ON THE STUDY OF USING SELF-ORGANIZING MAP FOR INTENSITY-BASED AERIAL COLOR IMAGE QUANTIZATION

SUPARAT WIRATTANAPORNKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2009

Thesis
Entitled

ON THE STUDY OF USING SELF-ORGANIZING MAP FOR INTENSITY-BASED AERIAL COLOR IMAGE QUANTIZATION

.....

Mr.Suparat Wirattanapornkul
Candidate

.....

Assist. Prof. Bunlur Emaruchi,
Ph.D. (Environmental Systems Engineering)
Major-Advisor

.....

Lect. Songpol Ongwattanakul,
Ph.D. (Computer Engineering)
Co-Advisor

.....

Lect. Rangsipan Marukatat,
Ph.D. (Computer Science)
Co-Advisor

.....

Prof. Banchong Mahaisavariya,
M.D.
Dean
Faculty of Graduate Studies
Mahidol University

.....

Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management),
Acting Chair
Master of Science Programme in
Technology of Information
System Management
Faculty of Engineering
Mahidol University

Thesis
Entitled

ON THE STUDY OF USING SELF-ORGANIZING MAP FOR INTENSITY-BASED AERIAL COLOR IMAGE QUANTIZATION

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science
(Technology of Information System Management)
on
October 14, 2009

.....
Mr.Suparat Wirattanapornkul
Candidate

.....
Lect. Yodchanan Wongsawat,
Ph.D. (Electrical Engineering)
Chair

.....
Assist. Prof. Bunlur Emaruchi,
(Environmental Systems Engineering)
Member

.....
Lect. Songpol Ongwattanakul,
Ph.D. (Computer Engineering)
Member

.....
Mr. Sanparith Marukatat,
Ph.D. (Computer Science)
Member

.....
Lect. Rangsipan Marukatat,
Ph.D. (Computer Science)
Member

.....
Prof. Banchong Mahaisavariya,
M.D.
Dean
Faculty of Graduate Studies
Mahidol University

.....
Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management)
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENTS

The success of this thesis can be succeeded by the attentive support from my major advisor, Asst. Prof. Dr. Bunlur Emaruchi. I would like to express my appreciation and deeply thank him for the valuable advices and several suggestions reinforced my perception and knowledge to practice this research.

I would like to express my appreciation and deeply thank to Dr. Sanparith Marukatat, the external examiner of the thesis defense and Mr. Noppadol Teeranachaideekul, my senior, for their kindness in valuable suggestion and time sacrifice through this study. Special thanks to Dr. Songpol Ongwattanakul and Dr. Rangsipan Marukatat, my co-advisor for attentiveness and time sacrifice for this research.

Thank to my friends and staffs in Technology of Information System Management department Mahidol University especially Miss Janjira Jamjan and Mr. Saksiri Jamsak, for their help, kindness, support and cooperation during this study.

Finally, I am grateful to my family, especially my parents for unconditional support, entirely care, love and encouragement through my life. I hope that this thesis will usefully for the Aerial Image Segmentation for GIS development regardless most or least. The use fullness of this thesis, I dedicate to my father, my mother and all of my teachers.

Suparat Wirattanapornkul

ON THE STUDY OF USING SELF-ORGANIZING MAP FOR INTENSITY-BASED AERIAL COLOR IMAGE QUANTIZATION.

SUPARAT WIRATTANAPORNKUL 4936781 EGTI/M

M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)

THESIS ADVISORY COMMITEE: BUNLUR EMARUCHI, Ph.D. (ENVIRONMENTAL SYSTEMS ENGINEERING), SONGPOL ONGWATTANAKUL, Ph.D. (COMPUTER ENGINEERING), RANGSIPAN MARUKATAT, Ph.D. (COMPUTER SCIENCE)

ABSTRACT

Aerial imagery has many applications in mapping land-use and cover, forestry, city planning and others. The methods that help computers to understand and interpret the information on aerial photographs directly as humans do in order to construct a digital map are very important in many Geographic Information System (GIS) applications. A Self-Organizing Map (SOM) is a powerful tool which has been used for the clustering of color images. SOM is an unsupervised neural network mapping a set of n-dimensional vectors to a two-dimensional topographic map. However, its application in aerial image color clustering is not widely known.

In this work, the study compares the two types of SOM: sequential training SOM that sequentially updates the weights of SOM; and batch learning SOM (BLSOM) that presents the whole data set to the map before any adjustments are made with K-Means and Fuzzy C-Means (FCM) algorithm. By benchmarking a performance and computational time of the clustering process the study demonstrates that BLSOM can outperform the sequential training SOM in computational time with no significant decrease in performance and outperform K-Means and FCM in road color clustering for up to 82% of detection. To ensure that this clustering method is efficient and highly reliable, the study compares the clustering result with hand clustering.

KEY WORDS: SELF-ORGANIZING MAP/ AERIAL IMAGE / IMAGE QUANTIZATION /

65 pages

การศึกษาเรื่องการนำแผนผังการจัดระบบตัวเองเพื่อใช้ในการแบ่งนับความเข้มของภาพถ่ายทางอากาศ
ON THE STUDY OF USING SELF-ORGANIZING MAP FOR INTENSITY-BASED AERIAL COLOR IMAGE QUANTIZATION

ศุภรัตน์ วิรัตนพรกุล 4936781 EGTI/M

วท.ม. (เทคโนโลยีการจัดการระบบสารสนเทศ)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : บันลือ เอมะรุจิ, Ph.D. (ENVIRONMENTAL SYSTEMS ENGINERING), ทรงพล องค์วัฒนกุล, Ph.D. (COMPUTER ENGINEERING), รังสิพรรณ มฤคทัต, Ph.D. (COMPUTER SCIENCE)

บทคัดย่อ

ภาพถ่ายอากาศมีการประยุกต์ใช้ในงานด้านการวางแผนการใช้งานพื้นที่ดิน ป่าไม้ การวางผังเมือง และอื่นๆ ขั้นตอนที่ช่วยให้คอมพิวเตอร์เข้าใจและแปรข้อมูลจากภาพถ่ายอากาศได้เหมือนมนุษย์ในการสร้างแผนที่ดิจิตอลจึงมีความสำคัญมากต่องานด้านระบบสารสนเทศภูมิศาสตร์ แผนผังการจัดระบบตัวเอง จัดเป็นโครงข่ายประสาทเทียมแบบไม่ต้องมีการฝึกที่ลดมิติของข้อมูลเวกเตอร์หลายมิติให้เป็นแผนผังแบบ 2 มิติ ซึ่งเป็นเทคนิคที่มีประสิทธิภาพสูงในการแยกองค์ประกอบของภาพสี อย่างไรก็ดีการประยุกต์ใช้ในการแยกกลุ่มสีของภาพถ่ายอากาศยังไม่เป็นที่แพร่หลายเท่าที่ควร

งานวิจัยนี้ได้มีการเปรียบเทียบแผนผังการจัดระบบตัวเองสองแบบ คือแผนผังการจัดระบบตัวเองแบบตามลำดับขั้นที่จะมีการปรับค่าน้ำหนักของแผนผังที่ละขั้น กับแผนผังการจัดระบบตัวเองแบบทีละรอบซึ่งข้อมูลทั้งหมดจะถูกส่งเข้าไปยังแผนผังการจัดระบบตัวเองก่อนที่จะมีการปรับค่าน้ำหนัก นอกจากนี้ยังเปรียบเทียบแผนผังการจัดระบบตัวเองทั้งสองกับอัลกอริทึมแบบ K-Means และ Fuzzy C-Means จากผลการทดสอบพบว่าแผนผังการจัดระบบตัวเองทีละรอบให้ผลที่ดีกว่าในแง่เวลาที่ใช้ในการคำนวณเมื่อเทียบการแผนผังการจัดระบบตัวเองแบบตามลำดับถึง 9 เท่า และให้ผลที่ดีกว่าอัลกอริทึมแบบ K-Means และ Fuzzy C-Means ในแง่ผลที่ได้จากการแยกกลุ่มสี โดยสามารถแยกส่วนที่เป็นถนนได้ถูกต้องมากถึง 82 เปอร์เซ็นต์ ในการวัดความถูกต้องนี้เราเปรียบเทียบผลที่ได้จากอัลกอริทึมกับการแยกกลุ่มสีของภาพถ่ายโดยมนุษย์

65 หน้า

CONTENTS

CONTENTS (cont)

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 General Introduction

Aerial imagery [1] is the form of remote sensing that captures images of objects using photographic or digital cameras and film from platforms in the atmosphere. It has many applications in mapping land-use and cover, agriculture, soils mapping, forestry, city planning, archaeological investigations, military observation, and geomorphologic surveying, among other uses. For example, foresters use aerial photographs for preparing forest cover maps, locating possible access roads, and measuring quantities of trees harvested.

Remote sensing images are recorded in digital forms and then processed by the computers to produce images for interpretation purposes. Images are available in two forms - photographic film form and digital form. Variations in the scene characteristics are represented as variations in brightness on photographic films. A particular part of scene reflecting more energy will appear bright while a different part of the same scene that reflecting less energy will appear black. Digital image consists of discrete picture elements called pixels. Associated with each pixel is a number represented as DN (Digital Number), which depicts the average radiance of relatively small area within a scene. The size of this area effects the reproduction of details within the scene. As the pixel size is reduced more scene detail is preserved in digital representation. For this reason, the methods that help computers do understand and interpret the information on aerial photographs directly as humans do are very important and widely interested by researchers.

Image Segmentation [2], the important step that help computers do understand and interpret the information on aerial photographs, means the partitioning of image into meaningful regions based on homogeneity or heterogeneity criteria, respectively. It represents the interface between image pre-processing and image understanding (object recognition). Image segmentation techniques can be differentiated into the

following basic concepts: Histogram Thresholding, Region Based Approaches, Edge Detection, Fuzzy based Techniques, and Neural Networks Approaches which we selected to use in this work. The readers can see [3] for further reading. Color clustering is the basic step for this process.

1.2 Statement of Problem

Until now, most approaches based on Neural Networks can separate to two criterions, the first is known as supervised learning that requires a priori knowledge of image to get a successful clustering process, and sometimes, the required information may not be available. The second is known as unsupervised learning that attempts to construct the “natural grouping” of the image without using any prior knowledge of what color represent an object, instead, separates the image based on the sample of object colors. The unsupervised segmentation is widely used in the applications where the image features are unknown, such as pixel colors and shape of objects. The supervised segmentation is commonly used in the applications where the sample of object colors can be acquired in advance. e.g., object tracking, face/gesture recognition, and image retrieval etc. From this reason, then we chose the unsupervised segmentation in this work.

Self-Organizing Map (SOM) has been used for the clustering of color image and well reported in literature [11-17], however, its application in aerial image color clustering is not widely known. SOM can serve as a clustering tool for high-dimensional data, which constructs a topology that the high-dimensional space is mapped onto map units in such a way that relative topology distances between input data are preserved. The map units usually form a two-dimensional regular lattice. The training process of SOM is quite simple. Each map unit is associated with a reference vector. At first, all the reference vectors are randomly designated. Each input vector is compared with all the reference vectors and the unit whose reference vector is most similar to the input vector is identified. Then, the reference vectors neighboring to that of the identified unit are moved to the input vector.

1.3 Objective

1. To employ the SOM in the clustering of color image especially in aerial image.

2. To benchmark a performance of the clustering process between SOM and other clustering algorithms.

3. To benchmark a performance of the color clustering in the clustering process between SOM and other clustering algorithms.

4. To benchmark a computational time of the SOM color reduction with other clustering algorithms.

1.4 Scope of Work

The scope of this study includes;

1. Comparing a performance and computational time of the SOM color reduction with other clustering algorithms.
2. Sample maps used in this study consist of 12 aerial color photographs.
3. All pixels in image are clustered and classified into one of these four classes: Buildings, Lands and Roads, Lands with grasses and Forest.

1.5 Expected Result

This work proposes the method of aerial image color clustering employed by SOM. The performance, computational time, advantage and disadvantage of this algorithm are analyzed.

CHAPTER II
LITERATURE REVIEW

2.1 Image Segmentation Definition

Image segmentation [4] is a process of partitioning an image into disjoint regions. These regions have the properties: (1) homogeneity within region, i.e., textures or colors that in a region should be as similar as possible, and (2) heterogeneity between the regions, i.e., textures of colors that in one region should distinguished from those in another region. A more formal definition of segmentation is given in the following way: Let I be an image and H be a homogeneity predicate then the segmentation of I is a partition of I into a set of regions $\{R_1, R_2, \dots, R_n\}$ such that:

$$\bigcup_{i=1}^{n} R_i = I \quad with\ R_i \cap R_j = \emptyset, i \neq j \tag{1}$$

$$H(R_i) = true, \forall i, \tag{2}$$

and

$$H(R_i \cup R_j) = false, \qquad \forall R_i\ and\ \forall R_{j\ adjacent} \tag{3}$$

A variety of approached have been proposed to solve this problem. H.D. Cheng et al [3] summarized these methods into the following basic concepts: Histogram Thresholding, Region Based Approaches, Edge Detection, Fuzzy based Techniques, and Neural Networks Approaches. Some researchers may be classified the methods in different way such as Xu et al [5] who summarized these method into two categories: (1) boundary detection-based approaches, which try to search closed boundary contours to partition an image, and (2) region clustering-based approach, which group “similar” neighboring pixels into clusters. Color image quantization in this research is the basic of clustering image into meaningful region.

2.2 RGB Color space

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

RGB is a device-dependent color space: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same color across devices without some kind of color management. Typical RGB input devices are color TV and video cameras, image scanners, and digital cameras. Typical RGB output devices are TV sets of various technologies (CRT, LCD, plasma, etc.), computer and mobile phone displays, video projectors, multicolor LED displays, and large screens as JumboTron, etc. Color printers, on the other hand, are not RGB devices, but subtractive color devices (typically CMYK color model).

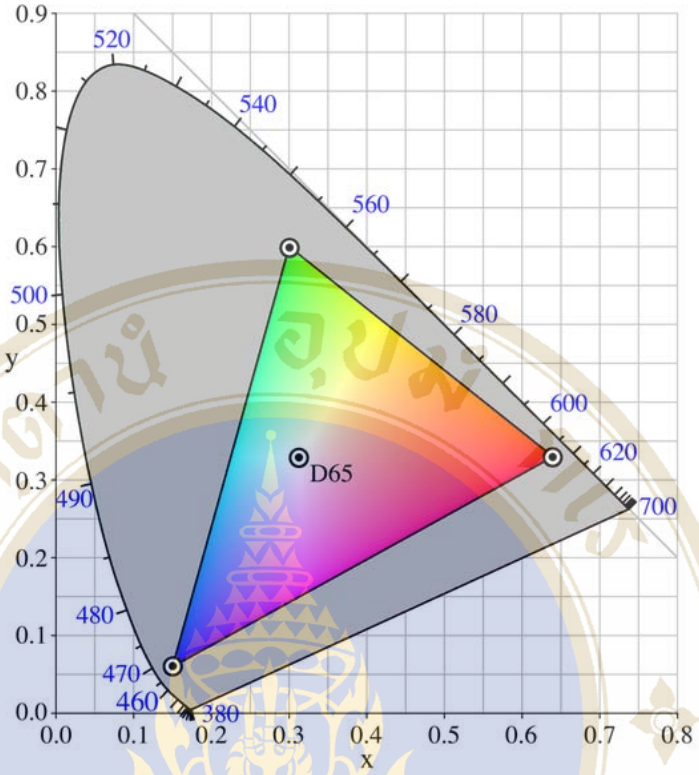


Figure 2.1 A set of primary colors, such as the sRGB primaries, define a color triangle; only colors within this triangle can be reproduced by mixing the primary colors. Colors outside the color triangle are therefore shown here as gray. The primaries and the D65 white point of sRGB are shown.

2.3 HSV Color space

Hue, Saturation and Value [6] are based on the artist concepts of Tint, Shade, and Tone, respectively. The HSV color space is really a cylinder, and not a cone or hexcone as usually pictured. However, the perceived change in color as saturation varies between 0 and 1 is less for dark colors (i.e. ones with a low Value parameter) than for light ones (i.e. ones with a high Value parameter), so the color space is usually distorted to form a cone to help compensate for this perception imbalance.

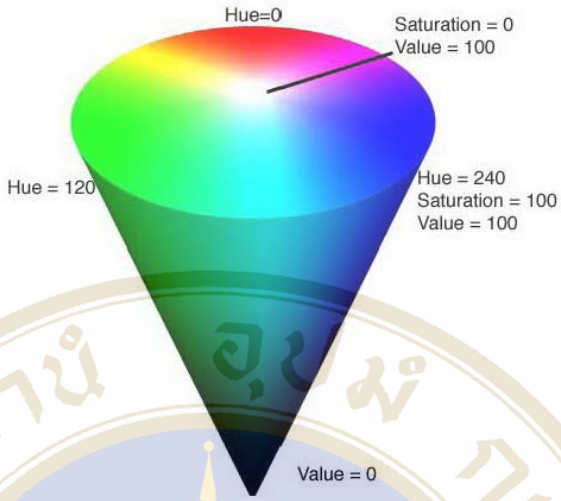


Figure 2.2 HSV Cone

HSV color space, like RGB, is a device-dependent color space, meaning the actual color we see on our monitor depends on what kind of monitor users are using, and what its settings are. One of the first color systems based on polar coordinates and perceptual parameters was Munsell's, which defined the terms: *Hue* is that quality by which we distinguish one color family from another, as red from yellow, or green from blue or purple. *Saturation* is that quality of color by which we distinguish a strong color from a weak one; the degree of departure of a color sensation from that of a white or gray; the intensity of a distinctive hue; color intensity. *Value* is that quality by which we distinguish a light color from a dark one.

