People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of the Requirements for the Degree of

MASTER

In **Telecommunication**

Option: **Telecommunications**

Title:

Building Detection From High Resolution Remote Sensing Imagery

Presented by:

- **BENTAALA Ali**
- **BOULEBNANE Lokman**

Supervisor:

Mr.DAAMOUCHE

Registration Number:……/2020

Dedication

I dedicate this work

To my mother, my father and

To all my family

To all people who have helped me to obtain my modest knowledge.

To all good people that I met in our institute and shared the last five years together.

To all my friends

ALI

Dedication

I dedicate this modest work

To my beloved mother, my dear father

To my brother, sisters.

To the people who made me who I am today.

To all the good people that shared five years of their lives

With me.

To all my friends.

LOKMAN

Acknowledgement

First and foremost, we are thankful to God, the most merciful, the most beneficent at times when it seemed impossible to go on.

We wish to express our sincere gratitude to our supervisor, Abdelhamid DAAMOUCHE for his enthusiasm, patience, helpful information, practical advice and unceasing ideas that have helped us tremendously at all times in our research and writing of this thesis. Without his support and guidance, this project would not have been possible.

We are infinitely grateful to our family members, particularly our parents for patience and encouragement.

Last but not least, we thank so much all people that contributed to the realization of the present work especially Rasha BELFEDHAL.

Table of content

CHAPTER 2: Feature Extraction

CHAPTER 3: Classification Algorithms in Machine Learning

List of Tables

List of Figures

List of abbreviations

SVM: Support Victor Machine.

GLCM: Gray Level Co-Occurrence Matrix

VHR: Very High Resolution

HSI: Hyperspectral Imaging

MSI: Multispectral Imaging

MM: Mathematical Morphology.

MO: Morphological Operator.

MF: Morphological Filter.

SE: Structuring Element.

WS: Window Size

KNN: K-Nearest Neighbors

NN: Neural Network

Abstract

Building detection is an important task in very high-resolution remote sensing image analysis. In recent years, availability of very high-resolution images raised new challenges to building detection algorithms.

In this report, we use a supervised method to detect buildings from remotely sensed images using spectral-spatial features. The morphological operations (MO), gray level co-occurrence matrix (GLCM) and Variogram techniques are used to extract the spatial features.
We concatenated spatial features and spectral features, and then we fed the Support Vector Machines (SVM) classifier with the resulting vector of features. We classified the image data into two classes **(Building** and **Non-Building)** using different combinations of features. The simulation results obtained on three different images showed that our approach achieved an acceptable performance in terms of accuracy.

Introduction

With the development of imaging technology, many airborne or satellite sensors have provided very high resolution (VHR) imagery. These images have been used in a wide range of remote sensing applications, including agriculture, land planning, defense, and surveillance.

Traditional classification algorithms have exposed weaknesses necessitating further research in the field of image classification, to classify the remote sensing imagery we need an efficient classifier, we used in our work support vector machines (SVM), which is widely used for classifying remote sensing data; also it makes the optimization process very rapidly. Due to its higher performance over other existing classifiers, SVM become a benchmark in the field of pattern recognition.

The feature extraction process is a necessary phase in image classification. In our approach, different features have been extracted using morphological operators, gray level co-occurrence matrix and Variograms.

Building detection from remote sensing images is one of the key pieces of cadastral information related to population and cities, and are fundamental to urban planning [1]. Researchers focused on building detection using automated image processing and computer vision techniques. The main problems that are raised in our work are as follows: buildings have diverse characteristics and their appearance (illumination, viewing angles, etc.) is uncontrolled.

In this study, we used a binary SVM classification approach for the detection of buildings. In addition to original bands of the image, the morphological operators (MO), the gray level co-occurrence matrix (GLCM) and variogram were also included in the classification process. With the use of original and additional bands, twenty different feature combinations were generated using the Beijing dataset, and then a binary SVM classification was performed. The proposed approach was implemented in three **Sub-Areas** which were selected from Beijing, the capital city of China.

The objectives of the study are as follows: (i) to detect buildings using the binary SVM classification technique (ii) to explore the effects of the additional features MO, GLCM, and VARIOGRAM on building detection (iii) to determine the best feature combination that achieves the highest classification accuracy.

This report is organized in four chapters, chapter 1, which is an introductory chapter, deals with the fundamentals of the remote sensing data and some generalities about classification in remote sensing. Chapter 2 explains the tools used in this work for features extraction MO, GLCM, and VARIOGRAM. Chapter 3 highlights generalities about classification algorithms in machine learning. Chapter 4 presents the experimental results of the work and discusses the obtained result.

Chapter 1

Remote Sensing

1.1. Introduction

This chapter is a general overview of the remote sensing field and classification techniques of images in remote sensing. Also, the different types of images and image feature extraction.

1.2. What is remote sensing?

The term "remote sensing," first used in the United States in the 1950s by Ms. **Evelyn Pruitt** of the U.S. Office of Naval Research [2]. *Remote sensing* is the process of detecting and monitoring the physical characteristics of an area by measuring its reflected and emitted radiation at a distance objects or materials as shown in figure 1.1.

Remote sensing technology are divided into two types active and passive remote sensing. Active sensors emit energy in order to scan objects and areas whereupon a sensor then detects and measures the radiation that is reflected or backscattered from the target. And passive sensors accumulate radiation that is emitted or mirrored by the object or surrounding areas. Reflected sunlight is the most common supply of radiation measured by way of passive sensors.

Remote sensing data provides essential information that helps in monitoring various applications such as mapping land use and cover, agriculture, soils mapping, forestry, city planning, archeological investigations, military observation, land cover changes, deforestation, vegetation dynamics, water quality dynamics, urban growth, etc.

Remote sensing is now one of the most useful sciences due to the fact that it is important in gathering, interpreting and processing information from inaccessible areas.

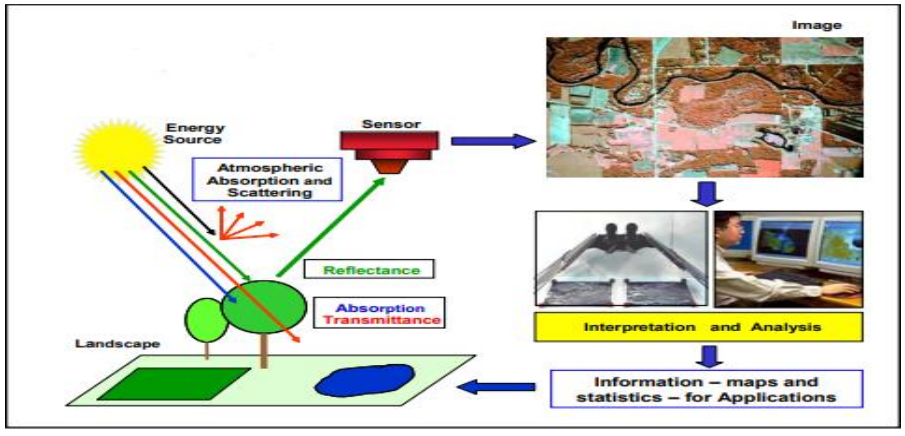


Figure1.1 Remote Sensing Process [2].

1.3. Multispectral and hyperspectral images

In remote sensing, the terms Hyperspectral imaging (HSI) and Multispectral imaging (MSI) are often conflated [3]. But Hyperspectral imaging systems acquire images in over one hundred contiguous spectral bands. While multispectral imagery is useful to discriminate land surface features and landscape patterns.

Multispectral and hyperspectral imaging provide the ability to see as human (red, green and blue), and what we cannot see like infrared and ultraviolet. In fact, we can see more than this reflected electromagnetic radiation to the sensor.

1.3.1. Key differences

The distinction between the MSI and the HSI system is that the HSI system uses continuous wavelengths in data collection while the MSI system concentrates on multiple wave bands that are pre-selected based on the application. Common RGB sensors help illustrate the differences between HSI and MSI systems. In an RGB sensor, a Bayer pattern — consisting of red, green, and blue filters — is layered over the pixels. The filters allow wavelengths from specific color bands to be absorbed by the pixels while the rest of the light is attenuated [2]. These bandpass filters have transmission bands in the range of 100 to 150 nm and have slight spectral overlap as shown on Figure 1.2.

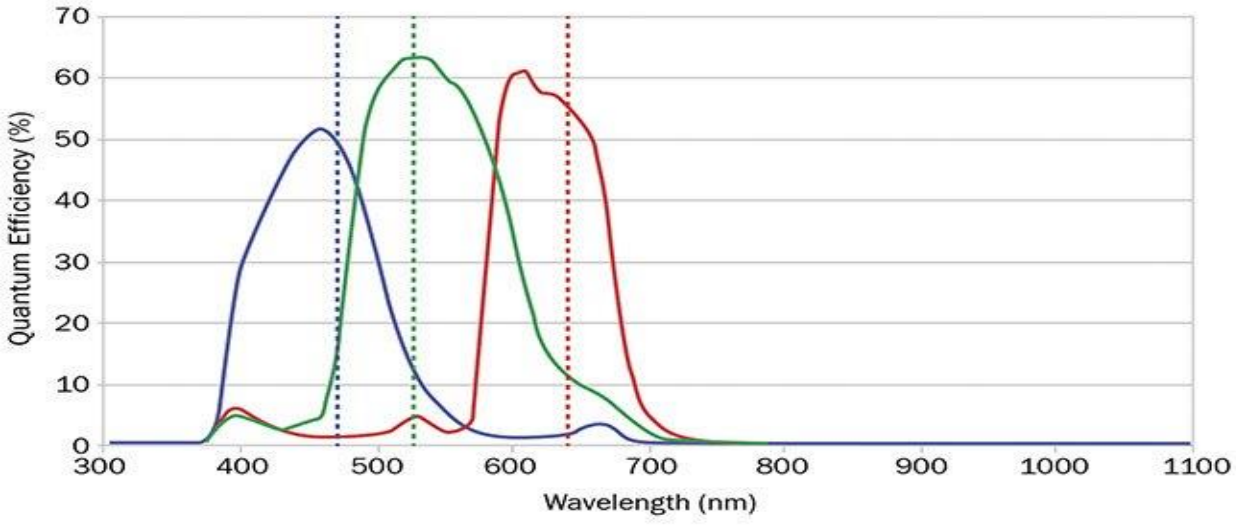


Figure 1.2 curves for an RGB camera showing the overlap between red, green, and blue [3].

The two technologies each present advantage, depending on the task. HSI is best suited for applications that are sensitive to subtle differences in signal along a continuous spectrum. However, some systems require that significant portions of the electromagnetic spectrum be blocked and that light be only selectively captured Figure 1.3.

