A Mobile Application for Plant Nutrient Deficiency Detection

Poomkawin Sihanatkathakul
Purit Thong-on

Bachelor of Engineering in Software Engineering,
International College,
King Mongkut's Institute of Technology Ladkrabang,
Academic Year 2015

Thesis - Academic Year 2015

Bachelor of Engineering in Software Engineering

Internation College

King Mongkut's Institute of Technology Ladkrabang

Title: A Mobile Application for Plant Nutrient Deficiency Detection

Authors:

1. Mr. Poomkawin Sihanatkathakul Student ID: 55090038
2. Mr. Purit Thong-on Student ID: 55090039

Approved for submission

....

(Dr. Ukrit Watchareeruetai)

Advisor

Date 30/3/2016

Abstract

This paper proposed an automatic system for analysing nutrient deficiency in black gram plants using its young leaf and old leaf as indicators. The classification classes are complete (no deficiency occurred), Nitrogen deficiency, Calcium deficiency, Potassium deficiency, Magnesium deficiency, and Iron deficiency. The reason why the system needs both a young leaf and an old leaf is because some of the nutrients are catagorized as mobile nutrients (nutrient that can traverses within the tree) which means the symptoms may occur in different time and places. Especially between young leaf and old leaf.

In this system, there are two components. First, a mobile application is used for interacting with a user and let the user takes plant leaves' photos and also performing segmentation. The resulting images are sent to the server which is the second component. The server is used for analyzing the deficiency of the leaves from both photos and sending the result back to be displayed on the mobile application.

In this project, GrabCut algorithm is used for leaf segmentation while evolutionary algorithm is for constructing young leaf and old leaf feature extraction programs to be used in nutrient deficiency analysis. For evolutionary algorithm, the dataset of black gram leaves are colltected for training and testing. The experiments of both algorithm indicate positive performance of GrabCut segmentation and the potential in using evolutionary algorithm to construct the feature extraction programs.

Acknowledgements

This project, A Mobile Application for Plant Nutrient Deficiency Detection, consumed a huge amount of work, research, and dedication. Still, implementation would not have been possible if we do not have supports of many individuals in so many ways. Therefore, we would like to extend our sincere gratitude to all of them.

First of all, we are grateful to our advisor, Dr. Ukrit Watchareeruetai, for provision of expertise and technical support in the implementation. Without his superior knowledge and experience, this project will not become reality.

Besides our advisor, our sincere thanks also go to Dr. Sutsawat Duangsrisai at the Kasetsart University for providing us with her knowledge in botany. She also prepares sample plants for us to use with our project.

We are thankful to our lecturers, Dr. Natthapong Jungteerapanich and Dr. Chaiwat Nuthong, for his logistical support and for providing necessary equipments.

In addition, we also would like to thank other lecturers and staffs at the International College for guiding us and for supporting us in so many ways.

Nevertheless, we express our gratitude toward our families and colleagues for their kind co-operation and encouragement which help us in completion of this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter clarifies problems in identifying a nutrient deficient plant using human visual (Section 1.1 and Section 1.2). A semi-automatic method will be described to confront with these problems (Section 1.3). Then, brief description of the system will be expressed (Section 1.4). Next, the scope of this project will be defined (Section 1.5). Ultimately, organization of this thesis will be shown (Section 1.6).

1.1 Problem Description

When looking in the perspective of a manufacturer who raises plants in large fields and sells them as a business, increased in efficiency, productivity, and quality are important factors in production system which concerns producing massive amount of product. Especially, raising plants requires delicate works and methodologies.

There is, in fact, a science and engineering approach for producing crops in controlled environment which helps manufacturers to obtain clean, dependable, and healthy plants for consumption. This integrated science and engineering approach is called controlled-environment agriculture.

However, during the process, in order to produce quality plants, it is also important to be able to identify the lack of nutrients whether to get rid of the one that cannot be cured and treated as casualties, or to prevent any further damage by trying to cure the ones that can still be reverted to normal. The symptoms that are resulted from nutrient deficiency can also be detected by observing the characteristics in plants leaves.

This process typically uses human visual or requires physical contacts to determine the deficiency. It is a labor-intensive and destructive task if handled without care. However, there is always a risk in using human engineering within the process. The earliest research describing an attempt for solving this problem can be found published in 1995. Therefore, it would be more profitable to use an automatic system to replace humans in this process. In fact, Finding the solution to this problem has just recently been active for the past few years.

1.2 Motivation

Agriculture is one of the primary businesses which bring in revenues for Thailand since a long time ago. The country has successfully exported a large amount of agricultural products, for example, rubber, cassava, meat, and processed food products to other countries around the world.

In 2014, 41% of the area in Thailand was owned for agriculture purposes, which is 51.4 million hectares accommodating with one-third of population. 60% of agriculture production business also contributed to crops raising.

Plants are important agricultural products. Amount and quality of the plants obtained involve with many factors. Some examples of which are plant species, weather, water quantity, sunlight, soil quality, and crops management and protection. Nutrients are one of the most important factors which affect the growth of a plant. If there is a nutrient deficiency occurred, the plants might be damaged, or in the worst case, results in death. On the other hand, giving an excessive amount of nutrients to plants can also have negative results, and it is a waste of money too. Therefore, the act of carefully maintaining the plants so that it receives an appropriate amount of nutrient is crucial.

As a result, it is one of the most challenging problem to apply modern tools and technologies into developing a more conveient ways or atleast take a step forward to improve the quality of agricultural products and also gain benefits from the knowledge acquired in this project.

1.3 Objective

The objective of this project is to develop an automatic system on a mobile application to determine whether a plant has any nutrient deficiency or healthy (does not lack any nutrient).

1.4 System Overview

In our system, there are two components. The first component is a mobile application, and the second component is a processing server, as shown in Fig. 1.1. The mobile application mostly acts as the system interface. A user interacts only with this part of the system. It also does some process which will be described later in this thesis. The processing server does computational expensive tasks, such as an image procesing algorithm which requires a lot of raw CPU and GPU processing power to finish executed in a short period of time, which cannot be carried out by a smartphone that runs the mobile application. The mobile application and the server communicate between each other using the Internet. Only some functions in the mobile application will work if there is no Internet connection with the processing server or the processing server is down.

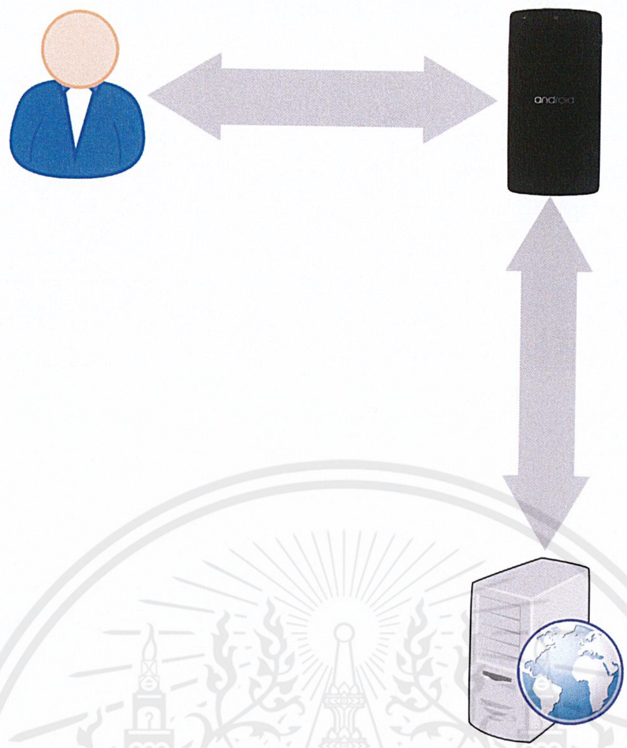


Figure 1.1: System overview

1.5 Scope of Work

The scopes of this project are as follows:

- The mobile application will be available only for Android operating system.
- The mobile application will be able to run on Android version 4.1 or newer.
- The mobile application will be able to run on smartphones not tablets.
- The plant used is Black Gram (*Vigna Mungo*).
- The system can only detect five nutrients deficiency, which are Nitrogen (N), Potassium (K), Magnesium (Mg), Calcium (Ca), and Iron (Fe) deficiency.

- The system can only detect a nutrient deficient plant when it lacks of only one nutrient.

1.6 Thesis Structure

The rest of the thesis is organized as follow:

- Chapter 2 explains the knowledge needed to be understood before reading further into this thesis.
- Chapter 3 describes the user interface of the mobile application and also justifies the algorithm for detecting nutrient deficiency in a plant.
- Chapter 4 illustrates the architecture of the system, including user interface designs and how the user interacts with the application.
- Chapter 5 explains and discusses experiment results for both image segmentation and nutrient deficiency detection mechanism.
- Chapter 6 sums up the propsed system.

Chapter 2

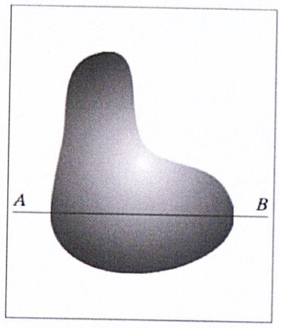
Background Knowledge and Literature Reviews

This chapter explains basic knowledge that helps understanding the rest of this thesis (Section 2.1). This chapter also give information that obtained from reviewing similar researches (Section 2.2).

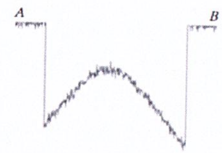
2.1 Background Knowledge

2.1.1 Digital Image

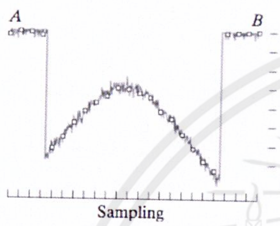
An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are *spatial*, or *plane*, coordinates, and the amplitude, or the measurement of changes, of f at any pair of coordinates (x, y) is called the *intensity*, or *gray level*, at that point in the image [1]. When x, y, and the intensity values of f are all definite, the image is called a *digital image*. A digital image is made up of a finite number of elements, each of which has particular location and value. These elements are called *picture elements* or *pixels*. In order to create a digital image from an image, the continuous spatial co-



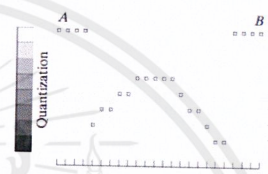
(a) Continuous image



(b) A scan line from A to B in the continuous image, used to illustrate the concepts of image sampling and image quantization.



(c) Sampling and quantization



(d) Digital scan line

Figure 2.1: Illustration of image sampling and image quantization. (image from [1])

ordinates and amplitude values need to be converted into a digital form. The method of digitalizing a continuous image associates two processes, which are *image sampling* and *image quantization*. Figure 2.1 gives an illustration of image sampling and image quantization. Figure 2.1a shows a continuous image with the continuous function f that will be converted into a digital form. An image is continuous with respect to the $x-$ and $y-$coordinates and is also continuous in amplitude. To convert it into a digital form, the function in both coordinates and amplitude have to be sampled. The process of digitizing the coordinate values is called *sampling*. The process of digitizing the amplitude values is called *quantization*.

A plot of intensity values of the continuous image along the line segment AB in Fig. 2.1a is shown in Fig. 2.1b as a one-dimensional function. The random variations are due to the noises in the image. To sample this function, equally spaced samples are taken along the line AB, as shown in Fig. 2.1c. The spatial location of each sample is indicated by a vertical tick marked in the bottom part of the figure. The samples are shown as small white squares laid on the function. However, the values of the samples still span a continuous range of intensity values. In order to form a digital function, the intensity values also must be converted into discrete quantities. In the left part of Fig. 2.1d, there is the intensity scale, which is divided into eight discrete intervals, sorted from black to white. Each of the eight intensity intervals is assigned with a specific value. The continuous intensity levels are quantized by assigning one of the eight values to each sample. Figure 2.1d shows the digital samples resulting from both sampling and quantizing. Beginning at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image. In addition to the number of discrete levels used, the accuracy achieved in quantization is highly dependent on the noise content of the sampled signal.

In representing a digital image, suppose that the continuous image is sampled into a two-dimensional function, $f(x, y)$, containing M rows and N columns, where (x, y) are discrete coordinates. These discrete coordinates are represented with integer values, where x is ranging from 0 to M, and y is ranging from 0 to N. In general, the value of the function at any coordinate (x, y) is represented as $f(x, y)$, where x and y are integers between 0 and M, and between 0 and N, respectively. As a result, the origin point of a digital image is $(0, 0)$ and the next coordinate along the first row is $(0, 1)$. Furthermore, there are two types of resolutions when representing a digital image. The first type is the *spatial resolution*. Spatial resolution is a measurement of an image size. Another type of resolution is the *intensity resolution*. Similarly with the spatial resolution, intensity resolution refers to the smallest discernible change in intensity level in an image. The number of intensity levels usually is an integer power of two. The most common intensity level is 256, ranging from 0 to 255. It is a common practice to refer to the number of bits used to quantize the intensity level as the intensity resolution. For example, an image whose intensity is quantized into 256 levels has 8 bits of intensity resolution, since 256 is equal to 2^8.

2.1.2 Digital Image Processing

Digital image processing [1] is a subcategory of *digital signal processing*. It is the use of *computer algorithms* to perform *image processing* on a digital image. Digital image processing sometimes can be confused with computer vision. Although the goal of computer vision and digital image processing is not the same, there is no obvious boundary between them. However, there is one way to categorize computerized processes into three levels, which are low-level processes, mid-level processes, and high-level processes, respectively, to give a better idea of how they are different from each other. Digital image processing only include processes from the low-level, and some from the second level. In the low-level processes, the input will be an image, and the output will also be an image. Some examples of low-level processes are image preprocessing to reduce noise in an image, contrast enhancement, and image sharpening. In mid-level processes, the input is still an image, but the output can be an image, information regarding the image, or characteristics regarding the image. Some examples of mid-level processes are image segmentation, and image classification or image recognition. Finally, in higher-level processes, the input can be anything from an image, information regarding the image, or characteristics regarding the image. The high-level processes will able to recognized an image. For example, the high-level processes can reconized a person in an image.

2.1.3 Image Segmentation

Image segmentation process [1] is categorized in the mid-level in computerized processes, since the input of this process is an image, but the output is information extracted from the image. Image segmentation divides an image into smaller regions or desired objects. This process carries on until it meets the detail level required by the problem being solved. In other words, image segmentation will stop when the objects or regions of interest have been detected. Image segmentation is a very important process, since it is one of the most earliest process done to an image, and its accuracy determines the eventual success or failure of further processes.

2.1.4 Genetic Programming

Genetic programming (GP) is a variant of *evolutionary algorithm* (EA), which adapts the concept of Darwin's principle of *evolutionary based natural selection* [3]. Eeach program is treated as a possible solution to the problem (individual). A group of individual programs is called a *population*.

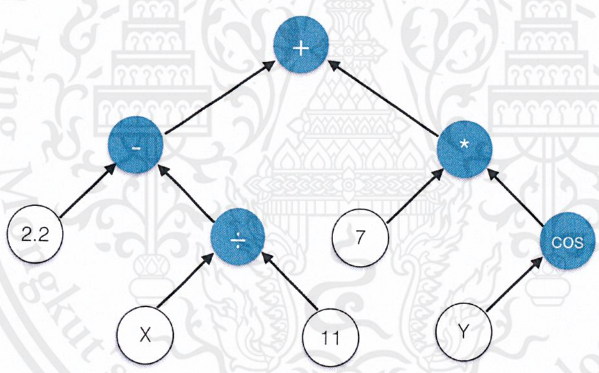


Figure 2.2: A tree structure representing an individual program in genetic programming

To represent each individual program, a tree structured is used. In Fig. 2.2 represents a single program, each terminal node (leaf nodes, white colored circle) is an input of the program. The other nodes are represented as an operation, for example, multiplication, summation, subtraction, etc.

In order to obtain the output of a program, the execution will start in a manner of bottom up procedure. The output will be obtained immediately after executing the root node. Thus, the result of the program in Fig. 2.3:

$$(2.2 \times (X \div 11)) + (7 \times \cos(Y)). \tag{2.1}$$

To determine how well each individual program solve the problem, fitness value for each individual is obtained and used for this purpose by executing every individual within the population. The method for obtaing fitness value will be furthur explained in the next chapter.

A general idea of how GP and other kind of EA operates can be described in the following procedures [2]:

1. **Initialization**: Randomly generate an individual program and group them together to build an initial population.

2. **Evaluation**: Go through and execute each individual program and obtain fitness value.

3. **Selection**: Select two individual programs based on the fitness values of each individual.

4. **Recombination**: Create new individuals from the selected individuals by applying genetic operations.

5. **Termination Criterion Checking**: Repeat the proceess starts from evaluation activity until the termination criteria is met. If it is, return an individual with the best fitness value.

2.1.5 Linear Genetic Programming

Linear GP is a type of GP in which a program is not a tree-based structure, but it is a sequence of instructions. These instructions will be executed in a consecutive order and operate on a set of registers. In conventional linear GP, registers are used to store numerical values.

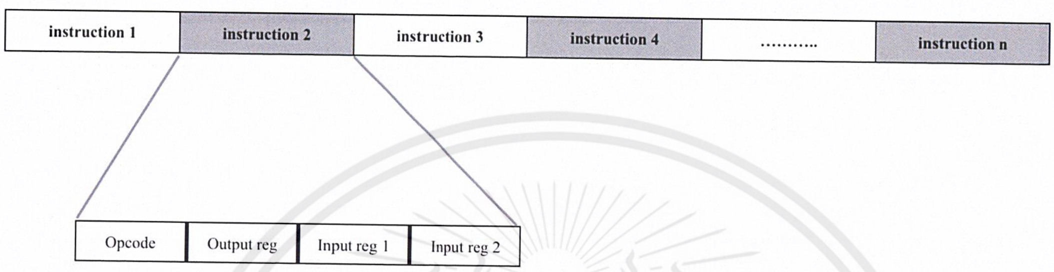


Figure 2.3: A typical representation of a program in genetic programming

As shown in Fig. 2.3, a typical instruction consists of:

- Operation code: indicates which operation to use.
- Output register: indicates which register to store the output after the operation.
- Input register 1: specifies which register to fetch the first input from.
- Input register 2: specifies which register to fetch the second input from.

2.2 Literature Reviews

2.2.1 Tomato Potassium and Nitrogen Deficiency

Xu et al. [4] studied and provided a diagnosis method for tomato disease caused by nutrient deficiencies, which are Nitrogen and Potassium.

Genetic Algorithm (GA) was used to find an optimal combination of features because less important features may decrease the classification capability.

There were three classes of leaf images, which are normal leaf images, Nitrogen deficient leaf images, and Potassium deficient leaf images. The images of different leaves were captured daily to cover deifferent degrees of deficiency symptoms. Eighty images from each classes were selected randomly. Fourty images from each were used as a test set, while the other fourty were for testing.

The features used were as follows:

- Intensity features using b* method [4]to distinguished characteristics between a pair of image classes.
- Texture features using the percent histogram of difference operator (Gradient operator, Robert operator, Sobel operator and Laplacian operator) method.
- Texture features using the *Fourier* rectangular annulus direction spectrum energy percent method which is using knowledge of rectangular annulus to calculate circumferential direction Fourior spectrum energy.
- Texture features using wavelet packet decomposition method. Dividing the image into components with different scales based on energy signals called wavelet spaces. This method can detect the spots appear on potassium deficient leaves.

Genetic Algorithm (GA) was used to select a set of effective features from previously mentioned which are an optimum combination of features. These features were encoded into a binary code which needs the chromosome length of twelve. The occupants of each population is equal to 20. Twenty chromosomes were produced by a random function. Thus, initial population was obtained. The number of evolutionary generations was 800.

To compute degree of fitness, the individual fitness was defined using the following equation:

$$J(x) = Sb - Sw \tag{2.2}$$

From Eq. (2.2), Sb is the between class variance while Sw is the within class variance. In this case, big value of Sb and small value of Sw was prefered. Then, the influence of dimension was eliminated by normalizing the feature vector which replaces x with x':

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.3}$$

where x is an arbitrary sample characteristic value of each class of feature, and x_{min} and x_{max} are the minimum and maximum characteristic values of each feature.

After the optimization using GA, the value of individual code was obtained. The feature combination was then used as an input to *fuzzy K-NN classifier* which the value of K is seven. The accuracies from recognizing normal leaves were 95%, 92.5%, and 92.5% for normal leaves, Nitrogen deficient leaves, and Potassium deficient leaves, respectively.

2.2.2 Lettuce Calcium Deficiency Detection

Ushada, Murase, and Fukuda [5] wanted to achieve two things. The first thing was to develop a methodology using morphological, textural, and temporal plant features with machine vision system for the automated non-contact monitoring of plant health and growth. The second thing was to evaluate the developed methodology for detecting tip burn resulted from calcium deficiency in a greenhouse-grown lettuce crop.

In image acquisition, to acquire an image, first a host computer will retrieved a list of target locations from a database. After that, the host computer sent a signal to a remote machine to position the image acquisition system to move to a specific location. When the camera is positioned at the center of the plant container which is the region of interest, five sequential images were then taken. After the images were acquired, image averaging will be applied to reduce noises and other disturbances in the acquired images.

Then, the images were segmented and the results were *binary images*. A binary image is a digital image that has only two possible values, intensity, for each pixel. The two possible values are 0 and 1, which represent black and white, respectively. The black color represents a background in an image, while the white color represents a foreground in an image. In this case, the foreground of the image is a plant canopy. This allows the plant portion to become visible. After analyzing the color feature, textural features were then extracted. However, four of texture features were proved to be useful for detecting the deficiency [6]. Thus, the final feature parameters were used in the experiments as follows:

- One morphological feature obtained by image segmentation method (the binary image) top projected canopy area (TPCA): represents the overall canopy expansion rate.

- Four textural features obtained by applying gray-level co-occurrence matrix.
  - Entropy: the randomness of gray-level distribution.
  - Energy: a numerical value represented by the level of gray-scale brightness.
  - Contrast: a measure of the local variations in an image.
  - Homogeneity: the determination of the related gray-level pixel distribution amongst the surrounding pixels in the canopy.

In order to determine the onset stress early, statistical computations using dual-segmented regression analysis was applied. The estimated change point of top projected canopy area, entropy, and homogeneity were on day 9.9, and Energy on day 9.7. Even though, contrast parameter was on day 9.5, the observed result in an interval (7.5, 11.5) caused doubt in its reliability.

Therefore, from statistical analysis, top projected canopy area, entropy, and homogeneity were promising features which can be observed as a warning sign of calcium deficiency, because the machine vision system can detect the sign on the 10th day, which is one day earlier than with using human eyes.

2.2.3 Conclusiton

In both research, the working systems were automatic but still needed to study the deficiency symptoms and characteristics to find a suitable features in which require an extent of problem expertise and observation. The first research paper from Xu et al.([4]) used the human determined features and genetic algorithm to select the most effective feature combinations out of the provided ones again. Similarly, in Section 2.2.2, Ushada, Murase, and Fukuda also used the a set features to that they determined to detect Calcium defeiciency in lettuces.

Chapter 3

Methodology

This chapter explains the methodology for implementing the application (Section 4.1.2) and the algorithm for detecting nutrient deficiency in a plant (Section 3.3).

3.1 System Overview

In the proposed system, there are two components, which are the mobile application and the processing server, combining together (see Section 1.4). The mobile application and the processing server are communicating between each other through the Internet, as illustrated in Fig. 3.1. If there is no Internet connection, some functions of the system will not be able to work.

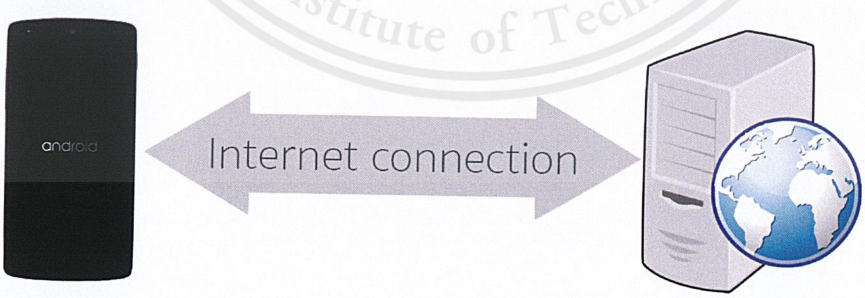


Figure 3.1: System overview

The main feature of the program is the function that allows users to analyze a nutrient deficiency in a plant. In order to make this function work, it requires collaboration between both the mobile application and the processing server.

First of all, the user takes photos of an old leaf and a young leaf of a plant needed to be analyzed. Then, the mobile application will send those images to the processing server. The reason for requiring both type of leaf is because of some nutrient are classified as a mobile nutrient which may traverse thorugh both type of leaf at differnt time therefore, the symptoms of each deficiency type can also varied periodically on both leaf as well.

However, the images sent from android application are the ones that have already been segmented by the segmentation algorithm programmed in the application itself. It is an image segmentation method which require user interaction by letting user draw contours around the leaf image (see Section 4.1.2) and it will automatically determine the boundary of leaf image. If and only if the result satisfied user, the segmented result will be sent to the server. The segmentation algorithm will be explained later in Section 3.2.

After the processing server received the images from the mobile application, there is an mechanism located within the server used for detect the nutrient deficiency. The processing server will receive two input images from the mobile application to process further. These images are old leaf and young leaf image in which may need different set of features in order to classify both type of leaves.

However, which feature and what kind of procedure to extract them is the problem. Therefore, evolutionary algorithm was studied and used as the approach to construct these two feature extraction programs.

The evolutionary run will be done offline to build a pair of optimized feature extraction programs before establishing any connections with the mobile application. After that, the classifier will be trained and created a nutrient deficiency detection program in which the structure of the program is described in Fig. 3.2. This is the actual program that will receive the input images.