When we try to mixing color by mentally varying RGB to some colors. It is not unusual to have difficulty. The HSV model mimics the way an artist mixes paint on his palette: he chooses a pure hue, or pigment, and lightens it to a tint of that hue by adding white, or darkens it to a shade of that hue by adding black, or in general obtains a tone of that hue by adding some mixture of white and black or gray.

2.4 Lab color space

A Lab color space [8] is a color-opponent space with dimension L for lightness and a and b for the color-opponent dimensions, based on nonlinearly-compressed CIE XYZ color space coordinates. The coordinates of the Hunter 1948 L, a, b color space are L, a, and b. However, Lab is now more often used as an informal abbreviation for the CIE 1976 (L*, a*, b*) color space (also called CIELAB, whose coordinates are actually L*, a*, and b*). Thus the initials Lab by themselves are somewhat ambiguous. The color spaces are related in purpose, but differ in implementation.

Both spaces are derived from the "master" space CIE 1931 XYZ color space, which can predict which spectral power distributions will be perceived as the same color (see metamerism), but which is not particularly perceptually uniform. Strongly influenced by the Munsell color system, the intention of both “Lab” color spaces is to create a space which can be computed via simple formulas from the XYZ space, but is more perceptually uniform than XYZ. Perceptually uniform means that a change of the same amount in a color value should produce a change of about the same visual importance. When storing colors in limited precision values, this can improve the reproduction of tones. Both Lab spaces are relative to the white point of the XYZ data they were converted from. Lab values do not define absolute colors unless the white point is also specified. Often, in practice, the white point is assumed to follow a standard and is not explicitly stated (e.g., for "absolute colorimetric" rendering intent ICC L*a*b* values are relative to CIE standard illuminant D50, while they are relative to the unprinted substrate for other rendering intents).

The lightness correlate in CIELAB is calculated using the cube root of the relative luminance, and using the square root in Hunter Lab (an older approximation). Except where data must be compared with existing Hunter L,a,b values, it is recommended that CIELAB be used for new applications.

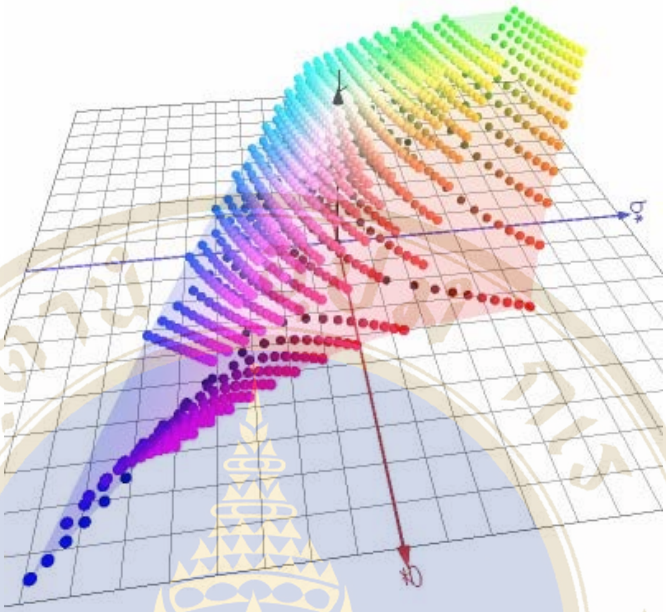


Figure 2.3 The CIE 1976 (*L**, *a**, *b**) color space (CIELAB), showing only colors that fit within the sRGB gamut (and can therefore be displayed on a typical computer display). Each axis of each square ranges from -128 to 128.

CIE XYZ to CIE *L*a*b (CIELAB) and CIELAB to CIE XYZ conversions**

The forward transformation

$$\begin{aligned} L^* &= 116\, f(Y/Y_n) - 16 \\ a^* &= 500\,[f(X/X_n) - f(Y/Y_n)] \\ b^* &= 200\,[f(Y/Y_n) - f(Z/Z_n)] \end{aligned} \tag{4}$$

where

$$f(t) = \begin{cases} t^{1/3} & t > (6/29)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise} \end{cases} \tag{5}$$

Here X_n, Y_n and Z_n are the CIE XYZ tristimulus values of the reference white point (the subscript n suggests "normalized").

The division of the $f(t)$ function into two domains was done to prevent an infinite slope at $t = 0$. $f(t)$ was assumed to be linear below some $t = t_0$, and was assumed to match the $t^{1/3}$ part of the function at t_0 in both value and slope. In other words:

$$t_0^{1/3} = at_0 + b \text{ (match in value)}$$

$$1/(3t_0^{2/3}) = a \quad \text{(match in slope)}$$

The value of b was chosen to be 16/116. The above two equations can be solved for a and t_0:

$$a = 1/(3\delta^2) = 7.787037\cdots$$

$$t_0 = \delta^3 = 0.008856\cdots$$

where $\delta = 6 / 29$. Note that the slope at the join is $b = 16 / 116 = 2\delta / 3$.

2.5 Self-organizing map

A self-organizing map (SOM) or self-organizing feature map (SOFM) [7] is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. Self-organizing maps are different than other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space. This makes SOM useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling. The model was first described as an artificial neural network by the Finnish professor Teuvo Kohonen, and is sometimes called a Kohonen map. Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector.

A self-organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest weight vector to the vector taken from data space and to assign the map coordinates of this node to our vector.

While it is typical to consider this type of network structure as related to feedforward networks where the nodes are visualized as being attached, this type of architecture is fundamentally different in arrangement and motivation. The structure of SOM is shown in figure 2.4.

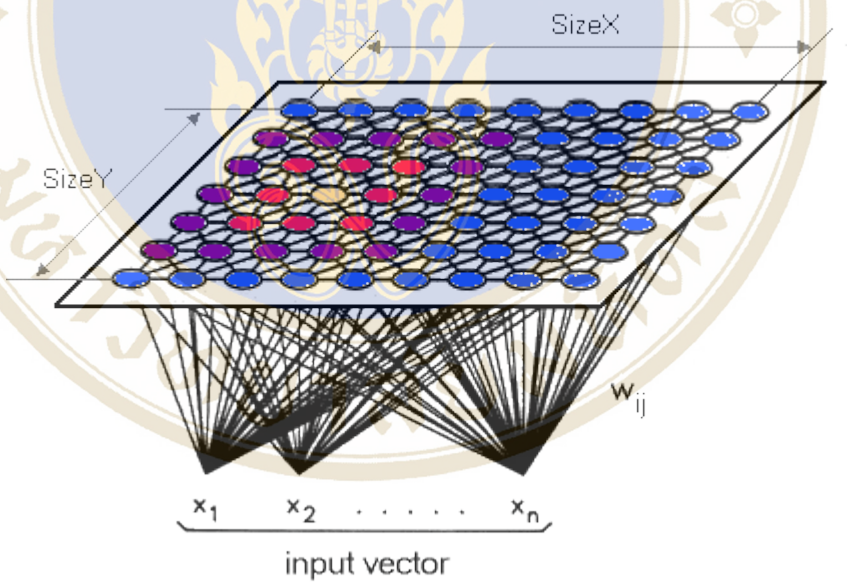


Figure 2.4 The structure of SOM network.

2.5.1 Ordinary SOM Algorithm

The SOM variant we will examine is the Gaussian-neighborhood, Euclidean distance, rectangular topology SOM, given by (6)-(9). The algorithm is, in brief, as follows: An input $x(t)$ is presented to the network at time (or timestep,

iteration) *t*. The 'winning node' $c(t)$, i.e., the node with weight vector that most closely matches the input at time *t*, is selected using (6)

$$c(t) = \arg\min_i(\|x(t) - w_i(t)\|_2) \quad (6)$$

where $w_i(t)$ is the weight vector of node *i* at time *t*. $\|.\|_2$ denotes the L^2-norm or *n*-dimensional Euclidean distance. (The SOM can use other distance measure, e.g., Manhattan distance.) The weights of all nodes are then updated using (9)-(11)

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \quad (7)$$

$$\Delta w_i(t) = \alpha(t) h_{c,i}(t)[x(t) - w_i(t)] \quad (8)$$

$$h_{c,i}(t) = e^{\frac{-d(i,c)^2}{\beta(t)^2}} \quad (9)$$

where $h_{c,i}(t)$ is referred to as the neighborhood function, and is a scaling function centered on the winning node *c* in the node grid. As is the case with the input/weight distance, the node distance can be calculated using some other distance measure than the Euclidean distance, e.g., the Manhattan distance or the link distance, and the grid need not be rectangular. $\alpha(t)$ is the learning rate at time *t*, $\beta(t)$ is the neighborhood size at time *t*.

Lastly, the learning rate (α) and neighborhood size (β) are decreased in accordance with the annealing scheme. One possible annealing scheme is given by (10) and (11) for the decrease of the learning rate and the neighborhood size, respectively. The important point is that the annealing scheme relies on the time step number *t* and not the actual fitness of the network

$$\alpha(t+1) = \alpha(t)\delta_\alpha \qquad 0 < \delta_\alpha < 1 \quad (10)$$

$$\beta(t+1) = \beta(t)\delta_\beta \qquad 0 < \delta_\beta < 1 \quad (11)$$

Here, δ_β and δ_α are scaling constants determined beforehand.

These steps are repeated until some preset condition is met, usually after a given number of iterations or when some measurement of error reaches a

certain level. The density of the nodes in input space are proportional to the density of input samples, however this may lead to undesired results.

2.5.2 The Batch SOM Algorithm (BSOM)

Also batch training algorithm [10] is iterative, but instead of using a single data vector at a time, the whole data set is presented to the map before any adjustments are made hence the name "batch". In each training step, the data set is partitioned according to the Voronoi regions of the map weight vectors, ie. each data vector belongs to the data set of the map unit to which it is closest. After this, the new weight vectors are calculated as:

$$w_i(t+1) = \frac{\sum_{j=1}^{n} h_{i,c}(t) x_j}{\sum_{j=1}^{n} h_{i,c}(t)} \tag{12}$$

where $c = \arg\min_i(\|x(t) - w_i(t)\|_2)$ is the index of the best match unit (BMU) of data sample x_j. The new weight vector is a weighted average of the data samples, where the weight of each data sample is the neighborhood function value $h_{i,c}(t)$ at its BMU c.

2.6 K-Means Algorithm

K-means clustering [9] is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a *d*-dimensional real vector, then *k*-means clustering aims to partition the *n* observations into *k* sets ($k < n$) $\mathbf{S}=\{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) given by (13):

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2 \tag{13}$$

where $\boldsymbol{\mu}_i$ is the mean of S_i.

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the k-means algorithm; it is also referred to as Lloyd's algorithm, particularly in the computer science community.

Given an initial set of k means m1(1),…,mk(1), which may be specified randomly or by some heuristic, the algorithm proceeds by alternating between two steps:
Assignment step: Assign each observation to the cluster with the closest means (i.e. partition the observations according to the Voronoi diagram generated by the means) as given by (14).

$$S_i^{(t)} = \left\{ x_j : \left\| x_j - m_i^{(t)} \right\| \leq \left\| x_j - m_{i^*}^{(t)} \right\| for\ all\ i^* = 1, \ldots, k \right\} \tag{14}$$

Update step: Calculate the new means to be the centroid of the observations in the cluster given by (15)

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j \tag{15}$$

The algorithm is deemed to have converged when the assignments no longer change.

As it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial clusters. As the algorithm is usually very fast, it is common to run it multiple times with different starting conditions. It has been shown that there exist certain point sets on which k-means takes superpolynomial time: $2^{\Omega(\sqrt{n})}$ to converge, but these point sets do not seem to arise in practice.

2.7 Fuzzy C-Means

In hard Clustering or non-fuzzy such as k-means, data is divided into crisp clusters, where each data point belongs to exactly one cluster. In fuzzy clustering, the data points can belong to more than one cluster, and associated with each of the points are membership grades that indicate the degree to which the data points belong to the different clusters. Fuzzy clustering belongs to the group of soft computing techniques. Membership degrees between zero and one are used in fuzzy clustering instead of crisp assignments of the data to clusters. The resulting data partition improves data understanding and reveals its internal structure. Partition cluster algorithms divide up a data set into clusters or classes, where similar data objects are assigned to the same cluster whereas dissimilar data objects should belong to different clusters.

The most prominent algorithm is the FCM or Fuzzy C-Means algorithm. The fuzzy C-means algorithm was proposed as an improvement of the classic Hard C-Means clustering algorithm. The FCM algorithm receives the data or the sample space, an $n \times m$ matrix where n is the number of data and m is the number of parameters. The number of clusters c, the assumption partition matrix U, the convergence value E all must be given to the algorithm. The assumption partition matrix has c number of rows and n number of columns and contains values from 0 to 1. The sum of every column has to be 1. The first step is to calculate the cluster centers. This is a matrix v of dimension c rows with m columns. The second step is to calculate the distance matrix D. The distance matrix constitutes the Euclidean distance between every pixel and every cluster center. This is a matrix with c rows and n columns. From the distance matrix the partition matrix U is calculated. If the difference between the initial partition matrix and calculated partition matrix is greater than the convergence value then the entire process from calculating the cluster centers to the final partition matrix. The final partition matrix is taken and is used for reconstructing the image. Let us assume as a fuzzy C-Means Functional,

$$J_m(U,Y) = \sum_{k=1}^{n}\sum_{j=1}^{c}(u_{jk})E_j(x_k) \quad (16)$$

where $\Omega = \{ x_k \mid k \in [1,c]\}$ is the set of centers of clusters; $E_j(x_k)$ is a dissimilarity measure (distance or cost) between the sample x_k and the center y_j of a specific cluster j; $U = [u_{jk}]$ is the fuzzy c-partition matrix, containing the membership values of all samples in all clusters; E is a control parameter of fuzziness.

The clustering problem can be defined as the minimization of J_m with respect to Y, under the probabilistic constraint:

$$\sum_{j=1}^{c}(u_{jk}) = 1 \quad (17)$$

The Fuzzy C-Means (FCM) algorithm consists in the iteration of the following formulas:

$$y_j = \frac{\sum_{k=1}^{n}(u_{jk})^m x_k}{\sum_{k=1}^{n}(u_{jk})^m} \quad \text{for all j} \quad (18)$$

$$u_{jk} = \begin{cases} \sum_{l=1}^{c}(E_j(x_k))^{2/m-1} \; if \; E_j(x_k) > 0 & \forall j,k \\ 1 \; if \; E_j(x_k) = 0 \;\; and \; u_{jk} = 0 & \forall l \neq j \end{cases} \quad (19)$$

Where in the case of the Euclidean space:

$$E_j(x_k) = \|x_k - y_j\|^2 \quad (20)$$

It is worth nothing that if one chooses $m = 1$ the fuzzy C-Means Functional J_m (Eq.16) reduces to the expectation of the global error (which we denote as $<E>$):

$$< E >= \sum_{k=1}^{n}\sum_{j=1}^{c}(u_{jk})E_j(x_k) \quad (21)$$

and the FCM algorithm becomes the classic Hard C-Means algorithm.

2.8 Related researches

J. Moreira, L.F. Costa [11] presented a segmentation method for color images based on the chromaticities of the objects. The segmentation makes use of a SOM, which was able to discriminate the main chromaticities of the image. Sample points are taken from the image (already normalized) and submitted to the network in its learning phase. When the feature map has been formed, i.e., tended to a stable state, the present features (the main chromaticities) are clustered to detect not only the number of significant clusters but also their centers. Then it is possible to classify the pixels in the image assigning them to the closest cluster center. The final segmentation is composed by the classified pixels, which were assigned to one of the determined classes. The results substantiate the feasibility of the method, which is able de detect the classes in a coherent manner. Also, the mismatch error measures allowed to evaluate the method as presenting a reasonable reliability.

Y. Jiang et al [12] proposed a method of SOM based image segmentation that use a set of five-dimension vectors to represent an image, which is fed to a SOM network. After the clustering result is gained, isolated pixels are eliminated and scattered blocks are merged. Their experiments show that this method can get preferable segmentation result.

Y. Jiang et al [13] proposed an SOM ensemble based image segmentation through adopting a scheme for aligning different clusterers, their method utilize the power of ensemble learning paradigms and achieves robust image segmentation performance. Experimental results show that the proposed method works better than the image segmentation methods based on either single SOM neural network or *k*-means clustering. A limitation of their proposed method is that the number of regions to be segmented out should be manually set since it follow Y. Jiang et al.'s framework [8]. It is evident that in order to improve the utility, their proposed method should be facilitated with mechanisms that could adaptively determine the appropriate number of regions to be segmented out.

