CHAPTER – 6

NATURAL TEXTURE CLASSIFICATION

6.1 Introduction

Classification of a scene (whether it is an indoor scene or an outdoor one) has always remained as a challenging problem in the field of pattern recognition. This is due to the extreme variability of the scene content and the difficulty in explicitly modeling scenes with indoor and outdoor content. Such a classification has applications in content based image and video retrieval from archives, robot navigation, large scale scene content generation and representation, generic scene recognition etc. Human beings are able to classify scenes based on certain local features along with the content or association with other features. This context is learned by experience (training). Some examples of such local features are presence of trees, water bodies, exterior buildings, sky in an outdoor scene and the presence of straight lines or regular flat shaded objects or regions such as walls, windows and artificial man-made objects in an indoor scene. Also, the types of features that humans perceive from images are based on colour, texture and shape of local regions or image segments.

A texture has a spatially uniform distribution of local gray-value variations. The images with uniform gray-values, white noise images and images of objects that are solely characterized by shape do not form a texture. A texture is a region in an image that has a set of local statistics or other local properties which are either constant or varying slowly.
or approximately periodic. The local statistics or the property that is repeated over the textured region is called the texture element or texel which can be used as a visual primitive. These texels occur in different positions of a given area with certain invariant properties. The invariant properties of the texels correspond to their size and variation. The texels vary slightly in their size and orientation across the texture region.

Texture is an efficient measure to estimate the structural, orientation, roughness, smoothness or regularity differences of diverse regions in an image scene. Characterizing a real-world view or an image into different texture classes is often a trivial task for the human visual system but is one of the most challenging problems in the field of computer vision and image processing.

Several approaches for texture analysis have been proposed by researchers working in this area during the last three decades or so. Our approach for the classification of texture images is shown in Figure 6.1. It has the following two main steps: (1) Feature extraction for parametric representation of texture images and (2) Classification system for recognition of texture images.

![Block diagram of general approach for classification of texture images.](image)

**Figure 6.1: Block diagram of general approach for classification of texture images.**

The texture classification task requires an efficient method for representation of texture images and development of a recognition model. We have used a simple 2-dimensional
discrete wavelet transform (DWT) representation which captures the small differences in the rotation or scale that is desired for the current applications. The DWT is used to generate feature images from individual wavelet sub-bands. As we have already stated in recent years SVM have demonstrated good performance in a variety of pattern recognition tasks. It has been shown in the case of face recognition problem described in Chapter 5. We have proposed a texture classification approach using a 2-dimensional discrete wavelet transform and SVMs. Multiclass recognition system using SVMs are built using one-against-the-rest approach. SVM based approach for texture classification has been demonstrated using Brodatz album texture database. The description of this database and its representation are discussed in the following section.

This chapter is organized as follows: In Section 6.2, we have described the database and its representation. We have discussed feature extraction techniques in Section 6.3. In Section 6.4, we have discussed the classification model using support vector machines followed by summary and conclusion in Section 6.5.

6.2 Texture Database and Representation

Five different types of texture classes have been considered from the Brodatz album [141]. These five classes are Bark, Beachsand, Beans, Burlap, and D10. Each class has 64 images which consist of the following subsets of images: 16 original images, 16 rotated versions of the original images, 16 scaled versions of the original images and 16 rotated and scaled versions of the original images. The size of each image is 64 x 64 pixels. The image is stored in xv format (Khoros Visualization image file). Image pixels are in gray level whose values range from 0 to 255. Pixel format is 8 bit monochrome. Figure-6.2 shows the sample original images of textures. It can be seen from the images that there
exists a lot of variability among training and testing examples of the same texture class. For each texture class, out of 64 examples, 48 examples are used for training and 16 examples are used for testing. The 48 training examples of a class include 16 original images, 16 rotated versions of the original images and 16 scaled versions of the original images. The testing example of a class includes 16 rotated and scaled versions of the original images.

Texture classification is basically the problem of classifying pixels in an image according to their textural cues. This is different from conventional image segmentation as the texture is characterized using both the gray value for a given pixel and the gray-level pattern in the neighborhood surrounding the pixel. Crucial to the success of texture classification are (a) the identification of features that differentiate textures in an image and development of their representations for further classification and (b) the construction of classification paradigms that operate on the above representations and
discriminate between texture features associated with different texture classes. Accordingly, the texture classification problem is conventionally divided into the two subproblems of feature extraction and classification. Several methods have been developed to extract textural features which can be classified as statistical, model-based, and signal processing methods. In statistical approaches, textures are described using statistical measures such as the co-occurrence or autocorrelation statistics of the gray levels for k-tuples of pixels [33]. The major drawback of this type of method is the enormous amount of data involved in kth order statistics which is especially hard to handle when k is large (k > 2). This is relevant as psychophysical experiments have demonstrated that the human visual system is able to extract some statistics of an order higher than two. Model-based methods characterize texture images based on probability distributions in random fields, such as Markov chains and Markov random fields (MRFs). MRFs are widely used because they yield a local and economical texture description. However, they also require intensive computation to determine the proper parameters [19]. Signal processing methods, also known as multichannel filtering methods, are attractive due to their simplicity. In these methods, a textured input image is decomposed into feature images using a bank of filters such as Gabor, wavelet or neural network-based filters [115]. As a result, a high-dimensional textural pattern can be represented using a relatively small set of feature statistics that need to be extracted using a set of well-selected filters. Therefore, the major issue for this type of method is the selection of a good set of filters for a given texture classification problem.
6.3 Feature Extraction