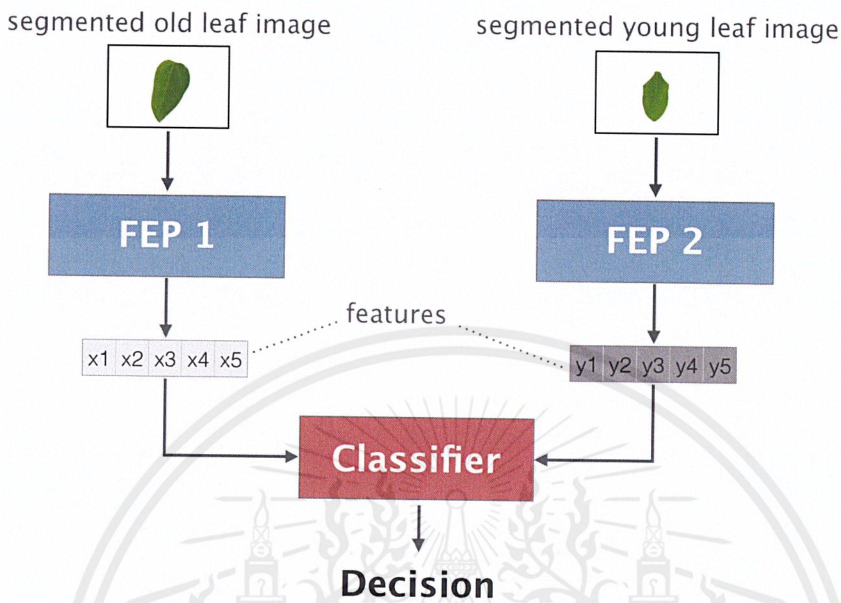


Figure 3.2: Block diagram

3.2 Leaf Segmentation

Generally, image segmentation can be divided into five categories [11], which are threshold-based segmentation, edge-based segmentation scheme, region-based segmentation, watershed-based segmentation, and energy-based segmentation.

To begin with, threshold-based segmentation [13, 14] is carried out by separating the image into two distinct parts which are foreground and background. The separation is done by determining each pixel in the image. If the pixel has an intensity value higher than the threshold value, it is classified as foreground. Otherwise, the pixel is classified as backgroupoond. Threshold-based segmentation is the simplest, easiest, and fastest segmentation method compared to other categories. However, this method has some difficulties. First, finding a proper threshold value is not easy. However, Otsu's algorithm [1] can overcome this difficulty and is the most widely used method for finding the threshold value. Another difficulty is that this method requires the image to has clearly different foreground and background intensities.

For the edge-based segmentation scheme, it is assumed that there are intensity differences in the pixels that connect foreground and background together. Therefore, the discontinuities can usually detected by the first and/or second order derivatives method, such as, gradient and Laplace [13–16]. Some of the most popular edge detection methods are Sobel, Roberts, Prewitt, Laplace of Gaussian, and Canny edge detectors [1, 13–16]. However, these methods sometimes include false edges to the result. Therefore, post-processing operations are usually required for the edge-based segmentation to eliminate the false edges.

The third category is the region-based segmentation [13, 14, 17]. Some of the common algorithms are region growing and region splitting-merging [1]. For the region growing theory, first, a group of origin pixels must be defined. Afterward, the adjacent pixels are grouped to the origin such that the pixel satisfies predefined criteria, such as, intensity, color, or texture. Therefore, the selection of the origin pixels is crucial for region growing scheme. For the region splitting-merging process, the image is split into many un-overlapped regions until it cannot be split anymore [11]. After that, these regions are merged with the adjacent regions such that it satisfies to predefined criteria.

Next, the watershed-based segmentation [1, 13, 14, 18] views the image as a topological surface and the intensity value as height [11]. The regional minimal values in the image are interpreted as catchment basins and the maximal values between every two neighboring catchment basins are viewed as ridge line. Watershed-based segmentation is to find the ridge line called watershed within the image. Consequently, in order to extract an object, watershed transform algorithm is usually applied to gradient image where the object is corresponding to the catchment basins while the boundary to watershed.

The last category is the energy-based segmentation. In this method, the energy function is established. This function will reach a minimal value when the image is segmented. Some of the algorithms that are categorized into this category are live wire [19], active contour [20], level sets [21, 22], and graph cut [23, 24]. For the graph cut segmentation, the energy function is constructed based on regional and boundary information, which can achieve globally optimal result. Boykov and Jolly proposed the graph-cut segmentation method in 2001 [23]. Since then, many varied methods based on graph-cut have been developed.

In this system, the technique called GrabCut is chosen as the segmentation algorithm. GrabCut [12] is an *image segmentation* (see Section 2.1.3) methods used for extracting a foreground or a desired object from background in an image. GrabCut segmentation is an extended image segmentation method based on graph-cut.

3.2.1 Concepts of Graph-Cut Segmentation

Let an *undirected graph*, a graph where all edges are bidirectional, denoted as $G = (V, E)$, where V is a series of nodes and E is an edge of the graph which connect every two adjacent nodes [11]. The vertex V is consist of two different types of nodes. The first type of node is neighborhood nodes. The neighborhood nodes are consisted of every pixels in the image. The other kind of nodes are called terminal nodes. The terminal nodes are consisted of s (source) and t (sink). This variety of graph is called a s-t graph. As shown in Fig. 3.3, the s node illustrates the object (foreground) and t node represents the background. In this type of graph, there are also two kinds of edges. The first kind of edges is called n-links. The n-links connect the adjacent pixels with in the image. The other kind of edges is called t-links. The t-links connect the terminal nodes with the neighborhood nodes. In the graph, each edge is assigned with a non-negative weight denoted as w_e. A *cut* is a subset of edges E which can be denoted as C and expressed as $C \subset E$. The cost of a the cut $|C|$ is the sum of the weights on edges in C

which is expressed as shown in Eq. (3.1)

$$|C| = \sum_{e \in C} w_e \tag{3.1}$$

A *minimum cut* is the cut that have the minimum cost, called min-cut. It can be achieved by finding the *maximum flow*, called max-flow, as was verified in [23–25] that the min-cut is equivalent to max-flow. The max-flow/min-cut algorithm developed by Boykov and Kolmogorov [25] can be used in order to find the minimum cut for the s-t graph. Therefore, the graph is divided by this cut separating the nodes into two disjoint subsets S and T, where $s \in S$, $t \in T$, and $S \cup T = V$. The two subsets represent foreground and background in the image. This kind of graph is illustrated in Fig. 3.3.

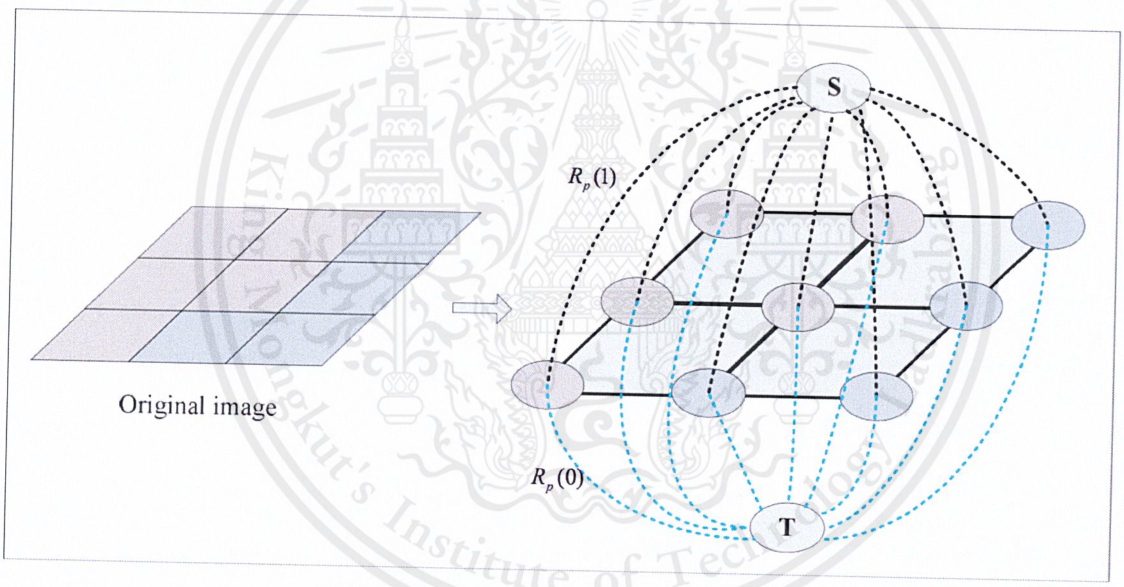


Figure 3.3: s-t graph. Each pixels in the image is correspond to the nodes in the graph (except s and t nodes). The solid lines in the graph are n-links and the dotted lines are t-links. (image from [11])

3.2.2 Graph-Cut Segmentation in Details

Image segmentation can be seen as pixels labeling problem. The label is set to be 1 for pixels representing the object (foreground)(s-node) while it is set to be 0 for pixels denoting the background (t-node) [11]. This process can be accomplished by minimizing the energy function through minimum graph cut. In order to make the segmentation acceptable the cut should be occurred exactly at the boundary between the object (foreground) and the background. That is to say, at the object boundary, the energy (cut) should be minimized. Let $L = \{l_1, l_2, l_3, ..., l_i, ..., l_p\}$ where p is the number of the pixels in the image and $l_i \in \{0, 1\}$. Thus, the set L will be divided into two parts, which are the pixels labeled with 1 and the pixels labeled with 0. The pixels labeled with 1 belong to the foreground while the pixels labeled with 0 are categorized into the background. Equation (3.2) shows the energy function which can be minimized by the min-cut in the s-t graph [23, 24].

$$E(L) = \alpha R(L) + B(L) \tag{3.2}$$

From Eq. (3.2), $R(L)$ is called regional term, which regional information incorporated into the segmentation, $B(L)$ is called boundary term, which the boundary constraint incorporated into the segmentation, and α is the relative importance factor between regional and boundary term. If α is set to 0, the regional information will be ignored and the segmentation will only consider the boundary information. In Eq. (3.3), the regional term is defined.

$$R(L) = \sum_{p \in P} R_p(l_p) \tag{3.3}$$

From Eq. (3.3), $R_p(l_p)$ is the penalty for assigning the label l_p to pixel p. The weight of $R_p(l_p)$ can be obtained by comparing the intensity of pixel p with the given intensity model of the object (foreground) and background.

$$R_p(1) = -\ln Pr(I_p|'object') \tag{3.4}$$

$$R_p(0) = -\ln Pr(I_p|'background') \tag{3.5}$$

Equations (3.4) and (3.5) define the weights of the t-links. As can be observed in Eqs. (3.4) and (3.5), if $Pr(I_p|'object')$ is larger than $Pr(I_p|'background')$, $R_p(1)$ will be smaller than $R_p(0)$. As a consequent, when the pixel is more likely to be the object, the penalty for grouping that pixel into the object should be smaller, which can reduce the energy in Eq. (3.2). Therefore, after all pixels in the image have been correctly separated into two subsets, the regional term would be minimized.

$B(L)$ in Eq. (3.2) is the boundary term which has the definition defined in Eq. (3.6)

$$B(L) = \sum_{\{p,q\}\in N} B_{(p,q)} \cdot \delta(l_p, l_q), \tag{3.6}$$

where p, q is neighboring pixels and $\delta(l_p, l_q)$ is defined as:

$$\delta(l_p, l_q) = \begin{cases} 1 & \text{if } l_p = l_q; \\ 0 & \text{if } l_p \neq l_q. \end{cases} \tag{3.7}$$

For regional constraint, it can as assigning labels l_p, l_q to the neighboring pixels. If the neighboring pixels have the same label. the penalty is 0, which means the regional term would only sum the penalty at the segmented boundary.

The term $B_{(p,q)}$ is defined in Eq. (3.8) as a non-increasing function of $|I_p - I_q|$

$$B_{(p,q)} \propto \exp(-\frac{(I_p - I_q)^2}{2\sigma^2}), \tag{3.8}$$

where σ is the camera noise. When the intensity of two neighboring pixels is very similar, the penalty is very high. On the other hand, it will be low. Therefore, when the energy function obtains the minimum value, it is more likely occurred at the object boundary. In [23, 24], Boykov and Jolly have showed that the minimized energy can be computed by min-cut through max-flow. Thus, the minimum energy problem is converted into the graph cut problem. In order to get a satisfied segmentation result, the weight assignment in the s-t graph is crucial. In their method, the weight of the s-t graph is assigned using the following equation:

$$weight = \begin{cases} B_{(p,q)} & \{p, q\} \in \text{neighboring pixels;} \\ \alpha \cdot R_p(0) & \text{for edge } \{p, S\}; \\ \alpha \cdot R_p(1) & \text{for edge } \{p, T\}. \end{cases} \tag{3.9}$$

Equation (3.9) explains, in the s-t graph, that if the intensity of the pixel is inclined to be the object, the weight between this pixel and s-node will be larger than that between the pixel and t-node. Thus, the cut is more likely to occurred at the edge with smaller weight. Furthermore, for the neighboring pixels, when their intensity is very similar, the weight becomes very large, which is not likely to be separated by the cut. Therefore, when the minimum cut is obtained from the s-t graph, the location of the cut is close to the object boundary The implementation of the graph cut can be achieved by the max-flow/min-cut as stated in [23–25]. Figure 3.4 gives an illustration of the graph cut for a 3×3 pixels image segmentation. The thickness of the edge represents the magnitude of the weight.

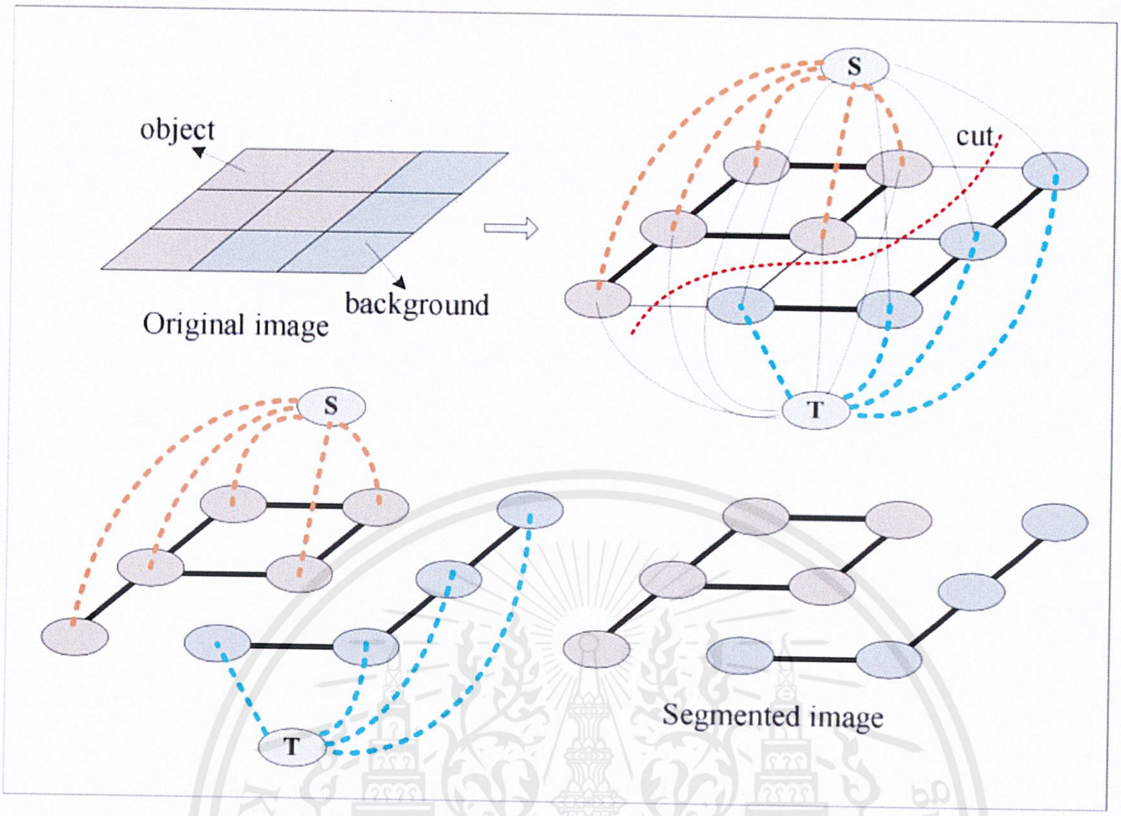


Figure 3.4: Graph-cut segmentation. (image from [11])

3.3 Construction of Nutrient Deficiency Detector

Linear Genetic Programming (LGP) is applied to construct a feature extraction program. It will be used to extract a set of plant features in order to classify a plant's deficiency condition. However, the nature of the problem stated that two input images are required from each plant in order to classify the deficiency. Thus, the concept of *Cooperative Coevolution* (CC) is adapted.

CC is an extention of EA. It describes a model of an ecosystem which consists of different species.

In this project, a combination of cooperative coevolution and linear GP, denoted as CCLGP, similar to the one explained in Section 2.1.4 is used to construct feature extraction programs. However, some procedures are ordered differently for this task. CC is different from other types of EA in which one evolutionary run consists of multiple populations and another difference is that population encodes only a part of the solution to the problem [3]. In CC, this partly encoded solution will be refered as sub-individual.

3.3.1 Initialization

For this task, a program takes two images as inputs (a young leaf and an old leaf from the same plant) in order to produce the output. In this step, members of population are generated randomly.

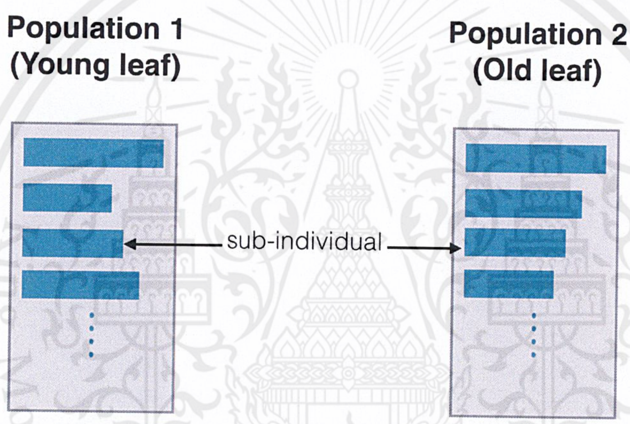


Figure 3.5: Population initialization

Two populations are generated with population members are conceptually called sub-individuals instead of individual because of the fact that a single linear program can not be used to extract features from both type of leafs. In CCLGP and in this project, a pair of linear program is required to evaluate the effectiveness of nutrient deficiency detection. Therefore, one population will be used for constructing a linear program for extracting young leaf features and another population for old leaf's.

A sub-individual (linear program) is a sequence of instructions with variable lenght. Each sub-individual will have its own set of registers for instructions to operate on when executed. The result of the execution will be a vector of extracted features. Sub-individuals in the first population are responsible for extracting features of young leaf images while the ones in the second population are also responsible for extracting features of old leaf images.

Registers are used as a temporary storage to store the inputs and outputs which each operations can operate on during the evolutionary process. They can be divided into different types of registers based on the datatype that can be store in each unit. In this project, there are two types: image registers and numeric registers. An image register stores an input image or a processed image which is an output from an operation. On the other hand, a numeric register stores a number, eg., integer or float, which results from performing primitive or image processing operations (see Table 3.1).

3.3.2 Selection and Recombination

The next process is to *evolve* the populations. This can be done by selecting two candidate sub-individuals within the same population and create new pair of sub-individuals based on the selected candidates using *one-point crossover* [2] (Fig. 3.6). A part of parent sub-individuals are exchanged according to the crossing point. However, the crossing point of each sub-individuals can varied as the lenght of a sub-individual is not fixed. It's also chosen among the joints of each sub-individuals which means that an individual instruction will not be cut apart. Repeating the process until the number of new sub-individuals is sufficient. Thus, the new population is obtained and can be used for further iteration processes.

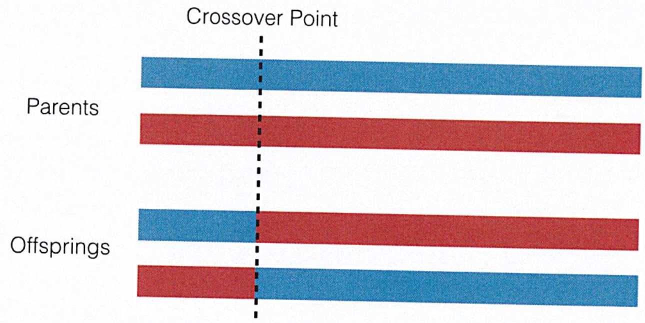


Figure 3.6: One-point crossover

3.3.3 Evaluation

To obtain the fitness value, the evaluation proceess starts with combining the results from executing each sub-individuals, which are feature vectors, from the first population with the one from the second population. This means combining the extracted features of young leaf and old leaf together to classify the Black Gram's leaf images.

Executing a sub-individual means executing a set of instructions by performing operations (Table 3.1) in which can be devided into different types based on its inputs parameter and output. As shown in Fig. 3.7, registers will be used to store inputs and outputs while the operations are underway. The implementations of these operations will be defined.

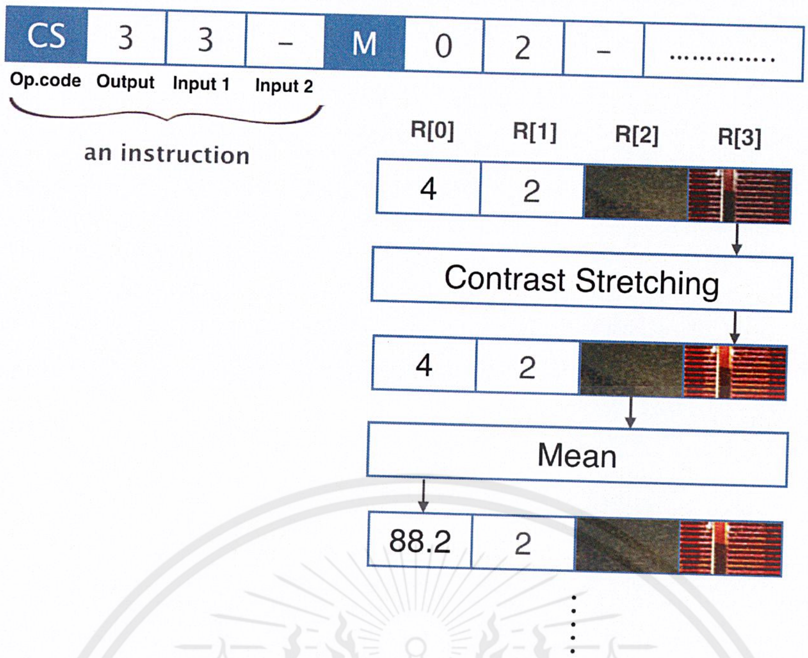


Figure 3.7: An example of instructions execution process within a sub-individual

After sub-invididual executions, the resulting feature vectors are used for classification process. Input the vectors and training images into classifier. Then, it will use the features to judge each of the training images (see Fig. 3.8). The results will fall under one of six categories, which are Fe, N, Ca, Mg, K deficiencies or complete (no deficiency). Then, the ground truths are used to measure the correctness of the results. In this case, *recognition percentage* (Eq. (3.10)) is used.

$$\frac{\text{no. of correctly classified leaf images}}{\text{no. of all the training within the same dataset}} \times 100 \qquad (3.10)$$

The recognition percentage of each individual will also become its *fitness value* in the end. Thus, obtaining fitness value of every individuals marks the end of evaluation process.

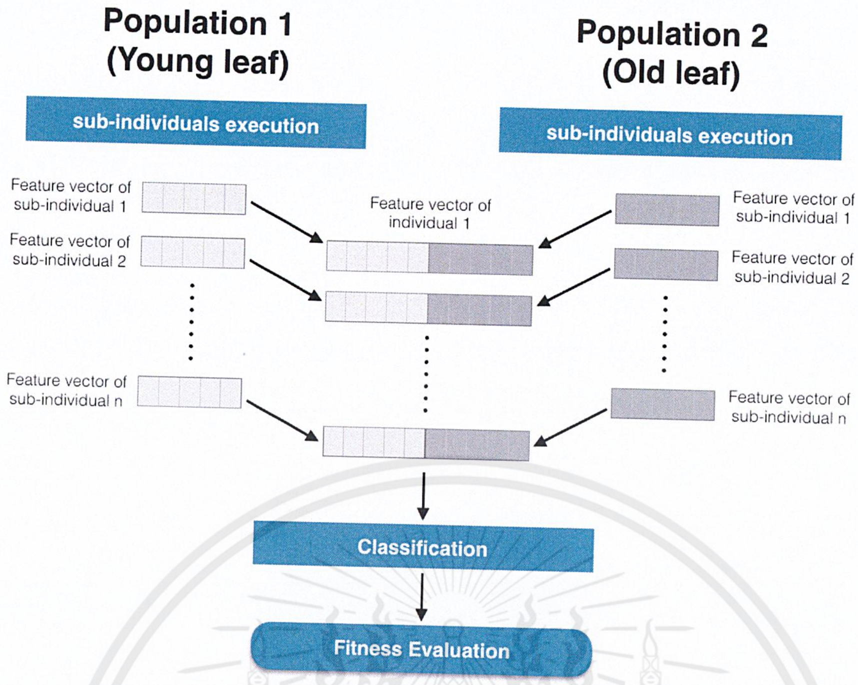


Figure 3.8: Summary of evaluation process

3.3.4 Termination Criteria Checking

This process simply checks fitness value of each individual if one of them reaches a certain threshold or not. If there is one or more individual where fitness value satisfies the condition, then the process is terminated. The appropriate set of features is obtainined. Otherwise, the iteration resumes by starting at selection and recombination process (Section 3.3.2) with the current populations. However, in this project, a value of generation limit will be set as a termination criteria.

Table 3.1: Different types of operations from primitive and image processing

Image $\longrightarrow$ Numeric	Image $\longrightarrow$ Image
Global mean Global variance Global standard deviation Global skewness Global kurtosis Global maximum Global minimum Global median Global mode Global range	Highpass filter Sobel operation Image negative Mean thresholding Entropy thresholding Histogram equalization
Image, Numeric $\longrightarrow$ Image	**Image, Image $\longrightarrow$ Image**
Lowpass filter Median filter Morphological dilation Morphological erosion Morphological opening Morphological closing Local histogram equalization Local thresholding Local variance Local skewness Local kurtosis Local maximum (max filter) Local minimum (min filter) Local mode Local range Local entropy	Image addition Image subtraction

Chapter 4

System Analysis and Design

This chapter describes system analysis and design including what the system can do, how does the interface look like, and how the system is modeled.

