LAND USE SEGMENTATION USING IMAGE EDGE DETECTION, MARKER-CONTROLLED WATERSHED AND K-MEANS CLUSTERING TECHNIQUES

CHONG ZI QING

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LAND USE SEGMENTATION USING IMAGE EDGE DETECTION, MARKER-CONTROLLED WATERSHED AND K-MEANS CLUSTERING TECHNIQUES

CHONG ZI QING

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Computer Networking) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

LAND USE SEGMENTATION USING IMAGE EDGE DETECTION, MARKER-CONTROLLED WATERSHED AND K-MEANS CLUSTERING TECHNIQUES

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ________________________ Date : 09/09/2021
(CHONG ZI QING)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of [Computer Science (Software Development)] with Honours.

SUPERVISOR : ________________________ Date : 09/09/2021
(GS. DR. OTHMAN BIN MOHD)

DEDICATION

This study is dedicated to my beloved mother, Loke Lai Thai for her support and understanding. Her care and accompany is necessary for me to complete this project on time. Moreover, it is also dedicated to my supervisor, Dr. Othman bin Mohd for his great guidance and advices. Lastly, this study is also dedicated to my fellow friends. They really give meaningful suggestion and share their knowledge on how to perform this project in a better way.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor, Dr. Othman Bin Mohd for giving his valuable guidance and advice. He inspired me greatly on how to proceed to this project and give me advices that really help me a lot when I met problems. well as inspired me greatly. His willingness that motivated me have contributed tremendously to this project. Besides, I would like to thank all my fellow friends who were willing to help when I asked for their helps.

Furthermore, I felt grateful to the school the authority of Universiti Teknikal Malaysia Melaka (UTeM) for giving me a good environment and chance to deal with this project. Finally, an honorable mention goes to my families and friends for their supporting and encouragement in completing this project. I am able to overcome a lot of problems and difficulties completed this project successfully on time with the help of everyone that are mentioned above.

ABSTRACT

Land use is one of the basic data in geographic information system (GIS). It is important for the planning to meet the needs of people and safeguard the future resources. Satellite image is an important source of input but it is difficult to segment the land based on different uses due to high resolution satellite data. Therefore, image processing is necessary to extract this basic information from satellite image as it is a method that use any algorithm to perform operation which can solve this problem. In this project, image segmentation is focused because it is defined as the process that partition or divide an image into homogeneous pixel groups (Verdonck, L., et. al., 2019). The main purpose of the project is to segment the Malacca land and then identify the different types of land use. There are many techniques for image segmentation thus several of them have been identified. Image Edge Detection, Marker-Controlled Watershed as well as K-means clustering are the techniques used to determine the contours of objects within the satellite image in this project. Then, evaluate the three (3) techniques of segmentation by conducting comparison on the output of each technique. The best technique is identified after the analysis conducted.

ABSTRAK

Penggunaan tanah adalah salah satu data asas dalam GIS. Penggunaan ini adalan penting untuk perancangan yang dapat memenuhi keperluan orang dan meletarikan sumber daya bagi masa depan. Imej satelit adalah salah satu sumber input yang penting tetapi pembahagian tanah berdasarkan penggunaan yang berbeza adalah sukar disebabkan data satelit yang beresolusi tinggi. Oleh itu, pemprosesan imej diperlukan untuk memperoleh maklumat asas ini dari imej satelit kerana proses ini merupakan kaedah yang menggunakan sebarang algoritma untuk melakukan operasi yang dapat menyelesaikan masalah ini. Projek ini akan berfokus kepada *image segmentation* kerana ia merupakan proses pembagian imej ke dalam kumpulan *homogeneous pixel* (Verdonck, L., et. al., 2019). Tugasan yang kena dilaksanakan adalah membahagikan tanah Melaka, dan kemudian mengenal pasti jenis-jenis penggunaan tanah. Beberapa teknik untuk segmentasi gambar telah dipilih dan diambil daripada kebanyakan teknik yang berwujud. Dalam projek ini, teknik *image Edge Detection, Marker-Controlled Watershed* dan *K-means clustering* digunakan untuk menentukan garisan kontur objek dalam gambar satelit. Selain itu, penilaikan tiga (3) teknik *segmentation* ini perlu dijalankan melalui perbandingan ke atas output setiap teknik dan akhirnya teknik yang terbaik dipilih.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PSM	-	Projek Sarjana Muda
UTeM	-	Universiti Teknikal Malaysia Melaka
GIS	-	Geographic Information System
LULC	-	Land Use/Land Cover
RMS	-	Rate-Monotonic Scheduling
CAGR	-	Compound Annual Growth Rate
PDE	-	Partial Differential Equation
ANN	-	Artificial Neural Network
FCM	-	Fuzzy C Means
EM	-	Expectation Minimization
PSNR	-	Peak Signal-to-Noise Ratio
USGS	-	United States Geological Survey
LISS	-	Linear Imaging Self Scanning Sensor
IRS	-	Indian Remote Sensing Satellites

LIST OF ATTACHMENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

Land use is one of the basic data in geographic information system (GIS). It is important for the planning to meet the needs of people and safeguard the future resources. Previous research done by Sowmya, D. R., et. al. (2018) stated that Land Use/Land Cover (LULC) mapping plays an important role in many land management applications, for example to create community map and proper urban planning. Satellite image is an important source of input for land use information but it is difficult to segment the land based on different uses due to high resolution satellite data. Image processing is a method that use any algorithm to perform some operations on an image to extract useful information from it, thus it can used to extract this basic information from satellite image. Satellite image processing consists of several techniques which are enhancement, fusion, segmentation, feature extraction, change detection, compression, feature detection, classification and. Image segmentation are focused to distinguish the object based on different types of land use from the satellite image.

Image segmentation is the partitioning of an image into homogeneous pixel groups. (Verdonck, et. al., 2019). Its goal is to change or simplify the representation of an image into more meaningful thing that is easier to be analyzed. It is the process of assigning a label to every pixel such that pixels with the same label share particular characteristics likes tone, texture or color. It involves detecting and classifying individual objects within the image, for example locate objects in satellite images such as road, forests, tree and etc.

Nowadays, image segmentation is widely applied for computer vision, face recognition, medical imaging, number plate identification and etc. Furthermore, it also has been used in many workplaces especially medical and security. In this project, image segmentation is used to segment the Malacca land and then identify the different types of land use. Supervised techniques perform classification through the training of machine or system where the training sets and target pattern are provided for the system. Unsupervised technique performs classification by generating clusters based on similar spectral features inherent in the image without training the provided samples.

There are several techniques in image segmentation including region based method, edge based method, thresholding method, watershed based method, clustering based method, Artificial Neural Network Based (ANN) based method and Partial Differential Equation Based (PDE) method. In this project, the covered techniques are image edge detection, Marker-controlled watershed and k-means clustering techniques. Comparison between the output of these 3 techniques to identify the best technique for land use segmentation of Malacca satellite image has been carried out. Besides, unsupervised classification is implemented since there is no dataset or samples are acquired and used to classify the different types of land use. It is fairly quick and easy to implement for non-professional but it has to spend time to identify and label the classes.

1.2 Problem Statement (PS)

Land use is the term which described the use of land for humans. Nowadays, there are several common land use which are recreational, transport, agricultural, residential and commercial. It is important data used in the planning to meet the needs of people and safeguard the future resources. Besides, the land use information is necessary for the land management practices which aim to solve the natural resource management issues for example, the salinity and water quality. There are many researches about land use. One of the found researches was conducted by Kim, C. (2016), its purpose is to use satellite imagery to carry on the classification of the land use and land cover status, the identification of land use changes especially deforestation and forest degradation in Lombok Island in the past 20 years.

Since it is difficult to distinguish the land based on different uses due to high resolution satellite data, this project aims to solve this problem to segment land and identify different types of land use for Malacca. In this process, some problems can directly influence the motives of this project. These kinds of problems would lead to poor result or output of the project. Therefore, it is important to overcome these problems in order to find out the best techniques for land use segmentation. A good output from the satellite image segmentation is necessary for us to identify the different types of land use correctly and efficiently. The problem statements (PS) are summarized in the Table 1.1.

Table 1.1: Summary of Problem Statement

PS	Problem Statement
PS1	Difficult to distinguish the object based on different types of land use from the satellite image.
PS2	Lack of understanding about different segmentation techniques.
PS3	There are too many techniques can be applied for image segmentation.

1.3 Project Question (PQ)

Based on the problem statement summarized in Table 1.1, the project questions have been figured out to overcome the issues that may happen in the land use segmentation. The problem questions are:

i. It is difficult to distinguish the objects from the satellite image with naked eyes. Therefore, it is necessary to find out the technology or techniques that allow us to do that. How to distinguish the objects from satellite image?

ii. Lack of understanding about different techniques would increase the possibilities of choosing the wrong or worse technique. Therefore, it is important to know the advantages and disadvantages of each technique. What are advantages and disadvantages for each of the selected segmentation techniques?

iii. From the present-day knowledge society, there are too many techniques can be applied for image segmentation. Therefore, the comparison of each output is necessary to be conducted to find out the best techniques. Which techniques can be applied to produce better results?

These project questions (PQ) are summarized in the Table 1.2.

Table 1.2: Summary of Project Question

PS	PQ	Project Question
PS1	PQ1	How to distinguish the objects from satellite image?
PS2	PQ2	What are advantages and disadvantages for each of the selected segmentation techniques?
PS3	PQ3	Which techniques can be applied to produce better results?

1.4 Project Objective (PO)

The problem statement as mentioned in Table 1.1 and project questions mentioned in Table 1.2 are used to embark the objectives of this project. This project objectives (PO) are summarized in Table 1.3.

Table 1.3: Summary of Project Objectives

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To identify techniques of land use segmentation.
		PO2	To apply three (3) techniques which are image Edge Detection, Marker-controlled Watershed and K-means Clustering to determine the contours of objects within the satellite image.
PS2 PQ3	PQ2 PQ3	PO3	To evaluate the 3 techniques to identify the best technique for land use segmentation.

1.5 Project Scope

The project study was bounded to the following scope as mentioned below:

The land use segmentation is conducted on the satellite image of Malacca city in Malaysia and the image is downloaded from Google Earth. Google Earth is selected because this platform is free and easy for us to download the image. This image has been downloaded on 26 March 2021 and it is with the highest resolution available in Google Earth which is 4800 * 2782 pixels.

The satellite image is processed with 3 different segmentation techniques in Matlab that is running on Windows system. Matlab is selected because this software allows us to code and debug mush faster.

There are six (6) segmentation regions, which consists of Residential, Green spaces, Commercial and industrial area, Sediment, Coastal area and Deep Sea.

1.6 Project Contribution (PC)

In this project, the best segmentation technique is identified to distinguish the object based on different types of land use of Malacca from the satellite image. This can contribute to the further research or study of image processing in the many fields of science and technology especially for remote sensing and computer vision. Furthermore, the knowledge obtained from this project can help non-professional in many workplaces likes land use mapping, environmental monitoring, and land resources planning and management.

1.7 Report Organization

In this project, there are 6 chapters to be discussed. The report started from chapter 1 that is introduction until chapter 6 which is conclusion.

Chapter 1: Introduction of the project is discussed in this chapter. Introduction states about the project background, problem statement, the project questions and the project objectives. Problem statement directly affected and decided what is the project questions and objectives. Furthermore, this chapter also states about the project scope and contribution.

Chapter 2: This chapter discuss about the literature review of the project. Literature review states about the related work done by some researcher previously. Moreover, this chapter also states about the critical review of current problem and justification, as well as the proposed solution based on the related researches.

Chapter 3: Methodology of the project is discussed in this chapter. Methodology describes every stage for the selected methodology and the activities carried on in each stage. Besides, all the activities are explained stage by stage in the project milestones.

Chapter 4: This chapter discuss about the implementation of this project. Implementation states about the hardware and software required in this project and describes all the steps involved in the implementation. Besides, the progress of the development status is needed to be recorded.

Chapter 5: This chapter discuss about the testing and analysis of this project. Testing and analysis state about the details on how to compare the output of each segmentation technique and the result of which is the best technique for the land use segmentation.

Chapter 6: The conclusion of this project is discussed in this chapter. Conclusion consists of the project summarization, project contribution, project limitation and future works which can be done to improve the project.

1.8 Conclusion

Land use is one of the basic data in GIS. It is important for the planning to meet the needs of people and safeguard the future resources. Satellite image is an important source of input but it is difficult to segment the land based on different uses due to high resolution. Therefore, image processing is method to extract this basic information from satellite image. In this project,

image segmentation is focused because it can partition an image into homogeneous pixel groups or image objects. There are many techniques for image segmentation thus image edge detection, Marker-controlled watershed and K-means clustering have been chosen to carry out comparison between them in order to identify the best technique.

The objectives of this project are to identify segmentation techniques for land use as well as apply 3 techniques which are image edge detection, Marker-controlled watershed as well as K-means clustering to determine the contours of objects within the satellite image. Then, evaluate the 3 techniques of segmentation. On the other hand, the scopes of the project are satellite image of Malacca downloaded from Google Earth and Matlab is used to carry out the project. The target at the end of this project is to segment the Malacca land and identify the different types of land use like residential, agricultural, transportation and etc. Through this, non-professional can have better understanding of the image processing technique and identify the best technique from the comparison between the selected techniques.

The next chapter discusses about the related work done by some researcher previously in Literature Review. In this chapter, three (3) selected segmentation techniques for land use will be studied and discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, there are 5 sub-chapter including introduction, related work, critical review of current problem and justification, the proposed solution and the conclusion. The related work done by some researchers previously are discussed in this chapter. Canny Edge Detection, Marker-Controlled Watershed as well as K-Means Clustering Techniques are the discussed techniques for land use segmentation of Malacca. The purpose of this chapter is to carry out the comparison by referring the previous related work to identify the best technique for the land use segmentation.

2.2 Related Work/Previous Work

The domain for this project is image processing which refers to a method that use any algorithm to perform some operation on an image to extract useful information from image. Nowadays, there are numerous methods and techniques for satellite image processing. It is necessary to do a lot of researches in order to choose three (3) techniques used for land use segmentation. Works related to satellite image processing which are done by the previous researches should be referred and studied. Based on Asokan, A., et. al. (2020), there are nine (9) methods in satellite image processing. Figure 2.1 shows the methods in satellite image processing and three (3) techniques for segmentation.

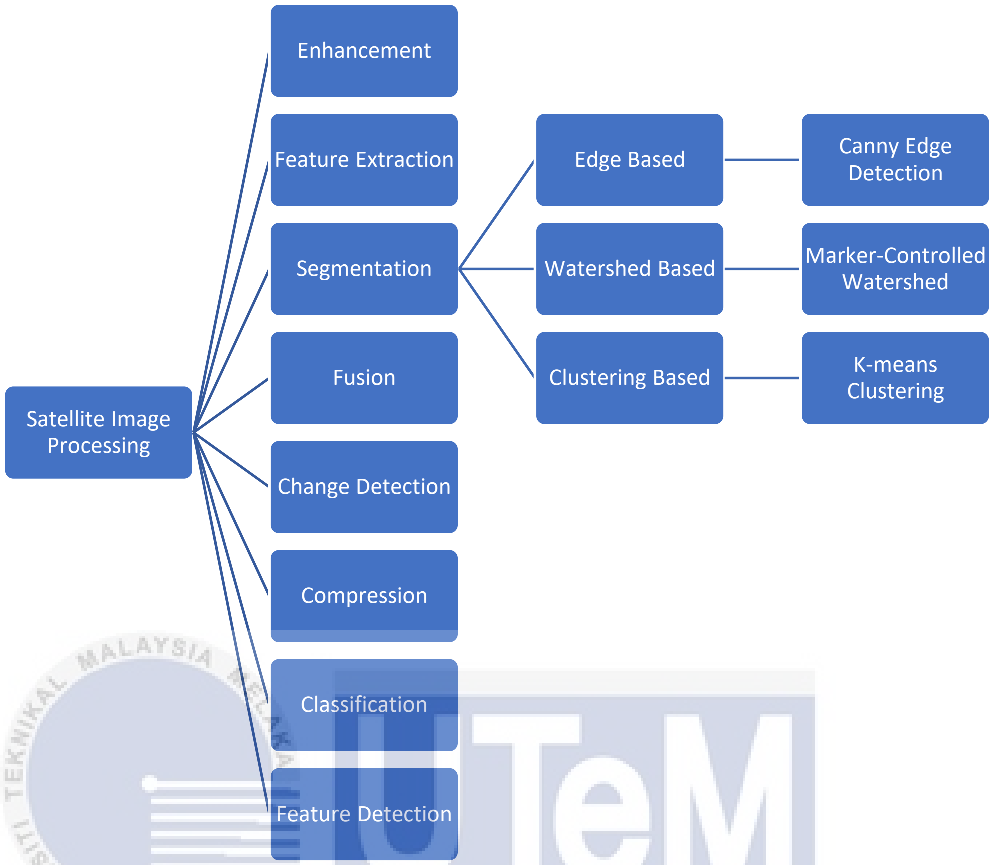


Figure 2.1: Summarize for The Satellite Image Processing

Based on Figure 2.1, this project focuses on satellite image processing that involve techniques including edge detection, watershed based and clustering based techniques. Only one method is selected among these nine (9) methods for study in this project. Canny Edge Detection, Marker-Controlled Watershed and K-means Clustering are the technique used for the land use segmentation.

2.2.1 Image Processing

Image processing is one of the methods that perform operations on an image using any algorithm for extracting useful information from it. In this process, an image will be the input and altered image as well as the report about the features/information obtained from that image will be the output. Image processing can be classified into two types which are digital and analog image processing. Analog image processing is the processing carried out on hard copies (two-dimensional analog signals) likes printouts and photographs by analog means. On the contrary, digital image processing is the manipulation of digital computer to process digital images using an algorithm.

The main reason why image processing is growing rapidly nowadays is due to its achievement in the extraction of useful information without altering the other features of the image. The digital image processing can be divided into 4 classes which are image enhancement, image analysis, image compression and image restoration. Image segmentation is one of the examples for image analysis. First thing to do in image processing is to import the image via acquisition tools, then analysis and manipulate the image.

There are many platforms provided for people to carry out image processing. MATLAB is one of the programming and numeric computing software that is available for users to apply various algorithms and mathematical formula to process the digital image. For digital image, the processing algorithms are used for many purposes. Converting signals from a sensor into digitals image, and extracting size, scale or numbers in the scene captured in an image are the example of its purpose.

2.2.2 Satellite Image Processing

Satellite Image Processing is a vital field for research and consists of the images of earth taken by satellites. According to the latest report from Allied Market Research, global commercial satellite imaging market was valued at $2,240.0 million in 2018. For now, it is registering with a CAGR (Compound Annual Growth Rate) of 11.2% from 2019 to 2026. This shows that the development of satellite image processing is growing time by time.

The images are captured in digital form by satellite and processed to exact the useful information from the Earth surface likes the environmental conditions and omen that can used to predict natural disasters. There are numerous methods and algorithms for satellite image processing including enhancement, fusion, segmentation, change detection, feature extraction, compression, feature detection and classification. There are many satellite image processing techniques and they are summarized in Table 2.1.

Table 2.1: Satellite Image Processing Techniques

Satellite Image Processing Techniques	**Description**
Image Enhancement	It is the way or method to improve the quality and the information content of original data. (Haldar, S. K. (2018). Numerous algorithms have been designed and developed to overcome various brightness related problems in image processing and achieve the contrast enhancement and (Asokan, A., et. al., 2020)
Feature Extraction	It involves creating features for selection and classification such as distinguishable features and forwarded to classification stage. It plays an important role in deciding the efficiency of the image classification process. (Asokan, A., et. al., 2020)
Image Segmentation	Image Segmentation is the partitioning of an image into homogeneous pixel groups such as objects and boundaries that can be located. (Verdonck, L., et. al., 2019).
Image Fusion	It is a process that gather all the important information from multiple images by combining them to form a new image. Its purpose is to make images to be more understandable for people as well as machine perception.
Change Detection	It is done to observe and identify the changes in a certain area given that satellite data over that area at different times or under different conditions from images in a easier way.
Image Compression	This technique can used to reduce the image size without reducing the image quality. It is the solution related to the transmission and storage of large

	amount of information for digital image. (Parmar, C. K., et. al., 2015)
Image Classification	It is a method for pattern recognition in which pixels/images are classified according to certain similarity measures. It can be divided three (3) approaches which are into supervised, unsupervised, as well as post-classification. (Asokan, A., et. al., 2020)
Feature Detection	It is a process of examining every pixel to see if there is any feature present in an image. It plays an important role for environmental analysis, urban monitoring and etc. It is mostly used to differentiate features such as roads, trees, buildings and etc.

Based on Table 2.1, different processing techniques for satellite images are explained. Each of these techniques can be performed for various applications of satellite images processing. In this project, image segmentation is mainly discussed.

There are many free sources of satellite images that are available for people, for example USGS (United States Geological Survey) EarthExplorer, Google Earth, Copernicus Open Access Hub, and Sentinel Hub. Before conducting any study or analysis on images, it is necessary to check and know the most applicable sources of the images for us. For this project, Google Earth was used for downloading the satellite image of Malacca.

2.2.3 Unsupervised Classification

Unsupervised technique performs classification by generating clusters based on similar spectral features inherent in the image without training the provided samples. In this classification, it is necessary to specify the number of classes or types, the spectral classes or types will be created separately based on the numerical information in the data, for example the pixel values for different bands. Clustering algorithm is one of the methods for unsupervised classification which used to determine the statistical and natural grouping of the data. Pixels are assembled together into according to their spectral similarity. Then, machine or computer will use feature space to analyze and group the data into different classes or clusters.

Unsupervised classification is implement since there is no dataset or samples are acquired and used to classify the different types of land use. It is fairly quick and easy to implement for non-professional but it has to spend time to identify and label the classes.

2.2.4 Image Segmentation

Image Segmentation is the process that partition or divide an image into homogeneous pixel groups (Verdonck, L., et. al., 2019). Each of the pixels in a group or segment are similar regarding the characteristic of image such as color, intensity or texture. The output of image segmentation is a set of segments with different characteristics or a set of contours which are extracted from the image.

It is the basic step for analyzing the images in order to extract data from them. Numerous image processing techniques are available and segmentation is a challenging field with the least complexity and effective implementation. (P.Jayapriya, et. al., 2019). There are many segmentation techniques that can be classified into two types based on the properties of an image which are similarity and discontinuity.

For discontinuity, its purpose is to extract regions that differ in properties such as color, intensity, texture, or other statistics of image based on abrupt changes in intensity likes the edges in an image. On the other hand, similarity aims to group pixels based on common properties. (P.Jayapriya, et. al., 2019).

Nowadays, various techniques can be applied for the segmentation of satellite images including region based method, edge based method, thresholding method, clustering based method, watershed based method, ANN based method and PDE Based method.

Table 2.2: Description of Segmentation Techniques

Segmentation Techniques	**Description**
Thresholding	It is the simplest way to segment the image by dividing the image pixels according to their intensity level. It is used

	over images which have lighter objects than background. (Kaur, D., et. al., Y, 2014)
Edge Based	This techniques recongnize the boundaries by detecting the sharp change in the intensity or brightness of the image. Hence the obtained boundary marks the edges or the contours of the desired object. (Matta, S. 2014)
Region Based	This technique is the method that segments/ divides the image into various sub regions which have similar characteristics.
Clustering Based	This technique segments the image into clusters which have pixels with similar characteristics such as color texture, size and etc. (Kaur, D., et. al., 2014). It follows the concept of unsupervised learning.
Watershed Based	The technique is considered as another region-based method that transforms an image into a gradient image. It decomposes an image completely, then assigns each pixel to a region or watershed.
Partial Differential Equation Based (PDE)	This technique is appropriate for time critical applications. The results of the PDE method are blurred edges and boundaries that can be shifted using close operators. (Kaur, D., et. al., 2014)
Artificial Neural Network Based (ANN)	This technique refers to the simulation of the learning process of human brain for the purpose of decision making. This method works in a manner that similar with neural network.

2.2.5 Image Segmentation based on Edge Detection Techniques

Kirti, R., et. al., (2017), stated that edge can be considered as the boundary between two different regions in an image. Edge detection is a process used to identify and locate sharp discontinuities in an image. According to Muthukrishnan, R., et. al., (2011), the immediate changes in pixel concentration that distinguish boundaries of objects in a scene result in the discontinuities. Edge detection usually engage convolution of an image through an operator. This operator is constructed to be responsive to great gradients in the image even though returning value of zero in uniform regions.

There are two types of edge detection operators, which are gradient and gaussian. The gradient-based operators consist of Sobel, Robert and Prewitt operator which computes first order derivations in digital image whereas the Gaussian-based operators includes Laplacian of Gaussian and Canny edge detector that computes second order derivations.

The operators used for Edge detection techniques is necessary to be selected to be receptive to such a regular change in some cases such as reduced focus or refraction that can result in objects through boundaries defined by a regular change of intensity. Noise and high computational time will cause problems likes missing true edges, edge localization and fake edge detection. Thus, there are a lot of research conduct the comparison of various edge detections and analyze the performance of the different techniques based on their different application.

