CHAPTER 6

UNSUPERVISED TEXTURAL CLASSIFICATION OF IMAGES FOR IDENTIFICATION OF DIFFERENT REGIONS

6.1 INTRODUCTION

The present study proposes a new unsupervised textural classification of images based on Texture Unit proposed by He and Li Wong [100]. The basic concept is that a texture image can be considered as a set of essential small units, called as termed texture units (TU), which characterize the local texture information for a given pixel and its neighborhood. The statistics of all the texture units over the whole image reveal the global texture aspects.

In a square raster digital image, each pixel is surrounded by eight neighboring pixels. The local texture information for a pixel can be extracted from a neighbourhood of 3 X 3 pixels, which represents the smallest complete unit (in the sense of having eight directions surrounding the pixel).

Given a neighbourhood of 3 X 3 pixels, which will be denoted by a set of containing nine elements: \( V = \{V_0, V_1, ..., V_8\} \), where \( V_0 \) represents the intensity value of the central pixel and \( V_i \) \( \{i = 1, 2, ..., 8\} \), is the intensity value of the neighboring pixel i. The corresponding texture unit (TU) is defined by a set
containing eight elements. \( \text{TU} = \{E_1, E_2, \ldots, E_8\} \), where \( E_i \), \( (i=1,2,\ldots,8) \) is determined by the formula:

\[
E_i = \begin{cases} 
0 & \text{if } V_i < V_0 \\
1 & \text{if } V_i = V_0 \\
2 & \text{if } V_i > V_0 
\end{cases} \quad \text{for } i = 1,2,\ldots,8
\]

and the element \( E_i \) occupies the same position as the pixel \( i \).

As each element of TU has one of three possible values, the combination of all the eight elements results in \( 3^8 = 6561 \) possible TU in total. There is no unique way to label and order the 6561 TU. The 6561 texture units are labeled by using the following formula:

\[
N_{\text{TU}} = \sum_{i=1}^8 E_i \times 3^{i-1}, \quad N_{\text{TU}} \in \{0,1,2,\ldots,(N_{\text{TU}})\}
\]

where \( N_{\text{TU}} \) represents the texture unit number and \( E_i \) is the \( i^{th} \) element of texture unit set \( \text{TU} = \{E_1, E_2, \ldots, E_8\} \).

To reduce the overall complexity of texture units in the present study instead of 3 values, 2 values are chosen. This reduces overall texture units from \( 3^8 = 6561 \) to \( 2^8 = 256 \) in the following way.

\[
E_i = \begin{cases} 
0 & \text{if } V_i < V_0 \\
1 & \text{if } V_i > V_0 \quad \text{for } i = 1,2,\ldots,8
\end{cases}
\]

In addition, the eight elements may be ordered differently. If the eight elements are ordered clock-wise as shown in Fig. 1, the first element may take eight possible positions from top left (a) to middle left (h), then the 256
texture units can be labeled by the formula under eight different ordering ways (from a to h).

\[ N_{TU} = \sum f_s * 2^{i} \]

Fig. 6.1 Eight clockwise successive ordering ways of the eight elements of a texture unit: the first element may take eight possible positions from a to h.

The defined set of 256 texture units describes the local texture aspect of a given pixel that is the relative gray level relationships between the central pixel and its neighbors. Thus, the statistics on frequency of occurrence of all the texture units over a large region of an image should reveal texture information. The texture spectrum can be termed as the frequency distribution of all the texture units, with the abscissa indicating the texture unit number \( N_{TU} \) and the ordinate representing its occurrence frequency.

It should be noted that the labeling method chosen may affect the relative positions of texture units in the texture spectrum, but will not change their frequency values in the texture spectrum. It should be also noted that the local texture for a given pixel is characterized by the corresponding texture unit, while the texture aspect for an uniform texture image is revealed by its texture spectrum calculated within an appropriate window. The size of the window depends on the
nature of the texture image. An example of a Texture unit calculated from the Fig. 6.2 is given below

\[
\begin{array}{c|c|c}
83 & 28 & 60 \\
76 & 55 & 23 \\
90 & 55 & 10 \\
\end{array} \rightarrow \begin{array}{c|c|c}
1 & 0 & 1 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
\end{array}
\]

Fig. 6.2 Example of transforming a neighbourhood to a Texture Unit

\[N_{TU} = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 1 \times 2^7\]

\[= 1 + 0 + 4 + 0 + 0 + 0 + 64 + 128\]

\[= 197.\]

6.2 SEGMENTATION APPROACH

Texture analysis plays an important role in the interpretation and understanding of terrain, biomedical or microscopic images. Most work on texture analysis has been devoted to the analysis of texture features of an entire image. However, one of the most important image processing operations is the textural classification of an image, that is, to partition the image space into a set of subregions, each of which is homogeneously textured. The constraint is that each pair of adjacent subregions is differently textured [31].