The steps involved in the overall methodology for texture classification are shown in Figure 6.3. First the image is filtered using a sequential combination of Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT). The filter coefficients (responses) are post-processed by a set of non-linear functions which rectify the filter responses and remove local fluctuations. These non-linear functions consist of two stages: (1) obtaining the magnitude followed by (2) smoothing by a large Gaussian function. The feature vector computed from the local energy estimate is the local mean which represents the local texture characteristic. Rest of the section describes the steps of processing texture classification using features extracted with DWT and DCT.

![Diagram](image)

Figure 6.3: Stages of processing texture classification using features extracted with DWT and DCT.
We have used a 2-dimensional discrete wavelet transform (DWT) representation which can also capture the small differences in the rotation or scale as desired for the current applications. Daubechies 8-tap filters with 1-level decomposition of DWT has been used to compute the wavelet subbands. The wavelet analysis on a texture image of size 64 x 64 results in the approximation coefficients matrix of size 32 x 32 and three detailed coefficient matrices each of size 32 x 32 corresponding to horizontal, vertical and diagonal details. These four wavelet coefficient matrices are preprocessed. The preprocessing includes mean smoothing and Gaussian smoothing using a window of size 13 x 13.

6.3.1 Filtering

Input image is sequentially filtered using discrete wavelet transform [69] and discrete cosine transform [92] as shown in the first stage of Figure 6.3. Input image I is first decomposed using discrete wavelet transform (Daubechies filter) to give four sub-band (A, V, H and D) images S1, S2, S3 and S4. DCT is then performed on the sub-band images.

An $N \times 1$ DCT basis vector $u_m$ is expressed as:

$$u_m(k) = \begin{cases} \sqrt{\frac{1}{N}} & m = 1 \\ \sqrt{\frac{2}{N}} \cos \left( \frac{(2k-1)(m-1)\pi}{2N} \right) & m = 2, ..., N \end{cases}$$

A two-dimensional DCT can be evaluated by simply multiplying column basis one-dimensional DCT vectors with the row vectors of identical length. Lu C., P. Chung, and C. Chen [69] has introduced nine 3 x 3, 2-D DCT masks generated using the following three 1-D DCT basis vectors (with $N=3$) $u_1 = \{1,1,1\}^T$, $u_2 = \{1,0,-1\}^T$, and $u_3 = \{1,-2,-1\}^T$. 

122
Each wavelet subband is convolved with these nine 2-D DCT masks to generate 36 filtered images, as:

\[ H_i = S_j ** U_r \quad j = 1,...,4; \quad r = 1,...,9 \]

where ** denotes 2D convolution, \( S_j \) (\( j = 1,...,4 \)) denotes wavelet subbands; \( U_r \) (\( r = 1,...,9 \)) denotes 2D-DCT masks and \( H_i \) (\( i = 1,...,36 \)) denotes the filtered images. DCT decomposes the wavelet decomposed images into different subbands having different importance with respect to the visual quality of the image [74].

### 6.3.2 Smoothing

The absolute value of the filter responses \( H_i \) are convolved with a low pass Gaussian post filter \( G \) to yield a post-filtered energy.

\[ F_i = |H_i|**G \]

where,

\[ G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \]

** denotes 2D convolution, \(|.|\) denotes absolute value and \( F_i \) (\( i = 1,...,36 \)) are 36 smoothed images. This post-processing is necessary to retain only the signal energy in each subband and eliminate the undesired ripples in the output images.

After preprocessing, each matrix is represented by a one dimensional vector of size 1024. The four one dimensional vectors are concatenated to give one dimensional pattern vector of size 1 x 4096. Thus each texture image example is represented by a one dimensional vector of size 1 x 4096. This vector is being used for further processing which is explained in the following section.
6.4 Classification System Using SVM

In this section, we will describe the classification system devised to assess the potential of SVMs in texture classification. The operation of an SVM for texture classification has two steps: (1) The nonlinear mapping of a texture space into a possibly high-dimensional feature space and (2) The construction of an OSH in the feature space.

