SOIL CLASSIFICATION FOR AGRICULTURE CROPS USING K-NEAREST NEIGHBORS (KNN)

SHWE YEE WIN

M.C.Sc. **JUNE 2022**

SOIL CLASSIFICATION FOR AGRICULTURE CROPS USING K-NEAREST NEIGHBORS (KNN)

By

SHWE YEE WIN

B.C.Sc(Hons:)

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Yangon JUNE 2022

ACKNOWLEDGEMENTS

To accomplish this thesis, many things are essential like my best effort, hard work, and important ideas from specific supporters while completing this thesis.

Firstly, I would like to express my special thanks to **Dr. Mie Mie Khin**, Rector of the University of Computer Studies, Yangon, for her kind permission and encouragement to submit this dissertation.

I would like to express my profound gratitude to my thesis supervisor, **Dr. Thin Lai Lai Thein**, Professor, Faculty of Information Science, University of Computer Studies, Yangon, for her close supervision, cooperation during my effort of study, understanding and generous encouragements during a long period of study during the progress of this work.

I would like to thank course coordinators, **Dr. Si Si Mar Win and Dr. Tin Zar Thaw**, Professors, Faculty of Computer Science, University of Computer Studies, Yangon, Dr. Moh Moh Khaing, Processor and Head of Electronics Engineering Department, Lashio and Dr. Myo Zaw Min, Pro-Rector of the Technological University, Lashio for their superior suggestions and administrative supports during my academic study.

I wish to express my deepest gratitude to **Daw Mya Hnin Mon**, Associate Professor, English Department, University of Computer studies, Yangon, for her advices for editing this thesis, and suggestions from the language point of view.

I am also so grateful to **all my teachers** who taught me computer science throughout the master’s degree course, and **my friends** for their corporation, and am extremely grateful to **my beloved parents** for their continuous support, patience, understanding, consideration and love that enable me to complete this thesis.

Last but not least, many thanks are extended to all persons who directly and indirectly contributed towards the success of this research.

ABSTRACT

Soil is the basis of our earth’s agroecosystems which provide us with fiber, food, fuel etc. Soil classification helps predict soil type and performance for growing agricultural crops that provide us with food. Soil classification is essential for a farmer who can know soil type, and plants the suitable crops depending soil type. The aim of this research is therefore to develop a method that automates soil classification by applying image processing techniques. In the proposed soil classification method, soil classification is performed by using color and texture of a soil image as features and by using the K-Nearest Neighbors (KNN) as a classifier.The proposed soil classification method firstly extracts color features: mean and standard deviation, and texture features: energy and contrast, from soil images in dataset. These features vectors are then saved as a features dataset. In testing phase, the texture and color features from the user input soil image are also extracted as a testing feature vector. The user input soil image is then classified based on this testing feature vector by comparing with all the features vectors in the features dataset using k-nearest neighbors (KNN) classifier. After classifying the user input soil image whether it is clay or clay loam or sandy loam, the system provides the list of crops and vegetations which can easily be grown in the predicted soil type.Soil RGB images dataset applied to our soil classification system contains “sandy loam” and “clay loam” (Red Earths and Yellow Earths) soil images has taken in plantations and farms in Lashio township and collected from Internet. Our own soil image dataset including 200 soil images is applied to the system with the purpose of building the features dataset and testing the system. 150 soil images in the dataset are used for building the features dataset and 50 soil images are, for testing the system, as unknown data. The overall accuracy of the system is over 88% for all 3 soil types: clay, clay loam and sandy loam. The system is implemented in MATLAB programming environment on Microsoft Windows platform.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

Page

CHAPTER 1

INTRODUCTION

Soil is a crucial ingredient of agricultural products. There are different kinds of soil exist. Each soil type can have various kinds of features and crops that can be cultivated on different soil types. The features and characteristics of different types of soils are needed to know so as to understand which crops can be grown healthier in a specific soil type [14].

Moreover, most of the countries conduct agricultural products exporting, in which those countries exporting agricultural products of higher standard are very much depending on these soil characteristics. Soil characteristics classification and identification are therefore very much important. Soil type identification helps to overcome quantity loss of agricultural products [16].

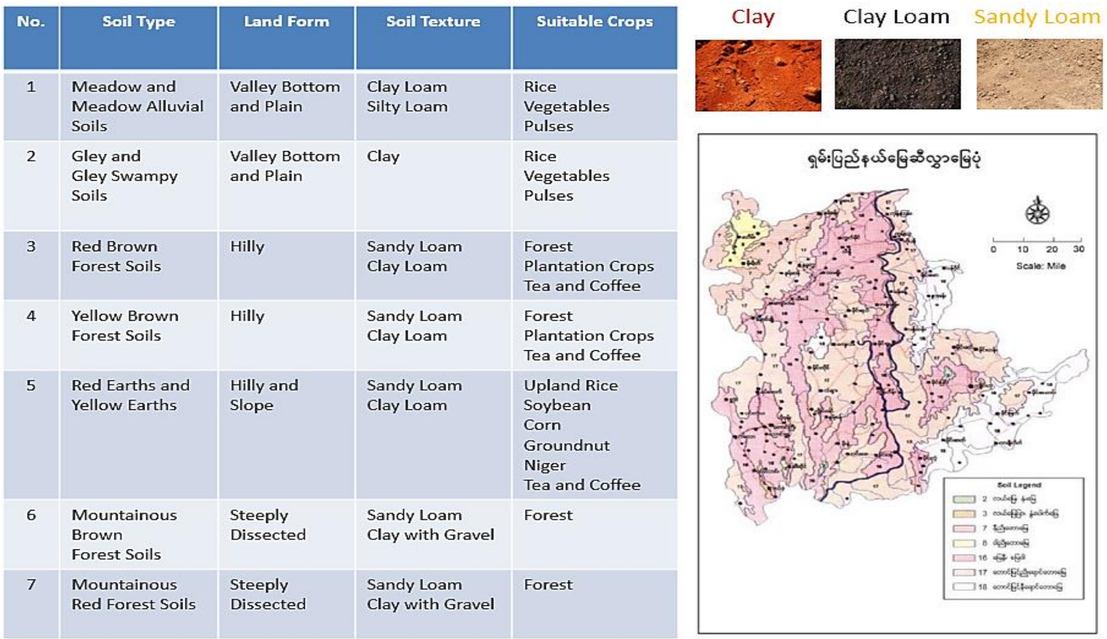
1.1 Soil Classification

Soil classification comprises steps like image acquisition, image pre-processing, feature extraction and classification [16]. Statistical features such as HSV histogram, mean and standard deviation, low-pass filter, Gabor filter and color quantization [16], pH, Potassium and Zinc, chemical features [14], color features, texture features, drainage class features and terrain features [15], can be applied as features to soil classification.

Machine learning (ML) algorithms can be useful to soil classification because it is progressed significantly in recent years. ML is still a challenging and emerging research field in agricultural data analysis. Several ML methods such as Gaussian Kernel-based Support Vector Machines (SVM), Bagged Trees, weighted K-Nearest Neighbor (k-NN), [14], K-means clustering and Self-organizing Maps (SOM) [15] are applied to soil classification.

Land Use Division of Myanmar Agriculture Service had prepared a new soil map of Myanmar based on the Rosanov soil system by definitely modifying the taxonomy, nomenclature, definition of types and properties and by adding additional data to correlate it with the FAO/UNESCO classification in 1970. According to the modern soil classification, 24 main types of soils are being recognized in Myanmar. The these soils' characteristics are determined upon (1) the mineral and physical composition of the parent material, (2) physical features, the relief, (3) the vegetation, (4) the climate under which the soil material has been developed [7].

Soil RGB images dataset applied to our soil classification system contains "sandy loam" and "clay loam" (Red Earths and Yellow Earths) soil images taken in plantations and farms in Lashio townships located in Shan state. The Red Earths is the typical soils in Shan state for agriculture. They are having a good structure, well drained, easy to plough, and very suitable for cultivation of perennial as well as seasonal crops. The Yellow Earths can be occurred on the Shan plateau's lower slopes. The Yellow Earths soil are suitable for forests, flowers and gardens [7]. In Figure 1.1, soil types, land forms, soil textures, and list of suitable corps for Shan plateau is depicted.



No.	Soil Type	Land Form	Soil Texture	Suitable Crops
1	Meadow and Meadow Alluvial Soils	Valley Bottom and Plain	Clay Loam Silty Loam	Rice Vegetables Pulses
2	Gley and Gley Swampy Soils	Valley Bottom and Plain	Clay	Rice Vegetables Pulses
3	Red Brown Forest Soils	Hilly	Sandy Loam Clay Loam	Forest Plantation Crops Tea and Coffee
4	Yellow Brown Forest Soils	Hilly	Sandy Loam Clay Loam	Forest Plantation Crops Tea and Coffee
5	Red Earths and Yellow Earths	Hilly and Slope	Sandy Loam Clay Loam	Upland Rice Soybean Corn Groundnut Niger Tea and Coffee
6	Mountainous Brown Forest Soils	Steeply Dissected	Sandy Loam Clay with Gravel	Forest
7	Mountainous Red Forest Soils	Steeply Dissected	Sandy Loam Clay with Gravel	Forest

Figure 1.1 Soil Types, Land Forms, Soil Textures, and List of Suitable Crops for Shan Plateau

1.2 Motivation of the Thesis

Soil is an important component in agriculture for cultivating crops. Soil type of a particular geographical area is generally analyzed by manually collecting soils samples, and classifying them into certain soil types using different methods. These tasks are experts and labor-intensive, time-consuming, and expensive also. The main point is to automate these procedures, and to achieve faster and more accurate information of soil.

With the emerging technologies concerning ML and image, it can be effectively classified a soil image into soil types which it belongs to. Therefore, an automated soil classification system is needed to developed which can automatically determine the soil type for particular crops or plants based on a soil image using ML and image processing techniques.

1.3 Objectives of the Thesis

The objectives of the thesis are as follow:

(i) To study the image processing techniques and soil classification systems.

(ii) To study the integration of image features such as mean, standard deviation, contrast and energy with the K-NN classifier.

(iii) To classify soil types of Myanmar in specific regions.

(iv) To implement an efficient soil classification system and analyze its performance with experimental results.

(v) To provide the list of crops which can be grown in the output soil type.

1.4 Contributions of the Thesis

A soil classification system and its methods for agricultural crops using KNN is developed in this thesis. The proposed soil classification system is essential to automatic soil classification using image processing techniques to be used in farms and plantation in the specific areas in Myanmar. The contributions of the thesis are as follows:

(i) Image Features extraction using color and texture is proposed.

(ii) Soil classification system using the extracted features and KNN is developed.

(iii) List of suitable crops in the predicted soil type is also provided for farms and plantations.

(iv) The accuracy of the proposed system is compared with the other two classifiers: Decision (Fine) Trees and Kernel Naïve Bayes.

1.5 Overview of the System

The proposed soil classification system comprises two processes: features extraction from soil RGB images and building dataset process, and testing or soil classification process.

In the first process, image color features: mean and standard deviation and image texture features: energy and contrast from soil RGB images are extracted as features vectors. The features dataset is then built by using these features vectors.

In the second process, a tested soil image is inserted into the proposed system, and the features extraction from it is performed. The tested soil image is classified by finding distances between its features vector and features vectors in the features dataset by using KNN. The system then outputs the soil type of the tested soil image and the list of crops that can easily grow on the soil in the tested soil image. The overview of the system is shown in Figure 1.2.

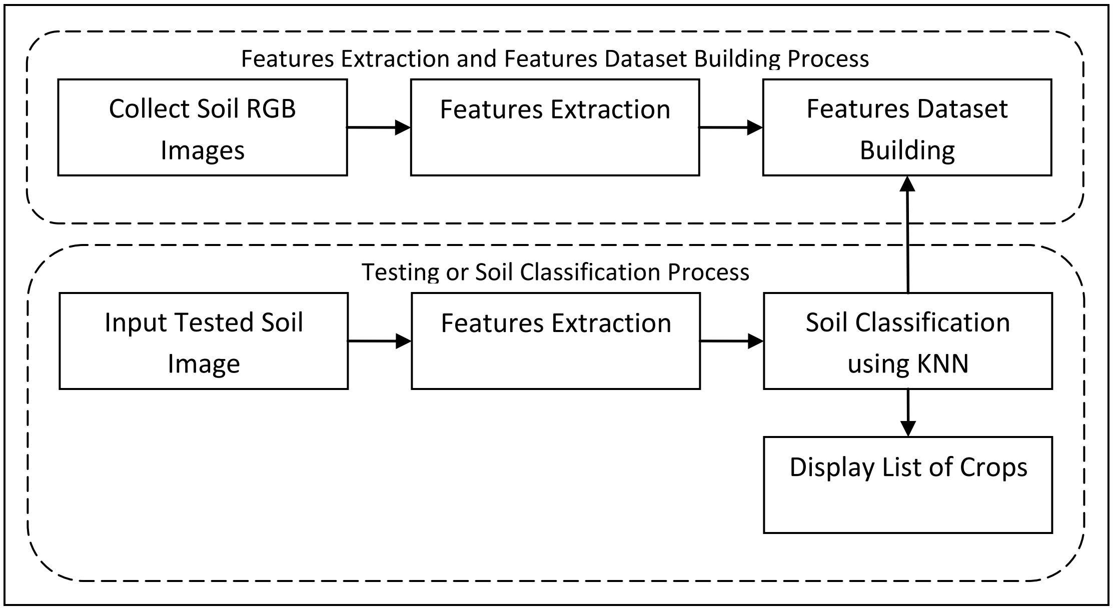

Features Extraction and Features Dataset Building Process
Collect Soil RGB Images
Features Extraction
Features Dataset Building
Testing or Soil Classification Process
Input Tested Soil Image
Features Extraction
Soil Classification using KNN
Display List of Crops

Fig. 1.2 The Overview of the System

1.6 Organization of the Thesis

This thesis is comprised of five chapters, abstract, acknowledgment and references.

Soil classification, machine learning and soil types of Myanmar are introduced in chapter one. This chapter also presents the motivation, objectives of the research work, contribution, and overview.

The basic concepts of digital image processing, types of images, the image features for the soil classification system, types of classifiers and related works are presented in chapter two. Fundamental steps in image processing are also briefly explained in this chapter.

The soil classification system is explained in detailed in chapter three. First of all, dataset collection is described, and the detail discussion about color and texture features extraction from the dataset is then done, and soil classification using KNN is finally presented.

Design and implementation of soil classification system using KNN are described in chapter four. First of all, the overview of system design and the architecture of the system are described in order to understand the process flow of the proposed soil classification system in this chapter. The implementation of all the processes in the proposed system using MATLAB programming environment is then discussed with User Interfaces (UI). The performance analysis and the experimental results are finally presented with charts, figures and tables.

In chapter five, the conclusion of the research work is presented. In this chapter, limitations of the system and further extensions that intend to make improvements on the research work are also discussed.

CHAPTER 2

BACKGROUND THEORY

This chapter first describes data structures of a digital image. Secondly, this chapter presents the color spaces and types of images. It then explains fundamental steps in digital image processing and machine learning for classification tasks. Finally, it presents related works with soil classification.

2.1 Digital Image Processing

Digital image processing is to develop a digital system that operates on a digital image with the aim of achieving an enhanced image or to get some useful information from it. It is also a kind of signal processing in which an image is considered as input, and its output could be an enhanced image or characteristics or features related to that image. The digital image processing is the processing of digital images by using a digital computer. Image processing is generally employed either to prepare an image for quantitative features measurement for object recognition or to enhance visual appearance. The digital image processing techniques are researched and developed for three major applications mainly:

- Image Processing (input and output are images),
- Image Analysis (input is an image, and output are measurements), and
- Image Understanding (input is an image, output is high-level description of the image).

2.1.1 Digital Image Data Structures

Digital images comprise picture elements or a set of points, pixels which are stored as an array of numbers. There are four types of digital image such as true color image (RGB image), intensity image (Grayscale image), binary image (Black/White Image), and index image. Images are spatial data indexed by two spatial coordinates, typically the variables x and y refer to the horizontal and vertical axes of an image.

A pixel value represents the intensity or color of each pixel. An image is often represented by using a multidimensional matrix [5]. For example, encoding of an RGB image comprises 3 matrices: one each for red, green and blue intensity as shown in Figure 2.1.

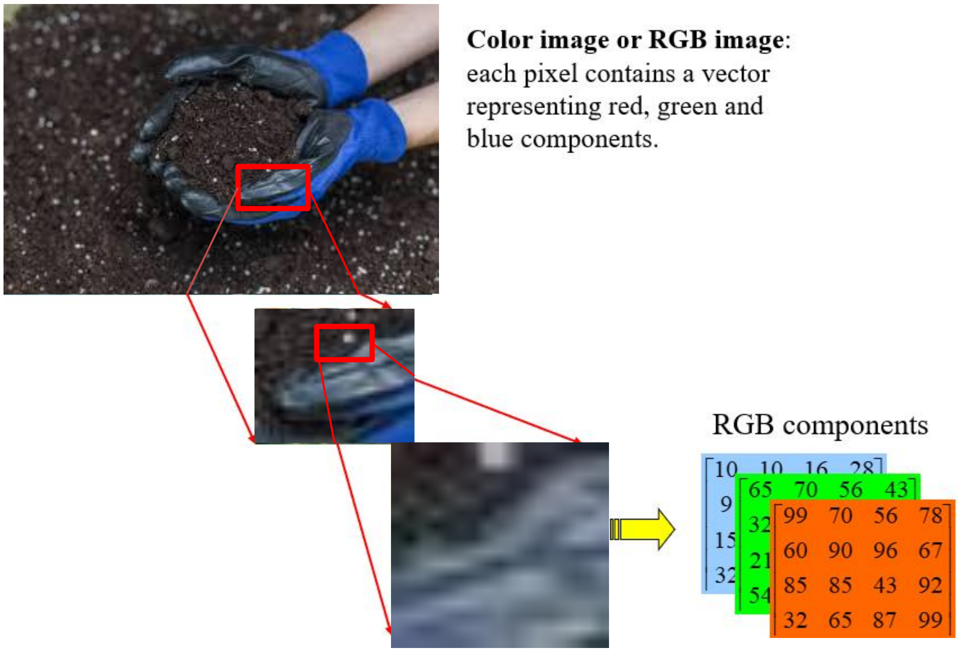

Color image or RGB image: each pixel contains a vector representing red, green and blue components.
RGB components
10 10 16 28
9
15
32
65 70 56 43
32
21
54
99 70 56 78
60 90 96 67
85 85 43 92
32 65 87 99

Figure 2.1 RGB Color Image Represented by Three Matrices

2.1.2 Types of Digital Images

People spend a lot of time in everyday while working with a large variety of digital raster images such as grayscale scans of printed documents, color photographs of people and landscapes, building plans, screenshots, faxed documents, and medical images such as X-rays and ultrasounds, and a multitude of others. All these images are, as a rule, ultimately represented by using two-dimensional ordered arrays of image elements, although all the different sources are used for these images.

