CHAPTER 4

TEXTURE CLASSIFICATION

4.1 INTRODUCTION

The presence of texture has been represented locally by the texture primitives and globally by the primitive spectrum. Various images have been included for the tests and experiments and the corresponding primitive spectrums have been presented. The properties of the primitive spectrum for various noise levels, window size variations of images and rotation features have been effectively studied in the previous chapter. Also, using the set of texture primitives and placement rules, a number of texture images have been generated. The usage of the proposed primitive spectrums shown for texture classification is discussed in this chapter. Both supervised and unsupervised texture classification has been attempted and very good results have been obtained. Standard texture images from the Brodatz and VisTex textural album have been included for the classification experiments. The results have been compared with the classification results obtained with a texture number scheme as illustrated in Chapter 3.

The success of any texture analysis scheme is inferred from the way it captures the textural features and their uniqueness. The textural properties of texture images carry more useful information for discrimination purposes; it is important to develop features for texture, based on suitable quantitative representation of some of these properties. Since the frequency of the occurrences of such primitives is computed for the entire image, the representation becomes a global descriptor.

Our proposed method has been successful in claiming the advantages in that, (i) the number of texture features are less and (ii) the time complexity of the proposed method for classification is lesser compared to the texture number scheme, as the number of features is lesser.

In the subsequent sections, the texture classification scheme has been explained and the classification confusion matrix has been obtained in the case of supervised classification. The problem of unsupervised classification has also been done with many created target images. Finally, the conclusion of our approach and its application for texture classification has been highlighted, along with the further scope of this work.

In applications such as scene analysis and remote sensing, it has long been recognized that texture analysis plays a fundamental role in establishing and classifying objects and regions. Real world images like landsat images are known to contain different textured regions. Classification of these textured regions is vital for the analysis and interpretation of images. The major approaches for texture classification include statistical, structural, model-based and transform methods, which have already been explained in Chapter 2 under literature survey.

Recently, Manian and Vasquez (2003) took into account the dynamics of visual perception of textures and emphasized the importance of contextual information of texture in the discrimination process. Their approach consists of a texture space whose components are textures with varying texture context such as regularity, randomness, directionality etc. The Gabor filters are convolved with the texture regions and from the responses, textural features like regularity etc. are extracted. These features form the components of the representational space.

Georgescu et al (2003) attempted to identify the textons, the basic structural units of textures, as the cluster centers in a feature space derived from the input. The feature space they build from a bank of filters, employ 48 anisotropic and isotropic filters, and 13 circular symmetric filters, but has higher time complexity.

This chapter explains texture classification obtained with a database, containing texture image regions, derived from the texture images of the Brodatz texture album, and the Vistex database.

The problem in classification is assigning each possible region or a pixel in the image to a known class, i.e., partitioning the image into mutually exclusive regions, so that each region corresponds to a particular class of texture. In general, classification can be broadly done, based on two major approaches, namely, (i) supervised classification , where a priori information about the classes of images to be recognized is available and (ii) unsupervised classification, where no priori information about the classes is available.

These two approaches have been implemented using the primitive spectrum as the feature vector, and are explained in the following sections.

4.2 SUPERVISED CLASSIFICATION

Any texture classification system will have basically two stages, namely, training and classification.

The steps involved in texture training and texture classification are shown in Figures 4.1 and 4.2 respectively.