M. Persson et al [14] describes a system for automatic detection of buildings in aerial images taken from a nadir view. The system builds two types of independent hypotheses based on the image contents. A segmentation process implemented as an ensemble of SOMs is trained and used to create a segmented image showing different types of roofs, vegetation and sea. A second type of hypotheses is based on an edge image produced from the aerial photo. A line extraction process uses the edge image as input and extracts line from it. From these edges, corners and rectangles that represent buildings are constructed. A classification process uses the information from both hypotheses to determine whether the rectangles are buildings, unsure buildings or unknown objects. The overall detection rate for the buildings is fairly good with 53% correctly found areas, of which 93% was classified correctly. The unsure buildings were left out.

G. Dong et al [15] proposed an image segmentation system for the segmentation of color image. They represented images color in a modified *CIE Luv* color space in order to measure the color difference properly and the unsupervised segmentation is achieved by a two-level approach, i.e., color reduction and color clustering. In color clustering, they applied simulated annealing (SA) to seek the optimal clusters from SOM prototypes. Their method is able to produce the near optimal segmentation result with lower computational cost than seeking the color clusters in the original image data.

M. Awad et al [16] proposed a new multicomponent image segmentation method using SOM and hybrid genetic algorithm (HGA). They used SOM to detect the main features that are representing in the image; then, HGA is used to cluster the image into homogeneous region without any a priori knowledge. Their experiments are performed on different satellite images and the results are compared with ISODATA, the results show that their SOM-HGA is more robust and efficient than ISODATA. The overall average efficiency of SOM-HGA is greater than 90% while ISODATA accuracy is less than 72%. The drawback of SOM-HGA is that it takes five times slower than ISODATA.

X. Zhang et al [17] applied SOM to cluster the color data. From their result, it is clear that SOM can not only cluster the color data very well, but also efficiently and effectively reveal the appropriate number of classes and hierarchical relations in the data sets.

CHAPTER III
RESEARCH METHODOLOGY

3.1 Research Methodology

A complete aerial image quantization system is developed for both ordinary SOM and Batch SOM in order to compare the performance of both techniques in color reduction. Additionally, we also compare the performance in color clustering of *K*-means. Figure 3.1 shows the structure of the system.

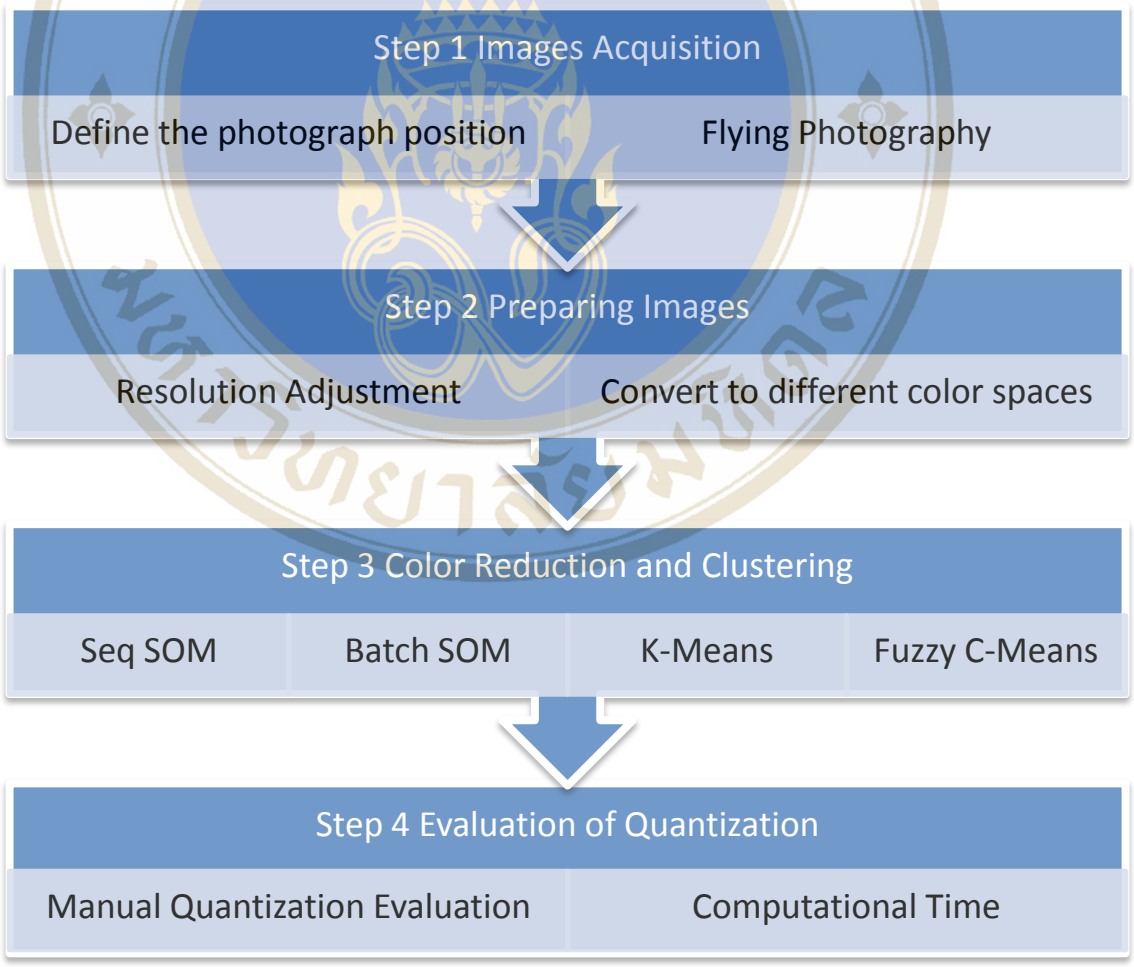


Figure 3.1 Research methodology

3.1.1 Images acquisition

3.1.1.1 Define the photograph position

In order to get the photograph that covers all area, we have to first define the right photograph position. An essential information for calculate the photograph point are (1) A focus length of camera (2) A Height of flying. The right analysis of photograph position can do in figure 3.2:

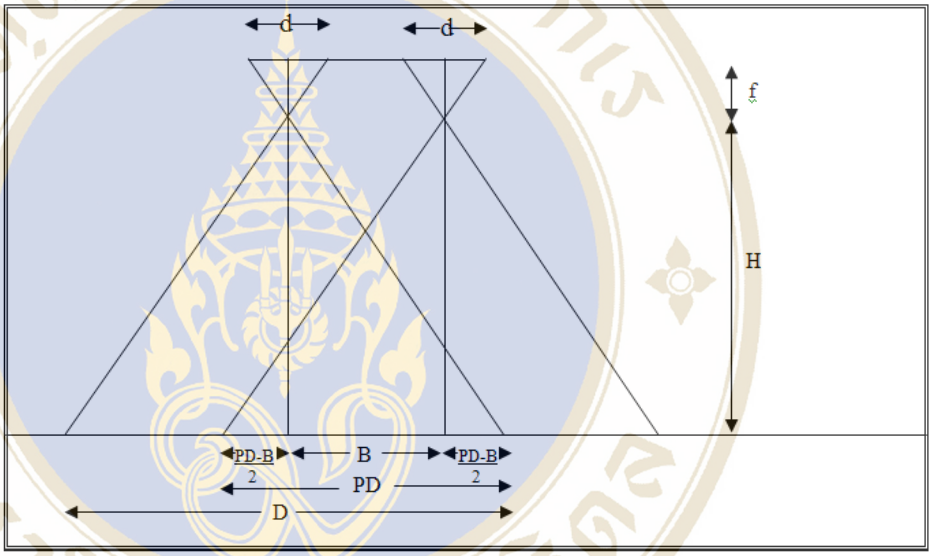


Figure 3.2 The analysis of photograph position

$$B = \frac{dH}{f}(1-p)$$

d = side length of aerial photograph (mm)

D = side length of area that photograph cover (m)

f = focus length of camera (mm)

H = height of flying (m)

P = ratio of photograph overlapped

B = distance between two photograph points (m) called overlap distance

For example, we fly to take the photograph at 900 m height with camera which focus length 447.675 mm, size of photograph is 3264×2448 pixels with 0.127 mm pixel size and the overlapped of photograph is 60%. According to above formula we can calculate the distance between two photograph points (overlap) as

$$(3264\times0.127)\times900\times (1-0.6)/447.675 = 333 \text{ m}$$

Besides from overlap, the above formula also use for calculate flying line (side lap). The same example, if we desire the side lap at 10% we can calculate side lap as

$$(2448\times0.127)\times900\times (1-0.1)/447.675 = 562 \text{ m}$$

When we know the photograph points we then define that points on map as show in figure 3.3 in order to pilot can fly correctly. From map ratio 1:50000 of interested area. We define flying points with flying line separate at 562 m and photograph points separate at 333 m.

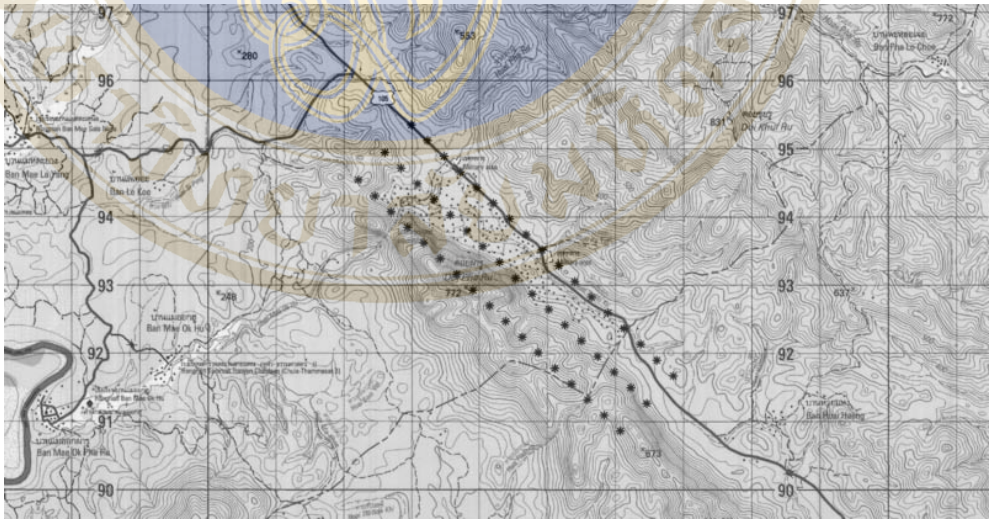


Figure 3.3 Photograph points

For the convenience of pilot to know what is the right photograph point. All coordinates of photograph points as show in above figure have to set to mobile GPS devices as show in figure 3.4 and bring them with pilot.

Figure 3.4 The mobile GPS devices

3.1.1.2 Flying Photography

This research uses the small plane as a vehicle for flying photography. The advantage of small plane instead of personal plane or flying brolly is cost-effective. Nowadays a lot of flying association has been widespread make it easy to uses the small plane for photography. So in this research we use the small plane as show in the following figure 3.5.

Figure 3.5 The small plane as a vehicle for flying photography

In general purpose of aerial photography, we prefer to use metric camera which design especially for aerial photography. This camera has the component that can define fiducial marks for conveniently measure of coordinate values beside the camera also has a high resolution. Disadvantage of the camera is the high price. In this research then we use digital camera instead and reduce discrepancy that may be occur just as the discrepancy that occur from lens distortion. The digital camera we use in this research can photography at 8 mega pixel resolutions as we show in figure 3.6:

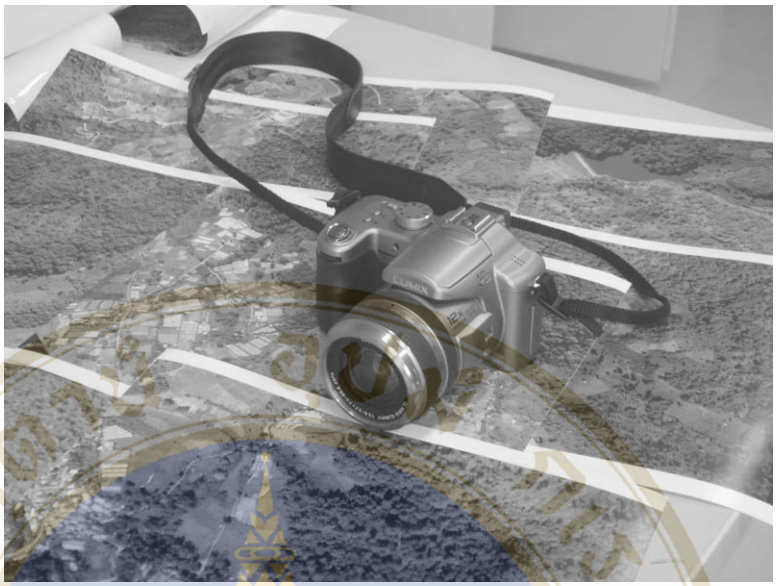
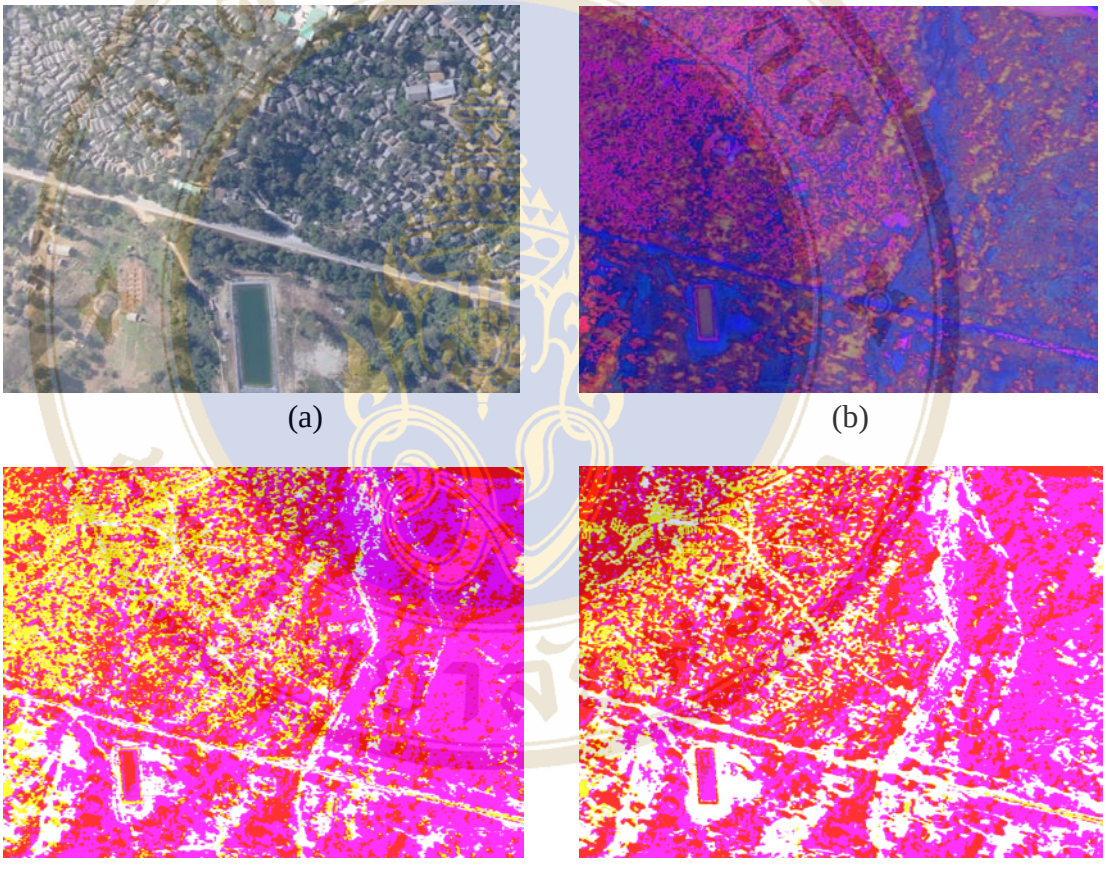


Figure 3.6 Digital Camera used for aerial photography

The results we get are color photograph which size 3264×2448 pixels at 200 dpi or the size of each pixel is 127 microns. We photograph at altitude 1150 m above the seawater level or 900 m above the mean ground level. The result is show in appendix A.

3.1.2 Preparing Images

For training and testing images, we use the digital images directly take from Panasonic LUMIX DMC-FZ30. Then, we adjust the photographs to appropriate resolution that is 25% of 3264×2448 pixel original size of photographs in order to reduce the computational time that spend in color clustering by SOM. After that we convert the photographs color space from RGB to HSV and CIELAB color space before feed to the algorithms in order to get a better perceptual uniformity.



(a) (b)

(c) (d)

Figure 3.7 Aerial images (a) Before convert to other color space (b) After convert to the HSV color space (c) After convert to the CIELAB color space (d) After convert to the CIELUV color space

Figure 3.8 The different colors represent the different object
From left to right, in the top line, Buildings, Roads and Land
In the bottom line, Land with grasses and Forest

3.1.3 Color Reduction

The color reduction maps a given set of 3-D color points into a surrogate lower dimensional space. Let $\Re^3$ be a 3-D color space, it carries out a projection of color points from color space $\Re^3$ into a two-dimensional (2-D) color space $\Re^2$ by the transformation *T*

$$X \in \Re^3 \overset{T}{\Rightarrow} Y \in \Re^2,\ d < 3 \tag{18}$$

Given a set of color points $X = \{x_1, x_2, \ldots, x_n\}$ in RGB color space $\Re^3_{rgb}$, it is trained to map X onto a 2-D array of nodes M such that color points neighboring in $\Re^3_{rgb}$ are projected to nearby nodes in M.