2.1.2.1 Grayscale Image (Intensity Image)

A grayscale image data includes a single layer or channel that represents the brightness, intensity, or density of the image. In most cases, only positive intensity values make sense because the brightness values represent the intensity of light energy or density of film; and thus, these values cannot be negative, so typically whole integers in the range 0, . . ., 2^k-1 are used. A typical grayscale image uses k = 8 bits (1 byte) per pixel, and its intensity values are in the range 0-255, where the value 0 represents the minimum brightness (black), and 255 the maximum brightness (white).

8 bits per pixel is not sufficient for many professional print applications and photographs, as well as in astronomy and medicine. Image depths of 12, 14, and even 16 bits are often encountered in these domains. An RGB-encoded color image with an 8-bit depth would require 8 bits for each channel for a total of 24 bits, while the same image with a 12-bit depth would require a total of 36 bits. A sample grayscale image is shown in Figure 2.2.

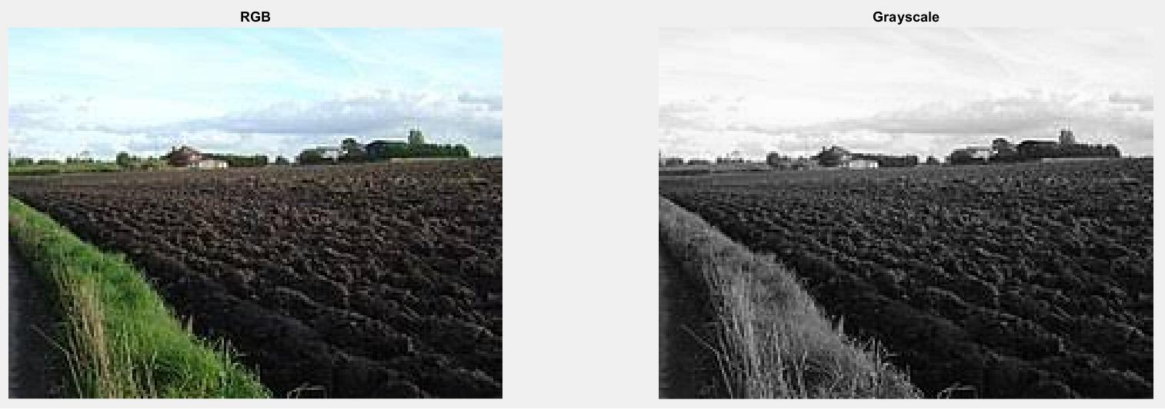


Figure 2.2 Grayscale Image (Intensity Image) and its Original RGB Image

2.1.2.2 Binary Image

Binary images are a special kind of intensity image in which pixels values can only take on one of two values, black or white.

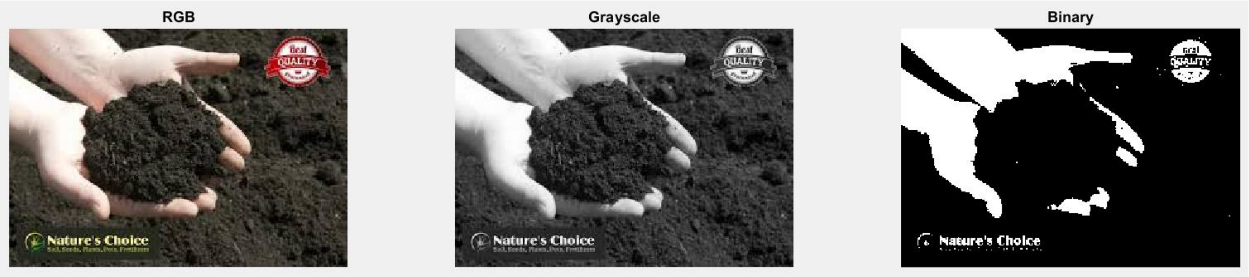


Figure 2.3 Binary Image and its Original RGB and Grayscale Images

Typically, these values are encoded using a single bit (0/1) per pixel. Binary images are often applied to representing archiving documents, line graphics, in electronic printing and encoding fax transmissions. A sample binary image is shown in Figure 2.3.

2.1.2.3 Color Image

Most color images are constructed by using the primary colors: red, green, and blue (RGB), in which typically 8 bits for each color component is used. Each pixel of a color image requires 3×8 = 24 bits to encode all three components, and each color component is ranging from 0 to 255. As with intensity images, color images with 30, 36, and 42 bits per pixel are commonly used in professional applications. Images with four or more color components are commonly applied to most pre-press applications, typically using the subtractive CMYK (Cyan-Magenta-Yellow-Black) color model while most color images include three components. A sample color image is shown in Figure 2.4.

Figure 2.4 Color Image

Palette or indexed images are a very special kind of color images. The difference between a true-color image and an indexed image is the number of different colors that can be applied to a particular image. In an indexed image, the image pixel values are only indices onto a specific table including selected full-color values.

2.1.3 Color Models

A color model is a color system that utilizes three primary colors to create a larger range of colors. Various kinds of color models used for different purposes exist, and each color model has a slightly different colors ranges which they can produce. The whole range of colors produced by a specific type of color model is called a color space. All color results from how our eye processes light waves, but depending on the type of media, creating that color comes from different methods [8].

2.1.3.1 RGB Color Space

The usage of the RGB color space has started in color television where Cathode Ray Tubes (CRT) was used. RGB color space is an example of a relative color standard. The primary colors (R-red, G-green, and B-blue) mimicked phosphor in CRT luminophore. The model uses additive color mixing to inform what kind of light needs to be emitted to produce a given color. The value of a particular color is expressed as a vector of three elements-intensities of three primary colors, and a transformation to a different color space is expressed by 3 × 3 matrices. Assume that values for each primary are quantized to $m = 2^n$ values; let the highest intensity value be $k = m-1$; then (0, 0, 0) is black, (k, k, k) is (television) white, (k, 0, 0) is pure red, and so on. The value k = 255 = 28 − 1 is common. There are 2563 = 224 = 16, 777, 216 possible colors in such a discretized space [8]. Figure 2.5 shows the RGB color space.

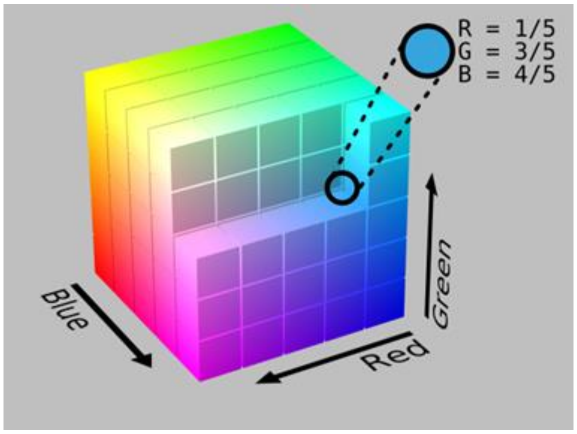


Figure 2.5 RGB Color Space

The RGB model may be thought of like a 3D coordinatization of color space; note the secondary colors which are combinations of two pure primaries. There are specific instances of the RGB color model such as RGB, Adobe RGB, and Adobe Wide Gamut RGB, which differ slightly in transformation matrices and the gamut [8]. One of the transformations between RGB and XYZ color spaces is as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.24 & -1.54 & -0.50 \\ -0.98 & 1.88 & 0.04 \\ 0.06 & -0.20 & 1.06 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.1}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.41 & 0.36 & 0.18 \\ 0.21 & 0.72 & 0.07 \\ 0.02 & 0.12 & 0.95 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.2}$$

2.1.3.2 HSV Color Space

HSV (Hue, Saturation and Value) color space is considerably closer to RGB color space in which humans describe color sensations and perceive color. Hue is the dominant color observably by humans. Saturation is the amount of white light assorted with hue. Value is the brightness (intensity). A HSV color space can be viewed as a geometric cylinder, where the angular dimension represents Hue (H), starting at the primary red at 0°, and moving to primary green at 120°, and then finally wrapping black to red at 360°. A saturation value moving towards the outer edge means that the colorfulness value is at the maximum for the color defined by the hue. The central vertical axis of HSV color space is the Value (V), ranging from black at the bottom with lightness or value 0, to white at the top with lightness or value 1 [8]. The HSV color space is illustrated in Figure 2.6.

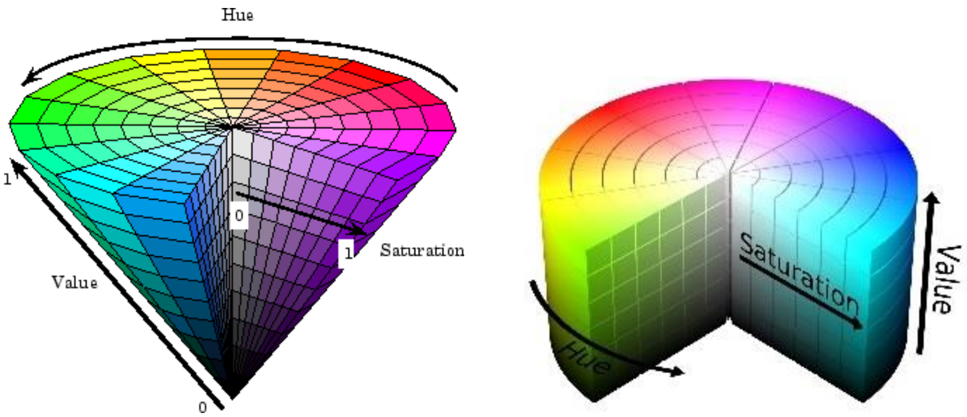


Figure 2.6 HSV Color Space

2.2 Phases of Digital Image Processing

Phases of Digital Image Processing are as follows. [12]

i. Acquisition

ii. Image Enhancement
iii. Image Restoration
iv. Color Image Processing
v. Wavelets and Multi-Resolution Processing
vi. Image compression
vii. Morphological Processing
viii. Segmentation Procedures
ix. Representation and Description
x. Object Detection and Recognition

Some of these fundamental steps are discussed in this section.

2.2.1 Image Enhancement

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

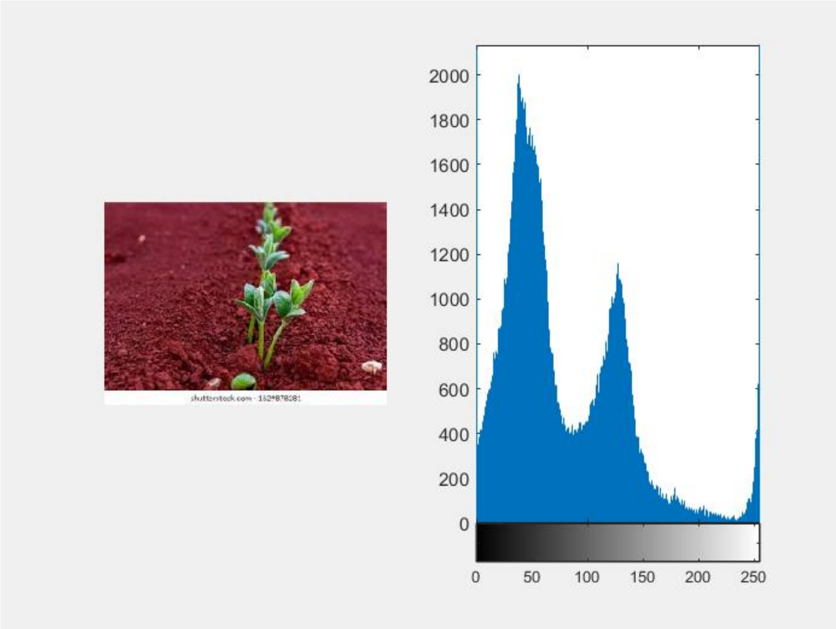


Figure 2.7 A Grayscale Image and its Histogram

The word specific is important here, because it establishes at the outset that enhancement techniques are problem oriented. Thus, for example, a method that is quite useful for enhancing X-ray images may not be the best approach for enhancing satellite images taken in the infrared band of the electromagnetic spectrum **[**12**]**.

A grayscale image and its histogram are depicted in Figure 2.8. Contrast of this image is enhanced by using the histogram equalization method. The contrast-enhanced resultant grayscale image and its histogram are as shown in Figure 2.9.

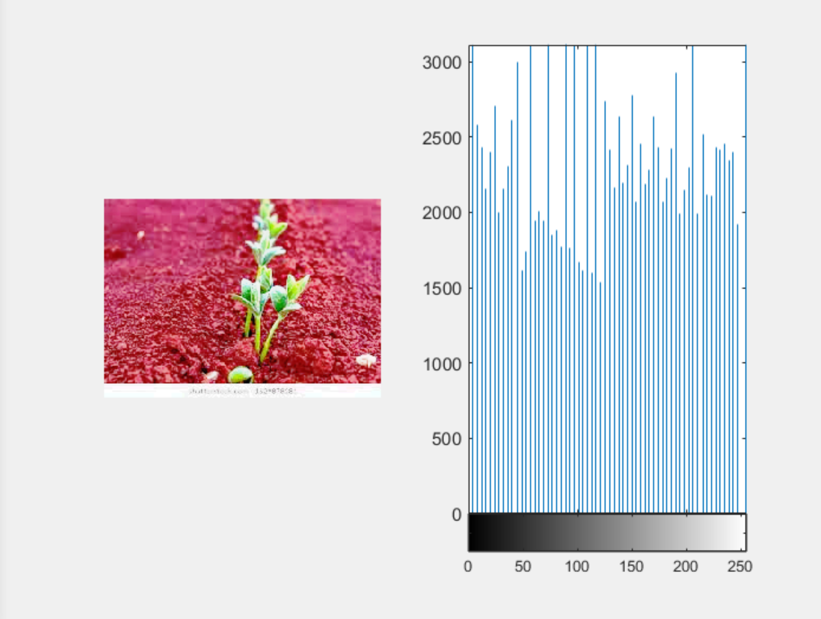


Figure 2.8 The Contrast-enhanced Grayscale Image and its Histogram

2.2.2 Color Image Processing

Color image processing is an area that has been gaining in importance because of the significant increase in the use of digital images over the internet. Color image processing composes of color models and color processing in a digital domain. Color is used also as the basis for extracting features of interest in an image.

The techniques of intensity (sometimes called density) slicing and color coding are the simplest and earliest examples of pseudocolor processing of digital images.

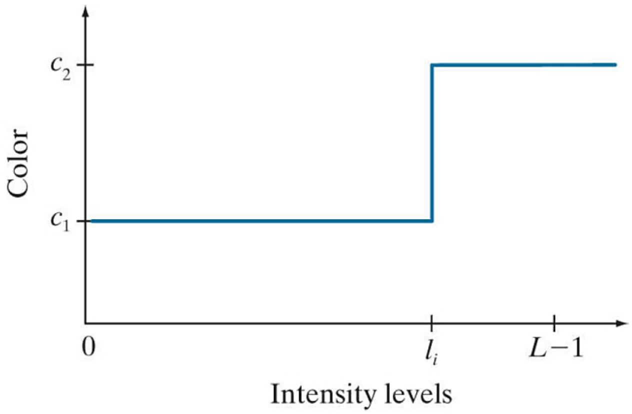


Figure 2.9 A Representation of the Intensity Slicing Technique

According to the mapping in this figure, any image intensity below level li is assigned one color, and any level above is assigned another. When more partitioning levels are used, the mapping function takes on a staircase form **[**12**]**. The intensity slicing is shown in Figure 2.10.

An example of intensity slicing is shown in Figure 2.11 including a terrain RGB image, its grayscale image, and a pseudo-colored image using intensity slicing into five colors.

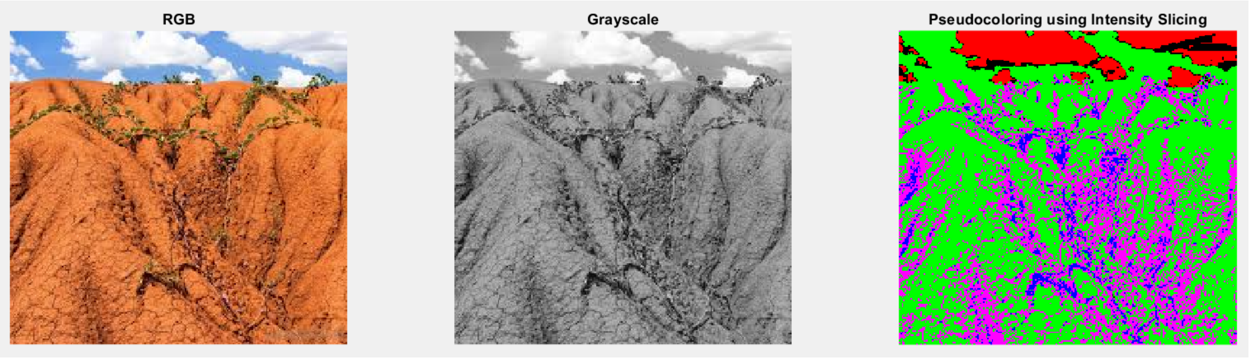


Figure 2.10 An Example of Color Image Processing, Pseudo Coloring using Intensity Slicing

2.2.3 Segmentation

The process of partitioning an image into different groups of pixels is called image segmentation. It segments an image into discrete regions in which having the pixel values with high similarity in each region, but high contrast forming edges among regions. There are many image segmentation techniques such as thresholding-based, edge-based, clustered-based, and neural network-based.

Among these, k-means clustering is popular, and it is an unsupervised machine learning algorithm. The k-means clustering algorithm can be applied to segment the region on interest (ROI) from the background or from other areas. As shown in Figure 2.12, the image is divided into two clusters using k-means clustering, and the foreground and the background are efficiently segmented.

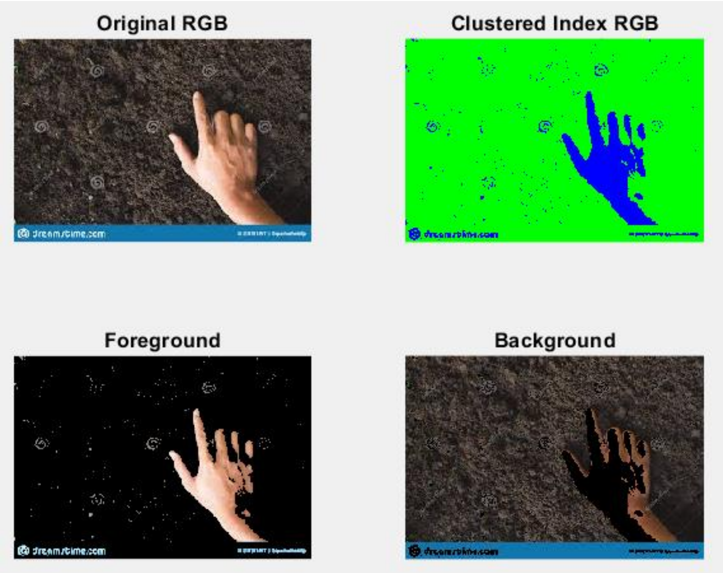


Figure 2.11 Segmentation in RGB Color Space (2 Clusters)

As shown in Figure 2.13, the image is divided into three clusters using k-means clustering, and three regions: black constituents, gray constituents and others components such as shovel and leaf are efficiently segmented.