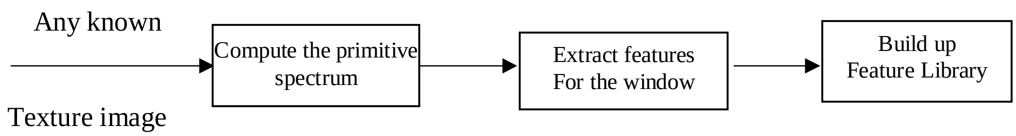


Figure 4.1 Texture Training Stage

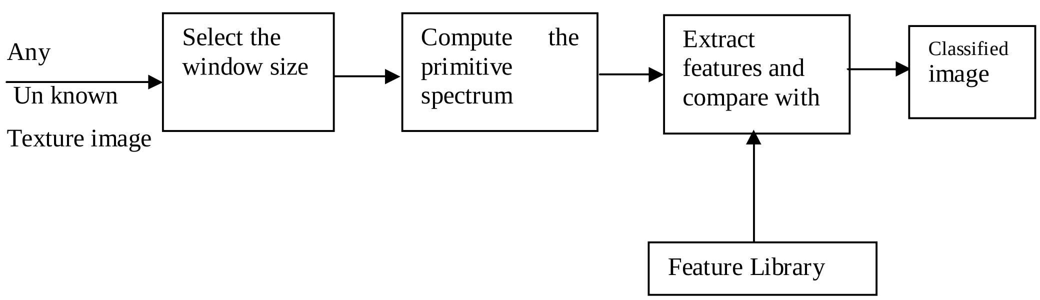


Figure 4.2 Texture Classification Stage

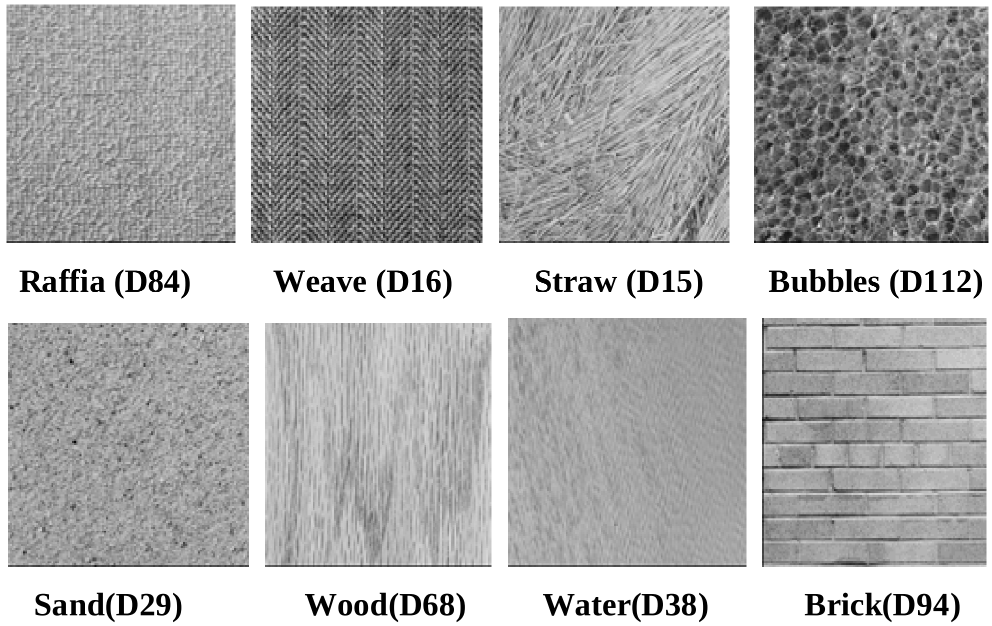
The feature library will be present for the supervised classification case whereas it will not be present for the unsupervised one. Correspondingly, the algorithm will be slightly different.

