,1)= Total_x1(1,1)+ x1(i, j);
    end
end
Mean_x1 = Total_x1(1,1)/6400;
 Total_y1=[0];
Mean_y1=[0];
for i=1:80
    for j=1:80
        Total_y1(1,1)= Total_y1(1,1)+ y1(i, j);
    end
end
Mean_y1 = Total_y1(1,1)/6400;
 Total_z1=[0];
Mean_z1=[0];
for i=1:80
    for j=1:80
        Total_z1(1,1)= Total_z1(1,1)+ z1(i, j);
    end
end
Mean_z1 = Total_z1(1,1)/6400;
 %mean r, g ,b
Total_r1=[0];
Mean_r1=[0];
for i=1:80
    for j=1:80
        Total_r1(1,1)= Total_r1(1,1)+ r1(i, j);
    end
end
Mean_r1 = Total_r1(1,1)/6400;
 Total_g1=[0];
Mean_g1=[0];
for i=1:80
```

```
    for j=1:80
        Total_g1(1,1)= Total_g1(1,1)+ g1(i, j);
    end
end
Mean_g1 = Total_g1(1,1)/6400;
 Total_b1=[0];
Mean_b1=[0];
for i=1:80
    for j=1:80
        Total_b1(1,1)= Total_b1(1,1)+ b1(i,j);
    end
end
Mean_b1 = Total_b1(1,1)/6400;
 %write to excel file.
%color indices: (w=ru-gu, x=gu-bu, y=(gu-bu)/(ru-gu) and z=2gu-ru-bu)
xlswrite('chromatic.xls',Mean_w1,'sheet1','A1');
xlswrite('chromatic.xls',Mean_x1,'sheet1','B1');
xlswrite('chromatic.xls',Mean_y1,'sheet1','C1');
xlswrite('chromatic.xls',Mean_z1,'sheet1','D1');
xlswrite('chromatic.xls',Mean_r1,'sheet1','E1');
xlswrite('chromatic.xls',Mean_g1,'sheet1','F1');
xlswrite('chromatic.xls',Mean_b1,'sheet1','G1');
```

3. Source code of image analysis algorithms for precision positioning of disease spots on cassava leaves

3.1 Algorithm I

```
clc
clear
files = dir('*.jpg');
num_files = numel(files);
images = cell(1, num_files);
for k = 1:num_files
    images{k} = imread(files(k).name);
end
 x = zeros(1,num_files);
y = zeros(1,num_files);
 for k = 1:num_files
% low pass filter by gaussian filter
F1 = fspecial('gaussian', [5 5], 10);
F2 = imfilter(images{k}, F1,'replicate');
 % RGB convert to HSI color model
rgb=imouble(F2);
r=rgb(:,:,1);
g=rgb(:,:,2);
b=rgb(:,:,3);
    % Implement the conversion equations
      num = 0.5*((r-g)+(r-b));
      den = sqrt((r-g).^2+(r-g).*(g-b));
      theta = acos(num./(den+eps));
      H = theta;
      H(b>g) = 2*pi-H(b>g);
      H=H/(2*pi);
      num=min(min(r,g),b);
```

```
    den = r+g+b;
    den(den == 0)=eps;
    S=1-3.*num./den;
    H(S == 0) = 0;
    I = (r+g+b)/3;
  % Combine all three results into an hsi image.
    hsi = cat(3, H, S, I);
  % Hue image segmentation
    Hue = 0;
    for i =  1:480
      for j = 1:640
        if H(i,j) > 0.22
          Hue(i,j) = 0;
        elseif H(i,j) <= 0.22
          Hue(i,j) = 1;
        end
      end
    end
% Fill holes in the Hue image
BW2 = imfill(Hue,'holes');
SE = strel('disk',5);
BW3 = imopen(BW2, SE);
% Object recognition
      % Detection of area value less than 1500 pixels (area1).
  cc1 = bwconncomp(BW3, 8);
  stats1 = regionprops (cc1, 'Area');
  idx1 = find([stats1.Area] < 800);
  area1 = ismember(labelmatrix(cc1), idx1);
% Detection of stems by ratio of MajorAxis and MinorAxis calculation(stems).
  cc2 = bwconncomp(area1, 8);
  stats2 = regionprops (cc2, 'MajorAxisLength', 'MinorAxisLength');
      idx2 = find([stats2.MajorAxisLength]./[stats2.MinorAxisLength]>3);
```

```
    stems = ismember(labelmatrix(cc2), idx2);
  % area of spots, which're less than 1500 pixels and without stems (area2).
    area2 = area1-stems;
   % Detection of area value less than 100 pixels (area3)
    cc3 = bwconncomp(area2, 8);
    stats3 = regionprops(cc3, 'Area');
    idx3 = find([stats3.Area] <100);
    area3 = ismember(labelmatrix(cc3), idx3);
    % area of spots in rage pixels value of 100-1500
    area4 = area2-area3;
    % hole filling
    area5 = imfill(area4);
    % spots enclosing
    perimeter = bwperim(area5);
overlay1 = imoverlay(images{k}, perimeter, [1 0 0]);
 % label spot regions
[labeled, numObjects] = bwlabeln(area5,8);
numob = numObjects;
 % Pixels calculation
sum_D = [0];
    for i = 1:480
       for j = 1:640
           sum_D(1,1) = sum_D(1,1) + area5(i,j);
       end
    end
% writting image & writting excel data
 fn1 = int2str(k);
 fn1 = strcat(fn1,'.jpg');
 imwrite(area5,fn1,'jpg');
 x(k) = numob;
y(k) = sum_D;
end
```