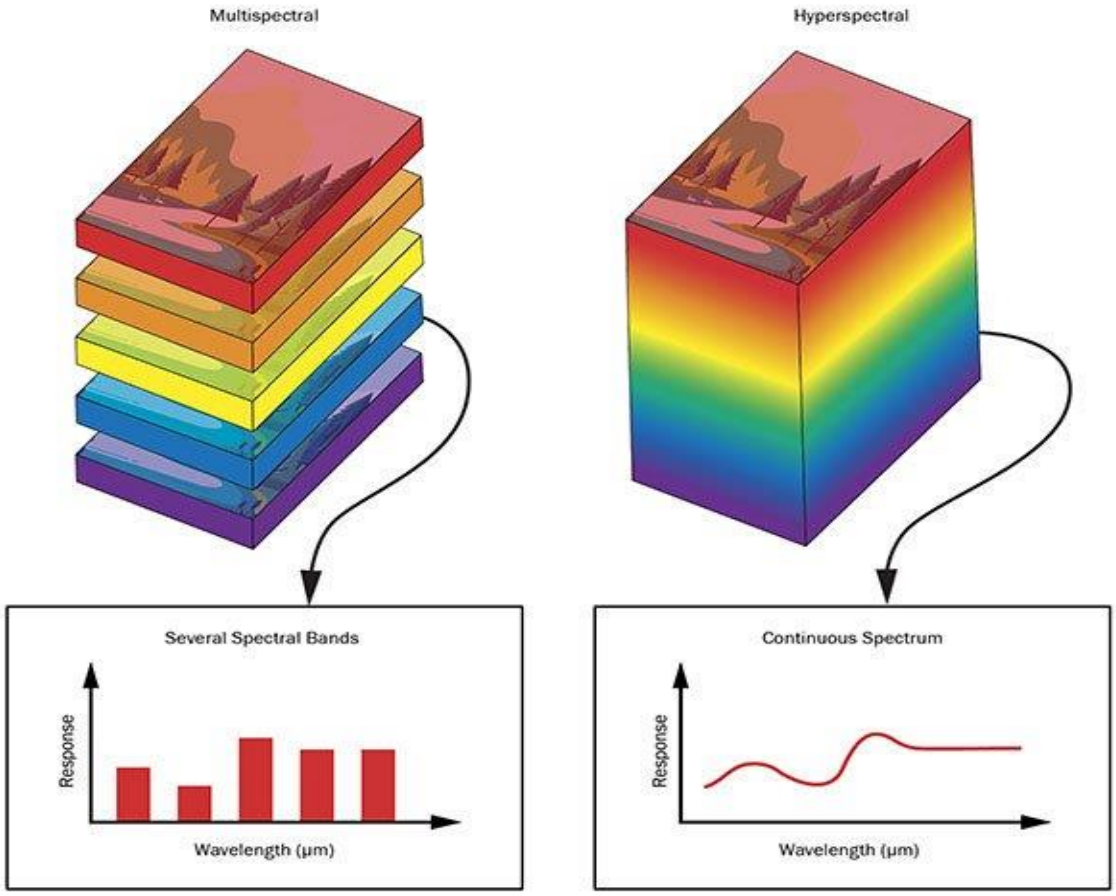


Figure 1.3 Multi- and hyperspectral stacks of images. Comparison of the image stacks in MSI, in which several images are taken across various discrete wavelength regions, and HSI, in which images are taken over a larger continuous range of wavelengths [3].

1.4. Image classification in remote sensing

The problem of costly and difficult remotely sensed products is a major challenge in our day, so it requires us to find other ways to improve the resolution of images, the image processing technique is one of the methods that can help to achieve that goal.

In Multispectral Image Classification, we combine the techniques of image processing and the classification method, typically grouping identical pixels into groups is the process of image

classification in remote sensing. This is demonstrated schematically in Figure 1.4, which shows a 2D scatter plot of image pixels into classes. The major steps of image classification may include determination of a suitable classification system, selection of training samples, image preprocessing, feature extraction, selection of suitable classification approaches, and accuracy assessment [4].

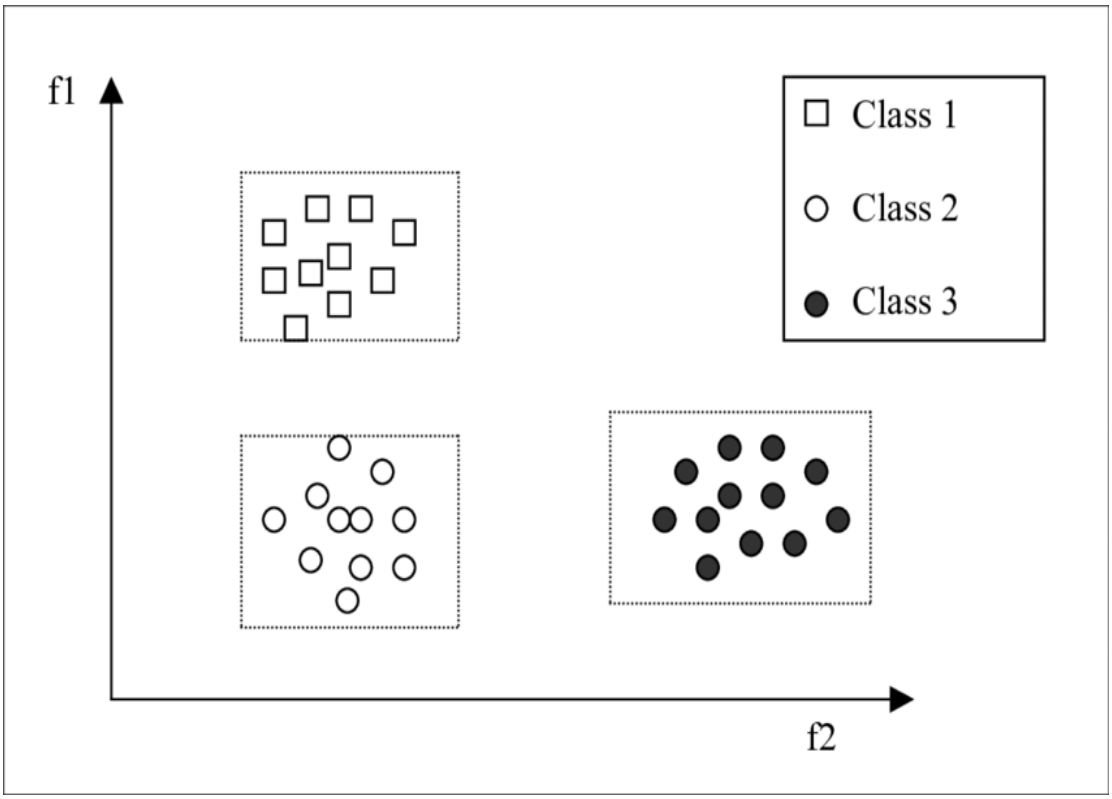


Figure 1.4 Image pixels grouped into classes in a 2D scatter plot.

Image classification is an important part of the remote sensing data mining, at present, it is not possible to identify which classifier is ideally suited to all cases, since the features of each picture and the conditions of each study differ too greatly. The understanding of the data and the Experimentation They are the cornerstones for getting the best classification.

1.5. Classification Techniques

A group of image classification techniques has been developed to classify pixels automatically with similar multispectral reflectance values into clusters that, ideally, correspond to usable land use and land cover categories [5]. There are many types of learning associated with machine learning, but the two widely used are Supervised and Unsupervised Learning.

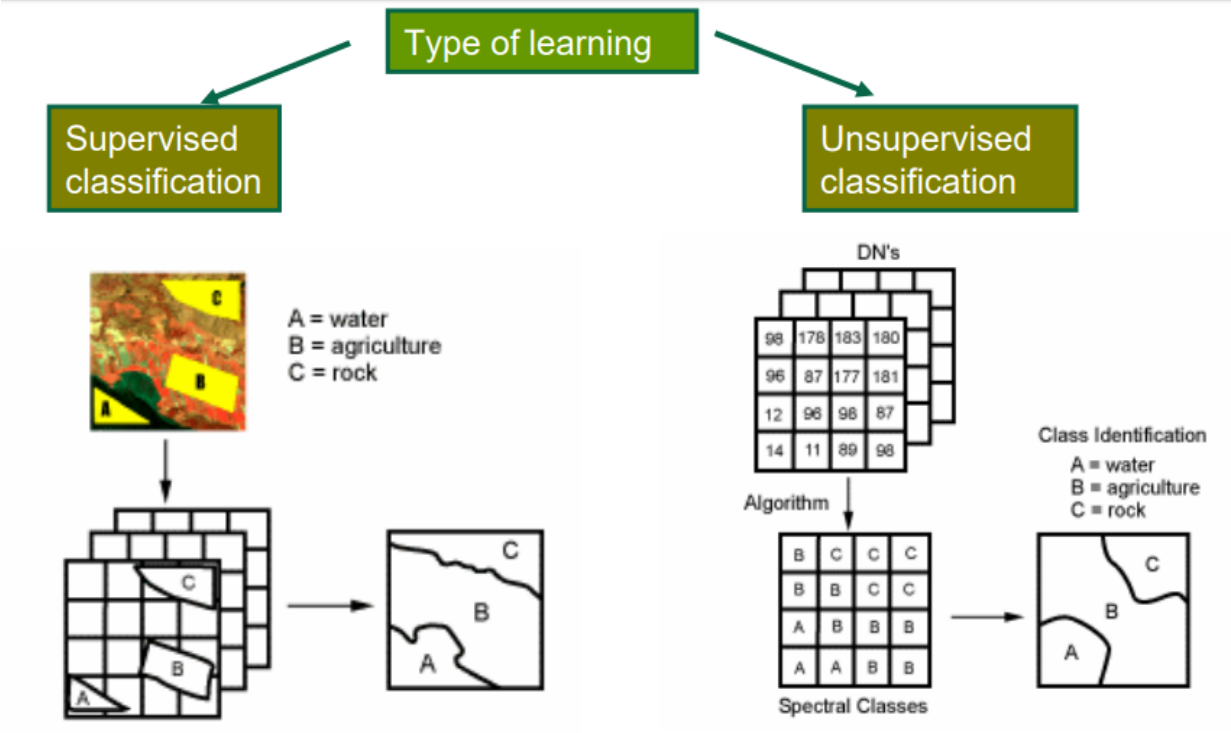


Figure 1.5: Image Classification Techniques [5].

1.5.1. Supervised classification

Supervised learning is the most practical machine learning in our days. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output [6].

$$Y = F(X). \quad (1.1)$$

The principle of supervised learning is to predict output variable Y based on new input data X, the algorithm iteratively predicts training and corrects this prediction until the algorithm reaches an appropriate level of performance.

1.5.2. Unsupervised classification

The difference between supervised learning and unsupervised learning is that the second does not have output Y data corresponding to input X data. The goal of unsupervised learning is to model the underlying structure or distribution of data in order to learn more about the data [6].

Unsupervised learning algorithms are left to their own devices to discover and present a fascinating structure of the data.

1.6. Feature Representation

We can describe features as a piece of information which is relevant for solving the computational task related to a certain application. This is the same sense as a feature in machine learning and pattern recognition generally, though image processing has a very sophisticated collection of features. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image.

The development of feature extraction methods has been one of the most important problems in the field of pattern analysis and has been studied extensively. Feature extraction methods can be both unsupervised and supervised. Several feature extractions approaches have been proposed and applied successfully in classification of remote sensing data.

1.7. Conclusion

The first chapter is an overview about image classification in remote sensing field and the different type of classification. Supervised technique has been successfully applied for a variety of machine vision applications, image classification, object recognition and object detection, we have used the supervised technique in our work for building detection algorithms.

Chapter 2

Features Extraction

2.1. Introduction to Features Extraction

Feature extraction describes the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. In pattern recognition and in image processing, feature extraction is a special form of dimensionality reduction, transforming the input data into the set of features is called feature extraction.

In feature extraction there is more than one technique to select the extracted features, in our work we use 3 different techniques to obtain the extracted features and those techniques are.

- Morphological Operators
- Gray-Level Co-occurrence Matrices (GLCM)
- Variogram

2.2. Morphological Operators

2.2.1. Introduction

Morphological operation is a set of image processing algorithms that acts on image pixels using pre-defined kernels. These kernels, known as structuring elements, a morphological operation is used to remove imperfections on the image.

2.2.2. Basic Set Operations

Morphological operations use a small shape or template known as structuring elements. This structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels [7]. Some operations test whether the element "fits" within the neighborhood, or its "hits" the neighborhood.

The most common morphological operations are **dilation, erosion, opening and closing**. They are defined in terms of the interaction of the original image **A** to be processed [8]. and the structuring element **B**. Next, the 4 basic operators are defined

❖ **Dilation**

Dilation is an operation used to grow or thicken objects in binary image.the dilation of a binary image **A** by a structuring element **B** is defined as:

$$\boldsymbol{A \oplus B = \{z: \left(\widehat{B}\right)_z \cap \mathbf{A} \neq \emptyset\}.} \quad \textbf{(2.1)}$$

This equation is based on obtaining the reflection of **B** about its origin and translating (shifting) this reflection by **z.** then the dilation of **A** by **B** is the set of all structuring element origin locations where the reflected and translated **B** overlaps with **A** at least one element.