4.2.1 Texture Training Stage

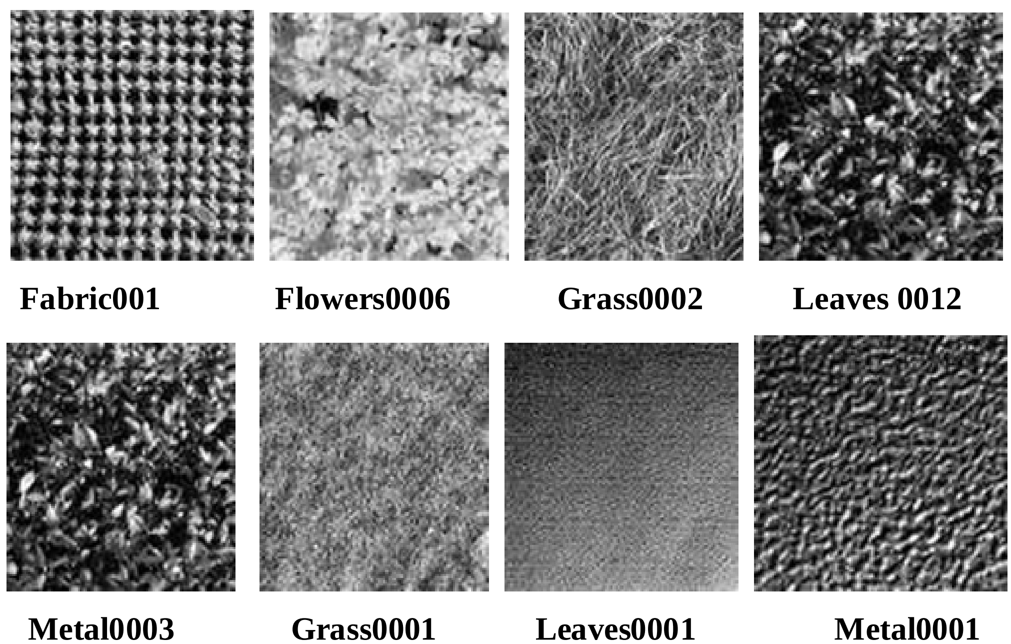
In the texture training stage, the primitive spectrums for known texture images collected from the Brodatz textural album and the VisTex Album have been obtained. The primitive spectrums have been computed for a window size of 30 x 30 image size using the procedure presented in Section 3.3.4. Since we have considered texture primitives of size 3 x 3, it is suggested to have the window size at least ten times, for representing the features of the image. The frequency of occurrences of each primitive for a set tolerance level is calculated as features and labeled in the feature library. The feature library, thus, will have the sample primitive spectrums for all the possible texture images likely to be included for the classification experiment. These features stored in the library are further used in the texture classification phase. It is proposed to use the images shown in Figure 4.3 for the classification experiment. Hence the primitive spectrums are obtained for these images for a specific window size of 30 x 30. The complete algorithm for creating the texture training is presented below.

Algorithm

Input	:	Texture images collected from the Brodatz and the VisTex album, say n images.
Output	:	Feature vectors, one for each texture image considered.
Step 1	:	Consider any 30 x 30 image region, from the first image.
Step 2	:	Compute the primitive spectrum as discussed in Section 3.3.4 for a set tolerance, for example 20.
Step 3	:	Repeat step 2 and obtain at least 10 to 15 primitive spectrums considering the window randomly over the entire image.
Step 4	:	Compute the average of all the features i.e. the primitive spectrums.
Step 5	:	Store the primitive spectrum PS in the Feature Library, with proper labeling.
Step 6	:	Repeat from step 1 to step 5 for all the known texture images
Step 7	:	If no more image exists, then close the training stage and the feature library will have n feature vectors, each corresponding to the image considered.



(a) Brodatz Images



(b) Vistex Images

Figure 4.3 Standard Texture Images used in the Classification Experiment (a) Brodatz Images (b) Vistex Images

Since the accuracy of the entire classification experiment depends on the training, utmost care has been paid in creating the training feature vector. In any natural texture image, the presence of texture varies slightly within the texture image, as is evident from the images shown in Figure 4.3. Hence the feature vector is generated by computing the average of 10 to 15 primitive spectrums obtained for the image regions collected randomly from each image. Thus the training stage will have the feature vectors of all the possible images included for the classification experiment.

4.2.2 Texture Classification Stage

In the second stage, namely, in the classification stage, the texture classification is done by computing the integrated absolute difference between this spectrum and all the spectrums available in the feature library, created using the procedure given in the previous section. The unknown texture is assigned to one of the known classes for which the difference is zero or closer to the present value.

The simple minimum distance decision criterion has been used to classify the input feature vector X among the m number of reference or training feature vectors R_1,R_2,..Rm with R_j associated with pattern class w_j. The image region that corresponds to the input feature vector X is assigned to w_i if □X-Ri□ is the minimum, where □X-Ri□ is the distance between X and Ri.