```
 xlswrite('number of spots.xls', x,'A1');
xlswrite('number of spots.xls', y,'A4');
```

3.2 Algorithm II

```
clc
clear
files = dir('*.jpg');
num_files = numel(files);
images = cell(1, num_files);
for k = 1:num_files
    images{k} = imread(files(k).name);
end
x = zeros(1,num_files);
y = zeros(1,num_files);
 for k = 1:num_files
 % rgb reading
rgb=imouble(images{k});
r=rgb(:,:,1);
g=rgb(:,:,2);
b=rgb(:,:,3);
% color operation
ExR = (1.4*r)-g;
ExG = (2*g)-r-b;
NDI = (g+r)./(g-r);
Ex = ExG-ExR;
%image filtering
F1 = medfilt2(Ex, [5 5]);
F2 = fspecial('unsharp');
F3 = imfilter(F1, F2, 'replicate');
%image segmentation
Hue = 0;
```

```
    for i =  1:480
       for j = 1:640
          if F3(i,j) > 0
            Hue(i,j) = 0;
          elseif F3(i,j) <= 0
            Hue(i,j) = 1;
          end
       end
    end
 % Fill holes in the Hue image
BW2 = imfill(Hue,'holes');
 % Object recognition
% Detection of area value less than 1500 pixels (area1).
   cc1 = bwconncomp(BW2, 8);
   stats1 = regionprops (cc1, 'Area');
   idx1 = find([stats1.Area] < 800);
   area1 = ismember(labelmatrix(cc1), idx1);
% Detection of stems by ratio of MajorAxis and MinorAxis calculation(stems).
   cc2 = bwconncomp(area1, 8);
   stats2 = regionprops (cc2, 'MajorAxisLength', 'MinorAxisLength');
    idx2 = find([stats2.MajorAxisLength]./[stats2.MinorAxisLength]>3);
   stems = ismember(labelmatrix(cc2), idx2);
     % area of spots, which're less than 1500 pixels and without stems (area2).
   area2 = area1-stems;
   % Detection of area value less than 100 pixels (area3)
   cc3 = bwconncomp(area2, 8);
   stats3 = regionprops(cc3, 'Area');
   idx3 = find([stats3.Area] <100);
   area3 = ismember(labelmatrix(cc3), idx3);
  % area of spots in rage pixels value of 100-1500
   area4 = area2-area3;
   % hole filling
```

```
    area5 = imfill(area4);
% spots enclosing
perimeter = bwperim(area5);
overlay1 = imoverlay(images{k}, perimeter, [1 0 0]);
% label spot regions
[labeled, numObjects] = bwlabeln(area5,8);
numob = numObjects;
% Pixels calculation
sum_D = [0];
   for i = 1:480
      for j = 1:640
         sum_D(1,1) = sum_D(1,1) + area5(i,j);
      end
   end
% writting image & writting excel data
 fn1 = int2str(k);
 fn1 = strcat(fn1,'.jpg');
 imwrite(area5,fn1,'jpg');
x(k) = numob;
y(k) = sum_D;
end
xlswrite('number of spots.xls', x,'A1');
xlswrite('number of spots.xls', y,'A4');
```

3.3 Algorithm III

```
clc
clear
files = dir('*.jpg');
num_files = numel(files);
images = cell(1, num_files);
for k = 1:num_files
```

```
    images{k} = imread(files(k).name);
end
x = zeros(1,num_files);
y = zeros(1,num_files);
for k = 1:num_files
%read image
% RGB convert to HSI color model
rgb=imouble(images{k});
r=rgb(:,:,1);
g=rgb(:,:,2);
b=rgb(:,:,3);
% color operation
ExR = (1.4*r)-g;
ExG = (2*g)-r-b;
NDI = (g+r)./(g-r);
Ex = ExG-ExR;
   % Implement the conversion equations
      num = 0.5*((r-g)+(r-b));
      den = sqrt((r-g).^2+(r-g).*(g-b));
      theta = acos(num./(den+eps));
      H = theta;
      H(b>g) = 2*pi-H(b>g);
      H=H/(2*pi);
      num=min(min(r,g),b);
      den = r+g+b;
      den(den == 0)=eps;
      S=1-3.*num./den;
      H(S == 0) = 0;
      I = (r+g+b)/3;
   % Combine all three results into an hsi image.
      hsi = cat(3, H, S, I);
% spots extraction
```