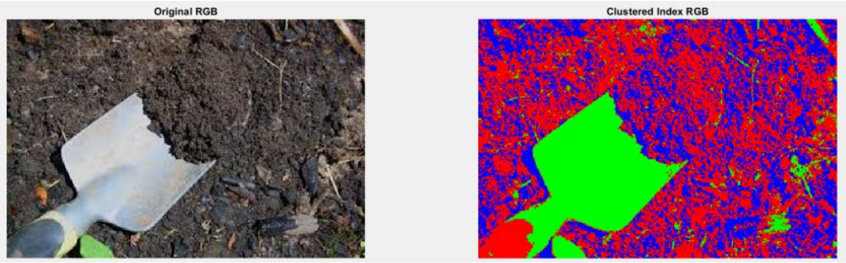


Figure 2.12 Segmentation in RGB Color Space (3 Clusters)

2.2.4 Feature Extraction

Features extraction can be used in many different domains such as diagnosis, identification, clustering, classification, detection and recognition. Image features extraction is used to get much information as feasible from the image. There exist many feature extraction methods, which may depend on color features, geometric features, statistical features, and texture features [20]. Feature is crucial in image processing. The different features of an image are domain specific features, or shape, texture and color [4].

Edges are very important features for representing the shape of a particular region or an object. There exist many edge detection methods. In Figure 2.14, edge detection result using high-pass filter. The high-pass passes over the high frequency components, and reduces or eliminates low frequency components of an image. By doing so, the high-pass filter can detect and enhance edges.

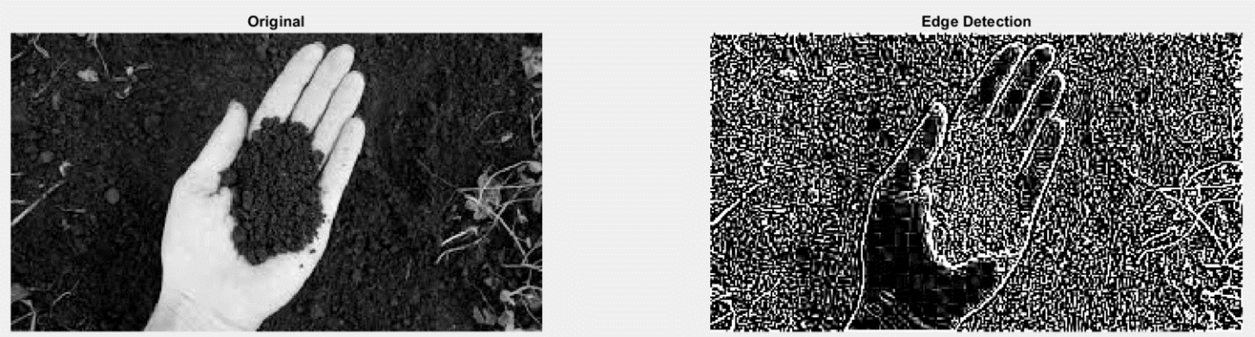


Figure 2.13 Edge Detection using High-pass Filter

In Figure 2.15, edge detection result using range filter is depicted. The Range filter detects edges and areas having different texture. The range filter result is the difference between local minimum and maximum values in the filter window multiplied by an adjustable gain factor that modulates the brightness and contrast of the result. The more the gain factor increases, the brighter the contrast of the filter result.

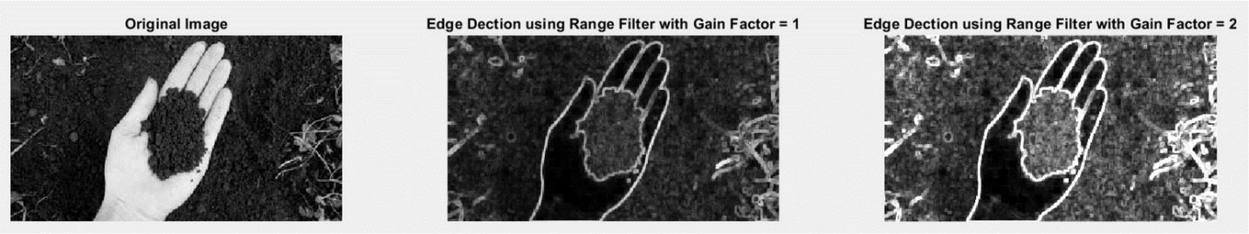

Original Image
Edge Dection using Range Filter with Gain Factor = 1
Edge Dection using Range Filter with Gain Factor = 2

Figure 2.14 Edge Detection using Range Filter

2.3 Artificial Intelligence (AI), Machine Learning (ML) and KNN

Artificial Intelligence (AI) is intelligence exhibited by machines. An ideal intelligent machine is a flexible rational agent that perceives its environment in computer science. It also takes actions that maximize its chance of success at some goal. Therefore, when a machine mimics a human-like behavior such as planning, learning, reasoning, the perception of the environment, natural language processing, problem-solving, and so on, it then falls under the category of Artificial Intelligence.

AI systems are not be pre-programmed. Instead, they use clever algorithms which can perform different kinds of tasks based on their own intelligence. They comprise ML algorithms such as deep neural networks, clustering algorithms and reinforcement learning algorithms. AI is being applied to many technologies such as optical character recognition (OCR), language translation, speech recognition, web search, objects detection and classification and health care. According to capabilities of the AI systems, they can be categorized into three types as follows:

(a) Weak AI

(b) General AI

(c) Strong AI

Weak AI and general AI are currently worked with. The future of AI is to create strong AI that will be more intelligent than humans.

Artificial intelligence and machine learning (ML) are included in computer science, and they are related each other. AI and ML are the most trending technologies used for developing intelligent systems. ML is a sub-field of AI, which enables machines to learn from past data or experiences without being explicitly programmed. ML works only for particular or specific domains. For example, if a ML model is created which is intended to detect pictures of dogs, it can detect only dog images, and, however, if a new or unknown data like a cat image is given to the ML model, it will then become misclassified or unresponsive.

Machine learning is being applied to different applications such as business intelligence, Facebook recommendation engines, human resource information systems (HRIS) and self-driving cars. It can be categorized into three types:

(a) Supervised learning

(b) Unsupervised learning and

(c) Reinforcement learning.

The KNN algorithm is one of the most broadly used machine learning algorithms for classification due to its simplicity and easy implementation. In many domain problems, it is also applied as the baseline classifier [18].

KNN is a lazy learner for it simply stores all the given training data as input without doing any calculations. It does classification only when the testing data is given to it. KNN is also known as a non-parametric algorithm because it does not require any prior knowledge concerning the training dataset. It assumes that instances or tuples in the dataset are identically and independently distributed. The instances or tuples which are close to each other, therefore, have the same classification. [19].

The distance between two points is calculated by Euclidean distance, a similarity measure, in KNN. The distance between two points: p and q with n elements is measured as in Equation 2.4:

$$d(p,q) = d(q,p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \qquad (2.4)$$
$$= \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

After measuring the distance between the testing data and the training data using the above equation, the k number of nearest neighbors to the testing data is then selected, and the majority class or label of the selected neighbors will become the predicted class or label for unknown testing data.

The K-NN working flow can be described with the following algorithm:

(a) Define the number of nearest neighbors, K

(b) Find Euclidean distances between the testing data and each of the training data

(c) Get the K nearest neighbors based on the Euclidean distances

(d) Count the number of the data points in each class or category among the K nearest neighbors

(e) Assign the new data points to that category for which the number of the neighbor is maximum.

Figure 2.15 shows the distance calculation between two data points in K-NN.

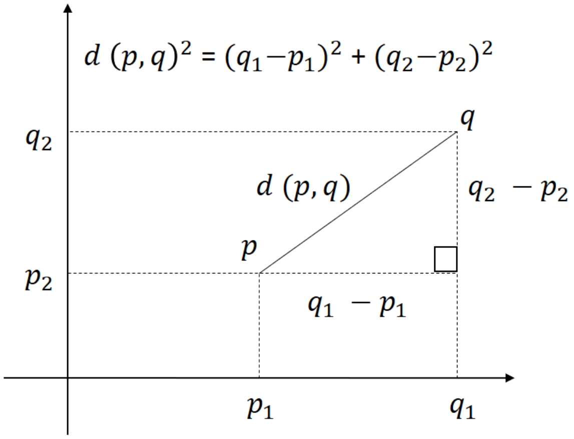


Figure 2.15 Distance Calculation between Two Data Points

The advantages of using KNN are insensitive to outliers, high accuracy, and no assumption about data. KNN does work with both nominal and numeric values. However, KNN requires an expensive computation and a lot of memory.

2.4 Related Works

With the advent of AI and ML, various methods have been proposed to facilitate soil classification task in literature. The research works related to this thesis in literature are discussed in this section.

2.4.1 Soil Classification using ML and Crop Suggestion

In [14], S. A. Z. Rahman and K. Chandra Mitra have proposed a model or a method that can predict soil series together with land types, and according to prediction results, their model can be suggested suitable crops for a certain soil type. Gaussian kernel-based Support Vector Machines (SVM) and the weighted KNN, and Bagged Trees are applied to soil classification in their research. The proposed SVM based method achieves better results than many existing classification methods according to the experimental results. The remark for their research is that prior survey on soil series of a country, chemical and geographical features are must be performed.

2.4.2 Soil Classification using Self-Organizing Map and k-means

In [15], Sofianita Mutalib and S. Abdul-Rahman have used k-means and self-organizing map (SOM) in the classification model. The inputs to their model are texture, color, terrain and drainage class. The classification rate for the SOM is 91.8%, and k–means, 79.8%, respectively. The remark for their research is that prior survey on soil texture, color, terrain and drainage class features are must be first measured, and the classification rate of the model based on k–means is about 79.8 %.

2.4.3 Performance of SVM Classifier for Image-based Soil Classification

In [16], Srunitha. K. and S. Padmavathi, have explained soil types classification using support vector machine. The texture features from soil images are taken using color quantization technique, Gabor filter and the low pass filter. The statistical parameters: Standard deviation, mean amplitude and HSV histogram, are also extracted. The remark for their research is that, although the proposed method works effectively with sand and clay soil types, it provides poor results for the peat soil with 58.7% accuracy.

2.5 Summary

This chapter describes the theory background concerned with digital image processing, machine learning and soil classification in detail. Digital image data structures, types of digital images, color models and fundamental steps in digital image processing are discussed and referenced in detail. Moreover, the relation among AI, ML and KNN is also described. Finally, the research works related to this thesis in literature are discussed using descriptions and remarks in this chapter.

CHAPTER 3

SOIL CLASSIFICATION SYSTEM

In this chapter, methods applied to the proposed soil classification system are described in detail. In features extraction method, color features: mean and standard deviation, and texture features: energy and contrast are extracted from soil RGB images. In soil classification method, features similarity between the features dataset and the tested features dataset is calculated using KNN.

3.1 Features Extraction

Feature extraction is almost always performed after achieving the output of a segmentation process, which usually is raw pixel data, including either all the points in the region itself or the boundary of a region. Feature extraction comprises feature detection and feature description. Firstly, feature detection means that the features in an image, region, or boundary is being found. Secondly, feature description means assigning quantitative attributes to the detected features [12].

Features extraction can be used in many different domains such as diagnosis, identification, clustering, classification, detection and recognition. Image features extraction is used to get much information as feasible from the image. There exist many feature extraction methods, which can be depended on color features, geometric features, statistical features, and texture features [20]. Feature is crucial in image processing. The different features of an image are domain specific features, or shape, texture and color [4].

Features extraction is to simplify resource numbers needed to represent a large dataset accurately. When complex data analysis is performed, one of the major problems is stemming from the involved variables number. When performing analysis on a large number of variables, it generally requires a large amount of computation power and memory, or a classification algorithm which may overfit the training data, and generalizes poorly to new testing data. Feature extraction is a general term for methods which construct variables combination to solve these problems while it can still represent the data with acceptable rate [11].

3.1.1 Image Resizing

Before features extraction, soil RGB images which are clay, clay loam and sandy loam soil images taken in plantations and farms in Pyin Oo Lwin and Lashio townships, and collected from Internet is first applied to our soil classification system. Some of the soil images in the dataset has high dimensions (resolution), and they should be resized because of time and space complexity. Image resizing is depicted in Figure 3.1.

Figure 3.1 Image Resizing

The three major soil types in Shan state: clay, clay loam and sandy loam [7] have different patterns: different texture and color features as shown in Figure 3.2. Therefore, color features and texture features of a soil image can be used to classify a particular soil type.

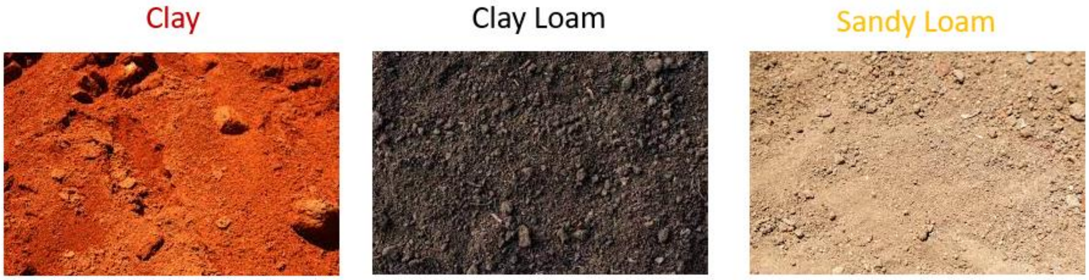


Figure 3.2 Three Soil Types used in the System

3.1.2 Texture Features

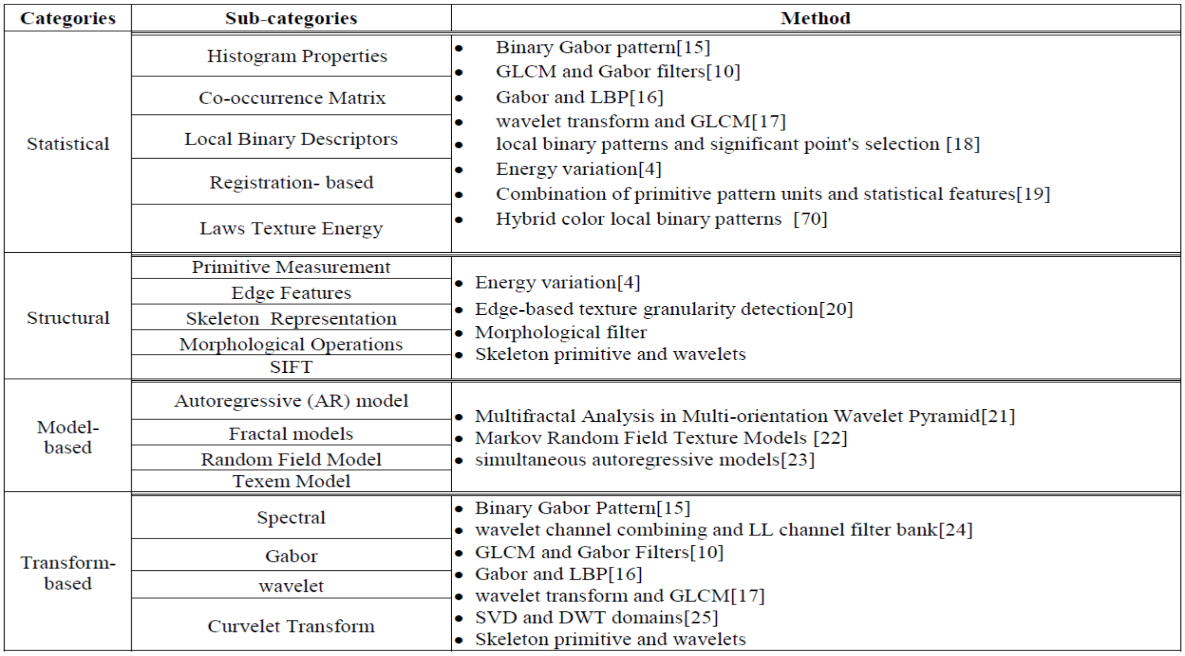
Texture includes a significant amount of information about the basic arrangement of the surface that is fabric, clouds, bricks, leaves etc. It also defines the relationship between the surface and its environment. The physical composition of surface is also described by the texture [3].

Texture is marked as one of the crucial features of every image. Gray level co-occurrence matrix (GLCM) is used to obtain the second order statistical features for any image, and GLCM operates on spatial domain. Some of the texture features proposed by Haralick are entropy, energy, correlation, homogeneity, maximum probability, contrast and dissimilarity [4].

The aim of texture analysis is to find a unique way of representing the underlying characteristics of textures. It then represents them in some simpler but unique form, so that they can be applied to robust classification and accurate objects segmentation. Although texture is very much important in pattern recognition and image analysis, only a few architectures concerned with on-board textural feature extraction has been implemented. GLCM, one of the few architectures is formulated to obtain statistical texture features [11].

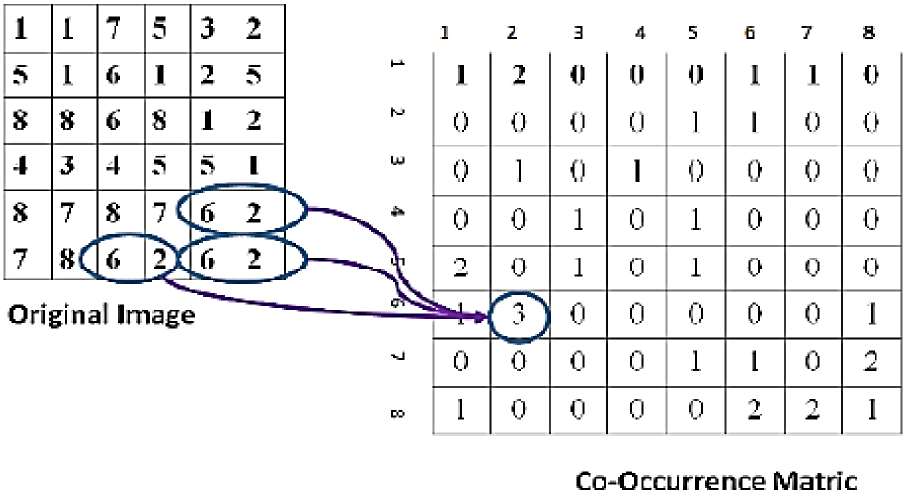
Many features and methods which are used to represent the texture are summarized in Table 3.1.

