CHAPTER 3

TEXTURE CLASSIFICATION BASED ON CENTRE SYMMETRIC FUZZY TEXTURE UNIT MATRIX

3.1 BRIEF OUTLINE

The present chapter combined the features of *'Centre Symmetric Fuzzy Texture Unit Matrix'* (CSFTUM) and GLCM and derived a new matrix called CSFTU-CM for texture classification. The proposed CSFTU-CM reduces the size of the TU matrix from 6561 to 67 in the case of original texture spectrum and 2020 to 67 in the case of *'Fuzzy Texture Spectrum'* (FTS) approach. Thus, it reduces the overall complexity. The co-occurrence features extracted from the CSFTU-CM provides complete texture information about an image.

3.2 FUZZY TEXTURE SPECTRUM (FTS) APPROACH

Recently, fuzzy based methods have been used in image/texture analysis and in image segmentation [2, 8, 66, 111, 112]. In original texture spectrum approach [23]; it contains a maximum of 6561 texture units. But a major inconvenient of this descriptor is the large range of its possible values (there are 6561) at the same time that these values are not correlated. Moreover, as images of the same underlying texture can vary significantly, textural features must be invariant to (large) image variations, and at the same time sensitive to intrinsic spatial structures that define textures. A possible solution to aforementioned problems should be the use of fuzzy logic and fuzzy techniques. Fuzzy logic is a simple and powerful tool to mathematical formalization of ill defined concepts that usually appear at texture analysis and segmentation. On the other hand, fuzzy techniques enable more flexible classifications because elements can have a characteristic only to some degree.

Taur et al. [93] proposed a texture feature based on the fuzzified relative gray levels between pixels. The simplest unit representing a pixel is a 3-vector called *'Texture Vector'* (TV). An image block is represented by the histogram of its TV, and its texture is then characterized using a neural network classifier. Later on, Aina Barcelo et al [2] used the concept of *'Fuzzy Texture Spectrum'* (FTS), to be used as the texture feature within texture analysis process.

To reduce the number of texture units and to have high discriminating power, two more membership functions (Greater and Lesser quantities) are introduced in FTS approach. The aim of fuzzy texture unit is to extract local texture information from pixels for representing the texture accurately. To deal accurately with the regions of natural images even in the presence of noise and the different processes of caption and digitization FTS is introduced. For example, even if the human eye perceives two neighboring pixels as equal, they rarely have exactly the same intensity values. However, the desirable situation would be that the TU of homogeneous images contain more number of ones because the human eye can perceive ones. Therefore, if there is lack of ones, the basic TU will take only 0 and 2 values, which means that the real number of possible textures is 2^8, i.e., 256 out of

6561, and the spectrum will never be totally covered, which misuses the power of TS method.

To overcome the above, the fuzzy membership function is represented as shown in Figure 3.1 [112]. A texture unit is represented by eight elements each of which has only five possible values {0, 1, 2, 3 and 4} obtained from a neighborhood of 3×3 image region. The elements are ordered in clockwise around the centre pixel as shown in Figure 3.1. The following Equation (3.1) is used to determine the elements, E_i of the texture unit. In FTU [111], the texture unit is reduced to 2020, so that the computation time is very less when compared to basic approach.

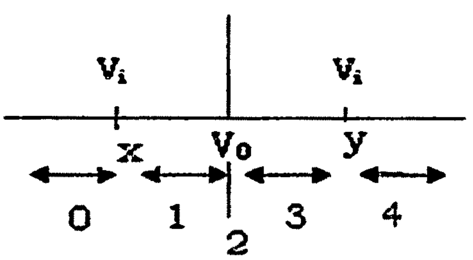


Figure 3.1: Fuzzy texture number (Base-5) representation.

$$E_i = \begin{cases} 0 & \text{if} \quad V_i < V_0 \quad \text{and} \quad V_i < x \\ 1 & \text{if} \quad V_i < V_0 \quad \text{and} \quad V_i > V_x \\ 2 & \text{if} \quad V_i = V_0 \\ 3 & \text{if} \quad V_i > V_0 \quad \text{and} \quad V_i > y \\ 4 & \text{if} \quad V_i > V_0 \quad \text{and} \quad V_i < y \end{cases} \quad \text{for } i = 1,2,3,\dots,8 \qquad (3.1)$$

where x, y are the user-specified values.