4.1 System Analysis

4.1.1 Requirements

The requirements of the system are shown in Table 4.1. Requirements of the system are classified into *FURPS+* categorization model. The FURPS+ model separates requirements into six categories, which are functionality, usability, reliability, performance, supportability, and plus. The plus category includes design constraints, implementation requirements, interface requirements, and physical requirements. Functionality defines what a user can does with the system. Usability is the system effectiveness from the perspective of the person who use it, or the user. Reliability defines the stability of the system. Performance defines the speed of the system. Supportability defines whether the system is testable, extensible, serviceable, installable, and configurable or not. Design constraints are the limitation of the software and/or hardware that the system must follow. Interface requirements tell that if there is any other systems that this system must be communicate with. Finally, physical requirements are the hardware specifications that the system must be deployable on. In this system, there are a total of 25 requirements. There are six functionality requirements, three usability requirements, one reliability requirement, one performance requirement, seven design constraints, and seven physical requirements.

For the functionality requirement, there are a total of six requirements in this category. First, the application allows the user to take pictures within the application so that the system can analyze the pictures for the user. Next, the application must contains five nutrient deficiencies information. Next, the application allows the user to search for nutrient deficiencies information in the browse page of the application. The user can view each nutrient deficiency in detail. Next, the application allows the user to select regions of interests for a better analysis result. Next, the application allows user to reselect regions of interests by clear the previous regions of interests. Lastly, the application must return the result to the user.

For the usability requirement, there are three requirements in this category. To begin with, the interfaces of the application are designed to be easy to understand. Next, the application must works properly with different screen sizes. Finally, there is a tutorial provided within the application.

For the reliability requirement, there is only one reliability requirement in this system. It stated that the system must be running 24/7 with minimum downtime as possible.

For the performance requirement, there is only one performance requirement in this system. The performance of the system is that the server should be able to handle 10 simultaneous connection from different users.

For the design constraint, there are ten design constraints in this system. Beginning with the operating system support by the application, the application is supported on Android 4.1 - 4.3 (JellyBean), Android 4.4 (KitKat), Android 5.0 - 5.1.1 (Lollipop), and

Android 6.0 - 6.0.1 (Marshmallow). Next, there are several constraints for the server to work perfectly. It requires OpenCV 2.4.11, Apache server, MySQL version 5.6.X, PHP version 5.6.X, and Windows 10. Lastly, the server must runs with full administrator privilege.

For the physical requirement, there are seven requirements in this category. First, some features, which are analyze plant and view nutrient deficiency information, require Internet connection to work. Next, there are some specifications required by the application to work flawlessly. The minimum processor speed is 1.0GHz single-core. The minimum random access memory space is 512MB. The minimum storage space needed to install the application is 60MB. Next, taking pictures requires smartphone with camera to work. Lastly, the server needs to be deployed on a high performance computer.

Table 4.1: Requirements

Requirement	FURPS+ Category
The application allows user to take a picture.	Functionality
The application must contains five nutrient deficiencies information.	Functionality
The application allows user to search for a nutrient deficiency information.	Functionality
The application allows user to select region of interest from a picture.	Functionality
The application allows user to reset region of interest of the picture.	Functionality
The application must show a result of the analysis.	Functionality

continued ...

...continued

Requirement	FURPS+ Category
The interface of the application should be easy to understand.	Usability
The application should be able to display correctly on different screen sizes.	Usability
The tutorial will be able to view from within the application.	Usability
The server should be running 24/7.	Reliability
The server should be able to handle 10 connections simultaneously.	Performance
The application requires smartphones running Android 4.1 - 4.3 (Jelly Bean), Android 4.4 (KitKat), Android 5.0 - 5.1.1 (Lollipop), and Android 6.0 - 6.0.1 (Marshmallow).	Design Constraints
The server requires OpenCV version 2.4.11.	Design Constraints
The server requires Apache server.	Design Constraints
The server requires MySQL version 5.6.X.	Design Constraints
The server requires PHP version 5.6.X.	Design Constraints
The server requires Windows 10.	Design Constraints
The server requires full administrator accesses.	Design Constraints
Some features of the application require Internet connection to work.	Physical Requirement
The application requires smartphones with the screen resolution higher than 800 × 480.	Physical Requirement
The application requires smartphones with 1.0Ghz single-core CPU or faster.	Physical Requirement

continued ...

...continued

Requirement	FURPS+ Category
The application requires smartphones with 512MB of RAM or more.	Physical Requirement
The application requires smartphones with 60MB of free internal storage or more.	Physical Requirement
Some features of the application require smartphones with a camera to work.	Physical Requirement
The server must be deployed on a high performance computer.	Physical Requirement

4.1.2 User Interface

As stated in Section 1.4, this system contains two components. The first component is a mobile application and the second component is a processing server. The mobile application provides all the interactions between a user and the system. As explained in Section 1.5, the mobile application will available for Android smartphones. For this system, we adapt the user interface design from the Leaf Identifier project [7].

Figure 4.1: Homepage of the application

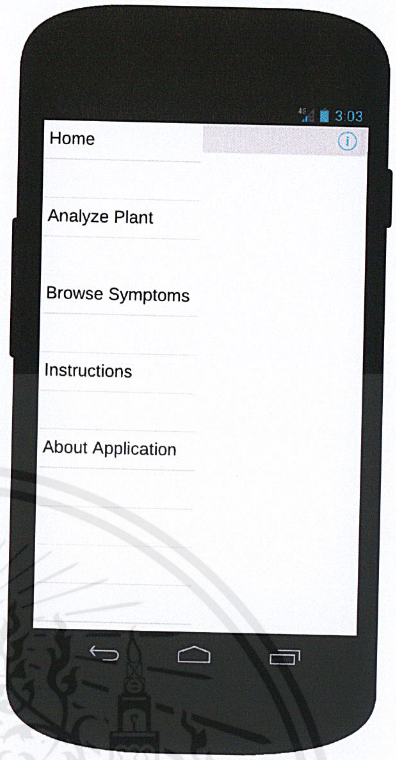


Figure 4.2: Navigation drawer

When a user initially launches the application, the application will land on the home page, as shown in Fig. 4.1. The homepage shows the logo of the application. The is also a toolbar located at the top of the screen. The toolbar contains three components. The first component, which is placed on the leftmost of the toolbar, is a navigation drawer button. The second component is the title of the current application screen, which occupies the center of the toolbar. The last component of the toolbar is the help button, which is situated on the rightmost of the toolbar.

The navigation drawer of the application, as shown in Fig. 4.2, will be opened if the user touches this button. The navigation drawer can also be opened by swiping a finger from the left of the screen. When it is opened, the user can close it easily by pressing the back button on the smartphone. In addition, the user can also close the navigation drawer by swiping a finger from the rightmost of the navigation drawer itself to the left of the screen. In the navigation drawer, there are five items listed as follow:

- **Home**: This button navigates the application to its homepage.
- **Analyze Plant**: This button goes to the analysis page of the application.
- **Browse Symptoms**: This button changes to the browse page of the application.
- **Tutorial**: This button switches to the tutorial page of the application.
- **About Application**: This button turns to the about page of the application.

Figure 4.3: Analysis page

Figure 4.4: Asking the user for the region of interest

If the user touches the analyze button, the application will redirect to the analysis page, as shown in Fig. 4.3. This page is composed up of the camera view and the take a photo button located at the bottom of the screen. After the user has taken a photo, the application will ask the user to draw a region of interest for that photo, as shown in Fig. 4.4. If the user does not satisfy with the region of interest that shows on the screen, there is a reset button located at the bottom-left on the screen. This button will reset the region of interest that the user drew on the screen. If the user does not satisfy with the taken photo, tapping the back button on the smartphone will navigate the application back to the analysis page, as shown in Fig. 4.3.

Figure 4.5: User-drawn region of interest

Figure 4.6: Application automatically adjust region of interest and display the result

When the user finished drawing the region of interest, as shown in Fig. 4.5, the user needs to press the continue button, which is lied at the bottom-right of the screen. If the continue button is pressed, the application will automatically adjust the region of interest to make it more precise. After the application finished adjusting the region of interest, the display will show as in Fig. 4.6.

Subsequently, the user can press the continue button located at the bottom-right of the screen. Then, the application will send all information, including photos and region of interest, to the processing server. When the processing server finish analyzing the information, the processing server will return the result to the mobile application and the application will display the result to the user, as shown in Fig. 4.7.

Figure 4.7: Information page shows information about a symptom

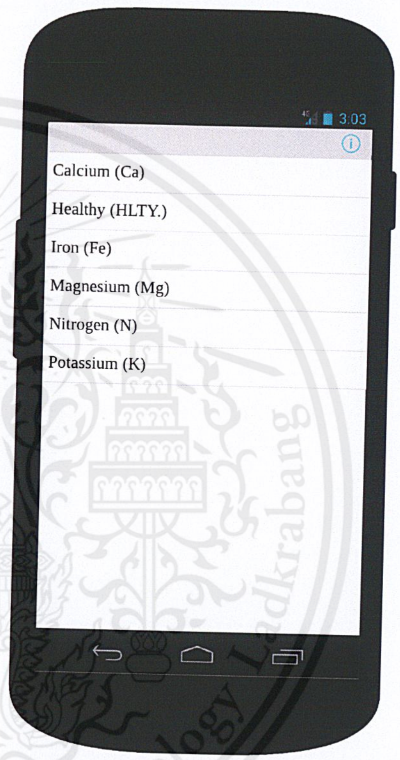


Figure 4.8: Symptoms browse page shows list of symptoms

Figure 4.7 shows information about each symptom of a *Black Gram*. There are two ways to reach the information page. The first way is from the analyze plant page. The second way is from the browse page, as shown in Fig. 4.8. This page composed up of three components. The first component is the name of the symptom that is showing, which is located at the top of the screen below the toolbar. The next component is the image gallery, which is situated below the name of the symptom. The image gallery shows images about the symptom. The user can change between images by swiping from left to right or right to left in the image gallery component. The last component is the detail, which is placed below the image gallery. The detail component shows information about the symptom.

The browse page, as shown in Fig. 4.8, contains a list of symptoms that are available in the database of this system. The user can view information about each symptom via this page.

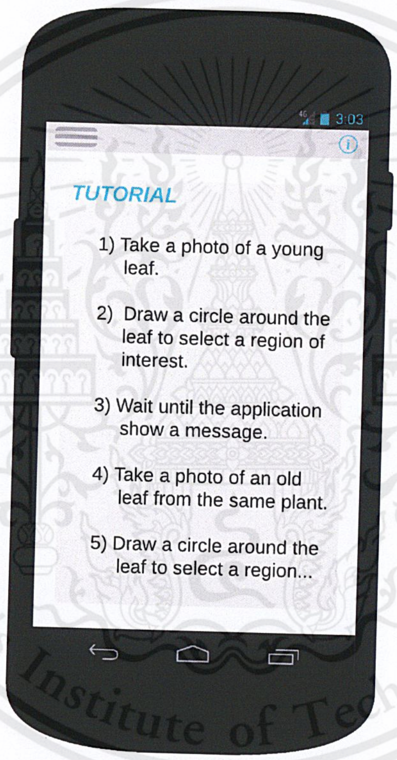


Figure 4.9: Tutorial page shows instructions of how to use the application

Figure 4.9 shows the tutorial page of the application. In this page, there are instructions that show the user how to use the application properly.

4.1.3 Use Case

The use case diagram of the system is shown in Fig. 4.10. There are six use cases in this system. First, a user can identify a nutrient deficiency in a plant by taking photos of that plant. After the user has taken the photos, the user has to select a region of interest for each photo. Then, the system will analyze the taken photos for the user. If the user want to know about a nutrient deficiency in a plant but does not want to analyze any plant, the user can search for a nutrient deficiency information using a nutrient name. Then, the user can view a nutrient deficiency description. In the description, there are information about that nutrient deficiency and some sample images of how the plant look like if the deficiency occurred. Lastly, the user can view tutorials, which tell the user how to use the application.

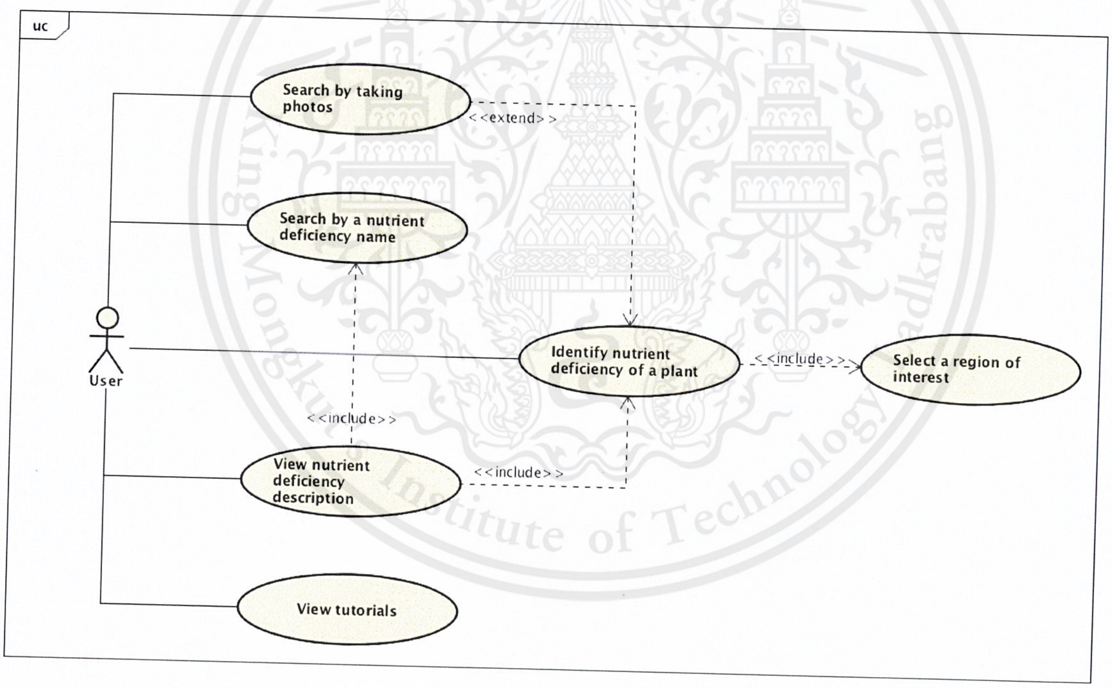


Figure 4.10: Use case diagram

4.1.4 Activity Diagram

The activity diagram of the system is shown in Fig. 4.11. After a user launch the application on their smartphone, the application will show the Home page. From the home page, the user can choose to view tutorials, open the navigation drawer of the application, or main menu of the application, or quit the application. If the user choice is to view tutorials, the application will show the tutorials to the user. If the user want to open the navigation drawer, the application will show the navigation drawer to the user. In the navigation drawer, there are five menus that can be chosen. The first menu goes to the Home page of the application. If the user taps on this menu, the application will navigate to the Home page. The second menu goes to the Analyze page. If the user wants to analyze a plant and taps this menu, the application will navigate to the camera interface that allows the user to take pictures. After the user had finished taking 2 pictures, one for an old leaf and one for a young leaf, and selecting region of interest for both pictures, the application will send all data to the processing server. Then, the server will analyze the received data. After the server finished, the server will return the analysis result back to the application. When the application receives the result from the server, it will display the result to the user. Next, the third menu navigates the user to the browse page. If the user chose to tapped on this menu, the application will show a list of symptoms that are available in the system database to the user. From the symptoms list, the user can tap on any symptom to view its description. Next, there is a menu to view the instructions of how to use the application. Lastly, if the user want to see information about the application, the user can tap on the last menu.

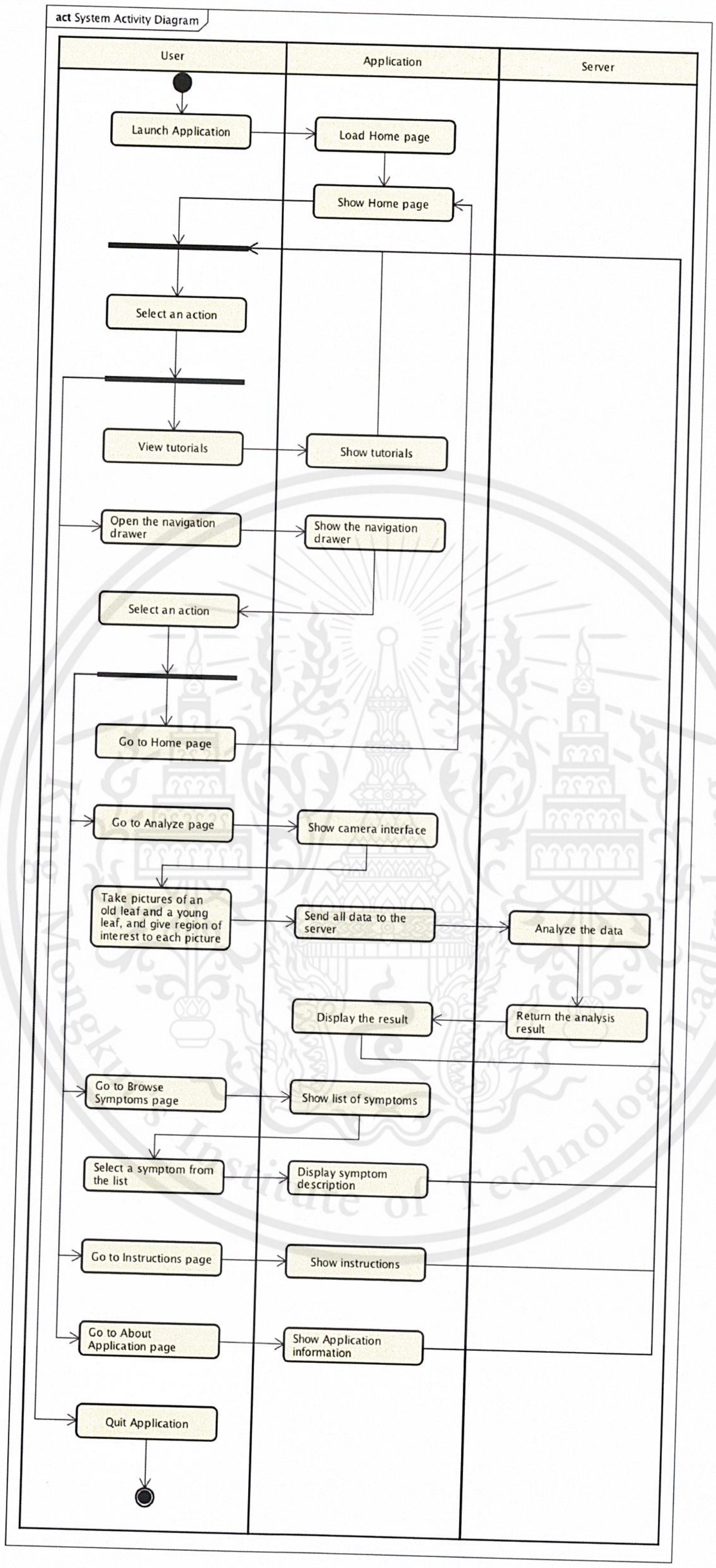


Figure 4.11: System activity diagram

4.2 System Design

4.2.1 Package Diagram

A package diagram shows *components* and *dependencies* between each component in the system. As shown in Fig. 4.12, there are three packages in this system, which are the Android System package, the Server System package, and the Nutrient Deficiency Analyzer System package. The three systems are organized using *Layered Architectural* style. In this architectural style, each layer knows the abstractions of the layer directly beneath it only. From the Fig. 4.12, the Android System package is at the top of the layer follows by the Server System package and the Nutrient Deficiency Analyzer System package, respectively. This arrangement implies that the Android System depends on the Server System and the Server System depends on the Nutrient Deficiency Analyzer System. The Android System will never know what happen between the Server System and the Nutrient Deficiency Analyzer System. The Nutrient Deficiency Analyzer System will also never know what happen between the Android System and the Server System.

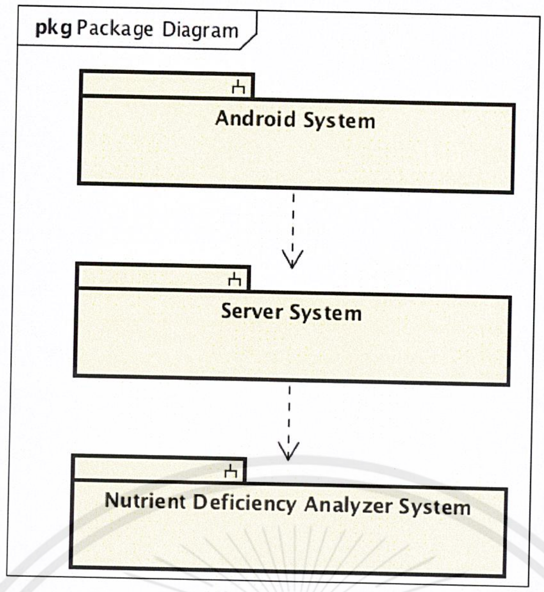


Figure 4.12: Package diagram

4.2.2 Class Diagram

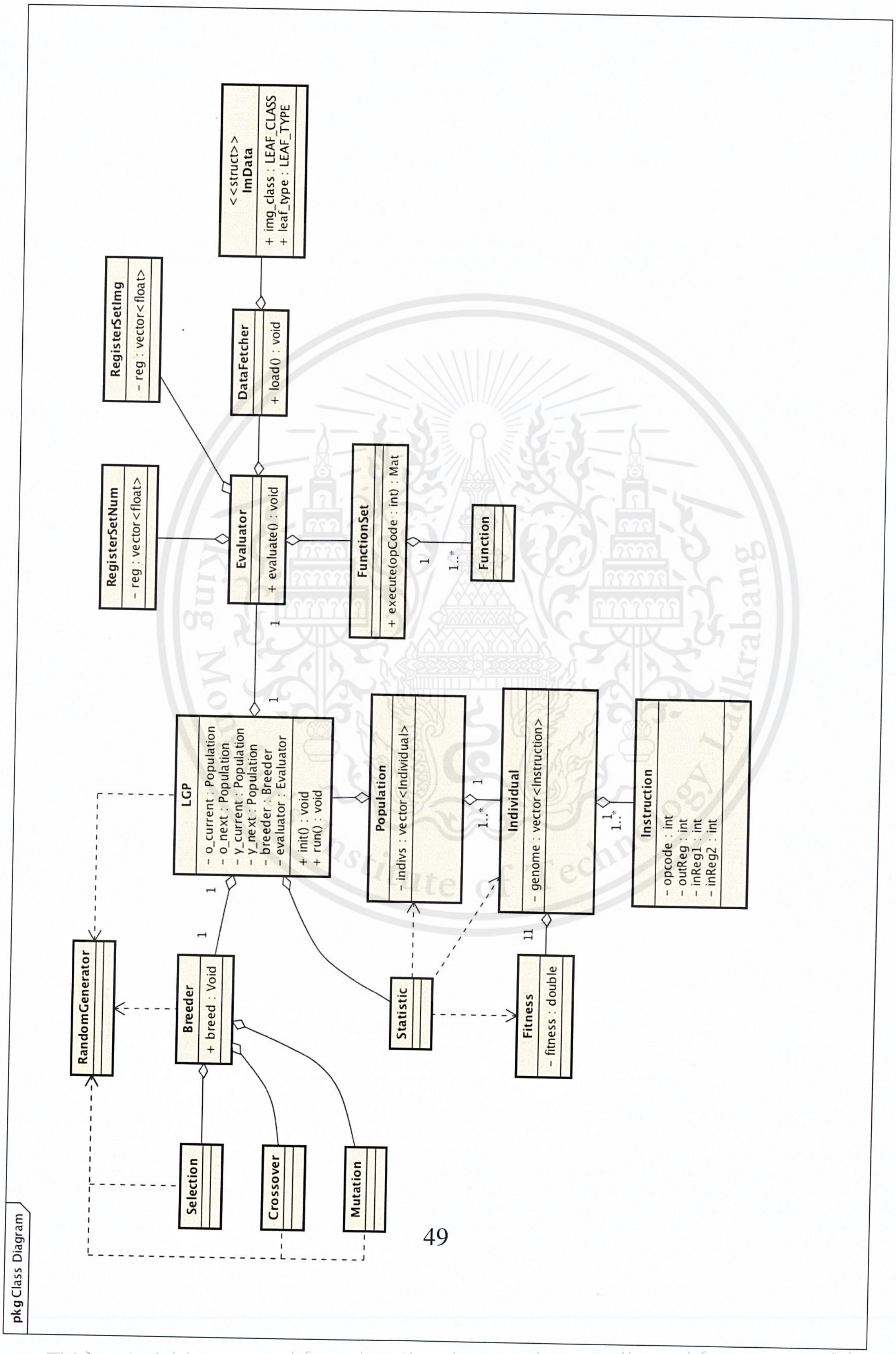


Figure 4.13: Class diagram

4.2.2.1 LGP class

LGP class responsible for running the whole evolutionary process. Containing primary classes for example, Breeder and Evaluator

Table 4.2: LGP class Attributes

Attribute	Type
*o_current	Population
*o_next	Populaiton
*y_current	Population
*y_next	Population
*evaluator	Evaluator
*breeder	Breeder
params	ParameterSet
stat	Statistic
randomGen	RandomGenerator

Table 4.3: LGP class operations

Operation	Return type	Parameters	Other details
init	void	void	initialize every attribute and parameters
run	void	void	starts the evolutionary run
evolve	void	void	evolve the populations

4.2.2.2 Population class

Population class responsible for storing Individuals.

Table 4.4: Population class attributes

Attribute	Type
indivs	vector<Individual>

Table 4.5: Population class operations

Operation	Return type	Parameters	Other details
getPopulationSize	int	void	return the size of the population
getIndividual	Individual	int	return the individual at a certain given index
setIndividual	void	int Individual	assign an individual to the given index

4.2.2.3 Individual Class

Individual class in LGP is a program in a form of instruction sequence. Therefore, Indivuual stores Instructions called genome. Operations of this class are mostly instructions wise. Every individual has its own fitness value.