2.2.5.1 Canny Edge Detection

Canny edge detection technique is a standard edge detection technique in which an optimal edge detector based on a set of benchmarks for its algorithm including minimizing the error rate to find the most edges, maximizing localization by marking edges to the actual edges as closely as possible, and marking edges once when a single edge exists for minimal response. (Al-Amri, S. S., et. al., 2010)

According to Muthukrishnan, R., et. al., (2011), Canny edge detection is a method that separate noise from the image to find edges of image. This method will not interfere the features of the edges in the image after it applying the tendency to find the serious value for threshold and the edge. This technique is selected to be implemented in this project because it is easy to be implemented and fast speed for the process. Besides, it is not sensitive to noise and thus can have better detection in noise condition. The algorithmic steps are as follows:

i. Convolve image, f(r, c) using a Gaussian function to obtain smooth image, f^(r, c). The formula is shown at below.

$$f\hat{}(r, c) = f(r,c) * G(r,c,6)$$

ii. Applying first difference gradient operator to compute edge strength, edge magnitude and direction by

iii. Applying critical suppression or non-maximal to the gradient magnitude.

iv. Applying threshold to the non-maximal suppression image

2.2.6 Image Segmentation based on Watershed Based Technique

One of the important Morphological techniques used to perform segmentation is Watershed Algorithm. It uses the concept of topological interpretation in which the gradient of image is considered as topographic surface and its gray levels of a pixel are interpreted by its altitude. Besides, the pixels with higher gradient are represented as boundaries which is also known as continuous. Watershed algorithm segments regions into catchment basins in which the adjacent basins will be merged together when water reaches the border of basin. Dams are required to maintain the separation between basins and these dams are constructed using dilation. The main idea of watershed based segmentation is to treat the regions to be extracted as catchment basins in topography.

According to Rizvi, I. A., et. al., (2011). There are two basic approaches to watershed image segmentation which are:

i. The first approach begins with the seeking of a downstream path from every pixel to a local minimum of the image surface altitude. Then, a catchment basin is determined as the set of pixels for which all corresponding downstream paths stop at same minimal altitude.

ii. The second approach is paired with the first one. A local minimum is recognized for each region, then the topographic surface is immersed in water. If two catchment basins would merge due to the further immersion, a dam will be built to avoid this.

2.2.6.1 Marker-Controlled Watershed

According to Rizvi, I. A., et. al., (2011), when the watershed algorithm is being applied to image segmentation, there are two main drawbacks occurs, which are sensitivity to high computational requirements to merge the over-segmented regions and strong noise. Marker-controlled segmentation was the strategy proposed by Meyer and Beucher to overcome these problems. This strategy is based on the idea that machine vision systems generally roughly ‘‘know’’ the location of the objects to be segmented from other sources. The concept of markers is a good way to control over segmentation problem.

Based on P.Jayapriya, et. al., (2019), image is converted into Grayscale before being segmented using Marker-Controlled Watershed Segmentation. Then reconstruction of image is made by opening and closing by reconstruction with selection of markers based on foreground and background objects, to found the definite boundaries that spatially locate object and background to ensure the interior of the object is kept up as a whole. This technique is selected to be implemented in this project because it can reduce and solve the over segmentation. Besides, it can reduce noise of output and has high accuracy when compared to some segmentation technique such as Sobel, Roberts, LoG, Canny, Otsu's thresholding segmentation and etc. In this process, markers act as the connected components in an image.

The proposed steps for Marker-Controlled Watershed Segmentation to be followed are given below:

i. Read the gradient image and apply closing and opening function by reconstruction.
   - Gray-scale Opening
     - The opening of a gray image f by a structuring element b. The formula is shown at below.

$$f \circ b = (f \Theta \mathrm{b}) \oplus \mathrm{b}$$

   - Gray-scale Closing
     - The closing of a gray image f by a structuring element b. The formula is shown at below.

$$f \bullet b = (f \oplus \mathrm{b}) \Theta \mathrm{b}$$

ii. Compute regional maximal gradient magnitude to obtain minima at the location of foreground and background marker
iii. Compute the foreground markers
iv. Compute the background markers
v. Compute the Watershed Transform using segmentation function.

2.2.7 Image Segmentation based on Clustering Based Technique

According to Kaur, D., et. al., Y, (2014), clustering based technique is the method which segments the image into clusters which have pixels with similar characteristics such as color texture, size and etc. It follows the concept of unsupervised learning. The quality of the results or output of this method depends on the similarity measure and the way of implementation.

This clustering method can be divided into two types which Partition based and Hierarchical method. Partition based method uses optimization methods repeatedly to minimize an objective function whereas the hierarchical method is based on the concept of trees.

Furthermore, various algorithms which consists of hard clustering and soft clustering are used in these two categories of method. For hard clustering, it is a simple clustering technique that partition or divide the image into set of clusters which means that one pixel can only belong to only one cluster. Therefore, sharp boundaries exist between different clusters. On the other hand, soft clustering is more natural because exact division is not possible in real life because of the presence of noise.

2.2.7.1 K-means Clustering

The most common and popular clustering algorithm used for image segmentation is k-means clustering. It is classified as one of the unsupervised algorithms and hard clustering based techniques that helps to improve performance and efficiency of image segmentation. This algorithm is used to estimate the number of clusters that depend on the values of pixels by aligning the pixels n into a set of clusters k, where k < n. The various pixels within an image are divided into k number of clusters based on certain image characteristics. (P.Jayapriya, et. al., 2019)

There are two separate phases for K-means algorithm. In the first phase, it calculates the k centroid. Then, it takes each point to the cluster which has nearest centroid from the respective data point in the second phase. This technique is selected to be implemented in this project because it is easy to be implemented and has high-speed performance. Besides, it can avoid the mixed pixel problem which happened on most of the pixel-based methods. According to Dhanachandra N., et. al., (2015), various steps are involved in the k-means algorithm as follows:

i. Initialize number of cluster k and centre.

ii. Calculate the Euclidean distance d, between the center and each pixel of an image using the relation shown below.

$$d= \| p(x,y)-c_k \|$$

iii. Assign all the pixels to the nearest centre based on the Euclidean distance d.

iv. Recalculate new position of the centre after all pixels have been assigned using the relation shown below.

$$c_k=\frac{1}{k}\sum_{y\in c_k}.\sum_{y\in c_k} \; p(x,y)$$

v. Repeat this process until the error value is satisfied.

vi. Restructure the cluster pixels into image.

2.3 Critical review of current problem and justification

Based on Sowmya, D. R., et. al. (2018) in their research paper with title "Land Use/Land Cover Segmentation of Satellite Imagery to Estimate the Utilization of Earth's Surface", they stated that LULC mapping plays an important role in many applications of land management, for example to create proper urban planning and community map. They have proposed Rate-Monotonic Scheduling (RMS) value based multi-thresholding method to segment different classes land cover or land use and implement the binning technique for the estimation of the utilization of earth's surface accurately. The proposed RMS algorithm are used to segment efficiently different land use/land cover classes likes trees, bare lands, buildings as well as water body.

Based on Bhadoria, P., et. al. (2020) in their research paper with title "Image Segmentation Techniques for Remote Sensing Satellite Images", they studied and analyzed which methods of image segmentation provides the best result for remote sensing satellite. Most of the researches that they have found for their study involved edge detection techniques. They concluded that Canny edge detection give better result compared to other methods of edge detection. However, it is not reliable for corner detection and sometimes gives the false detection, thus Harris is proposed to be used for the efficient corner detection. They have also studied other research that mention about Canny operator can be improved using morphological operator to reduces the effect of noise and return more accurate detection. However, Canny operator is still been called ‘Optimal Edge Detector’, it is more efficient to extract object because it can return fewer false edges. When comparing Canny and Sobel operators, Canny is more accurate than Sobel in the edge detection. Besides the choice of method for edge detection, the selection of parameter used to segment an image also plays an vital role in feature detection.

In research conducted by Malbog, M. A. F., et. al. (2020) with title “Edge Detection Comparison of Hybrid Feature Extraction for Combustible Fire Segmentation: A Canny vs Sobel Performance Analysis”, they proposed that Canny goes through five phases which are noise reduction, gradient computation, non-maximum suppression, double threshold as well as edge tracking by Hysteresis. There is another research discuss about Canny edge detection with title “Study of image segmentation by using edge detection techniques”, Abubakar, F. M. (2012) proposed that the Canny edge detection algorithm consists of four basic steps. First, it smooths the input image with a Gaussian filter. Second, it computes the gradient magnitude and angle images and then it applies non-maxima suppression to the gradient magnitude image. Lastly, it applies double thresholding to detect and link edges.

There are many research papers about Canny edge detection including the ones which have stated above, they all proposed similar steps or stages of Canny algorithm. Moreover, they also concluded that Canny result is the superior one when compared to the others such as Sobel, Robert, Prewitt and LoG edge detectors for the processed image since different edge detections work better in different conditions. A researcher Kirti, R., et. al. (2017) proposed that the Canny edge detection algorithm is the best and it is widely used method among the various edge detection because of its adjustable parameters which can affect the speed and effectiveness of the algorithm. There are many advantages for this technique, for example improved signal-to-noise ratio, good localization and rapid prototyping of the complex algorithms.

Based on P.Jayapriya, et. al. (2019) in their research paper with title “*Comparative Analysis of Image Segmentation Techniques And Its Algorithm*”, they have discussed and compare different segmentation techniques before starting to conduct the experiment of image segmentation. After that, they have used K-means, Canny Edge Detection, Fuzzy C-means, Neural Network, Morphological Watershed, Otsu’s Thresholding techniques to segment the same image.

The conclusion after the upcoming observation was Marker-Controlled Morphological Watershed Segmentation is superior in which the Marker-Controlled

Watershed segmentation helps to reduce and solve the over segmentation. Besies, it can reduce noise of output and its accuracy is high compared to Sobel, Roberts, LoG, Canny, active contour Morphological-based segmentation as well as Otsu's thresholding segmentation methods. Before performing segmentations on the image using Marker-Controlled algorithm, they have to input RGB image and converted it into grayscale. When applying this algorithm, the reconstruction of image involving the selection of markers is needed to be made first. Then, they only continued with the segmentation using Watershed Transform.

According to research conducted by Rizvi, I. A., et. al. (2011) with title "*Multi-Resolution Segmentation of High-Resolution Remotely Sensed Imagery using Marker-Controlled Watershed Transform*", they also proposed that the concept of markers has solved the over segmentation which is the main problem of watershed when applying directly to the gradient image. This is because it uses a set of morphological simplifications for detecting the presence of the homogenous regions from the image.

On the other hand, research conducted by Bhatia, S., et. al. (2007) in their research paper with title "Satellite Image Segmentation using Watershed based Algorithms" stated that the results obtained by the watershed method are usually severely over-segmented. They applied preprocessing that was geodesic reconstruction to the gradient image before using the watershed algorithm for solving this problem. First, the gradient image is reconstructed by erosion, then they used a modified version of reconstruction by closing. After making the observation before and after the geodesic reconstruction on the satellite image, they found that it has largely overcome the over-segmentation problem. It can be concluded that the preprocessing reconstruction to the gradient image is necessary to solve the over-segmentation problems since Marker-Controlled also involves this process.

In research conducted by Usman, B. (2013) with title "Satellite Imagery Land Cover Classification using K-Means Clustering", K-means clustering algorithm has been used to classify the satellite image into specific objects for Cadastral and Environmental Planning Purpose. The reason why they decided to use this algorithm is because it can avoid the mixed pixel problem which are suffered by most of the pixel-based methods and the accuracy percentage of its performance can reach as high as 88.889%. In this experiment, they have used a Quick Bird Satellite imagery with a 2.4m resolution for the classification based on three land-use classes including residential, agriculture and commercial.

Before started with the classification, the image was geo-rectified to remove some errors and make it to be assumed as a 2D plane surface. After that, K-means algorithms has been used for the satellite imagery classification that discrete objects with similar characteristics and assigned them into clusters. In this process, pattern representation and the definition of pattern proximity measure that appropriate to particular data domain are also involved. Lastly, this algorithm has also been used for color-base segmentation. For this process, marker with values have been defined and a similarity measure between a point and the neighboring marked region was found. This is the way how they define the color difference between the point and the neighbor within the markers. No post-processing is needed after the classification, thus they concluded that good results can be ensured using this algorithm.

According to research conducted by Mir, S. A., et. al. T. (2018) with title "Review about Various Satellite Image Segmentation", they have proposed different algorithms including Expectation Minimization (EM), Fuzzy C Means (FCM) as well as K-Means algorithm for the segmentation of satellite image. The research begins with a purpose which is to find out the methods that can overcome the problems happened during the satellite images segmentation such as data loss, mis-segmentation and etc. These three algorithms have been evaluated based on their Peak Signal-to-Noise Ratio (PSNR) value. Segmentation using K-means has the highest PSNR value where the higher the PSNR value, the better the result of the segmentation.

On the other hand, the explanation from the research conducted by Dhanachandra, N., et. al. (2015) make it more clear about the concept of K-means. they proposed that K-means is an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid for all clusters. The quality of the final results of K-means clustering depends on the arbitrary selection of initial centroid. Besides, the result of this clustering method relies on the number of data elements, the number of clusters as well as the number of iterations.

For software that being used in the previous research, MATLAB has been used to implement the Edge Detection, Marker-Controlled Watershed Transformation as well as K-means Clustering techniques for image segmentation in most of the previous researches that have been done. Different versions of the software have been used depending on the years that the researches were carried out. It provides strong mathematical and numerical support for the implementation of numerous algorithms as well as the functions for image processing. Thus, it allows user to debug and test the algorithms immediately without recompilation.

Besides, the image processing algorithms which are available under MATLAB seems to be more advanced compared with other image processing applications. This is why MATLAB is widely used in community of image processing and computer vision. Furthermore, research conducted by Rizvi, I. A., et. al. (2011) has used Code::Blocks IDE on Windows XP platform. It is a free C/C++ and Fortran IDE and it is designed to be fully configurable and very extensible. Its advances such as it is a great tool for beginning with programming as well as it is light and rather flexible, easy to travel with some excellent characteristics can be known from the users of this software. It allows them to implement the image processing efficiently with C++ language. The programming languages available in MATLAB is different from Code::Blocks IDE and MATLAB has his own language developed by MathWorks.

Digital single lens reflex camera and digital image scanner have been used by Mir, S. A., et. al. (2018). They used digital single lens reflex to capture the aggregated images to the segment these images. Besides, they used the digital image scanner to categorize and measure chalkiness for the evaluation of image information processing. Malbog, M. A. F., et. al. (2020, August) used cameras or other devices with photographic capability that able to detect and scan an image of flame. Digital image scanning is available in these devices which makes them to be used as the fire detection tool.

Moreover, there are several researches have used LISS (Linear Imaging Self Scanning Sensor) in their researches, LISS-II and III were used by them to acquire data from an Indian Remote Sensing Satellites (IRS). In this project, the satellite image of the Malacca land was acquired from Google Earth which currently uses the Landsat 8. Table 2.3 shows the comparative study of the satellite image segmentation techniques involved by the previous researches stated in critical review.

Table 2.3: Satellite Image Segmentation Techniques Involved by The Previous Researches Stated in Critical Review

Journal	Title	Year	Author	Image Segmentation Techniques							
				K-means clustering	Edge Detection	Fuzzy C-mean	Region Based	Thresholding	Watershed Segmentation	ANN method	Expectation Maximization
International Journal of Scientific & Technology Research Issn 2277-8616	Comparative Analysis of Image Segmentation Techniques and Its Algorithm	2019	P.Jayapriya, Dr. S.Hemalatha	X	X	X	X	X	X		
IOP Conference Series: Materials Science and Engineering	Image Segmentation Techniques for Remote Sensing Satellite Images	2020	Bhadoria, P., Agrawal, S., & Pandey, R.		X					X	
2020 11th IEEE	Edge Detection Comparison of	2020	Malbog, M. A. F.,		X						

Control and System Graduate Research Colloquium (ICSGRC)	Hybrid Feature Extraction for Combustible Fire Segmentation: A Canny vs Sobel Performance Analysis		Lacatan, L. L., Dellosa, R. M., Austria, Y. D., & Cunanan, C. F.								
International Journal of Engineering Research & Technology (IJERT) lssn 2278-0181	Study of image segmentation by using edge detection techniques	2012	Abubakar, F. M.		X						
International Journal of Techno-Management	Image Segmentation Using Canny	2017	Kirti, R., & Bhatnagar, A.		X						

Research lssn 2321-3744	Edge Detection Technique										
Proceedings of the International Conference & Workshop on Emerging Trends in Technology	Multi-resolution segmentation of high-resolution remotely sensed imagery using marker-controlled watershed transform	2011	Rizvi, I. A., Mohan, B. K., & Bhatia, P. R.						X		
International Conference on Soft computing and Intelligent Systems	Satellite Image Segmentation using Watershed based Algorithms	2007	Bhatia, S., & Saxena, K.						X		

Elixir International Journal of Computer Science and Engineering	Satellite imagery land cover classification using k-means clustering algorithm computer vision for environmental information extraction	2013	Usman, B.	X							
Indonesian Journal of Electrical Engineering and Computer Science Issn 2502-4752	Review about Various Satellite Image Segmentation	2018	Mir, S. A., & Padma, T.	X		X					X

Procedia Computer Science	Image Segmentation Using K - means Clustering Algorithm and Subtractive Clustering Algorithm	2015	Dhanachandra, N., Manglem, K., & Chanu, Y.	X							

2.4 Proposed Solution / Further Project

According to the previous researches, the techniques used for the land use segmentation in our project are Canny Edge Detection, Marker-Controlled Watershed and K-means clustering. Edge Detection has been chosen because it is generally used for seeking discontinuities in gray level images (key steps to segment the images) and Canny algorithms has been chosen because it is the superior one which can give better result compared to other methods of edge detection. Besides, Marker-Controlled Watershed has been chosen because it can reduce and solve the over segmentation which is the most common problem in image segmentation. Furthermore, K-means clustering has been chosen since it has high accuracy percentage for its performance and able to avoid the mixed pixel problem suffered by most of the pixel-based methods. Before perform image segmentation using all techniques mentioned above except for K-means clustering, image is needed to be converted into grayscale image. The platform used to conduct this project is MATLAB software since it enables users to debug and perform image processing algorithms efficiently. The best techniques for the land use segmentation is identified from the three techniques mentioned above.

2.5 Conclusion

This chapter discussed about the related works that have done by the previous researcher. Since one of the current problems of this project is lack of understanding about different segmentation techniques, it is necessary to study from the previous works so that it is possible to make decision to choose the techniques that want to be applied for the land use segmentation. After making critical reviews on some previous researches, Canny Edge Detection, Marker-Controlled Watershed and K-means clustering have been selected from the numerous techniques which exists nowadays for the land use segmentation of Malacca. This is because all of them have been recommended by the researchers that involved in the previous work. The best technique for the land use segmentation of Malacca can then be identified. Furthermore, what platform to be used for this project is the second problem, MATLAB is selected because of the recommendation in the previous researches.

For the next chapter, methodology that is going to be used in this project will be discussed. All activities involved in each phase from the start to the end of this project will be described clearly in this chapter.

CHAPTER 3

PROJECT METHODOLOGY

3.1 Introduction

There are four sub-chapter including introduction, methodology, project milestones and conclusion in this chapter. In this chapter, the methodology of project is discussed in details. Each stage of the methodology and every activity involved in each stage is described clearly. The methodology involves five processes which are Image Acquisition, Image Segmentation, Classification of Segments, Accuracy Assessment and Output Evaluation. The ways and flows to conduct the selected segmentation techniques which consists of Canny Edge Detection, Marker-Controlled Watershed and K-means clustering were discussed. This enables us to have a clear mind and basic concepts about the steps that are going to carry on throughout the project. This is to ensure the project can go smoothly from start to the end and can be completed within the time frame.

3.2 Methodology

The project methodology is vital to ensure the research is carried out according to the right and smooth procedure so that the project can be completed within the given time frame. There are six processes involved in the methodology which consists of Image Acquisition, Image Segmentation, Classification of Segments, Accuracy Assessment and Output Evaluation. Image segmentation is the main process that other processes run around it. Several techniques or methods have been used to conduct the land use segmentation of Malacca. Figure 3.1 shows the processes for the satellite image segmentation in this project.

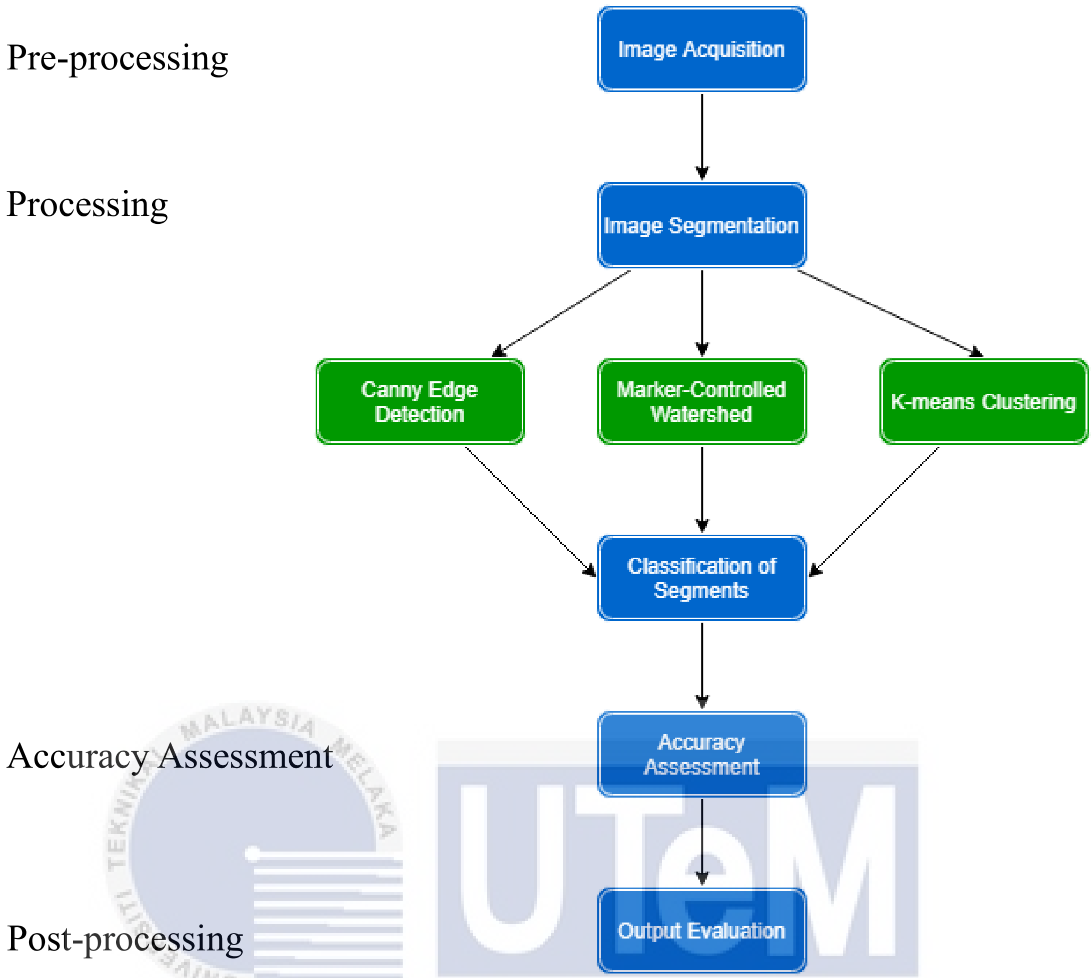


Figure 3.1: Project Methodology

Based on Figure 3.1, the first stage to be carried out for our project is the pre-processing which is image acquisition before the starting of satellite image segmentation. It is a process to retrieve the image which is used for land use segmentation from the selected sources. The satellite image of Malacca land was downloaded from the free-source platform that is Google Earth. Before starting to the processing, it is necessary to carry some steps to improve the quality of the image or enhances some features that is important for the further processing so that the analysis of the satellite data can be proceed well. It is a process to resize and remove noise of the satellite image.

The second stage in this project is processing which includes two processes. The first process of processing stage is image segmentation. There are three techniques which consists of Canny Edge Detection, Marker-Controlled Watershed and K-means clustering are used for the segmentation of Malacca satellite image. This process divides the satellite image into several segments based on different types of land use.

The output of this stage may allow us to distinguish between the different land use from the satellite image.

Next, the second process in processing stage which is classification of segments in the output image for each image segmentation technique was conducted. The numbers of segments can be known from the result, then try to classify the types of land use for each segment. It may spend time to identify and label the classes (types of land use) by referring the satellite image in Google Earth. Google Earth allows us to zoom in the satellite image to have a good and clear view like street view on the map of each segmented area.

Moreover, the third stage of the project is accuracy assessment. It is a process to analyze the output image of each technique and test its accuracy of segmenting and classifying different types of land use in the satellite image based on different image characteristics. Accuracy assessment is used to compare the classified image with a sample of actual points which is also known as ground truth. It provides important and essential information for the evaluation of the segmentation techniques.

Lastly, post-processing is the last stage in this project. In this stage, the output evaluation was conducted on the output of the three techniques which are Canny Edge Detection, Marker-Controlled Watershed and K-Means Clustering. This is to evaluate and determine the best output (segmented image) after undergoing the segmentation using these techniques. The advantages or disadvantages of all of the three techniques can be identified from the comparison of their outputs. Thus, the best technique for land use segmentation of satellite image can be identified.

Figure 3.2 is the flow chart that describes the entire flow of this project.