The FTU number (FTU_n) is computed in Base-5 as given in Equation (3.2):

$$FTU_{n5} = \sum_{i=1}^{8} E_i * 5^{(i-1)/2} \qquad (3.2)$$

The FTU numbers range from 0 to 2020.

For example, the process of evaluating FTU from a subimage of 3×3 is shown in Figure 3.2.

90	130	145
160	140	200
100	140	250

0	1	3
4		4
0	2	4

FTU= {0, 1, 3, 4, 4, 2, 0, 4}, FTU_{n5} = 1292

Figure 3.2: (a) Original subimage (b) Representation of fuzzy texture elements (c) Evaluate FTU.

Fuzzy Texture Spectrum (FTS) is termed as the frequency distribution of all fuzzy texture units, with the abscissa indicating the texture unit number and the ordinate representing its occurrence frequency. FTS means extraction of local texture information from pixels for describing the textural aspect of a digital image. FTS method is good to detect the classes having highest relevance for deciding the existence of a concrete textural feature within an image.

3.3 PROPOSED CENTRE SYMMETRIC FUZZY TEXTURE UNIT MATRIX (CSFTUM) METHOD

Both texture and fuzzy texture spectrum method of texture analysis gives the texture information using eight neighbouring pixels around the central pixel. The level of this information depends on ordering of the neighbouring pixels. The GLCM method gives reasonable texture information of an image that can be obtained from two pixels. Further, a little work has been reported in literature to produce strong texture information of an image by separating the neighbouring pixels into groups and to form a relationship among them. In the cross diagonal approach [4], texture information of the image is evaluated by separating the neighbourhood pixels into diagonal and corner pixels. The corner pixels are not connected pixels. The cross diagonal approach is evaluated with original texture unit but not with the FTU information. To overcome these, the present thesis derived a new matrix called *'Centre Symmetric Fuzzy Texture Unit Matrix'* (CSFTUM) for an efficient and accurate texture analysis with less time complexity. This feature is very much needed for image mining applications. The proposed matrix divides the fuzzy texture information of an image by separating the neighbouring pixels into two well connected equal groups containing four pixels based on the centre symmetric principle with central pixel of a 3×3 neighbourhood.

3.3.1 Derivation of Centre Symmetric Fuzzy Texture Unit Co-occurrence Matrix (CSFTU-CM)

The CSFTUM approach considers a set of four connected texture elements on a 3×3 grid for evaluating the FTU instead of non-connected and corner texture elements as in the case of *'Cross Diagonal Texture Matrix'* (CDTM), and *'Left Right Texture Unit Matrix'* (LRTUM). The CSFTUM method divides the 3×3 matrix in to, two well connected groups of four pixels called *'Top CSFTUM'* (TCSFTUM) and *'Bottom CSFTUM'* (BCSFTUM). These two groups of four pixels are symmetric with respect to central pixel as shown below. The TCSFTUM consists horizontal (H_1), right diagonal (RD_1) vertical (V_1), and left diagonal (LD_1) pixel elements. These TCSFTUM pixels form an angle 0, 45, 90 and 135 degrees with centre pixel as shown in Figure 3.3(b). The BCSFTUM consists of horizontal (H_2), left diagonal (LD_2) vertical (V_2), and right diagonal (RD_2) pixel elements as shown in Figure 3.3(c). These BCSFTUM pixels H_2, LD_2, V_2 and RD_2 form an angle 180, 225, 270 and 325 degrees respectively with centre pixel.

The present study considers TCSFTUM of a 3×3 neighborhood instead of 8 elements as in the case of TUM, FTUM and Cross diagonal matrix.