By introducing various kernel functions, various \( \Phi \) mappings can be used, which are implicitly imbedded in the SVMs. One of these mappings (induced by a polynomial kernel) takes the \( p \)-order correlations between the entries \( x_i \) of the input vector \( x \). If \( x \) represents a texture pattern whose entries are pixel values, this amounts to mapping the input space into the space of the \( p^\text{th} \) order products (monomials) of the input pixels. It should be noted that the direct computation of this type of feature is not easy even for a moderate sized problem. We note that \( N \)-dimensional input patterns include
\[
N_F = \frac{(N + p - 1)!}{p!(N-1)!}
\]
different monomials comprising a feature space \( F \) of dimensionality \( N_F \).

In multichannel filtering methods, a textured input image is decomposed into a set of feature images using a bank of filters. The decomposition is accomplished by filtering the input image, applying a nonlinear function to all the pixels in the filtered images and spatially smoothing each output image.

From Bayes classifiers to neural networks, there are many possible choices for an appropriate classifier. Among these, support vector machines (SVMs) would appear to be a good candidate because of their ability to generalize in high-dimensional spaces such as spaces spanned by texture patterns. The appeal of SVMs is based on their strong connection to the underlying statistical learning theory. That is, an SVM is an approximate implementation of the structural risk minimization (SRM) method [123].
For several pattern classification applications, SVMs have already been shown to provide better generalization performance when compared to traditional techniques, such as neural networks [101].

Figure 6.4 shows the block diagram of the model for recognition of texture images. It consists of two stages. In the first stage, the 1 x 4096-dimensional input pattern vectors \( (x) \) are used to train the SVM classifier. One-against-the-rest is used for decomposition of the learning problem in \( n \)-class pattern recognition into several two-class learning problems. An SVM is constructed for each class by discriminating that class against the remaining \((n-1)\) classes. Here each class refers to a texture class. The recognition system based on this approach consists of \( n \) number of SVMs. The set of training examples \( \{(x_i, y_i)\}_{i=1}^{N_k} \) consists of \( N_k \) number of examples belonging to \( k^{th} \) class, where the class label \( k \in \{1, 2, \ldots, n\} \). All the training examples are used to construct an SVM for a class.

The SVM for the class \( k \) is constructed using a set of training examples and their desired outputs, \( \{(x_i, y_i)\}_{i=1}^{N_k} \). The desired output \( y_i \) for a training example \( x_i \) is defined as follows:

\[
y_i = \begin{cases} 
+1 : x_i \text{ belongs to the } k^{th} \text{ class} \\
-1 : \text{otherwise}
\end{cases}
\]

\[ \text{SVM}_1 \quad \text{Evidence for Texture class 1 } \quad D_1(x) \]
\[ \text{SVM}_2 \quad \text{Evidence for Texture class 2 } \quad D_2(x) \]
\[ \vdots \quad \vdots \]
\[ \text{SVM}_{n-1} \quad \text{Evidence for Texture class } n-1 \quad D_{n-1}(x) \]
\[ \text{SVM}_n \quad \text{Evidence for Texture class } n \quad D_n(x) \]

\[ \text{Decision Logic} \]

\[ \text{Hypothesized Texture Class} \]

\[ \text{Figure 6.4: Block diagram of the texture image classification system using SVM models.} \]
The examples with $y_i = +1$ are called positive examples while those with $y_i = -1$ are called negative examples. An optimal hyperplane is constructed to separate positive examples from negative examples. The separating hyperplane is chosen in such a way as to maximize its distance from the closest training examples of different classes [101]. The support vectors are those training patterns that lie closest to the margin of separation (decision surface) of two classes and are therefore most difficult to classify. They are the most confusable patterns. They have a direct bearing on the optimum location of the decision surface. For a given test pattern $x$, the evidence $D_k(x), k = 1,2,\ldots,n$ is obtained from the $n$ SVMs. In the decision logic, the class label $k$ associated with the SVM that gives the maximum evidence is hypothesized as the class of the test pattern. The complexity of the SVM model for a class depends on the number of positive examples, the number of negative examples and the margin of separation between them in the kernel feature space.