```
% spots segmentation
   Hue1 = 0;
   for i =  1:480
      for j = 1:640
         if H(i,j) > 0.2
            Hue1(i,j) = 0;
         elseif H(i,j) <= 0.2
            Hue1(i,j) = 1;
         end
      end
   end
   % Fill holes in the Hue image
   BW2 = imfill(Hue1,'holes');
   %SE = strel('disk',5);
   %BW2 = imopen(BW1, SE);
%leaves extraction
   % low pass filter by gaussian filter
   F1 = fspecial('gaussian', [3 3], 10);
   F4 = imfilter(ExG, F1,'replicate');
   %F3 = fspecial('unsharp');
   %F4 = imfilter(F2, F3, 'replicate');
   %leaves segmentation
   level = graythresh(F4);
   Hue2 = 0;
   for i =  1:480
      for j = 1:640
         if F4(i,j) > level
            Hue2(i,j) = 1;
         elseif F4(i,j) <= level
            Hue2(i,j) = 0;
         end
      end
```

```
            end
          % Fill holes in the Hue image
          BW3 = imfill(Hue2,'holes');
        c= 0;
       for i=1:480
          for j=1:640
            if BW3(i,j)>=BW2(i,j)&BW3(i,j)<=BW2(i,j); % if you want to intersect the exact amplitudes, you don't have to use +/-0.05.
              c(i,j)=BW2(i,j);
            else
              c(i,j)=0;
            end
          end
       end
       SE = strel('disk',4);
       BW4 = imopen(c, SE);
        % Detection of stems by ratio of MajorAxis and MinorAxis calculation(stems).
          cc = bwconncomp(BW4, 8);
          stats = regionprops (cc, 'MajorAxisLength', 'MinorAxisLength');
              idx = find([stats.MajorAxisLength]./[stats.MinorAxisLength]>2);
          stems = ismember(labelmatrix(cc), idx);
           % area of spots, which're less than 1500 pixels and without stems (area2).
          area = BW4-stems;
       % spots enclosing
       perimeter = bwperim(area);
       overlay1 = imoverlay(images{k}, perimeter, [1 0 0]);
       % label spot regions
       [labeled, numObjects] = bwlabeln(area,8);
       numob = numObjects;
       % Pixels calculation
       sum_D = [0];
```

```
    for i = 1:480
        for j = 1:640
            sum_D(1,1) = sum_D(1,1) + area(i,j);
        end
    end
% writting image & writting excel data
 fn1 = int2str(k);
 fn1 = strcat(fn1,'.jpg');
 imwrite(perimeter,fn1,'jpg');
x(k) = numob;
y(k) = sum_D;
end
xlswrite('number of spots.xls', x,'A1');
xlswrite('number of spots.xls', y,'A4');
```

3.4 Image overlay function for marking the positions of detected spots

```
function out = imoverlay(in, mask, color)

DEFAULT_COLOR = [1 1 1];
if nargin < 3
    color = DEFAULT_COLOR;
end

% Make the uint8 the working data class.  The output is also uint8.
in_uint8 = im2uint8(in);
color_uint8 = im2uint8(color);

% Initialize the red, green, and blue output channels.
if ndims(in_uint8) == 2
% Input is grayscale.  Initialize all output channels the same.
    out_red   = in_uint8;
    out_green = in_uint8;
    out_blue  = in_uint8;
else
    % Input is RGB truecolor.
    out_red   = in_uint8(:,:,1);
    out_green = in_uint8(:,:,2);
    out_blue  = in_uint8(:,:,3);
end
```

```
% Replace output channel values in the mask locations with the appropriate
% color value.
out_red(mask)   = color_uint8(1);
out_green(mask) = color_uint8(2);
out_blue(mask)  = color_uint8(3);

% Form an RGB truecolor image by concatenating the channel matrices along
% the third dimension.
out = cat(3, out_red, out_green, out_blue);
```

4. Source code of image processing algorithm for the assessment of diseases infection severity

```
function varargout = rating3(varargin)
    gui_Singleton = 1;
    gui_State = struct('gui_Name',       mfilename, ...
                       'gui_Singleton',  gui_Singleton, ...
                       'gui_OpeningFcn', @rating3_OpeningFcn, ...
                       'gui_OutputFcn',  @rating3_OutputFcn, ...
                       'gui_LayoutFcn',  [] , ...
                       'gui_Callback',   []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
    % End initialization code - DO NOT EDIT


% --- Executes just before rating3 is made visible.
    function rating3_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to rating3 (see VARARGIN)
% Choose default command line output for rating3
handles.output = hObject;
 % Update handles structure
guidata(hObject, handles);
 % UIWAIT makes rating3 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = rating3_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
f = handles.A;
axes(handles.axes1)
rgb=imouble(f);
r=rgb(:,:,1);
g=rgb(:,:,2);
```

```
b=rgb(:,:,3);

  % Implement the conversion equations
    num = 0.5*((r-g)+(r-b));
    den = sqrt((r-g).^2+(r-g).*(g-b));
    theta = acos(num./(den+eps));
    H = theta;
    H(b>g) = 2*pi-H(b>g);
    H=H/(2*pi);
    num=min(min(r,g),b);
    den = r+g+b;
    den(den == 0)=eps;
    S=1-3.*num./den;
    H(S == 0) = 0;
    I = (r+g+b)/3;
  % Combine all three results into an hsi image.
    hsi = cat(3, H, S, I);
  % Segmented Image
    Ibw = 0;
    for i =  1:480
      for j = 1:640
        if H(i,j) > 0.22
          Ibw(i,j) = 0;
        elseif H(i,j) <= 0.22
          Ibw(i,j) = 1;
        end
      end
    end
  % Noise reduction (Ibw3 = the image was performed erosion and diation)
    SE = strel('rectangle', [5 5]);
    Ibw2 = imerode(Ibw,SE); %Ibw2 = erosion
    Ibw3 = imdilate(Ibw2,SE); %Ibw3 = diation after erosion
```

```
    % Spot area
       sum_d1 = [0];
       for i = 1:480
          for j = 1:640
             sum_d1(1,1) = sum_d1(1,1) + Ibw3(i,j);
          end
       end
    % connected components
    [labeled, numObjects] = bwlabeln(Ibw3,8);
numob=numObjects; % count all distinct objects in the images.