❖ **Erosion**

Erosion is used to shrink or thin objects in binary images. The erosion of a binary image **A** by a structuring element **B** is defined as:

$$\boldsymbol{A \ominus B = \{z : (B)z \ \cap \ A^C \neq \ \emptyset\}} \quad \textbf{(2.2)}$$

The erosion of **A** by **B** is the set of all structuring element origin locations where the translated **B** does not overlap with the background of **A**.

❖ **Opening**

The opening operation erodes an image and then dilates the eroded image using the same structuring element for both operations.

$$\boldsymbol{A \circ B = \{(A \ominus B) \oplus B\}} \quad \textbf{(2.3)}$$

The opening operation is used to remove regions of an object that cannot contain the structuring element, smooth object contours, and breaks thin connection.

❖ **Closing**

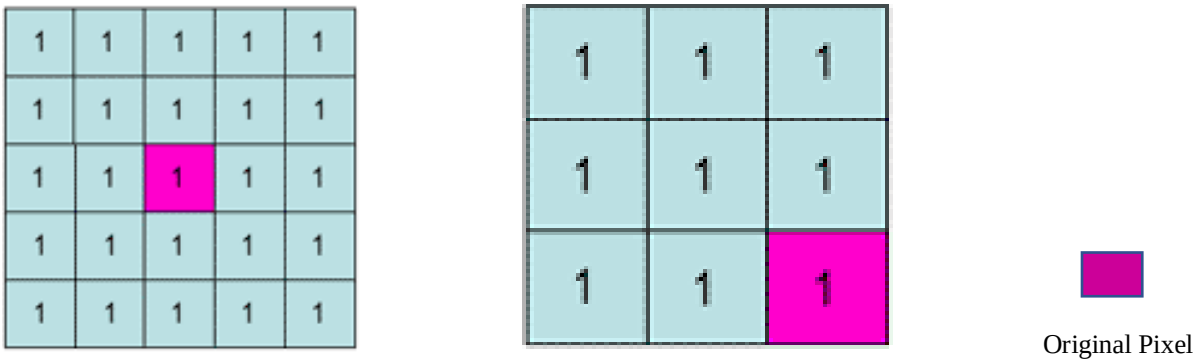
The closing operation dilates an image and then erodes the dilated image using the same structuring element for both operations. The closing operation fills holes that are smaller than the structuring element.

$$\boldsymbol{A * B = \{(A \ \oplus B) \ominus B\}} \quad \textbf{(2.4)}$$

2.2.3. Morphological Structuring Element

The structuring element of morphological operations is a small matrix of pixels, each with a value of zero or one as shown in Figure 2.1. The size of the structuring element depends upon what features you wish to extract from the image. Larger structuring elements preserve larger features while smaller elements preserve the finer details of image features, we have chosen to work in our simulation with four different square SE (5,5) (11,11) (21,21) and (51,51) in order to see the effect

of the SE on the remote sensed image and select the best SE between them in terms of accuracy. Figure 2.2 and Figure 2.3 shows different results using the morphological operations using (5*5) & (9*9) Structuring Elements, the original image was taken from the "MIT Building Data Set".



(5*5) structuring element. **(3*3) structuring element.**

Figure 2.1 Structuring elements with size (3*3) and (5*5).

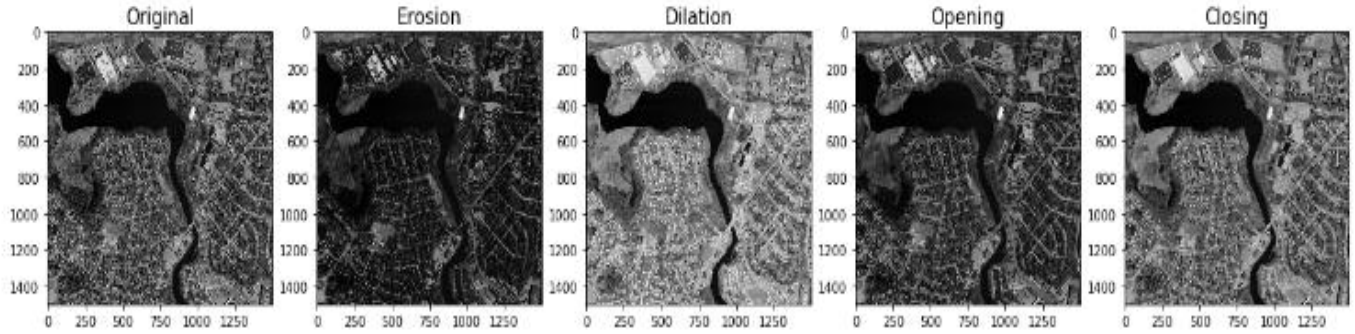


Figure 2.2 Results of MO with (5*5) structuring element

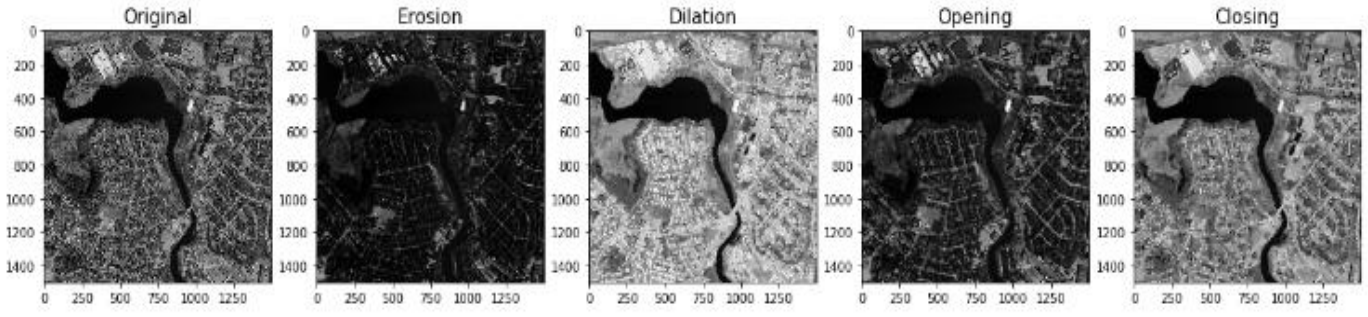


Figure 2.3 Results of MO with (9*9) structuring element

When a morphological operation is carried out, the origin of the structuring element is typically translated to each pixel position in the image in turn, and then the points within the translated structuring element are compared with the underlying image pixel values [9]. The details of this comparison and the effect of the outcome depend on which morphological operator are being used.

2.3. Gray-Level Co-occurrence Matrix (GLCM)

2.3.1. Definition

Gray-level co-occurrence matrix (GLCM) is a matrix showing different combinations of gray levels found within the image. The textural features extracted from the images by GLCM were helpful in identification of different regions in the images, It examines the spatial relationship among pixels and defines how frequently a combination of pixels are present in an image in a given direction Θ and distance d [10], The GLCM features were calculated along four of its neighbors in directions of 0°, 45°, 90°, and 135°. As shown in the Figure 2.4.

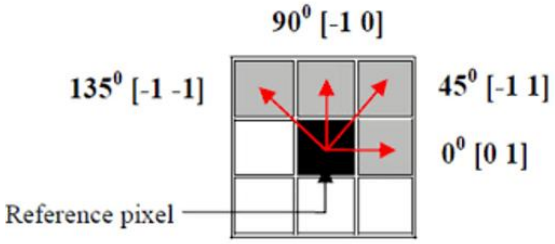


Figure 2.4 GLCM window 3*3

2.3.2. GLCM textural features

Usually the co-occurrence matrix is used to describe the textural properties. The matrix is calculated as P(i, j, d, Θ) where Θ can be 0, 45, 90, and 135 degrees, and d = 1, 2, 3, …, G, where G is the maximum gray value in the image and (i, j) indicates an entry in a cell the 2nd order [10]. GLCM features are computed considering the spatial relationship between any two intensity levels of the ROI (Region of Interest) using mathematical equations . The Extracted Features are

- **Contrast (CON):** Measures the local variations in the gray-level co-occurrence matrix is also known as variance and inertia. With Range = [0 (size (GLCM,1)-1) ^2]

$$\sum_{(i:j)} \boldsymbol{P(i;j)} \cdot \boldsymbol{(i-j)^2} \quad \textbf{(2.5)}$$

- **Correlation (COR):** Measures the joint probability occurrence of the specified pixel pairs. It is defined as follows with mean (μ_i, μ_j) and stander deviation (σ_i, σ_j).

$$\boldsymbol{\mu_i = \sum_{i,j=0}^{N-1} i\big(P(i,j)\big)} \qquad \boldsymbol{\sigma_i^2 = \sum_{i,j=0}^{N-1} P(i,j).(i-\mu_i)^2}$$

$$\boldsymbol{\mu_j = \sum_{i,j=0}^{N-1} j\big(P(i,j)\big)} \qquad \boldsymbol{\sigma_j^2 = \sum_{i,j=0}^{N-1} P(i,j).(j-\mu_j)^2}$$

$$\sum_{i,j=0}^{N-1} \boldsymbol{P\ (i,j).\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}} \quad \textbf{(2.6)}$$

- **Energy (ENG):** Provides the sum of squared elements in the GLCM. Also known as uniformity or the angular second moment.

$$\sqrt{\sum_{(i,j)} \boldsymbol{p(i,j)^2}}\ . \quad \textbf{(2.7)}$$

- **Homogeneity (HOM):** Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

$$\sum_{i,j=0}^{N-1} \frac{\boldsymbol{p_{(i,j)}}}{\boldsymbol{1+(i-j)^2}}\ . \quad \textbf{(2.8)}$$

- **Angular Second Moment (ASM):** Energy and MAX ASM and Energy use each $P_{(i,j)}$ as a weight for itself. High values of ASM or Energy occur when the window is very orderly.

$$\sum_{(i,j)} \boldsymbol{p(i,j)^2}. \quad \textbf{(2.9)}$$

- **Dissimilarity (Contrast Group):** GLCM Dissimilarity is the variation of gray-level pixels pairs.

$$\sum_{(\mathbf{i},\mathbf{j})} \boldsymbol{P\ (i,j).|i-j|}\ . \quad \textbf{(2.10)}$$

Figure 2.5 shows an example of image texture using GLCM with window size of (13*13).

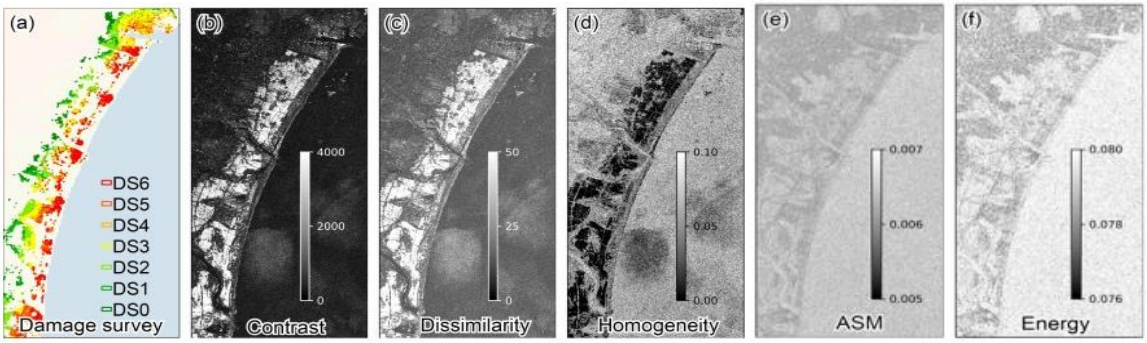


Figure 2.5 (a) damage buildings, the GLCM-based texture images obtained with window size of 13*13 are shown in the images (b) to (f) [11].