LD_1	E_1	RD_1
H_2		H_1
LD_2	V_2	RD_2

(a)

LD_1	E_1	RD_1
		H_1

(b)

H_2		
LD_2	V_2	RD_2

(c)

Figure 3.3:a) Representation of CSFTUM b)TCSFTUM c)BCSFTUM.

Equation (3.3) derives the elements, E_i of the texture unit. This method further reduces the FTU from 2020 to 67 i.e., CSFTUM (that is either TCFTUM or BCFTUM) values range from 0 to 66. This reduction is useful for formation of an efficient GLCM based on TU, for a good classification by reducing computational complexity.

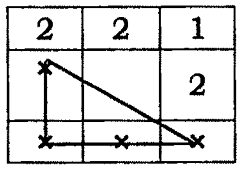
Any CSFTUM number (FTU_n) is computed in Base-5 as given in Equation (3.3):

$$CSFTU_{n5} = \sum_{i=1}^{4} E_i * 5^{(i-1)/2} \qquad (3.3)$$

Any CSFTUM number (that is either TCFTUM or BTCFTUM) range from 0 to 66. An example of TCSFTUM with eight neighbors is shown in Figure 3.4.

90	95	110
97	102	115
100	105	134

(a)



(b)

TCSFTU = {2, 2, 1, 2},
TCSFTUn5 = 32

(c)

Figure 3.4: (a) Original sub image (b) Representation of TCSFTUM elements (c) Evaluate TCSFTU.

Instead of comparing the all eight neighboring elements with the fuzzy rule, the proposed CSFTUM compares only with the four elements, as given in Figure 3.4. This method reduces the number of comparisons to half for the same number of neighbors (N=8). Compared to the original GLCM, the histogram dimension of the CSFTUM is greatly reduced.

The proposed method converts TCSFTUM into TCSFTU-CM. The derived TCSFTU-CM will be having a dimension of size 67×67. On TCSFTU-CM the Haralick features such as energy, entropy, contrast, local homogeneity, correlation, cluster shade and cluster prominence are evaluated as specified in the Equations (2.2) to (2.8) respectively.

This new method combines the merits of both GLCM and TCSFTUM of the texture analysis and gives complete texture information about an image. The proposed TCSFTU-CM reduces the computational time complexity, because of the reduced size of the TCSFTUM from 6561 to 67 as in the case of TU and 2020 to 67 as in the case of FTS. The entire process is furnished below in algorithm 3.1.

Based on the derived TCSFTU-CM the present study derives an algorithm for the efficient classification of textures. The algorithm is given below.

Algorithm 3.1: Proposed method of Efficient Classification of images based on the derived TCSFTU-CM features

Begin

1. Take the original textures T_k, O_k, $k= 1:20$
2. Subdivide the T_k, into 16 equal sized blocks. Name them as sub image T_kS_i, $k=1:20$ and $i=1:16$. They are used as sample textures for testing. For classification a LOOM classifier is used as discussed in Section 3.4.
3. Subdivide the O_k, into 4 equal sized blocks. Name them as sub image O_kS_i, $k=1: 20$ and $i = 1:4$.
4. Select at random, a training sample sub image from each O_k, $k= 1: 20$ and denote it as O_kS_j where 'j' is any of the sample pieces 1 to 4 of a particular O_k.
5. Evaluate TCSFTUM.
6. On step (5), evaluate TCSFTU-CM by moving the 3×3 matrix across the sample with overlapping (convolving) for each of the four connected neighbors.
7. Obtain Haralick features on TCSFTU-CM in four directions.
8. To classify a sample image T_kS_j, absolute difference $D(k)$ is calculated from Equation (3.4) where $D(k)$, denotes the absolute difference between TCSFTU-CM of testing and training set sample

images. The tested set falls into the *Class k, k= 1:20*, such that *D (k)* is minimum among all the *D (k), k=1: 20.*

9. Now for each texture T_k, $k=1:20$, evaluate the classification gain (G) as given in Equation (3.5) and list the output in the form of table.