% Total leaf area
G = graythresh(I);
Ibw4 = 0;
for i =  1:480
   for j = 1:640
      if I(i,j) > G
      Ibw4(i,j) = 0;
      elseif I(i,j) <= G
      Ibw4(i,j) = 1;
      end
   end
end
sum_L1 = [0];
for i = 1:480
   for j = 1:640
      sum_L1(1,1) = sum_L1(1,1) + Ibw4(i,j);
   end
end
% percentage of disease on leaf
percentage = (sum_d1/sum_L1)*100;
if (percentage >0 & percentage <5)
```

```
    gd =1;
  elseif (percentage >5 & percentage <10)
    gd =2;
  elseif (percentage >10 & percentage <15)
    gd =3;
  elseif (percentage >15 & percentage <20)
    gd =4;
  elseif (percentage >20 & percentage <25)
    gd =5;
  else gd = 0;
   end

  handles.image = f;
  handles.hsiimage = hsi;
  handles.spotimage = Ibw3;
  handles.leafimage = Ibw4;
  handles.red = r;
  handles.green = g;
  handles.blue = b;
  handles.H = H;
  handles.S = S;
  handles.I = I;
  set(handles.sp,'string',num2str(numob))
  set(handles.pd,'string',num2str(percentage))
  set(handles.rd,'string',num2str(gd))
  guidata(hObject,handles);

imshow(handles.image)

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton1
if get(handles.radiobutton1,'value')
      imshow(handles.image);
    elseif get(handles.radiobutton2,'value')
      imshow(handles.hsiimage);
    elseif get(handles.radiobutton9,'value')
      imshow(handles.spotimage);
    elseif get(handles.radiobutton10,'value')
      imshow(handles.leafimage);
    elseif get(handles.radiobutton3,'value')
      imshow(handles.red);
    elseif get(handles.radiobutton4,'value')
      imshow(handles.green);
    elseif get(handles.radiobutton5,'value')
      imshow(handles.blue);
    elseif get(handles.radiobutton6,'value')
      imshow(handles.H);
    elseif get(handles.radiobutton7,'value')
      imshow(handles.S);
    elseif get(handles.radiobutton8,'value')
      imshow(handles.I);
    else
      imshow(handles.image);
    end


% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton2
if get(handles.radiobutton1,'value')
      imshow(handles.image);
    elseif get(handles.radiobutton2,'value')
      imshow(handles.hsiimage);
    elseif get(handles.radiobutton9,'value')
      imshow(handles.spotimage);
    elseif get(handles.radiobutton10,'value')
      imshow(handles.leafimage);
    elseif get(handles.radiobutton3,'value')
      imshow(handles.red);
    elseif get(handles.radiobutton4,'value')
      imshow(handles.green);
    elseif get(handles.radiobutton5,'value')
      imshow(handles.blue);
    elseif get(handles.radiobutton6,'value')
      imshow(handles.H);
    elseif get(handles.radiobutton7,'value')
      imshow(handles.S);
    elseif get(handles.radiobutton8,'value')
      imshow(handles.I);
    else
      imshow(handles.image);
    end

% --- Executes on button press in radiobutton9.
function radiobutton9_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton9
```

```
if get(handles.radiobutton1,'value')
    imshow(handles.image);
  elseif get(handles.radiobutton2,'value')
    imshow(handles.hsiimage);
  elseif get(handles.radiobutton9,'value')
    imshow(handles.spotimage);
  elseif get(handles.radiobutton10,'value')
    imshow(handles.leafimage);
  elseif get(handles.radiobutton3,'value')
    imshow(handles.red);
  elseif get(handles.radiobutton4,'value')
    imshow(handles.green);
  elseif get(handles.radiobutton5,'value')
    imshow(handles.blue);
  elseif get(handles.radiobutton6,'value')
    imshow(handles.H);
  elseif get(handles.radiobutton7,'value')
    imshow(handles.S);
  elseif get(handles.radiobutton8,'value')
    imshow(handles.I);
  else
    imshow(handles.image);
  end

% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton3
```

```
if get(handles.radiobutton1,'value')
    imshow(handles.image);
  elseif get(handles.radiobutton2,'value')
    imshow(handles.hsiimage);
  elseif get(handles.radiobutton9,'value')
    imshow(handles.spotimage);
  elseif get(handles.radiobutton10,'value')
    imshow(handles.leafimage);
  elseif get(handles.radiobutton3,'value')
    imshow(handles.red);
  elseif get(handles.radiobutton4,'value')
    imshow(handles.green);
  elseif get(handles.radiobutton5,'value')
    imshow(handles.blue);
  elseif get(handles.radiobutton6,'value')
    imshow(handles.H);
  elseif get(handles.radiobutton7,'value')
    imshow(handles.S);
  elseif get(handles.radiobutton8,'value')
    imshow(handles.I);
  else
    imshow(handles.image);
  end