The SOM is structured as two-layer neural network with a rectangular topology as shown in figure 3.4. Three inputs (H, S and V) are fully connected to the neurons on a 2-D plane. Each neurons i is a cell containing a template against which inputs are matched. The template is the weight values to the neuron i, which is represented by $w_i = [w_{i1}, w_{i2}, w_{i3}]^T$.

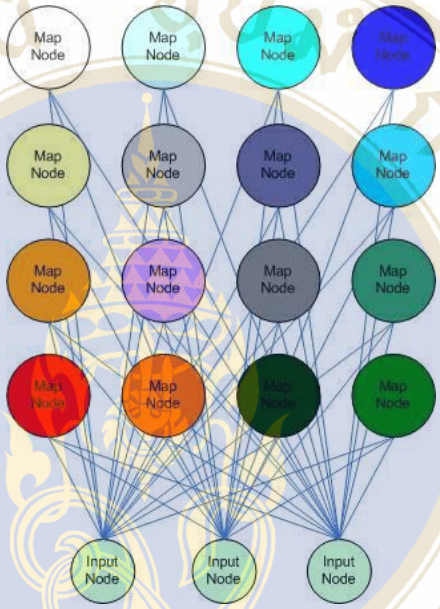


Figure 3.9 Example of SOM structure.

The Ordinary SOM training has the following procedure.

1. Initialization: Define the SOM map. The weight vector $w_i(0)$ of ith neuron is randomly initialized. We set the size of SOM map to 3 different sizes that is 8×8, 16×16 and 32×32 to find the best one. The topology type is rectangular. The neighborhood type is Gaussian. The SOM training is successively performed by two phases. The neighborhood radius r = 16, 5, and learning rate α = 0.05, 0.02. The weight vectors of the map are ordered in the first phase and fine-tuned in the second phase.
2. Input: The 12 aerial images are reiteratively train the network. During training, each color point X (H, S, and V) is cyclically

chosen from the data set, and fed to all neurons on the map simultaneously.

3. Competitive Process: At time *t*, color point $X(t) = [h(t), s(t), v(t)]^T$ is fed into the network. The winning neuron *c* is computed with the shortest Euclidean distance between color point and weight vectors by (5) in section 2.3.1.
4. Cooperative Process: The topological neighbors of 'winning neuron' *c* are determined by Gaussian function centered at neuron *c* with effective scope of $\beta(t)$.
5. Adaptive Process: The weights of 'winning neuron' *c* and its neighbor neurons are updated within the neighborhood by (6)-(8) in section 2.3.1.
6. Iteration: The next color point is fed into the network at time $t + 1$. According to [12], the learning rate α is linearly decreased to $\alpha(t+1) = \alpha(0)(1.0 - t/T)$. The neighborhood radius β is $\beta(t+1) = \beta(0)(2.0 - t/T)$, where *T* is the number of color points for training. The new 'winning neuron' is chosen by repeating the procedure from step 2 until all iterations have been made $(t = T, \alpha = 0, R = 1)$.

After the training process, all color points are presented to the SOM map, and then we assign the index values for every node in SOM map, for example the 8×8 map size, we can assign number 1-64 to each node and then assign these index to the pixel data. This process drastically reduces the computational cost from 3-D dimensional to 1-D dimensional for color clustering process; each color point is characterized by one neuron.

3.1.4 Color clustering for Image Quantization

The SOM clustering is very similar to SOM for color reduction but has some changes, has the following procedure.

1. Initialization: Define the SOM map as one-dimensional network. The weight vector $w_i(0)$ of *i*th neuron is randomly initialized. We set the size of SOM map to 3×3 which small enough to separate color of objects (the others such as 2×2, 3×2 and 2×3 are not work well from trial and error test). The topology type is linear. The neighborhood of radius is 1 and learning rate α = 0.05.
2. Input: The 12 aerial images that already performed color reduction are reiteratively used to train the network for few times. During training, each color point X is cyclically chosen from the data set, and fed to all neurons on the map simultaneously.
3. Competitive Process: At time *t*, color point $X(t)$ is fed to the network. The winning neuron *c* is computed with the shortest distance between color point and weight vectors by (5) in section 2.3.1.
4. Cooperative Process: The topological neighbors of 'winning neuron' *c* are determined.
5. Adaptive Process: The weights of 'winning neuron' *c* and its neighbor neurons are updated within the neighborhood by (6)-(8) in section 2.3.1.
6. Iteration: This process does the same as Ordinary SOM for color reduction.
7. Finally we obtain the color reduced image from the algorithm but the result contain 9 color and since we want to represent only 4 objects that are Building, Road, Grass and so we must reduce to 4 colors in order to identify the objects clearly. First, we pick the color of each object as the color reference, then we compare every color pixel with the color reference and set pixel to the most similar color with the minimum of Euclidean distance criterion.

The algorithm is implemented in MATLAB 7 environment with below code:

BLSOM Image Quantization

```
function [sMap,RES,DSOM2] = som_imsegment(rgb,size1,size2)

% This function use for image segmentation by The Batch
Learning Self-Organizing Map with size1 represent the first
size of SOM and size2 represent the last size of SOM

tic  % start count time

D = rgb2data(rgb);

% Convert the image to HSV or Lab color space and reduce to 2
Dimensional Matrix

D = som_normalize(D,'var');

% Normalize Data

sMap =
som_make(D,'msize',size1,'lattice','hexa','shape','sheet','neig
h','gaussian');

% Train The First SOM Map with essential parameters

[Bmus,Qerrors] = som_bmus(sMap,D);

% Find the first best match units and corresponding
quantization errors

DSOM = my_map2som(Bmus,sMap.codebook,D);

% Mapping data to best match units

sMapfinal =
som_make(DSOM,'msize',size2,'lattice','hexa','shape','sheet','n
eigh','gaussian');

% Train the Second SOM Map

[Bmus2,Qerrors2] = som_bmus(sMapfinal,DSOM);

% Find the second best match units and corresponding
quantization errors

DSOM2 = my_map2som(Bmus2,sMapfinal.codebook,DSOM);

% Mapping data to best match units

RES = my_data2hsv(DSOM2);

figure, imshow(RES);

% Show the result

toc % stop count time
```

Sequential SOM Image Quantization

```
function sMap = som_imsegment2(rgb,size1,size2)

% This function use for image segmentation by The Sequential
Self-Organizing Map

tic

D = rgb2data(rgb);

% Convert the image to HSV color space and reduce it to 2
Dimensional Matrix

D = som_normalize(D,'var');

% Normalize Data

sMap =
som_make(D,'algorithm','seq','msize',size1,'lattice','hexa','sh
ape','sheet','neigh','gaussian');

% Train The First SOM Map with essential parameters

 [Bmus,Qerrors] = som_bmus(sMap,D);

% Find the first best match units and corresponding
quantization errors

DSOM = my_map2som(Bmus,sMap.codebook,D);

% Mapping data to best match units

sMapfinal =
som_make(DSOM,'algorithm','seq','msize',size2,'lattice','hexa',
'shape','sheet','neigh','gaussian');

% Train The Second SOM Map

[Bmus2,Qerrors2] = som_bmus(sMapfinal,DSOM);

% Find the second best match units and corresponding
quantization errors

DSOM2 = my_map2som(Bmus2,sMapfinal.codebook,DSOM);

RGB2 = my_data2hsv(DSOM2);

figure, imshow(RGB2)

% Show the result

toc
```

Note: Since the function code is ponderous then we excluded.

We used SOM 2 times because we trial with 1 time SOM and found that the result cannot clearly separate the color of objects and 3 times is cost too much of computational time.

3.1.5 Image Color Quantization with K-Means

1. *k* initial "means" (in this case k=9) are randomly selected from the image pixel.
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.
3. The centroid of each of the *k* clusters becomes the new means.
4. Steps 2 and 3 are repeated until convergence has been reached.

3.1.6 Image Color Quantization with Fuzzy C-Means

1. Receiving the image data matrix x
2. Fixing number of clusters, c ($2 \leq c \leq n$), n is the length of the image data
3. Assuming the partition matrix, U(r)
4. Calculating the cluster centers, V using (18)
5. Calculating the Euclidean distance matrix, D using (20)
6. Updating the partition matrix, U(r) using equation (19)
7. Checking for convergence
   if max $|U(r) - U(r+1)| < E$ stop else repeat steps 3-6

3.1.7 Evaluation of the Color Quantization

In order to evaluate the performance, both automatic and manual evaluation techniques must be performed.

3.1.7.1 Evaluation with manual quantization

We ask the expert to label the right class of each components and perform manual image segmentation by determine the same objects to the same color then apply the color with the selection and paint tools

in Adobe Photoshop CS4. The result then compare with the results from SOM, BSOM and K-Means image color clustering pixel by pixel and count for the match then evaluates the correct percent.

3.1.7.2 Computational time

We compare the computational time of all techniques: SOM, Batch SOM and K-Means

3.1.7.3 Quantization error (QE) and Topographic error (TE)

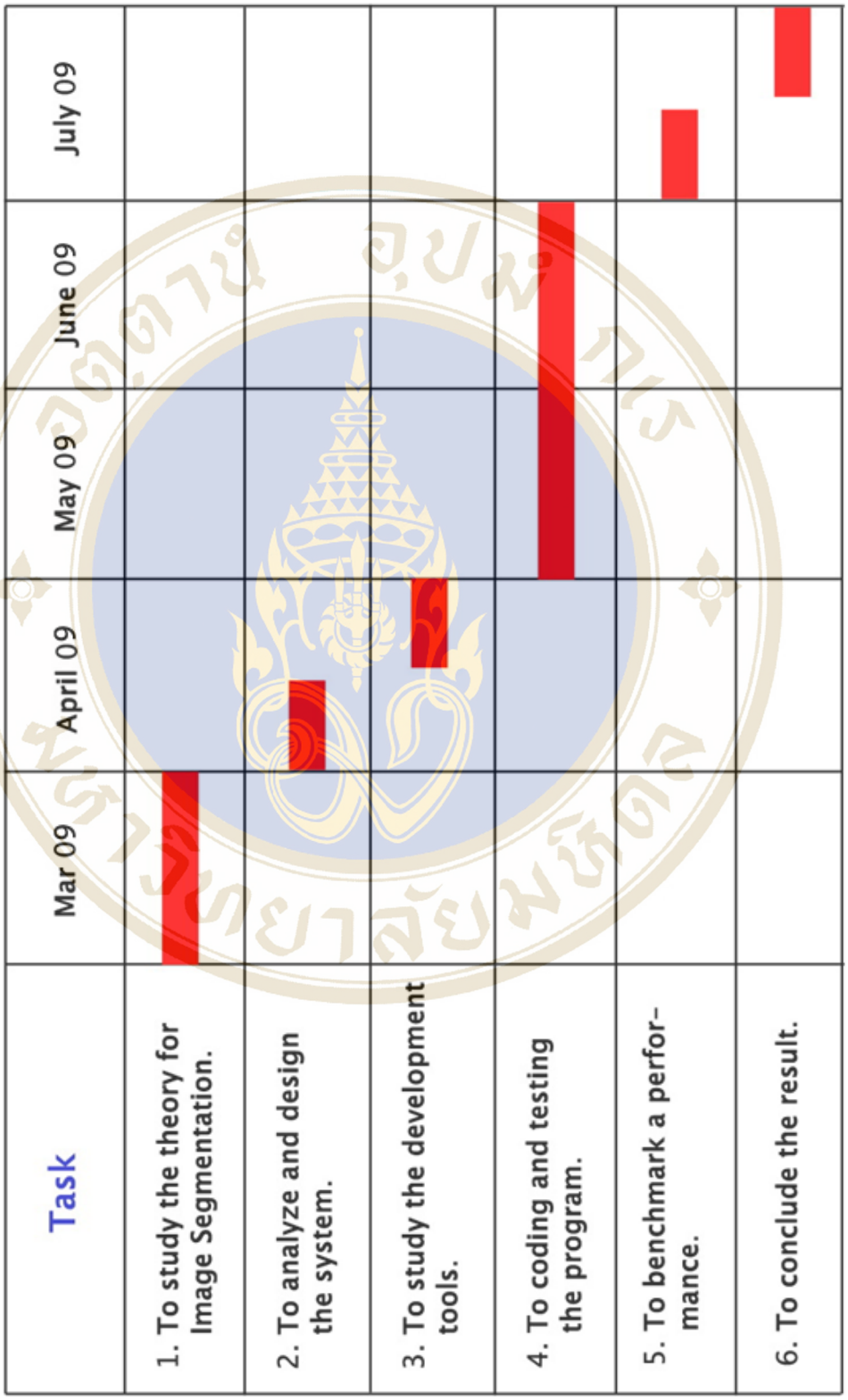
We measure the quality of the SOM map. The measures are data-dependent: they measure the map in terms of the given data. Typically, the quality of the map is measured in terms of the training data. The returned quality measures are average quantization error and topographic error. The issue of SOM quality is a complicated one. Typically two evaluation criteria's are used: resolution and topology preservation. There are many ways to measure them. The ones implemented here were chosen for their simplicity.

Quantization error (QE) is average distance between each data vector and its best match unit (BMU). Measures map resolution. Topographic error (TE) is the proportion of all data vectors for which first and second BMUs are not adjacent units. TE measures topology preservation.

3.2 Materials

- Aerial Images
- Hardware: The specification of hardware that used for image segmentation is consists of these following
    - CPU Intel Core2Duo E7200 OC@3.60 GHz
    - RAM DDR2-800 4.00 GB Dual Channel
    - Hard disk Seagate 500 GB SATAII 16 MB Buffer
- Development Tools
    - Operating System: Microsoft Windows 7 Ultimate x64
    - MATLAB 7.6 (R2008a)
    - Adobe Photoshop CS4

3.3 Research Schedule



CHAPTER IV
RESULTS

4.1 Algorithms and the size of SOM map

Firstly, and very important step, the SOM map size we choose is examined from the computational time and evaluation criterion; average Quantization error and Topographic error. The results clearly indicate that 8×8 SOM map size with batch training is the best choice of our work since the objective of this work is to examine the technique that compromise between the cost of time and the performance of color clustering.

Table 4.1 Computational time (CT) of BLSOM and SOM

SOM map size	Batch CT (seconds)	Seq CT (seconds)
8×8	11.54864	95.977629
16×16	28.158227	117.882711
32×32	114.835733	227.318496

Table 4.2 Quantization error (QE) and Topographic error (TE) at 32×32, 16×16 and 8×8 BLSOM map size

SOM map size	QE1	QE2	TE1	TE2
32×32	0.266	0.633	0.633	0.000
16×16	0.314	0.846	0.077	0.000
8×8	0.436	0.745	0.055	0.000

Table 4.3 Quantization error (QE) and Topographic error (TE) at 32×32, 16×16 and 8×8 SEQSOM map size

SOM map size	QE1	QE2	TE1	TE2
32×32	0.266	0.633	0.633	0.000
16×16	0.314	0.846	0.077	0.000
8×8	0.436	0.745	0.055	0.000

4.2 The result from applied the algorithm and map size to aerial photographs

The result of color image quantization is shown in Appendix B. Table 4.4 and Table 4.5 show the computational time (CT) and quantization error (QE) ,topographic error (TE) respectively and Figure 4.1 and 4.2 shows the result of color clustering for aerial image color clustering from chosen algorithm that is batch SOM algorithm with 8×8 SOM map size (we do not choose sequential SOM since the section 5.1 indicate that technique is not the good choice in this study) including K-Means and Fuzzy C-Means algorithm with the parameters that gave the good result. Table 4.6 show the numbers of iteration that make the centroids no longer move.

Table 4.4 Computational time (CT)

The number of aerial photos	Batch SOM CT (seconds)	K-Means CT (seconds)	FCM CT (seconds)
1	11.54864	44.45465	32.136610
2	11.51348	30.05392	40.242768
3	11.53482	26.63650	26.402302
4	11.59529	37.85836	30.617224
5	11.67961	27.04682	40.599922
6	11.53025	24.10086	41.898909
7	11.48852	33.60315	37.914750
8	11.53542	31.16347	40.743076
9	11.50315	35.94795	38.443402
10	11.51131	28.24108	21.121926
11	11.51583	28.54916	32.979598
12	11.50503	49.68306	40.283970
Average	**11.53845**	**33.11158**	**35.282038**
S.D.	**0.052293**	**7.696992**	**6.6384095**

Table 4.5 Quantization error (QE) and Topographic error (TE) of BLSOM

Aerial photos	QE1	QE2	TE1	TE2
1	0.436	0.745	0.055	0.000
2	0.391	0.782	0.087	0.000
3	0.399	0.754	0.022	0.000
4	0.423	0.758	0.013	0.000
5	0.398	0.730	0.020	0.000
6	0.410	0.722	0.007	0.000
7	0.409	0.753	0.020	0.000
8	0.406	0.747	0.025	0.000
9	0.391	0.760	0.019	0.000
10	0.353	0.698	0.043	0.000
11	0.427	0.707	0.019	0.000
12	0.446	0.673	0.014	0.000
Average	**0.407**	**0.736**	**0.029**	**0.000**
S.D.	**0.025**	**0.031**	**0.023**	**0.000**

Table 4.6 The numbers of iteration in K-Means that make the centroids no longer move.