2.3.3. GLCM Algorithm

The most important steps on which we have implement the GLCM algorithm are as follows:

- The window sizes.
- Place windows in the first position over top left of the image.
- For the pixels within this window in this position, count the pixels, transform and normalize the GLCM
- Calculate the texture measure (ASM, Energy, Contrast...)
- Move the window over one pixel, and repeat the last two steps
- Continue covering all possible window positions until the texture image complete.

2.4. VARIOGRAM

2.4.1. Definition

A Variogram, **2ɣ (t),** is a description of the spatial continuity of the data. The experimental Variogram is a discrete function calculated using a measure of variability between pairs of points at various distances. The exact measure used depends on the Variogram type selected [12]. The distances between pairs at which the Variogram is calculated are called lags.

The **semi VARIOGRAM ɣ (t)** was originally defined by **Matheron** (1963) as half the average squared difference between points separated by distance h the semi Variogram are calculated as [13]

$$\boldsymbol{\gamma(h) = \frac{1}{|2N(h)|}\sum_{N(h)}\left(z_i - z_j\right)^2.} \quad \textbf{(2.11)}$$

Where N(h) is the set of all pairwise Euclidian distance d(i, j)=h , $\left|N(h)\right|$ is the number of distinct pairs in N(h), z_i and z_j are data values at spatial location i & j , respectively in this formulation, h represent a distance measure with magnitude only. Sometimes it might be desirable to consider direction in addition to distance. In such cases, h will be represented as the vector h having both magnitude and direction.

2.4.2. VARIOGRAM model

In VARIOGRAM algorithms we must select the type of model for how it fits the data because it will provide a mathematical function to the relationship between values and distances. We use functions that are the best fit like linear, exponential, spherical and Gaussian.

Generally, the squared term in the VARIOGRAM, for instance $(z_i\text{-}z_j)^2$,is the most useful term . However, when you have an understanding of how the phenomenon behaves with distance, you can better choose which model to use. By replacing the square terms by different powers:

- **The Absolute** $\left|z_i - z_j\right|$
- **The square root** $\sqrt{(z_i - z_j)}$

2.5. Conclusion

In this chapter we have reviewed the three different techniques that we used in our work and their types and method of work. Also, we have presented the basic steps of developing our algorithm to extract features using these techniques.

Chapter 3

Classification Algorithms In Machine learning

Chapter 3 Classification Algorithm in Machine Learning

3.1. Introduction

Classification is technique to categorize our data into a desired and distinct number of classes where we can assign labels to each class. Due to their great significance, classification is currently used in many applications, such as document classification, speech recognition, biometric identification, handwriting recognition. Classifiers can be

- **Binary classifiers:** Classification with only 2 distinct classes or with 2 possible outcomes like male and Female, classification of spam email and non-spam email classification of **Building** and **Non-Building,** etc.

- **Multi-Class classifiers***:* Classification with more than two distinct classes. Like classification of types of soil, classification of types of crops classification of mood/feelings in songs/music.

In this chapter we are going to present some algorithms that used for classification in machine learning, starting by the traditional one K-nearest-neighbor (KNN) then we introduce the most recent stage in the development ANN; eventually we present the peace-of-the-art we used in our project for binary classification of building and nonbinding classes using SVM algorithms.

3.2. K-Nearest Neighbors

3.2.1. Introduction

K-nearest-neighbor (KNN) is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it. An algorithm, looking at one point on a grid, trying to determine if a point is in group A or B, looks at the states of the points that are near it. The range is arbitrarily determined, but the point is to take a sample of the data. If the majority of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa.

The k-nearest-neighbor is an example of a "lazy learner" algorithm because it does not generate a model of the data set beforehand. The only calculations it makes is when it is asked to poll the data point's neighbors. This makes KNN very easy to implement for data mining.

3.2.2 Distance Metrics

The distance metric is the effective hyper-parameter through which we measure the distance between data feature values and new test inputs.

Usually, we use the **Euclidean approach** which is the most widely used distance measure to calculate the distance between test samples and trained data values. We measure the distance along a straight line from the point (x1, y1) to point (x2, y2).

$$\textbf{Euclidean Distance} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)} \qquad (3.1)$$

In Figure 3.1, the test sample (inside the circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $K = 3$ (outside circle), it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If, for example K= 5 it is assigned to the first class (3 squares vs. 2 triangles outside the outer circle).

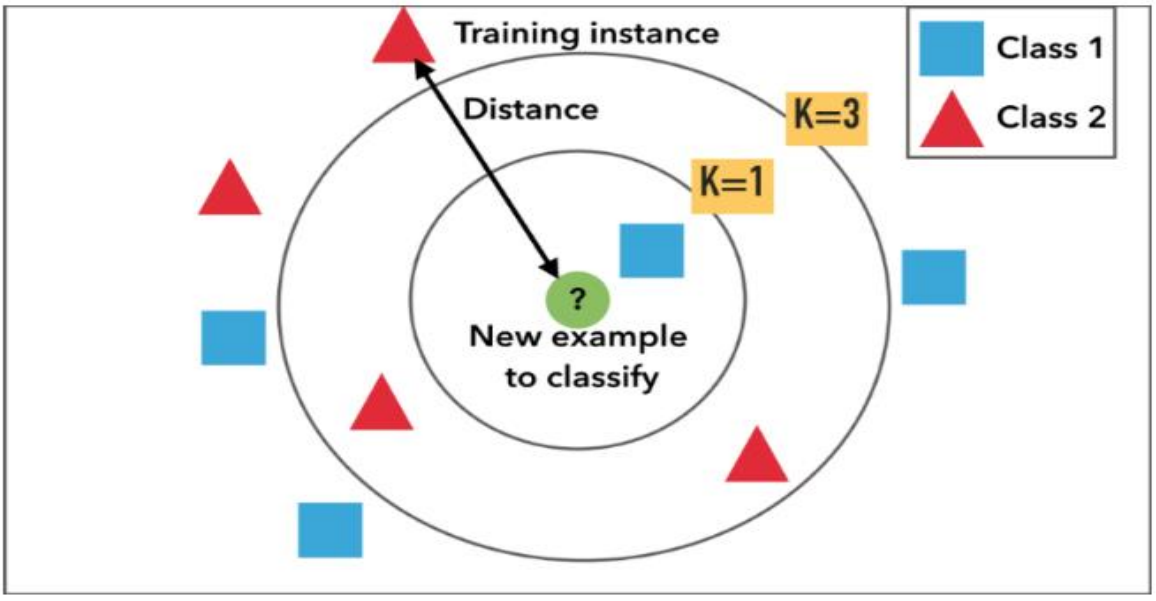


Figure 3.1: Example of *K*NN classification [14].

3.2.3. Choosing the best value of K

To select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before [15].

Here are some things to keep in mind

- As we decrease the value of K to 1, our predictions become less stable. Just think for a minute, imagine K=1 and we have a query point surrounded by several reds and one green (I'm thinking about the top left corner of the colored plot above), but the green is the single nearest neighbor. Reasonably, we would think the query point is most likely red, but because K=1, KNN incorrectly predicts that the query point is green.

- Inversely, as we increase the value of K, our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point). Eventually, we begin to witness an increasing number of errors. It is at this point we know we have pushed the value of K too far.

- In cases where we are taking a majority vote (e.g., picking the mode in a classification problem) among labels, we usually make K an odd number to have a tiebreaker.

3.3. Neural Networks (NN)

Neural networks are complex models, which try to mimic the way the human brain develops classification rules. A neural net consists of many different layers of neurons, with each layer receiving inputs from previous layers, and passing outputs to further layers. The way each layer output becomes the input for the next layer depends on the weight given to that specific link, which depends on the cost function, and the optimizer. The neural net iterates for a predetermined number of iterations, called epochs. After each epoch, the cost function is analyzed to see where the model could be improved. The optimizing function then alters the internal mechanics of the network, such

as the weights, and the biases, based on the information provided by the cost function, until the cost function is minimized.

Figure 3.2 shows a visual representation of such a network. The initial input is *x*, which is then passed to the first layer of neurons (the h bubbles in Figure 3.2*)*, where three functions consider the input that they receive, and generate an output. That output is then passed to the second layer (the g bubbles in Figure 3.2. There further output is calculated, based on the output from the first layer. That secondary output is then combined to yield a final output of the model.

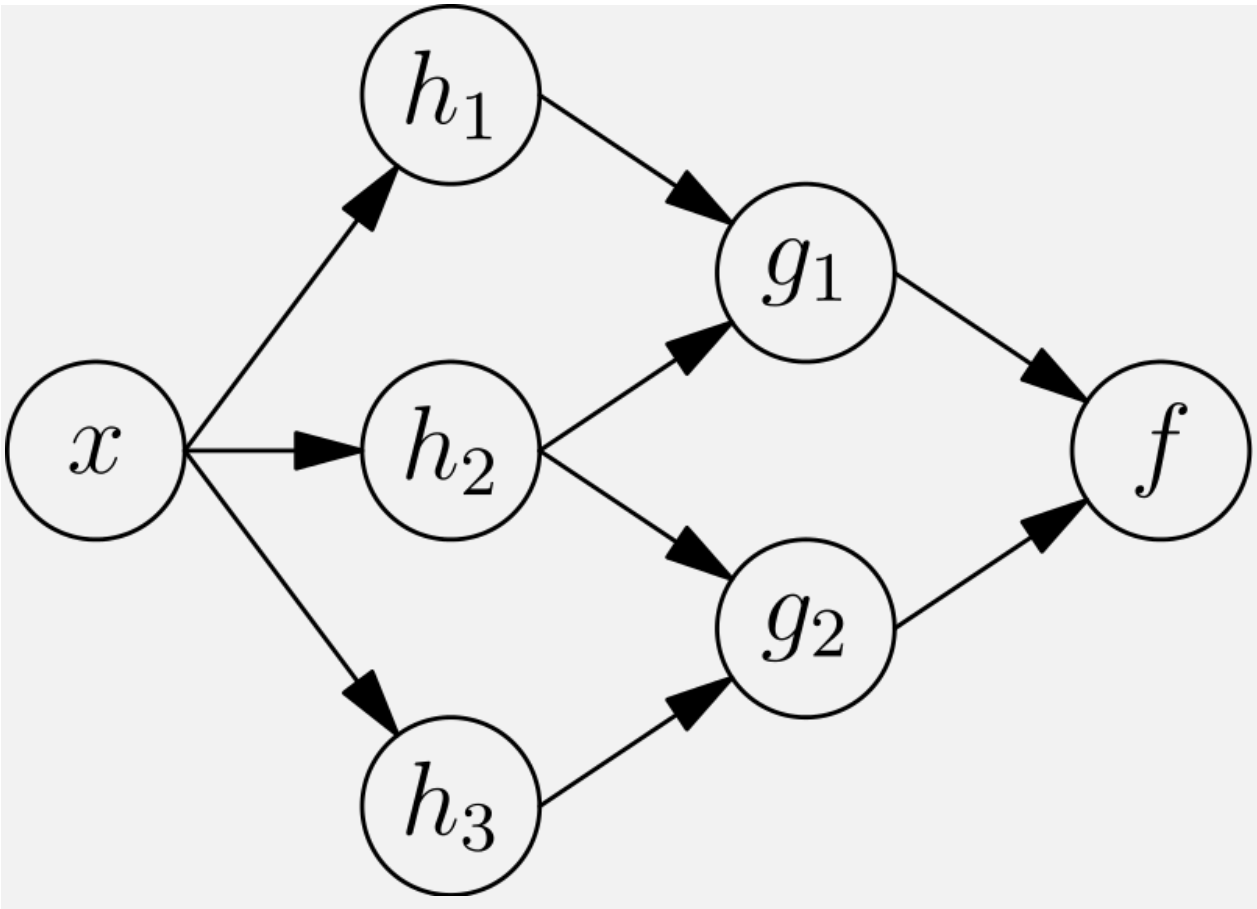


Figure 3.2: A **Visual Representation of a Simple Neural Net [16].**

A Convolution Neural Network (CNN) is a twist of a normal neural network, which attempts to deal with the issue of high dimensionality by reducing the number of pixels in image classification through two separate phases: the convolution phase and the pooling phase. After that it performs much like an ordinary neural network [16].