% --- Executes on button press in radiobutton4.
function radiobutton4_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton4
if get(handles.radiobutton1,'value')
    imshow(handles.image);
```

```
    elseif get(handles.radiobutton2,'value')
      imshow(handles.hsiimage);
    elseif get(handles.radiobutton9,'value')
      imshow(handles.spotimage);
    elseif get(handles.radiobutton10,'value')
      imshow(handles.leafimage);
    elseif get(handles.radiobutton3,'value')
      imshow(handles.red);
    elseif get(handles.radiobutton4,'value')
      imshow(handles.green);
    elseif get(handles.radiobutton5,'value')
      imshow(handles.blue);
    elseif get(handles.radiobutton6,'value')
      imshow(handles.H);
    elseif get(handles.radiobutton7,'value')
      imshow(handles.S);
    elseif get(handles.radiobutton8,'value')
      imshow(handles.I);
    else
      imshow(handles.image);
    end


% --- Executes on button press in radiobutton5.
function radiobutton5_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton5
if get(handles.radiobutton1,'value')
      imshow(handles.image);
    elseif get(handles.radiobutton2,'value')
```

```
    imshow(handles.hsiimage);
  elseif get(handles.radiobutton9,'value')
    imshow(handles.spotimage);
  elseif get(handles.radiobutton10,'value')
    imshow(handles.leafimage);
  elseif get(handles.radiobutton3,'value')
    imshow(handles.red);
  elseif get(handles.radiobutton4,'value')
    imshow(handles.green);
  elseif get(handles.radiobutton5,'value')
    imshow(handles.blue);
  elseif get(handles.radiobutton6,'value')
    imshow(handles.H);
  elseif get(handles.radiobutton7,'value')
    imshow(handles.S);
  elseif get(handles.radiobutton8,'value')
    imshow(handles.I);
  else
    imshow(handles.image);
  end

% --- Executes on button press in radiobutton6.
function radiobutton6_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton6
 if get(handles.radiobutton1,'value')
    imshow(handles.image);
  elseif get(handles.radiobutton2,'value')
    imshow(handles.hsiimage);
  elseif get(handles.radiobutton9,'value')
```

```
    imshow(handles.spotimage);
  elseif get(handles.radiobutton10,'value')
    imshow(handles.leafimage);
  elseif get(handles.radiobutton3,'value')
    imshow(handles.red);
  elseif get(handles.radiobutton4,'value')
    imshow(handles.green);
  elseif get(handles.radiobutton5,'value')
    imshow(handles.blue);
  elseif get(handles.radiobutton6,'value')
    imshow(handles.H);
  elseif get(handles.radiobutton7,'value')
    imshow(handles.S);
  elseif get(handles.radiobutton8,'value')
    imshow(handles.I);
  else
    imshow(handles.image);
  end

% --- Executes on button press in radiobutton7.
function radiobutton7_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton7
 if get(handles.radiobutton1,'value')
    imshow(handles.image);
  elseif get(handles.radiobutton2,'value')
    imshow(handles.hsiimage);
  elseif get(handles.radiobutton9,'value')
    imshow(handles.spotimage);
  elseif get(handles.radiobutton10,'value')
```

```
    imshow(handles.leafimage);
  elseif get(handles.radiobutton3,'value')
    imshow(handles.red);
  elseif get(handles.radiobutton4,'value')
    imshow(handles.green);
  elseif get(handles.radiobutton5,'value')
    imshow(handles.blue);
  elseif get(handles.radiobutton6,'value')
    imshow(handles.H);
  elseif get(handles.radiobutton7,'value')
    imshow(handles.S);
  elseif get(handles.radiobutton8,'value')
    imshow(handles.I);
  else
    imshow(handles.image);
  end

% --- Executes on button press in radiobutton8.
function radiobutton8_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton8
if get(handles.radiobutton1,'value')
    imshow(handles.image);
  elseif get(handles.radiobutton2,'value')
    imshow(handles.hsiimage);
  elseif get(handles.radiobutton9,'value')
    imshow(handles.spotimage);
  elseif get(handles.radiobutton10,'value')
    imshow(handles.leafimage);
```

```
    elseif get(handles.radiobutton3,'value')
      imshow(handles.red);
    elseif get(handles.radiobutton4,'value')
      imshow(handles.green);
    elseif get(handles.radiobutton5,'value')
      imshow(handles.blue);
    elseif get(handles.radiobutton6,'value')
      imshow(handles.H);
    elseif get(handles.radiobutton7,'value')
      imshow(handles.S);
    elseif get(handles.radiobutton8,'value')
      imshow(handles.I);
    else
      imshow(handles.image);
    end