Aerial photos	iter 1	iter 2	iter 3	iter 4	iter 5	S.D.
1	26	15	48	26	34	12.21475
2	21	23	53	32	33	12.68069
3	17	21	28	28	25	4.764452
4	25	30	18	33	32	6.188699
5	42	15	45	27	50	14.44645
6	15	24	13	35	40	11.92896
7	17	33	20	25	17	6.76757
8	26	30	29	19	26	4.301163
9	22	20	38	22	19	7.823043
10	20	20	15	10	13	4.393177
11	26	40	18	41	42	10.80740
12	29	53	44	25	21	13.55729
Average	**24**	**27**	**31**	**27**	**29**	**3.830285**

Table 4.7 The result of converting image to HSV color space and applied color quantization with BLSOM

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	50564	11275	9780	1570	69.08688464
Roads & Land	27132	278988	9110	22278	82.66115174
Grasses	7982	18004	22379	8014	39.69385764
Forest	9633	27854	16012	14121	20.88287489
Reliability	53.051589	83.0022522	39.0688012	30.7091751	

An average accuracy = 53.08119223, an average Reliability = 51.45795438.

Table 4.8 The result of converting image to HSV color space and applied color quantization with SEQSOM

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	61899	13211	7911	3148	71.83441841
Roads & Land	28539	240221	11739	14223	81.50765806
Grasses	7950	22788	19877	6311	34.91726101
Forest	10913	22153	18557	12785	19.85001863
Reliability	56.6316868	80.5103009	34.221128	35.0590945	

An average accuracy = 52.02733903, an average Reliability = 51.60555255.

Table 4.9 The result of converting image to HSV color space and applied color quantization with K-MEANS

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	29863	15463	6936	3107	53.93451209
Roads & Land	41689	177448	27845	11568	68.63198608
Grasses	8006	18744	16883	5089	34.65169739
Forest	7877	28735	25994	18227	22.54895896
Reliability	34.1545148	73.8167145	21.7401942	47.9771525	

An average accuracy = 44.94178863, an average Reliability = 44.422144.

Table 4.10 The result of converting image to HSV color space and applied color quantization with FCM

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	30122	16101	6339	2755	54.45342300
Roads & Land	42748	181441	28966	9787	69.00419104
Grasses	6234	20133	14789	4012	32.74220687
Forest	6997	30144	16483	13559	20.18218895
Reliability	34.984495	73.2151288	22.213377	45.0270647	

An average accuracy = 44.09550247, an average Reliability = 43.86001638.

Table 4.11 The result of converting image to Lab color space and applied color quantization with BLSOM

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	47851	30132	9870	2285	53.08637866
Roads & Land	90380	169754	23739	49223	50.96248529
Grasses	8340	13022	14327	17761	26.80449018
Forest	8833	20023	19344	21270	30.61753275
Reliability	30.7913567	72.8773757	21.2945898	23.4926385	

An average accuracy = 40.36772172, an average Reliability = 37.11399018.

Table 4.12 The result of converting image to Lab color space and applied color quantization with SEQSOM

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	40113	23537	7355	2310	54.71322376
Roads & Land	72539	169754	21739	74223	50.1852153
Grasses	7950	17332	17444	18001	28.72527871
Forest	10304	25023	14260	26630	34.93971161
Reliability	30.6425985	72.0377176	28.6917333	21.9784755	

An average accuracy = 42.14085735, an average Reliability = 38.33763123.

Table 4.13 The result of converting image to Lab color space and applied color quantization with K-MEANS

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	40350	22530	12705	4889	50.14041802
Roads & Land	70312	136788	27034	50044	48.13461985
Grasses	9910	14172	10018	16166	19.92997255
Forest	6833	31120	19935	32777	36.1517675
Reliability	31.6706566	66.8530375	14.3746772	31.5539682	

An average accuracy = 48.30837369, an average Reliability = 37.11399018.

Table 4.14 The result of converting image to Lab color space and applied color quantization with FCM

	Buildings	Roads & Land	Grasses	Forest	**Accuracy**
Buildings	30158	15473	10144	3017	51.29609471
Roads & Land	53112	144564	24789	49886	53.08003275
Grasses	9636	14556	11774	9888	25.67714921
Forest	5985	28456	27846	36369	36.86445832
Reliability	30.4962029	71.1966077	15.7927917	36.6770875	

An average accuracy = 41.72943375, an average Reliability = 38.54067245.

CHAPTER V
DISCUSSION

5.1 Comparison between algorithms

In algorithm comparison, we consider the result of each algorithm in the same color space. Since the SOM is a new, effective software tool for the visualization of high-dimensional data. It converts complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display we found that Batch Learning SOM algorithm (BLSOM) and Sequential SOM algorithm both provides us the best outcome with no significant difference. Beside, FCM is not very much better than K-Means. We can say that SOM algorithm beats FCM and K-Means in this research. BLSOM, another remark concerns faster algorithms. The incremental regression process defined by equations (8) in chapter 2 can often be replaced by the batch computation version which, especially with Matlab functions, is significantly faster. The computational time between BLSOM and Sequential SOM is significant difference in every SOM map size especially in 8×8 map size the BLSOM computational time is less than Sequential SOM approximately 9 times with no significant difference in performance and that is the reason we choose BLSOM in the next step and found that in MATLAB environment BLSOM is outperform SEQSOM. Next, we consider about relationship between Quantization error (QE) and Topographic error (TE) and SOM map size and we found that 32×32 SOM map size is the best (take less time) in Quantization error (QE) and the worst (take more time) in Topographic error (TE) but we found that these two measures is not significant related with performance of algorithm. From those reason then we chose 8×8 map size BLSOM in the next step. Finally we compare computational time of 8×8 map size BLSOM with K-Means and FCM algorithms and the result shown that 8×8 map size BLSOM outperforms both K-Means and FCM algorithms with less computational time approximately 3 times and much less in standard deviation (S.D.) value that means BLSOM is very consistence on implement.

5.2 Comparison between color spaces

In color space comparison, we consider the result separately by algorithm. In every algorithm, the HSV color space gives the satisfied result as well as the Lab color space. But the HSV color space is significantly more precise in separate the road from the buildings than the Lab space but the last is more precise in separate the grass from the forest but not significant so we can say that the HSV color space is better than the Lab color space in this research.

In the future research we may choose the different kind of aerial photo to see the change of performance when input changed. In the color space context we may separate the L value from a and b values in the Lab color space to see when lightness is excluded to the change in performance. Beside these we may combine two or more algorithms together to see the computational time and performance of combined algorithm.

CHAPTER VI
CONCLUSION

This research is related with the construct of digital map by separates the components according to pixel colors which we can separate them to layers for digital vector map that many GIS applications are depend on. We have proposed a two-step SOM-based network that combines the advantage of SOM-based unsupervised learning neural network with Batch Learning SOM algorithm and change image color space from RGB to HSV and Lab color space with compromise between computational time and performance of results. The results show that this technique has potentiality in real-time application of aerial image color clustering since the average time cost is only about 11 seconds (compared with k-means and FCM that take about 33 and 35 seconds respectively and both have high standard deviation) with about 50% in average accuracy and reliability and up to 82% accuracy in separate road and land from others components when applied HSV with BLSOM and SEQSOM. However, the drawback of technique using pixel for classification such as the pixels of different objects may have the same color that could lead to the false classification. The shape detection could improve the results but may trade-off with the computational time. In order to improve the work in future study the others type of SOM or the others clustering techniques could be compared and the more appropriate evaluation is the good one for measure that the SOM can cluster the color as the same way the human do.

REFERENCE

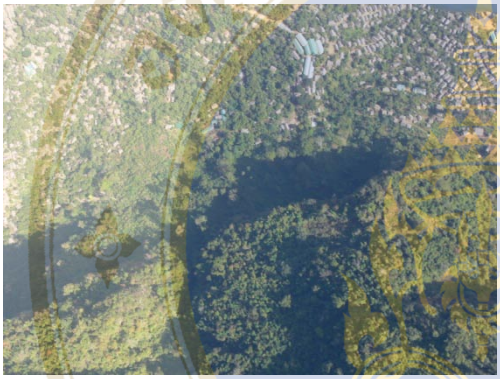
1. M, Pidwirny. "Introduction to Maps". Fundamentals of Physical Geography, 2nd Edition. 2006.
2. M. Neubert, H. Herold, G. Meinel. “Evaluation of Remote Sensing Image Segmentation Quality – Further Results and Concepts” In *Proceedings of the ISPRS Joint Workshop High Resolution*, 2006.
3. H.D. Cheng, X.H. Jiang, Y. Sun, J. Wang. “Color Image Segmentation: Advances and Prospects” *Pattern Recognition*, Vol.34, No.12, Dec. 2001. pp.2259-2281.
4. C. Lai, C. Chang. “A Hierarchical Genetic Algorithm Based Approach for Image Segmentation” In *Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control*, Mar. 2004, pp.1284-1287.
5. Y. Xu, V. Olman, E.C. Uberbacher, “A segmentation algorithm for noisy images: Design and evaluation” *Pattern Recognition Letters*, vol.19, pp.1213.
6. R.C. Gonzalez, R.E. Woods. “Digital Image Processing” Prentice-Hall, 2002.
7. T. Kohonen, “The self-organizing map” Proc. IEEE, vol.78, no.9, pp.1464–1480, Sep. 1990.
8. G. Hoffmann “CIELab Color Space”, 2008.
9. J. A. Hartigan, M. A. Wong. "A K-Means Clustering Algorithm" Applied Statistics 28 (1): 100–108, 1979.
10. J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas "SOM Toolbox for MATLAB 5", April 2000.
11. J. Moreira, L.F. Costa. “Neural-based color image segmentation and classification using self-organizing maps” In Anais do IX SIBGRAPI, 1996, pp.47–54.
12. Y. Jiang, K.J. Chen, Z.H. Zhou. “SOM Based Image Segmentation” Lecture Notes in Artificial Intelligence 2639, 2003, pp.640–643.

13. Y. Jiang, Z.H. Zhou. “SOM Ensemble-Based Image Segmentation” Neural Processing Letters Volume 20, Issue 3, Nov. 2004, pp.171–178.
14. M. Persson, M. Sandvall, T. Duckett. “Automatic Building Detection from Aerial Images for Mobile Robot Mapping” In Proceedings 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp.273–278, June 2005.
15. G. Dong, M.Xie. “Color Clustering and Learning for Image Segmentation Based on Neural Networks” IEEE TRANSACTIONS ON NEURAL NETWORKS, vol.16, no.4, July 2005.
16. M. Awad, K. Chehdi, A. Nasri. “Multicomponent Image Segmentation Using a Genetic Algorithm and Artificial Neural Network” IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, vol.4, no.4, October 2007.
17. X. Zhang, J. Chen, J. Dong. “Color Clustering using Self-Organizing Maps” In Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, IEEE, Beijing, China, Nov 2007.
18. N. Teeranachaideekul, P. Phokaratkul, S. Ongwattanakul, B. Emaruchi, “Feature Classification for Aerial Photography using Color Temperature” Master Thesis, Mahidol University, 2006.