3.4. Support vector machine

3.4.1. Introduction

Support vector machines (SVM) is a supervised learning algorithm that sorts data into two categories. It is trained with a series of data already labeled into two categories, building the model as it is initially trained. The task of an SVM algorithm is to determine which category a new data point belongs to.

In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiates the two classes very well (see figure 3.3).

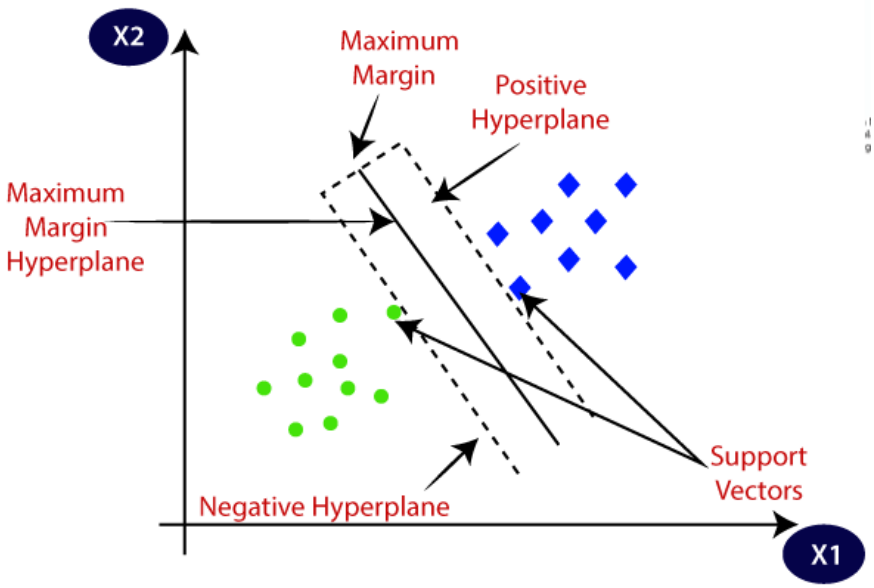


Figure 3.3: SVM hyperplane in 2D.

3.4.2. SVM parameters tuning

3.4.2.1. Kernel

Choosing the right kernel is crucial**,** because if the transformation is incorrect, then the model can have very poor results. As a rule of thumb, always check if you have linear data and, in that case, always use linear SVM (linear kernel). Linear SVM is a parametric model, but an Radial Basis Function (RBF) kernel SVM isn't**,** so the complexity of the latter grows with the size of the training set. Not only **is** more expensive to train an RBF kernel SVM**,** but you also have to keep the kernel matrix around, and the projection into this "infinite" higher-dimensional space where the data becomes linearly separable is more expensive as well during prediction [17].

3.4.2.2. Regularization (C)

The regularization parameter (often termed as C parameters in the python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example [18]. For large values of C, the optimization will choose a smaller-margin hyperplane. Conversely, a very small value of C will cause the optimizer to look for a larger margin separating hyperplane.

3.4.2.3. Gamma

The extent to which the impact of a single training instance reaches is called Gamma (γ), with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Whereas high gamma means the points close to plausible lines are considered in calculation.

3.4.2.4. Margin

A margin is a separation of the line to the closest class points [18]. When the separation between both classes is high, this provides a good margin that allows the points to be in their respective classes without going over to another class.

3.5. Conclusion

In this chapter, we have reviewed three classification algorithms which are used for classification in image processing, we focused on SVM and their parameters because it is the suitable one and is commonly used for image classification and easy in implementation for multispectral imagery.

Chapter 4

Experimental Results And Discussion

4.1. Introduction

After the presentation of the main tools used in developing our experiments in the previous chapter, we are going in this chapter to present and discuss the methodology and the results of our work. It is working noting that the software environment used in this work is **Python Version3**.

4.2. The study Area and Data set description

The dataset used in our experiments is Beijing Building Dataset is an elevation satellite image dataset which is integrated by satellite images and aerial photograph for building detection and identification. It contains 2000 images from Google Earth History Map of five different areas in Beijing taken on November 24, 2016,

In our work the proposed building detection procedure was implemented in three selected areas in the **Beijing Building Dataset** [19]**,** the capital city of China (Figure 4.1). The areas show different residential and/or industrial characteristics. The first study area (Sub-Area I) includes medium density of buildings with different shapes and sizes, the buildings falling within this area have different colored roofs. The second study area (Sub-Area II) includes high density of buildings also with different shapes and sizes. The third study area (Sub-Area III) includes low density of buildings with different shapes and sizes.

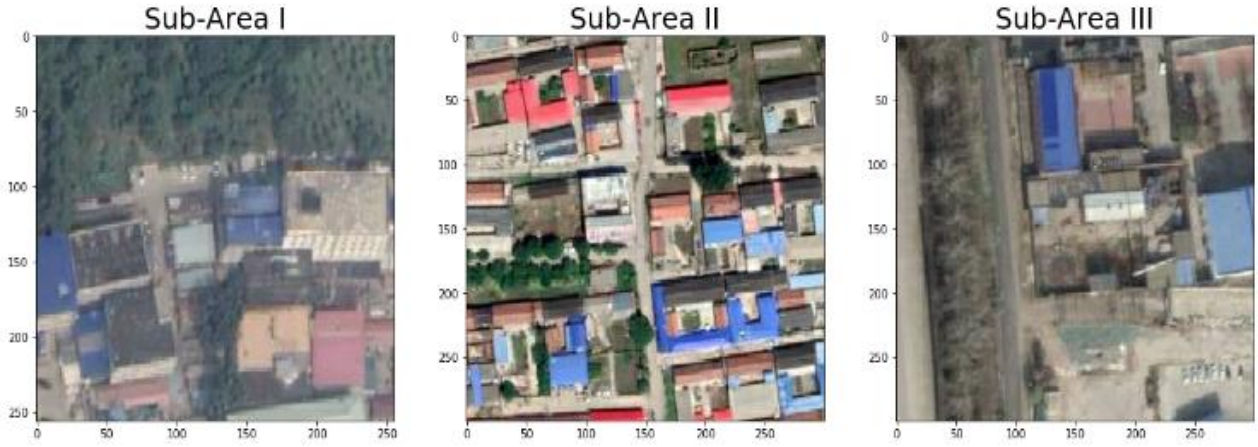


Figure 4.1 The Study Areas used in our work.

4.3 Flowchart of our work

The work we propose for building detection is illustrated in the Figure 4.2.

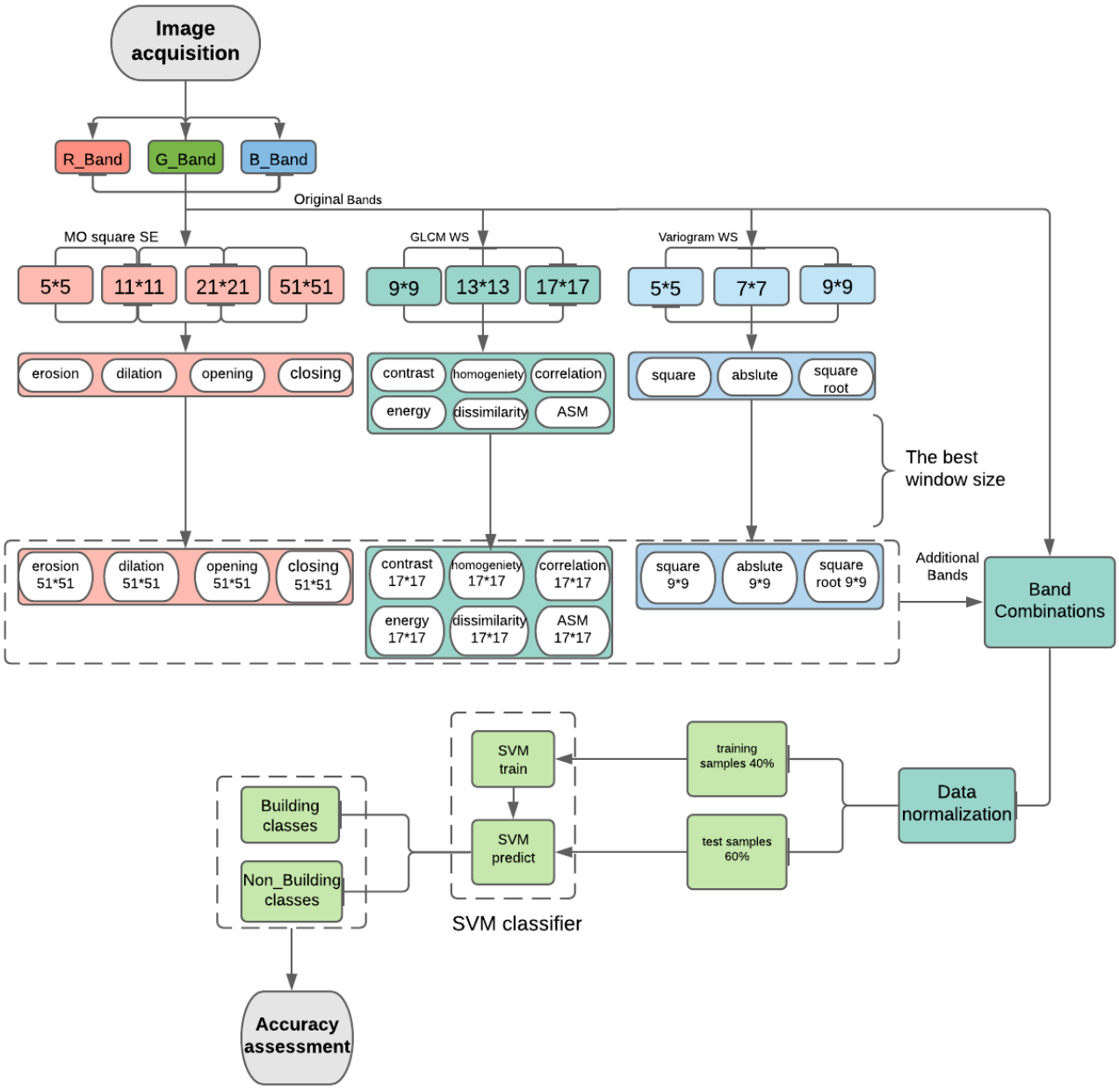


Figure 4.2. The Flowchart of the building detection procedure.

4.4. Spectral and spatial feature extraction

Our method of working is to fuse information from filtered data along with original data, this technique adds spatial features that filtered with the Morphological Filter, GLCM and Variogram to the existing spectral features in the original data to provide Spectral-spatial features to improve the classification process. The functions of the three techniques (MF, GLCM and VARIOGRAM) used in our procedure, only work on grayscale.

4.4.1. Spectral feature

First step is to extract the spectral features (Red, Green and Blue) bands of the remote sensed image as it is shown in Figure 4.3 for Sub-Area I.