Table 3.1 Methods and Features used to describe Texture [6]



Categories	Sub-categories	Method
Statistical	Histogram Properties Co-occurrence Matrix Local Binary Descriptors Registration- based Laws Texture Energy	• Binary Gabor pattern[15] • GLCM and Gabor filters[10] • Gabor and LBP[16] • wavelet transform and GLCM[17] • local binary patterns and significant point's selection [18] • Energy variation[4] • Combination of primitive pattern units and statistical features[19] • Hybrid color local binary patterns [70]
Structural	Primitive Measurement Edge Features Skeleton Representation Morphological Operations SIFT	• Energy variation[4] • Edge-based texture granularity detection[20] • Morphological filter • Skeleton primitive and wavelets
Model-based	Autoregressive (AR) model Fractal models Random Field Model Texem Model	• Multifractal Analysis in Multi-orientation Wavelet Pyramid[21] • Markov Random Field Texture Models [22] • simultaneous autoregressive models[23]
Transform-based	Spectral Gabor wavelet Curvelet Transform	• Binary Gabor Pattern[15] • wavelet channel combining and LL channel filter bank[24] • GLCM and Gabor Filters[10] • Gabor and LBP[16] • wavelet transform and GLCM[17] • SVD and DWT domains[25] • Skeleton primitive and wavelets

The co-occurrence matrix introduced by Haralick is popular for extracting texture features. The co-occurrence matrix of an intensity image is built depending on the correlations between image pixels. For an image whose size of k-bit with L=2k brightness levels, an L×L matrix is constructed and its elements are the number of occurrences of a pair of pixels with brightness of a and b, different pixels separated by d, some distance, pixels in a certain direction. The textual characteristics of the second statistic is calculated after calculating the matrix [6].



1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Original Image

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

Co-Occurrence Matric

Figure 3.3 Extraction a Co-Occurrence Matrix with 8 Intensity Levels

For example, in Figure 3.3, an image whose size of 3-bit with 8 intensity levels is shown, and its co-occurrence matrix has 8 columns and 8 rows. The matrix elements are the pixel occurrences number with gray intensity levels, *i* and *j*, which are represented by a displacement of 1 pixel in the direction of zero degrees [6].

The gained co-occurrence matrix is split by the summation of all the indecisions in order to gain a normalized GLCM matrix [17]. The normalized GLCM is shown in Figure 3.4.

$$G = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

GLCM

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} = \frac{1}{\sum a_{i,j}} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Normalized GLCM

Figure 3.4 Normalized GLCM [10]

Typically, the co-occurrence matrix is created by defining for the four main directions (0, 45, 90, and 135). In Figure 3.5, four possible angles between two pixels with angles (0, 45, 90 and 135) degrees are represented with a displacement of 3 between two pixels [6].

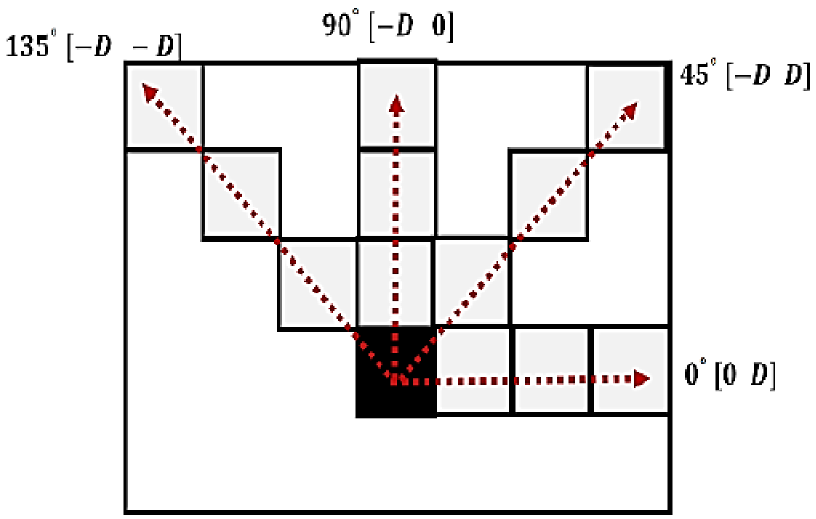


Figure 3.5 Four Different Directions with Displacement 3 between Two Pixels

After constructing the co-occurrence matrix, the Haralick statistical features can be calculated from the co-occurrence matrix output [6]. Haralick defined 14 statistical features from the GLCM to be applied to texture classification. These features are, however, strongly correlated [9]. Generally, the prominent four features are energy, homogeneity, entropy and contrast in 14 Haralick's features. These prominent features are calculated based on GLCM, and can be summarized as follows: [17]

Contrast is the measure of the local variations in the gray-level co-occurrence matrix.

$$Contrast = \sum P_{i,j}(i-j)^2 \qquad (3.1)$$

Entropy is the quantity of energy.

$$Entropy = \sum P_{i,j}\ x\ log P_{i,j} \qquad (3.2)$$

Homogeneity computes the not-zero in the GLCM, and it is the inverse of weight of the contrast. The value of the homogeneity is ranging from 0 to 1.

$$Homogeneity = \sum \frac{P_{i,j}}{1+(i-j)^2} \qquad (3.3)$$

Energy computes the local homogeneity, and the value of the energy is ranging from 0 to 1.

$$Energy = \sum(P_{i,j})^2 \tag{3.4}$$

Where: P = Normalized GLCM, i = row and j = column.

In this thesis, energy and contrast of a normalized GLCM transformed from a soil image are used as texture features in features extraction of the system.

3.1.3 Color Features

Greyscale texture features are popular in image processing domain, and provide enough information to solve many different tasks. Many researchers, however, have started to take color information and features into consideration for the human eye perceives any image as a combination of shape, color and texture [9]. Moreover, in [13], color features: mean, median and standard deviation are used to retrieve images. Therefore, the combination of texture and color features is applied to our proposed soil classification method.

Color distribution moments are the features which can be extracted from any image, and they can be used to represent the image, and applied to image matching and image retrieval. To be a robust representation of color distributions of all images, the first-order color moment, mean, the second order color moment, standard deviation, and the third-order color moment, skewness, have been proved. Color moments are very compact features to represent an image while comparing with other color features, as only 9, 3 values for each layer of a color image, numerical values are applied to representing the color content of a color image [1].

Color moment measures the color distribution in any image. It is mainly applied to color indexing purposes to compare how similar two images are by using color moments. It is a proven feature, and it can be used for images taken under changing illumination conditions, and suitable for all color models [2]. Two types of color moments or features: mean and standard deviation, are used as color features in our proposed method.

The first color feature or moment, mean, can be represented as the average or mean color in the image, and it can be calculated by using Equation 3.5:

$$\mu = \sqrt{\frac{\sum_{i=1}^{n} x_i}{N}} \tag{3.5}$$

Where: µ =mean, N = the number of pixels in the image, and x_i= value of the i^{th} pixel in the image.

The second color feature or moment is the standard deviation, which is achieved as the square root of the variance of the color distribution. The variance for every image pixel is calculated by subtracting the mean from each image pixel's intensity value. All the resulting values are then squared, and the squared results are summed. The summing result is then divided by the number of image pixels, and the variance is obtained. The standard deviation can be calculated using Equation 3.6.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.6}$$

Where: σ = standard deviation, N = the number of pixels in the image, µ = mean and x_i= value of the i^{th} pixel in the image.

3.1.4 Color and Texture Features Extraction in the System

The color and texture features of soil RGB images are extracted in the system by using the above color and texture features methods, and these features are as shown in Figure 3.6.

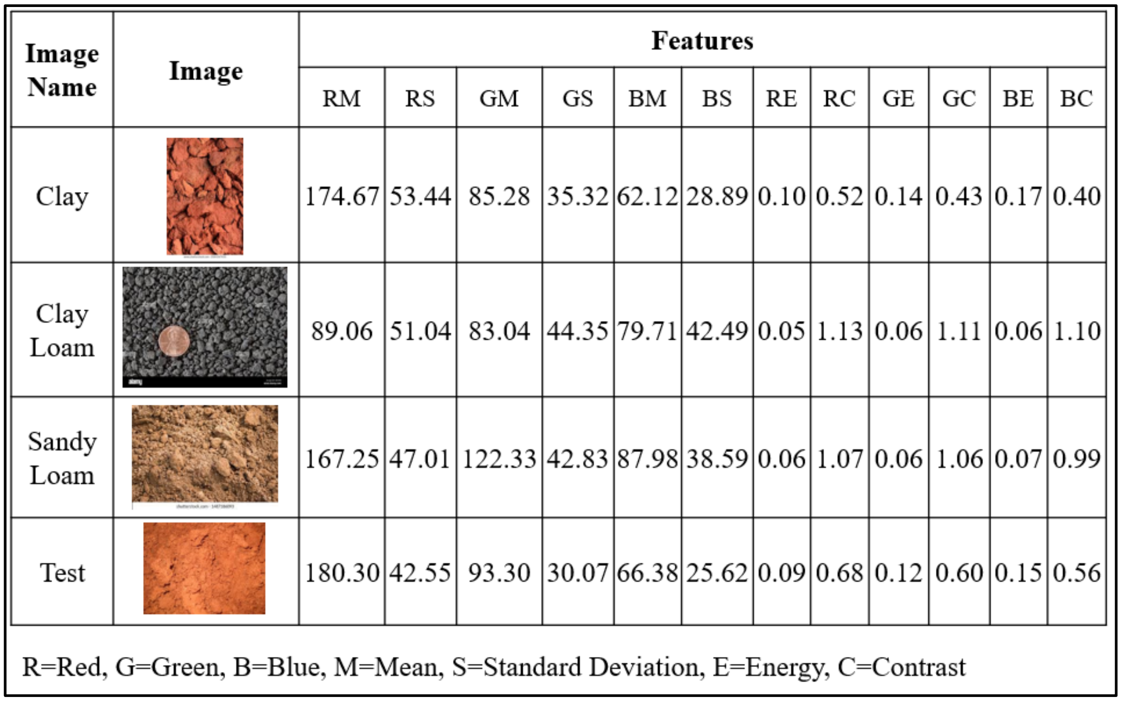


Image Name	Image	Features											
		RM	RS	GM	GS	BM	BS	RE	RC	GE	GC	BE	BC
Clay		174.67	53.44	85.28	35.32	62.12	28.89	0.10	0.52	0.14	0.43	0.17	0.40
Clay Loam		89.06	51.04	83.04	44.35	79.71	42.49	0.05	1.13	0.06	1.11	0.06	1.10
Sandy Loam		167.25	47.01	122.33	42.83	87.98	38.59	0.06	1.07	0.06	1.06	0.07	0.99
Test		180.30	42.55	93.30	30.07	66.38	25.62	0.09	0.68	0.12	0.60	0.15	0.56

R=Red, G=Green, B=Blue, M=Mean, S=Standard Deviation, E=Energy, C=Contrast

Figure 3.6 Color and Texture Features from Soil RGB Images

As depicted in Figure 3.6, there are 12 features for a soil RGB image and the range of values of 12 features is different from each other. The different in range of features can lead to a challenging classification task for KNN. The solution to this challenging task is to normalize features values by using the normalization technique used in Equation 3.7 as follows:

$$V_n = \frac{V_c - V_{min}}{V_{max}} \tag{3.7}$$

Where: V_n = normalized value, V_c = current value, V_{min}= minimum value, and V_{max} = maximum value. Normalization makes the values of all the features ranging from 0 to 1. The normalized features are as shown in Figure 3.7.

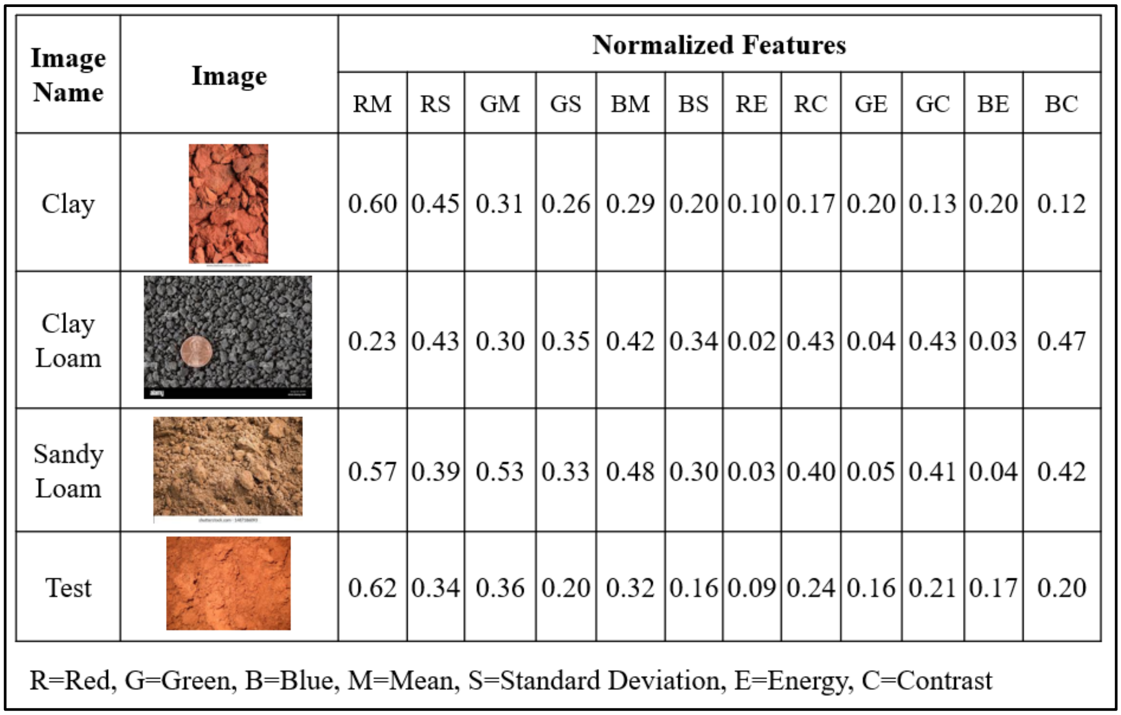


Image Name	Image	Normalized Features											
		RM	RS	GM	GS	BM	BS	RE	RC	GE	GC	BE	BC
Clay		0.60	0.45	0.31	0.26	0.29	0.20	0.10	0.17	0.20	0.13	0.20	0.12
Clay Loam		0.23	0.43	0.30	0.35	0.42	0.34	0.02	0.43	0.04	0.43	0.03	0.47
Sandy Loam		0.57	0.39	0.53	0.33	0.48	0.30	0.03	0.40	0.05	0.41	0.04	0.42
Test		0.62	0.34	0.36	0.20	0.32	0.16	0.09	0.24	0.16	0.21	0.17	0.20

R=Red, G=Green, B=Blue, M=Mean, S=Standard Deviation, E=Energy, C=Contrast

Figure 3.7 Normalized Color and Texture Features from Soil RGB Images

3.2 Soil Classification

The above normalized features are built as a features dataset, and these features are used in comparing with the normalized features of a tested soil image by using KNN for soil classification. The KNN algorithm is one of the most widely used machine learning algorithms for classification due to its simplicity and easy implementation. In many domain problems, it is also applied as the baseline classifier.

The distance between two points is calculated by Euclidean distance, a similarity measure, in KNN. The distance between two points: p and q with n elements is measured as discussed in Equation 2.4 in the previous chapter.

After measuring the distance between the testing data and the training data, the k number of nearest neighbors to the testing data is then selected, and the majority class or label of the selected neighbors will become the predicted class or label for unknown testing data.

As an example of our proposed soil classification, normalized features shown in Figure 3.7 are used to classify a tested input soil image using the following steps:

Step 1: Find the distance (d_1) between clay image and test image.

$$d_1 = \sqrt{(0.60-0.62)^2+(0.45-0.34)^2+\ldots+(0.20-0.17)^2+(0.12-0.20)^2} = 0.2035$$

Step 2: Find the distance (d_2) between clay loam image and test image.

$$d_2 = \sqrt{(0.23-0.62)^2+(0.43-0.34)^2+\ldots+(0.03-0.17)^2+(0.47-0.20)^2} = 0.6519$$

Step 3: Find the distance (d_3) between sandy loam image and test image.

$$d_3 = \sqrt{(0.57-0.62)^2+(0.39-0.34)^2+\ldots+(0.04-0.17)^2+(0.42-0.20)^2} = 0.4925$$

Step 4: Sort the distances in ascending order.

$$distances = [0.2035, 0.4925, 0.6519]$$

Step 5: Find the nearest distance(s) based on the number of nearest neighbor value, k. If k = 1, the nearest neighbor or distance is, $nearest\ distance = 0.2035$.

Step 6: The nearest distance, 0.2035, is the distance between test image and clay image. According to the nearest distance, the test image in Figure 3.7 is classified as a clay soil image by the proposed system.

3.3 Summary

In this chapter, the proposed methods applied to the system are described in detail. Color and texture features are first widely discussed because of their importance in representing a soil image to the system. Feature extraction using the texture and color features is then presented using figures. Moreover, the normalization method is also proposed to normalize features within the range of 0-1. Finally, soil classification using KNN is discussed using a sequence of steps. All the processes of the proposed soil classification system are presented in detail in this chapter.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

Design and implementation of the proposed soil classification system using KNN are described in this chapter. First of all, the overview of system design and the architecture of the system are described in order to understand the process flow of the proposed soil classification system. The implementation of all the processes in the proposed system using MATLAB programming environment is then discussed with User Interfaces (UI). The performance analysis and the experimental results are finally presented with charts, figures and tables in this chapter.