% --- Executes on button press in radiobutton10.
function radiobutton10_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton10
 if get(handles.radiobutton1,'value')
      imshow(handles.image);
    elseif get(handles.radiobutton2,'value')
      imshow(handles.hsiimage);
    elseif get(handles.radiobutton9,'value')
      imshow(handles.spotimage);
    elseif get(handles.radiobutton10,'value')
      imshow(handles.leafimage);
    elseif get(handles.radiobutton3,'value')
      imshow(handles.red);
```

```
    elseif get(handles.radiobutton4,'value')
      imshow(handles.green);
    elseif get(handles.radiobutton5,'value')
      imshow(handles.blue);
    elseif get(handles.radiobutton6,'value')
      imshow(handles.H);
    elseif get(handles.radiobutton7,'value')
      imshow(handles.S);
    elseif get(handles.radiobutton8,'value')
      imshow(handles.I);
    else
      imshow(handles.image);
    end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.vid = videoinput('winvideo',1,'YUY2_640x480');
guidata(hObject,handles);

set(handles.vid,'TriggerRepeat',Inf);
handles.vid.FrameGrabInterval = 5;
vid_src = getselectedsource(handles.vid);
set(vid_src,'Tag','motion detection setup');
start(handles.vid)
preview (handles.vid)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
im = getdata(handles.vid,1);

Y = single(im(:,:,1)); U = single(im(:,:,2)); V = single(im(:,:,3));

C = Y-16; D = U -128; E = V-128;

R = uint8((298*C+409*E+128)/256);
G = uint8((298*C-100*D-208*E+128)/256);
B = uint8((298*C+516*D+128)/256);

handles.newim = uint8(zeros(size(im)));
handles.newim(:,:,1)=R;
handles.newim(:,:,2)=G;
handles.newim(:,:,3)=B;

imshow(handles.newim)
imwrite(handles.newim,'d:\spd\IMG.jpg');
guidata(hObject, handles);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
stop(handles.vid),clear handles.vid;
clear

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over pd.
function pd_ButtonDownFcn(hObject, eventdata, handles)
```

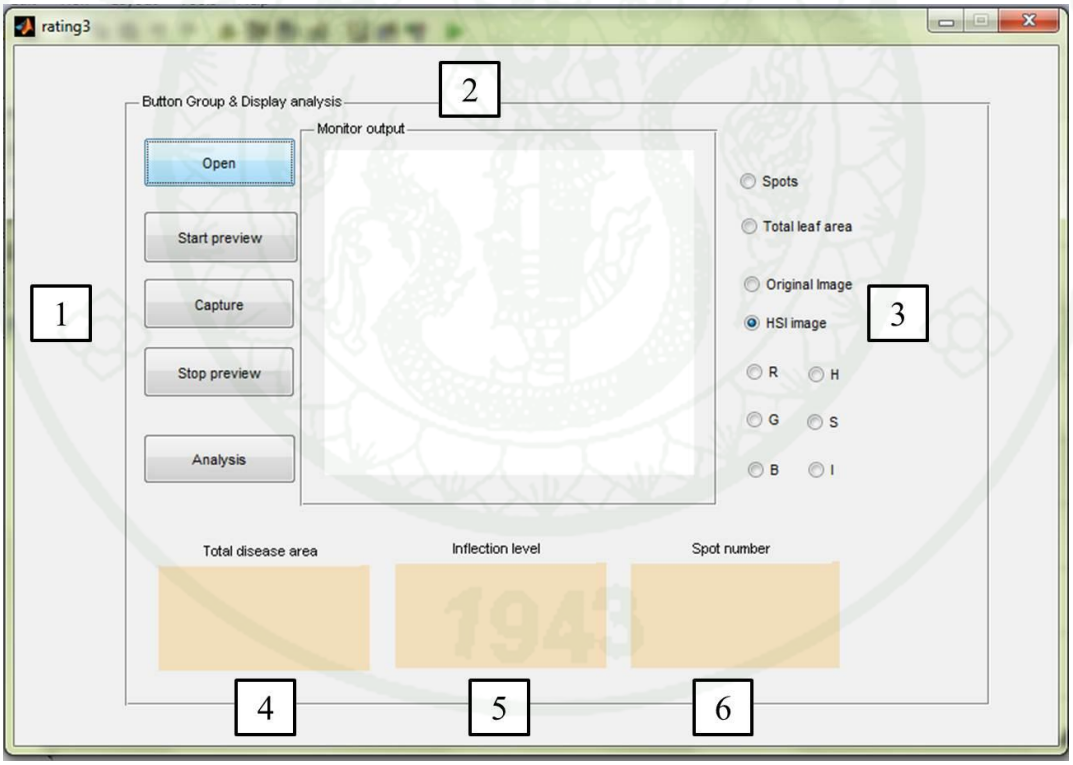
```
% hObject    handle to pd (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