APPENDICES

APPENDIX A
AERIAL IMAGE

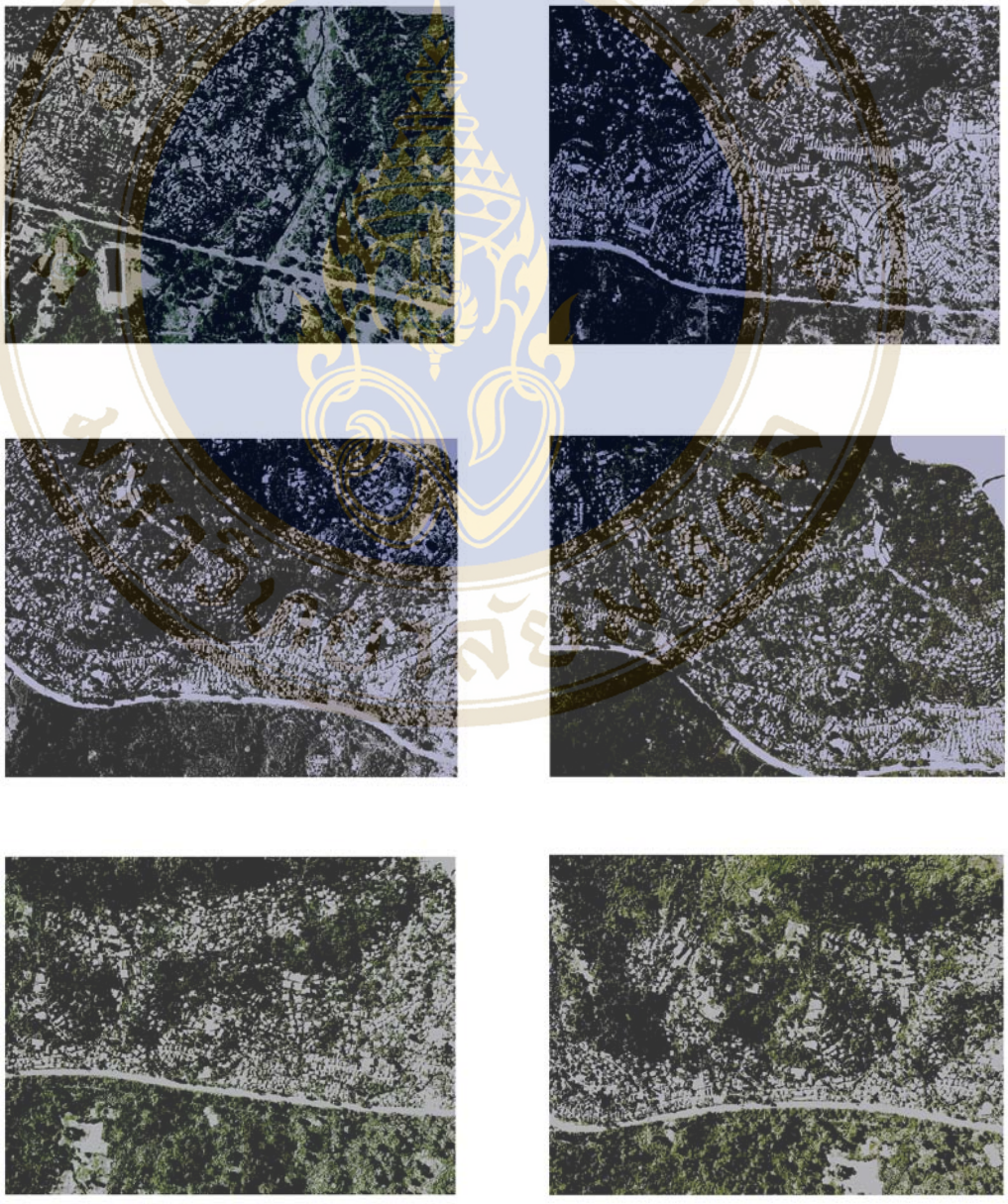


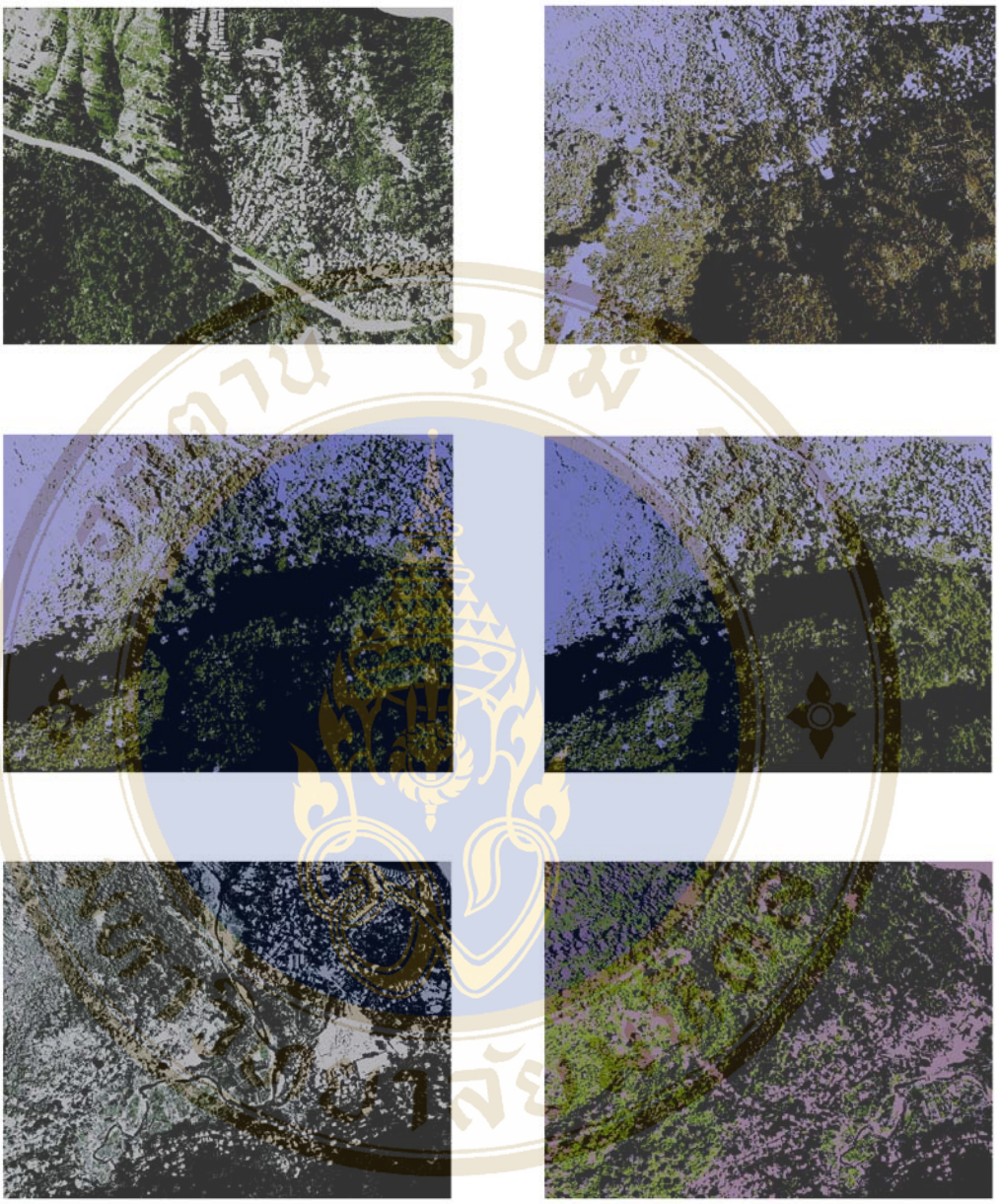



APPENDIX B
IMAGE SEGMENTATION RESULTS

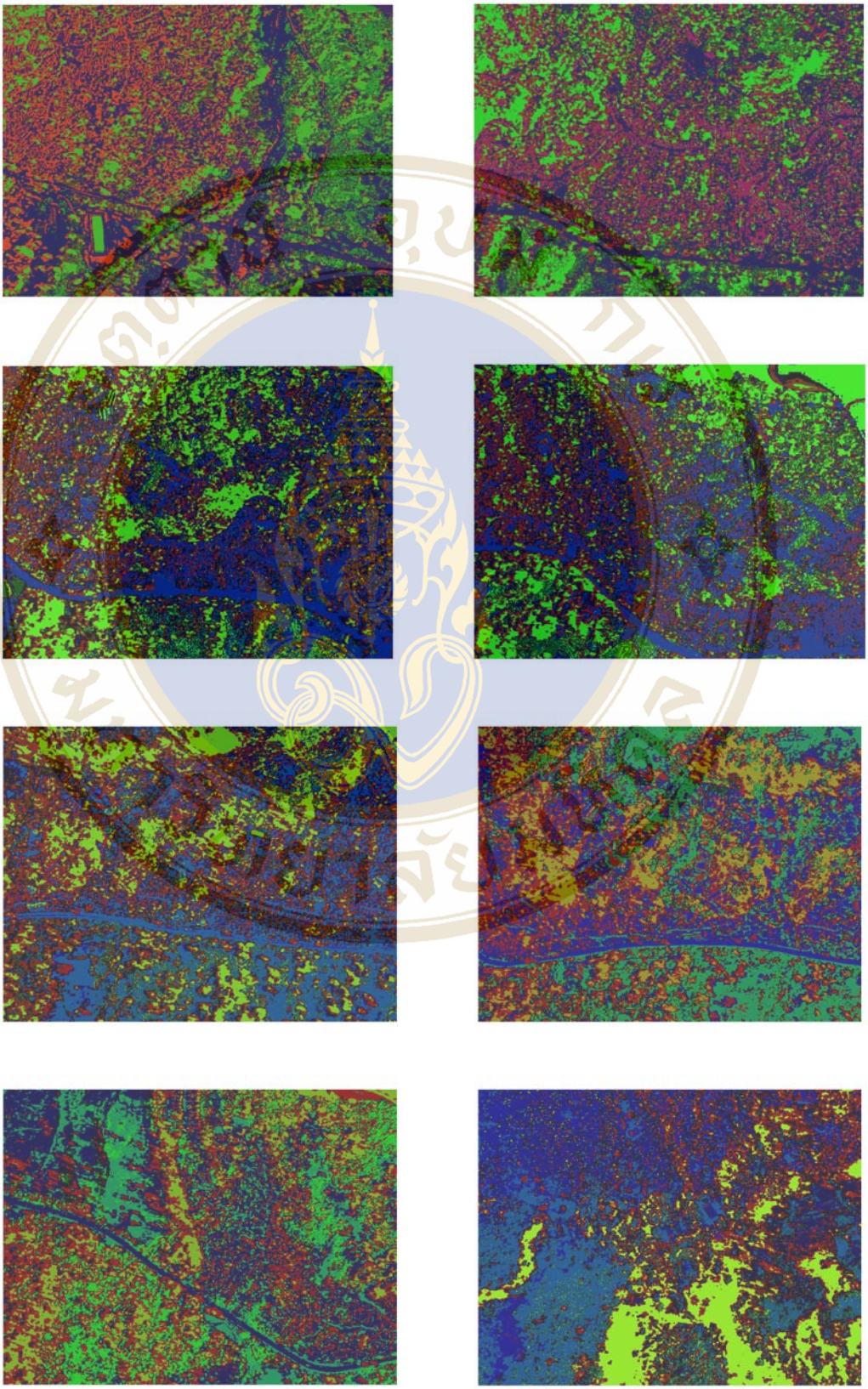
1. Color Quantization Result by Batch-Learning SOM (BLSOM)

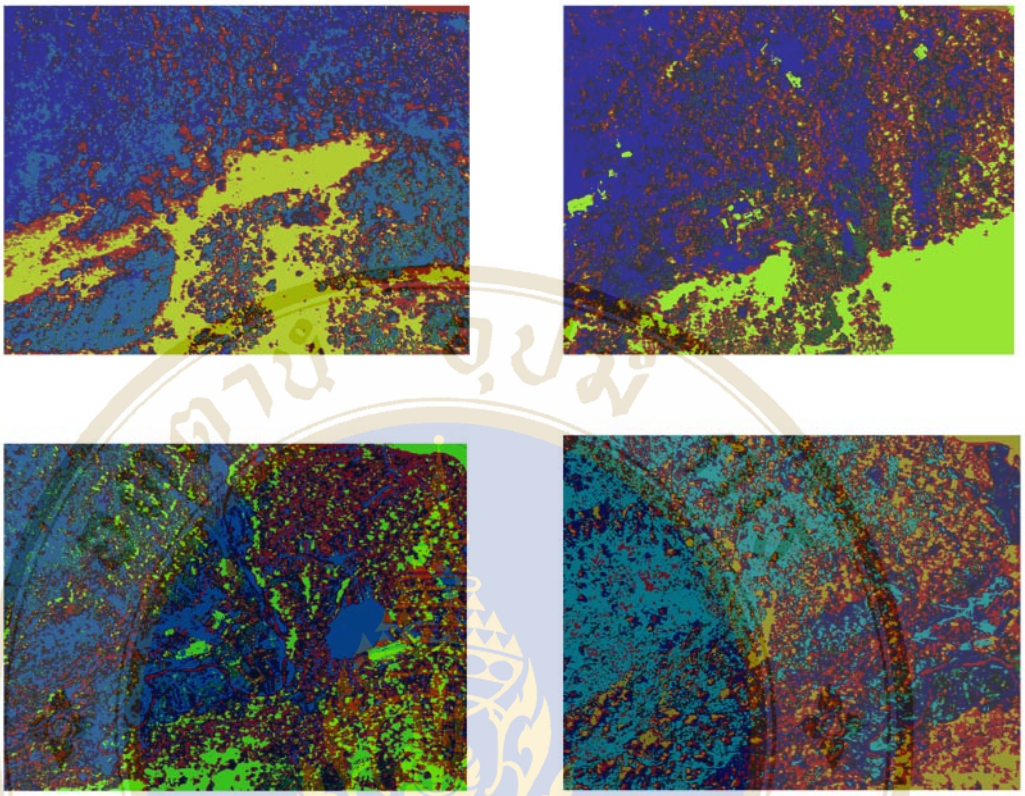
(a) The result from applied on RGB images



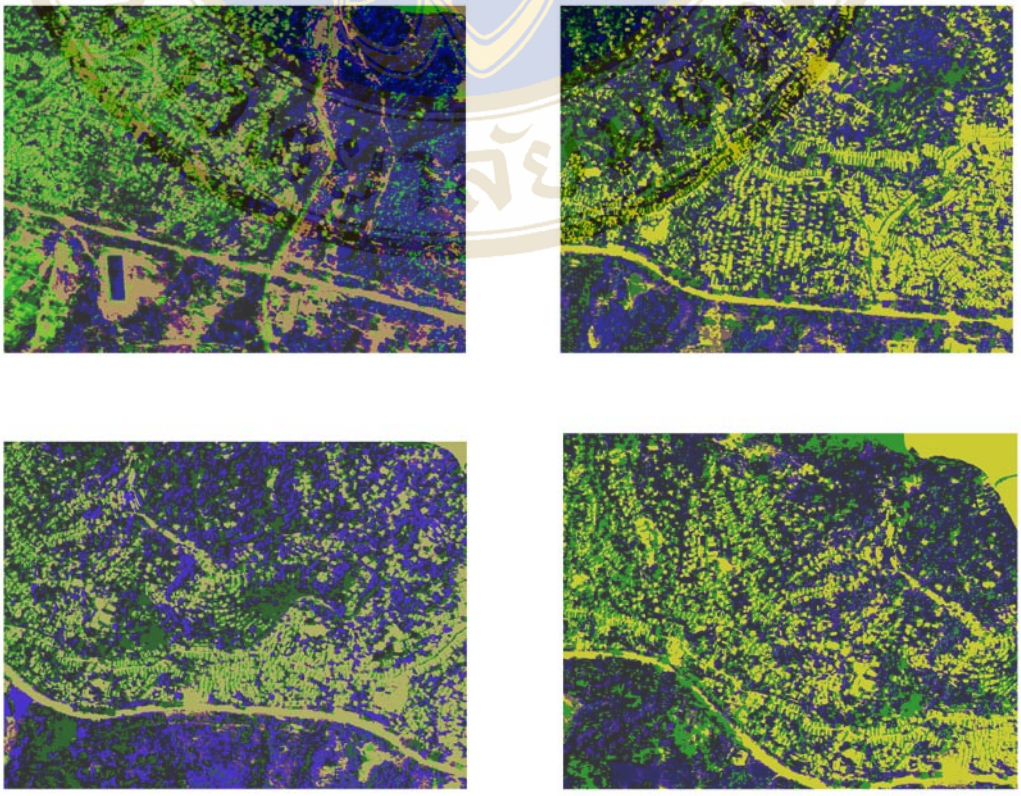


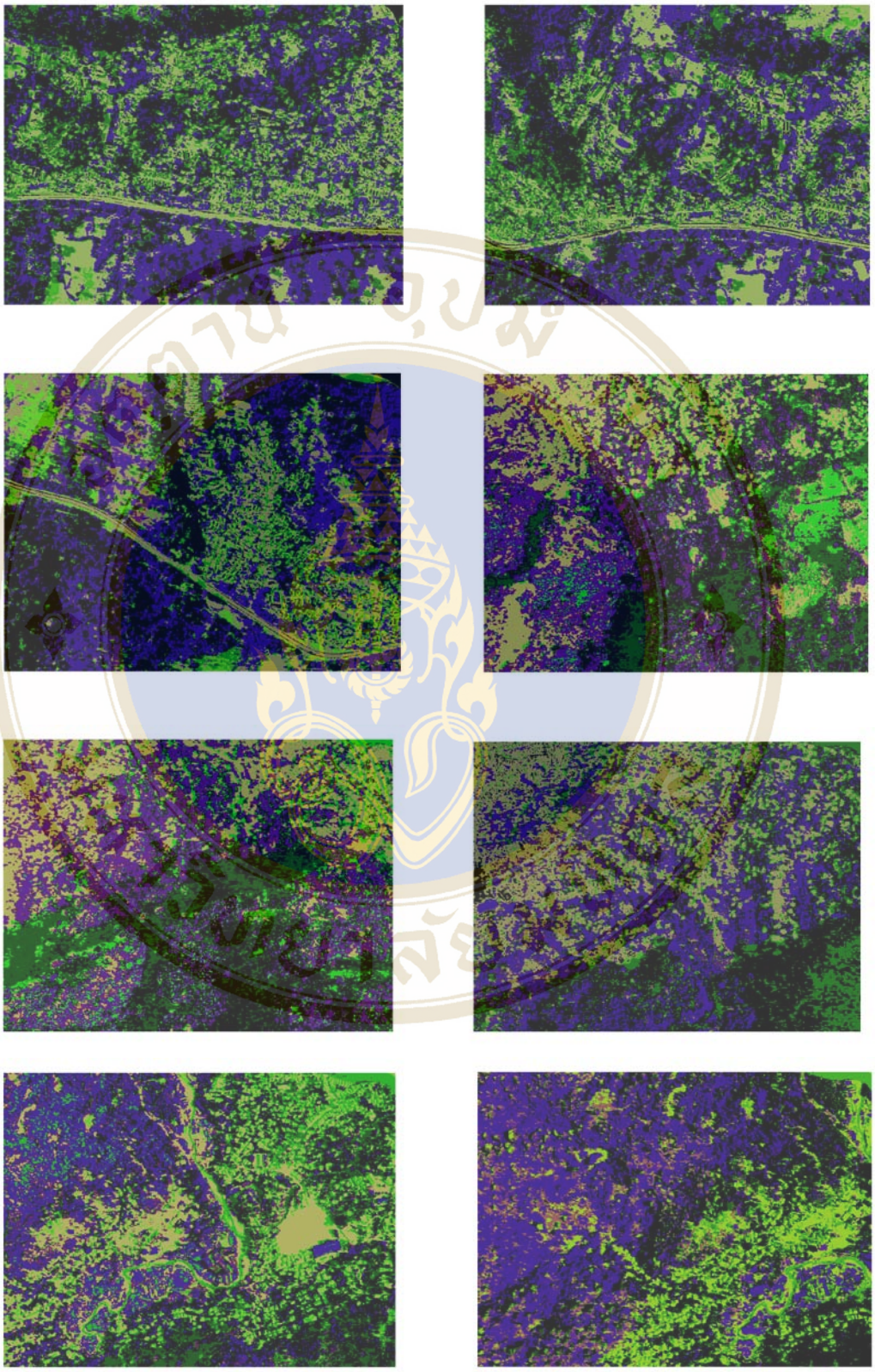
(b) The result from applied on HSV images





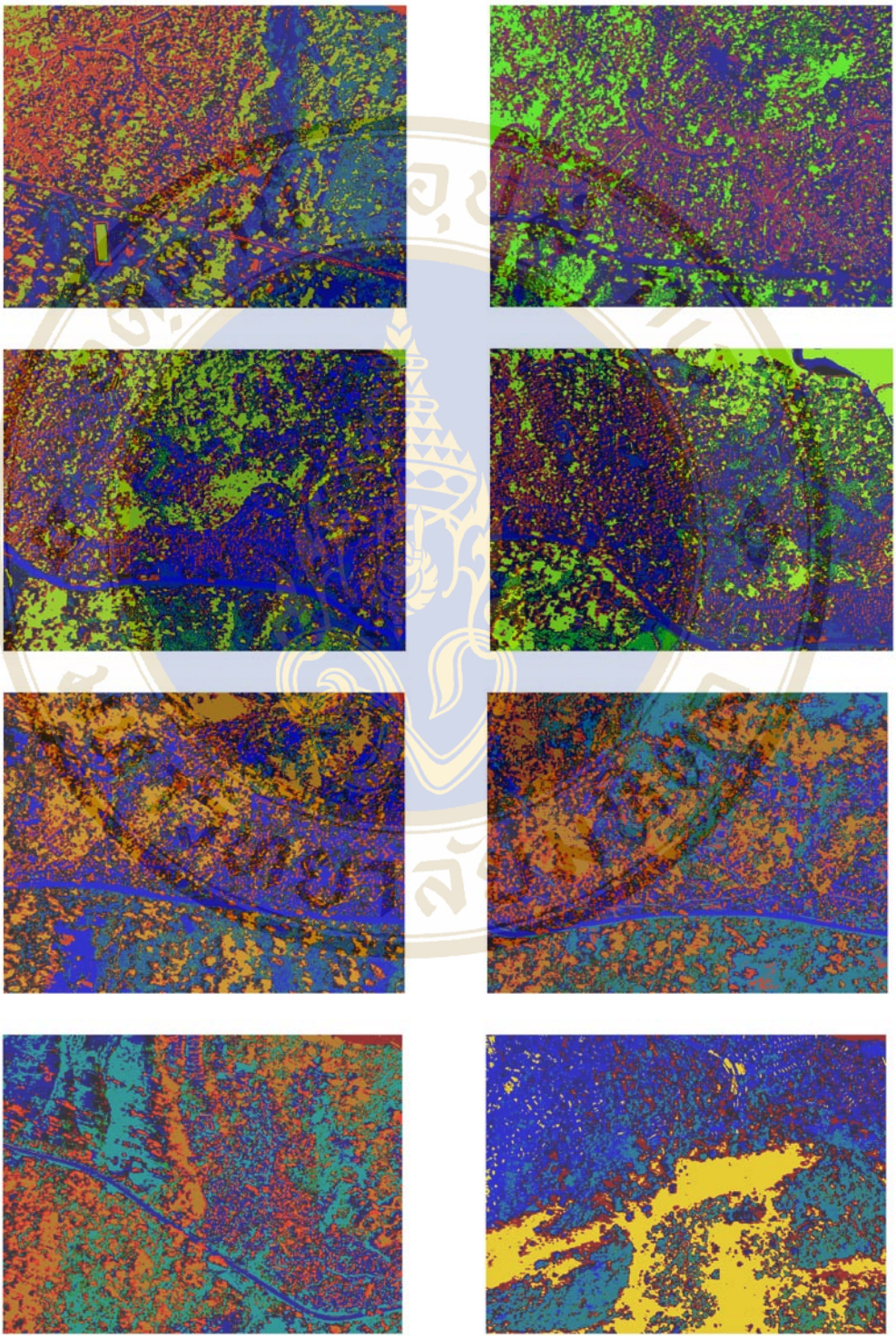
(c) The result from applied on Lab images