Methods of image classification can be broadly grouped into two major approaches [18]. The first is supervised classification, which is based on some discrimination function using several image features [24]. First, the data samples are collected and properly normalized from all the classes to be recognized. In the classifier design, training or learning stage, one will determine the discrimination functions to find the boundaries that separate the classes. Such functions will then be used to classify the whole image. This method requires a
priori information about the image classes to be recognized. The linear maximum likelihood discrimination function is widely employed because of its easy implementation.

The second major approach to image classification is the unsupervised clustering of image data, where no a priori information is required. A large number of clustering algorithms are available in the literature and widely used in practice to classify multispectral image data [1, 16, 17, 22, 43]. Cluster analysis is viewed as a process of partitioning an image into groups such that patterns belonging to the same group are more similar to each other than are patterns belonging to different groups.

In practice, the supervised approach is more frequently quoted than the unsupervised approach in the texture classification of images [10, 37, 46, 104]. This may be partly due to the fact that a large number of features are available in texture analysis, derived from structural or statistical approaches, [30] and that several powerful features can be pre-selected in the training stage with a priori knowledge of the texture classes. These features will then be used to classify the original image with a better performance of discrimination for the particular classification problem considered [37]. However, unlike selecting training sites for multispectral classes, it is sometimes difficult to locate all the texture patterns to be recognized in real image data, while the quality and precision of class samples take an important role in the elaboration of discrimination functions. What one need may be just a partition of an image based on texture similarity involving no a priori knowledge of the final classes. From
this the present study observes that unsupervised classification is also a good way to test the discriminating performance of the feature used in the classification.

The quality of texture classification depends a lot on the discriminating ability of the image features used in the classification. One should use those features which most completely embody information of texture in the original images. Structural and statistical approaches are the two major methods for textural feature extraction [30, 37, 38]. On one hand, we cannot use too many features in a classification algorithm and, on the other hand it is sometimes difficult to make a judicious and correct localization of texture classes in real image data to perform a training stage for selecting the most promising features. Recently the texture spectrum to the textural characterization of images is proposed [36, 101]. One of the advantages of this method is that the texture aspects of an image are characterized by the corresponding texture spectrum instead of a set of texture measures and the texture spectrum can directly be used for image classification. This feature has been evaluated with success for texture characterization through several natural images and remotely sensed images data [32, 33, 34, 35, 36, 99, 101].

6.3 UNSUPERVISED CLASSIFICATION ALGORITHM

A clustering algorithm has been designed to perform an unsupervised texture classification. Consider a cluster as an aggregation of points in the test space such that the distance between any points in the cluster is less than the distance between any point in the cluster and any point not in it. In this thesis, this minimum distance precision rule was employed and the integrated
absolute difference between two texture spectra has been taken as the distance between them. The present method provides to the user a flexible way of supplying the number of clusters desired (K) and an initial threshold \( T \). The algorithm will then determine, through an iterative process, the best value of \( D \), which would be the minimum between-cluster distance, such that a new cluster will be created once the minimum distance between a pattern and all cluster centers is greater than \( T \); and that the final number of clusters will be close or equal to the user-defined number (K). In practice, this can be realized as the procedure described as follows:

(1). Transform the original image into a texture unit image: that is, scan the whole image using a 3x3 matrix. The central pixel of the matrix will be assigned by the corresponding texture unit number (NTU) calculated through the method described in section above. This results into a new image in which the value of the pixel represents the corresponding NTU of the pixel. All the remaining processes will be realized in this NTU image instead of the original image.

(2). Input initial parameters, including the desired number of classes (K), a threshold value (T) and a step value (DS). The choice of these values will be discussed at the end of this section.

(3). Set the number of effective classes NC to zero (NC=0) (here, an effective classification means a not-empty classification) and scan the whole image using a window of \( M \) pixels \( \times M \) pixels:
(a) The first window will be chosen as the sample subimage of the first classification and let NC=1. The second windows will be:

1. Either classified to the first classification if the distance between the second window and the sample sub-image of the first class is less than or equal to the threshold value \( T' \) or

2. Chosen as the sample subimage of the second class if the distance is greater than \( T' \) and let NC=2.

(b) The integrated absolute difference between the texture spectrum of the window and the spectrum of the sample subimage is considered as the distance between them:

\[
D(i) = \sum_{j=1}^{256} |W(j) - S(i,j)|
\]

Where \( D(i) \) denotes the distance between the window \( W \) and the sample subimage of class \( i \); \( W(j) \) represents the occurrence value of the texture unit \( j \) in the window considered; \( S(i,j) \) represents the occurrence value of the texture unit \( j \) in the sample subimage of class \( i \).