4.1 Overview Design of the System

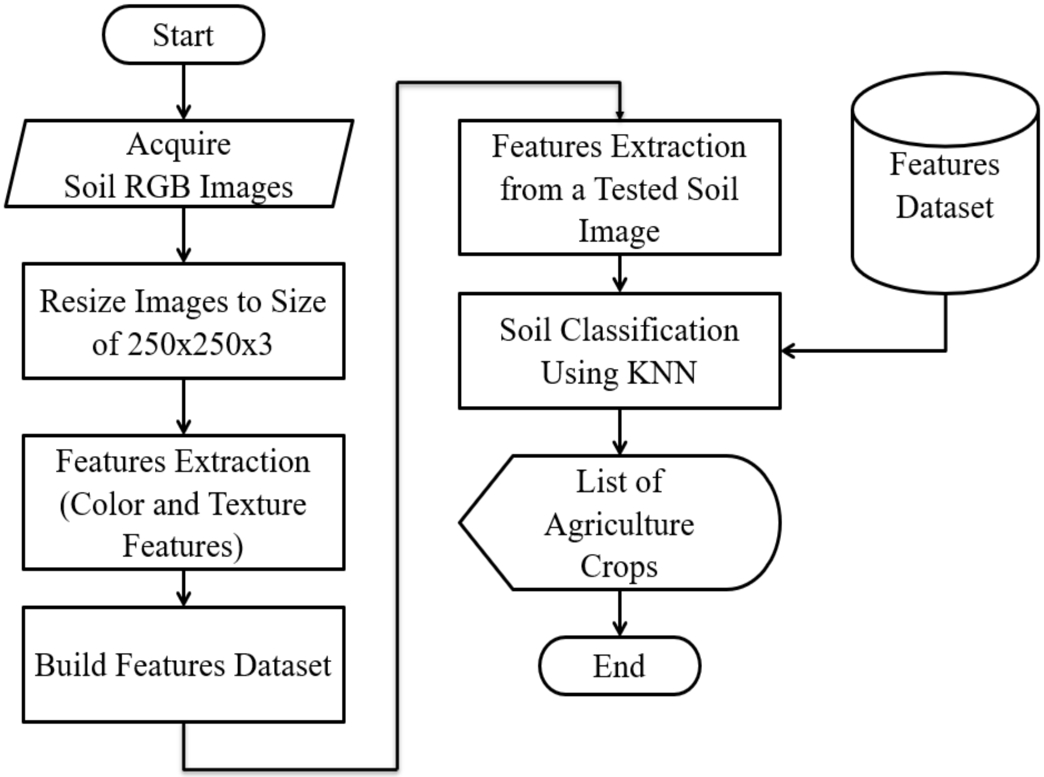

Start
Acquire Soil RGB Images
Resize Images to Size of 250x250x3
Features Extraction (Color and Texture Features)
Build Features Dataset
Features Extraction from a Tested Soil Image
Features Dataset
Soil Classification Using KNN
List of Agriculture Crops
End

Figure 4.1 Overview Design of the System

The overview design of the proposed soil classification system is shown in Figure 4.1. The proposed soil classification system is implemented as the soil classification system by using KNN. The contribution of the proposed system is color and texture features extraction and soil classification based on these extracted features using KNN as a classifier. In this system, three main steps are essential.

In the first step, pre-preprocessing, which consists of acquiring soil images and image resizing, is performed. The soil images are resized as size of 250x250x3. This system accepts various image types such as Portable Network Graphics, PNG (.png), Joint Photographic Experts Groups, JPEG (.jpg, .jpeg), JPEG File Interchange Format, JFIF (.jfif), and Tagged Image File Format, TIFF (.tif, .tiff).

In the second step, color and texture features extraction from the soil images is done and the features dataset is built using the extracted features vectors as described in section 3.1.4 of the previous chapter.

In the final step, a tested soil image is inserted into the system and the features from this image are also extracted. The tested soil image is classified based on distances between the features vector of the tested soil image and the features vectors in the features dataset by using KNN as a classifier.

After soil classification, the system also provides a list of agriculture crops that can be grown easily on the soil in the tested image.

4.2 Architecture of the System

The architecture of the system is shown in Figure 4.2.

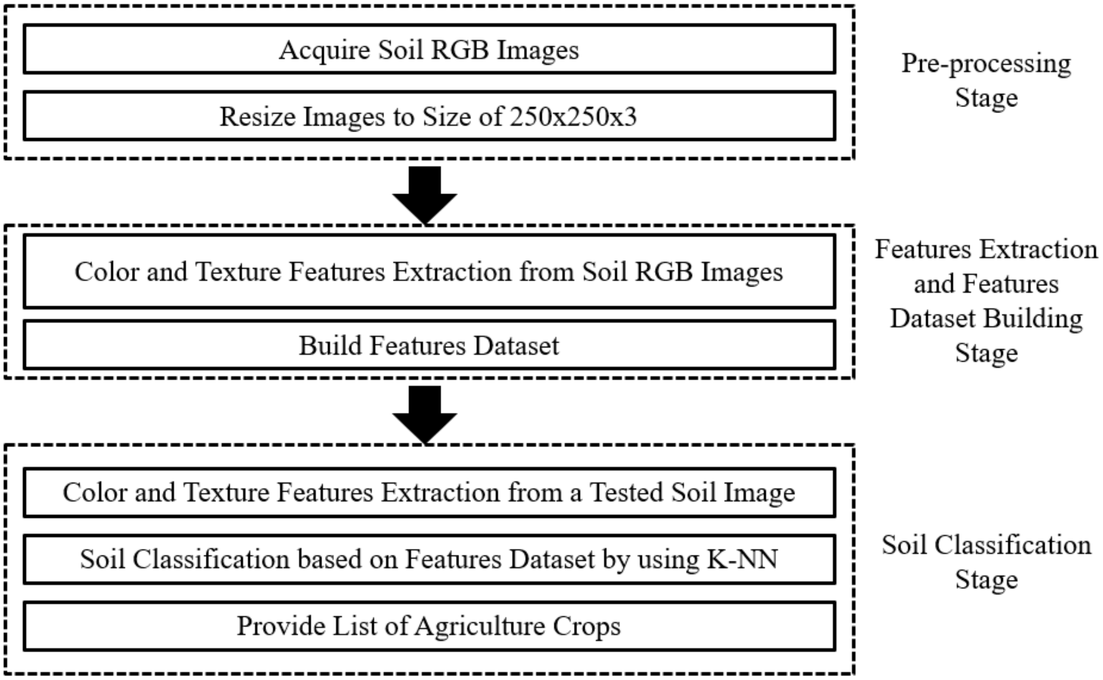

Acquire Soil RGB Images
Resize Images to Size of 250x250x3
Pre-processing Stage
Color and Texture Features Extraction from Soil RGB Images
Build Features Dataset
Features Extraction and Features Dataset Building Stage
Color and Texture Features Extraction from a Tested Soil Image
Soil Classification based on Features Dataset by using K-NN
Provide List of Agriculture Crops
Soil Classification Stage

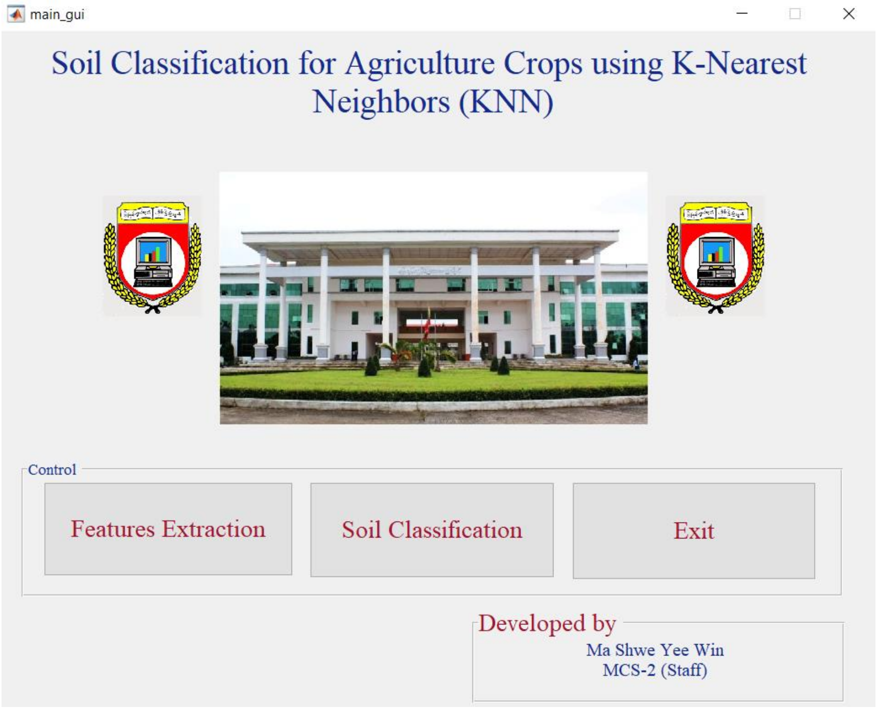
Figure 4.2 Architecture of the System

The architecture of the proposed system comprises pre-processing stage, features extraction and features dataset building stage, and soil classification stage. Image resizing in the pre-processing stage is important for time and space complexity of the system because the space and time complexity directly depends on the size of soil images. As a classification system, valid and useful features are essential to it. This stage is accomplished in the features extraction and features dataset building stage. In the soil classification stage, features from a tested soil images are gathered and it is classified by comparing with the features dataset using KNN. Finally, the system provides soil classification result along with a list of suitable corps for the resultant soil type.

4.3 Implementation of the system

The soil classification system for agriculture crops using KNN is implemented based on the overview design and the architecture of the system shown in Figure 4.1 and Figure 4.2, respectively. The MATLAB programming language is applied to implementation of the system.

The main graphical user interface (GUI) of the system consists of two important buttons: Features Extraction and Soil Classification as depicted in Figure 4.3.



Soil Classification for Agriculture Crops using K-Nearest Neighbors (KNN)

Developed by
Ma Shwe Yee Win
MCS-2 (Staff)

Figure 4.3 Main GUI of the System

4.3.1 Implementation of Features Extraction and Building Features Dataset

Before features extraction and building features dataset, as acquisition of soil RGB images, soil images dataset that contains "sandy loam" and "clay loam" (Red Earths and Yellow Earths) soil images taken in plantations and farms in Lashio township, and collected from Internet is applied to our soil classification system. Our own soil image dataset including 200 soil images is applied to the system with the purpose of building the features dataset and testing the system. 150 soil images in the dataset are used for building the features dataset and 50 soil images are, for testing the system, as unknown data. Some images from our own soil image dataset are shown in Figure 4.4.

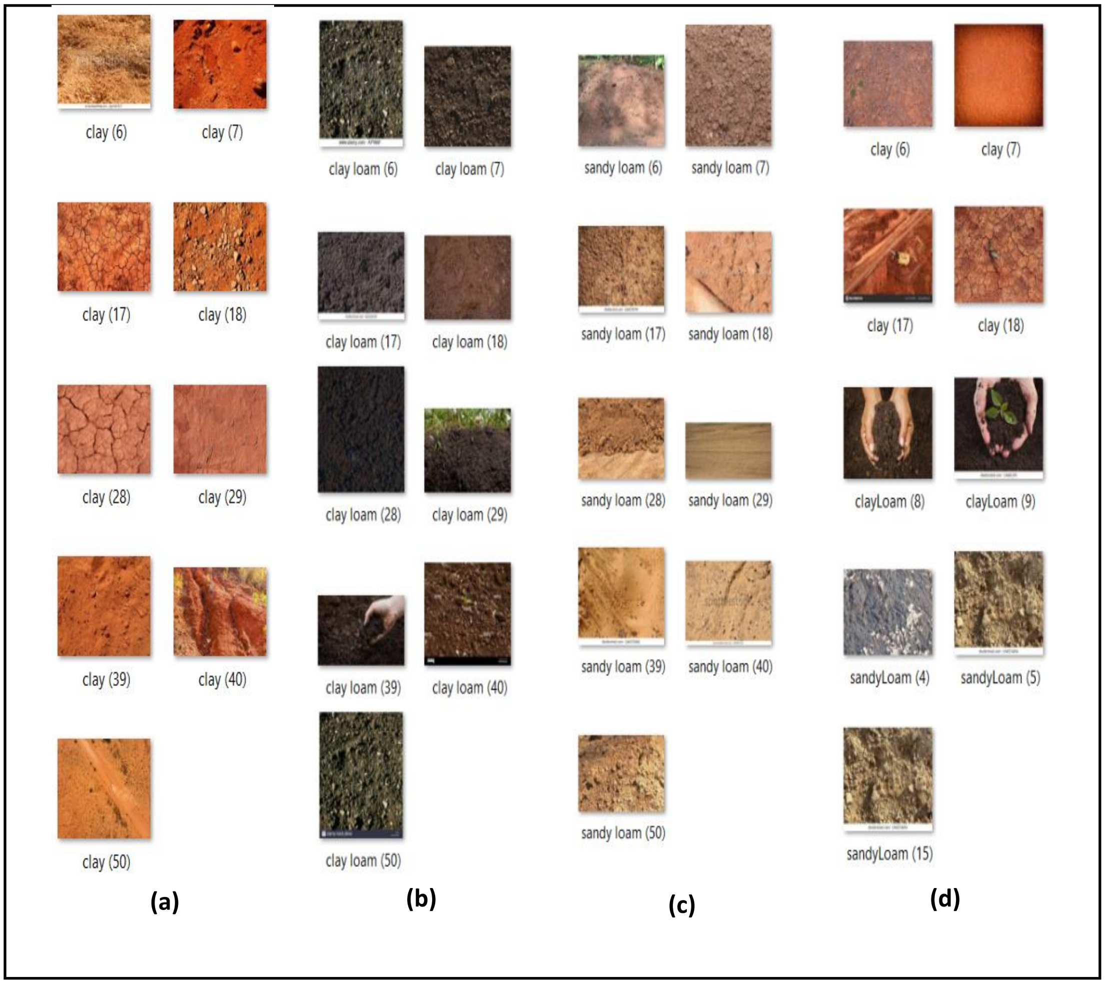
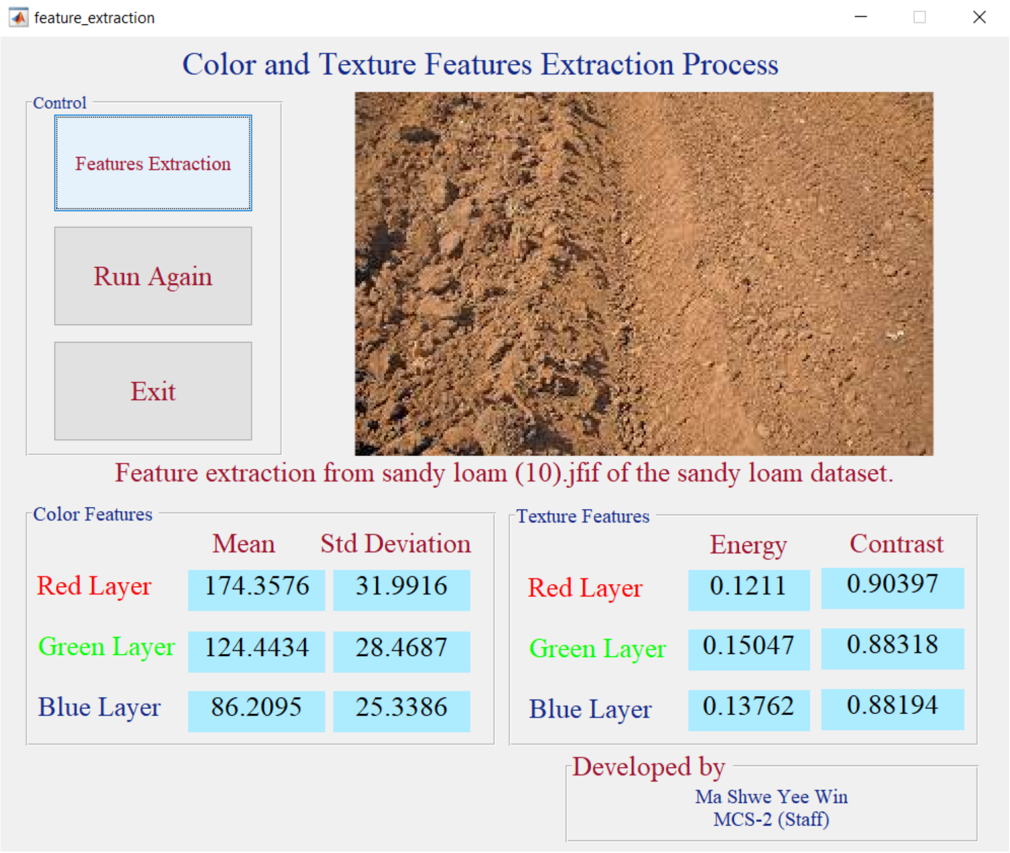


Figure 4.4 Some Images from our own Soil Image Dataset (a) Clay (b) Clay Loam (c) Sandy Loam (d) Tested Soil Images

In features extraction, color features: mean and standard deviation, and texture features: energy and contrast are extracted from each layer of RGB soil images as discussed in section 3.1 of the previous chapter. Feature extraction and its results are as illustrated in Figure 4.5.



Color and Texture Features Extraction Process

Feature extraction from sandy loam (10).jfif of the sandy loam dataset.

Color Features

	Mean	Std Deviation
Red Layer	174.3576	31.9916
Green Layer	124.4434	28.4687
Blue Layer	86.2095	25.3386

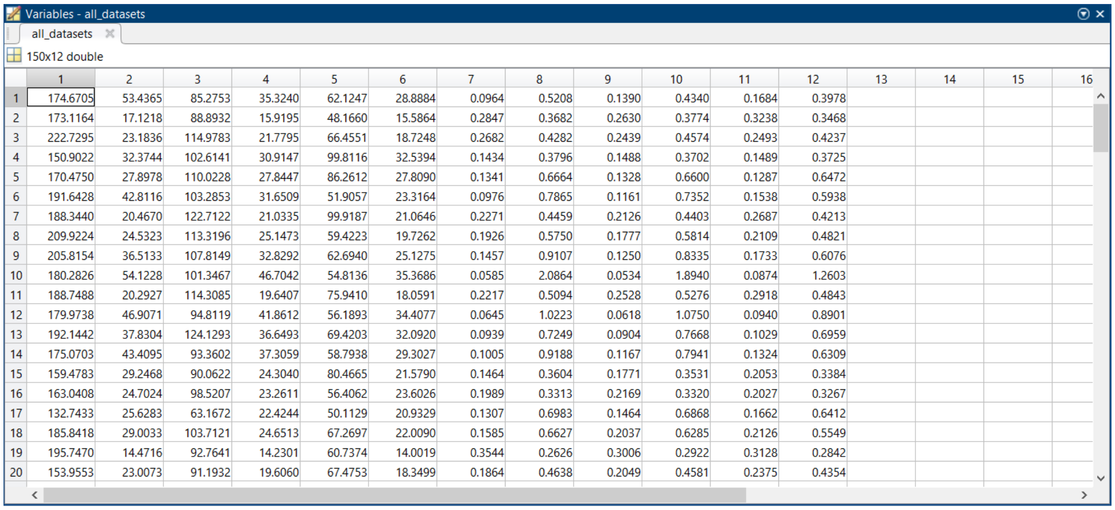
Texture Features

	Energy	Contrast
Red Layer	0.1211	0.90397
Green Layer	0.15047	0.88318
Blue Layer	0.13762	0.88194

Developed by
Ma Shwe Yee Win
MCS-2 (Staff)

