A Texture Classification Method based on Gray-Level Co-Occurrence Matrix and Support Vector Machines

Yu-Min Chiang\textsuperscript{a}, Jo-Mien Weng\textsuperscript{b}

\textsuperscript{a}Department of Industrial Engineering and Management, I-Shou University, No.1, Sec. 1, Syuecheng Rd., Dashu District, Kaohsiung City 84001, Taiwan, R.O.C.

\textsuperscript{b}Institute of Manufacturing Information and Systems, National Cheng Kung University No.1, University Road, Tainan City 701, Taiwan (R.O.C.)

*Corresponding author: Dr. Yu-Min Chiang, Associate Professor Department of Industrial Engineering and Management, I-Shou University, No.1, Sec. 1, Syuecheng Rd., Dashu District, Kaohsiung City 84001, Taiwan, R.O.C. E-mail: ymchiang@isu.edu.tw

Funding source: This research was supported by National Science Council of Taiwan, Republic of China, under the contract number NSC 98-2221-E-214-035-.
Abstract
Texture is a low-level image feature and is recognized to be an important hint for image classification. Texture analysis plays a significant role in many pattern recognition and computer vision applications. However, traditional texture analysis methods are sensitive to rotation, translation, or scaling. This paper develops a rotation and scale invariant texture analysis method in which texture features of an image are extracted from the gray-level co-occurrence matrix (GLCM). Followed by a support vector machine (SVM) classifier, textured images with variant scaling and orientation can be accurately recognized. 20 monochrome texture images, each of size 640 × 640, obtained from Brodatz texture database were randomly selected (overlapping or no overlapping) for each of these 20 Brodatz texture images. Experimental results showed that the average classification accuracy is 95.04% and the highest accuracy is 99.17% for the proposed approach. Overall speaking, the proposed texture analysis method can achieve good classification performance.

Keywords: Texture classification, gray-level co-occurrence matrix (GLCM), support vector machines (SVM)

1. Introduction
Texture analysis plays a significant role in many pattern recognition and computer vision applications. The physical surface features comprise textural structures or patterns such as roughness, smooth, grainy, etc. There is no formal definition for texture. Visual texture is referred as alterations of intensities with absolute repeated structures or patterns (Fan and Xia, 2003). This descriptor provides measure of properties such as smoothness, coarseness, and regularity. It has been widely applied for the analysis of many types of images including natural scenes, remotely sensed data, biomedical modalities and defect inspection, and plays an important role in the human visual system for recognition and interpretation (Huang, et al., 2006).

Textures are generally classified into two major types: structural and statistical (Pikaz and Averbuch, 1997). Structural textures are those that are composed of repetitions of some basic texture primitives. This type of textures arises in textile fabrics, machined surfaces, patterned wafers, etc. Statistical textures cannot be described with primitives and deterministic rules of displacement. The spatial distribution of gray levels in such textured images is rather stochastic. Leather, sandpaper and many magnified metallic surfaces, for instance, fall in this category(Tsai and Huang, 2003) Texture analysis approaches generally compute a set of textural features in the spatial domain or in the spectral domain. The spatial domain refers to the image plane itself, and methods in this category are based on direct manipulation of pixels in an image. One commonly used spatial approach is gray-level co-occurrence matrix (GLCM) (Haralick et al, 1973). Gray-level co-occurrence Matrices have been referred to as Co-occurrence matrices. Second-order statistics derived from spatial gray-level co-occurrence matrices have been applied to IRS LISS-II sensor data classification on forest analysis (Kushwaha et al., 1994), roughness measurement of machined surfaces (Ramana and Ramamoorthy, 1996), and carpet wear assessment (Siew and Hogdson, 1998). On the other hand, spectral domain methods of texture analysis are frequency-based approaches. Spectral-domain features are generally less sensitive to noise than spatial-domain features(Tsai and Huang, 2003). In spectral domain approaches, the textural features are generally derived from the Gabor transform (ClarkandBovik, 1987), Fourier transform(FT) (Azencott et al., 1997), and wavelet transform (Huang, et al., 2006).Mizapour&Ghassemian (2013) utilize two well-known methods for extracting texture features from single-band satelliteimages: GLCM and Gabor filters. They proposed a fast GLCMalgorithm which significantly improved the speed of GLCM about 30 to 180 times faster for images. Experimental results show good capabilities of the proposed method in classification of PAN images.
Many texture analysis methods have been proposed over the last three decades. A common limitation of traditional texture analysis schemes is that they analyze the image at one single scale. Moreover, most of the texture analysis papers assume that the texture has the same orientation, which is not always the case. Ideally, texture analysis should be invariant to translation, scaling, and rotation. Zhang and Tan (2002) studied existing invariant texture analysis algorithms and classified them into three categories: statistical methods, model based methods, and structural methods.