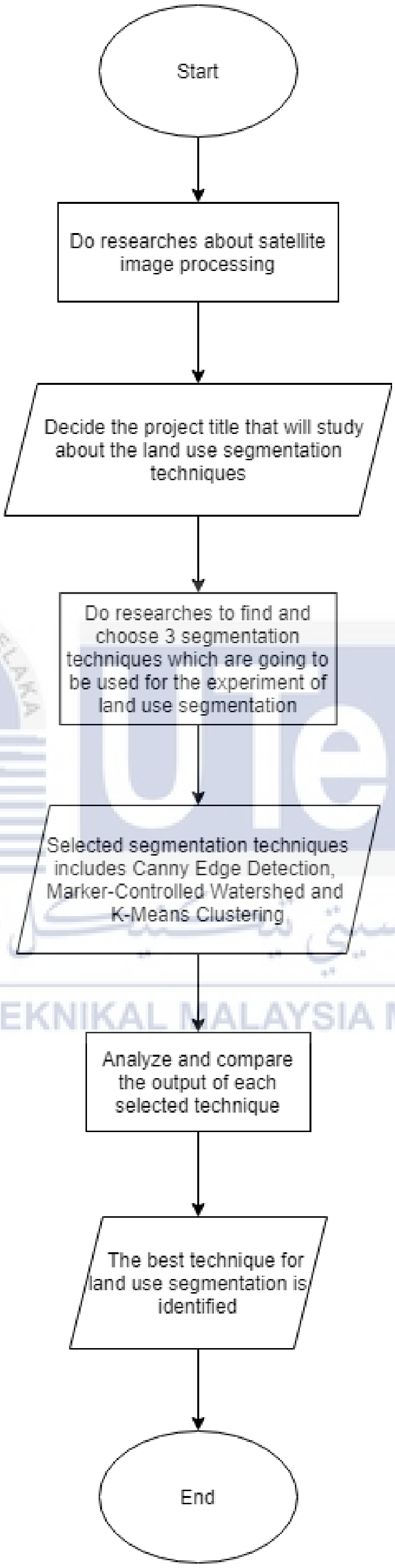


Figure 3.2: The flow of this project

The flow chart shown on the Figure 3.2 above has summarized the steps to proceed for this project. Based on the flow chart, did researches about the satellite image processing to understand more about it at the starting of the project. Then, decide the project title named as “Land Use Segmentation using image Edge Detection, Marker-Controlled Watershed And K-Means Clustering Techniques”. Next, do researches to find and choose three techniques which are used for land use segmentation. A lot of researches have been done until suitable techniques have been found. The selected segmentation techniques are Canny Edge Detection, Marker-Controlled Watershed as well as K-Means Clustering Techniques. After that, analysis and comparison between the outputs of the three techniques have been conducted. Lastly, the best technique for the land use segmentation is identified.

3.3 Project Milestones

The project milestone states about the activities of each project stage that should be completed according to the schedules. Setting up a project milestone is a very efficient way of project management. It estimates the progress toward the final goal of the project. Table 3.1 shows the milestone of this project.

Table 3.1: Project Milestone

PSM 1		
WEEK	ACTIVITY	NOTE / ACTION
< W0 **(< 15/3)**	Select a suitable project topic and potential Supervisor	• Action – Student
	Contact with supervisor to discuss about the details of the project.	• Action – Student
	Write and submit proposal to supervisor for checking	• Deliverable – Proposal
	Make correction according to supervisor comment and submit it again for approval	• Action – Student

	Proposal verification made by supervisor	• Action – Supervisor
	Proposal submission to Committee via email	• Action – Student
W 1 **(15/3 → 21/3)**	Proposal Approval	• Action – PSM/PD Committee
	List of Supervisor/Title	• Action – PSM/PD Committee
	Return the proposal with committee signature to student	• Action – Supervisor
	Proposal Submission via PSM ULearn	• Action – Student
	Chapter 1 (Research analysis begins)	
W2 **(22/3 → 28/3)**	Proceed with Chapter 1	• Action – Student
W3 **(29/3 → 4/4)**	Chapter 1 submission to supervisor for checking	• Action – Student, Supervisor
W4 **(5/4 → 11/4)**	Make correction for Chapter 1 based on supervisor's comment	• Action – Student
	Final check for Chapter 1	• Deliverable – Chapter 1 • Action –Supervisor
W5 **(12/4 → 18/4)**	Chapter 2 (Study on previous research papers for literature review)	• Action – Student
W6 **(19/4 → 25/4)**	Chapter 2 submission to supervisor for checking	• Action – Student, Supervisor
	Project Progress	• 1st Progress Presentation (PK 1) • Action – Student, Supervisor
	Student Status	• 1st Warning Letter

		• Action – Supervisor, PSM/PD Committee
W7 **(26/4 → 2/5)**	Make correction for Chapter 2 based on supervisor's comment	• Action – Student
	Final check for Chapter 2	• Deliverable – Chapter 2 • Action –Supervisor
	Chapter 3 (Develop project methodology)	• Action – Student
W8 **(3/5 → 9/5)**	Chapter 3 submission to supervisor for checking	• Action – Student, Supervisor
	Make correction based on supervisor's comment and do final check for Chapter 3	• Deliverable – Chapter 3 • Action – Student, Supervisor
W9 **(10/5 → 16/5)**	**MID SEMESTER BREAK**	
W10 **(17/5 → 23/5)**	Chapter 4 (Start Implementation of Project)	• Action – Student
	Project Progress	• 2nd Progress Presentation (PK 2) • Action – Student, Supervisor
	Student Status	• 2nd Warning Letter • Action – Supervisor, PSM/PD Committee
W11 **(24/5 → 30/5)**	Chapter 4 submission to supervisor for checking	• Action – Student, Supervisor
	Student status determination (Continue/Withdraw)	• Submission of student status to PSM/PD Committee • Action – Supervisor, PSM/PD Committee
	Make correction for Chapter 4 based on supervisor's comment	• Deliverable – Chapter 4 • Action – Student, Supervisor

W13 **(7/6 → 13/6)**	PSM1 Report Schedule the Presentation	• Action – Student, Supervisor • Action – PSM/PD Committee • Presentation Schedule
W14 **(14/6 → 20/6)**	PSM1 Report Project Demo Submission of Presentation Slides to Supervisor for checking	• Deliverable – Complete PSM1 Draft Report, Presentation Slides • Action – Student, Supervisor
W15 **(21/6 → 27/6)**	**FINAL PRESENTATION & PROJECT DEMOSTRATION** Submission of the PSM1 Report to Supervisor and Evaluator	• Action – Student, Supervisor, Evaluator, PSM/PD Committee
W16 **(28/6 → 4/7)**	**REVISION WEEK** Correction made on the draft report by referring the Evaluator and Supervisor's comments during the final presentation. Submit PSM1 Logbooks to PSM ULearn. Submit an EoS Survey form.	• Deliverable – Complete PSM1 Logbooks • Action – Student, Supervisor • EoS Survey • Action - Student
PSM 2		
W1 **(19/7 → 25/7)**	Make improvement for Chapter 4 based on supervisor's suggestion	• Deliverable – Chapter 4 • Action – Student, Supervisor
W2 **(26/7 → 1/8)**	Chapter 5 (Start Testing and Analysis)	• Deliverable – Chapter 5 • Action – Student
W3 **(2/8 → 8/8)**	Chapter 5	• Deliverable – Chapter 5 • Action – Student, Supervisor
	Student Status	• Warning Letter 1 • Action – Supervisor, PSM/PD Committee

W4 **(9/8 → 15/8)**	Chapter 5	• Deliverable – Chapter 5 • Action – Student, Supervisor
	Project Progress	• Deliverable – Chapter 5 • Action – Student, Supervisor
W5 **(16/8 → 22/8)**	Chapter 6 (Start Project Conclusion)	• Deliverable – Chapter 6 • Action – Student
	Chapter 5 & 6 submission to supervisor for checking	• Action – Student, Supervisor
	Student Status	• Warning Letter 2 • Action – Supervisor, PSM/PD Committee
	Presentation schedule	• Presentation Schedule • Action – PSM/PD Committee
W6 **(23/8 → 29/8)**	Project Demo PSM2 Draft Report	• Deliverable – Complete PSM2 Draft Report to SV & Evaluator • Action – Student, Supervisor, Evaluator
	Student status determination (Continue/Withdraw)	• Submission of student status to PSM/PD Committee • Action – Supervisor, PSM/PD Committee
W7 **(30/8 → 5/9)**	**FINAL PRESENTATION & PROJECT DEMONSTRATION**	• Final Presentation • Project Demontration • Action – Student, Supervisor, Evaluator
	Submission of the PSM Darft Report onto ULearn PSM2	• Deliverable – Complete PSM Draft Report • Action – Student, Supervisor
W8 **(6/9 → 12/9)**	**FINAL EXAMINATION WEEKS**	• Deliverable – Complete PSM2 Logbooks • Action – Student, Supervisor

	Correction made on the draft report by referring the Evaluator and Supervisor's comments during the final presentation. Submit PSM2 Logbooks to PSM2 ULearn. Submit an EoS Survey form.	• EoS Survey • Action - Student
W9 **(13/9 → 19/9)**	**INTER-SEMESTER BREAK** Submission for the final complete report onto the PSM2 ULearn	• Deliverable – Complete Final PSM Report, Complete PSM2 Logbooks, Plagiarism Report • Action – Student, Supervisor

Table 3.2 shows the Gantt Chart for PSM 1.

Table 3.2: Gantt Chart for PSM 1

No	Task	Week for PSM 1																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Prepare proposal																	
2	Improvement proposal																	
3	Chapter 1: Introduction																	
4	Chapter 2: Literature Review																	
5	Chapter 3: Methodology																	
6	Chapter 4: Implementation																	
7	Project Demonstration																	
8	PSM1 Report																	
9	Final Presentation PSM1																	
10	Correction on draft report based on supervisor's and evaluator's comments.																	
11	Submit PSM1 Logbooks																	

Table 3.3 shows the Gantt Chart for PSM 2.

Figure 3.3: Gantt Chart for PSM 2

No	Task	Week for PSM 2								
		1	2	3	4	5	6	7	8	9
1	Improvement of Chapter 4									
2	Chapter 5: Testing and Analysis									
3	Chapter 6: Project Conclusion									
4	Project Demonstration									
5	PSM2 Report									
6	Final Presentation PSM2									
7	Correction on draft report based on supervisor's and evaluator's comments.									
8	Submit PSM2 Logbooks									

3.4 Conclusion

All of the stages in the methodology and activities involved in each stage have been described clearly in this chapter. The purpose of project methodology is to allow us to understand how to carry out the project to achieve the goal. There are six processes involved in the project methodology which consists of involves Image Acquisition, Image segmentation, Feature Extraction, Image Evaluation and Accuracy Assessment.

Furthermore, project milestone has been created in this chapter for management purpose. This is prepared to remind ourselves what task should be done on what time for ensuring the smooth progress of the project to avoid project delays. All the activities in the milestone should be the possible tasks that can be achieved according to the timeline. Therefore, making a project milestone is an important knowledge or skill that should be handled.

The next chapter will discuss about the implementation of project. This chapter will discuss about the details such as hardware and software which are required as well as the procedures which will be carried out for each technique.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

There are five sub-chapter including introduction, project requirements, segmentation techniques algorithm, evaluation method and conclusion in this chapter. This chapter discussed and describe about the requirement and procedures for implementing the satellite image segmentation. Hardware and software required in this project are listed out. Besides, the steps to segment the satellite image using Canny Edge Detection, Marker-Controlled Watershed as well as K-Means Clustering Techniques are described. Codes running in MATLAB to carry out each segmentation techniques are explained and the output after applying each technique is put in section.

4.2 Project Requirement

The project requirements can be divided into two types which are software and hardware requirements. All the software and hardware used for the implementation of satellite image segmentation has been identified and installed. The specifications of the software and hardware are described as follows.

4.2.1 Software Requirements

There are three software used for the implementation of this project, they consist of MatlabR2021a, Microsoft Word 2019, Google Earth Pro and ArcMap 10.8. The function of each software is shown in Table 4.1 below.

Table 4.1: Software requirement of this project

Software	Functions
MatlabR2021a	This software is used to process the satellite image for segmenting the land based on different types of land use.
Microsoft Word 2019	This software is used for preparing proposal as well as report.
Google Earth Pro	This software is used to acquire satellite image for the implementation of project.
ArcMap	ArcMap is used to obtain classified data and ground truth for the accuracy assessment in the testing and analysis of project.

4.2.2 Hardware Requirements

In this project, there are only one hardware used to implement this project which is Lenovo Ideapad 320. All the installation of the software is done in this pc. The specifications and function of the hardware were summarized in Table 4.2 below.

Table 4.2: Hardware requirement of this project

Hardware	**Specifications & Functions**
Lenovo Ideapad 320 	**Specifications** Operating System: Windows 10 System Type: 64-bit Operating System. Processor: AMD FX-9800P RADEON R7, 12 COMPUTE CORES 4C+8G Processor Speed: 2.70 GHz Memory: 8.00 GB **Functions** Laptop is used to process the satellite image for land use segmentation, and writing proposal and report.

4.3 Satellite Image Segmentation

There are many techniques have been applied to segment the satellite image based on different types of land use. These techniques are Canny Edge Detection, Marker-Controlled Watershed as well as K-Means Clustering Techniques. In this chapter, all these techniques have been applied to segment the satellite image of Malacca which downloaded from Google Earth Pro based on the different types of land use. The purpose of this chapter is to obtain the output of each technique for the analysis and identification of the most suitable in the next chapter. Figure 4.1 shows the satellite image downloaded from Google Map.

Figure 4.1: Satellite Image Downloaded from Google Map

4.3.1 Canny Edge Detection Technique

Figure 4.2 shows the steps to segment satellite image using Canny Edge Detection technique by running code in Matlab.

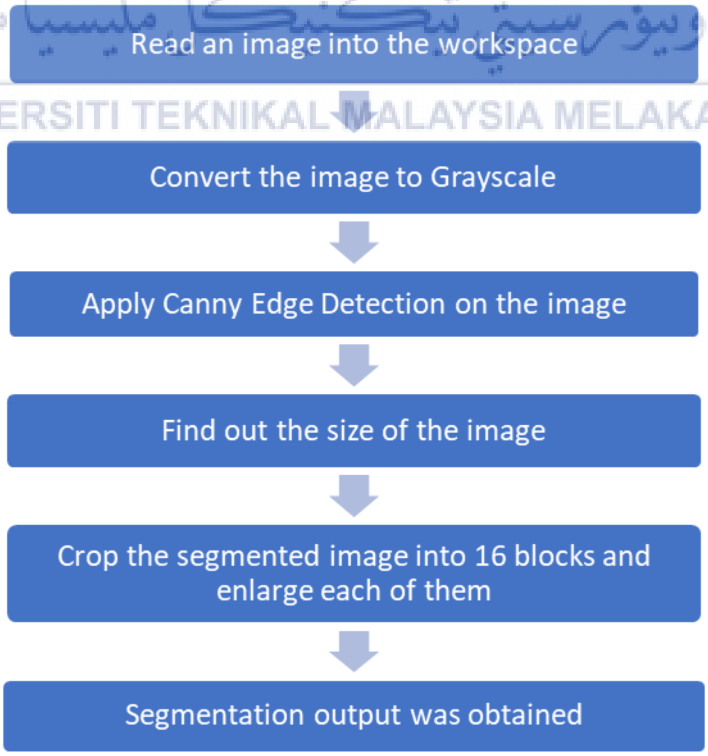


Figure 4.2: Steps to segment satellite image using Canny Edge Detection Technique

The whole process for the implementation of Canny Edge Detection segmentation was conducted on Matlab. Firstly, the image was read (imported) into the workspace of Matlab. Second, the image was converted to Grayscale since edge detection function only works with two-dimensional image. Third, the Canny Edge Detection segmentation was applied on the image but the edge cannot be seen clearly from the output image because the edge was too dense within high-resolution satellite image. In order to have a clear view of the output image, it was necessary to crop the image into 16 equal parts (4 rows and 4 columns) and enlarge each cropped part of the image. The image was cropped as shown in the Figure 4.7. First, it was necessary to find out the size (rows and columns) of the output image and then use imcrop function to crop and auto resize (magnify) the image. Since only the white point and black can be seen from the output image, Canny Edge Detection is concluded that it only segments the Malacca satellite image into two classes only which are edges and area. Figure 4.3 shows the coding to read or import the image.

```
% Read/import the image
m = imread ("malacca.jpg");
```

Figure 4.3: Coding for read or import the image

Figure 4.4 shows the coding to convert the RGB image into grayscale.

```
% Convert image to Grayscale
M = rgb2gray(m);
```

Figure 4.4: Coding for convert the RGB image into grayscale

Figure 4.5 shows the coding to apply Canny Edge Detection on the image.

```
% Apply Canny Edge Detection on the image and display output
canny = edge(M,'canny');
figure, imshow(canny), title('Image Processed with Canny Edge Detection')
```

Figure 4.5: Coding to apply Canny Edge Detection on the image

Figure 4.6 shows the output image of Canny Edge Detection after running code in Matlab.

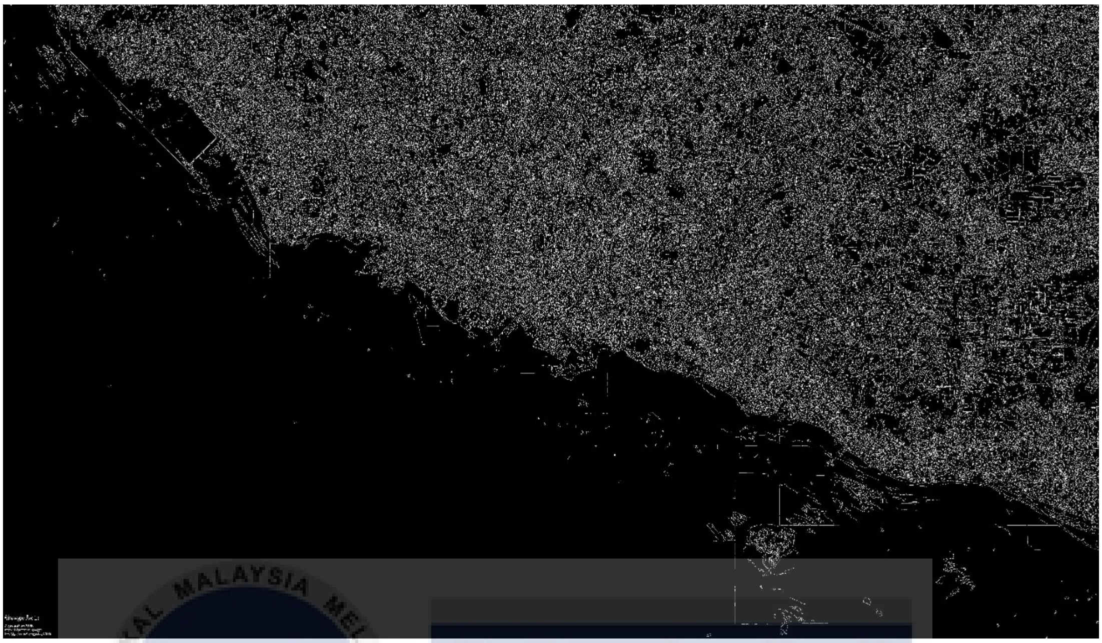


Figure 4.6: Output Image of Canny Edge Detection

Figure 4.7 shows the overview for the position of each cropped image.

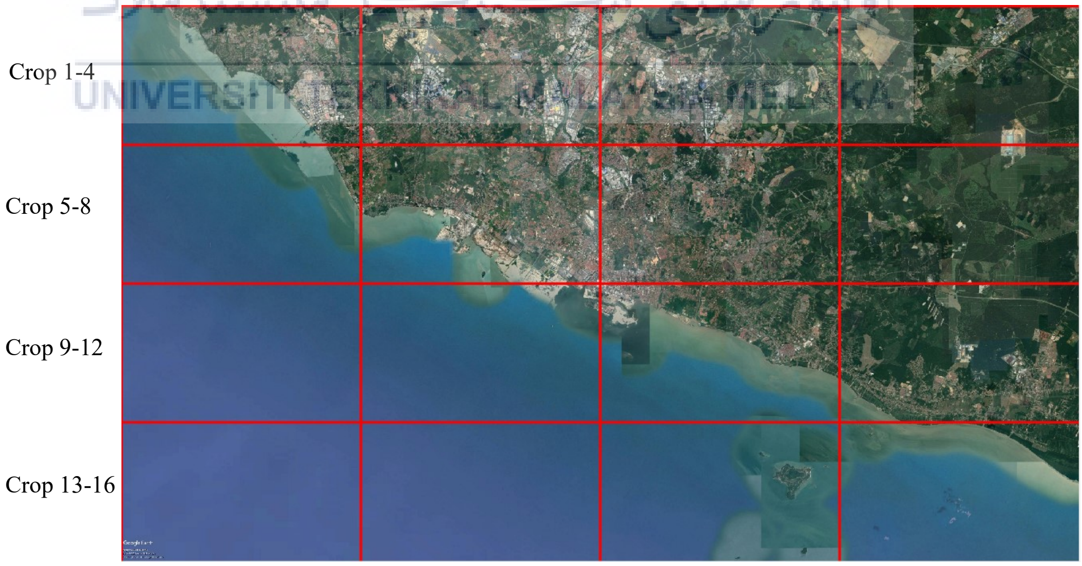


Figure 4.7: Overview for Position of Each Cropped Image

Figure 4.8 shows the coding to find out the size which are the rows & columns of the image.

```
% To find out the size of the output image
[rows, columns] = size(canny);
```

Figure 4.8: Coding to find out the size (rows & columns) of the image

Figure 4.9 shows the coding to crop the image into 16 equal cropped parts.

```
% Crops image into 16 equal parts and enlarge each cropped part (4 rows & 4 columns)
% Crops image into 4 columns for 1st rows
canny1 = imcrop(canny,[0 0 columns/4 rows/4]);
canny2 = imcrop(canny,[columns/4 0 columns/4 rows/4]);
canny3 = imcrop(canny,[columns/4*2 0 columns/4 rows/4]);
canny4 = imcrop(canny,[columns/4*3 0 columns/4 rows/4]);

% Crops image into 4 columns for 2nd rows
canny5 = imcrop(canny,[0 rows/4 columns/4 rows/4]);
canny6 = imcrop(canny,[columns/4 rows/4 columns/4 rows/4]);
canny7 = imcrop(canny,[columns/4*2 rows/4 columns/4 rows/4]);
canny8 = imcrop(canny,[columns/4*3 rows/4 columns/4 rows/4]);

% Crops image into 4 columns for 3rd rows
canny9 = imcrop(canny,[0 rows/4*2 columns/4 rows/4]);
canny10 = imcrop(canny,[columns/4 rows/4*2 columns/4 rows/4]);
canny11 = imcrop(canny,[columns/4*2 rows/4*2 columns/4 rows/4]);
canny12 = imcrop(canny,[columns/4*3 rows/4*2 columns/4 rows/4]);

% Crops image into 4 columns for 4th rows
canny13 = imcrop(canny,[0 rows/4*3 columns/4 rows/4]);
canny14 = imcrop(canny,[columns/4 rows/4*3 columns/4 rows/4]);
canny15 = imcrop(canny,[columns/4*2 rows/4*3 columns/4 rows/4]);
canny16 = imcrop(canny,[columns/4*3 rows/4*3 columns/4 rows/4]);
```


Figure 4.9: Coding to crop the image into 16 equal parts

Figure 4.10 shows the coding to display all the 16 equal cropped parts and Figure 4.11 – 4.26 are the images of all these parts.

```
% Display each cropped part
figure, imshow(canny1), title('Crop 1')
figure, imshow(canny2), title('Crop 2')
figure, imshow(canny3), title('Crop 3')
figure, imshow(canny4), title('Crop 4')
figure, imshow(canny5), title('Crop 5')
figure, imshow(canny6), title('Crop 6')
figure, imshow(canny7), title('Crop 7')
figure, imshow(canny8), title('Crop 8')
figure, imshow(canny9), title('Crop 9')
figure, imshow(canny10), title('Crop 10')
figure, imshow(canny11), title('Crop 11')
figure, imshow(canny12), title('Crop 12')
figure, imshow(canny13), title('Crop 13')
figure, imshow(canny14), title('Crop 14')
figure, imshow(canny15), title('Crop 15')
figure, imshow(canny16), title('Crop 16')
```