Figure 4.5 Features Extraction GUI of the System

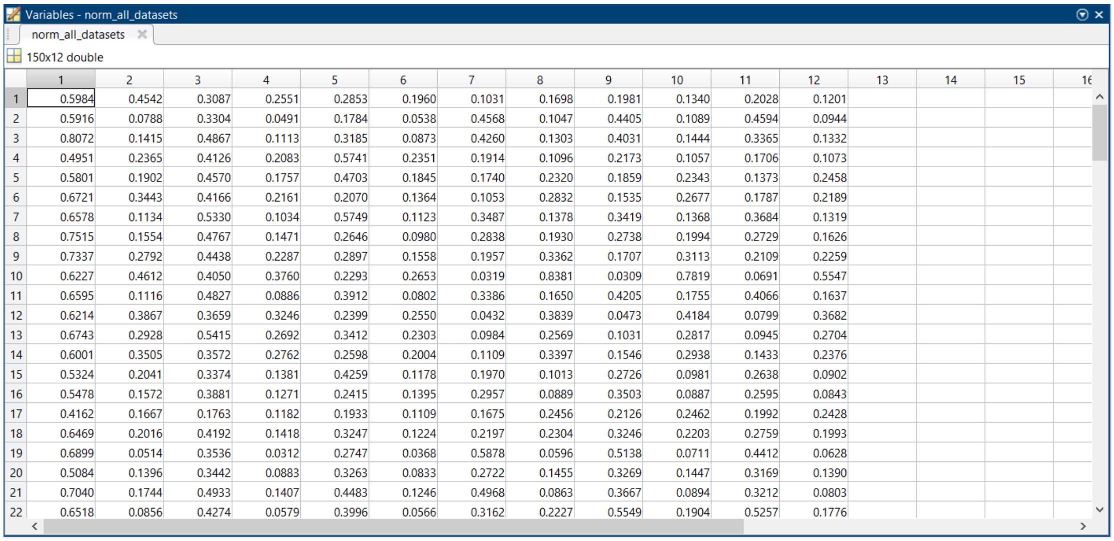
The extracted features are saved as features vectors in a MATLAB file with (.mat) extension for later use in soil classification. Each features vectors contains 12 features: mean, standard deviation, energy and contrast of each layer of a RGB soil image. The file which consists of the extracted features vectors is called the features dataset. Visualization of the features dataset is as shown in Figure 4.6



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	174.6705	53.4365	85.2753	35.3240	62.1247	28.8884	0.0964	0.5208	0.1390	0.4340	0.1684	0.3978				
2	173.1164	17.1218	88.8932	15.9195	48.1660	15.5864	0.2847	0.3682	0.2630	0.3774	0.3238	0.3468				
3	222.7295	23.1836	114.9783	21.7795	66.4551	18.7248	0.2682	0.4282	0.2439	0.4574	0.2493	0.4237				
4	150.9022	32.3744	102.6141	30.9147	99.8116	32.5394	0.1434	0.3796	0.1488	0.3702	0.1489	0.3725				
5	170.4750	27.8978	110.0228	27.8447	86.2612	27.8090	0.1341	0.6664	0.1328	0.6600	0.1287	0.6472				
6	191.6428	42.8116	103.2853	31.6509	51.9057	23.3164	0.0976	0.7865	0.1161	0.7352	0.1538	0.5938				
7	188.3440	20.4670	122.7122	21.0335	99.9187	21.0646	0.2271	0.4459	0.2126	0.4403	0.2687	0.4213				
8	209.9224	24.5323	113.3196	25.1473	59.4223	19.7262	0.1926	0.5750	0.1777	0.5814	0.2109	0.4821				
9	205.8154	36.5133	107.8149	32.8292	62.6940	25.1275	0.1457	0.9107	0.1250	0.8335	0.1733	0.6076				
10	180.2826	54.1228	101.3467	46.7042	54.8136	35.3686	0.0585	2.0864	0.0534	1.8940	0.0874	1.2603				
11	188.7488	20.2927	114.3085	19.6407	75.9410	18.0591	0.2217	0.5094	0.2528	0.5276	0.2918	0.4843				
12	179.9738	46.9071	94.8119	41.8612	56.1893	34.4077	0.0645	1.0223	0.0618	1.0750	0.0940	0.8901				
13	192.1442	37.8304	124.1293	36.6493	69.4203	32.0920	0.0939	0.7249	0.0904	0.7668	0.1029	0.6959				
14	175.0703	43.4095	93.3602	37.3059	58.7938	29.3027	0.1005	0.9188	0.1167	0.7941	0.1324	0.6309				
15	159.4783	29.2468	90.0622	24.3040	80.4665	21.5790	0.1464	0.3604	0.1771	0.3531	0.2053	0.3384				
16	163.0408	24.7024	98.5207	23.2611	56.4062	23.6026	0.1989	0.3313	0.2169	0.3320	0.2027	0.3267				
17	132.7433	25.6283	63.1672	22.4244	50.1129	20.9329	0.1307	0.6983	0.1464	0.6868	0.1662	0.6412				
18	185.8418	29.0033	103.7121	24.6513	67.2697	22.0090	0.1585	0.6627	0.2037	0.6285	0.2126	0.5549				
19	195.7470	14.4716	92.7641	14.2301	60.7374	14.0019	0.3544	0.2626	0.3006	0.2922	0.3128	0.2842				
20	153.9553	23.0073	91.1932	19.6060	67.4753	18.3499	0.1864	0.4638	0.2049	0.4581	0.2375	0.4354				

Figure 4.6 Features Dataset of the System

The range of values of 12 features in a feature vector is different from each other as seen in Figure 4.5. The different in range of features can lead to a challenging classification task for KNN. The solution to this challenging task is to normalize features values by using normalization technique presented in section 3.1.4 of the previous chapter. The normalized features dataset is shown in Figure 4.6. Values of all features are normalized within the range of 0 to 1 as seen in Figure 4.7.



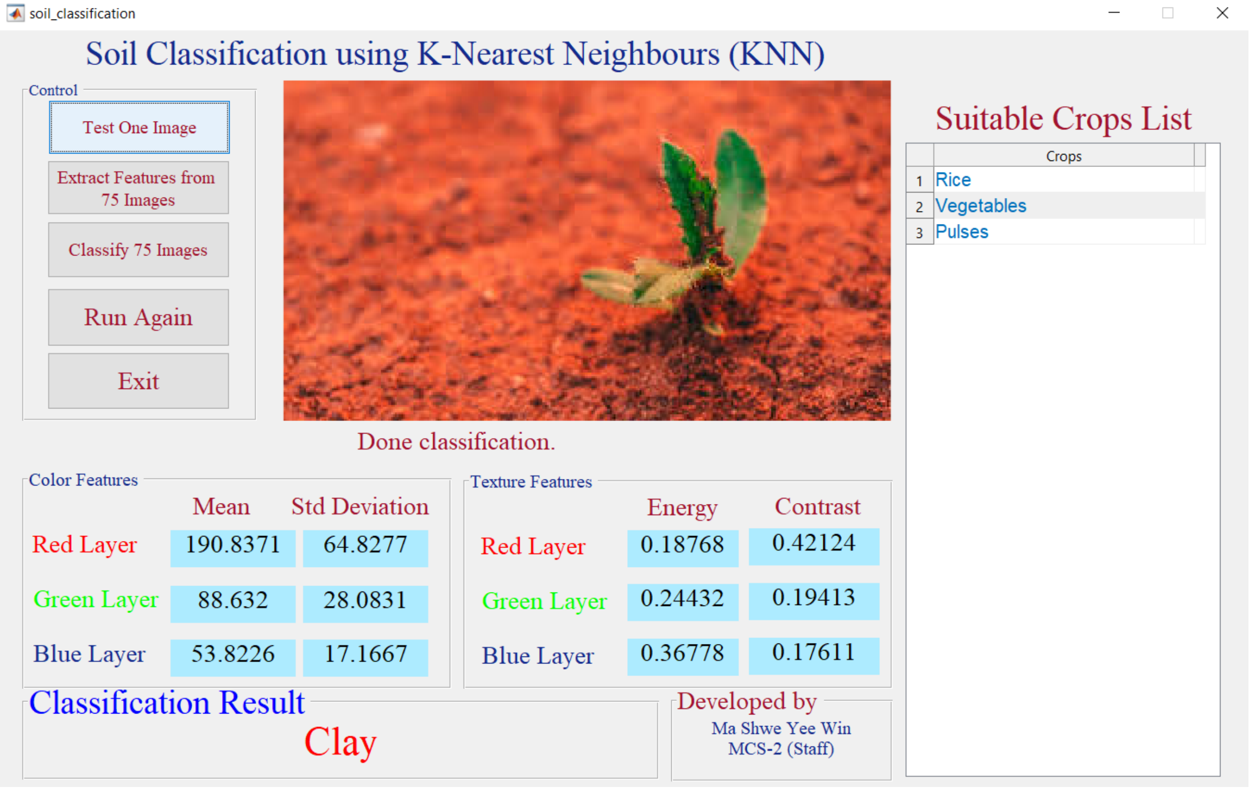
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.5984	0.4542	0.3087	0.2551	0.2853	0.1960	0.1031	0.1698	0.1981	0.1340	0.2028	0.1201				
2	0.5916	0.0788	0.3304	0.0491	0.1784	0.0538	0.4568	0.1047	0.4405	0.1089	0.4594	0.0944				
3	0.8072	0.1415	0.4867	0.1113	0.3185	0.0873	0.4260	0.1303	0.4031	0.1444	0.3365	0.1332				
4	0.4951	0.2365	0.4126	0.2083	0.5741	0.2351	0.1914	0.1096	0.2173	0.1057	0.1706	0.1073				
5	0.5801	0.1902	0.4570	0.1757	0.4703	0.1845	0.1740	0.2320	0.1859	0.2343	0.1373	0.2458				
6	0.6721	0.3443	0.4166	0.2161	0.2070	0.1364	0.1053	0.2832	0.1535	0.2677	0.1787	0.2189				
7	0.6578	0.1134	0.5330	0.1034	0.5749	0.1123	0.3487	0.1378	0.3419	0.1368	0.3684	0.1319				
8	0.7515	0.1554	0.4767	0.1471	0.2646	0.0980	0.2838	0.1930	0.2738	0.1994	0.2729	0.1626				
9	0.7337	0.2792	0.4438	0.2287	0.2897	0.1558	0.1957	0.3362	0.1707	0.3113	0.2109	0.2259				
10	0.6227	0.4612	0.4050	0.3760	0.2293	0.2653	0.0319	0.8381	0.0309	0.7819	0.0691	0.5547				
11	0.6595	0.1116	0.4827	0.0886	0.3912	0.0802	0.3386	0.1650	0.4205	0.1755	0.4066	0.1637				
12	0.6214	0.3867	0.3659	0.3246	0.2399	0.2550	0.0432	0.3839	0.0473	0.4184	0.0799	0.3682				
13	0.6743	0.2928	0.5415	0.2692	0.3412	0.2303	0.0984	0.2569	0.1031	0.2817	0.0945	0.2704				
14	0.6001	0.3505	0.3572	0.2762	0.2598	0.2004	0.1109	0.3397	0.1546	0.2938	0.1433	0.2376				
15	0.5324	0.2041	0.3374	0.1381	0.4259	0.1178	0.1970	0.1013	0.2726	0.0981	0.2638	0.0902				
16	0.5478	0.1572	0.3881	0.1271	0.2415	0.1395	0.2957	0.0889	0.3503	0.0887	0.2595	0.0843				
17	0.4162	0.1667	0.1763	0.1182	0.1933	0.1109	0.1675	0.2456	0.2126	0.2462	0.1992	0.2428				
18	0.6469	0.2016	0.4192	0.1418	0.3247	0.1224	0.2197	0.2304	0.3246	0.2203	0.2759	0.1993				
19	0.6899	0.0514	0.3536	0.0312	0.2747	0.0368	0.5878	0.0596	0.5138	0.0711	0.4412	0.0628				
20	0.5084	0.1396	0.3442	0.0883	0.3263	0.0833	0.2722	0.1455	0.3269	0.1447	0.3169	0.1390				
21	0.7040	0.1744	0.4933	0.1407	0.4483	0.1246	0.4968	0.0863	0.3667	0.0894	0.3212	0.0803				
22	0.6518	0.0856	0.4274	0.0579	0.3996	0.0566	0.3162	0.2227	0.5549	0.1904	0.5257	0.1776				

Figure 4.7 Normalized Features Dataset of the System

After normalizing the values of features, soil classification is ready to be performed.

4.3.2 Implementation of Soil Classification

Two types of soil classification can be performed in the system: testing one image and testing a sequence of images. In Figure 4.8, soil classification on a tested image is performed. The features from the tested image are also needed to extract and they are normalized, and soil classification is performed using normalized features vector of the tested image and features vectors in the normalized features dataset by using KNN. Moreover, a list of agriculture crops that is suitable for the output soil type is also shown in Figure 4.8.



Soil Classification using K-Nearest Neighbours (KNN)

Done classification.

Color Features

	Mean	Std Deviation
Red Layer	190.8371	64.8277
Green Layer	88.632	28.0831
Blue Layer	53.8226	17.1667

Texture Features

	Energy	Contrast
Red Layer	0.18768	0.42124
Green Layer	0.24432	0.19413
Blue Layer	0.36778	0.17611

Suitable Crops List

	Crops
1	Rice
2	Vegetables
3	Pulses

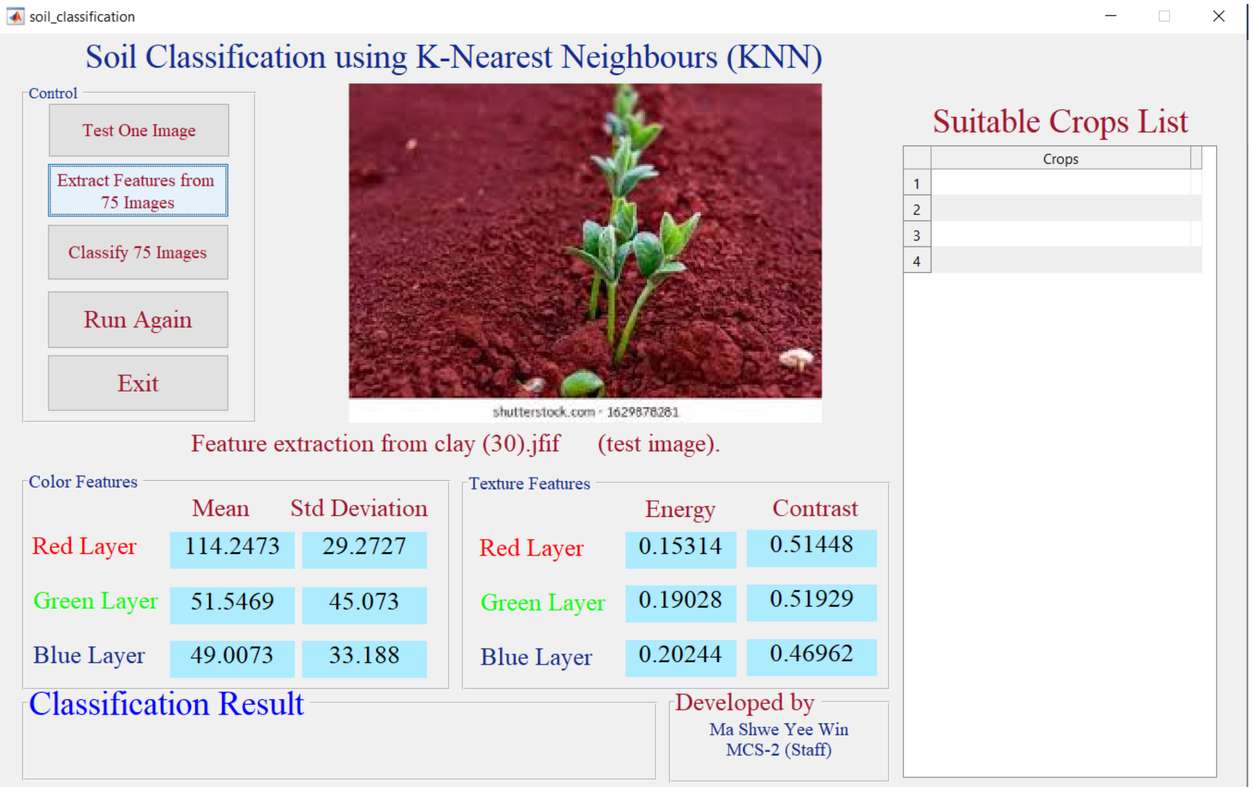
Classification Result

Clay

Developed by
Ma Shwe Yee Win
MCS-2 (Staff)

Figure 4.8 Soil Classification on a Tested Soil Image

As depicted Figure 4.9, before soil classification, features extraction from a sequence of tested images is first performed and they are then normalized. The file which consists of the extracted features vectors from a sequence of tested images is called the normalized tested features dataset.



Feature extraction from clay (30).jfif (test image).

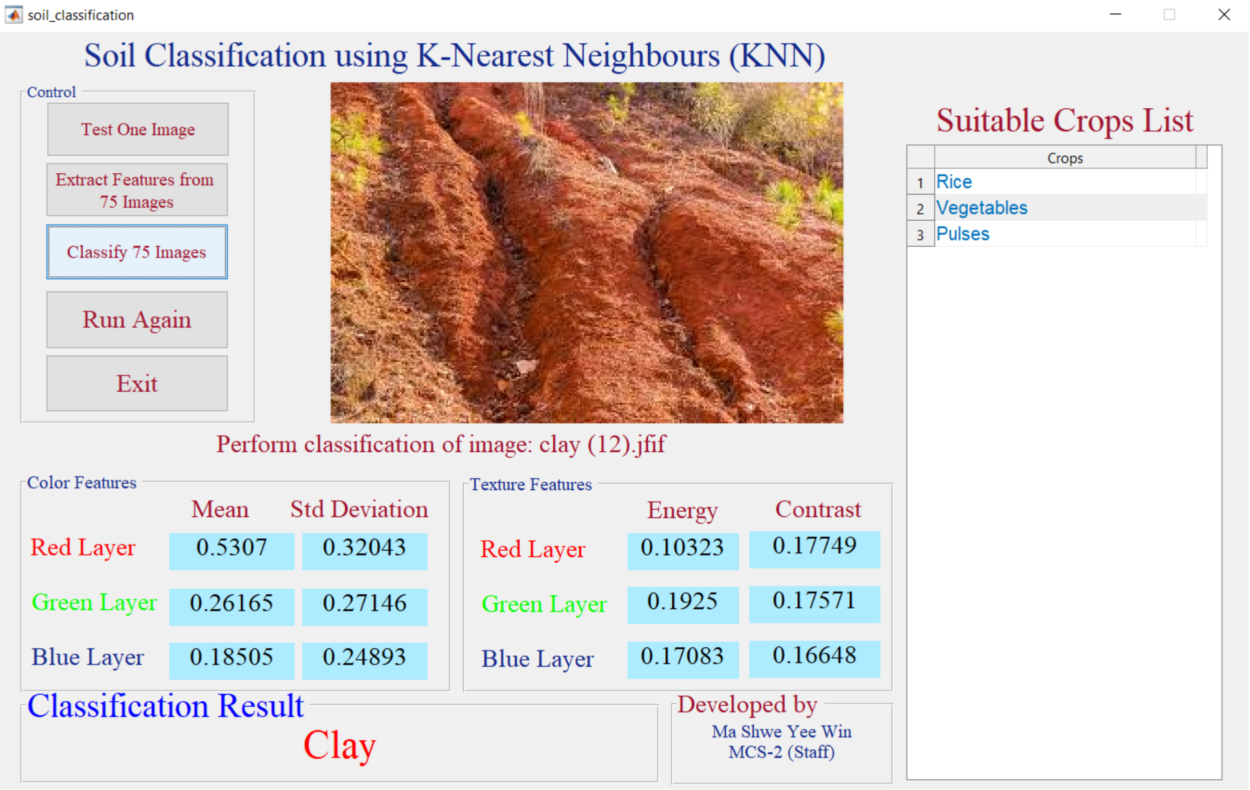
Color Features	Mean	Std Deviation
Red Layer	114.2473	29.2727
Green Layer	51.5469	45.073
Blue Layer	49.0073	33.188

Texture Features	Energy	Contrast
Red Layer	0.15314	0.51448
Green Layer	0.19028	0.51929
Blue Layer	0.20244	0.46962

	Crops
1	
2	
3	
4	

Figure 4.9 Features Extraction from a Sequence of Tested Images

Soil classification on each of the tested soil images in the sequence is performed as shown in Figure 4.10.