Recently, some researchers focused on rotation invariant texture analysis, and proposed various classification methods for recognizing image types (Lee, 2008; Turkoglu and Avci, 2008; Wang et al., 2010). For example, Wang et al. (2010) described a method for rotation invariant texture analysis. Radon transform was utilized first to project a texture image onto projection space to convert a rotation of the original texture image to a translation of the projection in the angle variable. A k-nearest neighbors’ classifier with Radon projection correlation distances was then employed to implement texture classification and orientation estimation. Experimental results show the classification accuracy of this approach is about 99%. The proposed method also presents high noise tolerance and yields high accuracy in orientation estimation.

Nowadays, a powerful pattern recognition techniques called Support vector machines (SVMs) have been successfully applied to classification and regression problems, such as isolated handwritten digit recognition (Schoölkopf, Burges, and Vapnik, 1995), face detection in images (Huang et al, 2002), recommender systems (Zhang and Iyengar, 2002), and text categorization (Wang and Chiang, 2009). SVMs can be analyzed theoretically using concepts from computational learning theory (Burges, 1998; Vapnik, 1998). They have outperformed many other machine learning methods such as k-nearest neighbors and artificial neural networks because they own the properties such as good generalization performance, robustness in the presence of noise, ability to deal with high dimensional data, and fast convergence.

Lee (2008) proposed a rotational texture classification method based on multi-model feature integration by ellipsoid support vector machines (ESVMs). Three feature sets based on three texture models - the Gabor filter, neighboring gray level dependence matrix (NGLDM), and minimum volume enclosing ellipsoids are used to describe the image properties. He used the intensity of decision function value to combine multiple ESVMs classifiers for image classification. 25 Brodatz textures were used to evaluate the classification performance. Experimental results show the average classification accuracy is 86.04%, which demonstrated the acceptable performance of the proposed system. Turkoglu and Avci (2008) compared wavelet-support vector machine (W-SVM) and wavelet-adaptive network based fuzzy inference system (W-ANFIS) approaches for texture image classification. Both W-SVM and W-ANFIS methods are used for classification of the 22 texture images obtained from Brodatz image album. 50 64 × 64 image regions were randomly selected for each of these 22 Brodatz texture images. In their experimental results, mean 92.81% and 91.27% recognition success was obtained for W-SVM and W-ANFIS intelligent systems, respectively.

In this paper, a GLCM combined with SVMs approach is proposed to overcome the rotation-invariant and scale-invariant problems as recognizing images with textures. The rotation-invariant and scale-invariant textural features are extracted from GLCM, and the SVM classifier is used for improving classification accuracy. Experiments are conducted with 20 monochrome texture images, each of size 640 × 640, obtained from Brodatz texture images, as shown in Figure 1.