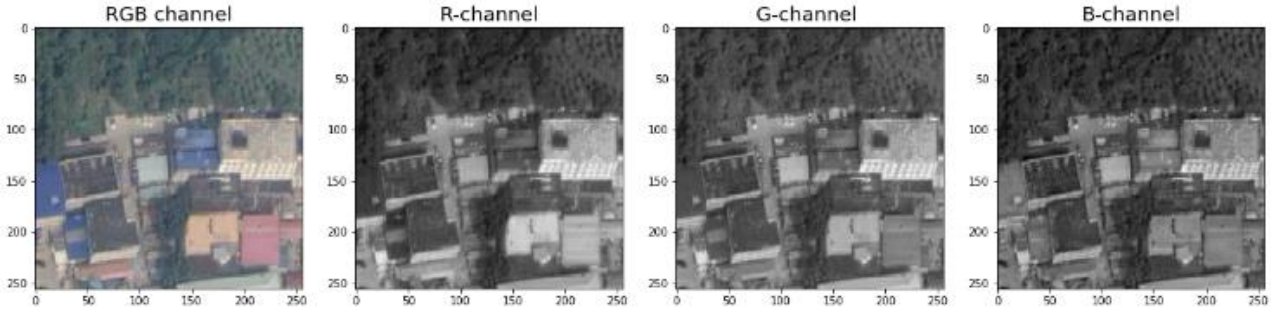


Figure 4.3. Spectral Features for Sub-Area I

4.4.2. Spatial feature

Morphological operators affect the accuracy of the classification, experimentally determining the best morphological operator for our application, our program was designed to apply different combinations using: opening, closing, erosion and dilation and different squares structuring. Element (SE). The results of the different morphological operators are shown in Figure 4.4.

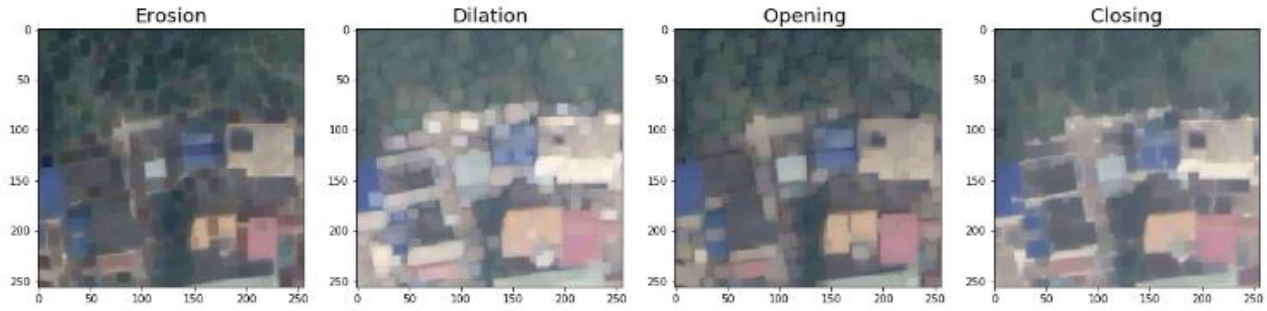


Figure 4.4. MO of Sub-Area I with SE (9*9)

The GLCM-derived texture information is anisotropic because the co-occurrence shift takes only one direction, to overcome this problem, the co-occurrence matrix should be computed in four directions and then the average should be taken. Therefore, we computed the co-occurrence matrix in horizontal (0 deg), vertical (90 deg), diagonal NE-SW (45 deg), and diagonal NW-SE (135 deg) directions and the average value were computed for each texture measure. Measured with different windows size (9*9, 13*13, 17*17), Figure 4.5 shows the GLCM-based texture measures with window size of (9*9) for Sub-Area I.

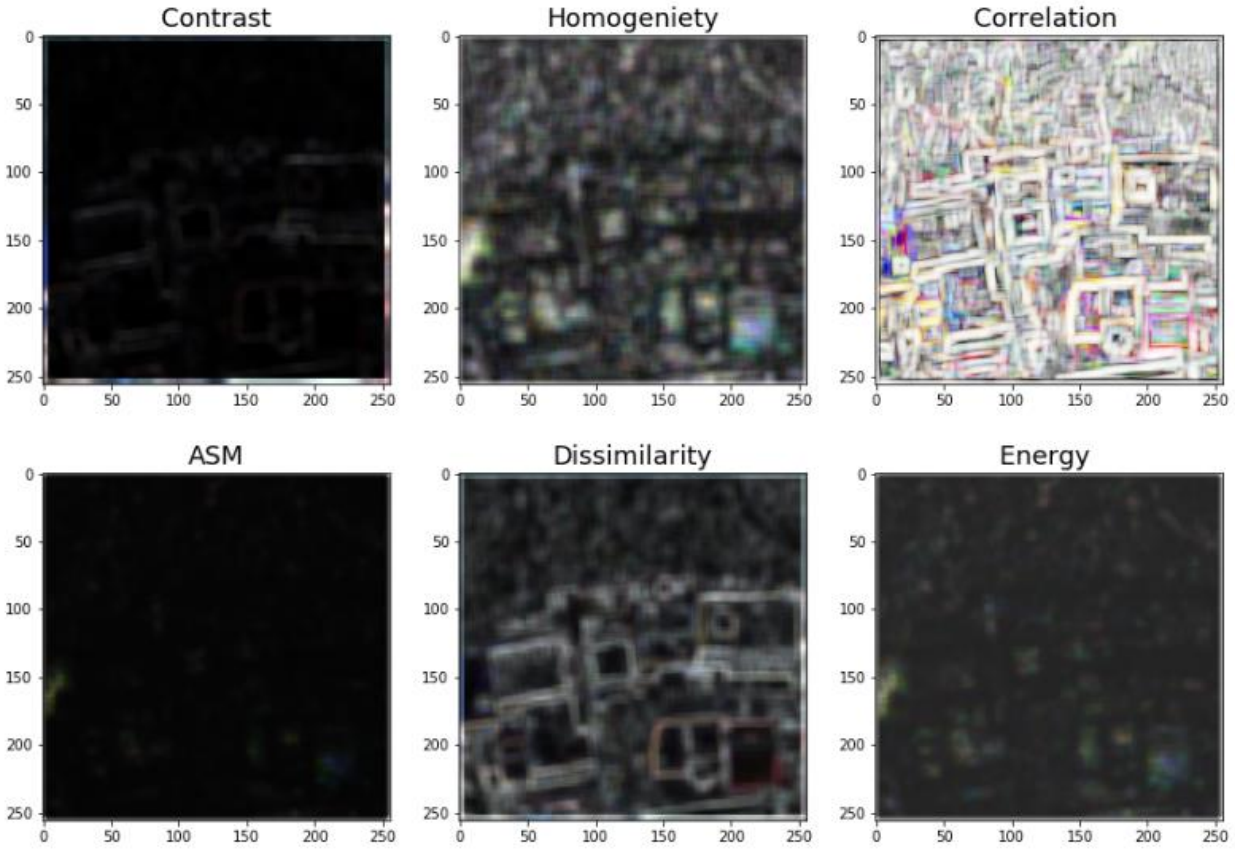


Figure 4.5. GLCM texture measures with WS (9*9)

The last technique we used for spatial feature extraction is the variogram the one that used to demonstrate the difference between data points as a function of distance. Our Variogram work method is to apply three distance functions (square, absolute, and square roots) to the pixel intensities, with respect to the four directions (0 °, 45 °, 90 °, 135 °), Figure 4.6 shows Variogram texture with window size (3*3).

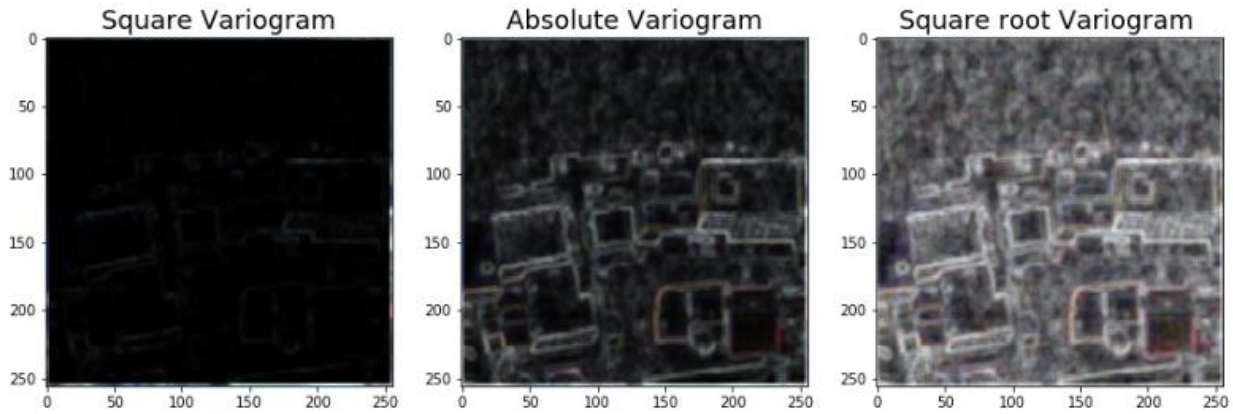


Figure 4.6. Variogram texture

4.5. Feature Normalization

For machine learning, the goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values, the normalization is required only when features have different ranges. The normalization technique is performed by dividing each extracted feature over the maximum value of that feature, taking into account R, G and B bands as one set from each.

4.6. SVM classification

In our study, a binary SVM classification was carried out using set of features to separate the class building from the class non-building, we used a combination of features as it showed in. Table 4.1**,** the original bands and the generated additional bands of the Sub-Areas were labeled as follows: B_1 (red band), B_2 (green band), B_3 (blue band), B_{DIL} (dilation), B_{ER} (erosion), B_{OP} (opening), B_{CLS} (closing), B_{CON} (contrast), B_{HOM} (homogeneity), B_{CORR} (correlation), B_{ENG} (energy), B_{ASM} (angular second moment), B_{DISS} (dissimilarity), B_{SQR} (square Variogram), B_{ABS} (absolute Variogram), B_{SQRT} (square root Variogram).

The ratios between the collected training samples and the whole pixels of the Sub-Areas are given in Table 4.2.

Table 4.1: The features set and their band combinations used in the SVM classification

Features set	Band combination
Features set 1	B_1, B_2, B_3
Features set 2	B_1, B_2, B_3, B_{DIL}
Features set 3	B_1, B_2, B_3, B_{ER}
Features set 4	B_1, B_2, B_3, B_{OPN}
Features set 5	B_1, B_2, B_3, B_{CLS}
Features set 6	B_1, B_2, B_3, B_{DIL}, B_{ER}, B_{OPN}, B_{CLS}
Features set 7	B_1, B_2, B_3, B_{HOM}
Features set 8	B_1, B_2, B_3, B_{CON}
Features set 9	B_1, B_2, B_3, B_{DISs}
Features set 10	B_1, B_2, B_3, B_{ENG}
Features set 11	B_1, B_2, B_3, B_{ASM}
Features set 12	B_1, B_2, B_3, B_{CORR}
Features set 13	B_1, B_2, B_3, B_{HOM}, B_{CON}, B_{DISs}, B_{ENG}, B_{ASM}, B_{CORR}
Features set 14	B_1, B_2, B_3, B_{SQR}
Features set 15	B_1, B_2, B_3, B_{ABS}
Features set 16	B_1, B_2, B_3, B_{SQRT}
Features set 17	B_1, B_2, B_3, B_{SQR}, B_{ABS}, B_{SQRT}
Features set 18	B_1, B_2, B_3, B_{DIL}, B_{ER}, B_{OPN}, B_{CLS}, B_{SQR}, B_{ABS}, B_{SQRT}
Features set 19	B_1, B_2, B_3, B_{DIL}, B_{ER}, B_{OPN}, B_{CLS}, B_{HOM}, B_{CON}, B_{DISS}, B_{ENG}, B_{ASM}, B_{CORR}
Features set 20	B_1, B_2, B_3, B_{DIL}, B_{ER}, B_{OPN}, B_{CLS}, B_{HOM}, B_{CON}, B_{DISS}, B_{ENG}, B_{ASM}, B_{CORR}, B_{SQR}, B_{ABS}, B_{SQRT}

Table 4.2. The percentages of the collected training pixels to whole pixels of the test sites.