Soil Classification using K-Nearest Neighbours (KNN)

Perform classification of image: clay (12).jfif

Color Features

	Mean	Std Deviation
Red Layer	0.5307	0.32043
Green Layer	0.26165	0.27146
Blue Layer	0.18505	0.24893

Texture Features

	Energy	Contrast
Red Layer	0.10323	0.17749
Green Layer	0.1925	0.17571
Blue Layer	0.17083	0.16648

Classification Result

Clay

Developed by

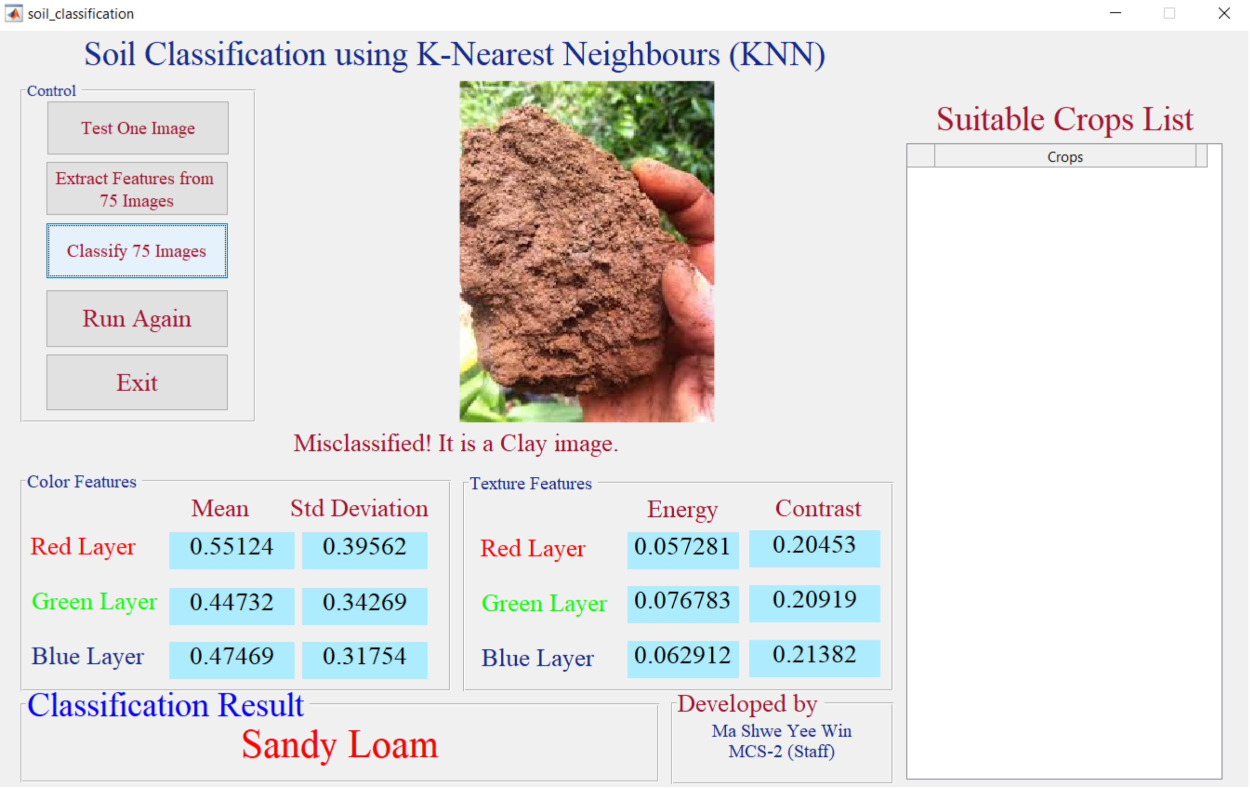
Ma Shwe Yee Win
MCS-2 (Staff)

Suitable Crops List

	Crops
1	Rice
2	Vegetables
3	Pulses

Figure 4.10 Soil Classification on a Sequence of Tested Soil Images

Some misclassification results are also found while testing on each of the tested soil images in the sequence as illustrated in Figure 4.11.



Misclassified! It is a Clay image.

Color Features

	Mean	Std Deviation
Red Layer	0.55124	0.39562
Green Layer	0.44732	0.34269
Blue Layer	0.47469	0.31754

Texture Features

	Energy	Contrast
Red Layer	0.057281	0.20453
Green Layer	0.076783	0.20919
Blue Layer	0.062912	0.21382

Classification Result

Sandy Loam

Figure 4.11 Misclassification Result

4.4 Performance Analysis

To evaluate the system performance, 75 soil images including 30 clay soil images, 22 clay loam soil images and 23 sandy loam soil images are tested in the system. To build the features dataset, 150 soil images including 50 clay soil images, 50 clay loam soil images and 50 sandy loam soil images are used. 30 percent out of 75 soil testing images, 20 percent out of 150 soil images used for features dataset are taken in farms and plantations located in Lashio township by using a main camera of Redmi Note 9 Pro 5G whose specifications are 64 MP, f/1.9, 26mm (wide), 1/1.72", 0.8µm, PDAF; the rest soil images are gathered from websites on the internet such as shutterstock.com, alamy.com, istockphoto.com, etc.

4.4.1 Processing Time for Features Extraction and Soil Classification

The average processing time for features extraction from 150 soil images is about 20 milliseconds. However, the processing time for features extraction can change depending on the size of the soil images. The processing time for features extraction on each soil image is as illustrated in Figure 4.12.

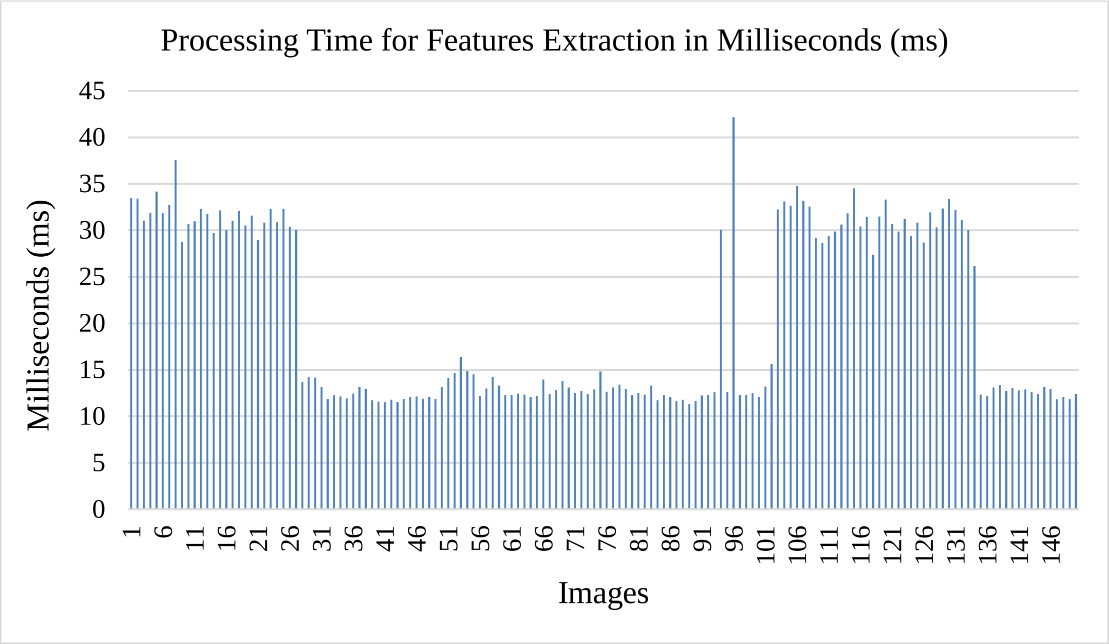


Figure 4.12 Processing Time for Features Extraction

The average processing time for soil classification on 75 soil images is about 0.32 millisecond. However, the processing time for soil classification can also change depending on the size of the features dataset because KNN, a lazy learner, is applied to the system as a classifier. The time complexity of the proposed soil classification can be denoted as $O(nd+k)$ because of KNN, where n is the number of features vectors in the features dataset, d is the dimension of features in a features vector, and k is the number of nearest neighbors. The processing time for soil classification on each soil image is as illustrated in Figure 4.13.

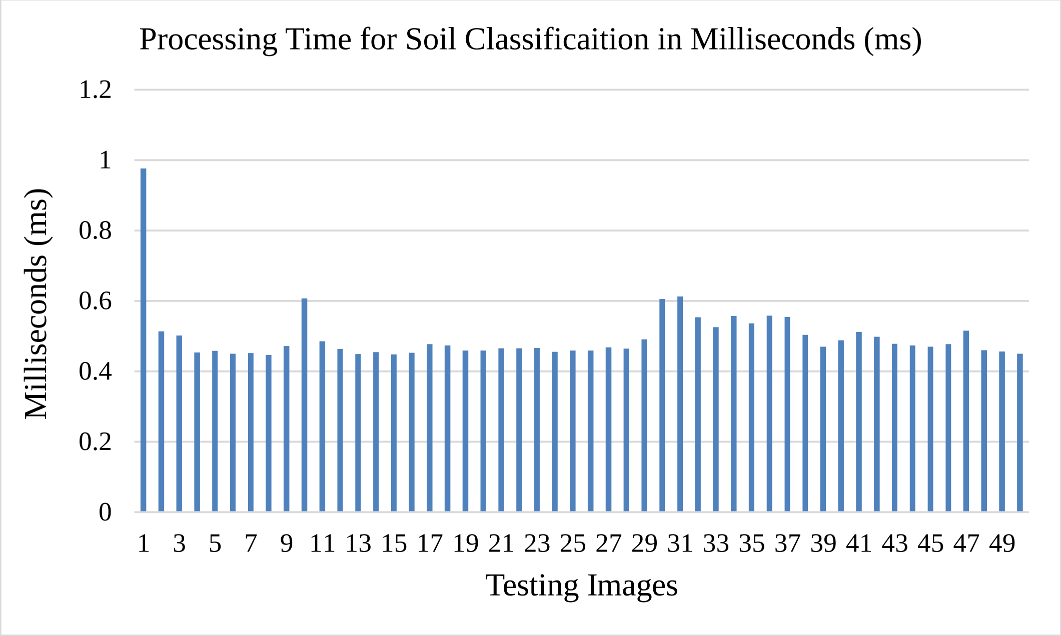


Figure 4.13 Processing Time for Soil Classification

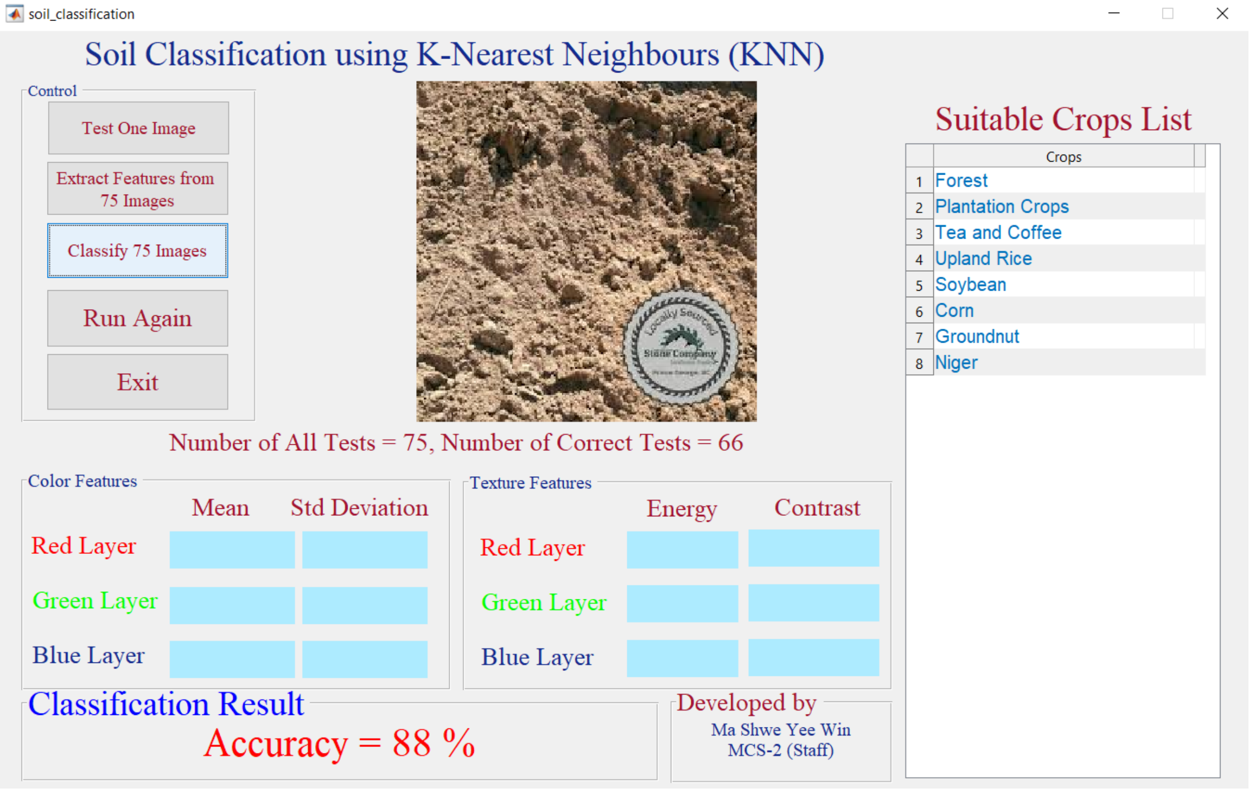
According to the rapid-processing time for features extraction and soil classification, the system can be proved as a fast soil classification system. The processing time for features extraction and soil classification of the system is verified and tested on a computer whose software and hardware specifications are as follows:

- Processor: Intel(R) Core (TM) i5-10200H CPU @ 2.40GHz 2.40 GHz
- Main Memory: 8.00 GB (7.80 GB usable)
- OS: Windows 10 Pro
- Software: MATLAB R2021a
- GPU: NVIDIA GEFORCE GTX with 4GB of memory

The processing time can also vary depending on the software and hardware specifications of a computer.

4.4.2 Soil Classification Accuracy

To evaluate the accuracy of the system, 75 soil images including 30 clay soil images, 22 clay loam soil images and 23 sandy loam soil images are tested in the system. As a result, number of correct tests is 60, and number of misclassified tests is 9 and number of all tests is 75. Therefore, soil classification accuracy is 88 % and the percent error is 12 %. The soil classification accuracy of the system is as shown in Figure 4.14.



	Crops
1	Forest
2	Plantation Crops
3	Tea and Coffee
4	Upland Rice
5	Soybean
6	Corn
7	Groundnut
8	Niger

Number of All Tests = 75, Number of Correct Tests = 66

Accuracy = 88 %

Figure 4.14 Soil Classification Accuracy

The accuracy and percent error are calculated using the following equations:

$$Accuracy = \frac{Number\ of\ Correct\ Tests}{Number\ of\ All\ Tests}\ x\ 100 \qquad (4.1)$$

$$Percent\ Error = \frac{|Number\ of\ Correct\ Tests - Number\ of\ All\ Tests|}{Number\ of\ All\ Tests} x100 \qquad (4.2)$$

In Table 4.1, the accuracy and percent error of the soil classification is shown and they are calculated using Equation 4.1 and 4.2.

Table 4.1 The Accuracy of Soil Classification

Soil Image	Clay	Clay Loam	Sandy Loam	Accuracy (%)	Percent Error (%)
Clay	25	2	3	83.3	16.7
Clay Loam	0	21	1	95.5	4.5
Sandy Loam	1	2	20	87.0	13.0
Accuracy and Percent Error for All Classes				88.0	12.0

The soil classification accuracy and percent error shown in Table 4.1 is also illustrated using confusion matrix as seen in Figure 4.15.