This paper is organized as follows: In Section 2, the features generated from gray-level co-occurrence is introduced. In Section 3, the theory of the SVM classifier is briefly reviewed. In Section 4, the texture classification experimental results obtained by using GLCM features and SVM classifier are given. The concluding remarks are given in Section 5.
2. Gray-level co-occurrence matrix

A gray-level co-occurrence matrix (GLCM), also referred to as a co-occurrence matrix or a co-occurrence distribution, is defined over an image to be the distribution of co-occurring values at a given offset. GLCM considers the relation between two pixels at a time, called the reference and the neighbor pixel. Mathematically, a co-occurrence matrix $C$ is defined over an $n \times m$ image $I$. The co-occurrence probabilities represent the conditional joint probabilities of all pairwise combinations of gray levels in the spatial window of interest given two parameters: interpixel distance (or displacement) ($d$) and orientation ($\theta$). In this research, four orientations ($\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$) were considered.

Suppose an image to be analyzed is rectangular and its size is $n \times m$, and the gray level appearing at each pixel is quantized to $N_g$ levels, thus the dimension of the relative matrix is $N_g \times N_g$. The texture-content information is specified by the matrix of relative frequencies $P_{i,j}$ with two neighboring pixels separated by interpixel distance $d$ occur on the image, one with gray level $i$ and the other with gray level $j$. Such matrices of gray-level co-occurrence frequencies are a function of the angular relationship and distance between the neighboring pixels (Soh and Tsatsouli, 1999).

Let $I(x,y)$ represent the gray-level of a pixel $(x,y)$. The following equations define the co-occurrence frequencies at 4 orientations ($\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$).

\[
P(i, j | d, \theta) = \# \{ k - m = 0, l - n = d, I(k,l) = i, I(m,n) = j \} \tag{1}
\]
\[
P(i, j | d, 45^\circ) = \# \{ (k - m = d, l - n = -d) \text{or}(k - m = -d, l - n = d), I(k,l) = i, I(m,n) = j \} \tag{2}
\]
\[
P(i, j | d, 90^\circ) = \# \{ k - m = d, l - n = 0, I(k,l) = i, I(m,n) = j \} \tag{3}
\]
\[
P(i, j | d, 135^\circ) = \# \{ (k - m = d, l - n = d) \text{or}(k - m = -d, l - n = -d), I(k,l) = i, I(m,n) = j \} \tag{4}
\]

where $\#$ denotes the number of elements in the set.

Let $p(i, j)$ be the $(i,j)$th entry in a normalized GLCM, which represents the co-occurrence probability.

\[
p(i, j) = \frac{P(i, j | d, \theta)}{N} \tag{5}
\]

where $N$ is the total number of possible outcomes.

The mean and standard deviations for the rows and columns of the matrix are:

\[
\mu_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i \times p(i, j) \tag{6}
\]
\[
\mu_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} j \times p(i, j) \tag{7}
\]
\[
\sigma_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)^2 \times p(i, j) \tag{8}
\]
\[
\sigma_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (j - \mu_y)^2 \times p(i, j) \tag{9}
\]

Let $P_x(i)$ be the sum of the probabilities of the $i$-th column in the normalized GLCM, $P_y(j)$ be the sum of the probabilities of the $j$-th row in the normalized GLCM.

\[
p_x(i) = \sum_{j=1}^{N_g} p(i, j), \; i = 1, 2, \ldots, N_g \tag{10}
\]
\[
p_y(j) = \sum_{i=1}^{N_g} p(i, j), \; j = 1, 2, \ldots, N_g \tag{11}
\]
According to the co-occurrence matrix, Haralick et al. (1973) proposed 14 textural features. This research calculated the 14 textural features together with the other 8 textural features adopted by Soh and Tsatsoulis (1999) and Clausi (2002) as the image features. The following equations define these 22 textural features.