Table 4.6: Individual class attributes

Attribute	Type
genome	vector<Instruction>
fitness	Fitness
maxlength	int

Table 4.7: Individual class operations

Operation	**Return type**	**Parameters**	**Other details**
getLength	int	void	return the length of genome
getGenomeSequence	vector<Instruction>	int int	return a sub sequence of genome in a limited interval
getFitness	Fitness	void	return fitness value of the individual
getMaxLength	int	void	return maximum possible length of individual
setInstruction	void	int Instruction	assign the new instruction at the given index
setFitness	void	Fitness	assign fitness to this individual
setMaxLenhth	void	int	set maximum possible length of individual
insert	void	int Individual	insert a new individual into this individual
insert	void	int Instruction	insert a new instruction into this individual
remove	void	int	remove an instruction from this individual at given index
remove	void	int int	remove sub sequnce of instructions from this individual at a specified length

4.2.2.4 Instruction Class

An Instruction consists of 4 parts namely, operation code, output register, input register 1 and input register 2.

Table 4.8: Instruction class attributes

Attribute	Type
opcode	int
outReg	int
inReg1	int
inReg2	int

Table 4.9: Instruction class operations

Operation	Return type	Parameters	Other details
getOpcode	int	void	return operation code
getOutReg	int	void	return output register number
getInReg1	int	void	return input register number 1
getInReg2	int	void	return input register number 2
setOpcode	void	int	assign a new opcode
setOutReg	void	int	assign a new output registe
setInReg1	void	int	assign a new input register number 1
setInReg2	void	int	assign a new input register number 2
set	void	int int int int	assige a new set of opcode, outReg, inReg1 and inReg2

4.2.2.5 Evaluator Class

Evaluator can executes every indivudal as well as calculate and assign its fitness value (represented by Fitness class).

Table 4.10: Evaluator class attributes

Attribute	**Type**
regSetImg	RegisterSetImg
regSetNum	RegisterSetNum
fnSet	FunctionSet
DataFetcher	fetcher

Table 4.11: Evaluator class operations

Operation	**Return type**	**Parameters**	**Other details**
evaluate	Fitness	Individual Individual int int	evaluate individuals
executeSubIndiv	cv::Mat	Individual vector<cv::Mat> cv::Mat int	execute the instructions in each individual
normalize	void	cv::Mat	nomalize the data in the given matrix

4.2.2.6 Fitness Class

Fitness class represents fitness value of an individual in evolutionary algorithm.

Table 4.12: Fitness class attributes

Attribute	Type
fit	double
higherIsBetter	bool

Table 4.13: Fitness class operations

Operation	Return type	Parameters	Other details
getFitness	double	void	return fitness value
setFitness	void	double	set new fitness value
setHigherIsBetter	void	bool	set if higher value is prefered or otherwise
reset	void	void	reset fitness value
isBetterThan	bool	Fitness	compare if current fitness is better than the given one or not
isWorseThan	bool	Fitness	compare if current fitness is worse than the given one or not
isEqaulTo	bool	Fitness	compare if current fitness is equal to than the given one or not

4.2.2.7 Breeder Class

The primary purpose of Breeder class is to perform selection and genetic operations which are crossover and mutaion. Selection method as well as the mentioned genetic operations have their own classes (Selection, Crossover and Mutator class).

Table 4.14: Breeder class attributes

Attribute	**Type**
randomGenerator	RandomGenerator
crossoverRate	double
mutationRate	double
selector	Selection
mutator	Mutation
crossover	Crossover
isElitist	bool

Table 4.15: Breeder class operations

Operation	**Return type**	**Parameters**	**Other details**
breed	void	Population Population Population Population Population	breed a new populations given the previous generation

Chapter 5

Experiments

This chapter explains how the experiments are carried out after the implementation was completed. This chapter also discusses the results of the experiments and conclusions. There are two kind of experiments which are leaf segmentation experiments and The CCLGP evolutionary run experiments.

5.1 Leaf Segmentation Experiment

5.1.1 Objective

This experiment is conducted because of two purposes. First, the performance of the image segmentation process needs to be measured. Second, to evaluate whether the quality of the image input to this process affects the result or not.

5.1.2 Experiment Setup

There are three set of photos. The first set contains 30 photos captured from an Android smartphone. The smartphone used for capturing the photo is the Asus ZenFone5 with 8 mega-pixel sensor. The photos were captured using the Android's camera application with the highest quality image setting and all other parameters are set to automatic. The second set of photos is from an iPhone 6. There are a total of 30 photos in the second set captured using the iOS's camera application. The last set of photos, which also contains 30 photos, is captured using a DSRL camera. The DSRL camera used is the Canon EOS 550D with 18–55 mm. and F/3.5–5.6 EF-S image stabilized lens. The settings of the camera are shown in Table 5.1. Other parameters that are now shown in the table are set to automatic.

Table 5.1: Camera settings

Format	PNG (converted from RAW (.CR2))
Resolution	5184 × 3456 pixels (width × height)
Shutter speed	1/50
F number	8.0
ISO	400
White balance	Automatic
Light optimization	Strong
Metering mode	Evaluative metering
Focal length	35 millimeters
Image stabilizer	On

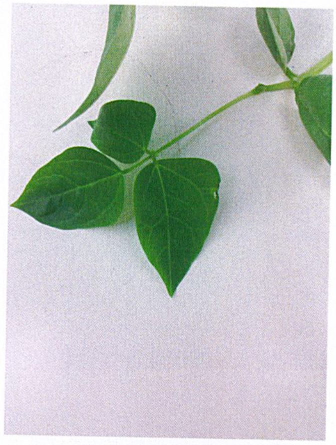
All photos from all sets were taken in a controlled environments, which include a controlled-illumination and a controlled-background.

Before starting the GrabCut segmentation, each photo is resized to less than 1,000 pixels in both width and height. Therefore, photos from the Asus ZenFone5, which was originally 2448 × 3264, were resized to 612 × 816 (width × height), photos from the iPhone 6, which was originally 2448 × 3264, were resized to 612 × 816 (width × height), and photos from the DSLR, which was originally 3456 × 5184, were resized to

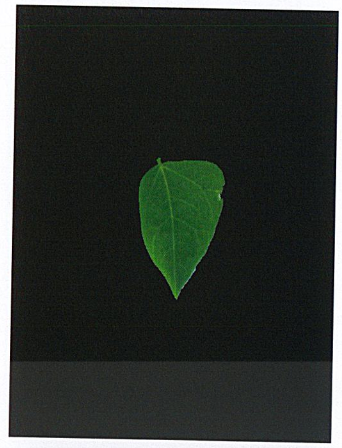
432×648 (width $\times$ height). This process was done because the computational power on mobile devices are limited. Therefore, reducing the size of an image significantly improved the time used to segment each image. For instance, the smartphone used to test the segmentation was the Asus ZenFone5. The segmentation process takes over a minute to complete using an original image. However, after the image was resized, the segmentation process only takes approximately 10 seconds, depending on the size of the area of interest.

5.1.3 Experimental Results and Discussions

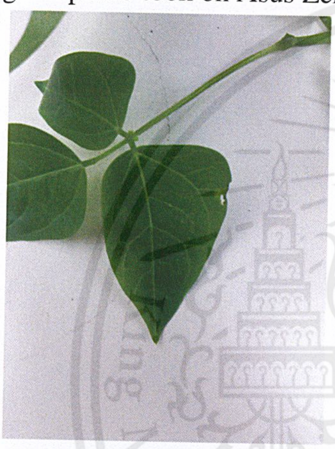
Figures 5.1 to 5.8 show the result of the segmentation process. Subfigures in each figure show original photo took on Asus ZenFone5, segmented image of the photo from the Asus ZenFone5, original photo took on iPhone 6, segmented image of the photo from the iPhone 6, original photo took on DSLR camera, and segmented image of the photo from the DSLR camera, respectively from (a) to (f).



(a) Original photo took on Asus ZenFone5

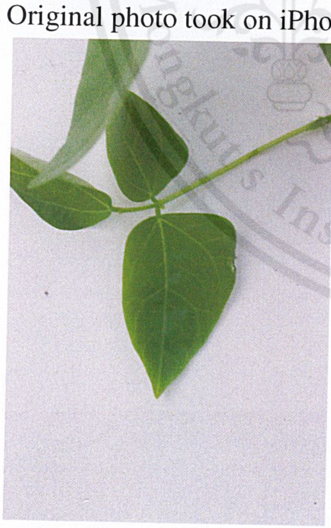


(b) Segment result from Fig. 5.1a



(c) Original photo took on iPhone 6

(d) Segment result from Fig. 5.1c

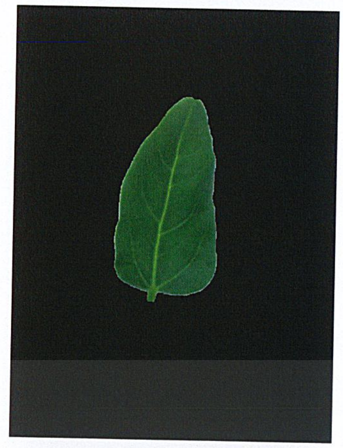


(e) Original photo took on Canon 550D DSLR

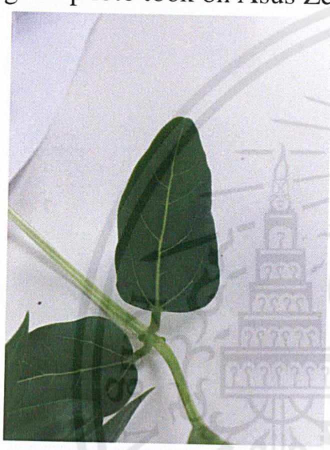
(f) Segment result from Fig. 5.1e

Figure 5.1: Test image 1

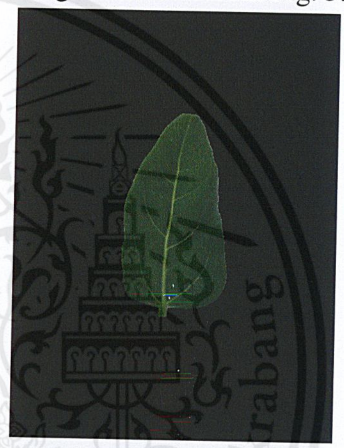
(a) Original photo took on Asus ZenFone5



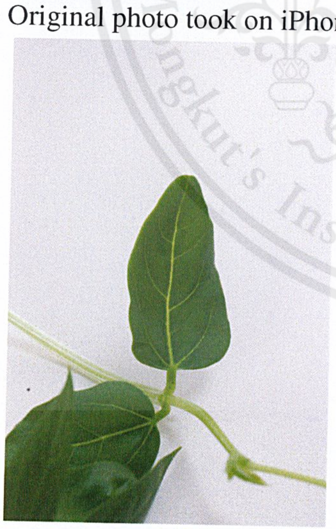
(b) Segment result from Fig. 5.2a



(c) Original photo took on iPhone 6



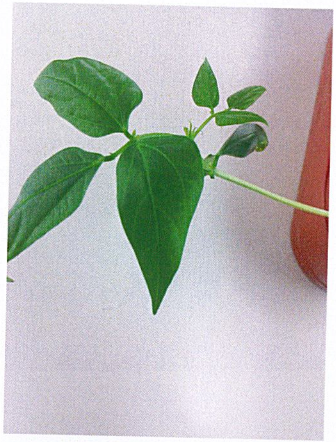
(d) Segment result from Fig. 5.2c



(e) Original photo took on Canon 550D DSLR

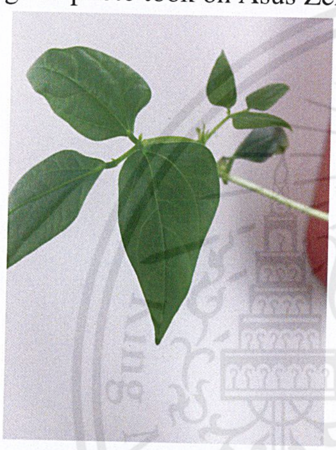
(f) Segment result from Fig. 5.2e

Figure 5.2: Test image 2



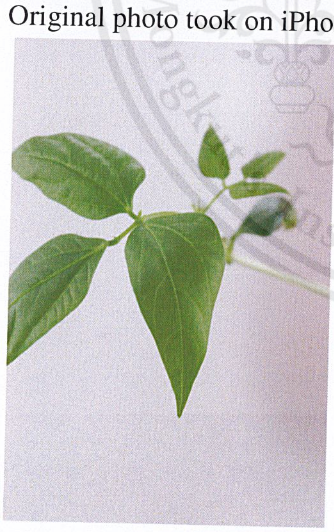
(a) Original photo took on Asus ZenFone5

(b) Segment result from Fig. 5.3a



(c) Original photo took on iPhone 6

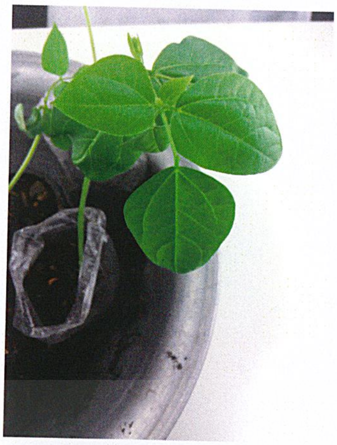
(d) Segment result from Fig. 5.3c



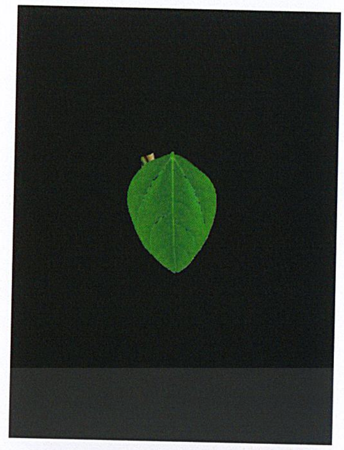
(e) Original photo took on Canon 550D DSLR

(f) Segment result from Fig. 5.3e

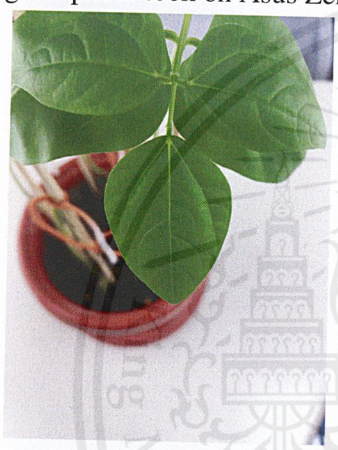
Figure 5.3: Test image 3



(a) Original photo took on Asus ZenFone5

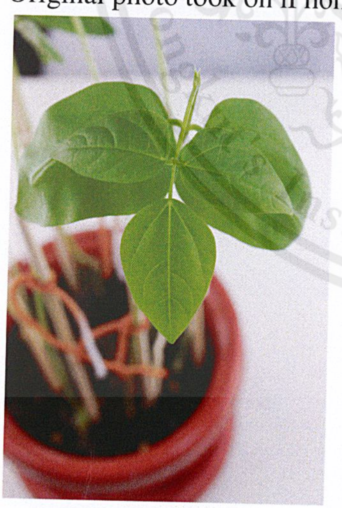


(b) Segment result from Fig. 5.4a



(c) Original photo took on iPhone 6

(d) Segment result from Fig. 5.4c

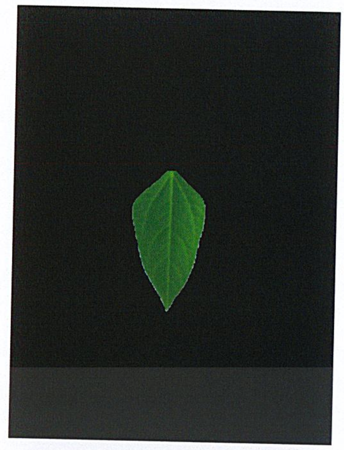


(e) Original photo took on Canon 550D DSLR

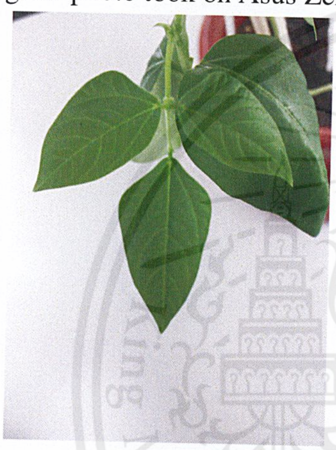
(f) Segment result from Fig. 5.4e

Figure 5.4: Test image 4

(a) Original photo took on Asus ZenFone5

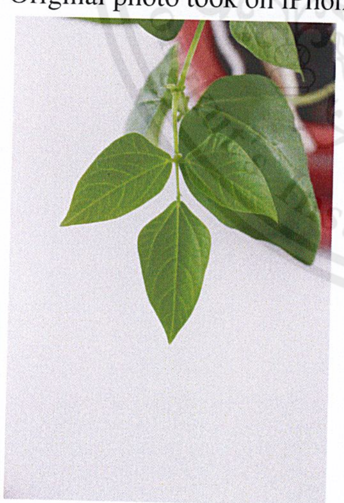


(b) Segment result from Fig. 5.5a



(c) Original photo took on iPhone 6

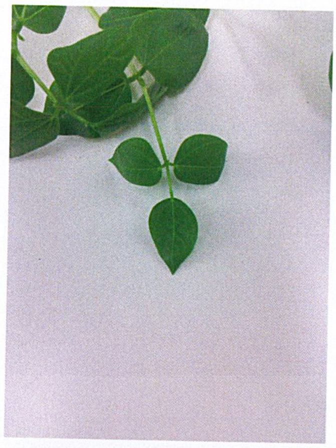
(d) Segment result from Fig. 5.5c



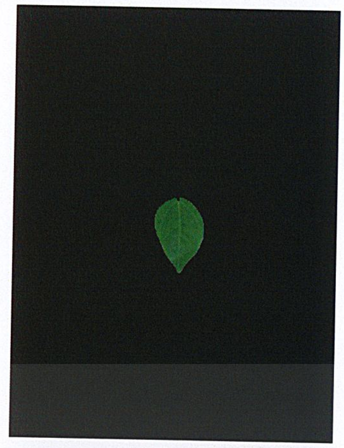
(e) Original photo took on Canon 550D DSLR

(f) Segment result from Fig. 5.5e

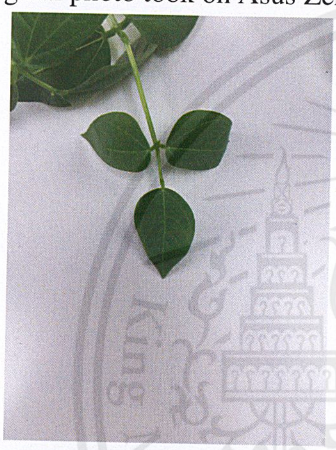
Figure 5.5: Test image 5



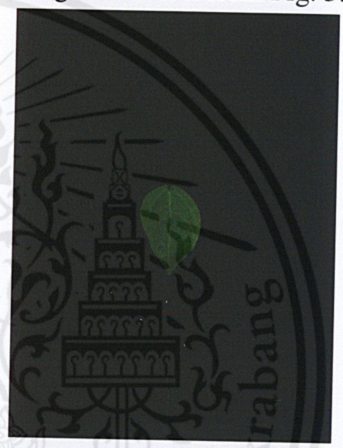
(a) Original photo took on Asus ZenFone5



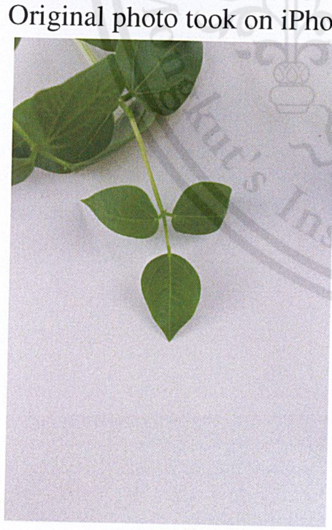
(b) Segment result from Fig. 5.6a



(c) Original photo took on iPhone 6



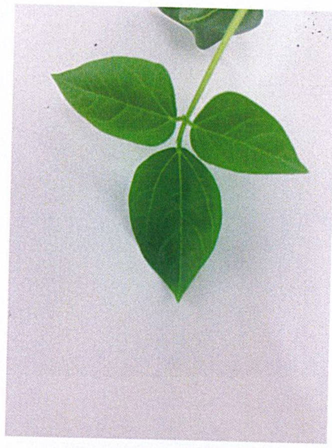
(d) Segment result from Fig. 5.6c



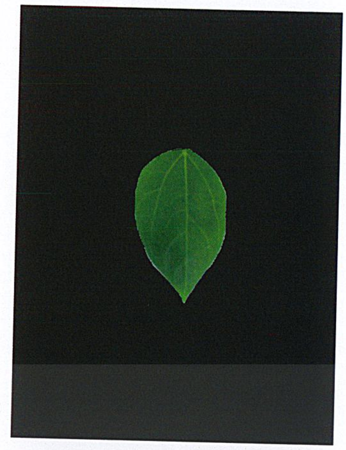
(e) Original photo took on Canon 550D DSLR

(f) Segment result from Fig. 5.6e

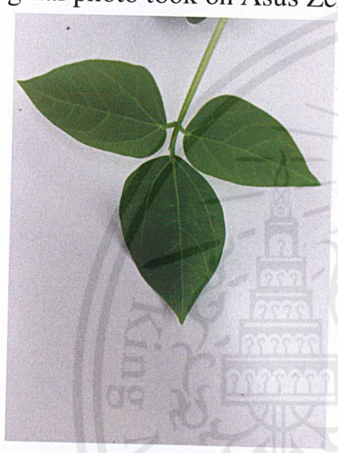
Figure 5.6: Test image 6



(a) Original photo took on Asus ZenFone5

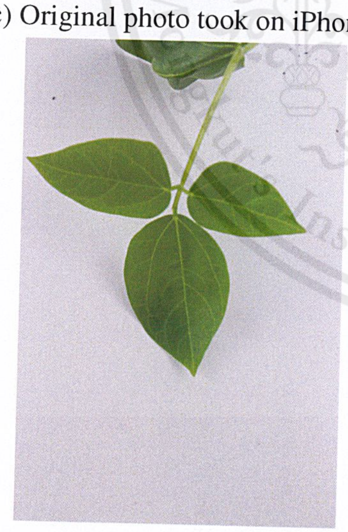


(b) Segment result from Fig. 5.7a



(c) Original photo took on iPhone 6

(d) Segment result from Fig. 5.7c

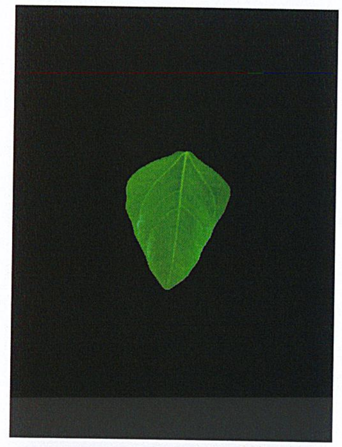


(e) Original photo took on Canon 550D DSLR

(f) Segment result from Fig. 5.7e

Figure 5.7: Test image 7

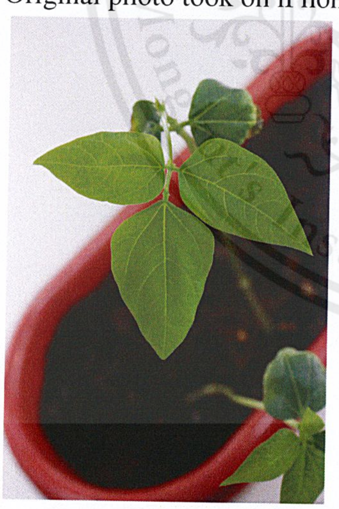
(a) Original photo took on Asus ZenFone5



(b) Segment result from Fig. 5.8a

(c) Original photo took on iPhone 6

(d) Segment result from Fig. 5.8c



(e) Original photo took on Canon 550D DSLR

(f) Segment result from Fig. 5.8e

Figure 5.8: Test image 8

Table 5.2: Segmentation result

Source	Number of Images	Number of Correctly Segmented Images	Percentage
Asus ZenFone5	30	12	40.00%
iPhone 6	30	20	66.67%
Canon 550D DSLR	30	26	88.67%
Total	90	58	64.44%

In the first set, images taken using the Asus ZenFone5, the segmented images gave satisfiable when the leaf in the image lied flat on a plain background, as in Fig. 5.1, Fig. 5.2, Fig. 5.5, Fig. 5.6, Fig. 5.7, and Fig. 5.8. In other situations, the segmented images from the Asus ZenFone5 gave undesirable results.

In the second set, images taken using the iPhone 6, more images providing preferable segmentation result than the Asus ZenFone5, as in Fig. 5.1, Fig. 5.2, Fig. 5.3, Fig. 5.5, Fig. 5.6, and Fig. 5.8. Only in some difficult circumstances that the images taken with the iPhone 6 deliver unfavorable results.

For the last set, images taken using the DSLR camera, almost every images yielded favorable segmented result, as in Figs. 5.1 to 5.8.

All in all, image quality significantly affects the segmentation results. As can be seen from Figs. 5.1 to 5.8, the Asus ZenFone5 gives the lowest quality among the others. For example, the color of the leaf edge in the image taken with the Asus ZenFone5 is diffused to the background, which does not happened with the images from the iPhone 6 and the DSLR camera. The best quality images came from the DSLR camera which give accurate color. The DSLR camera can also blurred out the background, which make the leaf in the image stood out. To sum up, the segmentation algorithm significantly depend on the image quality to give a delighted result.

5.2 Nutrient Analysis Experiment

5.2.1 Objective

The purpose of this experiment is using CCLGP to construct two feature extraction programs which responsible for extracting features of young leaf and old leaf images. The experiments were conducted with different parameter settings and quantity of training data used.

5.2.2 Experiment Setup

To obtain the black gram's leaf images as training and testing dataset. A set of black gram plants were raised and then capture photos of their leaves for 27 days. Therefore, 72 black gram specimens were obtained in which provided by Dr. Sutsawat Duansrisai, an expert botanist and a professor from Kasetsart University Bangkaen, Thailand. The specimens obtained are the ones that were already reaised for two weeks the heights of the plants were approximately 30 centimeters. Then applied the stress into the plants with treatment solutions.