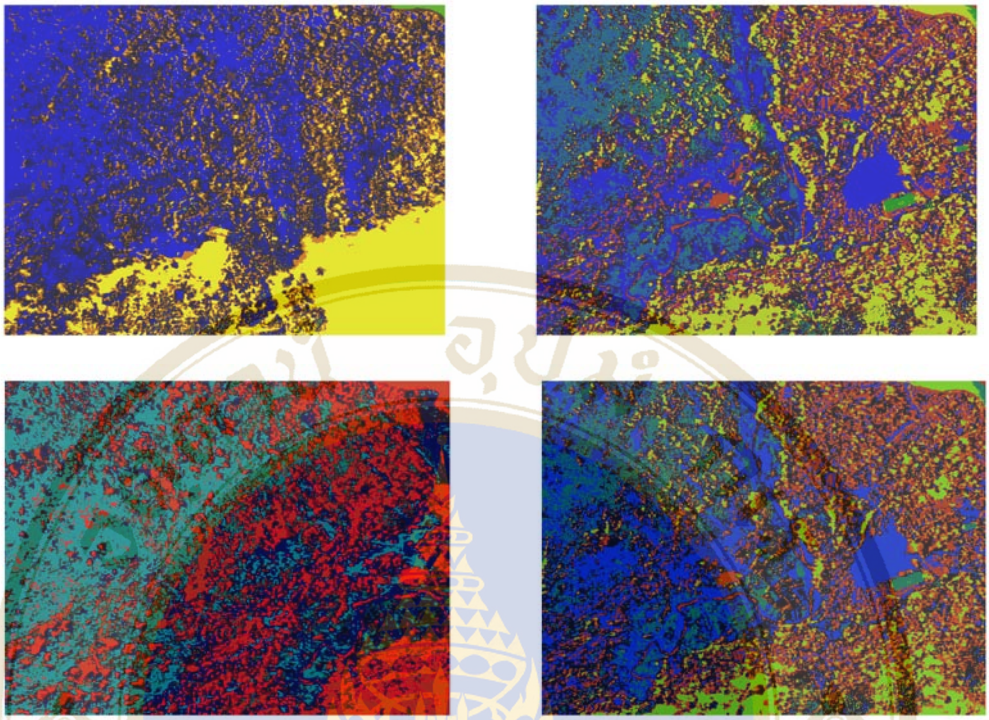




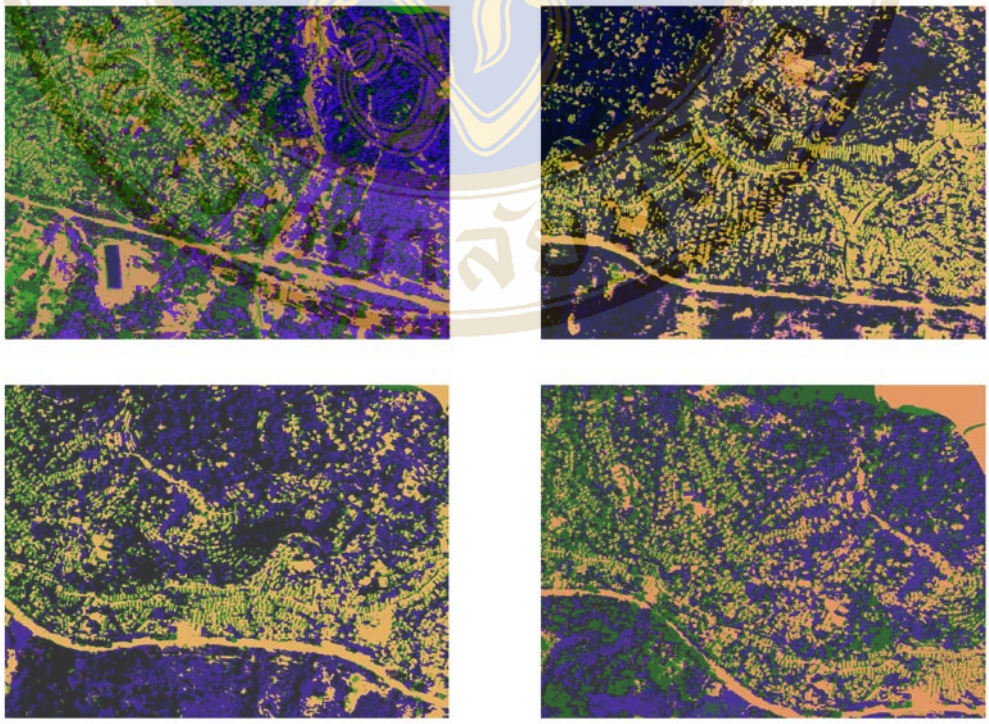
2. Color Quantization Result by Sequential SOM

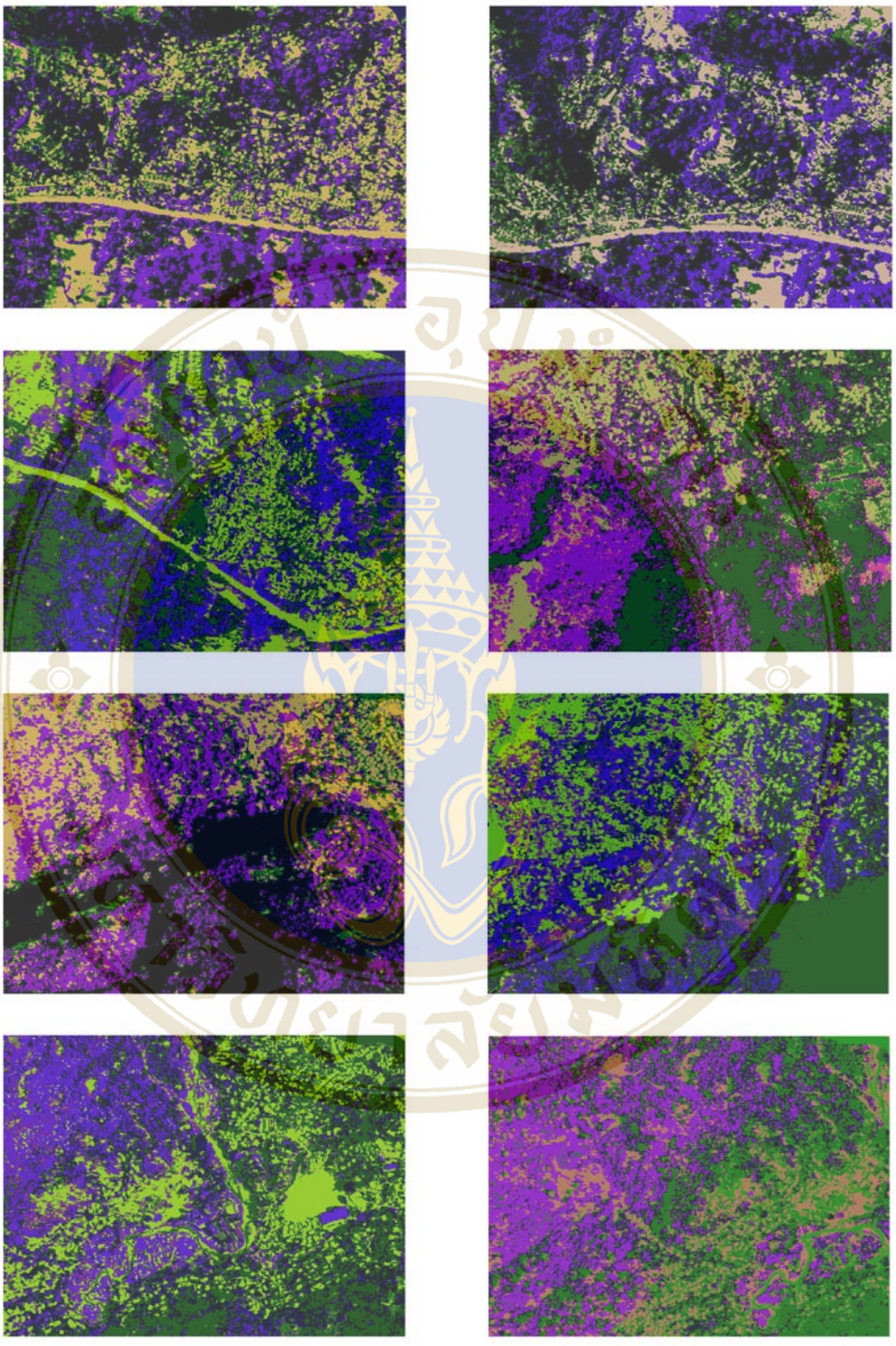
(a) The result from applied on HSV images





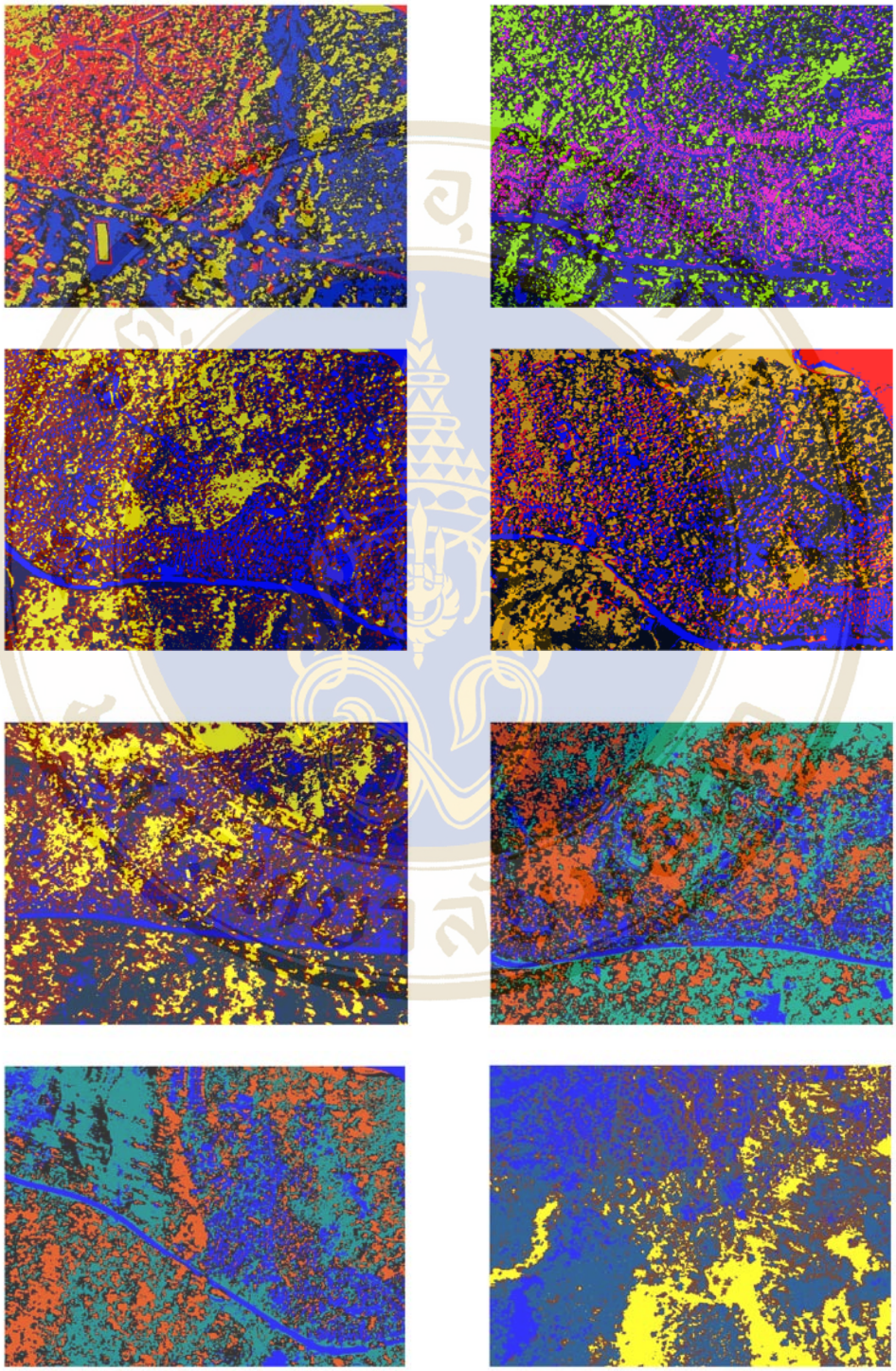
(b) The result from applied on Lab images