**F01. Uniformity / Energy / Angular Second Moment:**

\[ F_1 = \sum \sum \{p(i, j)\}^2 \]

(14)

**F02. Entropy:**

\[ F_2 = -\sum \sum p(i, j)\times \log \{p(i, j)\} \]

(15)

**F03. Dissimilarity:**

\[ F_3 = -\sum \sum |i - j|\times p(i, j) \]

(16)

**F04. Contrast / Inertia:**

\[ F_4 = \sum_{n=0}^{N_x-1} n^2 \times \left\{ \sum \sum p(i, j) \times |i - j| = n \right\} \]

(17)

**F05. Inverse difference:**

\[ F_5 = \sum \sum \frac{1}{1 + |i - j|}\times p(i, j) \]

(18)

**F06. Correlation:**

\[ F_6 = \sum \sum (i \times j)\times \left[ p(i, j) - \mu_x \mu_y \right] \]

\[ \sigma_x \sigma_y \]

(19)

**F07. Homogeneity / Inverse difference moment:**

\[ F_7 = \sum \sum \frac{1}{1 + (i - j)^2}\times p(i, j) \]

(20)

**F08. Autocorrelation:**

\[ F_8 = \sum \sum (i \times j)\times p(i, j) \]

(21)

**F09. Cluster Shade:**

\[ F_9 = \sum \sum (i + j - \mu_x - \mu_y)^3 \times p(i, j) \]

(22)

**F10. Cluster Prominence:**

\[ F_{10} = \sum \sum (i + j - \mu_x - \mu_y)^4 \times p(i, j) \]

(23)

**F11. Maximum probability:**

\[ F_{11} = \text{MAX}_{i,j} p(i, j) \]

(24)
F12. Sum of Squares:
\[ F_{12} = \sum_{i} \sum_{j} (i - \mu)^2 \times p(i, j) \]  
(25)

F13. Sum Average:
\[ F_{13} = \sum_{i=2}^{2N_x} i \times p_{xy}(i) \]  
(26)

F14. Sum Variance:
\[ F_{14} = \sum_{i=2}^{2N_x} (i - F_{13})^2 \times p_{xy}(i) \]  
(27)

F15. Sum Entropy:
\[ F_{15} = -\sum_{i=2}^{2N_x} p_{xy}(i) \times \log\{p_{xy}(i)\} \]  
(28)

F16. Difference variance:
\[ F_{16} = \text{variance of } p_{x-y}(i) \]  
(29)

F17. Difference entropy:
\[ F_{17} = -\sum_{i=0}^{N_x-1} p_{x-y}(i) \times \log\{p_{x-y}(i)\} \]  
(30)

F18. Information measures of correlation (1)
F19. Information measures of correlation (2)
\[ F_{18} = \frac{H_{XY} - H_{XY1}}{\max\{H_X, H_Y\}} \]  
(31)

\[ F_{19} = (1 - \exp[-2.0 \times (H_{XY2} - H_{XY})])^{1/2} \]  
(32)

where \( H_{XY} = -\sum_{i} \sum_{j} p(i, j) \times \log(p(i, j)) \)
(33)

\[ H_X = -\sum_{i} p_x(i) \times \log(p_x(i)), \quad H_Y = -\sum_{j} p_y(j) \times \log(p_y(j)) \]  
(34)

\[ H_{XY1} = -\sum_{i} \sum_{j} p(i, j) \times \log\{p_x(i) \times p_y(j)\} \]  
(35)

\[ H_{XY2} = -\sum_{i} \sum_{j} p_x(i) \times p_y(j) \times \log\{p_x(i) \times p_y(j)\} \]  
(36)

F20. Maximal correlation coefficient:
\[ F_{20} = \left(\text{Second largest eigenvalue of } Q\right)^{1/2} \]  
(37)

where \( Q(i, j) = \sum_{k} \frac{p(i, k) \times p(j, k)}{p_x(i) \times p_y(k)} \)  
(38)

F21. Inverse difference normalized:
\[ F_{21} = \sum \frac{p(i, j)}{1 + (i - j)} \]  
(39)
F22. Inverse difference moment:

\[ F_{22} = \sum \frac{p(i,j)}{1+(i-j)^2} \]  

(40)

3. Support Vector Machines

Support vector machine (SVM) is a classifier derived from statistical learning theory (Vapnik, 1998). SVMs have become a significant method for pattern recognition because they own the properties such as good generalization performance, robustness in the presence of noise, ability to deal with high dimensional data, and fast convergence.