The distance has been defined here as follows,

$$| X\text{-}Ri | = [(X\text{-}Ri)^t (X\text{-}Ri)]^{1/2} \qquad (4.1)$$

As the proposed texture descriptor, the primitive spectrum contains important textural information which has been used as a feature vector during classification. The reference feature vector Ri, $1 \le i \le m$, is considered here as the primitive spectrum of the texture class w_i. The input feature vector has been computed for each pixel in the target image as follows. A window of larger width (say 30X30 size) is slid over the target image. For each (30X30) image region selected randomly the primitive spectrum is computed. This primitive spectrum is subjected to the afore-mentioned minimum distance classifier as the input feature vector X .The texture class w_j, that is assigned by the classifier to the input vector X is assigned here to the region which is under analysis. The complete algorithm for performing supervised classification is presented below.

Algorithm

Input : Any unknown texture region

Output : Class identified

Step 1 : Consider the input image in the size of 30 x 30

Step 2 : Obtain the primitive spectrum at the same tolerance level

Step 3 : Compute the integrated absolute difference between this primitive spectrum and the reference vectors available in the feature library as

$$Diff\,(i) = \sum_{i=1}^{n} \left| f_i(x) - T(x) \right|$$

where $f_i(x)$ represents the primitive spectrum for the i th reference image and T(x) represents the primitive spectrum for the target image and x represents the feature value.

Step 4 : Compare the values of all Diff(i) for zero. Assign the target image to the class i if Diff(i) is zero or minimum.

Step 5 : Unknown image region is classified into a known class i.

Step 6 : If any more region is to be classified, go to step 1.

Step 7 : For each input image region considered, if it is correctly classified then the correct classification is 100 %. Practically, few samples may get misclassified into another class. Hence the classification efficiency is computed.

Step 8 : End

4.2.3 Experimentation and Results

In the supervised classification experiment 16 images are considered, 8 from each data base and are shown in Figure 4.3. Keeping the tolerance level at 20, the corresponding primitive spectrums are obtained and form the reference feature vector. Each 30 * 30 region considered from any target image, is subjected to classification. Overall, 1200 sample windows have been obtained randomly from these images and are subjected to the classification. Correct classification and misclassification have been tabulated as the confusion matrix in Table 4.1 and Table 4.2. This leads to an average correct classification of upto 95%. The experimentation gets repeated for different tolerance levels both for the Brodatz, and the Vistex images and this variation has been shown in Figure 4.4. It is quite evident from the graphs that as the tolerance level gets relaxed the classification efficiency is improved. This is so because, the feature vector has more components due to the relaxed conditions during the training stage. Hence the classification tends to be more accurate. Various experiments performed with different classes of texture images using different tolerance levels are presented below in the form of tables. The table shows the input image samples considered and these samples are classified into one of the known classes. The classification experiments have been performed with a texture spectrum as proposed by He and Li Wang just for comparison with our work. An average correct classification of 84 % has been obtained.

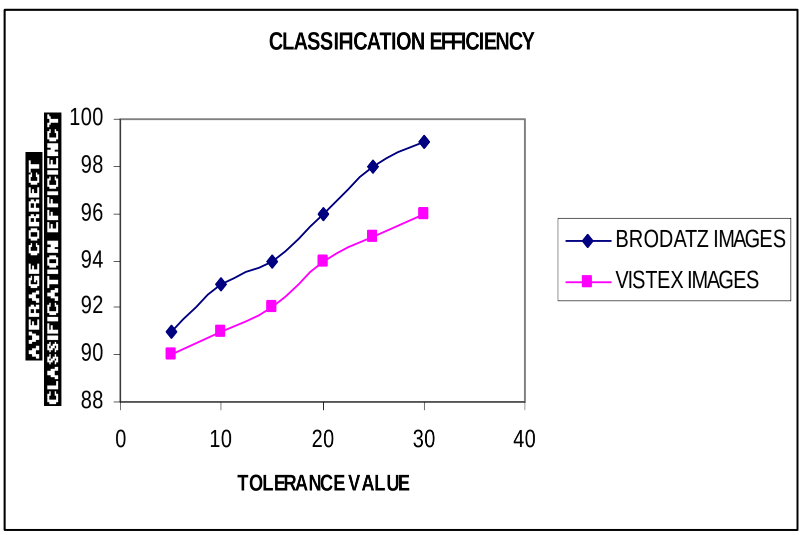


Figure 4.4 Correct Classifications with Tolerance Levels for Brodatz and Vistex Images