	Shape	Training sample	percentage
Sub-Area I	256*256	26214	40%
Sub-Area II	300*300	36000	40%
Sub-Area III	300*300	36000	40%

After random division data set into 40 % train and 60 % as examination (test), the binary SVM classification was performed to distinguish the class building from the non-building class. We fed the training and the testing data to the classifier in the SVM format, Figure 4.7 Figure 4.8 shows an example of SVM format of the data set 1(without additional band) with label classes of building "1" and non-building "0," data set 15 (spectral with MO).

For performing the SVM classification, the selection of the kernel method, determination of the C parameter, and the parameters related to the kernel are important. In our case, the Radial Basis Function (RBF) was selected as the kernel method. This function works well in most cases. In addition, Gamma term (**γ)** was determined as the inverse of the number of bands in the input image and 10000 were taken for the value of the parameter penalty parameter (C).

	red	green	bleu	label
0	0.286275	0.349020	0.349020	0
1	0.294118	0.356863	0.356863	0
2	0.286275	0.349020	0.349020	0
3	0.274510	0.337255	0.337255	0
4	0.286275	0.349020	0.349020	0
...	...	...	...	...
65531	0.643137	0.584314	0.572549	0
65532	0.682353	0.627451	0.615686	0
65533	0.686275	0.631373	0.619608	0
65534	0.647059	0.592157	0.580392	0
65535	0.615686	0.560784	0.549020	0

65536 rows × 4 columns

Figure 4.7 Normalized matrix of Features set 1 for Sub-Area I (SVM format)

	red	green	bleu	er_R	er_G	er_B	dil_R	dil_G	dil_B	opn_R	opn_G	opn_B	cls_R	cls_G	cls_B
0	0.286275	0.349020	0.349020	0.309735	0.380531	0.380531	0.294118	0.356863	0.356863	0.309735	0.380531	0.380531	0.294118	0.356863	0.356863
1	0.294118	0.356863	0.356863	0.283186	0.353982	0.353982	0.294118	0.356863	0.356863	0.309735	0.380531	0.380531	0.294118	0.356863	0.356863
2	0.286275	0.349020	0.349020	0.278761	0.349558	0.349558	0.294118	0.356863	0.356863	0.309735	0.380531	0.380531	0.294118	0.356863	0.356863
3	0.274510	0.337255	0.337255	0.278761	0.349558	0.349558	0.309804	0.372549	0.372549	0.283186	0.353982	0.353982	0.294118	0.356863	0.356863
4	0.286275	0.349020	0.349020	0.278761	0.349558	0.349558	0.309804	0.372549	0.372549	0.278761	0.349558	0.349558	0.294118	0.356863	0.356863
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
65531	0.643137	0.584314	0.572549	0.606195	0.526549	0.526549	0.686275	0.631373	0.619608	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608
65532	0.682353	0.627451	0.615686	0.606195	0.526549	0.526549	0.686275	0.631373	0.619608	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608
65533	0.686275	0.631373	0.619608	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608
65534	0.647059	0.592157	0.580392	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608
65535	0.615686	0.560784	0.549020	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608	0.637168	0.566372	0.566372	0.686275	0.631373	0.619608

65536 rows × 16 columns

Figure 4.8 Normalized matrix of features set 6 for Sub-Area I (SVM format).

4.7 Classification result

The accuracy of the high-resolution sub-area I that is defined using the original function (spectral function 'Features Set 1') is 81.57 %, enhancing accuracy by incorporating the spatial features previously extracted using MO, GLCM and VARIOGRAM with different SE ad window sizes are shown in Table 4.3, Table 4.4, and Table 4.5.

Table 4.3. Accuracies of the features set that relates to MO with different square SE.

Features set	Accuracy assessment (SE = 5*5) %	Accuracy assessment (SE =11*11) %	Accuracy assessment (SE=21*21) %	Accuracy assessment (SE=51*51) %
Features set 1	81.57	81.57	81.57	81.57
Features set 2	82.29	82.93	83.55	85.89
Features set 3	82.27	82.67	83.22	82.83
Features set 4	82.48	82.27	84.49	85.70
Features set 5	82.41	82.85	83.85	86.25
Features set 6	83.61	86.15	89.04	94.26

Table 4.4 Accuracies of the features set that relates to GLCM with different WS

Features set	Accuracy assessment (WS = 9*9) %	Accuracy assessment (WS = 13*13) %	Accuracy assessment (WS = 17*17) %
Features set 7	82.15	82.23	82.41
Features set 8	82.51	82.77	82.47
Features set 9	81.89	82.02	82.42
Features set 10	82.05	82.40	83.02

Features set 11	**82.11**	**82.30**	**83.46**
Features set 12	**82.42**	**82.77**	**83.40**
Features set 13	**83.94**	**85.75**	**87.27**

Table 4.5 Accuracies of the Features set that relates to Variogram with different WS.

Features set	Accuracy assessment (WS = 5*5) %	Accuracy assessment (WS = 7*7) %	Accuracy assessment (WS = 9*9) %
Features set 14	**82.57**	**82.67**	**82.85**
Features set 15	**82.19**	**82.62**	**82.01**
Features set 16	**81.54**	**82.04**	**82.91**
Features set 17	**82.62**	**82.71**	**83.80**

From the above tables, we can deduce that

- The best structuring element for the morphological filter is **51*51**.
- The best windows size for the GLCM is **17*17**
- The best windows size for the variogram is **9*9**

After extracting the best SE and WS, as shown in Table 4.6 the most effective combination that gives the best accuracy result for our procedure is the combination of all spatial features.

Table 4.6. Accuracies of the features set 18, 19, and 20 with optimum SE and WS

Features set	Features set 18	Features set 19	Features set 20
Accuracy %	**95.00**	**94.58**	**95.48**

The classification result of the Features set 20, that provides us the best accuracy is shown below in Figure 4.9**.**

```
{'C': 10000, 'gamma': 0.023809523809523808, 'kernel': 'rbf'}
SVC(C=10000, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.023809523809523808,
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
              precision    recall  f1-score   support

           0     0.9665    0.9630    0.9648     16854
           1     0.9339    0.9400    0.9369      9361

    accuracy                         0.9548     26215
   macro avg     0.9502    0.9515    0.9508     26215
weighted avg     0.9549    0.9548    0.9548     26215
```

Figure 4.9. Classification result for Sub-Area I with highest accuracy.

Figure 4.10 shows the classified image carried out using Features set 1, 6, 18, 19 and 20 .

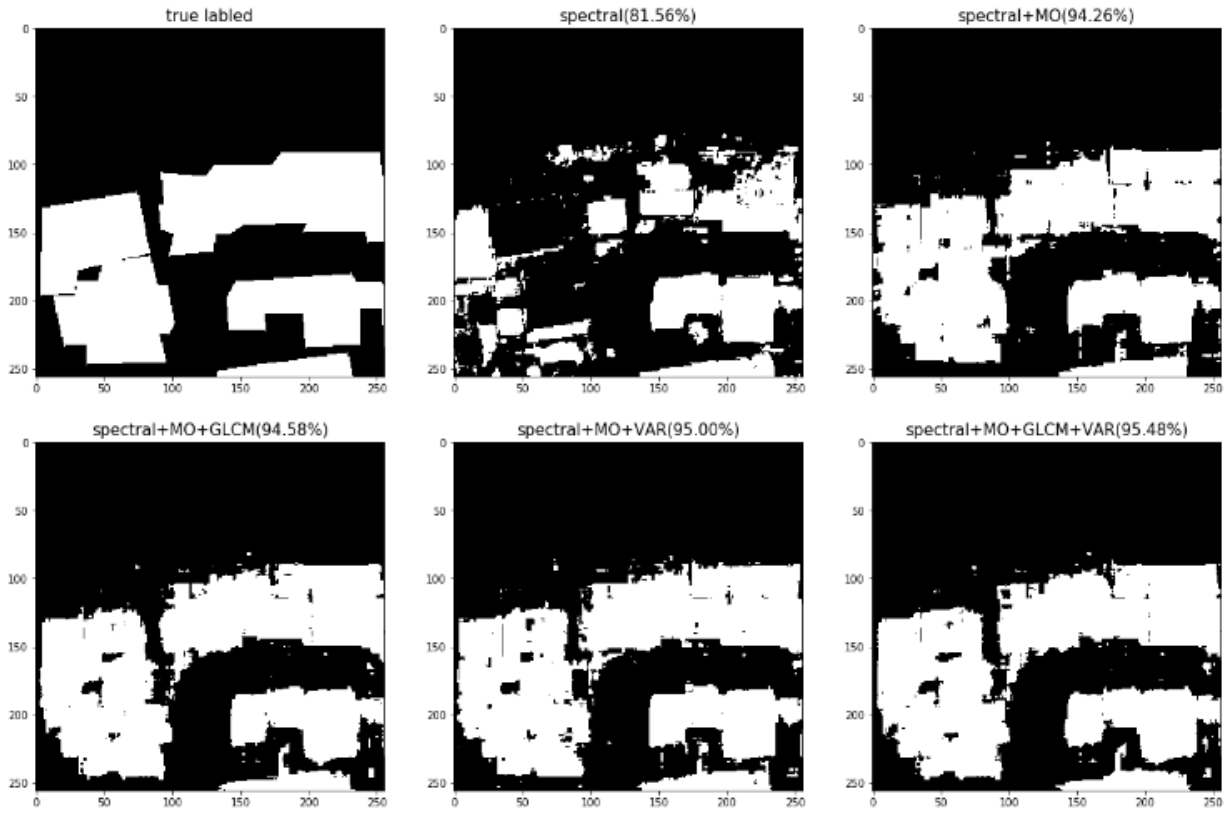


Figure 4.10 The classified image for Sub-Area I with different combinations of spectral-spatial features for SVM classifiers.

The same procedure has been applied to Sub-Area II and Sub-Area III with a picture scale of 300 * 300 with different building density and different urban characteristics to test our algorithm. The results of the SVM classifier for Sub-Area II and Sub-Area III with the optimum SE and WS are shown in Table 4.7.

Table 4.7 Accuracy Table for Sub-Area II and Sub-Area III with (SE =51*51, WS = 17*17 (GLCM), and WS = 9*9 (VARIOGRAM)).

Features sets	Accuracy % (Sub-Area II)	Accuracy% (Sub-Area III)
Features set 1	75.87	88.75
Features set 2	78.98	89.32
Features set 3	77.01	89.48
Features set 4	80.77	89.01
Features set 5	77.94	91.3
Features set 6	86.02	94.02
Features set 7	78.10	89.49
Features set 8	78.18	89.94
Features set 9	78.65	90.73
Features set 10	76.94	89.46
Features set 11	76.76	88.83
Features set 12	77.27	89.37
Features set 14	77.63	89.37
Features set 15	78.08	89.62
Features set 16	77.63	89.03
Features set 18	86.86	95.82
Features set 19	88.61	96.94
Features set 20	**90.23**	**96.94**

The reshaping of the SVM classifier result of the two Sub-Areas are shown in Figure 4.11 and Figure 4.12 respectively.