End

The feature vector derived from the unknown image is compared with the feature vectors in the database using the distance vector formula, given Equation (3.4).

$$D(i) = \sum_{j=0}^{N} [f_j(k) - f_j(i)] \tag{3.4}$$

where N is the total number of features in f, i = 1 to Q (Q is the number of images in the database), $f_j(k)$ represents the jth feature of unknown texture image (k) and $f_j(i)$ represents the jth feature of texture belonging to i^{th} texture. In classification, the unknown texture is assigned to n^{th} texture image if $D(n) < D(i)$ for all i; $i \neq n$.

3.3.2 Leave One Out Method Classifier Technique (LOOM)

For the classification aspect training set is needed. In most scenarios, a training set is comprised of half of the entire database. LOOM consists of leaving one image from the database "out", and using all the other samples for training. After the classifier has been changed, the left out image is classified by the algorithm. The process is iterated and each image of the database is left out once. This permits the computation of the classification accuracy for the entire database. The success of classification is measured using the classification gain (G) and is calculated using Equation (3.5).

$$G(\%) = \frac{C_{corr}}{M} \times 100\% \tag{3.5}$$

where C_{corr} is the number of sub-images correctly classified and M is the total number of sub images, derived from each texture image.

3.4 EXPERIMENTAL ANALYSIS

The proposed TCSFTU-CM approach is tested with a set of two databases one from the OuTex and the other from Granite texture database as given in Chapter 1 of Figure 1.2 and Figure 1.3 respectively. In Dataset 1, each image is subdivided into 16 non-overlapping sub-images, giving a database of 720 texture samples (16 for each class) and in Dataset 2, the texture images are subdivided into smaller non-overlapping texture patches.

To evaluate the influence of the patch size on the classification procedures, three different block sizes are considered in the present approach, namely, 64×64, 32×32, and 16×16. The various Haralick features are averaged along horizontal, vertical and diagonal directions. Given all features, the feature selection method is performed. For classification, LOOM classifier is used to guarantee strict separation of test and training set with the maximization of number of training images.

Tables 3.1 and 3.2 summarize the results obtained for each classification procedure, using the three different patch sizes. From the results, it is observed that the classification accuracy increases as the sample size increases.

Table 3.1: mean (%) classification of proposed TCSFTU-CM algorithm on OuTex database with the dataset-1 of different block sizes.

S.No	Texture Name	16×16	32×32	64×64
1	Canvas-005	95	96.7	96.9
2	Canvas-021	90.2	92.7	95.8
3	Carpet-005	90.5	92.8	95.8
4	Granite-001	91.5	91.7	95.8
5	Granite-003	91.1	97.5	97.5
6	Granite-004	91.7	91.7	95.8
7	Granite-005	91.7	95.8	91.7
8	Granite-006	91.7	97.5	92.7
9	Granite-007	83.3	91.7	97.4
10	Granite-008	79.2	91.7	97.7
11	Granite-009	83.3	89.7	99.2
12	Granite-010	91.7	91.7	95.8
13	Paper-006	95.8	93.3	96.5
14	Plastic-001	87.5	91.7	94.2
15	Plastic-002	90.8	91.7	98.2
16	Plastic-003	91.7	91.7	91.7
17	Plastic-004	91.8	91.7	93.8
18	Plastic-005	95.8	97.5	96.7
19	Plastic-009	87.5	91.5	97.5
20	Plastic-016	87.5	91.7	92.6
21	Plastic-017	91.7	91.7	96.7
22	Plastic-018	95.8	91.7	96.8
23	Plastic-019	92.5	93.5	96.4
24	Plastic-020	81.5	91.5	95.4
25	Plastic-021	91.1	93.4	97.6

Table 3.1: (continued) mean (%) classification of proposed TCSFTU-CM algorithm on OuTex database with the dataset-1 of different block sizes.