Table 4.1 Texture Classification Experiment Results (in Percentage) for Brodatz Images

Images	Tolerance levels	Total number of samples correctly classified								Total number of samples correctly classified %	Total number of samples misclassified	Correct Classification %
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**			
1	5	90	0	0	5	5	0	0	0	90	10	
2		0	93	7	0	0	0	0	0	93	7	
3		0	6	94	0	0	0	0	0	94	6	
4		5	0	0	92	3	0	0	0	92	8	
5		4	0	0	13	83	0	0	0	83	17	
6		0	0	0	0	6	84	10	0	84	16	
7		0	0	0	0	4	0	96	0	96	4	
8		0	0	0	0	0	4	4	92	92	8	
Average correct classification												91 %
1	10	92	0	0	4	4	0	0	0	92	8	
2		0	98	2	0	0	0	0	0	98	2	
3		0	1	99	0	0	0	0	0	99	1	
4		4	0	0	94	2	0	0	0	94	6	
5		4	0	0	12	84	0	0	0	84	16	
6		0	0	0	0	4	86	10	0	86	14	
7		0	0	0	0	0	0	100	0	100	0	
8		0	0	0	0	0	3	3	94	94	6	
Average correct classification												93 %
1	15	93	0	0	4	3	0	0	0	93	7	
2		0	98	2	0	0	0	0	0	98	2	
3		0	1	99	0	0	0	0	0	99	1	
4		3	0	0	94	3	0	0	0	94	6	
5		4	0	0	11	85	0	0	0	85	15	

Table 4.1 (Continued)

Images	Tolerance levels	Total number of samples correctly classified								Total number of samples correctly classified %	Total number of samples misclassified	Correct Classification %
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**			
6		0	0	0	0	5	90	5	0	90	10	
7		0	0	0	0	0	0	100	0	100	0	
8		0	0	0	0	0	4	2	94	94	6	
Average correct classification												94 %
1	20	94	0	0	3	3	0	0	0	94	6	
2		0	100	0	0	0	0	0	0	100	0	
3		0	0	100	0	0	0	0	0	100	0	
4		0	0	0	93	7	0	0	0	93	7	
5		4	0	0	10	86	0	0	0	86	14	
6		0	0	0	0	0	100	0	0	100	0	
7		0	0	0	0	0	0	100	0	100	0	
8		0	0	0	0	0	2	2	96	96	4	
Average correct classification												96%
1	25	95	0	0	3	2	0	0	0	95	5	
2		0	100	0	0	0	0	0	0	100	0	
3		0	0	100	0	0	0	0	0	100	0	
4		0	0	0	93	7	0	0	0	93	7	
5		3	0	0	3	94	0	0	0	94	6	
6		0	0	0	0	0	100	0	0	100	0	
7		0	0	0	0	0	0	100	0	100	0	
8		0	0	0	0	0	2	1	97	97	3	
Average correct classification												98 %
1	30	98	0	0	2	0	0	0	0	98	2	
2		0	100	0	0	0	0	0	0	100	0	
3		0	0	100	0	0	0	0	0	100	0	
4		0	0	0	98	2	0	0	0	98	2	
5		2	0	0	2	96	0	0	0	96	4	
6		0	0	0	0	0	100	0	0	100	0	
7		0	0	0	0	0	0	100	0	100	0	
8		0	0	0	0	1	1	0	98	98	2	
Average correct classification												99 %
Average correct classification for BRODATZ Images												95%

Table 4.2 Texture Classification Experiment Results (in Percentage) for VISTEX Images