Figure 4.10: Coding to display the outputs for 16 equal cropped parts

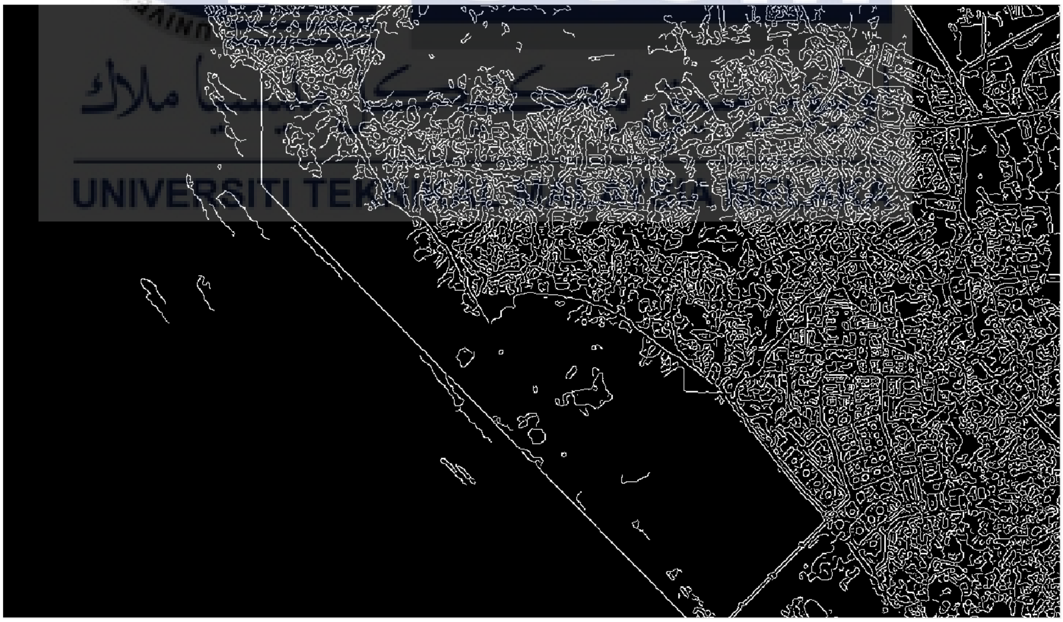


Figure 4.11: Output for Crop 1

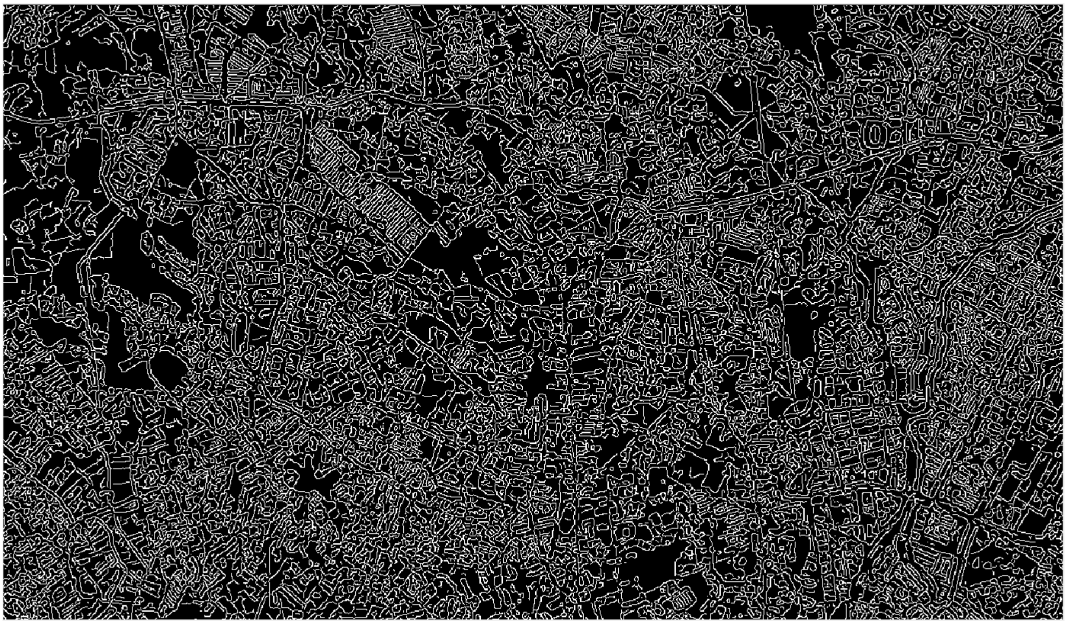


Figure 4.12: Output for Crop 2

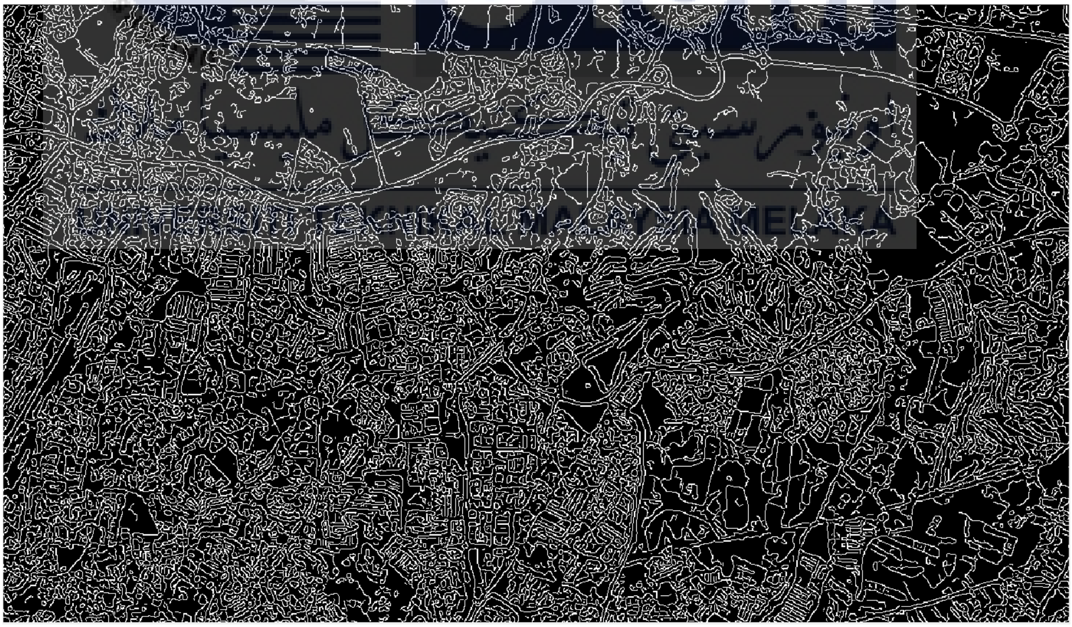


Figure 4.13: Output for Crop 3

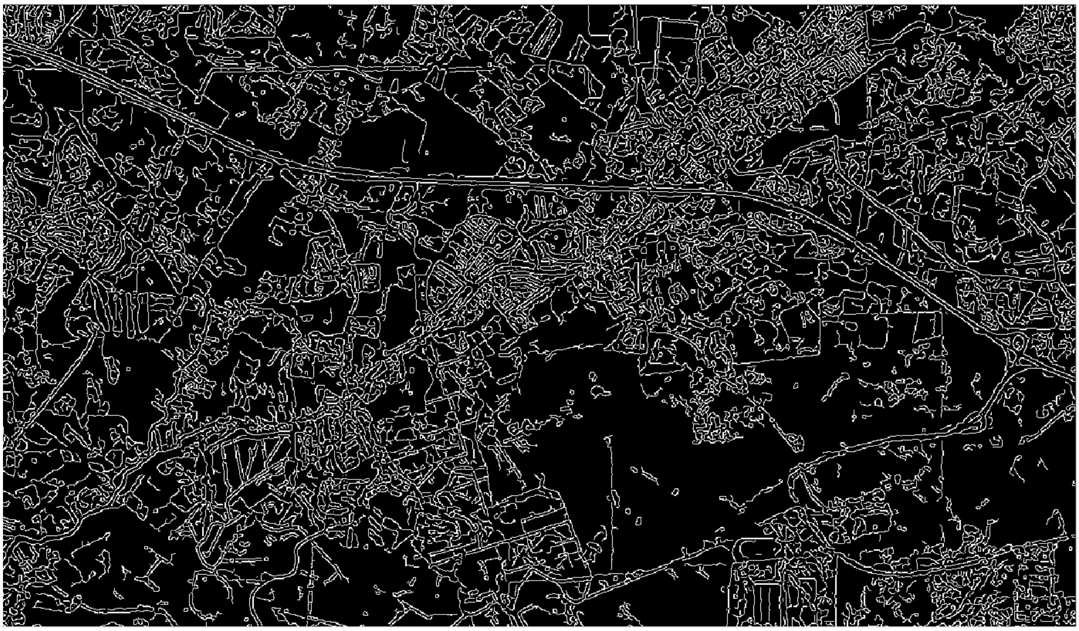


Figure 4.14: Output for Crop 4

Figure 4.15: Output for Crop 5

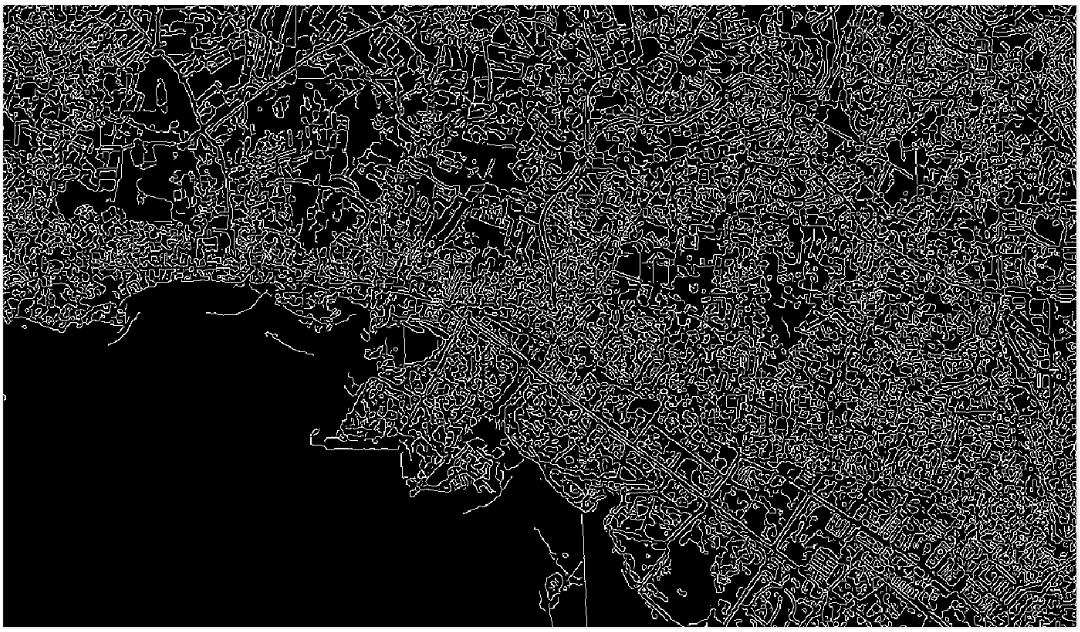


Figure 4.16: Output for Crop 6

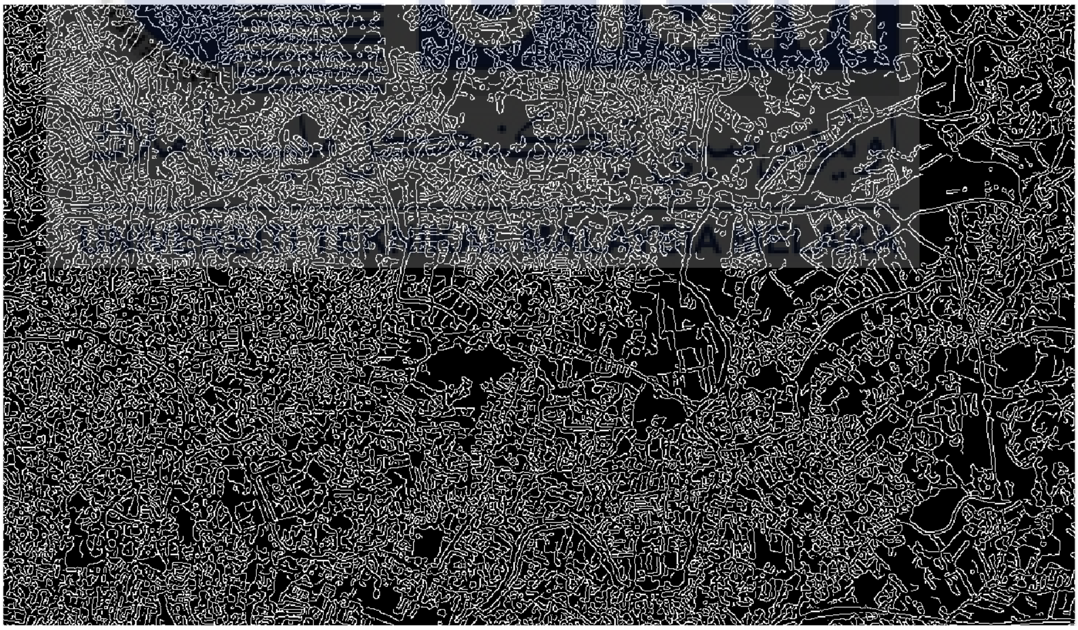


Figure 4.17: Output for Crop 7

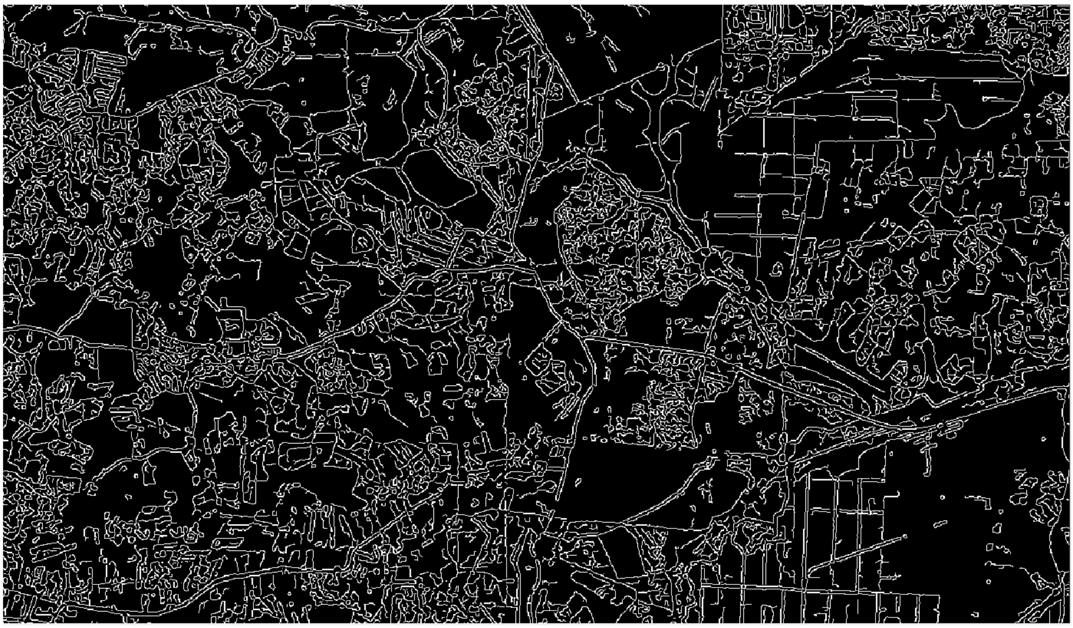


Figure 4.18: Output for Crop 8

Figure 4.19: Output for Crop 9 (Part of sea)

Figure 4.20: Output for Crop 10

Figure 4.21: Output for Crop 11

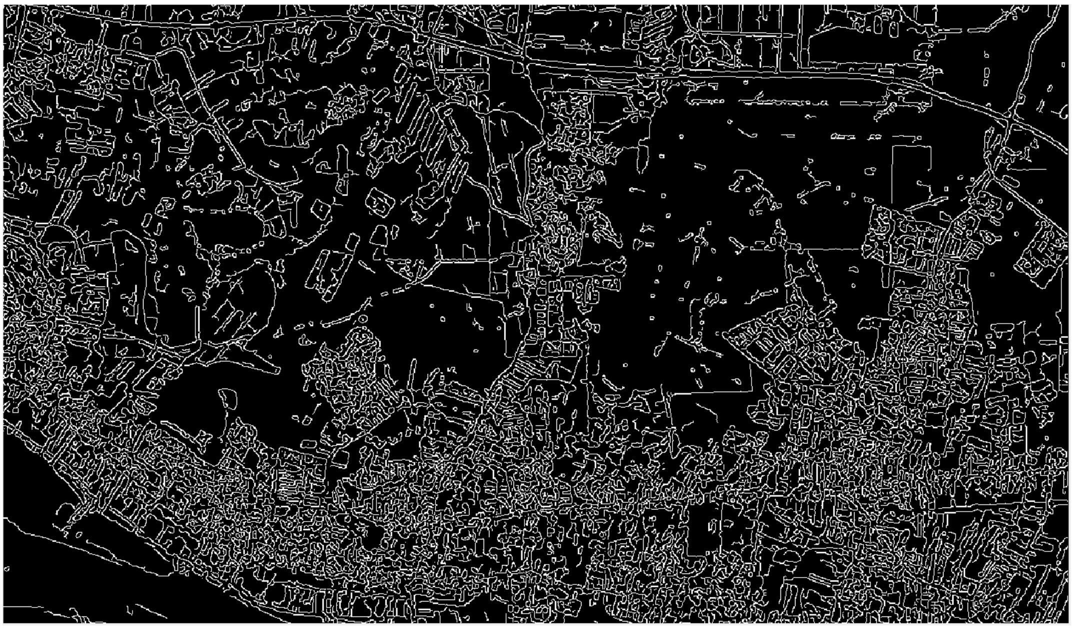


Figure 4.22: Output for Crop 12

Figure 4.23: Output for Crop 13 (Part of sea)

Figure 4.24: Output for Crop 14 (Part of sea)

Figure 4.25: Output for Crop 15 (Part of sea with small island)

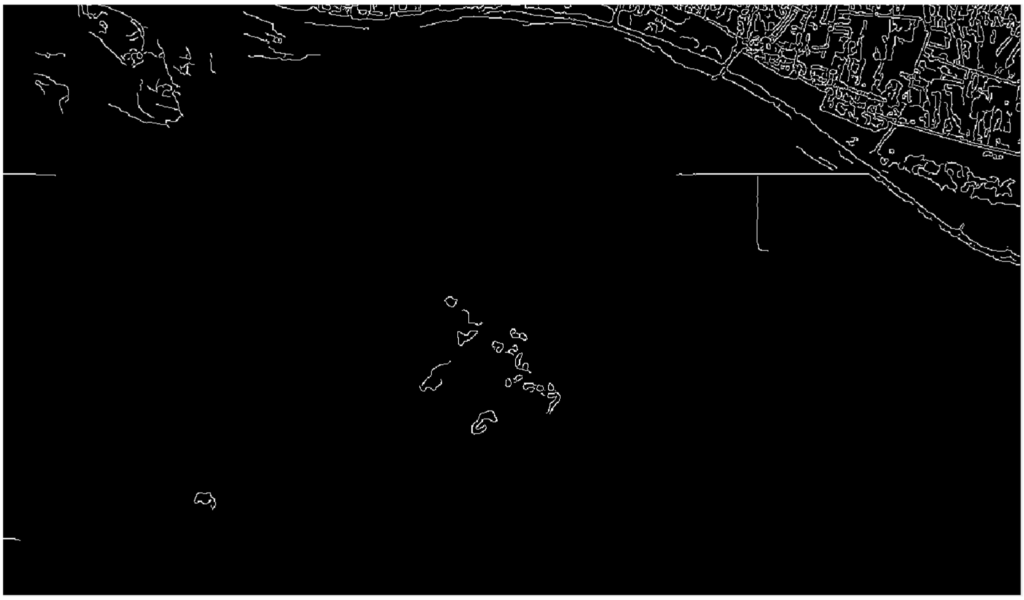


Figure 4.26: Output for Crop 16

4.3.2 Marker-Controlled Watershed Technique

Figure 4.27 shows the steps to segment satellite image using Marker-Controlled Watershed technique by running code in Matlab.

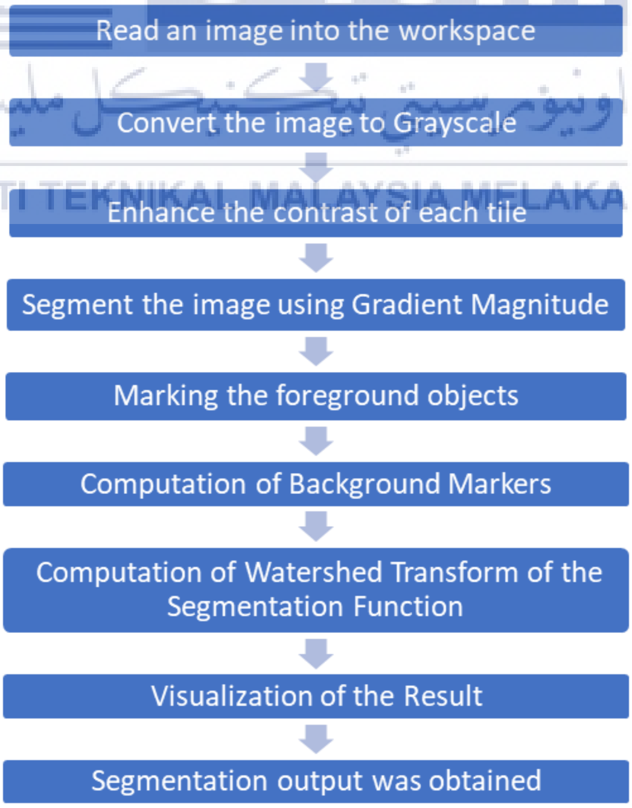


Figure 4.27: Steps to segment satellite image using Marker-Controlled Watershed Technique

The whole process for the implementation of Marker-Controlled Watershed segmentation was conducted in Matlab. The concept of marker here has made watershed transform segmentation works better because it can "mark" or identify the foreground objects and background locations. Firstly, the image was read (imported) into the workspace of Matlab. Secondly, the image was converted to Grayscale since gradient magnitude was used to detect the edges for segmentation and it only works with two-dimensional image. Thirdly, the contrast of each tile in the image was enhanced. Fourthly, the computation of gradient magnitude was proceeded to detect the edges in the image for segmentation. Then, an additional pre-processing was performed to prevent over-segmentation. Marker computations was carried out for the additional preprocessing, the foreground object was marked using opening-by-reconstruction. It is a morphological technique and clean out the image using closing-by-reconstruction. In this case, the comparison between opening-closing and opening-closing by reconstruction was made. Both opening and opening-by-reconstruction involves erosion but opening is followed by dilation after the erosion whereas opening-by-reconstruction is followed by morphological reconstruction.

The conclusion done from this comparison is the reconstruction-based opening and closing perform better in removing small flaws without disturbing the overall shapes of the objects compared to the standard opening and closing. In order to obtain good foreground markers, it was necessary to calculate the regional maxima of the output of reconstruction-based opening and closing. After that, the foreground was superimposed and some modification was made to ensure some of the shadowed and mostly-occluded objects were marked. Afterward, background markers were computed by staring with a thresholding operation. In order to make sure the background markers are not too close to the edges of the objects that wanted to segment, the watershed transform of the distance transform of the background markers was computed, and then the watershed ridge lines of the result were sought by treating the image as a surface with high and low levels where high level is for light pixels and low level is for dark pixels. Then, the modification to the gradient magnitude image was carried out and the Watershed Transform of the segmentation function was computed.

Lastly, the result was visualized using dilation to make certain aspects to be more visible. During the visualization, segmented object boundaries, foreground markers and background markers on the image were superimposed. Besides, another technique for visualization was to assign different colors to regions with different values inside the label matrix. The label matrix had been produced by watershed, then it was displayed as a color image. In order to have a good and clear view on the output for further study, the label matrix was superimposed transparently on the top of the original intensity image. From the output image, it concludes that the Marker-Controlled Watershed can segment the image into three classes which are land and coastal area, sediment and deep sea. Figure 4.28 shows the coding for read or import the image.

```
% Read/import the image
m = imread("malacca.jpg");
```

Figure 4.28: Coding for read or import the image

Figure 4.29 shows the coding for convert the satellite image to grayscale and enhance it.

```
% Convert image to Grayscale
M = rgb2gray(m);
% Enhances the contrast of each tile
M = adapthisteq(M);
```

Figure 4.29: Coding for convert the image to grayscale and enhance it

Figure 4.30 shows the coding to segment the image using gradient magnitude.

```
% Use gradient magnitude as segmentation fucnction to detect edges
gradmag = imgradient(M);
imshow(gradmag,[]), title('Gradient Magnitude')
```

Figure 4.30: Coding to segment the image using gradient magnitude (detecting edges)

Figure 4.31 shows the output image of gradient magnitude.

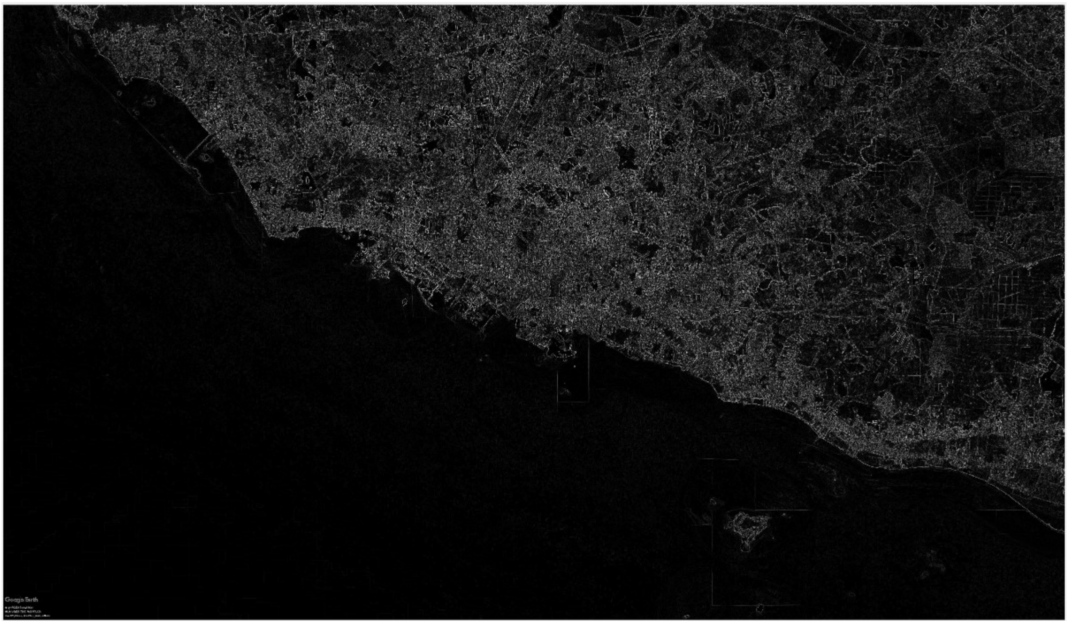


Figure 4.31: Output for Gradient Magnitude

Figure 4.32 shows the coding for the whole process of applying and comparing the opening and opening-by-reconstruction, as well as marking the foreground objects.

```
% Compare reconstruction-based, standard opening and closing
% Use the best opening and closing to mark the foreground objects
% Compute the standard opening
se = strel('disk',95);
Mo = imopen(M, se);
figure, imshow(Mo), title('Opening')
% Compute the opening-by-reconstruction
Me = imerode(M, se);
Mobr = imreconstruct(Me, M);
figure, imshow(Mobr), title('Opening-by-reconstruction')
% Close based on opening
Moc = imclose(Mo, se);
figure, imshow(Moc), title('Opening-closing')
% Close based on open imresconstruct
Mobrd = imdilate(Mobr, se);
Mobrcbr = imreconstruct(imcomplement(Mobrd), imcomplement(Mobr));
Mobrcbr = imcomplement(Mobrcbr);
figure, imshow(Mobrcbr), title('Opening-closing by reconstruction')
% Calculate the regional maxima to obtain good foreground markers
fgm = imregionalmax(Mobrcbr);
figure, imshow(fgm), title('Regional maxima of opening-closing by reconstruction')
% Superimpose the foreground marker image on the original image
M2 = labeloverlay(M,fgm);
figure, imshow(M2), title('Regional maxima superimposed on original image')
% Clean the edges of the marker blobs and then shrink them a bit using erosion
se2 = strel(ones(1,1));
fgm2 = imclose(fgm, se2);
fgm3 = imerode(fgm2, se2);
% Removes blobs having fewer than a certain number of pixels
fgm4 = bwareaopen(fgm3, 20);
M3 = labeloverlay(M,fgm4);
figure, imshow(M3), title('Modified regional maxima superimposed on original image')
```

Figure 4.32: Coding to apply opening and opening-by-reconstruction, and marking the foreground objects

Figure 4.33 – 4.39 shows the output image for each process during the marking of foreground objects.