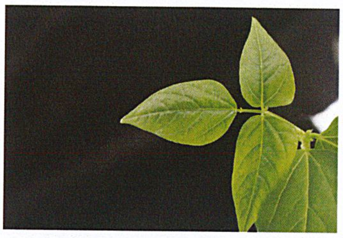
The plants will be divided into 6 catagories with 12 plants each which corresponded to deficiency types and were raised in a plastic bottle containing chemical solution according to their deficiency type as well (Fig. 5.9) .For example, plants that will be classified as Calcium deficiency will be raised with a solution which lacks Calcium only and so on. Therefore, there were 6 types of solution according to all the deficiency types. The treatment solutions were prepared by staffs from Kasetsart universitiy Bankaen. The solution will be changed weekly to prevent the plants from lack of other nutrients.

Figure 5.9: Six different trays are also prepared for each type of deficiency of the plant. Each tray can stored up to 12 plants.

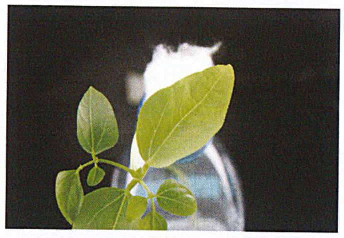
One young leaf and one old leaf photos will be taken from each plant (Camera specifications describrd in Table 5.3) under the condition that the leaf must be positioned in the middle of the photo.

Table 5.3: Camera sepcifications

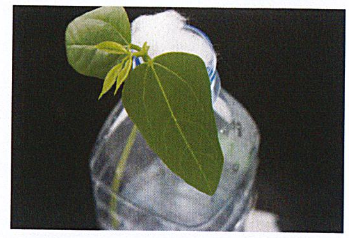
Name	Canon EOS 550D
Lens	18-55mm f/3.5-5.6 EF-S image stabilized lens
sensor	18.7 million pixels sensor
resolution	5184 $\times$ 3456 pixels (width $\times$ height)
dpi	72 pixels/inch



(a) Young leaf from Calcium deficiency catagory



(b) Old leaf from complete catagory



(c) Old leaf from Iron deficiency catagory

Figure 5.10: Sample of leaf photos

One by one, each plant photos will be taken in a controlled environment with controlled illumination condition (Fig. 5.11).

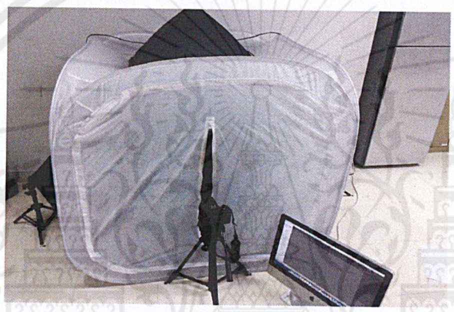


Figure 5.11: Controlled illumination environment

Photos of each plant will be taken daily started from December 1st, 2015 until December 28th, 2015 (December 20th, 2015 is omitted because of technical problem occurance). However, the number of captured images will not be fixed because of some uncontrollable factors. For example, at that period, many young leaves cannot be taken because it hadn't grown enough to be recognizable or the later day, most of the plants with nutrient deficiency symptoms were withered away or dead, especially with Calcium deficiency ones (Table 5.4).

Table 5.4: The records of the collected leaf photos in which the values in the table represents the number of leaf photos taken

Day	COM		-Ca		-Mg		-Fe		-N		-K	
	O	Y	O	Y	O	Y	O	Y	O	Y	O	Y
1	12	10	12	11	12	8	12	8	12	10	12	7
2	12	11	12	11	12	8	12	8	12	10	12	7
3	12	12	12	11	12	10	12	9	12	10	12	8
4	12	12	12	11	12	10	12	9	12	10	12	8
5	12	12	12	12	12	11	12	9	12	12	12	9
6	12	12	12	12	12	11	12	9	12	12	12	9
7	12	12	12	12	12	11	12	11	12	12	12	11
8	12	12	12	12	12	11	12	12	12	12	12	11
9	12	12	12	12	12	11	12	12	12	12	12	11
10	12	12	11	11	11	11	11	12	12	12	12	11
11	12	12	11	11	11	11	11	12	12	12	12	11
12	12	12	11	11	11	11	11	12	12	12	12	11
13	12	12	9	9	11	11	11	12	12	12	12	11
14	12	12	9	9	12	12	11	12	12	12	12	11
15	12	12	9	9	12	12	11	12	11	12	12	11
16	12	12	9	9	10	12	10	12	11	12	12	11
17	12	12	9	9	10	12	10	12	11	12	11	12
18	12	12	9	9	10	12	10	12	11	12	11	12
19	12	12	9	9	10	12	9	12	11	12	10	12
20	-	-	-	-	-	-	-	-	-	-	-	-
21	11	12	9	9	10	12	9	12	5	12	10	12
22	11	12	2	4	12	11	10	11	11	10	12	12

continued ...

...continued

Day	COM		-Ca		-Mg		-Fe		-N		-K	
	O	Y	O	Y	O	Y	O	Y	O	Y	O	Y
23	11	12	2	4	12	11	11	12	11	10	12	12
24	11	12	2	4	12	11	11	12	11	10	12	12
25	9	12	2	4	10	11	11	12	8	11	12	12
26	9	12	2	4	10	11	9	12	8	11	12	11
27	8	12	2	4	9	10	9	11	8	10	10	11
28	8	12	2	4	9	9	8	11	8	10	10	11
Total	306	321	225	237	300	293	291	289	293	304	314	287

Each photo were resized 4 times to 1296 × 864 pixels because it could reduce the time used for processing each data during the evolutionary run. The format are also changed from CR2 to .PNG format because CR2 format is not a universal format for images store in a computer, however PNG is a universal and a lossless one.

However, the images were resized again to 648 × 432 because the previous size still produced slow processing time in the experiments. This also results in information loss and therefore, guarantees the drop in accuracy.

After that, the photos were undergone segmentation. Current set of photos contained a background and some other objects. However, only the leaf in the middle of the photo was needed. Therefore, using photoshop, other part of the image aside from the leaf in the middle were manually removed and replaced with white pixels acted as its background.

(a) RS4_COM_D11_R10_P1.png (b) Seg_RS4_COM_D11_R10_P1.png

Figure 5.12: Sample image data before(a) and after(b) segmentation

Noticed from Fig. 5.12b that the name of the file was changed after the segmentation. Every images' file name are also changed into the same format: $Seg\_RS4\_(Deficiencyclass)\_(Day)\_(Plantnumber)\_(Leaftype).png$

- **Seg**: indication which shows that this image file had already been segmented.
- **RS4**: indication which shows that this image file had already been resized 4 times original resolution
- **Deficiency class**:
  - COM: complete (no deficiency)
  - Ca : Calcium deficiency
  - Mg : Magnesium deficiency
  - N : Nitrogen deficiency
  - K : Potassium deficiency
  - Fe : Iron deficiency
- **Day**: day number counted from the first day of photo taking
- **Plant number**: The number of plant in the tray (1 - 12)
- **Leaf type**: Old leaf or young leaf

5.2.3 Experimental Results and Discussion

The experiments were conducted for executing the evolutionary run. Three different parameter sets with eight evolutionary runs for each one were executed.

Table 5.5: The experiment results from three sets of evolutionary runs

Variable	Set 1	Set 2	Set 3
Avearage best fitness (%)	34.99	26.25	41.66
Maximum fitness (%)	43.33	30.00	53.33
Minimum fitness (%)	30.00	23.33	36.66
Standard deviation (%)	3.98	2.13	5.34
Parameters			
Generation limit	50	50	30
Population size	30	30	100
Maximum length	20	20	20
Minimum length	3	3	3
Numeric registers	8	8	8
Image registers	8	8	8
Maximum operation	33	33	33
Training data per class	18	18	30
Training data per class	30	30	30
Crossover rate	0.8	0.8	0.8
Mutation rate	0.02	0.02	0.2
Elitist	False	True	True
Tournament size	2	2	2

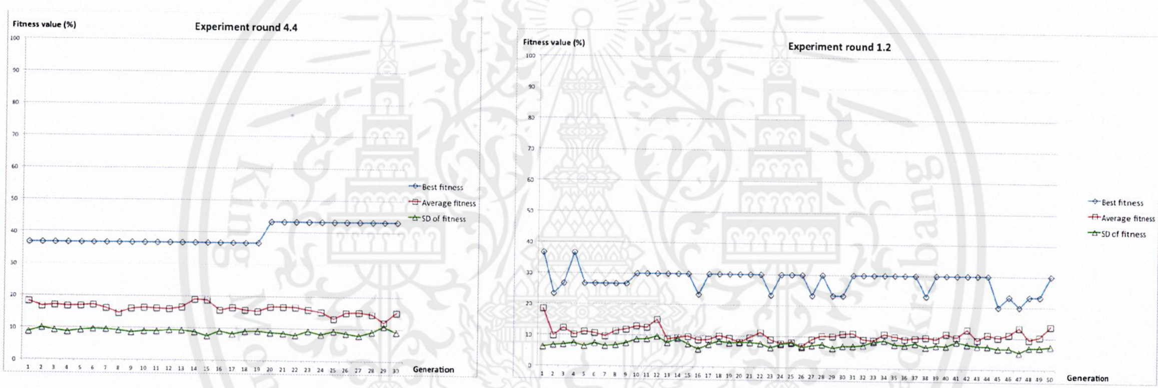
From Table 5.5, average best fitness is calculated from taking the best fitness value from that set (all eight evolutionary runs) and average them. Maximum fitness and Minimum fitness both means maximum and minimum fitness value found in the same set as well as Standard deviation which calculated from best fitness values together.

The first set results in the highest fitness value at 43.33% while the average was 34.99% in which unfortunately, was not yet half of the fitness range while the minimum fitness was a little bit better than randomizing.

The second set produced even less preferable results with maximum fitness value of

30% and the average is 26.25%. There was only one parameter that had changed which is the elitist parameter set from false to true. In this case, it might be a drawback from the elitist mechanism.

Elitist ensures the best individual would always be passed on to the next generation (example see figure 1.). Therefore an increase in exploitation of individual. However, it seemed in this case, that single individual might had gained influence over the population and prevented other individuals which might had produced high fitness value further on. Another possibility which might had prevented the growth of fitness value was the mutation rate with only 0.02 because compared to the maximum length which was 20, the rough probability calculation showed that 0.4 of instructions in an individual would be mutated in which the chance was next to not happening at all.



(a) The best individual fitness in each generations with elitist applied

(b) The best individual fitness in each generations without elitist

Figure 5.13: Fitness growth comparison with and without elitist

The last set, the experiment set up undergone quite a bit of modification. First, the training set were expanded from 18 to 30 in hope that it might help increase the average fitness produced. Another modification is the enhancement approach for pairing an individuals. As explained in Section 3.3.3 that a single pair of individuals selected from each population were to be evaluated together. However, in this round, each one individual would be paired with 3 other non-duplicate individuals from the other population for evaluation. Then, the pair which had maximum fitness value out of the other 2 combinations were selected instead. The last modifications were changing the generation limit from 50 to 30, mutation rate from 0.02 to 0.2.

The experiment results this time showed slight improvements. The maximum fitness value was 53.33 and the average with 41.66. However, even though there were positive improvements in the evolutionary run results, the fitness values were still far from effective.

5.2.3.1 Conclusion

From the experiments results, we can conclude that this approach still needs more improvement and study to be done. The challenge of this research is improving the fitness value of individuals. The problem is there were total of 5 classification data to operate on and there were also deficiency symptoms that possesses similar characteristics.

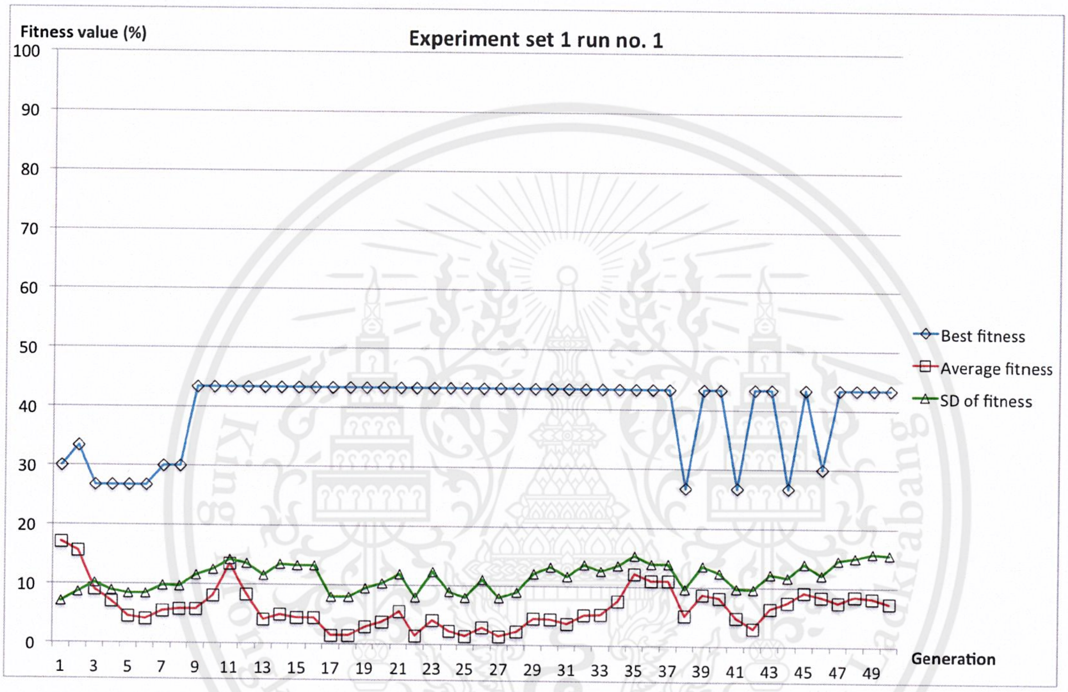
The image data used was in grayscale therefore, information. Mainly, color information. Removing this limitation means more image processing operation to used in which might also increase the diversity in individuals given that there were five classification data to work with.

However, the very first simple approach is to try changing and observing other parameters. The last experiments might also be able to improve by try removing the elitist mechanism, expand the training and testing dataset as well as population size and generation limit. There were also other fixed parameters for example, tournament size which currently is two. Despite the potential of the results, a lot of time for the evolutionary run are required as a tradeoff.

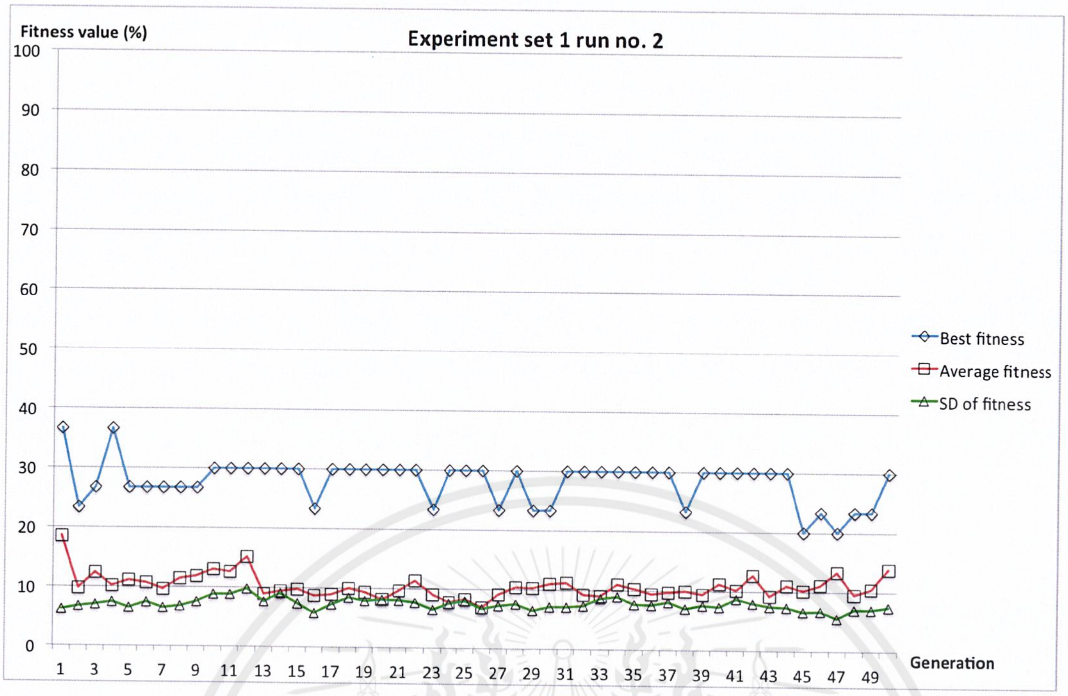
5.2.3.2 Evolutionary Run Statistics

This section displays all the statistical data derived from each evolutionary run which are the best fitness, average fitness and standard deviation of fitness. The statistical data of set 1 experiments are from Figures 5.14a to 5.14h, set 2 experiments are from Figures 5.14i to 5.14p and set 3 experiments are from Figures 5.14q to 5.14x

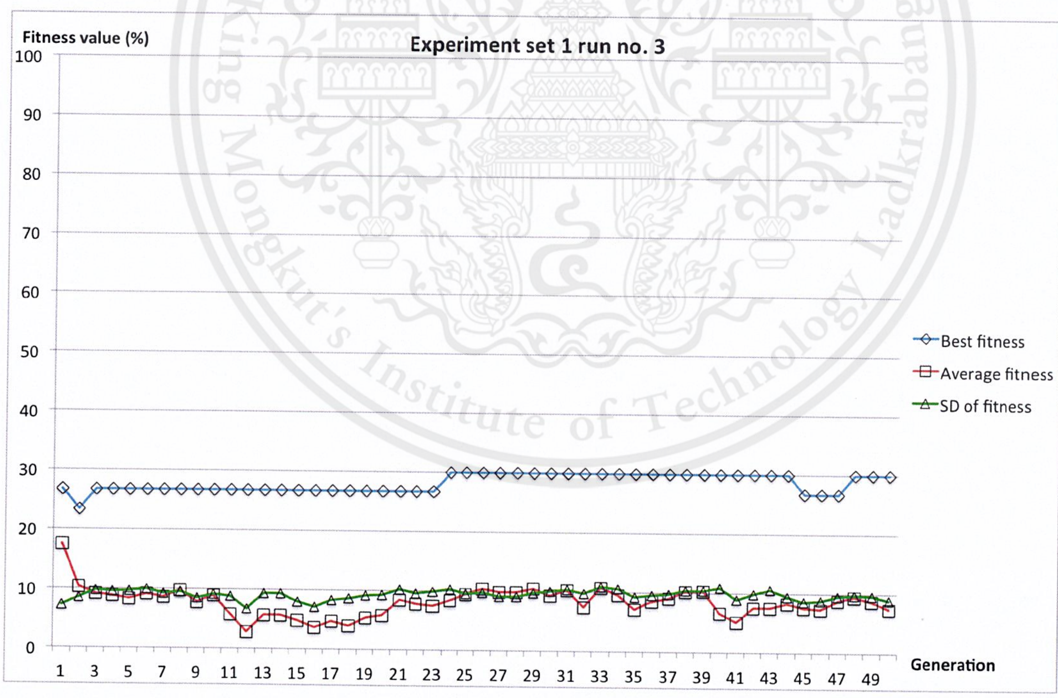
Figure 5.14: Statistical data from all three experiment rounds



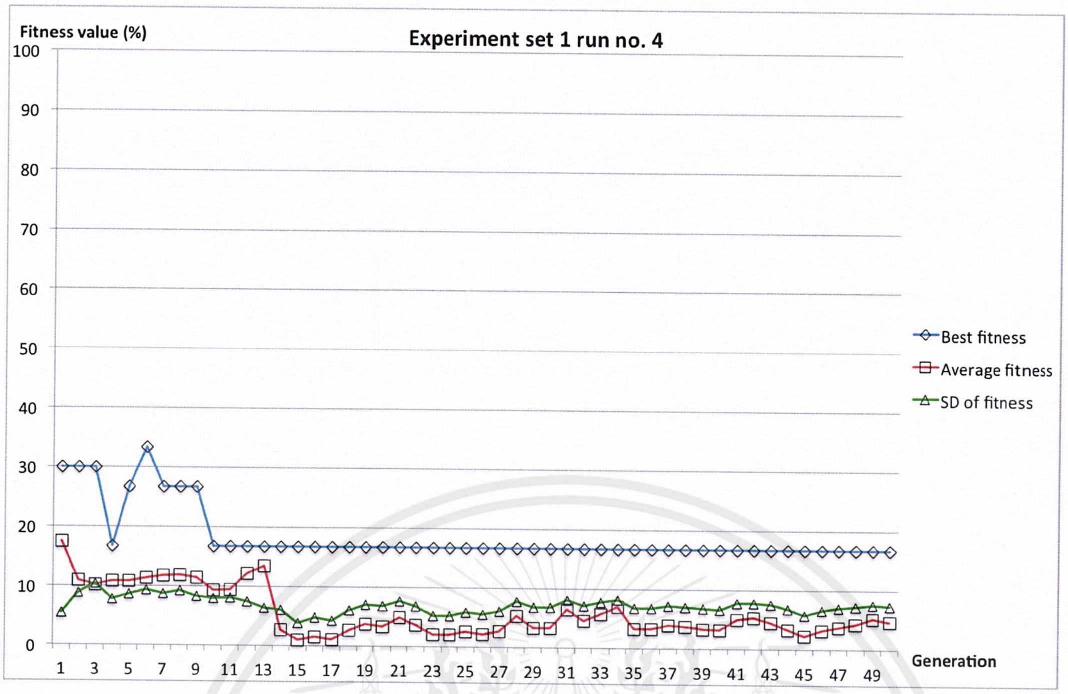
(a) Experiment set 1, Evolutionary run: 1



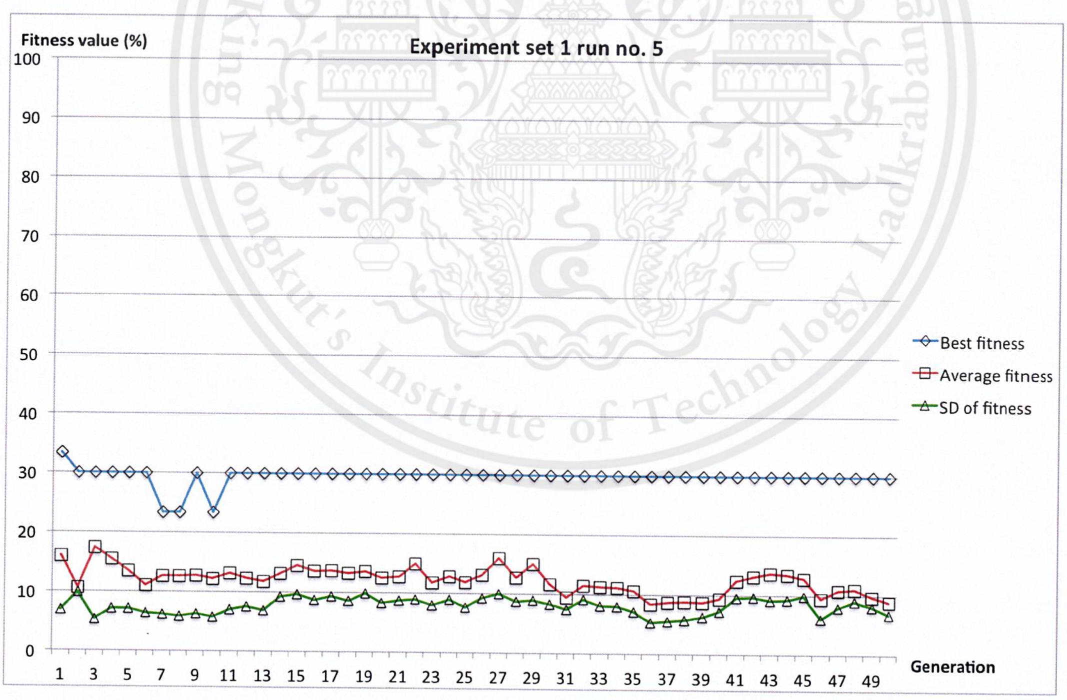
(b) Experiment set 1, Evolutionary run: 2



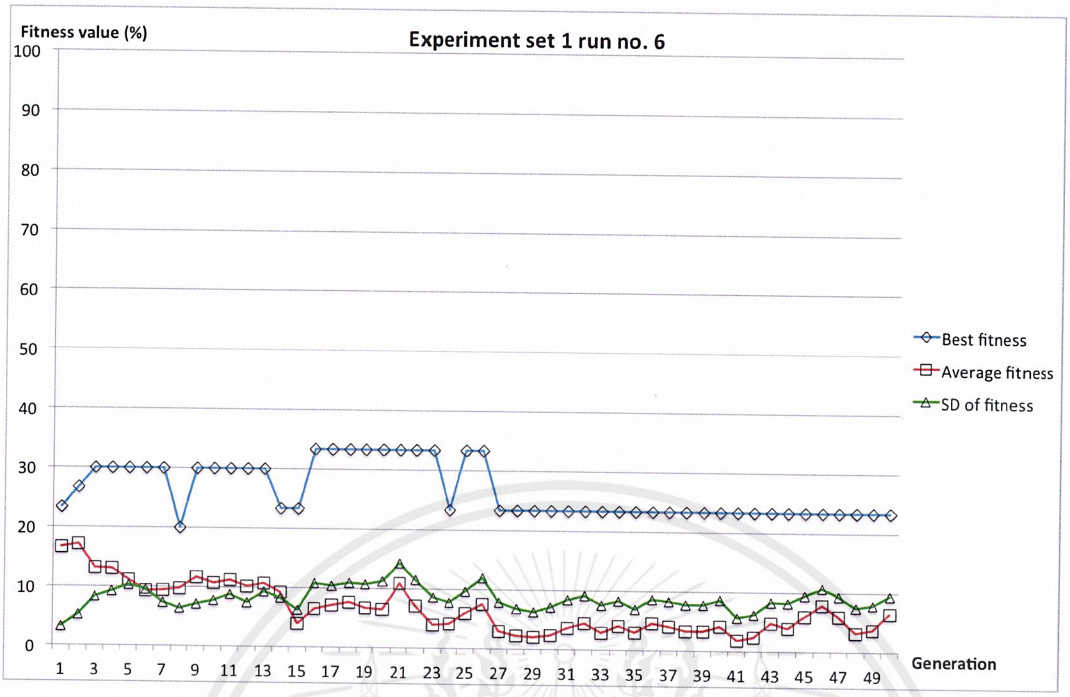
(c) Experiment set 1, Evolutionary run: 3