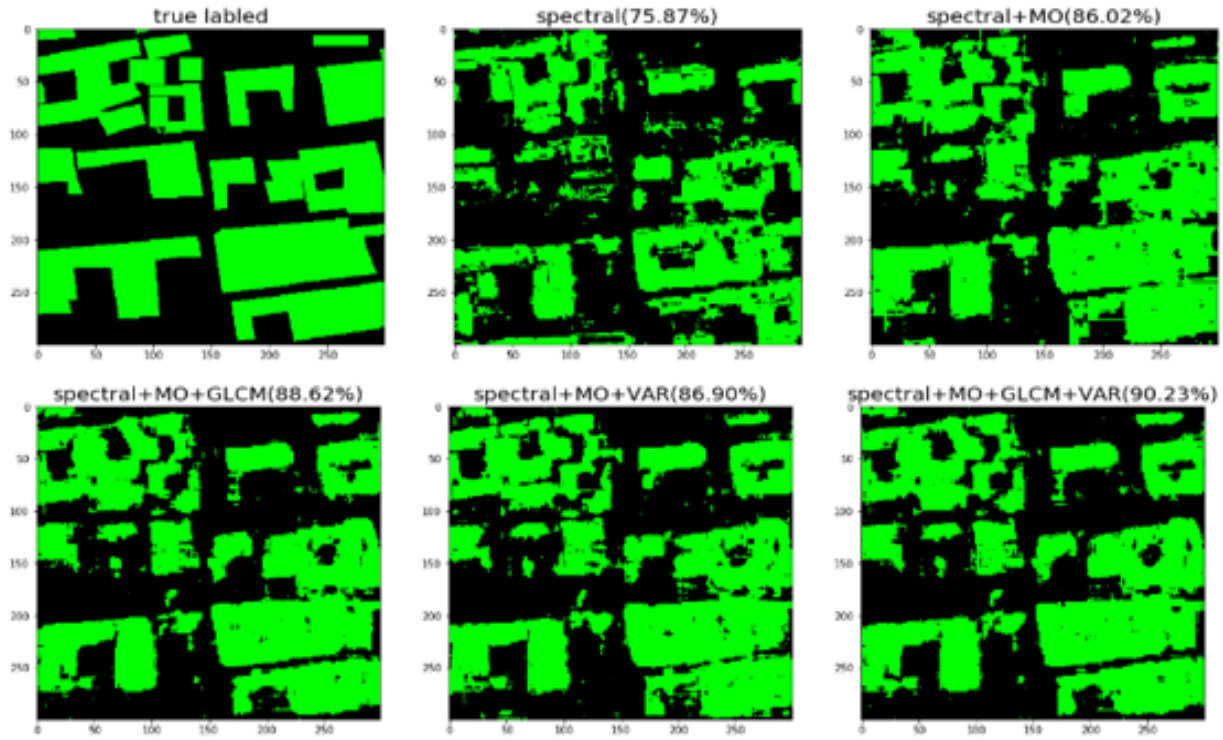


Figure 4.11 the classified images for Sub-Area II.

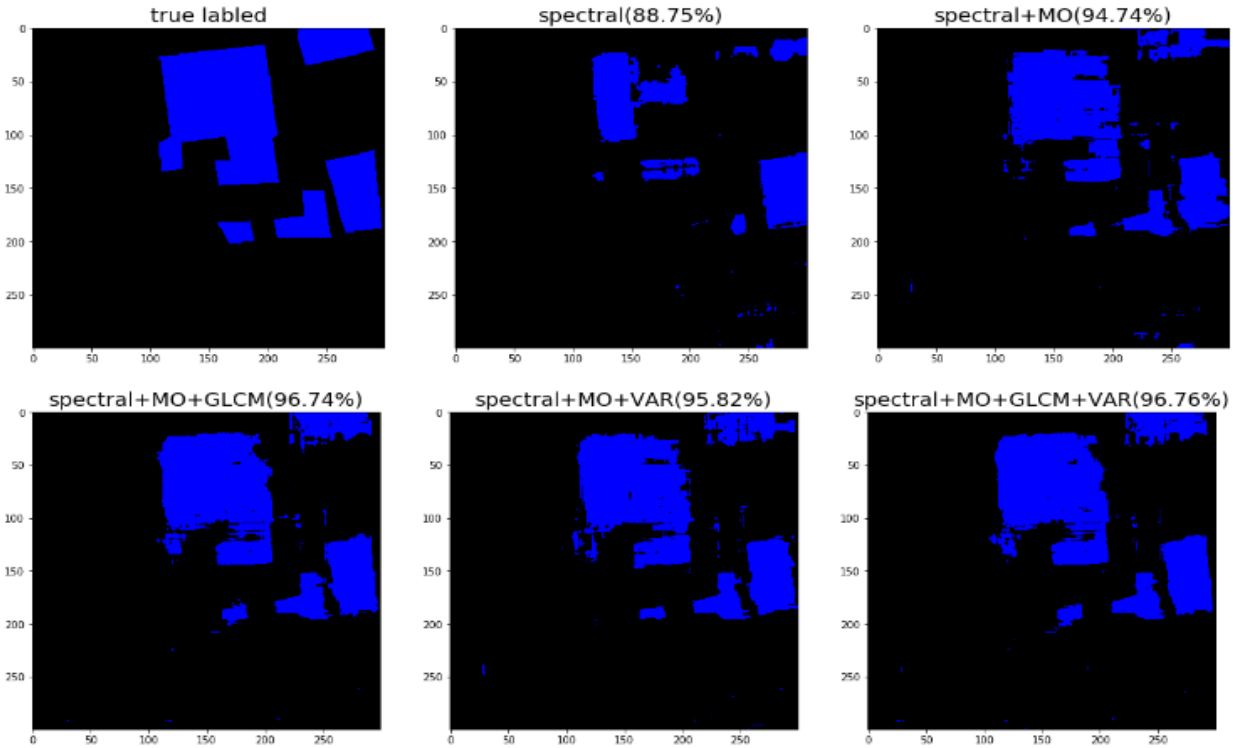


Figure 4.12 The classified images for Sub-Area III.

4.8 Results discussion

In this experiment, we performed the extraction of features for the original data, beginning with the original features (spectral) and then using the additional features extracted by morphological operators, gray level co-occurrence matrices and variogram. Our goal was to improve the accuracy as much as possible. The results of this study have shown that the SVM classification is very satisfactory for building detection from high-resolution satellite imagery. For all of the sub-areas we tested.

Through the use of only the original bands (red, green and blue), buildings through distinct reflectance values from other objects were effectively identified, while buildings with identical spectral values with other classes, such as paths, could not be accurately detected. With the use of the spectral-spatial features in the classification process, it was possible to detect buildings that have different spectral characteristics.

We found that the use of spectral-spatial features in SVM classification improved the accuracy of the building detection. we can also see that the change of the used (MO, GLCM and VARIOGRAM) parameters has a direct impact on the spatial features, which affects the classification accuracy, the optimum parameters that provided the best accuracy are square SE (51*51), WS (17*17) and (9,9) for MO, GLCM and VARIOGRAM respectively.

The SVM classifier produced quite accurate results for the proposed building detection procedure. For all test areas, the overall accuracies were computed to be between 81% and 95%. The findings were very encouraging where the accuracy of the **Sub-area I** improved from **81.57%** to **95.48%,** from **75.87%** to **90.20%** and **88.75%** to **96.94%** for **Sub-area II** and **Sub-area III**, respectively.

Conclusion

In this project, we worked on the classification of images using the support vector machine (SVM). We used the original image features (spectral features) for classification, and the results obtained were used as a reference during the entire experiment. Then, we used different combinations of features extracted from MO, GLCM, and VARIOGRAM as a spatial feature extraction technique to deliver spectral-spatial feature vectors as inputs to the support vector machine classifier, and then searched for parameters that provide the best data representation in terms of classification accuracy.

Using the above techniques, the overall accuracy of 95.48% was achieved with specific parameters which are very promising in comparison to original data Support Vector Machine Classification which had an overall accuracy of 81.56% in our case, with the same dataset. Other datasets also tested and we got promising results. The results of this study have shown that the SVM classification is very satisfactory for building detection from high-resolution satellite imagery.

It can be concluded at this point that the classification of multispectral data on the basis of spectral-spatial information gives better results with almost **14%** compared to the classification of the original spectral data. We can conclude through this modest work that we have succeeded in applying the extraction technique of spectral-spatial features as a model for the classification of multispectral data.

Buildings can be detected in difficult scenes where the building has similar color to the background, and the other buildings have different textural or geometric properties than the background. In this case, color features may not provide useful information to the SVM classifier for the detection, but the classifiers can extract discriminative information from the texture and shape features.

Finally, deep learning approach for building detection is proven to bring highly accurate results for a fraction of time and cost that otherwise would be spent on manual work.

References

[1]. Roger Fong and Frank de Morsier, Deep learning approach for building detection, 18 June 2020. Available : https://picterra.ch/blog/deep-learning-approach-for-building-detection/

[2]. Jensen, J. R, "Introduction and History", Remote Sensing of the Environment, 2007. Available: https://earthobservatory.nasa.gov/features/RemoteSensing

[3]. R. Paschotta, article on 'multispectral imaging' in the Encyclopedia of Laser Physics and Technology, 1. Edition October 2008, Wiley-VCH, ISBN 978-3-527-40828-3

[4]. C. Palaniswami, A. K. Upadhyay and H. P. Maheswarappa, Spectral mixture analysis for subpixel classification of coconut, Current Science, Vol. 91, No. 12, pp. 1706–1711, 25 December 2006.

[5] David DiBiase, "Image classification", The Nature of Geographic Information, chapter 8, March 2020, Available : https://www.e-education.psu.edu.

[6]. Jason Brownlee, "Supervised and Unsupervised Algorithms", Machine Learning Algorithms, March 16, 2016. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

[7]. Nick Efford, "Morphological Image Processing", Digital Image Processing: A Practical Introduction using Java, 23 MAI 2000.

[8].https://uotechnology.edu.iq/ce/lecture%202013n/4th%20Image%20Processing%20_Lectures/DIP_Lecture11.pdf. Accessed on Juin,2020

[9]. R. Fisher, S. Perkins, A. Walker and E. Wolfart, "Structuring Elements", Morphological Image Processing, Vol. 2, Issue 2, 2011, PP-35-43.

[10]. P.K. Bhagat, Prakash Choudhary, Kh. Manglem Singh, "A comparative study for brain tumor detection in MRI images using texture features", Sensors for Health Monitoring, pages 267–269, 2019

[11]. ISPRS Journal of Photogrammetry and Remote Sensing.Volume 149, March 2019, Pages 14-28.

[12]. Cameron, K, and P Hunter. 2002. Using Spatial Models and Kriging Techniques to Optimize Long-Term Ground-Water Monitoring Networks: A Case Study. Environmetrics 13:629-59.

[13]. Journel, A.G., 1988, Geostatistics for the Environmetal Sciences, EPA Project Report, Project No. CR 811893, EPA/EMSL, Las Vegas, Nevada.

[14]. Adi Bronshtein, "A Quick Introduction to K-Nearest Neighbors algorithm", Journal Blog, Apr 11, 2017.

[15]. Onel. Harrison, Machine Learning Basics with the K-Nearest Neighbors Algorithm, Sep 10, 2018.Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[16]. Oliver Knocklein, Classification Using Neural Network, Jun 5, 2019. Available: https://towardsdatascience.com/classification-using-neural-networks-b8e98f3a904f/.

[17]. Czako Zoltan," Tuning Parameters", SVM and Kernel SVM, Nov 13, 2018. Available : https://towardsdatascience.com/svm-and-kernel-svm-fed02bef1200

[18]. Savan Patel, "Tuning parameters: Kernel, Regularization, Gamma and Margin", SVM Support Vector Machine Theory, May 3, 2017.

[19]. Beijing building datasets. https://ieee-dataport.org/documents/beijing-building-dataset. Accessed on 05 November 2019.

People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University M'Hamed
BOUGARA - Boumerdes

Institute of Electrical and
Electronic Engineering

Université de Boumerdes
University of Boumerdes

Department of Electronics

Authorization for Final Year Project Defense

Academic year: 2019/2020

The undersigned supervisor: A. DAAMOUCHE
authorizes the student(s):

LOKMAN	BOULEBNANE	Option: Telecommunications
ALI	BENTAALA	Option: Telecommunications

to defend their final year Master program project entitled:

Building detection from high resolution remote sensing imagery

during the ☐ June ☑ September session.

Date: 19/09 / 2020

The Supervisor

The Department Head

رئيس قسم الإليكترونيك بالنيابة

ع. زيتوني