26	Plastic-022	90.6	90.3	97.6
27	Plastic-023	89.9	90.6	97.4
28	Plastic-024	88	95.1	96.3
29	Plastic-025	90.4	96.2	97.1
30	Plastic-026	92.6	94.9	98.2
31	Plastic-027	92.1	97.4	96.9
32	Plastic-028	87.8	96.7	95.5
33	Plastic-029	91.8	91.8	97.6
34	Plastic-030	89.7	90.3	97.6
35	Plastic-031	90.6	94.2	96.9
36	Plastic-032	88.4	97.4	97.5
37	Plastic-033	91.1	95.9	96.2
38	Plastic-034	88.8	91.9	98.2
39	Plastic-035	92.7	92.7	96.8
40	Plastic-036	91.4	97.2	99.6
41	Plastic-038	97.7	97.6	99.7
42	Plastic-040	91.5	94.5	97.1
43	Plastic-041	91.7	93.6	96.5
44	Wood-006	91.7	91.7	94.7
45	Wood-008	93.2	92.4	98.2
	Average	90.5	93.5	96.5

Table 3.2: mean (%) classification of proposed TCSFTU-CM algorithm on Granite database with the dataset-2 of different block sizes.

S.No	Texture Name	16×16	32×32	64×64
1	Acquamarina-1	92.1	97.4	96.9
2	Acquamarina-2	87.8	89.7	95.5
3	Acquamarina-3	91.8	93.8	97.6
4	Acquamarina-4	89.7	90.3	97.6
5	Azul Capixaba-1	90.6	96.2	96.9
6	Azul Capixaba-2	88.4	97.4	97.5
7	Azul Capixaba-3	91.7	95.9	96.2
8	Azul Capixaba-4	88.8	91.9	98.2
9	Bianco Cristal-1	92.7	92.7	96.8
10	Bianco Cristal-2	97.4	97.2	99.6
11	Bianco Cristal-3	91.7	91.7	95.8
12	Bianco Cristal-4	95.8	97.3	98.5
13	Bianco Sardo-1	90.5	91.7	94.2
14	Bianco Sardo-2	90.8	91.7	98.2
15	Bianco Sardo-3	91.7	93.7	91.7
16	Bianco Sardo-4	91.8	94.7	95.8
17	Rosa Beta-1	95.8	95.5	96.7
18	Rosa Beta-2	87.5	91.5	97.5
19	Rosa Beta-3	87.5	91.7	94.6
20	Rosa Beta-4	91.7	93.6	96.5
21	Azul Platino-1	91.7	91.7	94.7
22	Azul Platino-2	93.2	94.4	98.2
23	Azul Platino-3	90.7	92.2	96.5
24	Azul Platino-4	92.4	95.6	96.67
25	Giallo Ornamentale-1	92.1	97.4	96.9

Table 3.2: (Continued) mean (%) classification of proposed TCSFTU-CM algorithm on Granite database with the dataset-2 of different block sizes.

26	Giallo Ornamentale-2	91.8	96.7	95.5
27	Giallo Ornamentale-3	91.8	92.8	97.6
28	Giallo Ornamentale-4	89.7	90.3	97.6
29	Giallo Napoletano-1	90.6	96.2	96.9
30	Giallo Napoletano-2	88.4	97.4	97.5
31	Giallo Napoletano-3	91.1	95.9	96.2
32	Giallo Napoletano-4	88.8	91.9	98.2
33	Giallo Santa Cecilia-1	92.7	92.7	96.8
34	Giallo Santa Cecilia-2	90.4	97.2	99.6
35	Giallo Santa Cecilia-3	94.7	96.6	99.7
36	Giallo Santa Cecilia-4	83.3	91.7	97.4
37	Giallo Veneziano-1	83.3	91.7	97.4
38	Giallo Veneziano-2	89.2	91.7	97.7
39	Giallo Veneziano-3	89.3	92.7	99.2
40	Giallo Veneziano-4	91.7	91.7	95.8
41	Rosa Porri˜no A-1	92.8	93.3	96.5
42	Rosa Porri˜no A-2	87.5	91.7	94.2
43	Rosa Porri˜no A-3	90.8	91.7	98.2
44	Rosa Porri˜no A-4	91.7	91.7	94.7
45	Rosa Porri˜no B-1	91.8	91.7	93.8
46	Rosa Porri˜no B-2	95.8	94.5	96.7
47	Rosa Porri˜no B-3	95.8	93.5	96.7
48	Rosa Porri˜no B-4	87.5	91.5	97.5
	Average	91	93.6	96.8