(c) The algorithm continues by scanning the rest of the image. For a window encountered, the distances between this window and the sample subimage of each effective class will be calculated. Three possible situations will be considered:

- The central pixel of the window will be assigned to the class \( L \) such that \( D(L) \) is minimum among all the \( D(i) \), for \( i = 1, 2, \ldots, NC \) (where \( NC \) represents the number of effective classes), under the condition that \( D(L) \leq T \).

The algorithm will continue by considering the next windows.
• The window will be chosen as the sample subimage of the (NC+1)th class if \( D(L) > T \) and \( (NC+1) \leq K \). Then, let \( NC = NC + 1 \) and the algorithm will continue by considering the next windows.

• If \( D(L) > T \) and \( (NC+1) > K \), let \( T = T + DS \) and the algorithm will go back to the beginning of the scan, that is back to step (3).

(4) After certain iterations, this process becomes stable and the algorithm stops with the current value of \( NC \) and \( T \). All the pixels will be classified to one of the \( NC \) classes.

As for the choice of the initial parameters, the ideal value of \( I' \) would be the distance between the two closest classes, that is the minimum between class distances. A too large initial value of \( I' \) will result in a small number of effective classes, that is \( NC \ll K \), a small initial value of \( I' \) will not affect the final classification result except for the number of iterations. The value of \( DS \) affects the convergent speed of the algorithm. The choice of the size of the scanning window \((M \times M)\) depends on the texture structure of the original image to be classified. A small window characterizes small textures while a larger window should characterize large textures.
Fig. 6.3 Result of the unsupervised texture classification of TBK-1 with window size 2 and no. of samples (a) 2, (b) 3, (c) 5.
Fig. 6.4 Result of the unsupervised texture classification of TBR-1
with window size 3 and no. of samples (a) 2 (b) 3 (c) 5
Fig. 6.5 Result of the unsupervised texture classification of TBK-1 with window size 5 and no. of samples: (a) 2, (b) 3, (c) 5.
Fig. 6.6 Result of the unsupervised texture classification of 1B Ke with window size 2 and no. of samples (a) 2, (b) 3, (c) 5.
Fig. 6.7 Result of the unsupervised texture classification of TBK-2 with window size 3 and no. of samples (a) 2, (b) 3, (c) 5
Fig. 6.8 Result of the unsupervised texture classification of HRK-2 with window size 5 and no. of samples (a) 2 (b) 3 (c) 5.
Fig. 6.9 Result of the unsupervised texture classification of TBK-3 with window size 2 and no. of samples (a) 2, (b) 3, (c) 5.
Fig. 6.10 Result of the unsupervised texture classification of HBK 3 with window size 3 and no. of samples: (a) 2, (b) 3, (c) 5.
Fig. 6.11 Result of the unsupervised texture classification of HK with window size 5 and no. of samples 2.
Fig. 6.12 Result of the unsupervised texture classification of HKP4 with window size 2 and no. of samples (a) 2, (b) 3, (c) 5.
Fig. 6.13 Result of the unsupervised texture classification of HBR with window size 3 and no. of samples (a) 2, (b) 3, (c) 5.
(a) Window size = 5
No. of samples = 2

(b) Window size = 5
No. of samples = 3

(c) Window size = 5
No. of samples = 3

Fig. 6.44 Result of the unsupervised texture classification of TBR 1 with window size 5 and no. of samples 64. 20/4/02 10:34 PM
Fig. 2: Result of the unsupervised texture classification of HHK
with window size 2 and no of samples 6. (a) Initial.
Fig. 6.10: Result of the unsupervised texture classification of HRK with window size 3 and no. of samples 1. (a) 3x3, (b) 3x3, (c) 3x3.
Fig. 6.1 Result of the unsupervised texture classification of HHK with window size 5 and no. of samples 5; (a) 5, (b) 6, (c) 7.
Fig. 9.18 Result of the unoperated texture classification of TRK-
449P, window size 5 and no. of samples 6. (a) - (c)
Fig. 10 Results of the unaperiodic texture classification of TBK

(a) Window size = 3 and no. of samples = 2.

(b) Window size = 3 and no. of samples = 3.

(c) Window size = 3 and no. of samples = 5.