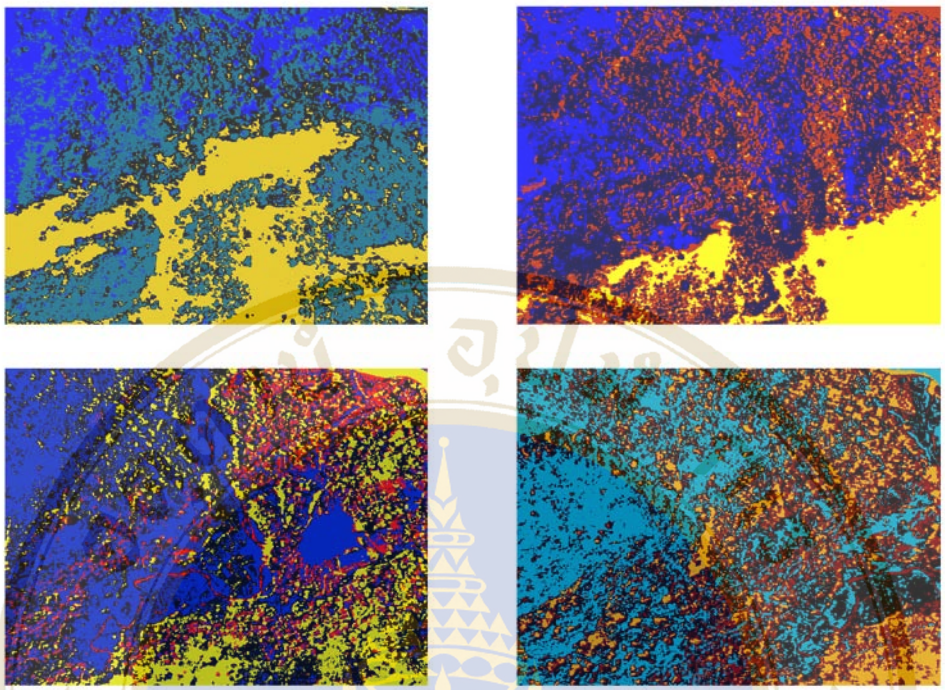




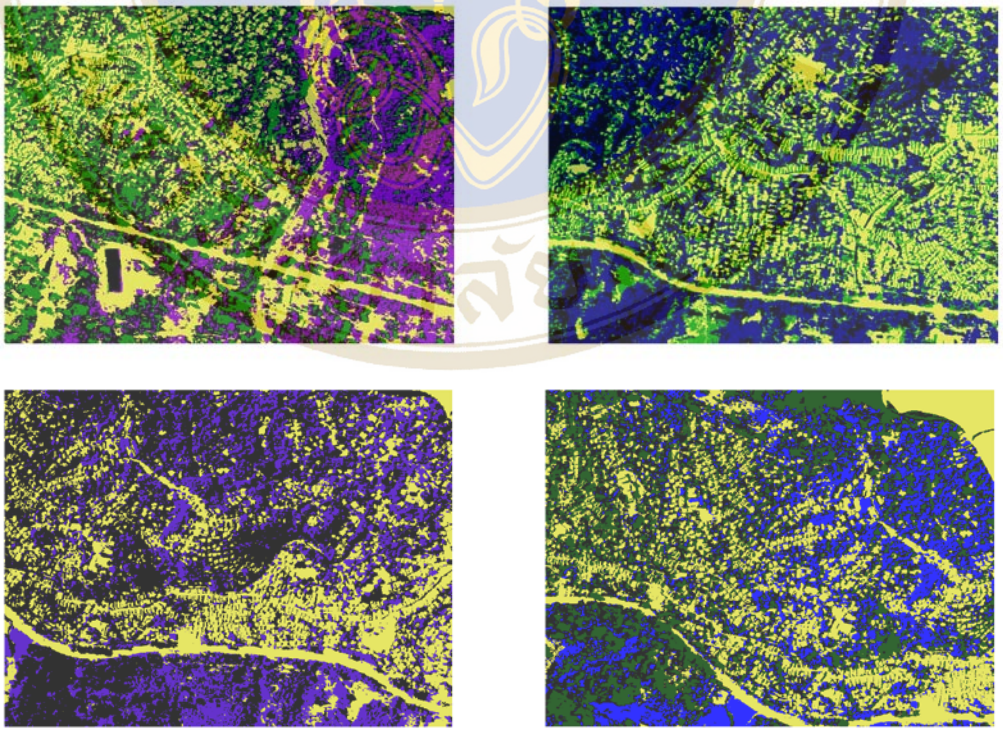
3. Color Quantization Result by K-Means

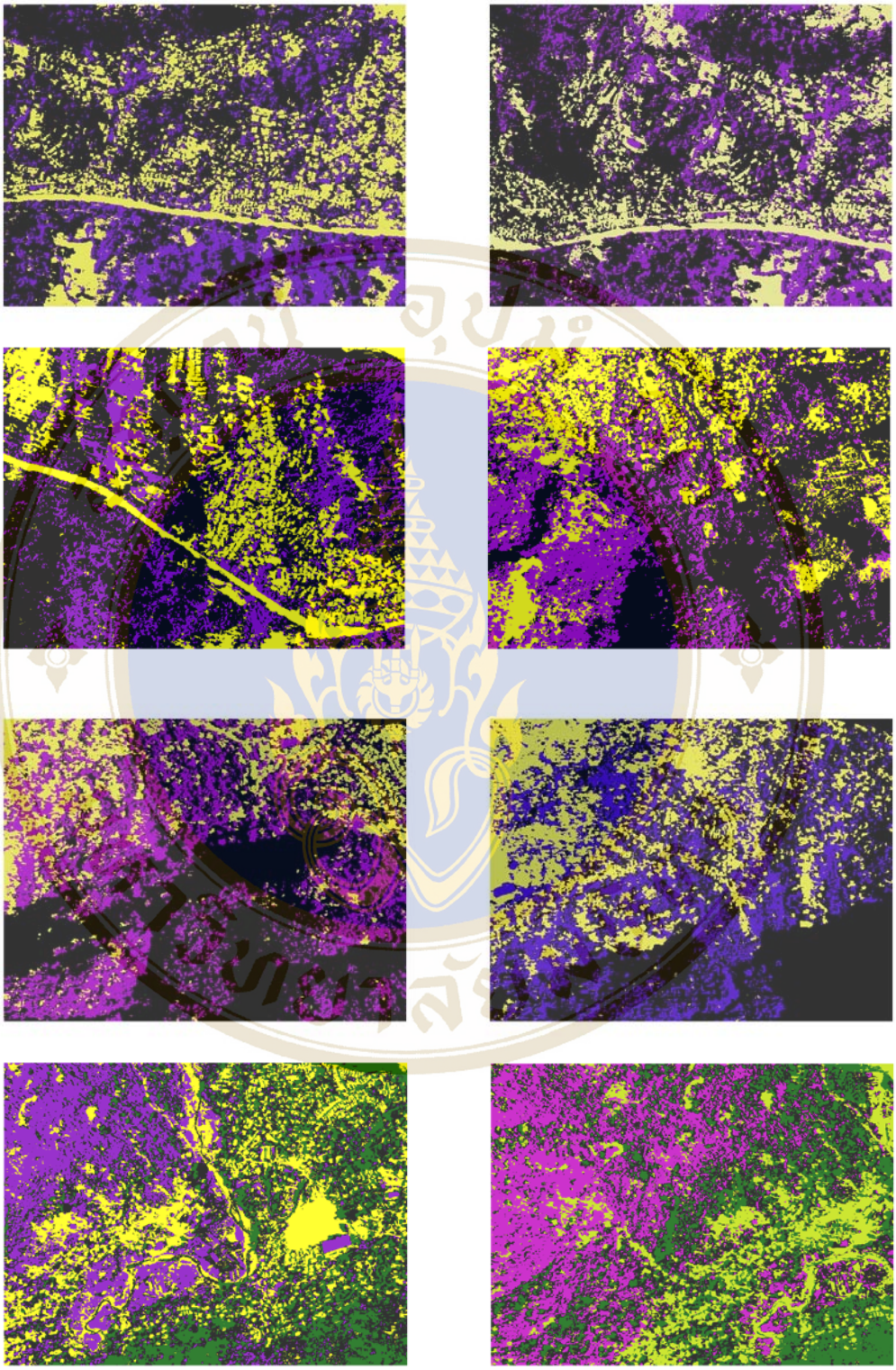
(a) The result from applied on HSV images





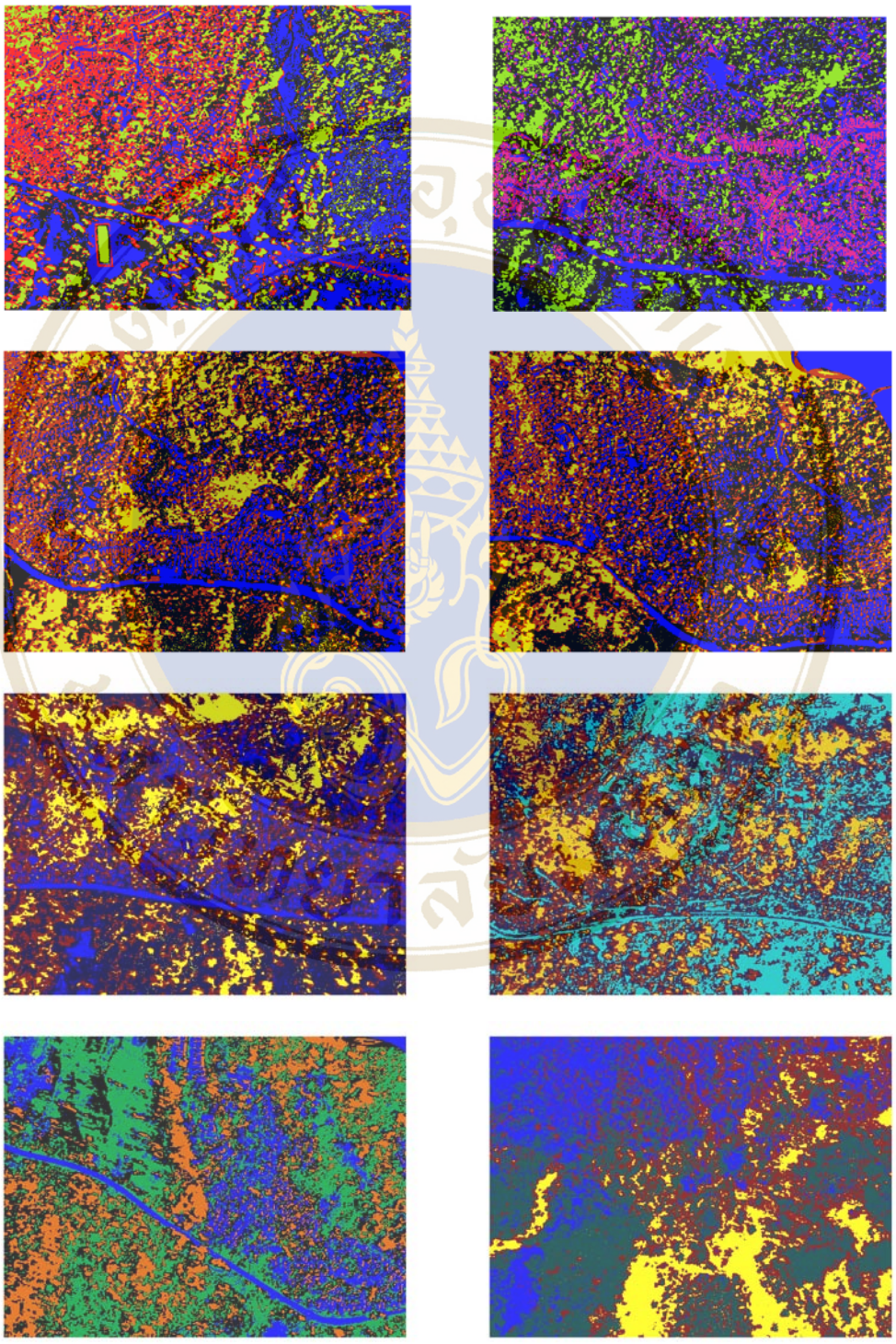
(b) The result from applied on Lab images

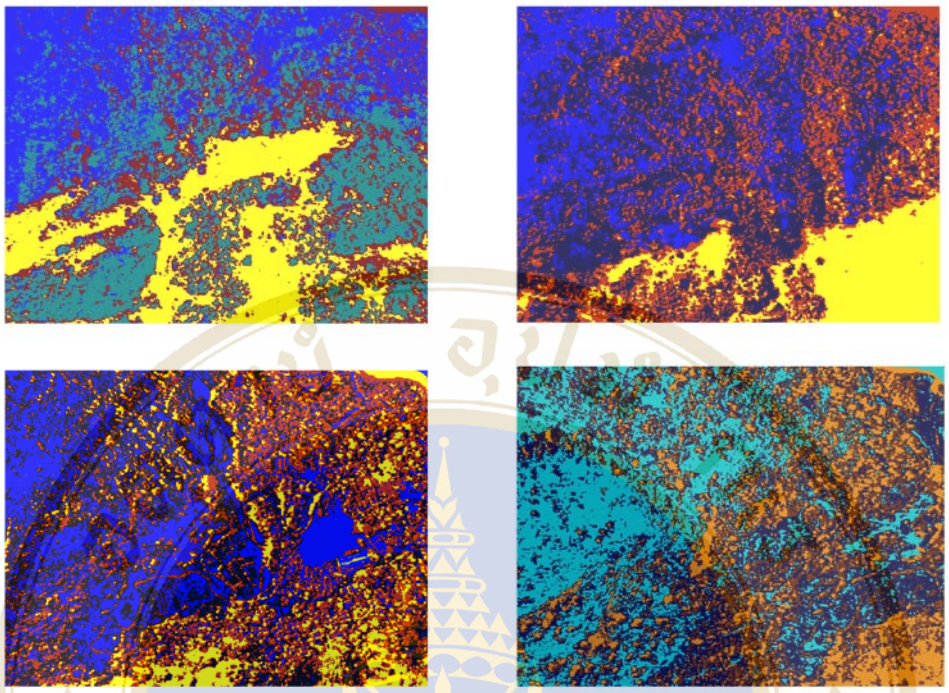




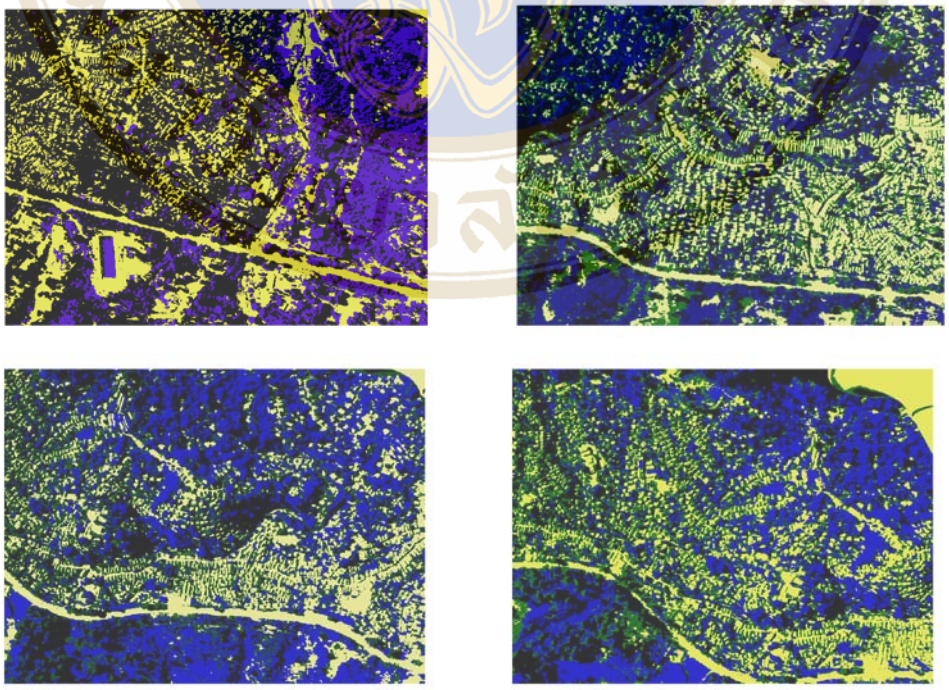
4. Color Quantization Result by FCM

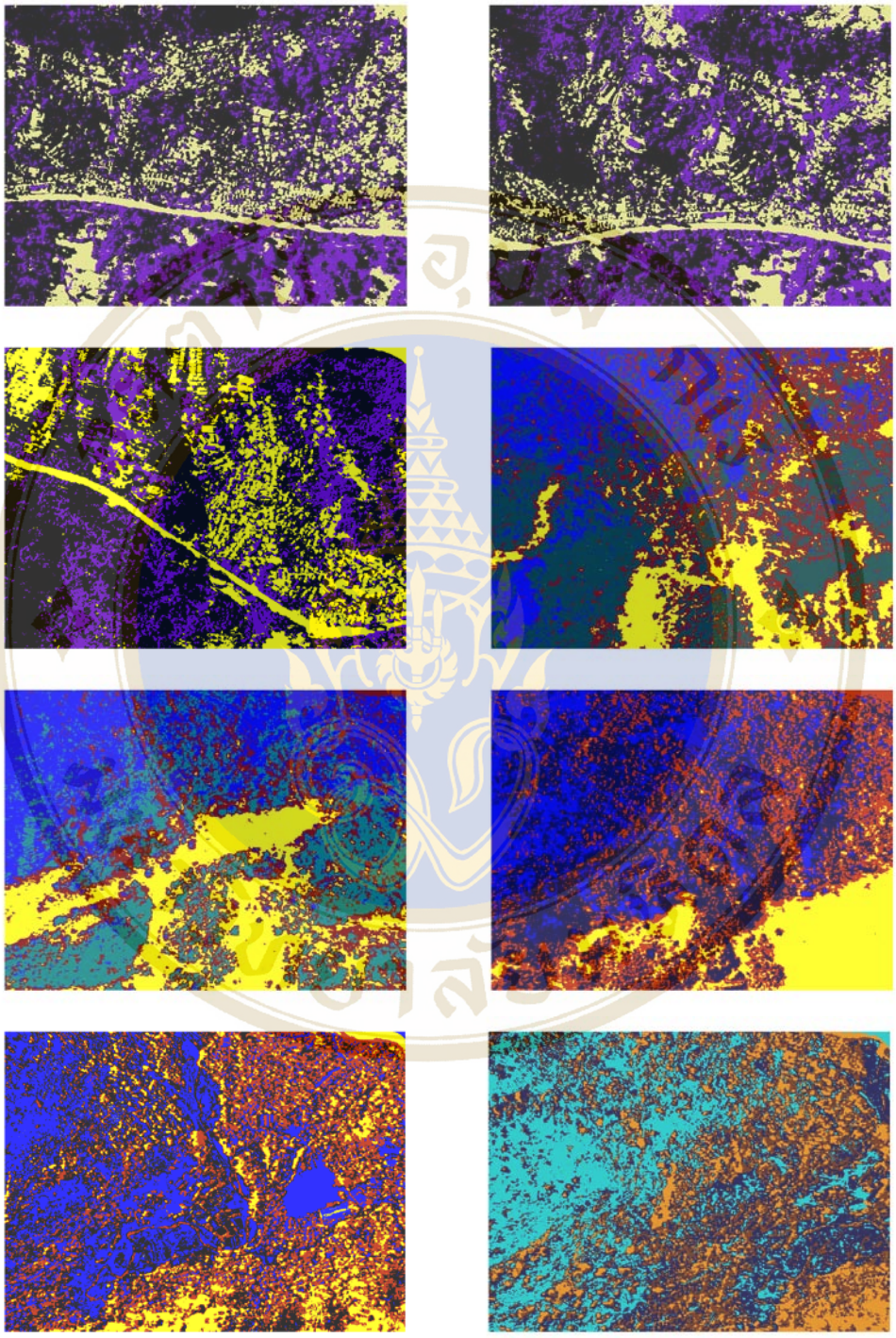
(a) The result from applied on HSV images





(b) The result from applied on Lab images





BIOGRAPHY

NAME	Mr.Suparat Wirattanapornkul
DATE OF BIRTH	10 December 1983
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Kasetsart University, 2001-2004: Bachelor Degree of Science (Biology) Mahidol University, 2006-2009: Master of Science (Technology of Information System Management)
HOME ADDRESS	16/18, Moo 9, Bangpood, Pakret, Nonthaburi 11120 Email: kongmed@gmail.com