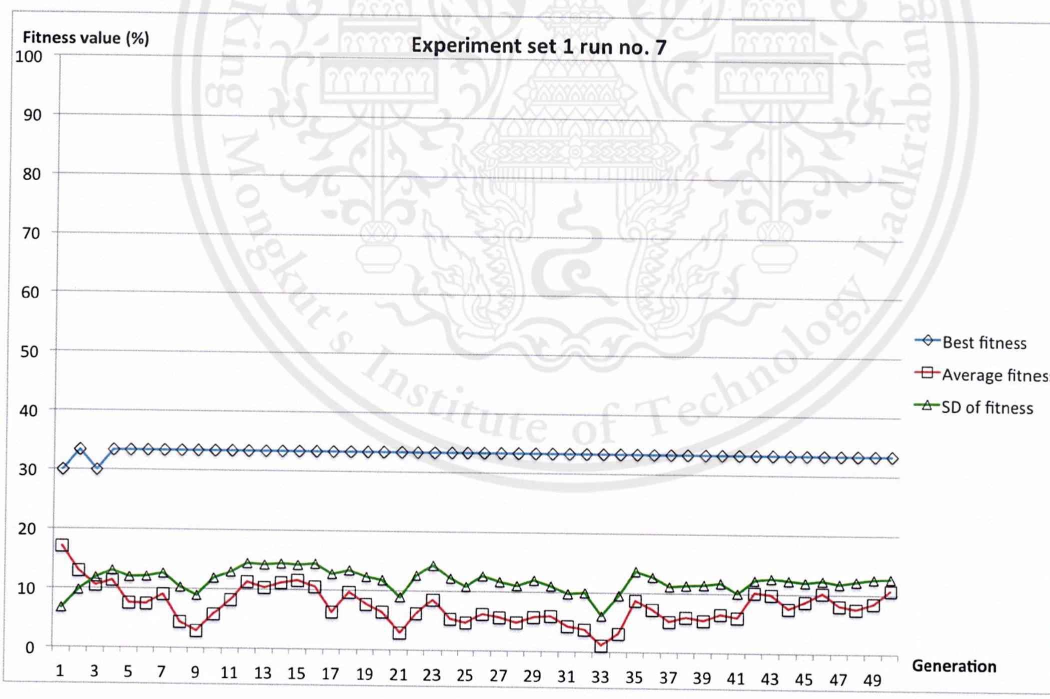
(d) Experiment set 1, Evolutionary run: 4



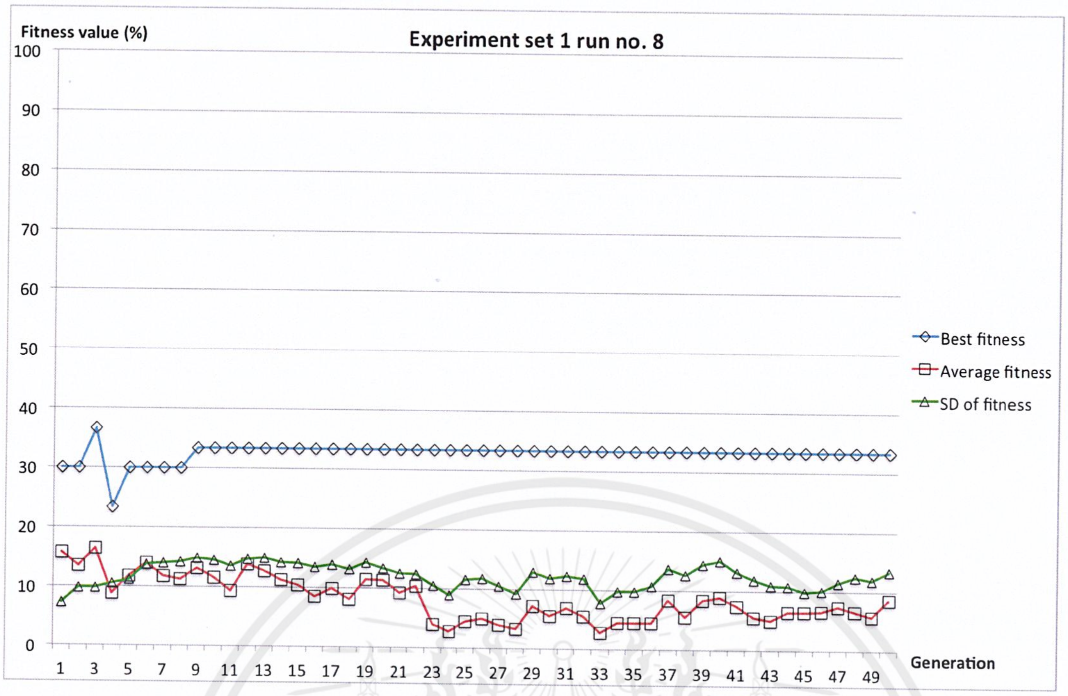
(e) Experiment set 1, Evolutionary run: 5



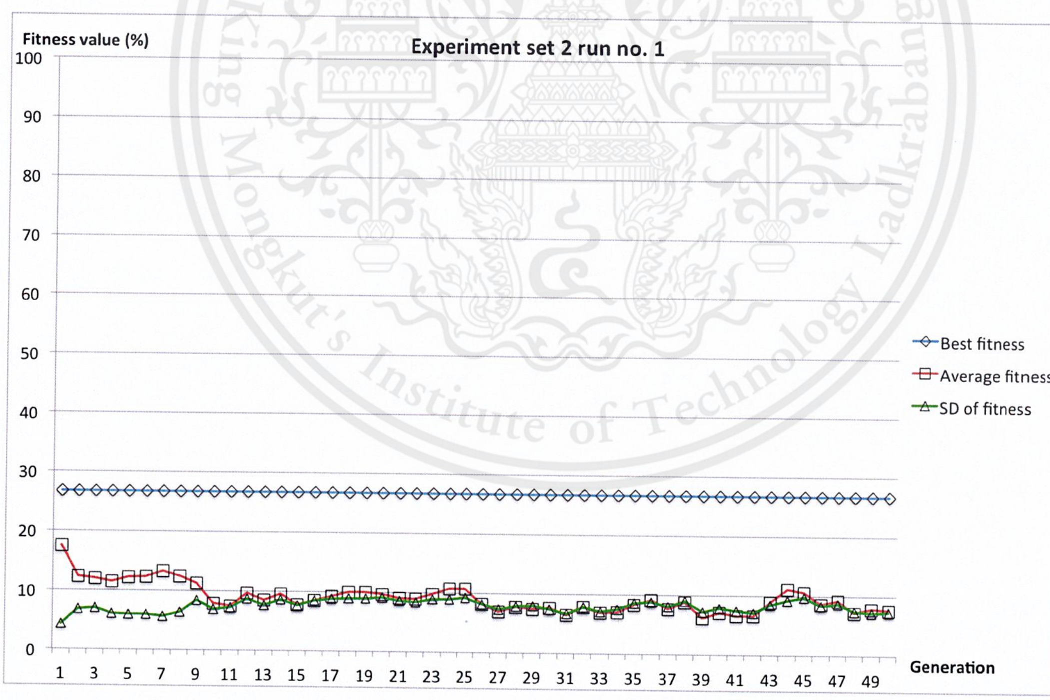
(f) Experiment set 1, Evolutionary run: 6



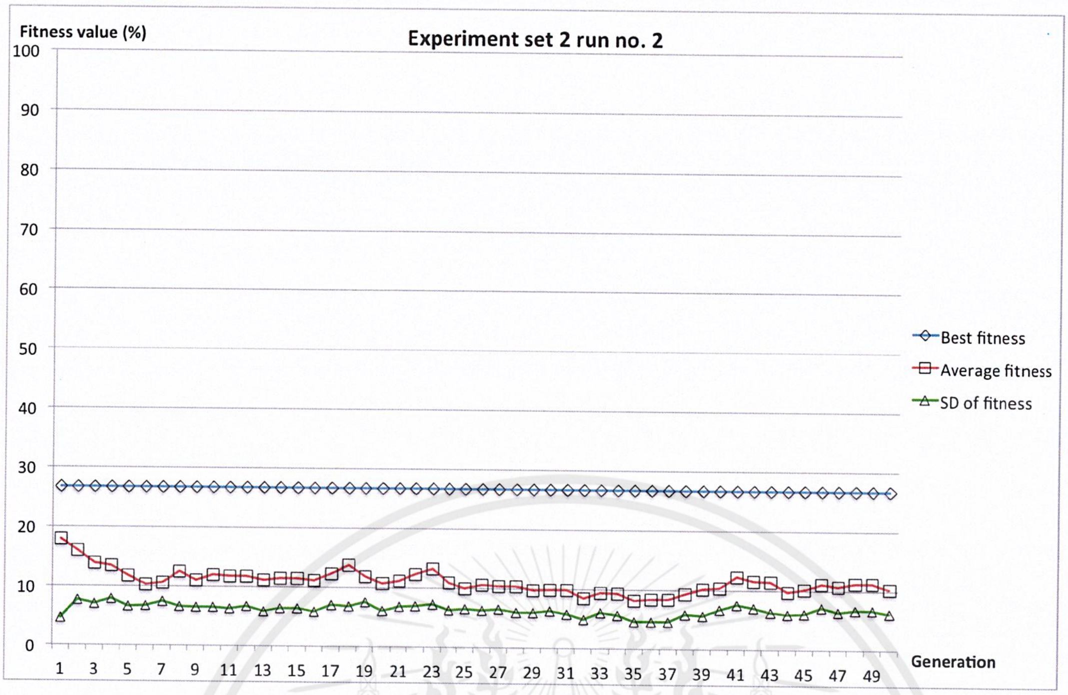
(g) Experiment set 1, Evolutionary run: 7



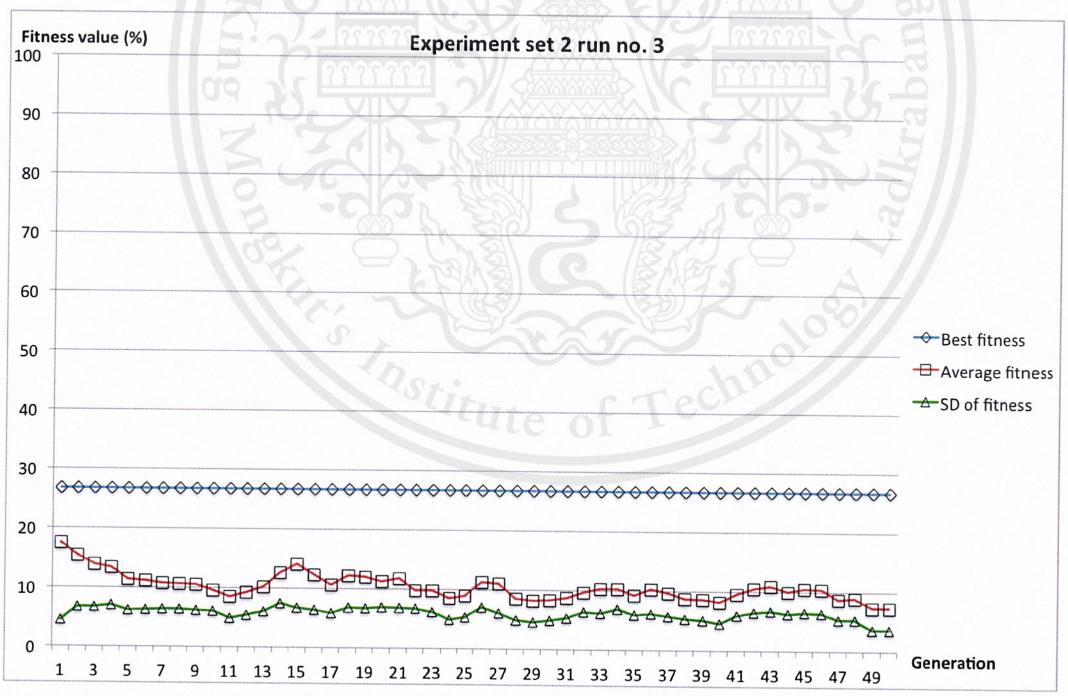
(h) Experiment set 1, Evolutionary run: 8



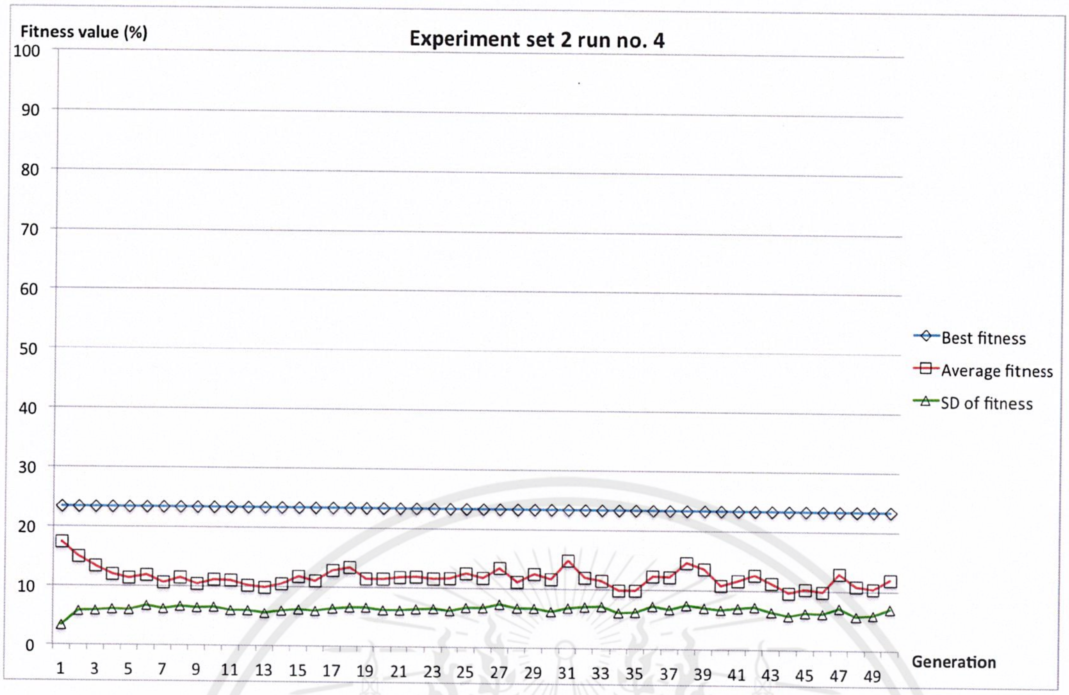
(i) Experiment set 2, Evolutionary run: 1



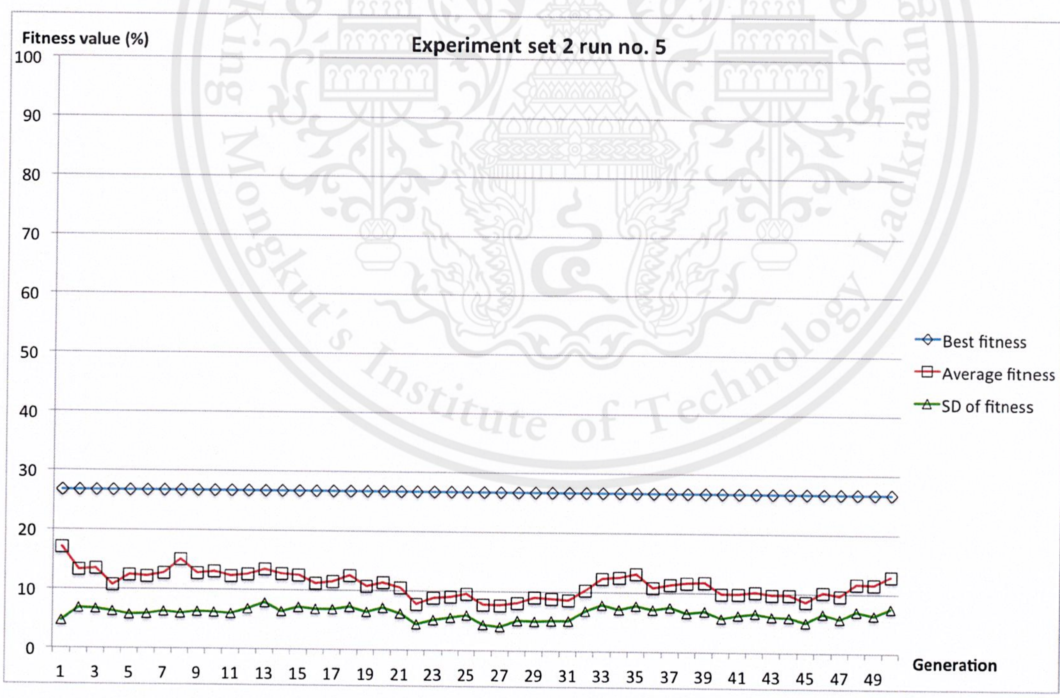
(j) Experiment set 2, Evolutionary run: 2



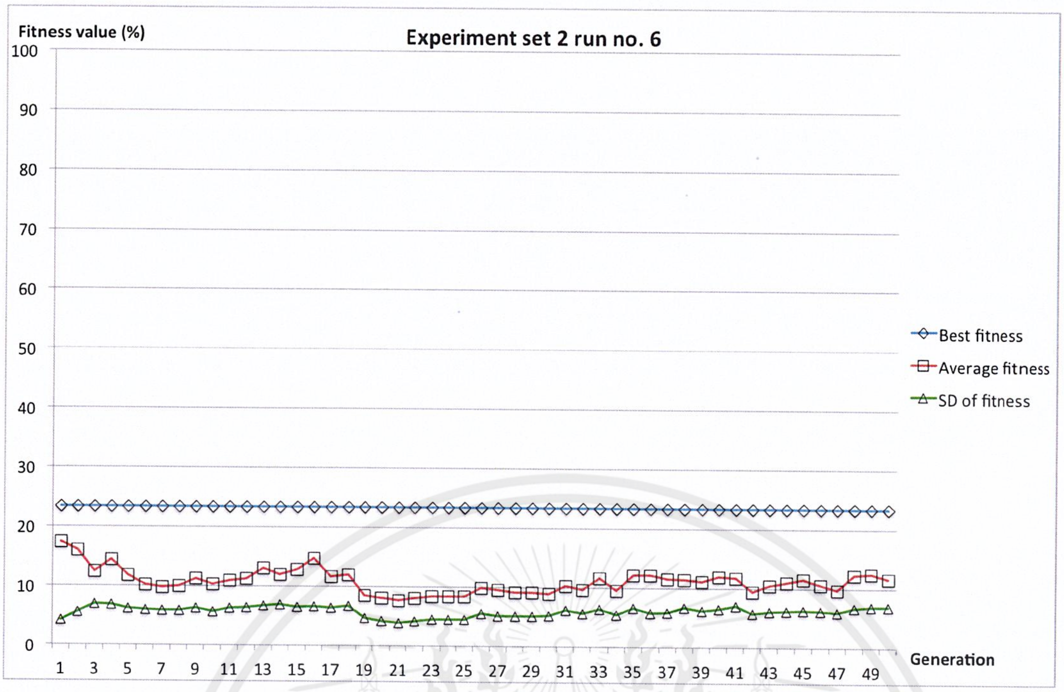
(k) Experiment set 2, Evolutionary run: 3



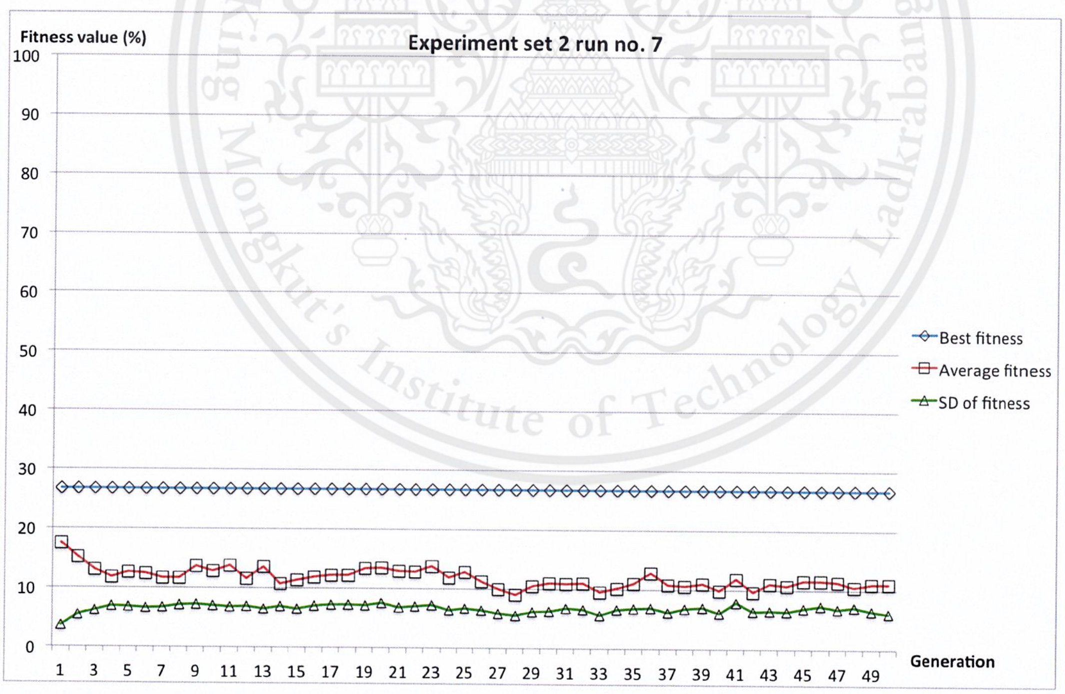
(l) Experiment set 2, Evolutionary run: 4



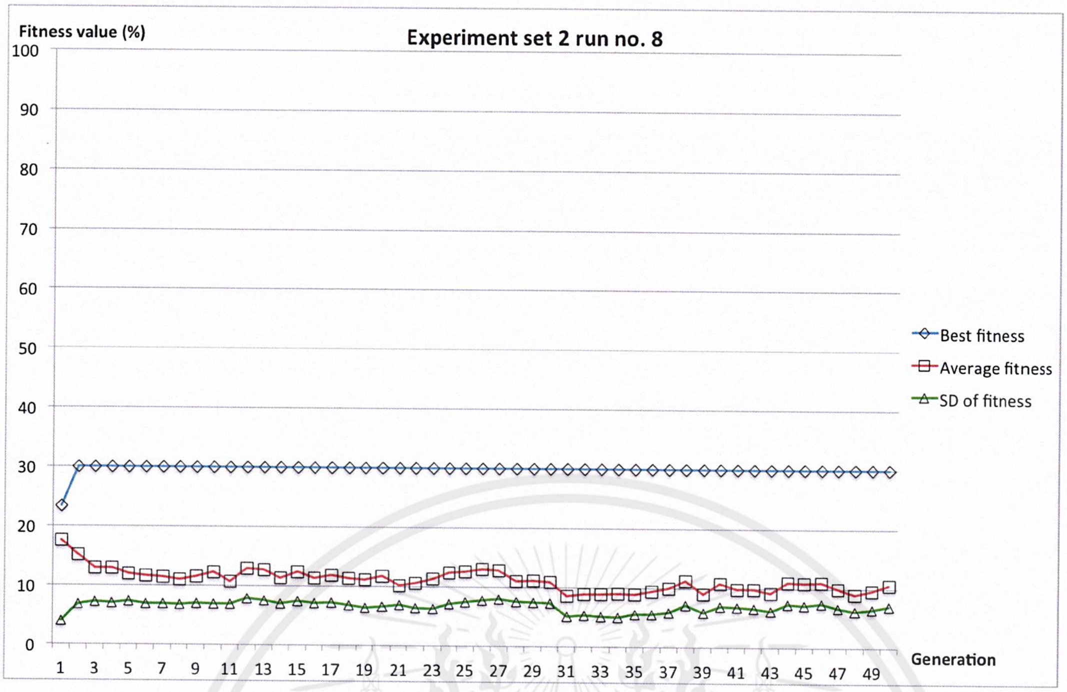
(m) Experiment set 2, Evolutionary run: 5



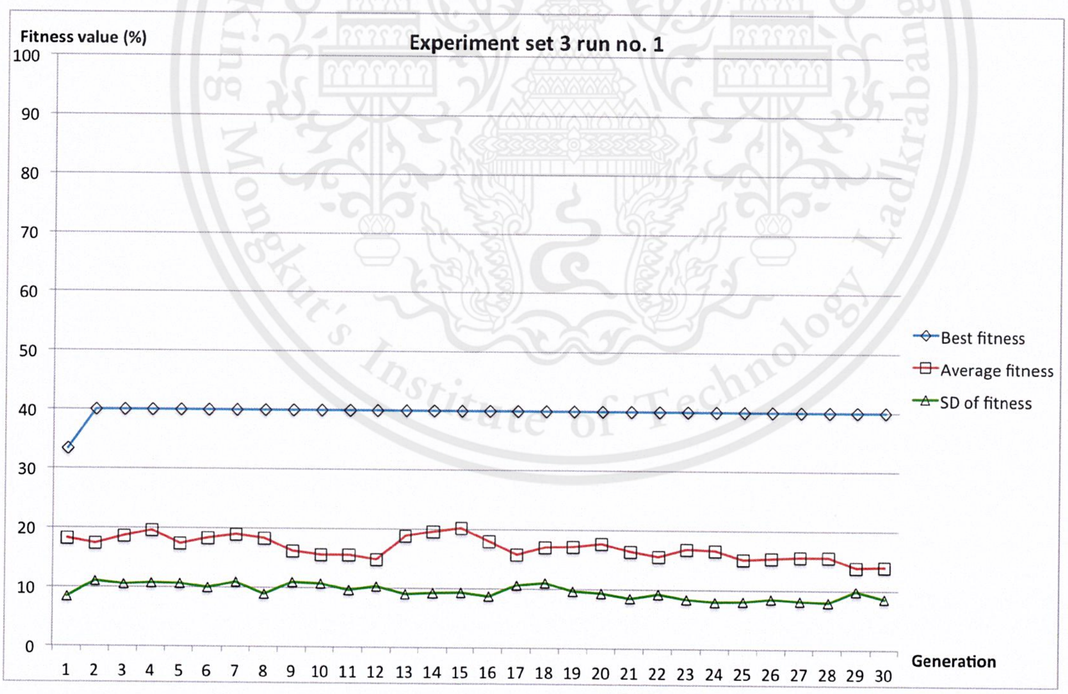
(n) Experiment set 2, Evolutionary run: 6