Images	Tolerance levels	Total number of samples correctly classified								Total number of samples correctly classified	Total number of samples misclassified	Correct Classification %
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**			
1	5	91	0	3	0	0	0	0	6	91	9	
2		0	83	0	5	0	0	12	0	83	17	
3		0	0	87	0	3	0	10	0	87	13	
4		0	0	0	94	4	0	0	2	94	6	
5		0	2	0	3	91	0	0	4	91	9	
6		0	9	4	0	0	87	0	0	87	13	
7		0	0	4	0	3	4	89	0	89	11	
8		6	0	0	0	0	0	0	94	94	6	
Average correct classification												90 %
1	10	93	0	2	0	0	0	0	5	93	7	
2		0	90	0	4	0	0	6	0	90	10	
3		0	0	91	0	4	0	5	0	91	9	
4		0	0	0	96	2	0	0	2	96	4	
5		0	1	0	3	92	0	0	4	92	8	
6		0	8	4	0	0	88	0	0	88	12	
7		0	0	3	0	3	4	90	0	90	10	
8		2	0	0	0	0	0	0	98	98	2	
Average correct classification												92%
1	15	94	0	1	0	0	0	0	5	94	6	
2		0	92	0	3	0	0	5	0	92	8	
3		0	0	90	0	3	0	7	0	90	10	
4		0	0	0	97	3	0	0	0	97	3	
5		0	1	0	3	92	0	0	4	92	8	
6		0	7	3	0	0	90	0	0	90	10	
7		0	0	4	0	3	3	90	0	90	10	
8		2	0	0	0	0	0	0	98	98	2	

Table 4.2 (Continued)

Images	**Tolerance levels**	**Total number of samples correctly classified**								**Total number of samples correctly classified**	**Total number of samples misclassified**	**Correct Classification %**
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**			
Average correct classification												93 %
1	20	95	0	1	0	0	0	0	4	95	5	
2		0	87	0	2	0	0	11	0	87	13	
3		0	0	91	0	0	3	6	0	91	9	
4		0	0	0	100	0	0	0	0	100	0	
5		0	0	0	3	93	0	0	4	93	7	
6		0	6	2	0	0	92	0	0	92	8	
7		0	0	3	0	3	3	91	0	91	9	
8		0	0	0	0	0	0	0	100	100	0	
Average correct classification												94%
1	25	95	0	2	0	0	0	0	3	95	5	
2		0	92	0	4	0	0	4	0	92	8	
3		0	0	94	0	2	0	4	0	94	6	
4		0	0	0	100	0	0	0	0	100	0	
5		0	0	3	0	93	2	0	2	93	7	
6		0	6	0	0	0	94	0	0	94	6	
7		0	0	4	0	2	2	92	0	92	8	
8		0	0	0	0	0	0	0	100	100	0	
Average correct classification												95 %
1	30	97	0	3	0	0	0	0	0	97	3	
2		0	96	0	2	0	0	2	0	96	4	
3		0	0	96	0	0	2	2	0	96	4	
4		0	0	0	100	0	0	0	0	100	0	
5		0	0	3	0	93	2	0	2	93	7	
6		0	3	0	2	0	95	0	0	95	5	
7		0	0	0	0	2	2	96	0	96	4	
8		0	0	0	0	0	0	0	100	100	0	
Average correct classification												97 %
Average correct classification for VISTEX Images												94%

4.3 UNSUPERVISED CLASSIFICATION

The supervised texture classification experiment has been already performed with standard texture images collected from the Brodatz and the Vistex textural album. The problem of un-supervised classification is explained in this section. By texture classification is meant assigning each possible region or a pixel in the image to a class; i.e., partitioning the image into mutually exclusive regions so that each region corresponds to a particular class of texture. Any texture description scheme is successful only if it has a discriminating capability.