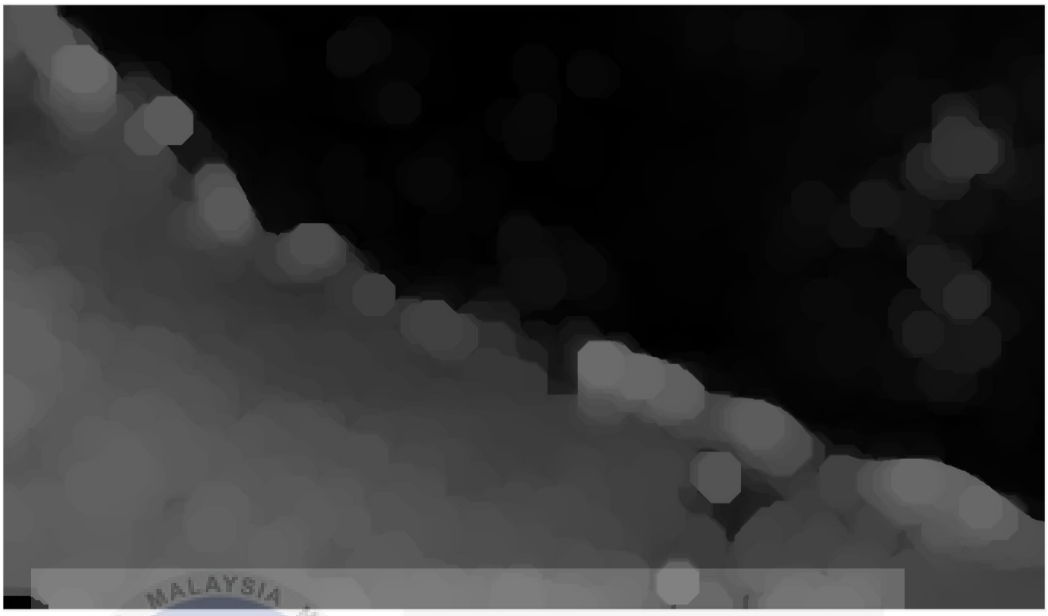


Figure 4.33: Output for Opening

Figure 4.34: Output for Opening-by-reconstruction

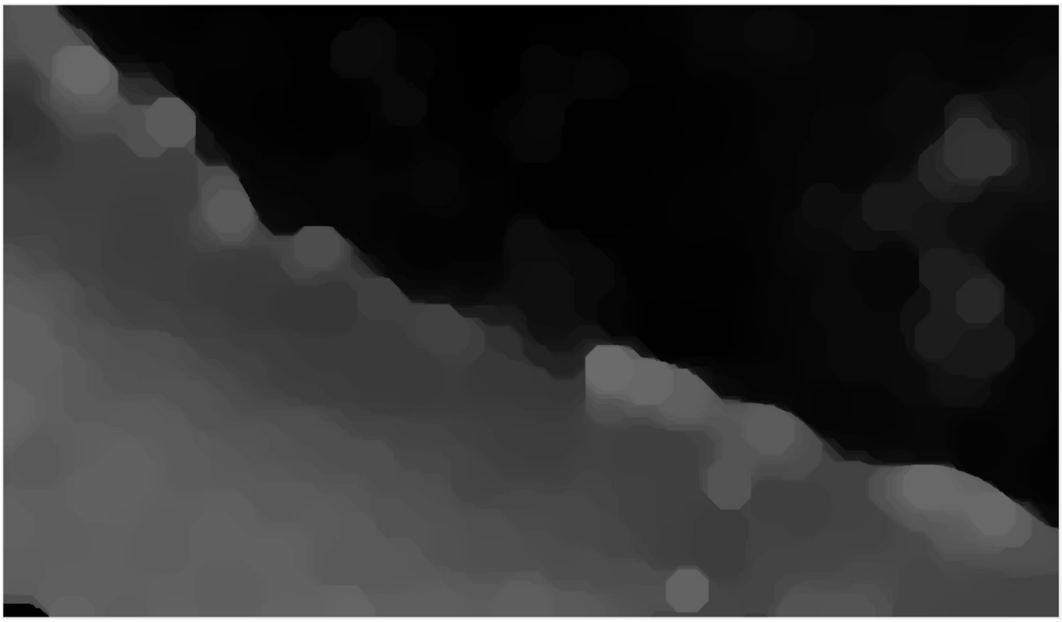


Figure 4.35: Output for Opening-Closing

Figure 4.36: Output for Opening-Closing by reconstruction

Figure 4.37: Output for regional maxima of Opening-Closing by reconstruction

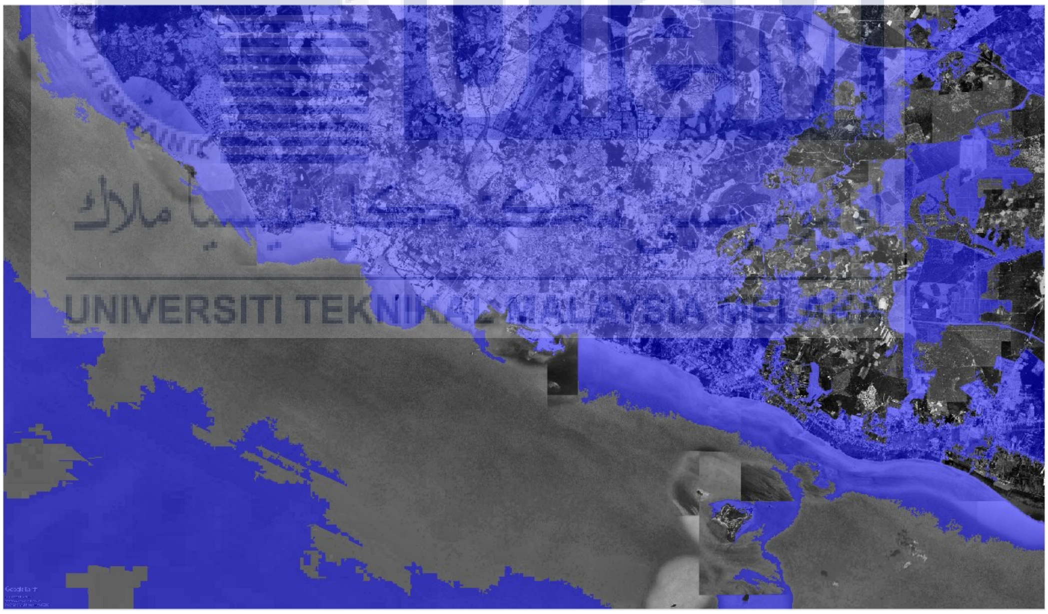


Figure 4.38: Output for regional maxima Superimposed on original image

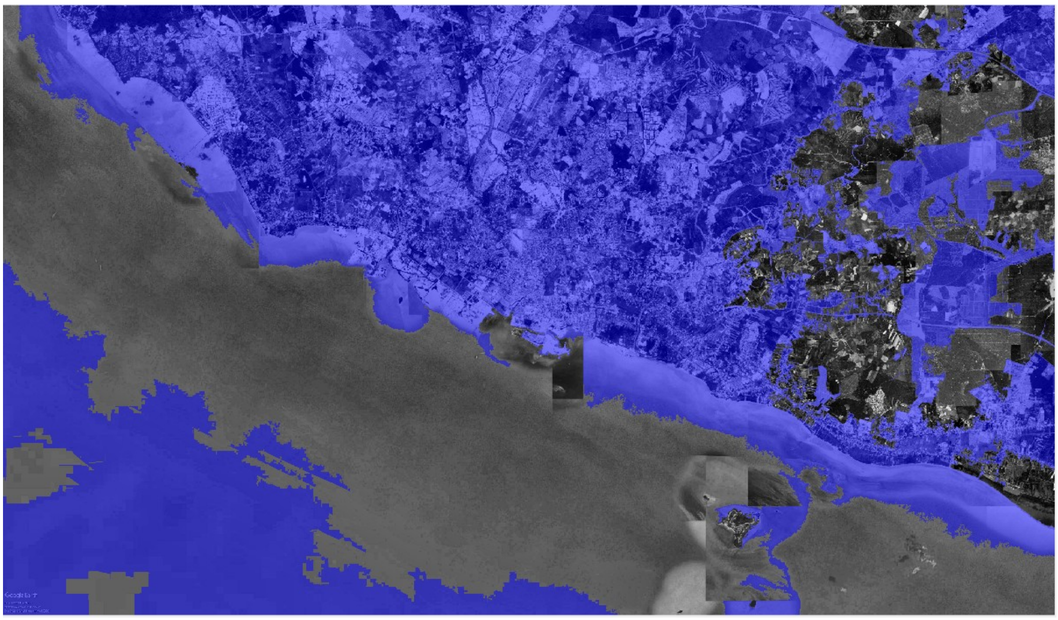


Figure 4.39: Output for modified regional maxima superimposed on original image

Figure 4.40 has shown the coding for computing the background markers.

```
% Compute background markers
% Thresholding operation
bw = imbinarize(Mobrcbr);
figure, imshow(bw), title('Thresholded opening-closing by reconstruction')
% Compute the watershed transform of the distance transform of bw
D = bwdist(bw,'euclidean');
DL = watershed(D);
% Looking for the watershed ridge lines of the result
bgm = DL == 0;
figure, imshow(bgm), title('Watershed ridge lines')
```

Figure 4.40: Compute Background Markers

Figure 4.41 – 4.42 shows the output image for each process during the computation of background objects. Figure 4.41 shows the output image of thresholded opening-closing by reconstruction.

Figure 4.41: Output for Thresholded Opening-Closing by reconstruction

Figure 4.42 shows the Watershed Ridge lines.

Figure 4.42: Output for Watershed Ridge lines

Figure 4.34 has shown the coding to compute the Watershed Transform of the Segmentation Function.

```
% Modify the gradient magnitude image
gradmag2 = imimposemin(gradmag, bgm | fgm4);
% Compute the Watershed Transform segmentation
L = watershed(gradmag2);
```

Figure 4.43: Compute the Watershed Transform of the Segmentation Function

Figure 4.44 shows the coding to visualize the segmented result with different way.

```
% Visualize the result
% Superimpose the markers and segmented object boundaries
labels = imdilate(L==0,ones(3,3)) + 2*bgm + 3*fgm4;
M4 = labeloverlay(M,labels);
figure, imshow(M4), title('Markers and object boundaries superimposed on original image')
% Assigns colors to regions in the label matrix
[m, n] = size (L);
area1 = zeros(m,n,3);
area2 = zeros(m,n,3);
area3 = zeros(m,n,3);
for i=1:m
      for j=1:n
          if L(i,j)== 1
                area1(i,j,1) = 255;
                area1(i,j,2) = 0;
                area1(i,j,3) = 0;
          end
          if L(i,j)== 2
                area2(i,j,1) = 0;
                area2(i,j,2) = 255;
                area2(i,j,3) = 255;
          end
          if L(i,j)== 3
                area3(i,j,1) = 0;
                area3(i,j,2) = 255;
                area3(i,j,3) = 0;
          end
      end
end
allclass = area1 + area2 + area3;
figure, imshow(allclass)
title('Colored watershed label matrix')
% Use transparency to superimpose label matrix on top of the original image

figure, imshow(M), hold on
himage = imshow(allclass);
set(himage, 'AlphaData', 0.4);
% title('Colored Labels superimposed transparently on original image')
```

Figure 4.44: Visualize the result

Besides, Figure 4.45 – 4.47 shows the output image for different way of visualization. Figure 4.45 shows the output image in which markers and object boundaries are superimposed on original image. Figure 4.46 shows the output image in which the watershed Label matrix are colored and Figure 4.47 shows the output image in which colored labels are superimposed transparently on original image.

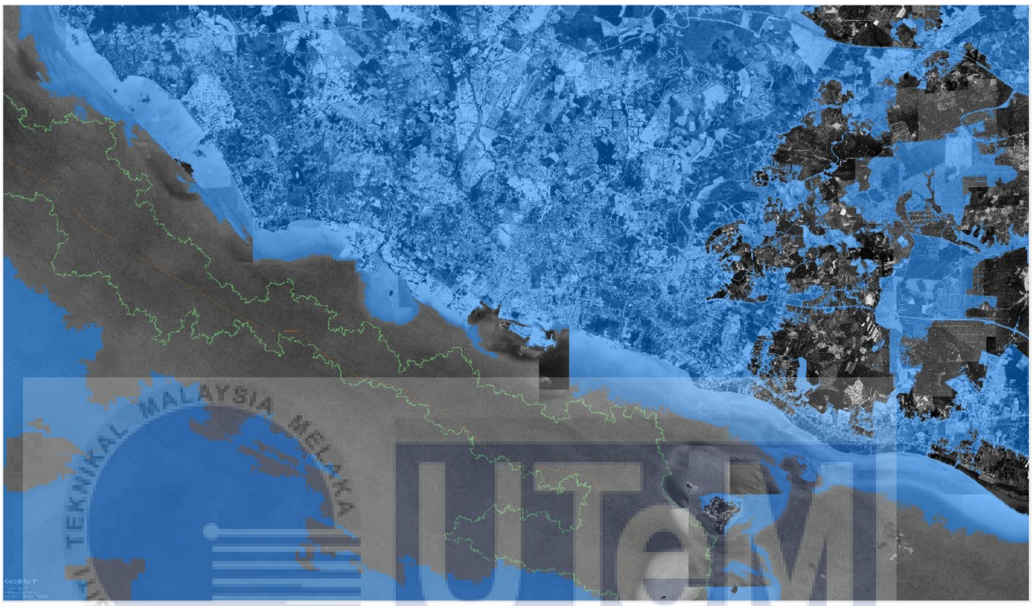


Figure 4.45: Markers and Object boundaries superimposed on original image

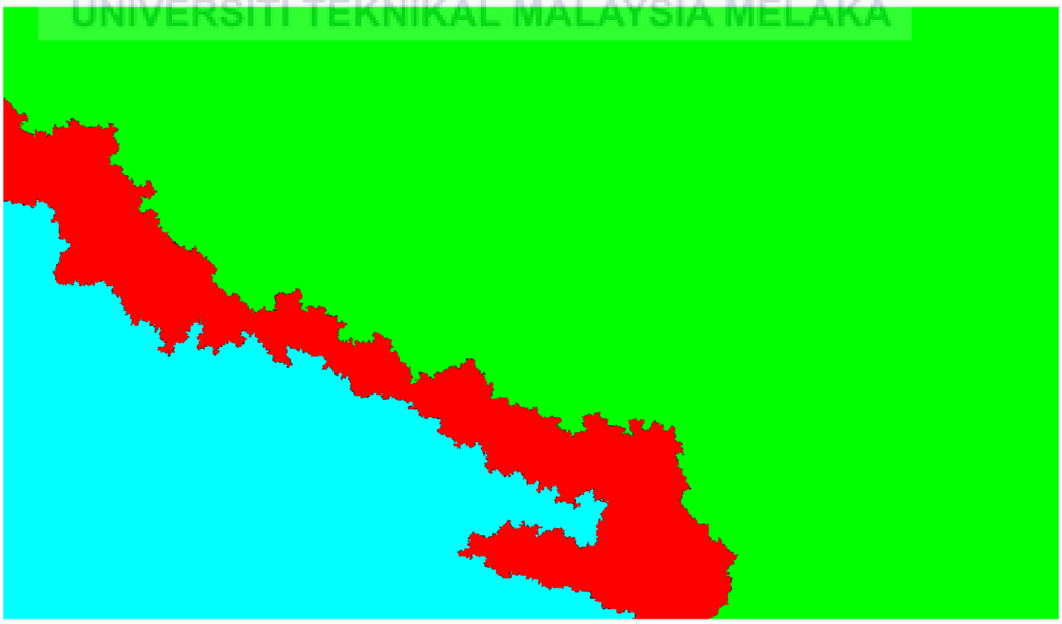
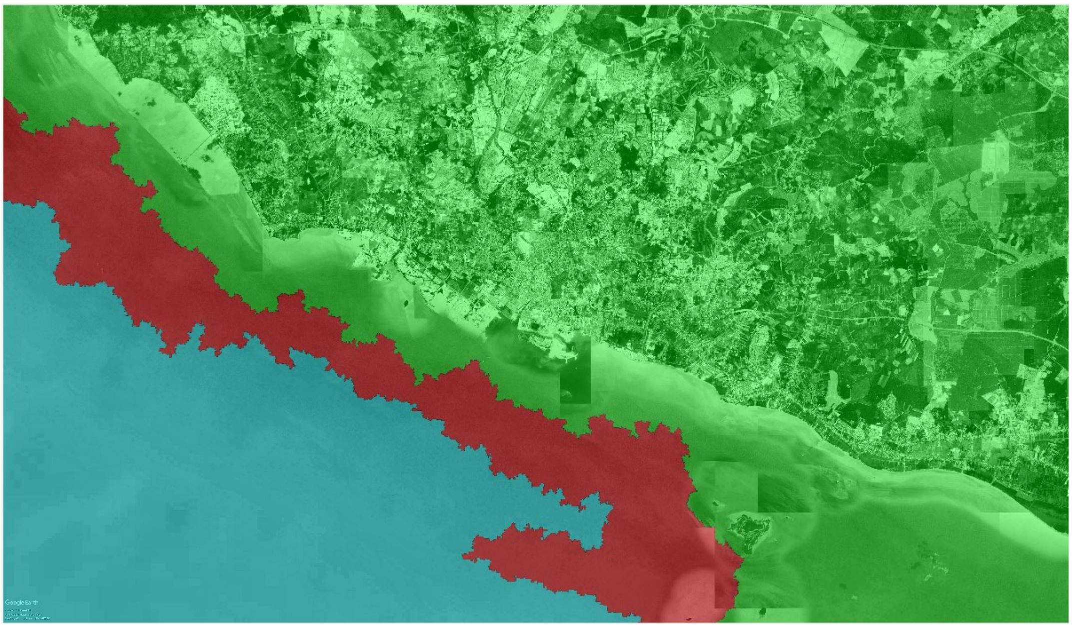


Figure 4.46: Colored Watershed Label Matrix



Legends:

Land & Coastal Area

Sediment

Deep Sea

Figure 4.47: Colored Labels superimposed transparently on original image

4.3.3 K-Means Clustering Techniques

Figure 4.48 shows the steps to segment satellite image using K-Means Clustering technique by running code in Matlab

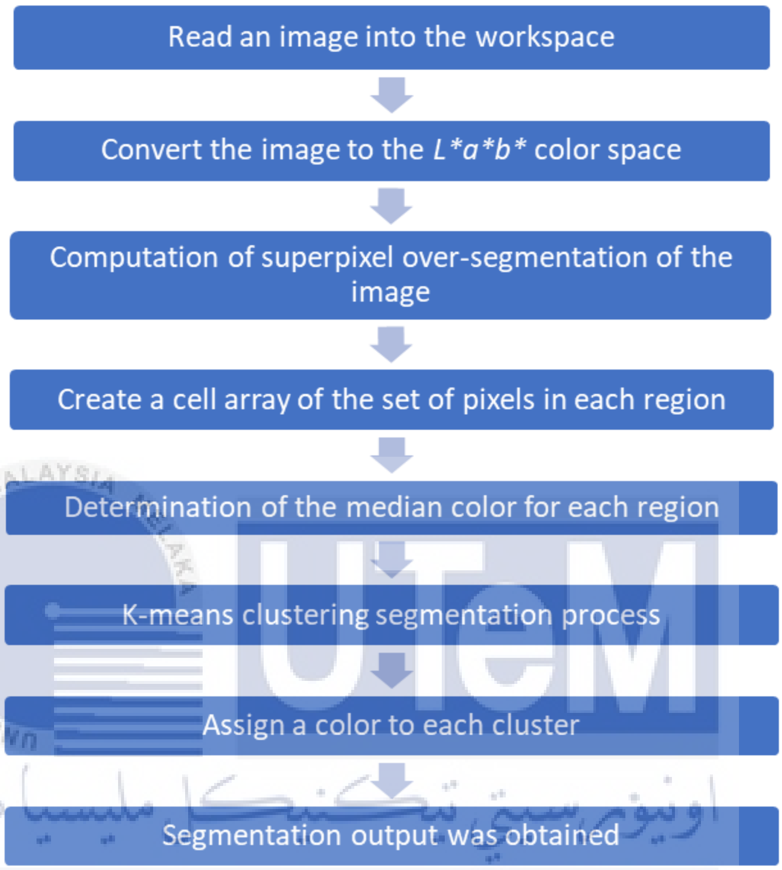


Figure 4.48: Steps to segment satellite image using K-means clustering

The whole process for the implementation of K-Means Clustering segmentation was conducted on Matlab. At the beginning of the process, the image was read (imported) into the workspace of Matlab. Secondly, the image was converted to the L*a*b* color space where L* stand for the perceptual lightness and a* and b* stand for the red, green, blue and yellow colors. Thirdly, the superpixel over-segmentation of the original image was computed. Superpixels was applied to the image first before performing K-means clustering. It is a technique used to divide the image into a set of structurally meaningful regions. The edge information in the original image is taken into consideration as the boundaries between each. After performing superpixels, algorithms only can be used to classify each region without having any classification problems over the grid of the full original image. Fourthly, a cell array was created to store the set of pixels in each region (segment). Fifthly, the median color for every superpixel region in the L*a*b color space was determined using the mean () function of Matlab. Lastly, the imsegkmeans function was used to cluster the color feature of each superpixel on the image. In this case, 6 clusters were divided since six types of land use including residential area, green spaces, commercial and industrial area, sediment, coastal area, and deep sea are wanted to be distinguished.

Each cluster was assigned with a color. Commercial area includes buildings that house business and land that generate profit. Some examples for commercial buildings are offices, company, clinics, farm land, hotels, malls, warehouses, school and etc. Furthermore, green spaces include area of grass, trees and vegetation, for example forests, shrubs, street trees and parks. Sediment can be considered as ocean floor sediment which are formed with insoluble particles likes soil and rocks that have been accumulated on the seafloor. Figure 4.49 shows the coding for read or import the image.

```
% Read/import the image
M = imread('malacca.jpg');
```

Figure 4.49: Coding for read or import the image

Figure 4.50 shows the coding for converting the image to the L*a*b color space.

```
% Convert the image to the L*a*b* color
Mlab = rgb2lab(M);
```

Figure 4.50: Coding for converting the image to the L*a*b color space

Figure 4.51 shows the coding to compute the superpixel over-segmentation of the image and display the output.

```
% Compute the superpixel oversegmentation of the image
[L,N] = superpixels(Mlab,20000,'isInputLab',true);
BW = boundarymask(L);

% Display it
figure,imshow(imoverlay(M,BW,'green')), title('Superpixel oversegmentation')
```

Figure 4.51: Coding to compute the superpixel over-segmentation of the image and display the output

Figure 4.52 shows the output image of superpixel over-segmentation of the image.