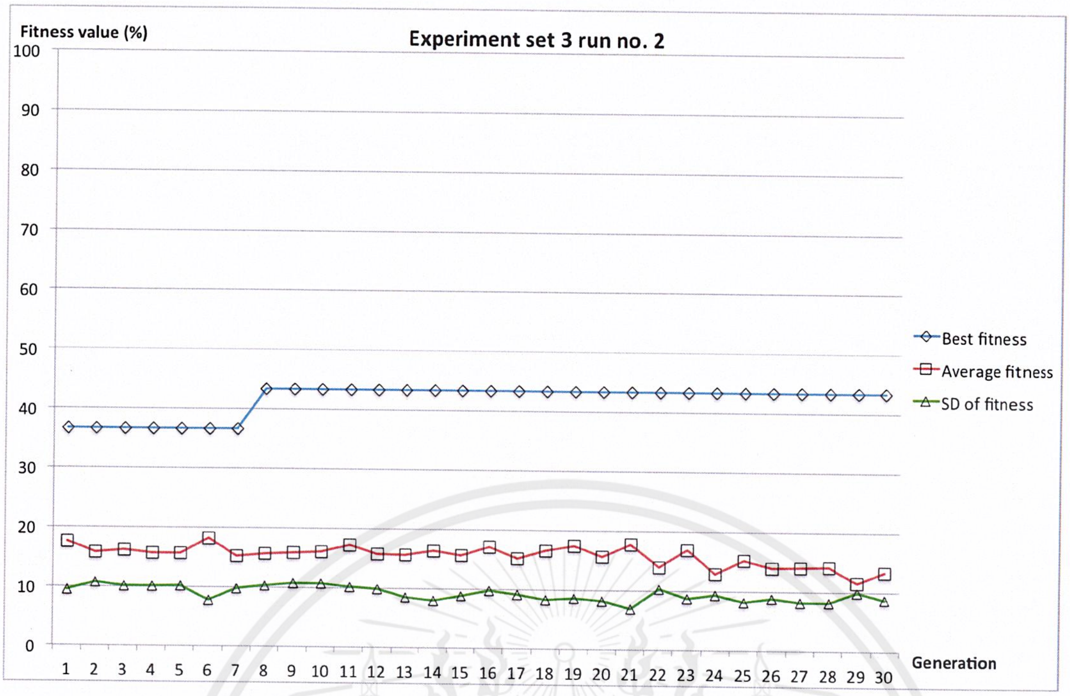
(o) Experiment set 2, Evolutionary run: 7



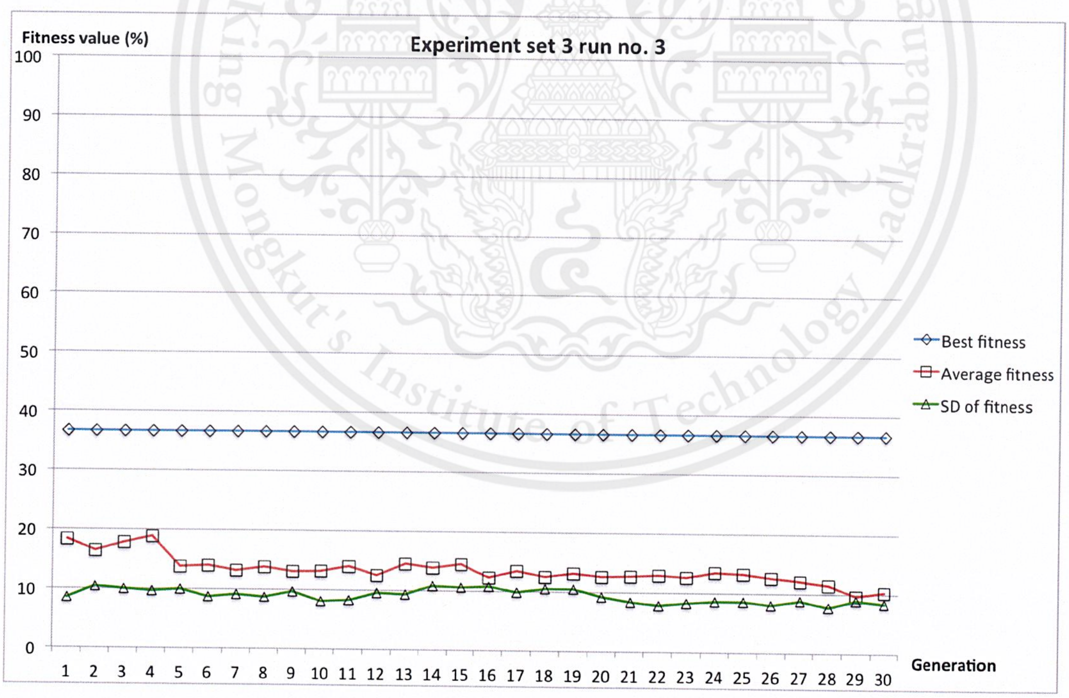
(p) Experiment set 2, Evolutionary run: 8



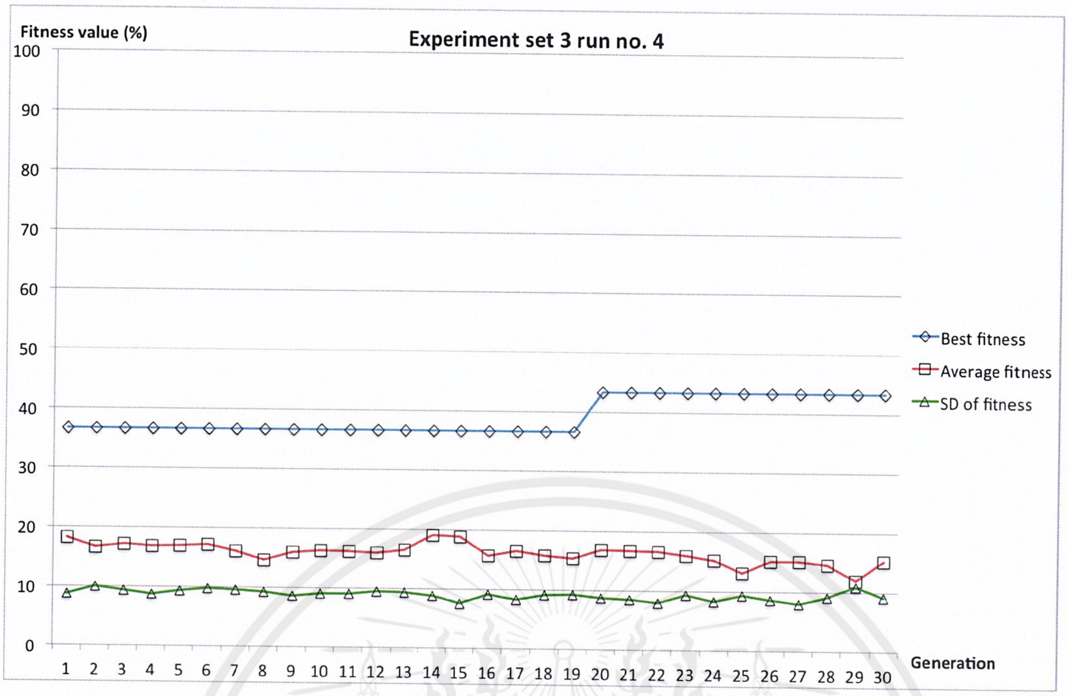
(q) Experiment set 3, Evolutionary run: 1



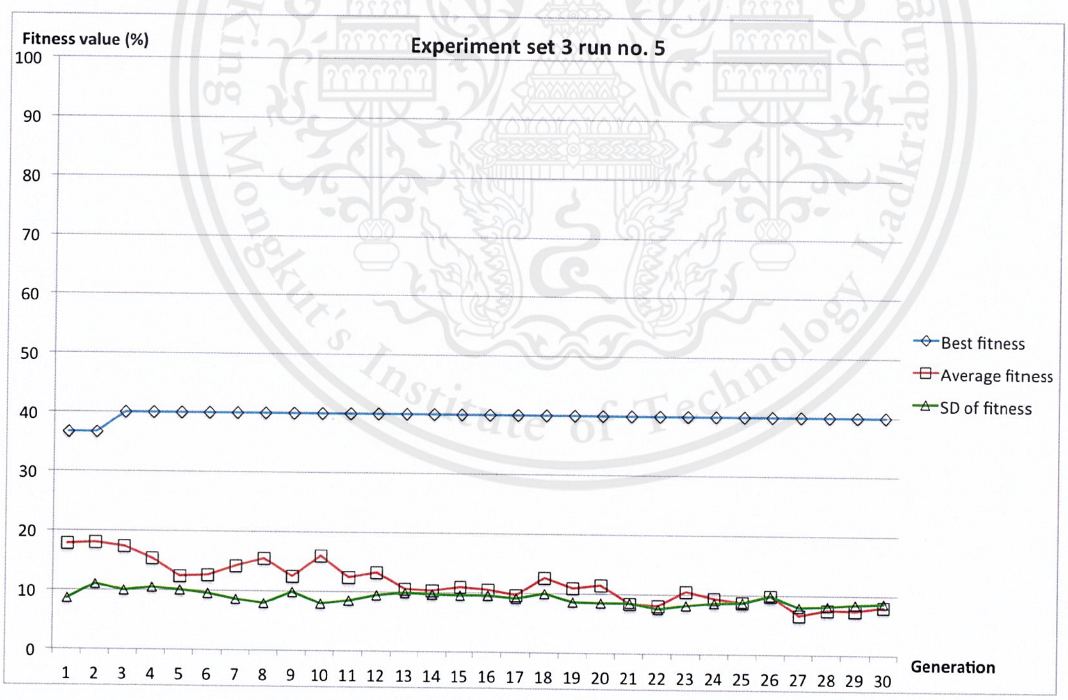
(r) Experiment set 3, Evolutionary run: 2



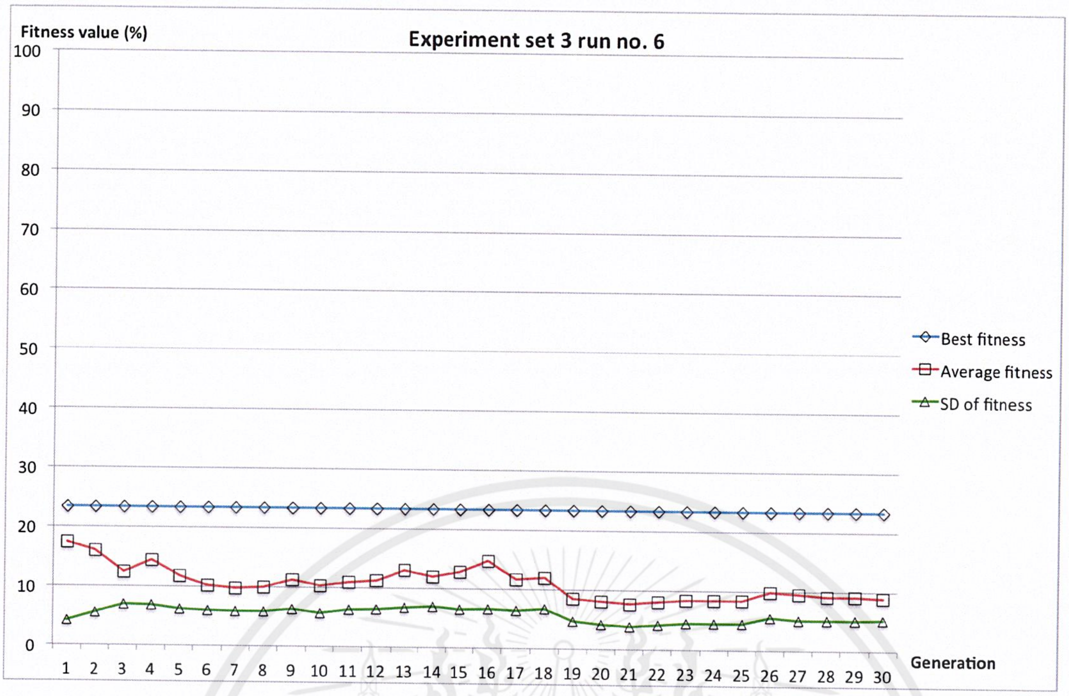
(s) Experiment set 3, Evolutionary run: 3



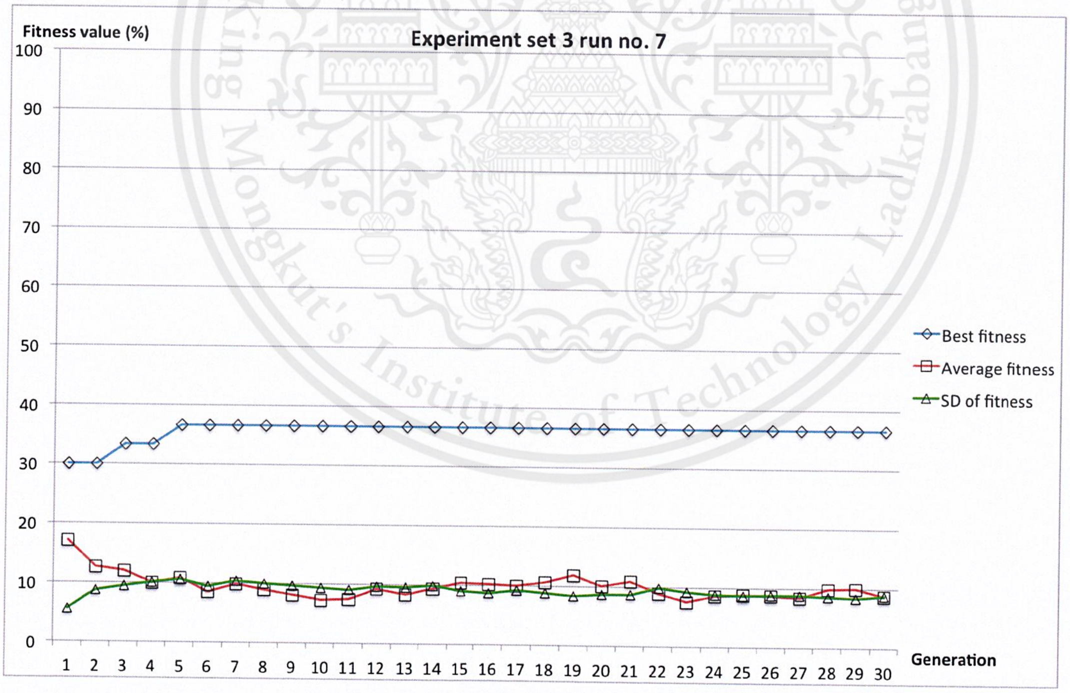
(t) Experiment set 3, Evolutionary run: 4



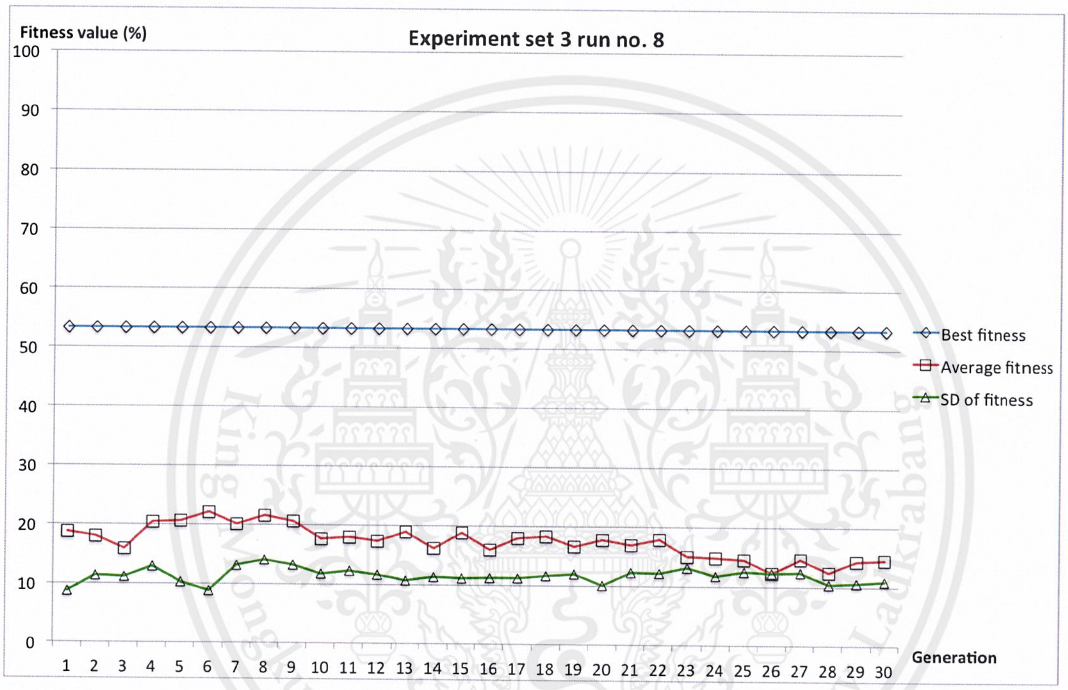
(u) Experiment set 3, Evolutionary run: 5



(v) Experiment set 3, Evolutionary run: 6



(w) Experiment set 3, Evolutionary run: 7



(x) Experiment set 3, Evolutionary run: 8

5.3 Integration Experiment

5.3.1 Objective

After implemented and conducted experiments on both algorithms, the last part is the experiments for testing the performance of the whole system. The objective of this experiment is to observe the performance of the whole system after both mobile and server were implemented.

5.3.2 Experiment Setup

From our dataset, as shown in Section 5.2.2, we chose some part of it to create a new dataset used for this experiment. The image in the dataset were categorized into days. In each day, the ones with an odds number were chosen. Therefore, in each day, 12 images per classes (six young leaves and six old leaves) were selected.

Furthermore, in this experiments, there were two users who conducted the experiments. The first person is the developer himself (developer) while another is the user who was newly introduced to the system (normal). The developer operates with the dataset on day number 1, 5, 9, 13, 17, 21, and 25. The normal user operates on day number 3, 7, 11, 15, 19, 23, and 27.

5.3.3 Experimental Results and Discussion

The results of this experiments are displayed in Tables 5.6 to 5.19.

If focusing on the segmentation results, comparing between the normal user and the developer, it seems the number of accepted segmentation results (shown in the segmentation column) for the normal user in the early days were worse than the developer. The possible cause is that the user took some time to get used to the touch screen to draw the contour. The number of accepted segmentation results increased after the third day

(day 11). However, the early stage segmentation results of the normal user could also reduce the ability to analyze the plants because of poor quality segmented images.

For nutrient analysis results, the program can display the result into most of the classes excepts for Nitrogen and Potassium classes. The reason might be because of the similarity of the symptoms characteristics which yields very close to some other deficiency class. In contrast, Complete and Calcium classes were the results which occurred the most. However, most of them were not accurate. Another observable changes were the analysis results which improved over the days of 9 - 13 (see column nutrient analysis of table Tables 5.10 to 5.12) and decrease afterwards. This means that the analysis results could yield more accurate results when trying to analyze the leaves that belong to the plant of some certain ages.

In conclusion, the segmentation process is significantly depended on users. For the experienced users like the developer of the system who created the application himself, the result is more satisfiable. However, normal users might require a certain amount of time to get to used to drawing the contour on the touch screen. For nutrient analysis program on the server, it could also be improved further to adapt so that the deficiency result could be more effective when detecting the deficiency on the earlier days.

The abbreviations of the Tables 5.6 to 5.19 are explained as follows:

- CS: Correct segmentation
- IS: Incorrect segmentation
- CC: Correct classification
- IC: Incorrect classification

Table 5.6: Experimental Result: Day 1

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	Ca	✓	-	-	-
Ca	3	Unacceptable	COM	-	-	-	✓
Ca	5	Unacceptable	Fe	-	-	-	✓
Ca	7	Acceptable	COM	-	✓	-	-
Ca	9	Acceptable	Ca	✓	-	-	-
Ca	11	Acceptable	Mg	-	✓	-	-
COM	1	Acceptable	Ca	-	✓	-	-
COM	3	Acceptable	Fe	-	✓	-	-
COM	5	Unacceptable	Fe	-	-	-	✓
COM	7	Acceptable	Fe	-	-	-	✓
COM	9	Acceptable	Ca	-	✓	-	-
COM	11	Acceptable	COM	✓	-	-	-
Fe	1	Unacceptable	COM	-	-	-	✓
Fe	3	Acceptable	Fe	✓	-	-	-
Fe	5	Unacceptable	COM	-	-	-	✓
Fe	7	Unacceptable	COM	-	-	-	✓
Fe	9	Acceptable	COM	-	✓	-	-
Fe	11	Unacceptable	COM	-	-	-	✓
K	1	Acceptable	Ca	-	✓	-	-
K	3	Acceptable	Mg	-	✓	-	-
K	5	Unacceptable	Ca	-	-	-	✓

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Unacceptable	Fe	-	-	-	✓
K	9	Acceptable	COM	-	✓	-	-
K	11	Acceptable	COM	-	✓	-	-
Mg	1	Acceptable	Ca	-	✓	-	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	Ca	-	✓	-	-
Mg	7	Unacceptable	COM	-	-	-	✓
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Unacceptable	COM	-	-	-	✓
N	1	Acceptable	COM	-	✓	-	-
N	3	Acceptable	Ca	-	✓	-	-
N	5	Acceptable	Fe	-	✓	-	-
N	7	Acceptable	COM	-	✓	-	-
N	9	Acceptable	COM	-	✓	-	-
N	11	Acceptable	Ca	-	✓	-	-

Table 5.7: Experimental Result: Day 3

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Unacceptable	Mg	-	-	-	✓
Ca	3	Unacceptable	Ca	-	-	✓	-
Ca	5	Acceptable	COM	-	✓	-	-
Ca	7	Unacceptable	COM	-	-	-	✓
Ca	9	Unacceptable	COM	-	-	-	✓
Ca	11	Unacceptable	COM	-	-	-	✓
COM	1	Unacceptable	COM	-	-	✓	-
COM	3	Unacceptable	COM	-	-	✓	-
COM	5	Unacceptable	Ca	-	-	-	✓
COM	7	Unacceptable	Fe	-	-	-	✓
COM	9	Unacceptable	Ca	-	-	-	✓
COM	11	Unacceptable	COM	-	-	✓	-
Fe	1	Unacceptable	COM	-	-	-	✓
Fe	3	Unacceptable	COM	-	-	-	✓
Fe	5	Unacceptable	COM	-	-	-	✓
Fe	7	Unacceptable	COM	-	-	-	✓
Fe	9	Unacceptable	COM	-	-	-	✓
Fe	11	Unacceptable	COM	-	-	-	✓
K	1	Unacceptable	COM	-	-	-	✓
K	3	Unacceptable	COM	-	-	-	✓
K	5	Acceptable	Ca	-	✓	-	-

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Unacceptable	Mg	-	-	-	✓
K	9	Unacceptable	COM	-	-	-	✓
K	11	Acceptable	COM	-	-	-	✓
Mg	1	Acceptable	COM	-	✓	-	-
Mg	3	Unacceptable	COM	-	-	-	✓
Mg	5	Unacceptable	COM	-	-	-	✓
Mg	7	Unacceptable	COM	-	-	-	✓
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Unacceptable	Ca	-	-	-	✓
N	1	Unacceptable	COM	-	-	-	✓
N	3	Acceptable	Ca	-	✓	-	-
N	5	Unacceptable	COM	-	-	-	✓
N	7	Acceptable	COM	-	✓	-	-
N	9	Unacceptable	COM	-	-	-	✓
N	11	Unacceptable	COM	-	-	-	✓

Table 5.8: Experimental Result: Day 5

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Unacceptable	Ca	-	-	✓	-
Ca	3	Acceptable	Ca	✓	-	-	-
Ca	5	Acceptable	Ca	✓	-	-	-
Ca	7	Unacceptable	Mg	-	-	-	✓
Ca	9	Acceptable	Ca	✓	-	-	-
Ca	11	Acceptable	Ca	✓	-	-	-
COM	1	Acceptable	Ca	-	✓	-	-
COM	3	Acceptable	Mg	-	✓	-	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	Ca	-	✓	-	-
COM	9	Acceptable	Ca	-	✓	-	-
COM	11	Acceptable	COM	✓	-	-	-
Fe	1	Acceptable	Fe	✓	-	-	-
Fe	3	Acceptable	Fe	✓	-	-	-
Fe	5	Unacceptable	Ca	-	-	-	✓
Fe	7	Acceptable	COM	-	✓	-	-
Fe	9	Acceptable	COM	-	✓	-	-
Fe	11	Unacceptable	COM	-	-	-	✓
K	1	Unacceptable	COM	-	-	-	✓
K	3	Unacceptable	Fe	-	-	-	✓
K	5	Acceptable	Mg	-	✓	-	-

continued ...