In the previous section, the use of the texture primitive spectrums has been shown to be effective in the case of supervised classification. The unsupervised classification scheme, algorithm and experimentation are presented in this section. By unsupervised classification is meant, that there is no apriori information about the number of texture classes present in the target images. The objective of this classification is to recognize the different texture classes. Given a target image, the algorithm proceeds under the presumption that the number of classes are, say, N. The main idea of the scheme is that the distance between one region and the adjacent region will be very small if both are from the same texture image. The distance will be more if they are from two different texture regions. Based on these distances, the target image is classified in to meaningful regions. That is, within a texture region the properties will be more or less uniform and they will be different when the regions are from two different textures. This principle is used for unsupervised classification.

As the proposed texture descriptor, the primitive spectrum contains important textural information which has been used as a feature vector during classification. The reference feature vector is considered here as the primitive spectrum of the texture class considered from the top left corner of the target image. A window of larger width (say, 30X30 size) is slid over the target image. For each 30 X 30 region selected, the primitive spectrum is computed. This primitive spectrum is subjected to a minimum distance classifier with the input feature vector computed for the previous region. The texture class, that is assigned by the classifier to the input vector is the same as that of the previous one if the differences are zero or closer to zero. This process gets repeated while considering the adjacent regions of the target image. Comparing the feature vector of the present window and the previous window, if the difference is greater than a threshold, then a new region starts to form. Finally when the entire target image is completely scanned, it gets classified in to a number of regions. Thus the proposed primitive spectrum has been successfully used for un-supervised classification. The entire process for performing unsupervised classification is presented in the form of an algorithm below.

Algorithm for Unsupervised Texture Classification

Input	:	Any texture image to be classified with size M x N where M,N >= 30.
Output	:	The regions classified in the image
Step1	:	Extract a 30 x 30 image region from the input image from the top left corner.
Step 2	:	Obtain the primitive spectrum for the window at a fixed tolerance
Step 3	:	Leaving 3 columns, obtain the adjacent window from the target image and repeat step 2
Step 4	:	Compute the integrated absolute difference between the two primitive spectrums

Step 5 : If the distance is less than a threshold, conclude both are from the same texture region. Else form another region.

Step 6 : Identified different regions are labeled as 1,2,…

Step 7 : Repeat the process till the end of the image.

Step 8 : Compute the classification accuracy based on the number of classes included for the test and how many classes get assigned. That is, if there are 100 samples included in the experiment and there are only two classes, based on the distance measures and the threshold, some of the samples may get assigned to the third or even the fourth class. Accordingly, the classification accuracy is computed.

4.3.1 Experimentation and Results

In the classification experiment four target images are considered, collected from Brodatz data bases and are shown in Figure 4.5. The target images (A to D) have 2, 4, 2 and 5 regions respectively. Keeping the tolerance level at 20, the corresponding primitive spectrums are obtained and form the initial feature vector. Subsequently, each 30 X 30 region considered from the first target image is subjected to classification by comparing the primitive spectrum and the adjoining texture primitive spectrum using the algorithm in Section 4.3. If the computed integrated absolute difference is well within the tolerance level, the adjacent region is the same as the previous region. This procedure is repeated till the end of the image is reached. For example, in the target image (A) shown in the figure, out of 100 samples considered, the integrated absolute differences have been computed and are compared with the threshold. About 56 samples fall in one category and 41 samples have been included in another region, 2 samples in another region and one sample falls into the fourth category. In short, 97 samples are correctly classified and three samples are misclassified. Similarly, for the image (B), there are four classes included in the target image. Here also, 97% of classification accuracy is obtained. Similarly, the other two images are also considered. The overall classes into which the target image gets classified, are listed in the Table 4.3. Correct classification and misclassification have been tabulated. This leads to an average correct classification of upto 96%. The experiment gets repeated at different tolerance levels both for the Brodatz, and the Vistex images. Unsupervised texture classification has been performed with the texture number scheme and an average correct classification of 80% obtained. It has been already described that the time complexity of our proposed scheme is less when compared to the texture number scheme. The same has been elaborately illustrated with time complexity metrics, in the following sub section.

4.3.2 Time Complexity Analysis

The Time complexity analysis for texture spectrum approach (He and LiWaNg's) and our proposed method is presented below.