 % --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile({'*.bmp';'*.jpg';'*.tif'});
A = imread([pathname filename]);
handles.A = A;
axes(handles.axes1);
imshow(A);
handles.output = hObject;
guidata(hObject, handles);
```

Appendix C

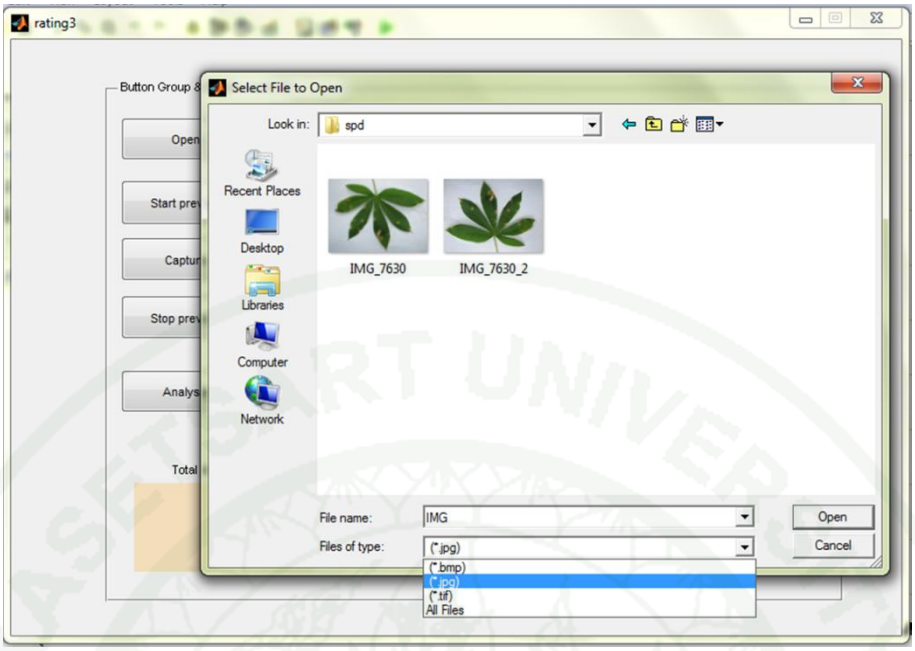
Graphical user interface (GUI) for the assessment of Brown Leaf Spot disease level

The GUI developed for the assessment of BLS disease level: A users' guide

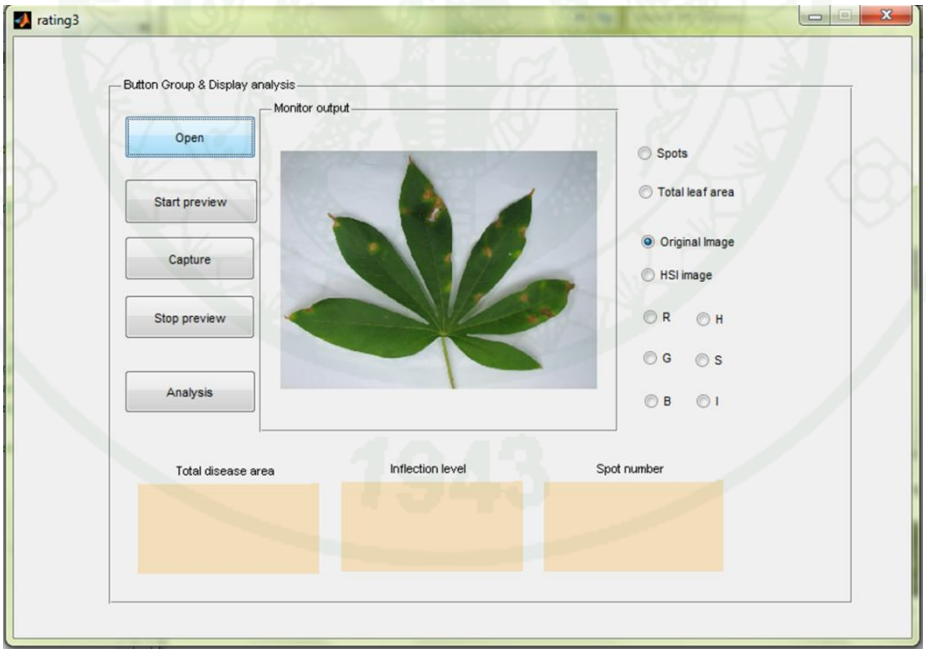
1. Click Open button (Appendix Figure C1) then the window would appear and allowing to select an image to analyze. The program allows to analyzing images of following types: BMP, JPG and TIF (Appendix Figure C2).
2. After selecting an image, the selected image will appears on frame (Figure C3).
3. Click Analysis button to run the program.
4. All parameters including total disease area, infection area, and number of spots will be calculated and reported. (Figure C4).
5. The GUI can display RGB and HSI image by choosing the corresponding radio button on the right side of the GUI.



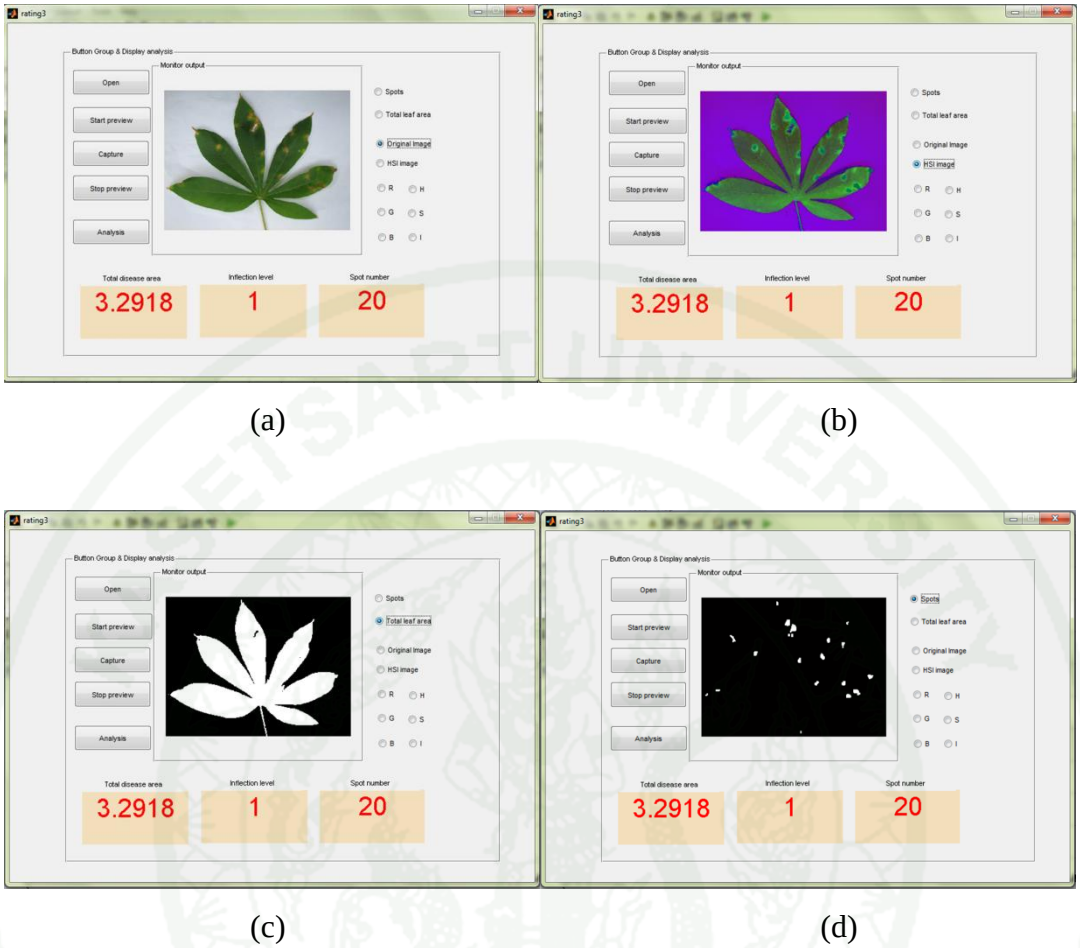
Appendix Figure C1 Graphical user interface of the developed program for the assessment of disease serverity. (1. Control buttons, 2. Output display, 3. Radio buttons for color mode selection, 4. Display for total disease area, 5. Display for inflection level, and 6. Display for number of spots)



Appendix Figure C2 Image selection



Appendix Figure C3 The selected image displayed in the GUI



Appendix Figure C4 The display of analysis outputs: (a) Original image in RGB mode, (b) HSI image, (c) Binary image of leaf, and (d) Binary image of spots.