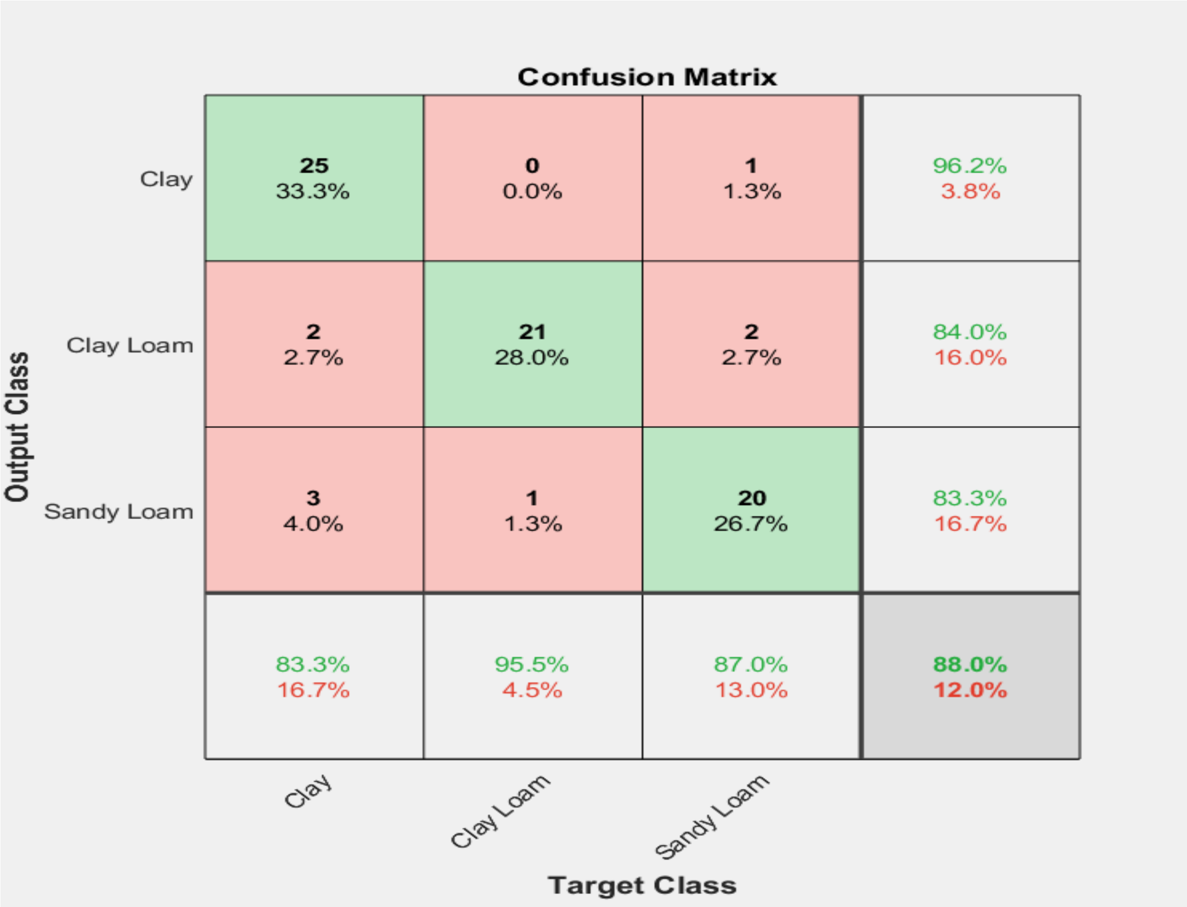


Figure 4.15 Confusion Matrix depicting Soil Classification Accuracy

4.4.3 Comparison with Other Classifiers

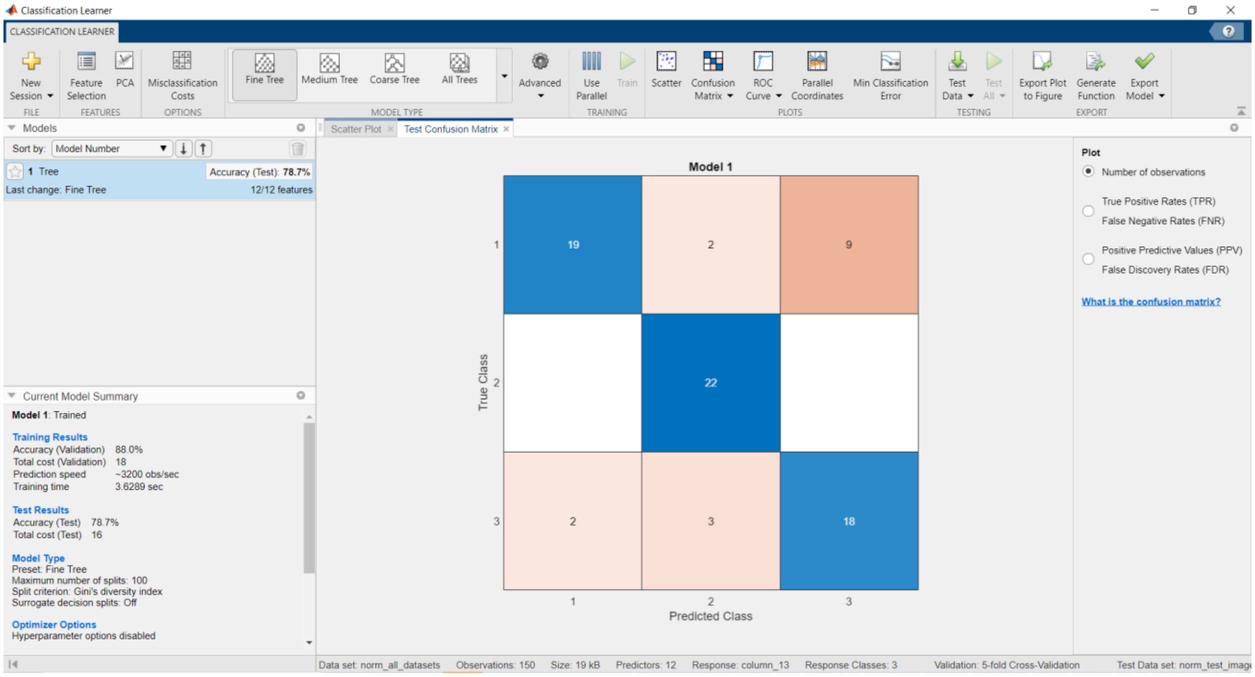
In this section, the soil classification accuracy of the proposed system based on KNN is compared with other two classifiers: Decision (Fine) Trees and Gaussian Naïve Bayes Classifier. The normalized features dataset is first trained using the other two classifiers, and the normalized tested features dataset is then used to evaluate their accuracies. These processes are conducted by using the Classification Learner App of the MATLAB.

The accuracy of the Decision (Fine) Trees classifier is 78.7 % as depicted in Figure 4.16.

The accuracy of the Kernel Naïve Bayes classifier is 76 % as depicted in Figure 4.17.

The accuracy comparison among the proposed system and the other two classifiers is depicted in Figure 4.18.

According to accuracy value, the proposed KNN-based soil classification is assumed to be more reliable and efficient than using the other two classification methods.



1 Tree Accuracy (Test): 78.7%
Last change: Fine Tree 12/12 features

Model 1

True Class \ Predicted Class	1	2	3
1	19	2	9
2		22	
3	2	3	18

Current Model Summary

Model 1: Trained

Training Results
Accuracy (Validation) 88.0%
Total cost (Validation) 18
Prediction speed ~3200 obs/sec
Training time 3.6289 sec

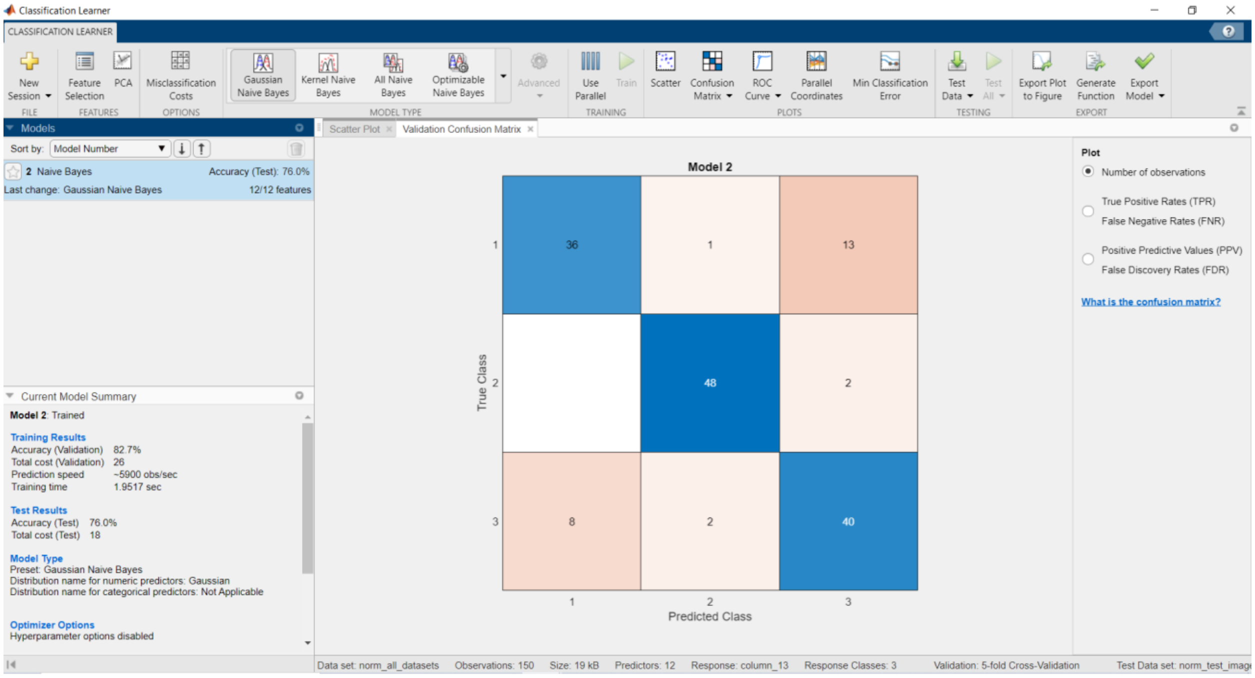
Test Results
Accuracy (Test) 78.7%
Total cost (Test) 16

Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Optimizer Options
Hyperparameter options disabled

Data set: norm_all_datasets Observations: 150 Size: 19 kB Predictors: 12 Response: column_13 Response Classes: 3 Validation: 5-fold Cross-Validation Test Data set: norm_test_image

Figure 4.16 Accuracy of the Decision (Fine) Trees Classifier



Sort by: Model Number

2 Naive Bayes Accuracy (Test): 76.0%
Last change: Gaussian Naive Bayes 12/12 features

Current Model Summary

Model 2: Trained

Training Results
Accuracy (Validation) 82.7%
Total cost (Validation) 26
Prediction speed ~5900 obs/sec
Training time 1.9517 sec

Test Results
Accuracy (Test) 76.0%
Total cost (Test) 18

Model Type
Preset: Gaussian Naive Bayes
Distribution name for numeric predictors: Gaussian
Distribution name for categorical predictors: Not Applicable

Optimizer Options
Hyperparameter options disabled

Model 2

True Class \ Predicted Class	1	2	3
1	36	1	13
2		48	2
3	8	2	40

● Number of observations
○ True Positive Rates (TPR) False Negative Rates (FNR)
○ Positive Predictive Values (PPV) False Discovery Rates (FDR)

Data set: norm_all_datasets Observations: 150 Size: 19 kB Predictors: 12 Response: column_13 Response Classes: 3 Validation: 5-fold Cross-Validation Test Data set: norm_test_imag

Figure 4.17 Accuracy of the Gaussian Naïve Bayes Classifier

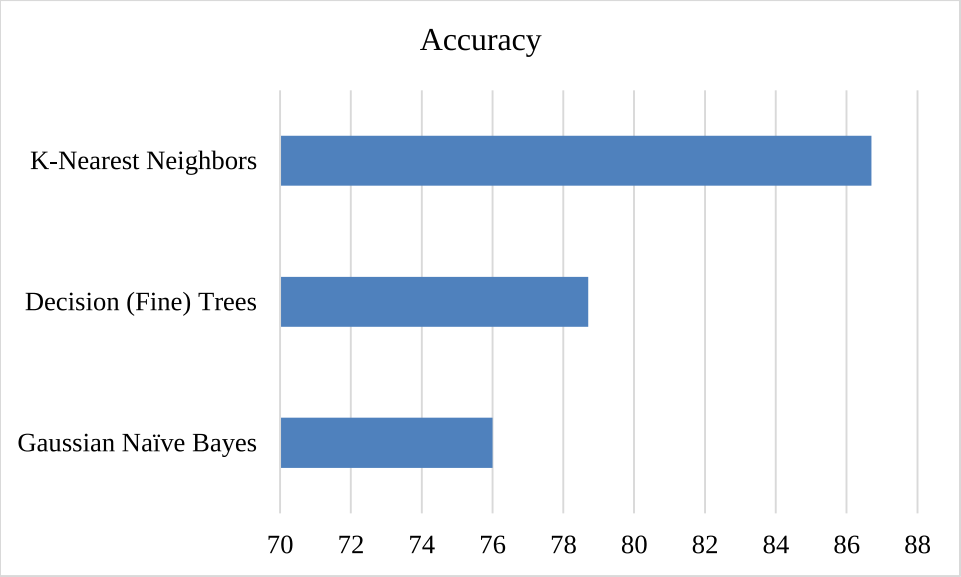


Figure 4.18 Accuracy Comparison with Other Classifiers

4.5 Summary

The overview design of the proposed soil classification system is discussed in this chapter. The proposed soil classification system is developed and implemented as the soil classification system for agriculture crops for farms and plantations in specific areas. The main point of the proposed system is color and texture features extraction and soil classification based on these extracted features using KNN as a classifier. The architecture of the proposed system comprises pre-processing stage, features extraction and features dataset building stage, and soil classification stage. To evaluate the accuracy of the system, 75 soil images including 30 clay soil images, 22 clay loam soil images and 23 sandy loam soil images are tested in the system. According to the accuracy comparison among the proposed system and the other two classifiers: Decision (Fine) Trees and Kernel Naïve Bayes, the proposed KNN-based soil classification is assumed to be more reliable and efficient than using the other two classification methods.

CHAPTER 5

CONCLUSION AND FURTHER EXTENSIONS

In this thesis, a soil classification system for agriculture crops using KNN is implemented and proposed. In the proposed soil classification method, color and texture features extraction is first performed, and soil classification is then achieved by using KNN as a classifier. In this chapter, the main points of the research work are concluded. Moreover, the advantages , limitations of the system and future work are also discussed in this chapter.

5.1 Conclusion

Traditional soil classification methods in laboratory or in-situ are expensive, experts- and labor-intensive, and time-consuming. With the rapid advent of computer technology, many researchers in the field of artificial intelligence (AI) have been trying to automate soil classification in order to reduce human efforts as much as possible. This thesis has also proposed a method and implemented a system that can efficiently and automatically classify soil types using trending computer technologies such as Machine Learning and image processing. In addition, the system can be provided farmers a list of agriculture crops that can easily grow in their farms and plantations.

5.2 Advantages and Limitations of the System

The proposed soil classification system is simple but highly user-friendly and useful because all of the user needs is just to take a soil image of his farm or plantation. The system then can provide him soil type of his farm or plantation, and moreover a list of agriculture crops that can easily grow there.

In the proposed soil classification method, color and texture features extraction from soil images and their classification using KNN achieves a fast and effective soil classification system. The proposed soil classification system can reach up to 88% accuracy in soil classification.

Unfortunately, the proposed soil classification system can classify only 3 types of soil. If more soil types are needed to classified, more soil images must be collected; more features from soil images must be extracted; and more sophisticated classifiers such as deep neural networks (DNN) must be applied to the system.

5.3 Further Extensions

The proposed soil classification system is tested by using only 3 types of soil: clay, clay loam and sandy loam. The features dataset can be extended with other soil type images such as silty clay and clay with gravel. Moreover, in features extraction process, more image features such as color histogram, color moment, correlation and homogeneity are needed to be extracted. Furthermore, in soil classification process, instead of using KNN as a classifier, more sophisticated classifiers in machine learning such as deep neural networks (DNNs) and support vector machines (SVMs) must be applied to soil classification. Achieving a plausible result in the soil classification is a motivation to do further research works such as soil classification using convolutional neural networks (CNNs).

AUTHOR'S PUBLICATIONS

[1] Shwe Yee Win, Thin Lai Lai Thein"Soil Classification for Agriculture Crops using K-Nearest Neighbors (KNN)", The National Journal of Parallel and Soft Computing(NJPSC 2022),2022.

REFERENCE

[2] Ahmed J. Afifi, Wesam M. Ashour, “Image Retrieval Based on Content using Color Feature”, International Scholarly Research Network, ISRN Computer Graphics, Volume 2012, Article ID 248285, January 2012.

[3] Chu Thinzar, “Content-based Image Classification and Retrieval using Support Vector Machine”, M.C.Sc. Thesis, Faculty of Computer Science, The University of Computer Studies, Yangon, March 2019.

[4] Danashree S. Kalel, Pooja M. Pisal, Ramdas P. Bagawade, “Color, Shape and Texture Feature Extraction for Content-based Image Retrieval System: A Study”, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), ISSN (Online) 2278-1021, ISSN (Print) 2319 5940, Volume 5, Issue 4, April 2016.

[5] J. C. Kavitha and A. Suruliandi, “Texture and color feature extraction for classification of melanoma using SVM”, 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), IEEE, 2016, pp. 1-6, DOI: 10.1109/ICCTIDE.2016.7725347.

[6] Kristine A. Thomas, “Image Processing as applied to Medical Diagnosis”, M.Sc. Thesis, Department of Computer and Information Science, The Graduate School of University of Oregon, USA, June 2010.

[7] Laleh Armi, Shervan Fekri-Ershad, “Texture Image Analysis and Texture Classification Methods- A Review”, International Online Journal of Image Processing and Pattern Recognition, Volume 2, No. 1, pp. 1-29, 2019.

[8] Land Use Division, “Soil Types and Characteristics of Myanmar”, Ministry of Agriculture and Irrigation, Myanmar.

[9] Milan Sonka, Vaclav Hlavac, Roger Boyle, “Image Processing, Analysis, and Machine Vision”, 4th Edition, Cengage Learning, USA, 2015.

[10] Miroslav Benco, Robert Hudec, Patrik Kamencay, Martina Zachariasova, Slavomir Matsuka, “An Advanced Approach to Extraction of Color Texture Features based on GLCM”, 2014 International Journal of Advanced Robotic Systems, 2014, DOI: 10.5772/58692.

[11] Nay Zar Aung Dr., “Image Processing and Analysis Using MATLAB”, Myanmar, January 2018.

[12] P. Mohanaiah, P. Sathyanarayana, L. GuruKumar, “Image Texture Feature Extraction using GLCM Approach”, International Journal of Scientific and Research Publications, Volume 3, Issue 5, ISSN 2250-3153, May 2013.

[13] Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 4th Edition, Pearson, New York, USA, 2018.

[14] R. Venkata Ramana Chary, D. Rajya Lakshmi, K. V. N. Sunitha, “Feature Extraction Methods for Color Image Similarity”, Advanced Computing: International Journal (ACIJ), Volume 3, No. 2, March 2012, DOI: 10.5121/acij.2012.3215.

[15] S. A. Z. Rahman, K. Chandra Mitra and S. M. Mohidul Islam, "Soil Classification Using Machine Learning Methods and Crop Suggestion Based on Soil Series," 2018 21st International Conference of Computer and Information Technology (ICCIT), IEEE, 2018, pp. 1-4, DOI: 10.1109/ICCITECHN.2018.8631943.

[16] S. Mutalib, S. Abdul-Rahman, “Soil Classification: An Application of Self Organizing Map and k-means”, 2010 10th International Conference on Intelligent Systems Design and Applications, Malaysia, 2010.

[17] Srunitha. K, S. Padmavathi, “Performance of SVM Classifier for Image based Soil Classification”, International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016, IEEE, ISBN: 978-1-5090-4620-1.

[18] Sundos Abdulameer Alazawi, Narjis Mezaal Shati, Amel H. Abbas, “Texture features extraction based on GLCM for face retrieval system”, Periodicals of Engineering and Natural Sciences, Volume 7, No. 3, pp. 1459-1467, October 2019, ISSN 2303-4521.

[19] T. M. Cover, P. E. Heart, “Nearest Neighbor Pattern Classification”, IEEE Transactions on Information Theory, Volume 13, No. 1, January 1967.

[20] T. M. Mitchell, “Machine Learning”, 1st Edition, McGraw-Hill Science/Engineering/Math, March 1997, ISBN: 0070428077.

[21] Wamidh K. Mutlag, Shaker K. Ali, Zahoor M. Aydam, Bahaa H.Taher, “Feature Extraction Methods: A Review”, Journal of Physics: Conference Series, 2020, DOI:10.1088/1742-6596/1591/1/012028.