As an illustration, evidence values obtained for test image of a texture class using SVM based classification system is given in Table 6.1. The values in each row are the evidences of the test image of a texture class against all the texture classes. It is seen that the highest evidence is obtained from the SVM model corresponding to the texture class to which the test image actually belongs. In the next subsection, we explain our studies on classification of texture images.
Test Pattern Evidence for

<table>
<thead>
<tr>
<th>Belongs to</th>
<th>Bark</th>
<th>Beachsand</th>
<th>Beans</th>
<th>Burlap</th>
<th>D10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bark</td>
<td>-0.120473</td>
<td>-0.922350</td>
<td>-0.506746</td>
<td>-0.865124</td>
<td>-0.682842</td>
</tr>
<tr>
<td>Beachsand</td>
<td>-0.229219</td>
<td>0.115243</td>
<td>-1.090169</td>
<td>-1.105183</td>
<td>-0.986143</td>
</tr>
<tr>
<td>Beans</td>
<td>-0.687008</td>
<td>-1.101708</td>
<td>-0.243064</td>
<td>-0.398958</td>
<td>-0.543236</td>
</tr>
<tr>
<td>Burlap</td>
<td>-0.941558</td>
<td>-0.598187</td>
<td>-0.660151</td>
<td>-0.415721</td>
<td>-0.462052</td>
</tr>
<tr>
<td>D10</td>
<td>-0.832802</td>
<td>-1.203701</td>
<td>-0.370205</td>
<td>-0.585507</td>
<td>-0.056796</td>
</tr>
</tbody>
</table>

Table 6.1. Evidence values for texture image classification system using SVM models for the sample test images.

6.4.1 Experimental Results

In this section, we describe the performance of the texture classification system for the database described in Section 6.2. For the development of SVM models, we consider 240 training examples. This corresponds to five texture classes and 48 examples from each class. The complexity of SVM models in terms of average number of support vectors per class is 199.6. For testing, we consider 80 images. This corresponds to five texture classes and 16 images from each class. It is seen from Table 6.2 that the classification system has provided 67.50% recognition. This performance is obtained for regularization parameter (\( c \)) value 400 and standard deviation (\( \sigma \)) value 8. These two parameters play an important role in this approach. These parameter values are obtained empirically.

<table>
<thead>
<tr>
<th>Category</th>
<th>Bark</th>
<th>Beachsand</th>
<th>Beans</th>
<th>Burlap</th>
<th>D10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classwise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>classification</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>05</td>
<td>15</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentage</td>
<td>67.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Classification performance for texture image classification system using SVM models. For each class, 16 examples are used for testing. Each example is represented using 1 x 4096-dimensional wavelet coefficients.
A similar study is conducted by representing each texture image using $1 \times 4096$ dimensional mean smoothed and Gaussian smoothed Gabor features. The best performance obtained using Gabor features is 45.00%. This is obtained for regularization parameter $(c)$ value 400 and standard deviation $(\sigma)$ value 4. Table 6.3 shows the percentage of correct classification of texture images for Gabor and wavelet types of feature representations.

<table>
<thead>
<tr>
<th>Feature representation</th>
<th>Classification performance (%)</th>
<th>Regularization Parameter $(c)$</th>
<th>Standard Deviation $(\sigma)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabor features</td>
<td>45.00</td>
<td>400</td>
<td>4</td>
</tr>
<tr>
<td>Wavelet features</td>
<td>67.50</td>
<td>400</td>
<td>8</td>
</tr>
</tbody>
</table>

*Table 6.3: Classification performance for Gabor and wavelet based features used in representation of texture images.*

We have made a comparative study of our proposed classification system based on SVM with the other classification system based on neural networks and KNN reported in the literature. The study shows that our proposed model based on SVM classification system gives a better result of 67.50% classification when compared to other classification models as shown in Table 6.4.

<table>
<thead>
<tr>
<th>Classification Model</th>
<th>Classification (%)</th>
<th>Misclassification (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural SOFM</td>
<td>61.02</td>
<td>38.98</td>
</tr>
<tr>
<td>K-Nearest Neighbor</td>
<td>64.15</td>
<td>35.85</td>
</tr>
<tr>
<td>SVM</td>
<td>67.50</td>
<td>32.50</td>
</tr>
</tbody>
</table>

*Table 6.4: Comparison of texture image classification models*
Table 6.4 tabulates the classification results for the neural self organizing feature map (SOFM), K-nearest neighborhood and SVM classifiers. Best results were obtained with SVM classifier with the classification results of 67.5% of texture images correctly.

6.5 Summary and conclusions

For texture classification purposes, we have used a 2-dimensional discrete wavelet transform (DWT) and support vector machines (SVMs). We have shown that Daubechies DWT based representation captures the differences in the rotation and scaled versions of the original texture images. Also, we have studied the representation of texture images using Gabor features. Support vector machines are used for classification. One-against-the-rest decomposition is adapted to apply binary SVMs to multiclass classification. The results show that it is possible to recognize the images of five texture classes correctly upto 67.50% using wavelet features. It is seen from Table 6.4 that in most cases, our proposed method provides better performance when compared to others. However, there is further scope for improvement in classification. One can explore a pyramidal approach using multi-level decomposition to improve the results. Further, it is necessary to perform some more studies on representation and classification for handling large number of texture classes.