CURRICULUM VITAE

NAME : Mr. Kittipong Powbunthorn

BIRTH DATE : July 31, 1987

BIRTH PLACE : Bangkok, Thailand

EDUCATION :

YEAR	INSTITUTE	DEGREE/DIPLOMA
2009	Kasetsart Univ.	B.Eng.(Agricultural Engineering)

POSITION/TITLE : -

WORK PLACE : -

SCHOLARSHIP/AWARDS : Kasetsart University Scholarship, under the Capacity Building for KU students on Internationalization Program 2010.

JASSO scholarship of Japan 2010.

The Graduate School, KU Scholarship 2011.

PUBLICATIONS :

Powbunthorn, K., W. Abdullakasim and J. Unartngam. 2010. Cassava Disease Detection by Fractal Analysis of Digital Images. **J. ISSAAS** 16 (1): 127-128.

Powbunthorn, K., W. Abdullakasim, J. Unartngam and T. Takigawa. 2010. Detection of Brown Leaf Spot Disease in Cassava by Digital Images Analysis, pp. 32-33. *In* **Proceeding of the 3rd Thailand-Japan International Academic Conference**. 19 November 2010, Nagoya, Japan.

Abdullakasim, W., K. Powbunthorn, J. Unartngam and T. Takigawa. 2011. An Images Analysis Technique for Recognition of Brown Leaf Spot Disease in Cassava. **Journal of Agricultural Machinery Science** 7(2): 165-169.

Powbunthorn, K., W. Abdullakasim and J. Unartngam. 2011. Rating of Brown Leaf Spot Disease in Cassava by Digital Image Analysis, pp. 152-160. *In* **Proceeding of The 8 th KU-KPS Conference.** 8-9 December 2011, Nakhon Pathom, Thailand. in Thai.

Powbunthorn, K., W. Abdullakasim and J. Unartngam. 2012. Assessment of the Severity of Brown Leaf Spot Disease in Cassava using Image Analysis, pp. 826-833. *In* **Proceeding of the international conference of the Thai Society of Agricultural Engineering**. 4-5 April 2012, Chiangmai, Thailand.