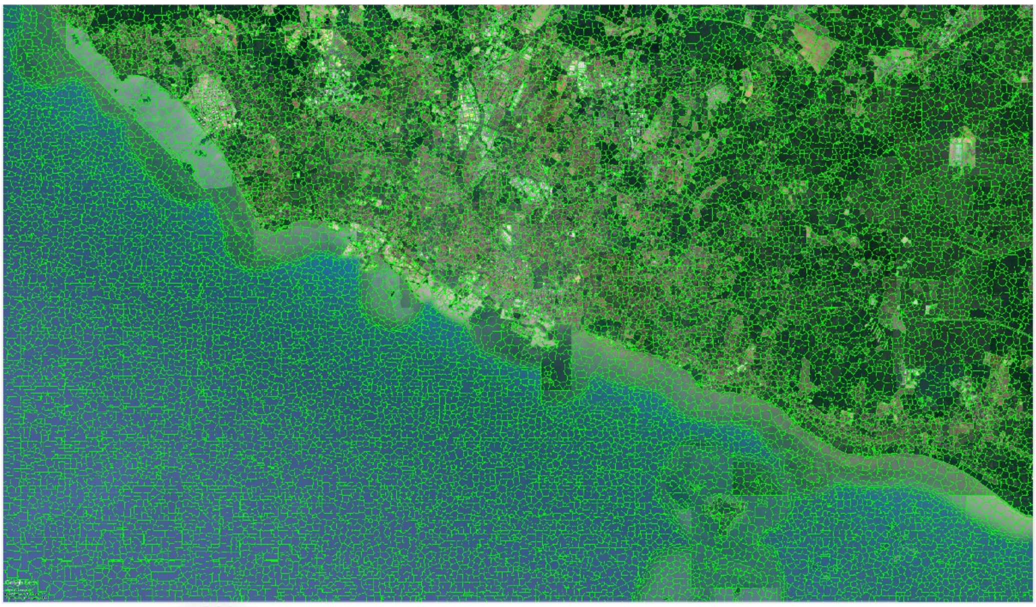


Figure 4.52: Output of superpixel over-segmentation of the image

Figure 4.53 shows the coding to create a cell array and determine the median color of each region.

```
% Create a cell array for the set of pixels
pxIdxList = label2idx(L);

% Determine the median color of each region in the L*a*b* color
[m,n] = size(L);
meanColor = zeros(m,n,3,'single');
for i = 1:N
    meanColor(pxIdxList{i}) = mean(Mlab(pxIdxList{i}));
    meanColor(pxIdxList{i}+m*n) = mean(Mlab(pxIdxList{i}+m*n));
    meanColor(pxIdxList{i}+2*m*n) = mean(Mlab(pxIdxList{i}+2*m*n));
end
```

Figure 4.53: Coding to create a cell array and determine the median color of each region

Figure 4.54 shows the coding to use K-means for clustering the color property of each superpixel and display output.

```
% Use k-means function to cluster the color property of each superpixel
numCluster = 6;
Lout = imsegkmeans(meanColor,numCluster,'numAttempts',2);
```

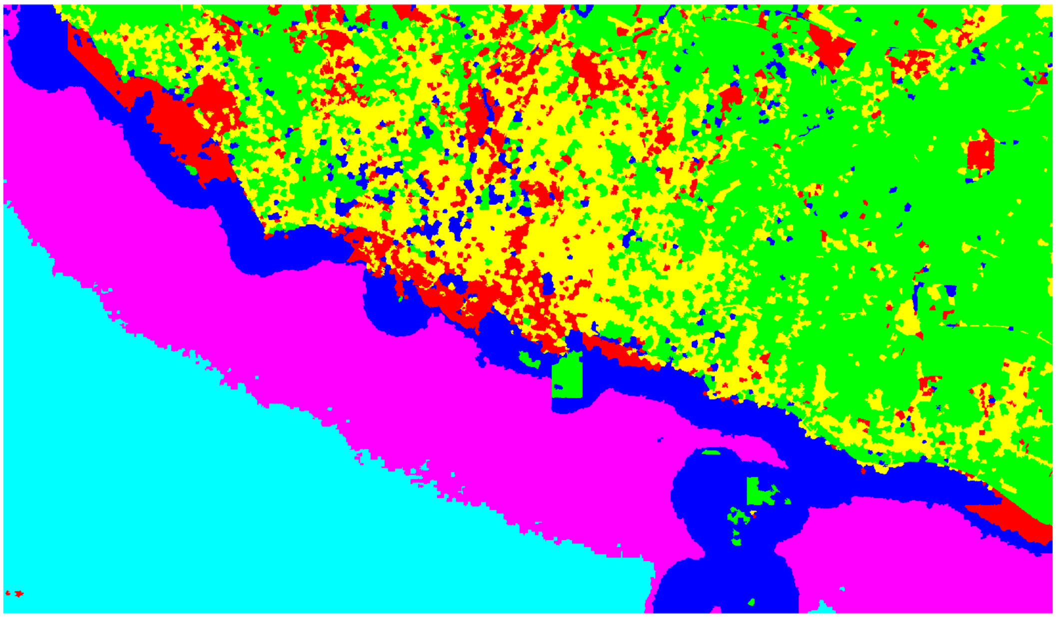
Figure 4.54: Coding to use k-means for clustering the color property of each superpixel

Figure 4.55 shows the coding to assign a color to each cluster.

```
% Assigns color to each cluster
cluster1 = zeros(m,n,3);
cluster2 = zeros(m,n,3);
cluster3 = zeros(m,n,3);
cluster4 = zeros(m,n,3);
cluster5 = zeros(m,n,3);
cluster6 = zeros(m,n,3);
for i=1:m
      for j=1:n
          if Lout(i,j)== 1
                cluster1(i,j,1) = 0;
                cluster1(i,j,2) = 255;
                cluster1(i,j,3) = 0;
          end
          if Lout(i,j)== 2
                cluster2(i,j,1) = 0;
                cluster2(i,j,2) = 0;
                cluster2(i,j,3) = 255;
          end
          if Lout(i,j)== 3
                cluster3(i,j,1) = 0;
                cluster3(i,j,2) = 255;
                cluster3(i,j,3) = 255;
          end
          if Lout(i,j)== 4
                cluster4(i,j,1) = 255;
                cluster4(i,j,2) = 255;
                cluster4(i,j,3) = 0;
          end
          if Lout(i,j)== 5
                cluster5(i,j,1) = 255;
                cluster5(i,j,2) = 0;
                cluster5(i,j,3) = 0;
          end
          if Lout(i,j)== 6
                cluster6(i,j,1) = 255;
                cluster6(i,j,2) = 0;
                cluster6(i,j,3) = 255;
          end
      end
end
allclass = cluster1 + cluster2 + cluster3 + cluster4 + cluster5 + cluster6;
figure, imshow(allclass)
```

Figure 4.55: Coding to assign a color to each cluster

Figure 4.56 shows the output image of K-means Clustering segmentation.



Legends:

Residential area | Sediment

Green spaces | Coastal area

Commercial & Industrial area | Deep Sea

Figure 4.56: Output of K-means Clustering segmentation

4.4 Conclusion

All the software and hardware required for the implementation of project have been listed in this chapter. The software required are MatlabR2021a, Microsoft Word 2019 and Google Earth Pro. Matlab is used to process the satellite image for segmenting the land based on different types of land use, Microsoft Office 2019 is used to preparing proposal and report, and Google Earth Pro is used to download the required satellite image. Besides, the only required hardware is Lenovo Ideapad 320 laptop. All of the tasks in this project are proceed with this laptop.

Moreover, the procedures for the land use segmentation in satellite image was explained in this chapter. Canny Edge Detection, Marker-Controlled Watershed and K-means Clustering are the techniques that have been applied for the segmentation. The steps to conduct all these three techniques have been explained. Besides, the code applied for each technique in Matlab as well as the output image have been captured and shown in this section.

For the next chapter, the testing and analysis will be discussed. This chapter will be discussed about the testing to be carried for identifying the accuracy of each technique in segmenting the satellite based on the different types of land use. In addition, analysis to find out the best technique for land use segmentation will also be discussed.

CHAPTER 5

TESTING AND ANALYSIS

5.1 Introduction

In this chapter, the output images or the results obtained from the previous chapter are tested and analyzed. Accuracy assessment is necessary to be carried out to obtain the accuracy of each technique on classifying the different kinds of classes or land use in satellite image. Confusion matrix is the method to be used for the accuracy assessment. This is the process for us to do the output evaluation in order to identify the best technique to use for the land use classification of satellite image.

5.2 Result and Analysis

Accuracy assessment is used to compare the classified image to another data source which is accurate data (ground truth). Since the Canny Edge Detection can segment the Malacca satellite image into two classes, Marker-Controlled Watershed can segment the image into three classes and K-means clustering can segment the image into six classes from the previous chapter, accuracy assessment is necessary to be conducted to test the accuracy of the land use classification of these three techniques using confusion matrix.

5.2.1 Pre-process of Testing

Before conducting the accuracy assessment, pre-process of testing is necessary for us to collect the ground truth and classified data which are used for testing later. Ground truth is the accurate and true classes (land use) whereas classified data is the predicted classes or land use from the segmented or classified image. In this section, ArcMap and Google Earth are used, ArcMap is one of the primary applications in ArcGIS which is used to create, display, edit and analyze map to obtain information. A set of points has been extracted to compare their predicted or classified data with the ground truth using confusion matrix since each point should contains its own id number, predicted class and true class. The steps to collect the ground truth and classified data:

i. First, convert the segmented or classified satellite image from binary or RGB image to indexed image with associated colormap by running code in Matlab.
ii. Georeferencing the segmented or classified satellite image in ArcMap to cross reference with Google Earth in order to determine the classification of classes (land use) is true or not. In this process, coordinates of Malacca are added into the image.
iii. Open the segmented or classified image in ArcMap and generate a set of accuracy assessment points with certain number by creating a shape file of randomly selected points on the image. Edit the attribute table of the shape file to input the predicted class or land use.
iv. Save the shape file and open it in Google Earth, then find and identify the true class or land use (ground truth) of the set of selected sample points by zooming in to view the map. Then, input the true classes inside the attribute table.
v. After finish collecting the predicted and true classes of all the sample points, copy the needed information from the attribute table into an excel file.

5.2.1.1 Canny Edge Detection

For Canny Edge Detection, the segmented image was converted from binary image to indexed image using coding in Matlab. This is to allow ArcMap can recognize different classes based on different colors of the image. Then, ninety (90) accuracy assessment points are generated to determine the classification of their classes (land use) is true or not. There are only two (2) classes which consist of edge and area are classified for the output image of Canny Edge Detection. Figure 5.1 shows the coding to convert segmented image from binary image to indexed image. Figure 5.2 shows the 90 sample points (classified data) shown in ArcMap whereas Figure 5.3 shows the 90 sample points (ground truth) shown in Google Earth.

```
% Read/import the image
M = imread('cannyEdge.jpg');

% Convert binary image to indexed image with 2 colors
[IND,map] = gray2ind(M, 2);
figure, imshow(IND,map);

% Write or save output image to a TIFF file
imwrite(IND, map, 'canny.tif');
```

Figure 5.1: Coding to convert binary image to indexed image

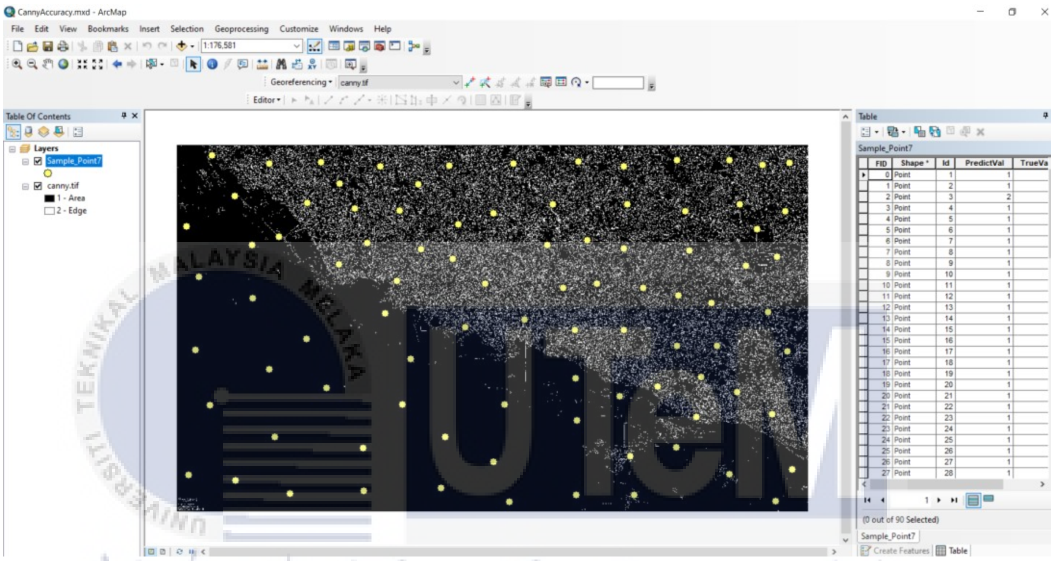


Figure 5.2: 90 sample points shown in ArcMap (Classified Data)

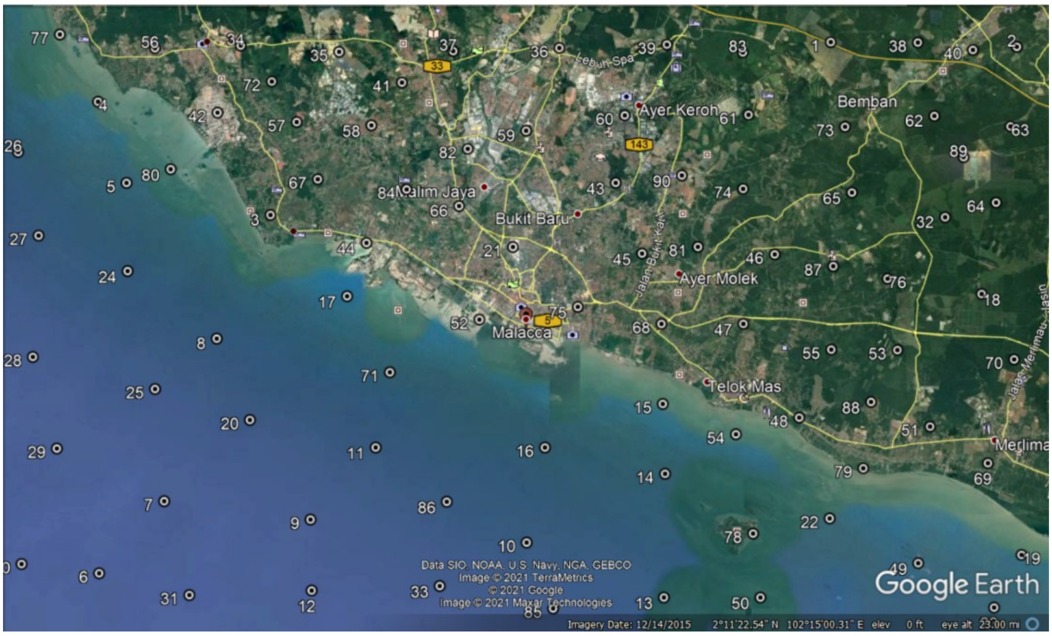


Figure 5.3: 90 sample points shown in Google Earth (Ground Truth)

5.2.1.2 Marker-Controlled Watershed Technique

For Marker-Controlled Watershed, the classified image was converted from RGB image to indexed image using coding in Matlab. This is to allow ArcMap can recognize different classes based on different colors of the image. Then, one hundred (100) accuracy assessment points were generated to determine the classification of their classes (land use) are true or not. Three (3) classes which consist of land and coastal area, sediment and deep sea were classified for the output image of Marker-Controlled Watershed. Figure 5.4 shows the coding to convert classified image from RBG image to indexed image. Figure 5.5 shows the 100 sample points (classified data) shown in ArcMap whereas Figure 5.6 shows the 100 sample points (ground truth) shown in Google Earth.

```
% Read/import the image
M = imread('watershed.jpg');

% Convert RGB image to indexed image with 3 colors
[IND,map] = rgb2ind(M, 3);
figure, imshow(IND,map);

% Write or save output image to a TIFF file
imwrite(IND, map, 'MarkerCW.tif');
```

Figure 5.4: Coding to convert RGB image to indexed image

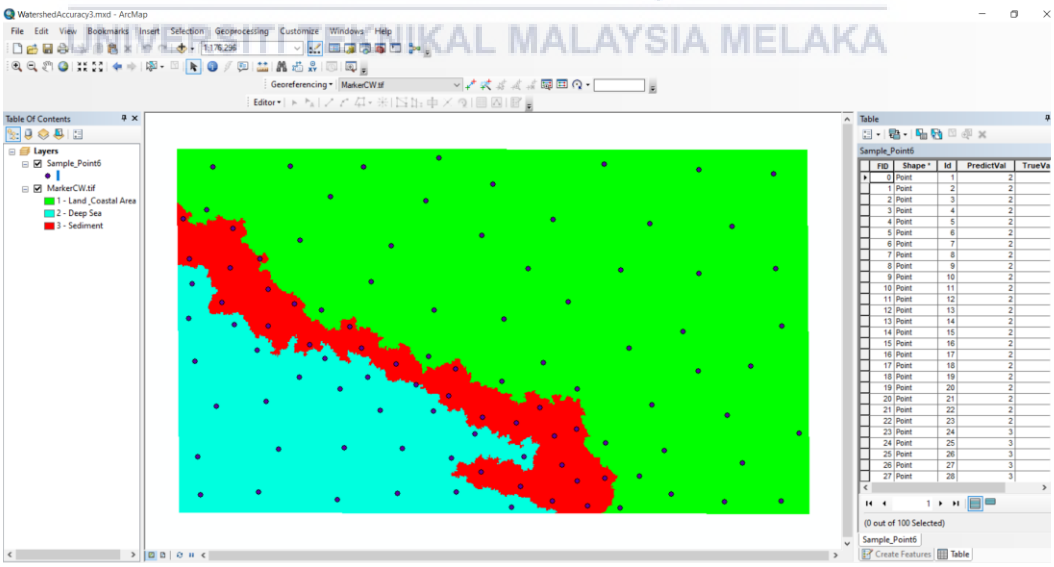


Figure 5.5: 100 sample points shown in ArcMap (Classified Data)

Figure 5.6: 100 sample points shown in Google Earth (Ground Truth)

5.2.1.3 K-means Clustering Technique

For K-means Clustering, the classified image was converted from RGB image to indexed image using coding in Matlab. This is to allow ArcMap can recognize different classes based on different colors of the image. Then, one hundred and twenty (120) accuracy assessment points were generated to determine the classification of their classes (land use) is true or not. Six (6) classes which consist of residential area, green spaces, commercial and industrial area, sediment, coastal area and deep sea were classified for the output image of K-means Clustering. Figure 5.7 shows the coding to convert classified image from RBG image to indexed image. Figure 5.8 shows the 120 sample points (classified data) shown in ArcMap whereas Figure 5.9 shows the 120 sample points (ground truth) shown in Google Earth.

```
% Read/import the image
M = imread('kmeansCluster.jpg');

% Convert RGB image to indexed image with 6 colors
[IND,map] = rgb2ind(M, 6);
figure, imshow(IND,map);

% Write or save output image to a TIFF file
imwrite(IND, map, 'kmeans.tif');
```

Figure 5.7: Coding to convert RGB image to indexed image

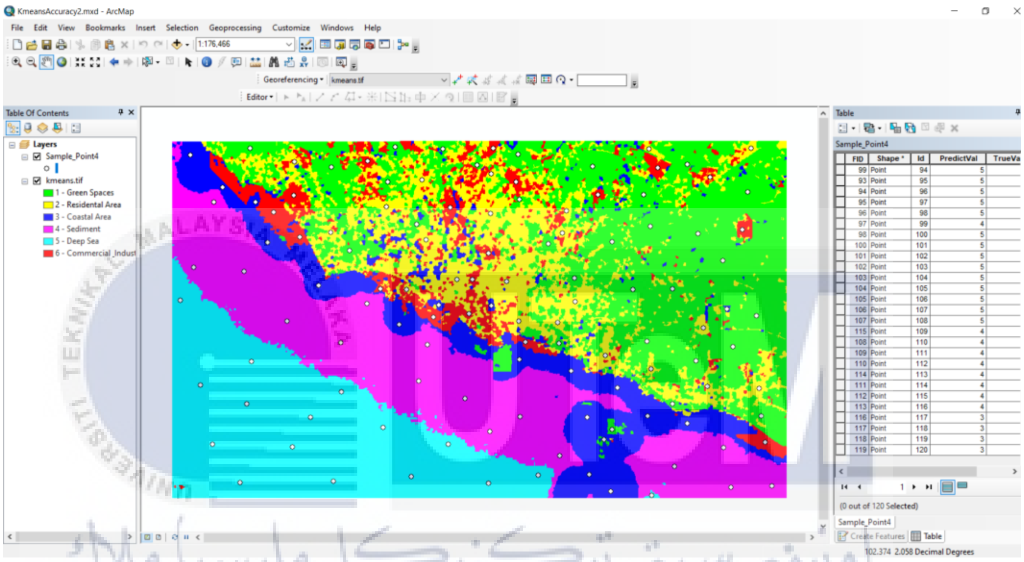


Figure 5.8: 120 sample points shown in ArcMap (Classified Data)

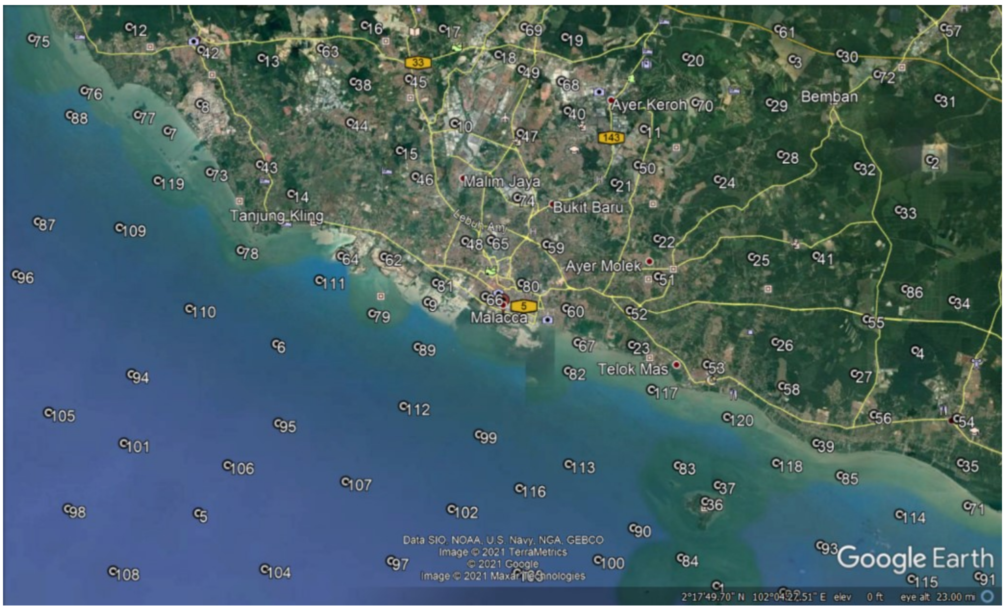


Figure 5.9: 120 sample points shown in Google Earth (Ground Truth)

5.2.2 Testing

Accuracy assessment is the testing conducted on the output image of Canny Edge Detection, Marker-Controlled Watershed and K-means Clustering technique. This is the process to obtain accuracy of each technique on classifying the different kinds of classes or land use in satellite image. This process is considered as accuracy assessment. The data and information containing classified data and ground truth which have been collected from the pre-process of testing are used to undergo the accuracy assessment.