3.4.1 Comparison of Proposed TCSFTU-CM with Existing Methods

The proposed TCSFTU-CM is compared with the recent classification methods CDTM [4], Modified CDTM [5] and LRTUM [89]. Table 3.3 show the mean percentage classification rate of the proposed TCSFTU-CM and existing methods. The graphical analysis of this is shown in Figure 3.5. Table 3.3 and Figure 3.5 clearly indicate that, the proposed TCSFTU-CM exhibits a high classification rate than the existing methods.

Table 3.3: Comparison of the proposed TCSFTU-CM method with other existing methods.

Texture Group	CDTM	Modified CDTM	LRTUM	Proposed TCSFTU-CM
OuTex	94.6	94.7	95.7	96.5
Granite	92.5	93.5	94.5	96.8
VisTex	91.8	93.5	92.0	94.4
Brodatz	91.3	93.5	95.0	95.7
Average (%) of Classification	91.5	93.8	94.3	95.8

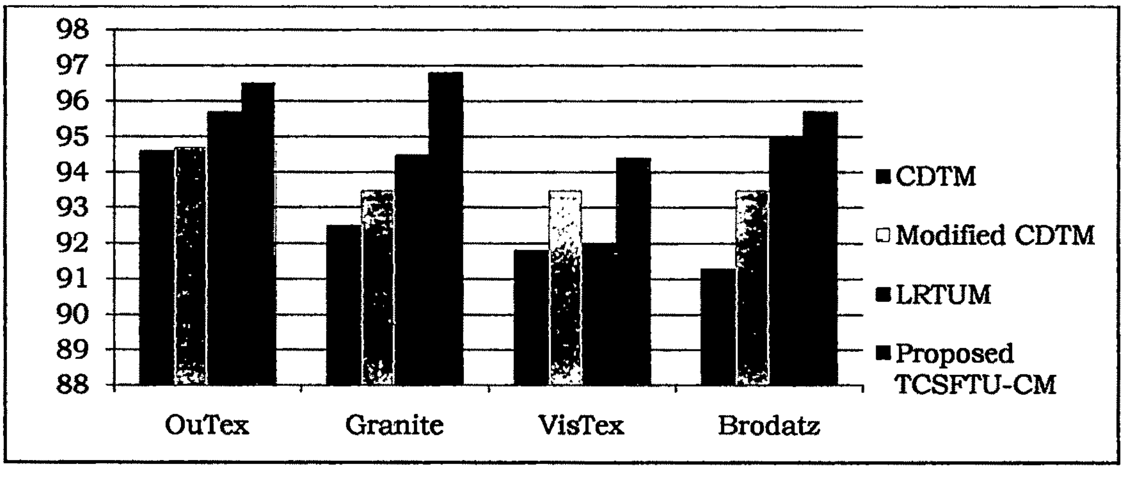


Figure 3.5: Comparative analysis of proposed TCSFTU-CM with existing methods.

SUMMARY

The proposed TCSFTU-CM reduces the size of texture unit matrix from 6561 to 67 as in the case of OTS and 2020 to 67 as in the case of FTS approach. The feature extraction process of the proposed TCSFTU-CM is quite efficient with less complexity. When compared with other approaches, the proposed TCSFTU-CM scheme is more effective, exhibiting increased classification ability by using smaller feature vectors. The experimental results based on different images show that the implemented TCSFTU-CM classification scheme is quite robust to noise and it is more efficient than the existing methods.