4.3.2.1 In the case of the texture spectrum approach

The total number of features in the feature vector: 6561

For a 30 X 30 image region the number of features in the feature vector : 6561

Subtraction between any two feature vectors : $f(n_1) - f_i(n_1) \approx O(n_1)$

Where n1 is proportional to the feature vector length.

If there are k number of feature vectors in the library, the subtraction is to be performed k times, i.e., $\approx$ k $O(n_1)$

This procedure is for a single window.

If this is to be repeated for an N X N image size, the total time complexity for the classification will be = (N- 30) X (N- 30) X k X $O(n_1)$.

The overall time complexity is proportional to $\approx O(n_1)$ as N and k are assumed to be constant.

4.3.2.2 In the case of our proposed texture primitive spectrum approach

The total number of features in the feature vector: 92.

For a 30 X 30 image region, the number of features in the feature vector : 92.

Subtraction between any two feature vectors : $f(n_2)$-$f_i(n_2)$.

Where $f_i(n_2)$ represents the feature vectors in the feature library, the subtraction is to be performed k times, i.e., k.O (n_2).

This time complexity is for a single window. If this is to be repeated for an N X N image size, then the total time complexity for the entire classification will be = (N- 30) X (N-30) X k $(O(n_2))$.

Then the overall time complexities are proportional to $\approx O(n_2)$ as N and k are assumed to be constant.

Comparing the two time complexities T_1 and T_2 where,

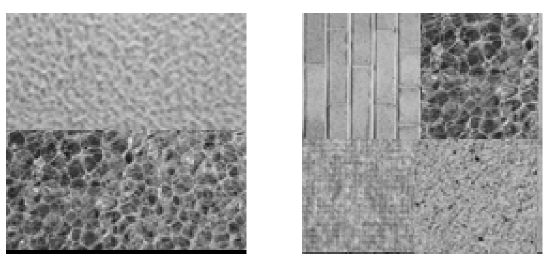
T_1 is proportional to $O(n_1)$ and

T_2 is proportional to $O(n_2)$,

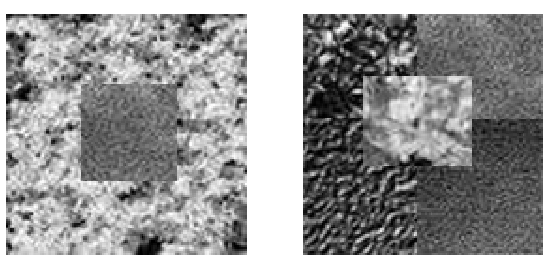
The time taken for performing classification will be based on the values of the feature vector.

As both the computation and classification experiments are carried out in the same hardware specification configuration, the computation time required (CPU time) is based purely on the feature vector length. Hence $n_1 = 6561$ and $n_2 = 92$. It is quite obvious that our algorithm, using the proposed texture primitive spectrum, takes lesser time as $n_2 < n_1$.

It is experimentally observed on a P-IV 2.66 GHz based personal computer where the algorithms are implemented.



(a) **(b)**



(c) **(d)**

Figure 4.5 Target Images (A TO D) used for Texture Unsupervised Classification

Table 4.3 Results of Unsupervised Classification

Target images as in Figure 4.5	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Misclassified samples	Correct Classification in %
A	56	41	2	1	---	---	---	---	3	97
B	24	41	14	18	2	1	---	---	3	97
C	54	41	4	1	----	----	----	---	5	95
D	19	19	18	20	19	1	2	2	5	95

4.4 SUMMARY

Supervised and unsupervised texture classification have been successfully performed and explained in this chapter. The proposed method of texture classification is based on computing the feature vector from the texture images, namely, the primitive spectrums. In the first stage, the features are computed and form the feature vector which have been used for the training. Then, based on the algorithms they are classified, and the results are presented. The experiments are performed with the Brodatz and the VisTex images. The average correct classification of about 95 % has been obtained. The results of classification have been compared with those of He and Li Wang’s texture spectrum approach. Our algorithms work better for the test cases considered. The use of the proposed texture primitive spectrum will be highlighted for texture edge detection and is illustrated in the next chapter.