Confusion matrix is used to do the accuracy assessment of the project. It is a table which is used to show the performance of a classification model on a set of sample data for which the actual or true values are known. User's and producer's accuracy, average user's and producer's accuracy, overall accuracy and Kappa Coefficient are the accuracy metrics in the accuracy assessment which are needed to be computed by referring the confusion matrix. User's accuracy is the probability that a value that is predicted to belong to a certain class is really in that class whereas producer's accuracy is the probability that a value in a certain class was classified correctly. The formulas for user's and producer's accuracy are:

$$\text{User's accuracy} = \frac{\textit{Number of correctly classified points in each class}}{\textit{Total number of points that were classified in that class}}$$

$$\text{Producer's accuracy} = \frac{\textit{Number of correctly classified points in each class}}{\textit{Total number of reference points which is known to be of that class}}$$

The Kappa Coefficient is used to measure the agreement or consistency between classification and true values. Therefore, the higher the Kappa Coefficient, the more accurate the classification is. On the other hand, overall accuracy refers to the ratio of the number of correctly classified values to the total number of values. Matlab was used to perform confusion matrix. The formulas for overall accuracy and Kappa Coefficient are:

$$\text{Overall Accuracy} = \frac{\textit{Total Number of Correctly Classified Points}}{\textit{Total Number of Reference Points}} \times 100\%$$

$$\text{Kappa Coefficient} = \frac{N \sum_{i=1}^{n} m_{i,i} - \sum_{i=1}^{n} (G_i \, C_i)}{N^2 - \sum_{i=1}^{n} (G_i \, C_i)}$$

Where:

- **i** = Class number
- **N** = Total number of classified values compared to truth values
- $\mathbf{m_{i,i}}$ = Number of values which is under the truth class **i** and have also been classified as class **i** (values within the diagonal of the confusion matrix)
- $\mathbf{C_i}$ = Total number of predicted values which is under class **i**
- $\mathbf{G_i}$ = Total number of truth values which is under class **i**

First, data was read or imported from the spreadsheet file which contains the predicted and true classes (land use) of all the sample points for each technique. Second, the numbers of rows for data inside the spreadsheet file were found out and the number of classes to be tested for accuracy was defined. In this case, the classification result of Canny Edge Detection, Marker-Controlled Watershed and K-means Clustering have different numbers of classes. Third, the confusion matrix was initialized, obtained and displayed. Forth, the numbers of rows and columns of the confusion matrix were found out, as well as variables to store the sum of the row values and column values were created. Fifth, variables to store the sum of diagonal, the user's and producer's accuracy for each class, as well as the average producer's and user's accuracy were created. Sixth, the total number of the points to be tested and the sum of the points which are rightly classified were computed. Lastly, user's and producer's accuracy for each class, average producer's and user's accuracy, Kappa Coefficient as well as overall accuracy were computed. The spreadsheet file for Canny Edge Detection is CannyData.xlsx, for Marker-Controlled Watershed is WatershedData.xlsx and for K-means clustering is KmeansData.xlsx. Figure 5.10 shows the coding to read or import variables from the spreadsheet file, different file has been imported for different techniques.

```
% Read variables from the spreadsheet file
% Different technique has different spreadsheet file
[Id, TrueVal, PredictVal] = readvars('KmeansData.xlsx');
```

Figure 5.10: Read variables from the spreadsheet file

Figure 5.11 shows the coding to find out numbers of rows for true value and define the numbers of classes for the testing of accuracy (each technique has different number of classes).

```
% Find out number of row for true value
[numRow] = size(TrueVal);
% Define the number of class for the testing of accuracy
% Different technique has different no. of classes
Numofclass = 6;
```

Figure 5.11: Find out numbers of rows for true value and define the numbers of classes for the testing of accuracy

Figure 5.12 shows the coding to initialize, obtain and display the confusion matrix.

```
% Initialize the confusion matrix
CM = zeros(Numofclass,Numofclass);
% Obtain the confusion matrix
for i = 1:numRow
    if(TrueVal(i) == 0)
        continue;
    end
        t = PredictVal(i);          % Obtain the predicted label from the classified image
        k = TrueVal(i);             % Obtain the true label
        CM(k,t) = CM(k,t)+1;        % Confusion matrix assignment
end
% Display the confusion matrix
figure
confusionchart(CM, ...
    'Title', 'Confusion Matrix', ...
    'YLabel', 'True Class', ...
    'XLabel', 'Predicted Class', ...
    'RowSummary', 'row-normalized', ...
    'ColumnSummary','column-normalized');
```


Figure 5.12: Initialize, obtain and display the confusion matrix

Figure 5.13 shows the coding to find out the no. of rows & columns of confusion matrix, and create variable to store the sum of the row values and column values.

```
[m,n] = size(CM);                    % Find out number of row and column of the confusion matrix
sumofRow = zeros(1,m);               % To store the sum of the row values
sumofColumn = zeros(1,n);            % To store the sum of the column values
```

Figure 5.13: Find out the no. of rows & columns of confusion matrix, and create variable to store the sum of the row values and column values

Figure 5.14 shows the coding to create variable to store the sum of diagonal, the user's and producer's accuracy for each class, and the average producer and user accuracy.

```
N = 0;
sumofDiag = 0;                  % To store the sum of diagonal
ProAcc = zeros(m,1);            % To store the producer accuracy for each class
UserAcc = zeros(m,1);           % To store the user accuracy for each class
AveAcc = zeros(2,1);            % To store the average producer and user accuracy
```

Figure 5.14: Create variable to store the sum of diagonal, the user's and producer's accuracy for each class, and the average producer and user accuracy

Figure 5.15 shows the coding to compute the total number of the points to be tested and compute the sum of the points which are rightly classified.

```
for i = 1:m
    for j=1:n
        sumofRow(i) = sumofRow(i) + CM(i,j);
        sumofColumn(j) = sumofColumn(j) + CM(i,j);
        % Compute the total number of the points to be tested
        N=N+CM(i,j);
        if(i==j)
            % Compute the sum of the points which are rightly classified
            sumofDiag = sumofDiag + CM(i,i);
        end
    end
end
```

Figure 5.15: Compute the total number of the points and compute the sum of the points which are rightly classified

Figure 5.16 shows the coding to compute Kappa Coefficient, overall accuracy, user's and producer's accuracy for each class, and average producer's and user's accuracy.

```
PC = 0;
 for i = 1:m
      PC = PC+sumofRow(i)*sumofColumn(i);
 end
% Calculate the Kappa Coefficient
Kappa = (N*sumofDiag-PC)/(N*N-PC);
% Calculate the Overall Accuracy
OveAcc = sumofDiag/N*100;
 for i = 1:m
     for j = 1:m
         % Calculate the user and producer accuracy for each class
         UserAcc(i) = CM(i,i)/sumofRow(i)*100;
         ProAcc(i) = CM(i,i)/sumofColumn(i)*100;
     end
 end
% Calculate average user accuracy
AveAcc(1) = sum(UserAcc)/Numofclass;
% Calculate average producer accuracy
AveAcc(2) = sum(ProAcc)/Numofclass;
```

Figure 5.16: Compute Kappa Coefficient, overall accuracy, user's and producer's accuracy for each class, and average producer's and user's accuracy

5.2.2.1 Accuracy Assessment for Canny Edge Detection Technique

User's and producer's accuracy, average user's and producer's accuracy, overall accuracy and Kappa Coefficient are obtained by performing Confusion Matrix and calculation in Matlab. Figure 5.17 shows the Confusion Matrix for Canny Edge Detection in Matlab.

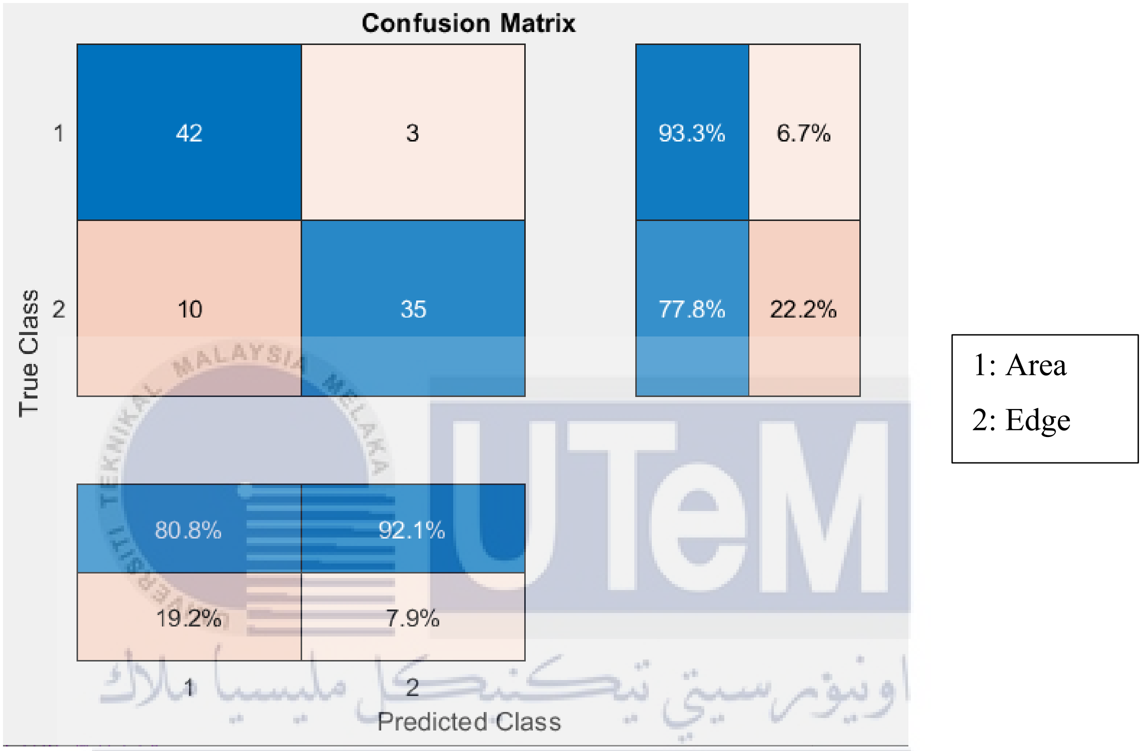


Figure 5.17: Confusion Matrix for Canny Edge Detection in Matlab

Table 5.1 shows the Confusion Matrix for Canny Edge Detection. This table has recorded the true and predicted classes for all the ninety (90) accuracy assessment points generated in the pre-process of testing before this. Based on this table, all of the points are classified into two (2) classes which are Area and Edge. User value is the numbers of points that is predicted in each class using the executive program running in Matlab whereas producer value is the numbers of points which are actually belonged to each class. Among the 45 points which are classified as Area, 42 points are correctly classified and 3 points are wrongly classified. Besides, there are 35 correctly classified points and 10 wrongly classified points among the 45 points which are classified as Edge.

Table 5.1: Confusion Matrix for Canny Edge Detection

Classes	Area	Edge	User value (Total)
Area	42	3	45
Edge	10	35	45
Producer value (Total)	52	38	90

Table 5.2 shows the user's and producer's accuracy for classification result of Canny Edge Detection. For Area class, its user's accuracy is 93.33% and its producer's accuracy is 80.77%. For Edge class, its user's accuracy is 77.78% and its producer's accuracy is 92.11%.

Table 5.2: User's and Producer's accuracy

Accuracy/ Classes	Area	Edge
User's accuracy (%)	93.33	77.78
Producer's accuracy (%)	80.77	92.11

Table 5.3 shows the accuracy metrics computed by referring the confusion matrix. Based on the table, the average user's accuracy is 85.56%, average producer's accuracy is 86.44%, Kappa Coefficient is 0.71 and the Overall Accuracy is 85.56%.

Table 5.3: Accuracy metrics computed by referring the Confusion Matrix

Average User's Accuracy (%)	85.56
Average Producer's Accuracy (%)	86.44
Kappa Coefficient	0.71
Overall Accuracy (%)	85.56

5.2.2.2 Accuracy Assessment for Marker-Controlled Watershed Technique

User's and producer's accuracy, average user's and producer's accuracy, overall accuracy and Kappa Coefficient are obtained by performing Confusion Matrix and calculation in Matlab. Figure 5.18 shows the Confusion Matrix for Marker-Controlled Watershed in Matlab.

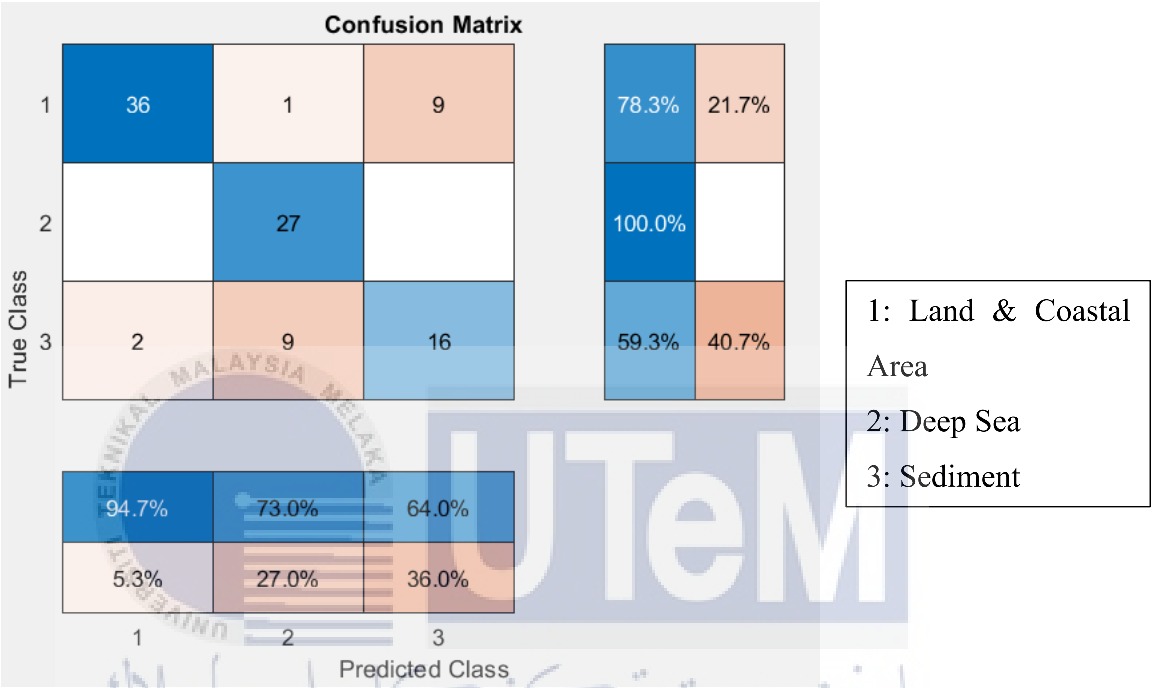


Figure 5.18: Confusion Matrix for Marker-Controlled Watershed in Matlab

Table 5.4 shows the Confusion Matrix for Marker-Controlled Watershed. This table has recorded the true and predicted classes for all the one hundred (100) accuracy assessment points generated in the pre-process of testing before this. Based on this table, all of the points are classified into three (3) classes which are Land & Coastal Area, Deep Sea and Sediment. User value is the numbers of points that is predicted in each class using the executive program running in Matlab whereas producer value is the numbers of points that are actually belonged to each class. Among the 46 points which are classified as Land & Coastal Area, 36 points are correctly classified and 10 points are wrongly classified. Besides, there are 27 points which are correctly classified as Deep Sea. For the 27 points which are classified as Sediment, 16 points are correctly classified and 11 points are wrongly classified.

Table 5.4: Confusion Matrix for Marker-Controlled Watershed

Classes	Land & Coastal Area	Deep Sea	Sediment	User value (Total)
Land & Coastal Area	36	1	9	46
Deep Sea	0	27	0	27
Sediment	2	9	16	27
Producer value (Total)	38	37	25	100

Table 5.5 shows the user's and producer's accuracy for classification result of Marker-Controlled Watershed. For Land & Coastal Area class, its user's accuracy is 78.26% and its producer's accuracy is 94.74%. For Deep Sea class, its user's accuracy is 100% and its producer's accuracy is 72.97%. For Sediment class, its user's accuracy is 59.26% and its producer's accuracy is 64%.

Table 5.5: User's and producer's accuracy

Accuracy/ Classes	Land & Coastal Area	Deep Sea	Sediment
User's accuracy (%)	78.26	100	59.26
Producer's accuracy (%)	94.74	72.97	64.00

Table 5.6 shows the accuracy metrics computed by referring the confusion matrix. Based on the table, the average user's accuracy is 79.17%, average producer's accuracy is 77.24%, Kappa Coefficient is 0.68 and the Overall Accuracy is 79%.

Table 5.6: Accuracy metrics computed by referring the Confusion Matrix.

Average User's Accuracy (%)	79.17
Average Producer's Accuracy (%)	77.24
Kappa Coefficient	0.68
Overall Accuracy (%)	79.00

5.2.2.3 Accuracy Assessment for K-means Clustering Technique

User's and producer's accuracy, average user's and producer's accuracy, overall accuracy and Kappa Coefficient are computed by performing Confusion Matrix in Matlab. Figure 5.19 shows the Confusion Matrix for K-means Clustering in Matlab.

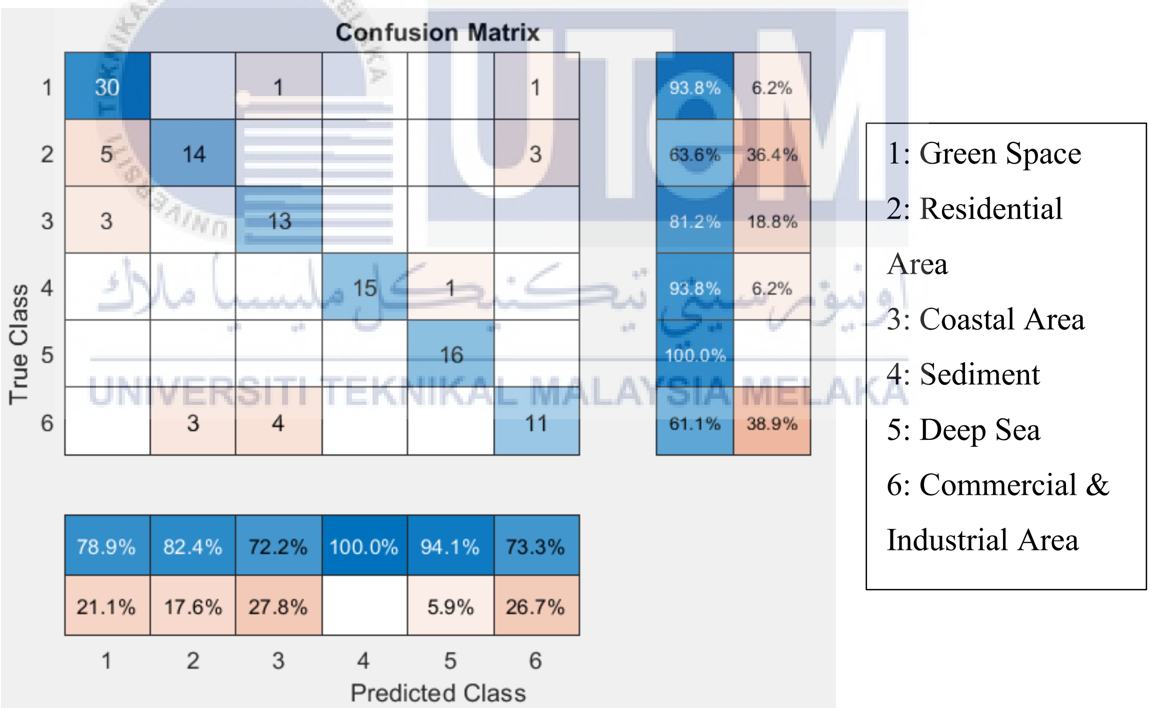


Figure 5.19: Confusion Matrix for K-means Clustering in Matlab

Table 5.7 shows the Confusion Matrix for K-means Clustering. This table has recorded the true and predicted classes for all the one hundred and twenty (120) accuracy assessment points generated in the pre-process of testing before this. Based on this table, all of the points are classified into six (6) classes which are Green Space, Residential Area, Coastal Area, Sediment, Deep Sea, and Commercial & Industrial

Area. User value is the numbers of points that is predicted in each class using the executive program running in Matlab whereas producer value is the numbers of points that are actually belonged to each class. Among the 32 points which are classified as Green Space, 30 points are correctly classified and 2 points are wrongly classified. Besides, there are 14 correctly classified points and 8 wrongly classified points among the 22 points which are classified as Residential Area. For the 16 points which are classified as Coastal Area, 13 points are correctly classified and 3 points are wrongly classified. There are 15 correctly classified points and only 1 wrongly classified point among the 16 points which are classified as Sediment. Moreover, there are 16 points which are correctly classified as Deep Sea. There are 11 correctly classified points and 7 wrongly classified points among the 18 points which are classified as Commercial & Industrial Area.

Table 5.7: Confusion Matrix for K-means Clustering

Classes	Green Space	Residential Area	Coastal Area	Sediment	Deep Sea	Commercial & Industrial Area	User value (Total)
Green Space	30	0	1	0	0	1	32
Residential Area	5	14	0	0	0	3	22
Coastal Area	3	0	13	0	0	0	16
Sediment	0	0	0	15	1	0	16
Deep Sea	0	0	0	0	16	0	16
Commercial & Industrial Area	0	3	4	0	0	11	18
Producer value (Total)	38	17	18	15	17	15	120

Table 5.8 shows the user's and producer's accuracy for classification result of K-means Clustering. For Green Space class, its user's accuracy is 93.75% and its producer's accuracy is 78.95%. For Residential Area class, its user's accuracy is 63.34% and its producer's accuracy is 82.35%. For Coastal Area class, its user's accuracy is 81.25% and its producer's accuracy is 72.22%. For Sediment class, its user's accuracy is 93.75% and its producer's accuracy is 100%. For Deep Sea class, its user's accuracy is 100% and its producer's accuracy is 94.12%. For Commercial & Industrial Area class, its user's accuracy is 61.11% and its producer's accuracy is 73.33%.

Table 5.8: User's and producer's accuracy

Accuracy/ Classes	Green Space	Residential Area	Coastal Area	Sediment	Deep Sea	Commercial & Industrial Area
User's accuracy (%)	93.75	63.64	81.25	93.75	100	61.11
Producer's accuracy (%)	78.95	82.35	72.22	100	94.12	73.33

Table 5.9 shows the accuracy metrics computed by referring the confusion matrix. Based on the table, the average user's accuracy is 82.25%, average producer's accuracy is 83.50%, Kappa Coefficient is 0.79 and the Overall Accuracy is 82.50%.

Table 5.9: Accuracy metrics computed by referring the Confusion Matrix

Average User's Accuracy (%)	82.25
Average Producer's Accuracy (%)	83.50
Kappa Coefficient	0.79
Overall Accuracy (%)	82.50

5.2.3 Techniques Analysis and Evaluation

From the result obtained from the testing part, enough information can be collected to analyze and evaluate the output of the land use segmentation using Canny Edge Detection, Marker-Controlled Watershed as well as K-Means Clustering techniques. From the process of analysis, advantages and disadvantages of these techniques can be concluded. Then, it can evaluate and identify the best technique to segment the satellite image based on different types of land use. Table 5.10 shows the comparison between Canny Edge Detection, Marker-Controlled Watershed as well as K-Means Clustering techniques.

Table 5.10: Comparison between Canny Edge Detection, Marker-Controlled Watershed and K-Means Clustering

	Canny Edge Detection	Marker-Controlled Watershed	K-Means Clustering
Overall Accuracy for Classification (%)	85.56	79.00	82.50
Kappa Coefficient in Classification	0.71	0.68	0.79
Time taken to perform in Matlab	2.91 sec	60.43 sec	118.79 sec
Numbers of classes	2	3	6
Ability to control the numbers of classes (land use)	No	No	Yes

Figure 5.20 shows the graph of overall accuracy for each technique while Figure 5.21 shows the graph of Kappa Coefficient for each technique.

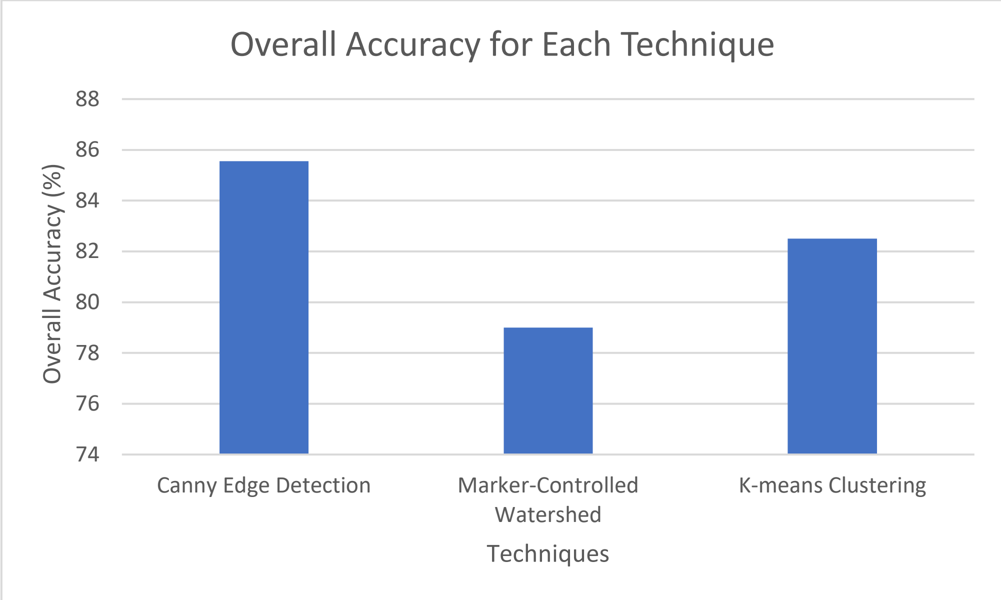


Figure 5.20: Graph of Overall Accuracy for Each Technique

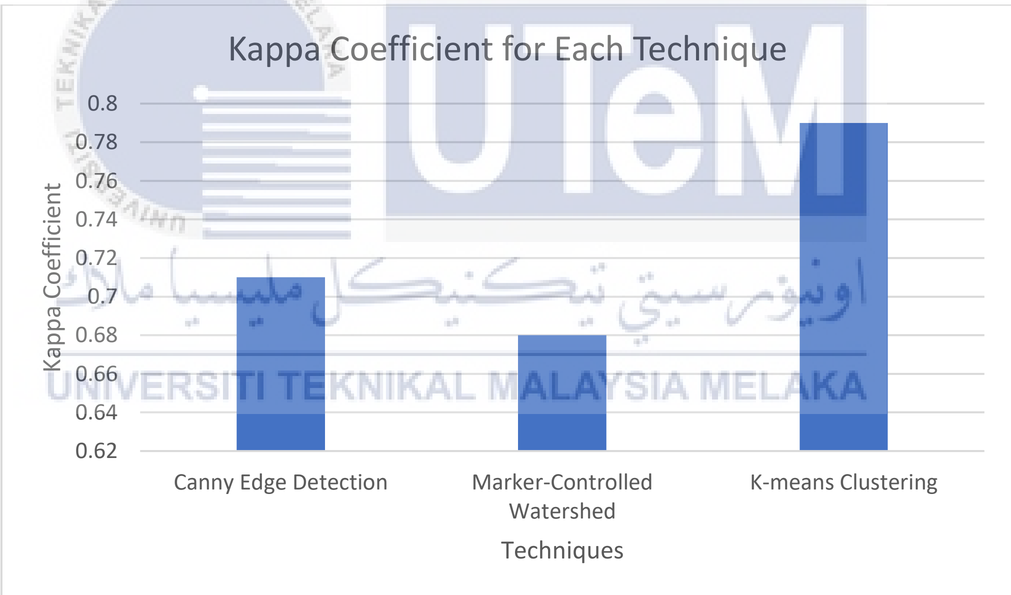


Figure 5.21: Graph of Kappa Coefficient for Each Technique

After making comparison between the three techniques which are Canny Edge Detection, Marker-Controlled Watershed and K-Means Clustering, advantages and disadvantages are identified. Table 5.11 shows the advantages and disadvantages of Canny Edge Detection, Marker-Controlled Watershed and K-Means Clustering techniques.

Table 5.11: Advantages and Disadvantages of Canny Edge Detection, Marker-Controlled Watershed and K-Means Clustering

	Canny Edge Detection	Marker-Controlled Watershed	K-Means Clustering
Advantages	• It has the highest accuracy when detecting the edges in satellite image. • It takes the shortest time for running.	• It can segment and classify different types of land use in satellite image. • It takes shorter time than K-means Clustering for running.	• It can segment and classify different types of land use in satellite image. • It has the Kappa Coefficient that is closest to 1, this means it has highest agreement between classification (predicted value) and truth value. • It has higher accuracy than Marker-Controlled Watershed when classifying the different types of land use in satellite image. • It allows users to control the numbers of classes (land use) that want to classify in satellite image.