(d)
Fig. 1: Result of the unsupervised texture classification of H8K 6 with window sizes 3 and no. of samples 2 at (a), 3 at (b), and 5 at (c).
Fig. 6.21 Result of the uncorrected texture classification of TOE.
(a) Window size 2 and no. of samples 2, 5, and 10.
The results of the unsupervised texture classification of TCU with window size 3 and number of samples (a) 2, (b) 4, (c) 6.
Fig. 23: Result of the unsupervised texture classification of TCI with window size 5 and no. of samples (a): 2, (b): 3, (c): 5.
FIG. 6.23 Result of the unsupervised texture classification of FC1
2 with window size 2 and no. of samples 6. (a) = (b) = (c) = (d) = (e) = (f) = (g) = (h) = (i) = (j) = (k) = (l) = (m) = (n) = (o) = (p) = (q) = (r) = (s) = (t) = (u) = (v) = (w) = (x) = (y) = (z) = (A) = (B) = (C) = (D) = (E) = (F) = (G) = (H) = (I) = (J) = (K) = (L) = (M) = (N) = (O) = (P) = (Q) = (R) = (S) = (T) = (U) = (V) = (W) = (X) = (Y) = (Z) = (\text{ABC}) = (\text{DEF}) = (\text{GHI}) = (\text{JKL}) = (\text{MNO}) = (\text{PQRS}) = (\text{TU}) = (\text{VW}) = (\text{XYZ}) = (123) = (456) = (789) = (012) = (345) = (678) = (901) = (234) = (567) = (890)
Fig. 6.23: Results of the unsupervised texture classification of LIDAR data at a window size of 3 and no. of samples. (a) and (b) show the different classes.
Figure 26 Result of the unsupervised texture classification of UCI 2 with window size 5 and no. of samples (a) 2 (b) 5 (c) 5.
Fig. 6.2(c): Result of the unsupervised texture classification of HUT with window size 2 and no. of samples on 2 - (b) = 3 - (c) = 5.
Fig. 6.28 Result of the unsupervised texture classification of 101
35 with window size 3 and no. of samples 3 in (a) the 3rd class.
Fig. 6.29 Result of the unsupervised texture classification of BFL-5 with window size 5 and no. of samples on 2 (a) 3 (b) 5 (c) 5
Fig. 6.30 Result of the unsupervised texture classification of 1CL-4 with window size 2 and no. of samples (a) 2 (b) 5.
Fig. 9. (a) Result of the unsupervised texture classification of TCL-4 with window size 5 and no. of samples 2.

(b) Result of the unsupervised texture classification of TCL-4 with window size 5 and no. of samples 3.

(c) Result of the unsupervised texture classification of TCL-4 with window size 5 and no. of samples 5.
Figure 6.1: Result of the unsupervised texture classification of TCI with window size 2 and no. of samples 6, (a) \( \text{Window Size} = 2 \text{, No. of Samples} = 2 \), (b) \( \text{Window Size} = 2 \text{, No. of Samples} = 3 \), (c) \( \text{Window Size} = 2 \text{, No. of Samples} = 5 \).
Fig. 6.4 Result of the unsupervised texture classification of TEI 5 with window size 3 and no. of samples 2, 3, and 5.
Fig. 8: Results of the unsupervised texture classification of UCId with window size 5 and no. of samples at 2, 3, and 5.

WINDOW SIZE = 5
NO. OF SAMPLES = 2

WINDOW SIZE = 5
NO. OF SAMPLES = 3

WINDOW SIZE = 5
NO. OF SAMPLES = 5
Fig. 6. Results of the unsupervised texture classification of ICI 6 with window size 2 and no. of samples for 2-10% 10 for 5.
Fig. 6.5: Result of the unsupervised texture classification of TLC with window size 3 and no. of samples 20 in (a).
Fig. 5. Results of the unsupervised texture classification of ICI95 with window size 5 and no. of samples 5.
6.4 SUMMARY

The tree bark and cloth textures have been classified using the above algorithm. To identify classes, colors are given to texture images. The proposed unsupervised segmentation algorithm is experimented with window size 2, 3 and 5 and with No. of samples 2, 3, 5. The segmented tree bark textures and cloth textures are shown from Figures 6.3 to 6.20 and 6.21 to 6.38 respectively.

It is clearly evident from the unsupervised segmented texture images that the above classification rates are calculated over all the pixels including the regions near the boundaries of the texture images. At these boundaries, the scanning window crosses two kinds of textures, resulting in a mixed spectrum, thus giving a lower accuracy of classification.

The promising result obtained shows the success of both the texture spectrum and the classification algorithm. Requiring a minimum input of initial parameters, this unsupervised clustering algorithm could easily be used in practice for processing a large number of data. As for the choice of the number of classes (K), it might be based upon the analyst's knowledge of the scene, or upon the user's requirements that the classification displays a certain number of classes.

From the above it is clear that this method requires minimum prior information on the final classes (the number of classes k). The algorithm will create automatically the necessary centers of classes according to the structure of the texture image data.

From the figures of the proposed unsupervised segmentation algorithm it is clearly evident that 95% of the textures shows a précised
segmentation with good number of regions as the number of samples decreased and window size increases. The proposed unsupervised segmentation algorithm is flexible because one can change the window size and number of classes.