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	Mg	-	✓	-	-
K	9	Unacceptable	COM	-	-	-	✓
K	11	Acceptable	Ca	-	✓	-	-
Mg	1	Acceptable	Ca	-	✓	-	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	Ca	-	✓	-	-
Mg	7	Unacceptable	COM	-	-	-	✓
Mg	9	Acceptable	Ca	-	✓	-	-
Mg	11	Acceptable	Ca	-	✓	-	-
N	1	Acceptable	Ca	-	✓	-	-
N	3	Acceptable	Fe	-	✓	-	-
N	5	Acceptable	Ca	-	✓	-	-
N	7	Acceptable	COM	-	✓	-	-
N	9	Acceptable	Ca	-	✓	-	-
N	11	Unacceptable	COM	-	-	-	✓

Table 5.9: Experimental Result: Day 7

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	Fe	-	✓	-	-
Ca	3	Unacceptable	COM	-	-	-	✓
Ca	5	Unacceptable	COM	-	-	-	✓
Ca	7	Unacceptable	COM	-	-	-	✓
Ca	9	Unacceptable	COM	-	-	-	✓
Ca	11	Unacceptable	Mg	-	-	-	✓
COM	1	Unacceptable	COM	-	-	✓	-
COM	3	Unacceptable	COM	-	-	✓	-
COM	5	Unacceptable	COM	-	-	✓	-
COM	7	Unacceptable	Mg	-	-	-	✓
COM	9	Unacceptable	Fe	-	-	-	✓
COM	11	Acceptable	COM	✓	-	-	-
Fe	1	Acceptable	Mg	-	✓	-	-
Fe	3	Acceptable	COM	-	✓	-	-
Fe	5	Acceptable	Fe	✓	-	-	-
Fe	7	Acceptable	Fe	✓	-	-	-
Fe	9	Acceptable	COM	-	✓	-	-
Fe	11	Acceptable	Ca	-	✓	-	-
K	1	Acceptable	Mg	-	✓	-	-
K	3	Acceptable	COM	-	✓	-	-
K	5	Acceptable	Ca	-	✓	-	-

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Unacceptable	COM	-	-	-	✓
K	11	Acceptable	Mg	-	✓	-	-
Mg	1	Acceptable	Ca	-	✓	-	-
Mg	3	Unacceptable	COM	-	-	-	✓
Mg	5	Acceptable	COM	-	✓	-	-
Mg	7	Acceptable	Ca	-	✓	-	-
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Acceptable	COM	-	✓	-	-
N	1	Unacceptable	COM	-	-	-	✓
N	3	Acceptable	Ca	-	✓	-	-
N	5	Acceptable	COM	-	✓	-	-
N	7	Unacceptable	COM	-	-	-	✓
N	9	Unacceptable	COM	-	-	-	✓
N	11	Acceptable	COM	-	✓	-	-

Table 5.10: Experimental Result: Day 9

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	COM	-	✓	-	-
Ca	3	Acceptable	COM	-	✓	-	-
Ca	5	Acceptable	Ca	✓	-	-	-
Ca	7	Acceptable	COM	-	✓	-	-
Ca	9	Acceptable	Ca	✓	-	-	-
Ca	11	Acceptable	Mg	-	✓	-	-
COM	1	Acceptable	Ca	-	✓	-	-
COM	3	Acceptable	Ca	-	✓	-	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	COM	✓	-	-	-
COM	9	Acceptable	COM	✓	-	-	-
COM	11	Acceptable	Ca	-	✓	-	-
Fe	1	Acceptable	Ca	-	✓	-	-
Fe	3	Acceptable	Ca	-	✓	-	-
Fe	5	Acceptable	Ca	-	✓	-	-
Fe	7	Acceptable	Fe	✓	-	-	-
Fe	9	Unacceptable	COM	-	-	✓	-
Fe	11	Acceptable	Fe	✓	-	-	-
K	1	Acceptable	Ca	-	✓	-	-
K	3	Acceptable	COM	-	✓	-	-
K	5	Acceptable	Fe	-	✓	-	-

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Acceptable	COM	-	✓	-	-
Mg	1	Unacceptable	Mg	-	-	✓	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	COM	-	✓	-	-
Mg	7	Acceptable	COM	-	✓	-	-
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Acceptable	Mg	✓	-	-	-
N	1	Acceptable	COM	-	✓	-	-
N	3	Acceptable	Ca	-	✓	-	-
N	5	Acceptable	Fe	-	✓	-	-
N	7	Acceptable	COM	-	✓	-	-
N	9	Acceptable	COM	-	✓	-	-
N	11	Acceptable	Ca	-	✓	-	-

Table 5.11: Experimental Result: Day 11

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	Ca	✓	-	-	-
Ca	3	Acceptable	Ca	✓	-	-	-
Ca	5	Acceptable	Ca	✓	-	-	-
Ca	7	Acceptable	Mg	-	✓	-	-
Ca	9	Acceptable	COM	-	✓	-	-
Ca	11	Unacceptable	COM	-	-	-	✓
COM	1	Acceptable	COM	✓	-	-	-
COM	3	Unacceptable	COM	-	-	✓	-
COM	5	Unacceptable	Fe	-	-	-	✓
COM	7	Acceptable	Mg	-	✓	-	-
COM	9	Acceptable	Fe	-	✓	-	-
COM	11	Acceptable	COM	✓	-	-	-
Fe	1	Acceptable	Ca	-	✓	-	-
Fe	3	Acceptable	Fe	✓	-	-	-
Fe	5	Acceptable	Ca	-	✓	-	-
Fe	7	Unacceptable	COM	-	-	-	✓
Fe	9	Acceptable	Fe	✓	-	-	-
Fe	11	Unacceptable	COM	-	-	-	✓
K	1	Acceptable	Ca	-	✓	-	-
K	3	Acceptable	COM	-	-	-	-
K	5	Acceptable	Ca	-	✓	-	-

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Unacceptable	Mg	-	✓	-	-
K	9	Acceptable	Mg	-	✓	-	-
K	11	Acceptable	COM	-	✓	-	-
Mg	1	Acceptable	COM	-	✓	-	-
Mg	3	Acceptable	Fe	-	✓	-	-
Mg	5	Acceptable	Mg	✓	-	-	-
Mg	7	Acceptable	COM	-	✓	-	-
Mg	9	Acceptable	Fe	-	✓	-	-
Mg	11	Acceptable	Ca	-	✓	-	-
N	1	Acceptable	COM	-	✓	-	-
N	3	Acceptable	Ca	-	✓	-	-
N	5	Acceptable	Fe	-	✓	-	-
N	7	Acceptable	COM	-	✓	-	-
N	9	Acceptable	COM	-	✓	-	-
N	11	Acceptable	Mg	-	✓	-	-

Table 5.12: Experimental Result: Day 13

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	Ca	✓	-	-	-
Ca	3	-	-	-	-	-	-
Ca	5	Acceptable	Ca	✓	-	-	-
Ca	7	Acceptable	Ca	✓	-	-	-
Ca	9	Acceptable	COM	-	✓	-	-
Ca	11	Acceptable	Ca	✓	-	-	-
COM	1	Acceptable	Ca	-	✓	-	-
COM	3	Acceptable	COM	✓	-	-	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	COM	✓	-	-	-
COM	9	Acceptable	COM	✓	-	-	-
COM	11	Acceptable	COM	✓	-	-	-
Fe	1	Acceptable	Ca	-	✓	-	-
Fe	3	Acceptable	Ca	-	✓	-	-
Fe	5	Acceptable	COM	-	✓	-	-
Fe	7	Acceptable	COM	-	✓	-	-
Fe	9	Acceptable	Ca	-	✓	-	-
Fe	11	Acceptable	Ca	-	✓	-	-
K	1	Acceptable	Ca	-	✓	-	-
K	3	Acceptable	Ca	-	✓	-	-
K	5	Acceptable	COM	-	✓	-	-

continued ...

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Acceptable	COM	-	✓	-	-
Mg	1	Acceptable	COM	-	✓	-	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	Ca	-	✓	-	-
Mg	7	Acceptable	COM	-	✓	-	-
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Acceptable	Ca	-	✓	-	-
N	1	Unacceptable	Ca	-	-	-	✓
N	3	Acceptable	COM	-	✓	-	-
N	5	Acceptable	Ca	-	✓	-	-
N	7	Acceptable	COM	-	✓	-	-
N	9	Acceptable	COM	-	✓	-	-
N	11	Acceptable	Mg	-	✓	-	-

Table 5.13: Experimental Result: Day 15

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	COM	-	✓	-	-
Ca	3	-	-	-	-	-	-
Ca	5	Unacceptable	COM	-	-	-	✓
Ca	7	Acceptable	COM	-	✓	-	-
Ca	9	Acceptable	COM	-	✓	-	-
Ca	11	Unacceptable	COM	-	-	-	✓
COM	1	Acceptable	COM	✓	-	-	-
COM	3	Acceptable	Mg	-	✓	-	-
COM	5	Acceptable	COM	✓	-	-	-
COM	7	Unacceptable	COM	-	-	✓	-
COM	9	Acceptable	COM	✓	-	-	-
COM	11	Unacceptable	Mg	-	-	-	✓
Fe	1	Unacceptable	COM	-	-	-	✓
Fe	3	Acceptable	Fe	✓	-	-	-
Fe	5	Acceptable	Ca	-	✓	-	-
Fe	7	Acceptable	COM	-	✓	-	-
Fe	9	Acceptable	COM	-	✓	-	-
Fe	11	Unacceptable	COM	-	-	-	✓
K	1	Acceptable	Ca	-	✓	-	-
K	3	Acceptable	COM	-	✓	-	-
K	5	Acceptable	COM	-	✓	-	-

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Unacceptable	Fe	-	-	-	✓
Mg	1	Acceptable	COM	-	✓	-	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	COM	-	-	-	-
Mg	7	Acceptable	COM	-	✓	-	-
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Acceptable	COM	-	✓	-	-
N	1	Unacceptable	COM	-	-	-	✓
N	3	Acceptable	Mg	-	✓	-	-
N	5	Acceptable	COM	-	✓	-	-
N	7	Unacceptable	COM	-	-	-	✓
N	9	Acceptable	COM	-	-	-	✓
N	11	Acceptable	Ca	-	✓	-	-

Table 5.14: Experimental Result: Day 17

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	Fe	-	✓	-	-
Ca	3	-	-	-	-	-	-
Ca	5	Acceptable	Ca	✓	-	-	-
Ca	7	Acceptable	Fe	-	✓	-	-
Ca	9	Acceptable	Ca	✓	-	-	-
Ca	11	Unacceptable	COM	-	-	-	✓
COM	1	Unacceptable	COM	-	-	✓	-
COM	3	Acceptable	COM	✓	-	-	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	COM	✓	-	-	-
COM	9	Acceptable	COM	✓	-	-	-
COM	11	Acceptable	COM	✓	-	-	-
Fe	1	Unacceptable	COM	-	-	-	✓
Fe	3	Acceptable	Fe	✓	-	-	-
Fe	5	Acceptable	Ca	-	✓	-	-
Fe	7	Acceptable	Ca	-	✓	-	-
Fe	9	Acceptable	Fe	✓	-	-	-
Fe	11	Acceptable	Ca	-	✓	-	-
K	1	Acceptable	Ca	-	✓	-	-
K	3	Acceptable	COM	-	✓	-	-
K	5	Unacceptable	COM	-	-	-	✓

continued ...

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	Ca	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Acceptable	COM	-	✓	-	-
Mg	1	Acceptable	Ca	-	✓	-	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	COM	-	✓	-	-
Mg	7	Unacceptable	COM	-	-	-	✓
Mg	9	Acceptable	Ca	-	✓	-	-
Mg	11	Acceptable	COM	-	✓	-	-
N	1	Acceptable	Ca	-	✓	-	-
N	3	Acceptable	Fe	-	✓	-	-
N	5	Acceptable	COM	-	✓	-	-
N	7	Unacceptable	COM	-	-	-	✓
N	9	Acceptable	Ca	-	✓	-	-
N	11	Unacceptable	Fe	-	-	-	✓

Table 5.15: Experimental Result: Day 19

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Acceptable	Ca	✓	-	-	-
Ca	3	-	-	-	-	-	-
Ca	5	Unacceptable	Mg	-	-	-	✓
Ca	7	Unacceptable	COM	-	-	-	✓
Ca	9	Acceptable	Ca	✓	-	-	-
Ca	11	Acceptable	Mg	✓	-	-	-
COM	1	Unacceptable	COM	-	-	✓	-
COM	3	Acceptable	COM	✓	-	-	-
COM	5	Unacceptable	COM	-	-	✓	-
COM	7	Unacceptable	COM	-	-	✓	-
COM	9	Acceptable	COM	✓	-	-	-
COM	11	Acceptable	Ca	-	✓	-	-
Fe	1	Unacceptable	COM	-	-	-	✓
Fe	3	Unacceptable	COM	-	-	-	✓
Fe	5	Acceptable	COM	-	✓	-	-
Fe	7	Acceptable	COM	-	✓	-	-
Fe	9	Unacceptable	Fe	-	-	✓	-
Fe	11	Acceptable	Ca	-	✓	-	-
K	1	Acceptable	COM	-	✓	-	-
K	3	Acceptable	COM	-	✓	-	-
K	5	Unacceptable	COM	-	-	-	✓

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Acceptable	Ca	-	✓	-	-
Mg	1	Acceptable	COM	-	✓	-	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	COM	-	✓	-	-
Mg	7	Acceptable	COM	-	✓	-	-
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Acceptable	COM	-	✓	-	-
N	1	Acceptable	COM	-	✓	-	-
N	3	Acceptable	COM	-	✓	-	-
N	5	Acceptable	Fe	-	✓	-	-
N	7	Acceptable	COM	-	✓	-	-
N	9	Acceptable	COM	-	✓	-	-
N	11	Acceptable	COM	-	✓	-	-

Table 5.16: Experimental Result: Day 21

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	Unacceptable	Ca	-	-	✓	-
Ca	3	-	COM	-	-	-	-
Ca	5	Acceptable	COM	-	✓	-	-
Ca	7	Unacceptable	Ca	-	-	✓	-
Ca	9	Acceptable	Mg	-	✓	-	-
Ca	11	Unacceptable	COM	-	✓	-	-
COM	1	Acceptable	Ca	-	✓	-	-
COM	3	Acceptable	COM	✓	-	-	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	COM	✓	-	-	-
COM	9	Acceptable	Ca	-	✓	-	-
COM	11	Acceptable	COM	✓	-	-	-
Fe	1	Acceptable	COM	-	✓	-	-
Fe	3	Acceptable	Ca	-	✓	-	-
Fe	5	Acceptable	COM	-	✓	-	-
Fe	7	Acceptable	COM	-	✓	-	-
Fe	9	Acceptable	Fe	✓	-	-	-
Fe	11	Acceptable	Ca	-	✓	-	-
K	1	Acceptable	Fe	-	✓	-	-
K	3	Acceptable	Ca	-	✓	-	-
K	5	Unacceptable	COM	-	-	-	✓

continued ...

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Unacceptable	Mg	-	-	-	✓
Mg	1	Acceptable	Ca	-	✓	-	-
Mg	3	Acceptable	Ca	-	✓	-	-
Mg	5	Acceptable	Ca	-	-	-	-
Mg	7	Acceptable	Fe	-	✓	-	-
Mg	9	Acceptable	Mg	✓	-	-	-
Mg	11	Acceptable	COM	-	✓	-	-
N	1	Acceptable	Mg	-	✓	-	-
N	3	Acceptable	COM	-	✓	-	-
N	5	Acceptable	COM	-	✓	-	-
N	7	Acceptable	Fe	-	✓	-	-
N	9	Acceptable	Ca	-	✓	-	-
N	11	Acceptable	Ca	-	✓	-	-

Table 5.17: Experimental Result: Day 23

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	-	-	-	-	-	-
Ca	3	-	-	-	-	-	-
Ca	5	-	-	-	-	-	-
Ca	7	-	-	-	-	-	-
Ca	9	Acceptable	COM	-	✓	-	-
Ca	11	Acceptable	Ca	✓	-	-	-
COM	1	Unacceptable	COM	-	-	✓	-
COM	3	Unacceptable	COM	-	-	✓	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	Ca	-	✓	-	-
COM	9	Acceptable	Ca	-	✓	-	-
COM	11	Acceptable	Ca	-	✓	-	-
Fe	1	Unacceptable	COM	-	-	-	✓
Fe	3	Unacceptable	Ca	-	✓	-	-
Fe	5	Acceptable	Fe	✓	-	-	-
Fe	7	Acceptable	COM	-	✓	-	-
Fe	9	Unacceptable	COM	-	-	-	✓
Fe	11	Acceptable	Ca	-	✓	-	-
K	1	Acceptable	COM	-	✓	-	-
K	3	Acceptable	COM	-	-	-	✓
K	5	Unacceptable	COM	-	✓	-	-

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	Fe	-	✓	-	-
K	11	Acceptable	Ca	-	✓	-	-
Mg	1	Acceptable	Ca	-	✓	-	-
Mg	3	Unacceptable	COM	-	-	-	✓
Mg	5	Acceptable	Ca	-	✓	-	-
Mg	7	Acceptable	COM	-	✓	-	-
Mg	9	Unacceptable	COM	-	-	-	✓
Mg	11	Unacceptable	COM	-	-	-	✓
N	1	Acceptable	Fe	-	✓	-	-
N	3	Acceptable	Ca	-	✓	-	-
N	5	Unacceptable	Fe	-	-	-	✓
N	7	Unacceptable	COM	-	-	-	✓
N	9	Acceptable	Ca	-	✓	-	-
N	11	Acceptable	COM	-	✓	-	-

Table 5.18: Experimental Result: Day 25

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	-	-	-	-	-	-
Ca	3	-	-	-	-	-	-
Ca	5	-	-	-	-	-	-
Ca	7	-	-	-	-	-	-
Ca	9	Acceptable	COM	-	✓	-	-
Ca	11	Acceptable	Ca	✓	-	-	-
COM	1	Acceptable	Ca	-	✓	-	-
COM	3	Acceptable	Ca	-	✓	-	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	Mg	-	✓	-	-
COM	9	Acceptable	Ca	-	✓	-	-
COM	11	Acceptable	Ca	-	✓	-	-
Fe	1	Unacceptable	Fe	-	-	✓	-
Fe	3	Acceptable	Fe	✓	-	-	-
Fe	5	Acceptable	COM	-	✓	-	-
Fe	7	Acceptable	Ca	-	✓	-	-
Fe	9	Acceptable	Fe	✓	-	-	-
Fe	11	Unacceptable	COM	-	-	-	✓
K	1	Acceptable	COM	-	✓	-	-
K	3	Acceptable	COM	-	✓	-	-
K	5	Acceptable	Ca	-	✓	-	-

continued ...

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Acceptable	Ca	-	✓	-	-
Mg	1	Acceptable	Ca	-	✓	-	-
Mg	3	Acceptable	COM	-	✓	-	-
Mg	5	Acceptable	COM	-	✓	-	-
Mg	7	Acceptable	COM	-	✓	-	-
Mg	9	Acceptable	Fe	-	✓	-	-
Mg	11	Acceptable	COM	-	✓	-	-
N	1	Acceptable	Fe	-	✓	-	-
N	3	Acceptable	Fe	-	✓	-	-
N	5	Acceptable	Ca	-	✓	-	-
N	7	Acceptable	Ca	-	✓	-	-
N	9	Acceptable	Ca	-	✓	-	-
N	11	Acceptable	Ca	-	✓	-	-

Table 5.19: Experimental Result: Day 27

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
Ca	1	-	-	-	-	-	-
Ca	3	-	-	-	-	-	-
Ca	5	-	-	-	-	-	-
Ca	7	-	-	-	-	-	-
Ca	9	Acceptable	Fe	-	✓	-	-
Ca	11	Acceptable	Ca	✓	-	-	-
COM	1	Acceptable	COM	✓	-	-	-
COM	3	Acceptable	COM	✓	-	-	-
COM	5	Acceptable	Ca	-	✓	-	-
COM	7	Acceptable	Ca	-	✓	-	-
COM	9	Acceptable	Ca	-	✓	-	-
COM	11	Acceptable	Ca	-	✓	-	-
Fe	1	Acceptable	Ca	-	✓	-	-
Fe	3	Unacceptable	COM	-	-	-	✓
Fe	5	Acceptable	COM	-	✓	-	-
Fe	7	-	-	-	-	-	-
Fe	9	Unacceptable	COM	-	-	-	✓
Fe	11	Acceptable	Fe	✓	-	-	-
K	1	Acceptable	COM	-	✓	-	-
K	3	Acceptable	COM	-	✓	-	-
K	5	Unacceptable	Ca	-	-	-	✓

continued . . .

...continued

Image Class	Plant Number	Segmentation Result	Nutrient Analysis	CS and CC	CS but IC	IS but CC	IS and IC
K	7	Acceptable	COM	-	✓	-	-
K	9	Acceptable	COM	-	✓	-	-
K	11	Unacceptable	Mg	-	-	-	✓
Mg	1	Acceptable	COM	-	✓	-	-
Mg	3	Acceptable	Ca	-	✓	-	-
Mg	5	Acceptable	COM	-	✓	-	-
Mg	7	Unacceptable	Ca	-	-	-	✓
Mg	9	Acceptable	COM	-	✓	-	-
Mg	11	Acceptable	Ca	-	✓	-	-
N	1	Acceptable	Fe	-	✓	-	-
N	3	Unacceptable	COM	-	-	-	✓
N	5	Acceptable	COM	-	✓	-	-
N	7	Acceptable	Fe	-	✓	-	-
N	9	Acceptable	Ca	-	✓	-	-
N	11	Acceptable	Ca	-	✓	-	-

Chapter 6

Summary

To sum up, the proposed system, which composed of a mobile application and a processing server, is successfully integrated into a single automatic system. For the mobile application, it is used for interacting with the users and segmenting the leaves' images. The processing server is used for analyzing a nutrient deficiency in a black gram.

The proposed segmentation algorithm is the GrabCut segmentation method. This method requires the users to draw the contour on the photo. Then, the algorithm will accordingly segmented the image as the received contour and the result will be sent to the processing server. As a result of this algorithm, the segmentation result depends on the user. If the user can perfectly draw the contour, the segmentation result will be favorable. Otherwise, the segmentation result will be poor. Furthermore, the segmentation result is significantly affected the result of the nutrient analysis.

For the nutrient analysis algorithm, after it receives the inputs from the mobile application, the program will process the input image to analyze the deficiency symptom. The deficiency symptoms include Complete, Calcium, Nitrogen, Potassium, Iron, and Magnesium deficiency classes. However, from the experiments, some of the deficiency classes could not be detected, while most of the classes were able to be identified by the algorithm.

Ultimately, both algorithms, especially the nutrient analysis algorithm, possess the potentials to be developed further to improve its result. In the future, the scope of the system could be expanded further so that it will become more generic, which mean the system will not only work with the black gram plant, but also other plants as well given there is a sample data to operate on.

Bibliography

[1] R. C. Gonzalez and R. E. Woods, Digital Image Processing (3rd Edition), Pearson Education, 2010.

[2] R. Poli, W. B. Langdon, and N. F. McPhee, A Field Guide to Genetic Programming, Lulu Enterprises, 2008.

[3] K. Krawiec and B. Bhanu, "Visual learning by evolutionary feature synthesis," Proceedings of the Twentieth International Conference on Machine Learning, pp.376–383, 2003.

[4] G. Xu, F. Zhang, S. G. Shah, Y. Ye, and H. Mao, "Use of leaf color images to identify nitrogen and potassium deficient tomatoes," Pattern Recognition Letters, vol.32, pp.1584–1590, 2011.

[5] D. Story, M. Kacira, C. K. Ali Akoglu, and L. An, "Lettuce calcium deficiency detection with machine vision computed plant features in controlled environments," Computers and Electronic in Agriculture, vol.74, pp.238–243, 2010.

[6] D. Ushada, H. Murase, and H. Fukuda, "Non–destructive sensing and its inverse model for canopy parameters using texture analysis and artificial neural network," Computer and Electronics in Agriculture, vol.57, pp.149–165, 2007.

[7] M. Rojanamontien, N. Piemkaroonwong, P. Sihanatkathakul, and S. Kamales,

Leaf Identifier, Team Software Project Report, International College, King Mongkut's Institute of Technology Ladkrabang, 2015.

[8] M. A. Potter and K. A. De Jong, "Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents," Evolutionary Computation, vol.8, pp.1–29, 2000.

[9] U. Watchareeruetai, Y. Takeuchi, T. Matsumoto, H. Kudo, and N. Ohnishi, "Redundancies in linear GP, canonical transformation, and its exploitation: a demonstration on image feature synthesis," Genetic Programming and Evolvable Machines, vol.12, no.1, pp.49–77, 2011.

[10] M. Brameier and W. Banzhaf, Linear Genetic Programming, Springer, 2006.

[11] F. Yi and I. Moon, "Image Segmentation: A Survey of Graph–Cut Methods," International Conference on Systems and Informatics (ICSAI), pp.1936–1941, 2012.

[12] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut — Interactive Foreground Extraction using Iterated Graph Cuts," ACM Trans. Graph, vol.23, no.3, pp.309–314, 2004.

[13] S. Raut, M. Raghuvanshi, R. Dharaskar, and A, Raut, "Image Segmentation — A State-of-Art Survey for Prediction," International Conference On Advances In Computing & Communications (ICACC), pp.420–424, 2009.

[14] S. Naz, H. Majeed, and H. Irshad, "Image segmentation using fuzzy clustering: A survey," International conference on emerging technologies, pp.181–186, 2010.

[15] L. Wang, "Comparison for Edge Detection of Colony Image," International Journal of Computer Science and Network Security (IJCSNS), vol.6, no.9A, 2006.

[16] Z. Musoromy, R. Soodamani, and B. Nico, "Edge Detection Comparison for License Plate Detection," Int.Control, Automation, Robotics, and Vision, 2010.

[17] I. Karoui, J. Boucher, and J. Augustin, "Variation Region-Based Segmentation Using Multiple Texture Statistics." IEEE trans on image processing, vol.19, no.12, 2010.

[18] C. Rambabu, I. Chakrabarti, and A. Mahanta, "Flooding-based watershed algorithm and its prototype hardware architecture," IEEE Proceedings — vision image and signal processing, vol.151, no.3, pp.224–234, 2004.

[19] A. X. Falcao, J. K. Udupa, and F. K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: live wire on the fly," IEEE trans on Medical Imaging, vol.19, no.1, pp.55–62, 2002.

[20] G. Sundaramoorthi, A. Yezzi, and A. C. Mennucci, "Coarse-to-Fine Segmentation and Tracking Using Sobolev Active Contours," IEEE trans on Pattern Analysis and Machine Intelligence, vol.30, no.5, pp.851–864, 2008.

[21] R. M. Willett and R. D. Nowak, "Minimax Optimal Level-Set Estimation," IEEE trans on Image Processing, vol.16, no.12, pp.2965–2979, 2007.

[22] H. Yu, D. Wang, and Z. Tan, "Level Set methods and image segmentation," International workshop on medical imaging and augmented reality, pp.204–208, 2002.

[23] Y. B. Yuri and G. F. Lea, "Graph Cuts and Efficient N-D Image Segmentation," International Journal of Computer Vision, vol.70, no.2, pp.109–131, 2006.

[24] Y. B. Yuri and M. P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," Proceedings of International Conference on Computer Vision, vol.1, pp.105–112, 2001.

[25] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," IEEE trans on Pattern Analysis and Machine Intelligence (PAMI), vol.26, no.9, pp.1124–1137, 2004.