			• It is easy to have a good output because its coding is easy to be done in Matlab.
Disadvantages	• It cannot classify between different types of land use because it only able to segment the land by detecting the edges in satellite image. • It does not allow users to control the numbers of classes (land use) that want to classify in satellite image.	• It has the Kappa Coefficient that is furthest to 1, this means it has lowest agreement between classification (predicted value) and truth value. • It has the lowest accuracy when classifying different types of land use in satellite image. • It does not allow users to control the numbers of classes (land use) that want to classify in satellite image. • It is difficult to have a good output because	• It takes the longest time for running.

		it needs frequent adjust on its coding in Matlab.	

Based on Table 5.11, it can conclude that there are only two image segmentation techniques are successful in segmenting and classifying different types of land use in satellite image, they are Marker-Controlled Watershed and K-means Clustering techniques. Canny Edge Detection only can segment the image by identifying the points of edges in image but cannot classify different types of land use. Thus, it can conclude that Canny Edge Detection is not suitable for land use segmentation of satellite image.

Therefore, the next thing to do is to evaluate Marker-Controlled Watershed and K-means Clustering for identifying which is the best technique. Marker-Controlled Watershed has the Kappa Coefficient (0.68) that is furthest to 1, this means it has lowest agreement between classification (predicted value) and truth value. It has the lowest accuracy (79.00%) when classifying different types of land use in satellite image. It does not allow users to control the numbers of classes or land use that want to classify. It is difficult have a good output because it needs frequent adjust on its coding in Matlab.

On the other hand, K-Means Clustering has the Kappa Coefficient (0.79) that is closest to 1, this means it has highest agreement between classification (predicted value) and truth value. It has higher accuracy (82.50%) than Marker-Controlled Watershed (79.00%) when classifying the different types of land use in satellite image. Besides, it allows users to control the numbers of classes or land use that want to classify and it is easy to have a good output because its coding is easy to be done in Matlab.

By comparing the advantages and disadvantages of Marker-Controlled Watershed and K-Means Clustering techniques, it can conclude that K-means Clustering is the best technique for land use segmentation and classification of satellite image.

5.3 Conclusion

In this chapter, all the testing and analysis was successfully completed. In this testing, accuracy assessment is carried out to find out the accuracy of each technique on classifying the different kinds of classes or land use in satellite image. Applications including ArcMap and Google Earth were used for pre-process to obtain data used for accuracy assessment. Then, confusion matrix is the method used in accuracy assessment and it was performed by running code in Matlab. Analysis and evaluation on these three techniques may need to refer their accuracy.

Based on the result of analysis, Canny Edge Detection has the highest accuracy (85.56%) but it cannot classify between different types of land use because it only able to segment the land by detecting the edges in satellite image. On the other hand, K-means has the highest accuracy (82.50%) among the remaining two techniques which can segment and classify different kinds of land use. It is better than Marker-Controlled Watershed because it allows users to define or control the numbers of classes that want to classify in the satellite image. In order to obtain good output, the coding for K-means Clustering is easier to be done in Matlab than Marker-Controlled Watershed.

In conclusion, all the objectives of this project have been achieved which is to identify techniques of land use segmentation, to apply image Edge Detection, Marker-controlled Watershed and K-means Clustering technique to determine the contours of objects within the satellite image, as well as to evaluate the 3 techniques to identify the best technique for land use segmentation of Malacca satellite image. At the previous chapters, Edge Detection, Marker-controlled Watershed and K-means Clustering have been identified and applied as techniques of land use segmentation. Then, these three techniques have been evaluated to identify the best technique. Finally, K-means Clustering is identified as the best technique for land use segmentation of Malacca satellite image.

CHAPTER 6

PROJECT CONCLUSION

6.1 Introduction

In this chapter, there are 5 sub-chapter including introduction, project summarization, project contribution, project limitation, future works and conclusion. All these sub-chapters are discussed.

6.2 Project Summarization

At the beginning of this project, the problem statement and project question have been defined. Based on the problem statement and project question, three project objectives have been defined. These objectives are to identify techniques of land use segmentation, to apply 3 techniques which are image Edge Detection, Marker-controlled Watershed and K-means Clustering to determine the contours of objects within the satellite image and to evaluate the 3 techniques to identify the best technique for land use segmentation. The first objective had been achieved in chapter 2 where the techniques used for land use segmentation has been identified by doing researches. Techniques including Canny Edge Detection, Marker-controlled Watershed and K-means Clustering are selected by referring the introduction and suggestion of the previous researchers from their research papers. Then, the second objective was achieved in Chapter 4 which is project implementation. Canny Edge Detection, Marker-controlled Watershed and K-means Clustering had been applied to segment the Malacca satellite image based on different types of land use. These techniques were implemented by running code in Matlab. The third objective has been achieved in chapter 5 where testing and analysis have been conducted. Accuracy assessment was carried out to obtain important information, the advantages and disadvantages of each technique have been identified for the technique evaluation and analysis. At the end of the project, it can conclude that K-means Clustering is the best technique for land use segmentation.

6.3 Project Contribution

This project can contribute to find out the best technique for land use segmentation among the Canny Edge Detection, Marker-controlled Watershed and K-means Clustering techniques. This may help in some further researches or studies of satellite image processing in the fields of science and technology especially for remote sensing. Furthermore, non-professionals can obtain knowledge in workplaces likes land use mapping, environmental monitoring, as well as land resources planning and management from this project.

6.4 Project Limitation

There are some limitations in this project. There is lack of reference about coding of the executive programs for the three techniques can be found from Internet, this made the project more difficult or tougher to obtain good output. Besides, Matlab is the platform used to implement the three techniques including Canny Edge Detection, Marker-controlled Watershed and K-means Clustering but it does not integrate with GUI. Thus, it is not user-friendly because it needs users to learn and type coding to develop the executive program of these techniques for land use segmentation. Moreover, the coding is needed to be changed or adjusted when users apply these techniques on different satellite image.

6.5 Future Works

This project can be improved in the future by developing a better and light application program of land use segmentation using Canny Edge Detection, Marker-controlled Watershed and K-means Clustering. For instance, make the program to be user-friendly use by integrate it with GUI. Besides, do more researches in order to find out the methods which can improve these three techniques for land use segmentation. Lastly, apply and compare more techniques to identify the best technique for land use segmentation of satellite image.

6.6 Conclusion

Finally, this project has been completed on time and all of the three objectives has been achieved. This project can contribute to find out the best technique for land use segmentation among the Canny Edge Detection, Marker-controlled Watershed and K-means Clustering techniques. Moreover, this project may help the future researches in studying satellite image processing. They may obtain some tips to helps them to undertake and improve their researches from this project even though there are some limitations.

REFERENCES

Asokan, A., Anitha, J., Ciobanu, M., Gabor, A., Naaji, A., & Hemanth, D. J. (2020). Image processing techniques for analysis of satellite images for historical maps classification—an overview. Applied Sciences, 10(12), 4207.

Al-Amri, S. S., Kalyankar, N. V., & Khamitkar, S. D. (2010). Image segmentation by using edge detection. *International journal on computer science and engineering*, *2*(3), 804-807.

Abubakar, F. M. (2012). Study of image segmentation by using edge detection techniques. *International Journal of Engineering Research & Technology (IJERT)*, *1*(9), 1-5.

Anonymous. (n.d). Marker-Controlled Watershed Segmentation. Retrieved May 17, 2021, from https://www.mathworks.com/help/images/marker-controlled-watershed-segmentation.html

Anonymous. (n.d). Plot Land Classification with Color Features and Superpixels. Retrieved May 18, 2021, from https://www.mathworks.com/help/images/land-classification-with-color-features-and-superpixels.html

Anonymous. (2019). AccuracyEvaluationByConfusionMatrix_MATLAB. Retrieved August 5, 2021, from https://github.com/lantingsheng/AccuracyEvaluationByConfusionMatrix_MATLAB/blob/master/AccuracyEvaluationByConfusionMatrix.m

Bhatia, S., & Saxena, K. (2007). Satellite Image Segmentation using Watershed based Algorithms. In *International Conference on Soft computing and Intelligent Systems*.

Bhadoria, P., Agrawal, S., & Pandey, R. (2020, December). Image Segmentation Techniques for Remote Sensing Satellite Images. In *IOP Conference Series: Materials Science and Engineering* (Vol. 993, No. 1, p. 012050). IOP Publishing.

Dhanachandra, N., Manglem, K., & Chanu, Y. (2015). Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Computer Science*, *54*, 764-771.

Esri. (n.d). Accuracy Assessment for Image Classification. Retrieved August 3, 2021, from https://desktop.arcgis.com/en/arcmap/latest/manage-data/raster-and-images/accuracy-assessment-for-image-classification.htm

Haldar, S. K. (2018). Mineral exploration: principles and applications. Elsevier.

Kim, C. (2016). Land use classification and land use change analysis using satellite images in Lombok Island, Indonesia. Forest Science and Technology, 12(4), 183-191.

Kaur, D., & Kaur, Y. (2014). Various image segmentation techniques: a review. *International Journal of Computer Science and Mobile Computing*, *3*(5), 809-814

Kirti, R., & Bhatnagar, A. (2017). Image Segmentation Using Canny Edge Detection Technique. *porc. of international journal of techno-management research*, 8-14.

L3Harris. (n.d). Calculate Confusion Matrices. Retrieved August 3, 2021, from https://www.l3harrisgeospatial.com/docs/calculatingconfusionmatrices.html

Malbog, M. A. F., Lacatan, L. L., Dellosa, R. M., Austria, Y. D., & Cunanan, C. F. (2020, August). Edge Detection Comparison of Hybrid Feature Extraction for Combustible Fire Segmentation: A Canny vs Sobel Performance Analysis. In *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)* (pp. 318-322). IEEE.

Matta, S. (2014). Various image segmentation techniques. *International Journal of Computer Scienceand Information Technologies (IJCSIT)*, *5*(6), 7536-7539.

Muthukrishnan, R., & Radha, M. (2011). Edge detection techniques for image segmentation. *International Journal of Computer Science & Information Technology*, *3*(6), 259.

Mir, S. A., & Padma, T. (2018). Review about Various Satellite Image Segmentation. *Indonesian Journal of Electrical Engineering and Computer Science*, *9*(3), 633-636.

Parmar, C. K., & Pancholi, K. (2015). A Review on Image Compression Techniques. Journal of Information. Knowledge and Research in Electrical Engineering, 2(2), 281-284.

P.Jayapriya, Dr. S.Hemalatha. (2019). Comparative Analysis Of Image Segmentation Techniques And Its Algorithm. *International Journal of Scientific & Technology Research* (IJSTR).

Parvati, K., Rao, P., & Mariya Das, M. (2008). Image segmentation using gray-scale morphology and marker-controlled watershed transformation. *Discrete Dynamics in Nature and Society*, *2008*.

Rekik, A., Zribi, M., Benjelloun, M., & ben Hamida, A. (2006, December). A k-means clustering algorithm initialization for unsupervised statistical satellite image segmentation. In *2006 1st IEEE International Conference on E-Learning in Industrial Electronics* (pp. 11-16). IEEE.

Rizvi, I. A., Mohan, B. K., & Bhatia, P. R. (2011, February). Multi-resolution segmentation of high-resolution remotely sensed imagery using marker-controlled watershed transform. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology* (pp. 674-678).

Sonawane, M. S., & Dhawale, C. A. (2015). A brief survey on image segmentation methods. *International Journal of Computer Applications*, *975*, 8887.

Saha, I., Maulik, U., Bandyopadhyay, S., & Plewczynski, D. (2011). SVMeFC: SVM ensemble fuzzy clustering for satellite image segmentation. *IEEE Geoscience and remote sensing letters*, *9*(1), 52-55.

Sowmya, D. R., Kulkarni, A. N., Sandeep, S., Shenoy, P. D., & Venugopal, K. R. (2018). Land Use/Land Cover Segmentation of Satellite Imagery to Estimate the Utilization of Earth's Surface. In Proceedings of the Second International Conference on Computational Intelligence and Informatics (pp. 27-36). Springer, Singapore.

Usman, B. (2013). Satellite imagery land cover classification using k-means clustering algorithm computer vision for environmental information extraction. *Elixir International Journal of Computer Science and Engineering*, *63*, 18671-18675.

Verdonck, L., De Smedt, P., & Verhegge, J. (2019). Making sense of anomalies: Practices and challenges in the archaeological interpretation of geophysical data. In *Innovation in Near-Surface Geophysics* (pp. 151-194). Elsevier.

APPENDICES

Appendix A - Coding for implementation in Matlab

Coding for Canny Edge Detection technique

```
% Read/import the image
m = imread ("malacca.jpg");

% Convert image to Grayscale
M = rgb2gray(m);

% Apply Canny Edge Detection on the image and display output
canny = edge(M,'canny');
figure, imshow(canny), title('Image Processed with Canny Edge Detection')

% To find out the size of the output image
[rows, columns] = size(canny);

% Crops image into 16 equal parts and enlarge each cropped part (4 rows & 4
columns)
% Crops image into 4 columns for 1st rows
canny1 = imcrop(canny,[0 0 columns/4 rows/4]);
canny2 = imcrop(canny,[columns/4 0 columns/4 rows/4]);
canny3 = imcrop(canny,[columns/4*2 0 columns/4 rows/4]);
canny4 = imcrop(canny,[columns/4*3 0 columns/4 rows/4]);

% Crops image into 4 columns for 2nd rows
canny5 = imcrop(canny,[0 rows/4 columns/4 rows/4]);
canny6 = imcrop(canny,[columns/4 rows/4 columns/4 rows/4]);
canny7 = imcrop(canny,[columns/4*2 rows/4 columns/4 rows/4]);
canny8 = imcrop(canny,[columns/4*3 rows/4 columns/4 rows/4]);

% Crops image into 4 columns for 3rd rows
canny9 = imcrop(canny,[0 rows/4*2 columns/4 rows/4]);
canny10 = imcrop(canny,[columns/4 rows/4*2 columns/4 rows/4]);
canny11 = imcrop(canny,[columns/4*2 rows/4*2 columns/4 rows/4]);
canny12 = imcrop(canny,[columns/4*3 rows/4*2 columns/4 rows/4]);

% Crops image into 4 columns for 4th rows
canny13 = imcrop(canny,[0 rows/4*3 columns/4 rows/4]);
canny14 = imcrop(canny,[columns/4 rows/4*3 columns/4 rows/4]);
canny15 = imcrop(canny,[columns/4*2 rows/4*3 columns/4 rows/4]);
canny16 = imcrop(canny,[columns/4*3 rows/4*3 columns/4 rows/4]);

% Display each cropped part
figure, imshow(canny1), title('Crop 1')
```

```
figure, imshow(canny2), title('Crop 2')
figure, imshow(canny3), title('Crop 3')
figure, imshow(canny4), title('Crop 4')
figure, imshow(canny5), title('Crop 5')
figure, imshow(canny6), title('Crop 6')
figure, imshow(canny7), title('Crop 7')
figure, imshow(canny8), title('Crop 8')
figure, imshow(canny9), title('Crop 9')
figure, imshow(canny10), title('Crop 10')
figure, imshow(canny11), title('Crop 11')
figure, imshow(canny12), title('Crop 12')
figure, imshow(canny13), title('Crop 13')
figure, imshow(canny14), title('Crop 14')
figure, imshow(canny15), title('Crop 15')
figure, imshow(canny16), title('Crop 16')
```

Coding for Marker-Controlled Watershed technique

```
% Read/import the image
m = imread("malacca.jpg");

% Convert image to Grayscale
M = rgb2gray(m);
% Enhances the contrast of each tile
M = adapthisteq(M);

% Use gradient magnitude as segmentation fucnction to detect edges
gradmag = imgradient(M);
imshow(gradmag,[]), title('Gradient Magnitude')

% Compare reconstruction-based, standard opening and closing
% Use the best opening and closing to mark the foreground objects
% Compute the standard opening
se = strel('disk',95);
Mo = imopen(M, se);
figure, imshow(Mo), title('Opening')
% Compute the opening-by-reconstruction
Me = imerode(M, se);
Mobr = imreconstruct(Me, M);
figure, imshow(Mobr), title('Opening-by-reconstruction')
% Close based on opening
Moc = imclose(Mo, se);
figure, imshow(Moc), title('Opening-closing')
% Close based on open imresconstruct
Mobrd = imdilate(Mobr, se);
Mobrcbr = imreconstruct(imcomplement(Mobrd), imcomplement(Mobr));
Mobrcbr = imcomplement(Mobrcbr);
figure, imshow(Mobrcbr), title('Opening-closing by reconstruction')
```

```
% Calculate the regional maxima to obtain good foreground markers
fgm = imregionalmax(Mobrcbr);
figure, imshow(fgm), title('Regional maxima of opening-closing by
reconstruction')
% Superimpose the foreground marker image on the original image
M2 = labeloverlay(M,fgm);
figure, imshow(M2), title('Regional maxima superimposed on original image')
% Clean the edges of the marker blobs and then shrink them a bit using erosion
se2 = strel(ones(1,1));
fgm2 = imclose(fgm, se2);
fgm3 = imerode(fgm2, se2);
% Removes blobs having fewer than a certain number of pixels
fgm4 = bwareaopen(fgm3, 20);
M3 = labeloverlay(M,fgm4);
figure, imshow(M3), title('Modified regional maxima superimposed on original
image')


% Compute background markers
% Thresholding operation
bw = imbinarize(Mobrcbr);
figure, imshow(bw), title('Thresholded opening-closing by reconstruction')
% Compute the watershed transform of the distance transform of bw
D = bwdist(bw,'euclidean');
DL = watershed(D);
% Looking for the watershed ridge lines of the result
% Produce label matrix
bgm = DL == 0;
figure, imshow(bgm), title('Watershed ridge lines')

% Modify the gradient magnitude image
gradmag2 = imimposemin(gradmag, bgm | fgm4);
% Compute the Watershed Transform segmentation
L = watershed(gradmag2);

% Visualize the result
% Superimpose the markers and segmented object boundaries
labels = imdilate(L==0,ones(3,3)) + 2*bgm + 3*fgm4;
M4 = labeloverlay(M,labels);
figure, imshow(M4), title('Markers and object boundaries superimposed on
original image')
% Assigns colors to regions in the label matrix
[m, n] = size (L);
area1 = zeros(m,n,3);
area2 = zeros(m,n,3);
area3 = zeros(m,n,3);
for i=1:m
    for j=1:n
       if L(i,j)== 1
```

```
            area1(i,j,1) = 255;
            area1(i,j,2) = 0;
            area1(i,j,3) = 0;
        end
        if L(i,j)== 2
            area2(i,j,1) = 0;
            area2(i,j,2) = 255;
            area2(i,j,3) = 255;
        end
        if L(i,j)== 3
            area3(i,j,1) = 0;
            area3(i,j,2) = 255;
            area3(i,j,3) = 0;
        end
    end
end
allclass = area1 + area2 + area3;
figure, imshow(allclass)
title('Colored watershed label matrix')
% Use transparency to superimpose label matrix on top of the original image
figure, imshow(M), hold on
himage = imshow(allclass);
set(himage, 'AlphaData', 0.4);
title('Colored Labels superimposed transparently on original image')
```

Coding for K-means Clustering technique

```
% Read/import the image
M = imread('malacca.jpg');

% Convert the image to the L*a*b* color
Mlab = rgb2lab(M);

% Compute the superpixel oversegmentation of the image
[L,N] = superpixels(Mlab,20000,'isInputLab',true);
BW = boundarymask(L);

% Display it
figure,imshow(imoverlay(M,BW,'green')), title('Superpixel oversegmentation')

% Create a cell array for the set of pixels
pxIdxList = label2idx(L);

% Determine the median color of each region in the L*a*b* color
[m,n] = size(L);
meanColor = zeros(m,n,3,'single');
for i = 1:N
    meanColor(pxIdxList{i}) = mean(Mlab(pxIdxList{i}));
```

```
    meanColor(pxIdxList{i}+m*n) = mean(Mlab(pxIdxList{i}+m*n));
    meanColor(pxIdxList{i}+2*m*n) = mean(Mlab(pxIdxList{i}+2*m*n));
end

% Use k-means function to cluster the color property of each superpixel
numCluster = 6;
Lout = imsegkmeans(meanColor,numCluster,'numAttempts',2);


% Assigns color to each cluster
cluster1 = zeros(m,n,3);
cluster2 = zeros(m,n,3);
cluster3 = zeros(m,n,3);
cluster4 = zeros(m,n,3);
cluster5 = zeros(m,n,3);
cluster6 = zeros(m,n,3);
for i=1:m
    for j=1:n
      if Lout(i,j)== 1
          cluster1(i,j,1) = 0;
          cluster1(i,j,2) = 255;
          cluster1(i,j,3) = 0;
      end
      if Lout(i,j)== 2
          cluster2(i,j,1) = 0;
          cluster2(i,j,2) = 0;
          cluster2(i,j,3) = 255;
      end
      if Lout(i,j)== 3
          cluster3(i,j,1) = 0;
          cluster3(i,j,2) = 255;
          cluster3(i,j,3) = 255;
      end
      if Lout(i,j)== 4
          cluster4(i,j,1) = 255;
          cluster4(i,j,2) = 255;
          cluster4(i,j,3) = 0;
      end
      if Lout(i,j)== 5
          cluster5(i,j,1) = 255;
          cluster5(i,j,2) = 0;
          cluster5(i,j,3) = 0;
      end
      if Lout(i,j)== 6
          cluster6(i,j,1) = 255;
          cluster6(i,j,2) = 0;
          cluster6(i,j,3) = 255;
      end
    end
end
```

```
allclass = cluster1 + cluster2 + cluster3 + cluster4 + cluster5 + cluster6;
figure, imshow(allclass)
```

Appendix B – Coding for pre-process of testing in Matlab

Coding for converting binary image of Canny Edge Detection to indexed image in pre-process of testing

```
% Read/import the image
M = imread('cannyEdge.jpg');

% Convert binary image to indexed image with 2 colors
[IND,map] = gray2ind(M, 2);
figure, imshow(IND,map);

% Write or save output image to a TIFF file
imwrite(IND, map, 'canny.tif');
```

Coding for converting RGB image of Marker-Controlled Watershed to indexed image in pre-process of testing

```
% Read/import the image
M = imread('watershed.jpg');

% Convert RGB image to indexed image with 3 colors
[IND,map] = rgb2ind(M, 3);
figure, imshow(IND,map);

% Write or save output image to a TIFF file
imwrite(IND, map, 'MarkerCW.tif');
```


Coding for converting RGB image of K-means Clustering to indexed image in pre-process of testing

```
% Read/import the image
M = imread('kmeansCluster.jpg');

% Convert RGB image to indexed image with 6 colors
[IND,map] = rgb2ind(M, 6);
figure, imshow(IND,map);

% Write or save output image to a TIFF file
imwrite(IND, map, 'kmeans.tif');
```

Appendix C - Coding for testing in Matlab

Coding for Accuracy Assessment of each technique

```
% Read variables from the spreadsheet file
% Different technique has different spreadsheet file
[Id, TrueVal, PredictVal] = readvars('KmeansData.xlsx');

% Find out number of row for true value
[numRow] = size(TrueVal);
% Define the number of class for the testing of accuracy
% Different technique has different no. of classes
Numofclass = 6;

% Initialize the confusion matrix
CM = zeros(Numofclass,Numofclass);
% Obtain the confusion matrix
for i=1:numRow
   if(TrueVal(i)==0)
      continue;
   end
      t=PredictVal(i);          % Obtain the predicted label from the classified image
      k=TrueVal(i);             % Obtain the true label
      CM(k,t)=CM(k,t)+1;  % Confusion matrix assignment
end
% Display the confusion matrix
figure
confusionchart(CM, ...
   'Title', 'Confusion Matrix', ...
   'YLabel', 'True Class', ...
   'XLabel', 'Predicted Class', ...
   'RowSummary', 'row-normalized', ...
   'ColumnSummary','column-normalized');

[m,n] = size(CM);                    % Find out number of row and column of the
confusion matrix
sumofRow = zeros(1,m);          % To store the sum of the row values
sumofColumn = zeros(1,n);       % To store the sum of the column values

N = 0;
sumofDiag = 0;                  % To store the sum of diagonal
ProAcc = zeros(m,1);            % To store the producer accuracy for each class
UserAcc = zeros(m,1);           % To store the user accuracy for each class
AveAcc = zeros(2,1);            % To store the average producer and user accuracy
 for i = 1:m
     for j=1:n
         sumofRow(i) = sumofRow(i) + CM(i,j);
         sumofColumn(j) = sumofColumn(j) + CM(i,j);
         % Compute the total number of the points to be tested
         N=N+CM(i,j);
```

```
        if(i==j)
            % Compute the sum of the points which are rightly classified
            sumofDiag = sumofDiag + CM(i,i);
        end
    end
end


PC = 0;
for i = 1:m
    PC = PC+sumofRow(i)*sumofColumn(i);
end
% Calculate the Kappa Coefficient
Kappa = (N*sumofDiag-PC)/(N*N-PC);
% Calculate the Overall Accuracy
OveAcc = sumofDiag/N*100;
for i = 1:m
    for j = 1:m
        % Calculate the user and producer accuracy for each class
        UserAcc(i) = CM(i,i)/sumofRow(i)*100;
        ProAcc(i) = CM(i,i)/sumofColumn(i)*100;
    end
end
% Calculate average user accuracy
AveAcc(1) = sum(UserAcc)/Numofclass;
% Calculate average producer accuracy
AveAcc(2) = sum(ProAcc)/Numofclass;
```