We now provide a brief introduction of SVMs here. More detailed information can be found in Burges (1998) or Kumar (2005). In the two class classification problem, suppose we are given a training set of examples \( x_i \in \mathbb{R}^n \ (i = 1,...,n) \), with corresponding labels \( d_i \in \{-1,+1\} \ (i = 1,...,n) \), where +1 and -1 stand for the positive class (C1) and negative class (C2). For a linearly separable set of data, there exists a set of separating hyperplanes \( w \cdot x + b = 0 \), where \( w \) is a weight vector and \( b \) is the bias or threshold. SVMs find a unique separating hyperplane between the two classes in input space that maximizes the margin between the hyperplane and the classes, as shown in Figure 2. The total margin can be expressed as \( 2/\|w\| \), and the data points lie on the margin are called support vectors.

-- Insert Figure2 about here -- 

To have good generalization, the total margin should be as large as possible. Therefore, the objective is then to minimize the squared Euclidean norm of \( w \), \( \|w\|^2 \), subject to the constraints: \( d_i (w \cdot x_i + b) - 1 \geq 0 \ (i = 1,...,n) \). This problem is called as standard quadratic optimization problem. The Lagrange function is used for solving of this optimization problem. The weight vector is generally expressed in terms of the linear combination of the training patterns: \( w = \sum_{i=1}^{n} \lambda_i d_i x_i \ (\lambda_i \geq 0) \). When \( \lambda_i > 0 \), the corresponding pattern contributes to the hyperplane and it is a support vector. The decision hyperplane can be written as:

\[ f(X) = \text{sgn} \left[ \sum_{i=1}^{n} \lambda_i d_i (w \cdot x_i) + b \right] \]  

(41)

where for a number \( x \), \( \text{sgn}(x) \) is 1 if \( x \) is larger than 0 and is -1 otherwise.

For linearly separable data, the weight vector \( w \) can be obtained from the labeled training data set and the classification of any unknown data point \( X \) can be directly calculated by Eq. (41).

However, in many cases, the data are non-linearly separable. The way to go about performing the optimization is to permit the algorithm to misclassify some of the data points with an increased cost. The idea is integrated into the optimal margin algorithm by introducing slack variables \( \xi_i \geq 0 \ (i = 1,...,n) \), into the constraints. The optimization problem thus gets modified to the following form:

\[ \text{Minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \]  

(42)

\[ \text{Subject to} \quad d_i (w \cdot x_i) + b - 1 + \xi_i \geq 0 \quad i = 1,...,n \]  

(43)

\[ \xi_i \geq 0 \]

Where \( C \) is the soft margin constant penalty parameter.

Another way to solve the non-linearly separable problem is to perform a nonlinear mapping of the input data into a new feature space, usually with a higher dimension, where the input vectors are linearly separable.
However, it is sometimes difficult to define the mapping function explicitly. One can observe the final decision function in the new high dimensional feature space is only in the form of dot product of the two feature vectors. Fortunately, inner products in the high dimensional feature space are computable using a kernel function in input space. In SVM literature, there are many kernel functions such as linear, polynomial, radial basis function (RBF), sigmoid, etc. The RBF kernel function is most commonly of these kernel functions. The typical kernel functions are given as below (Hsu et al., 2010):

- **linear kernel function:** \( K(X_i, X_j) = X_i^T X_j \)  
- **polynomial kernel function:** \( K(X_i, X_j) = (\gamma X_i^T X_j + r)^d \), \( \gamma > 0 \)
- **radial basis function (RBF) kernel:** \( K(X_i, X_j) = \exp\left(-\gamma \|X_i - X_j\|^2\right) \), \( \gamma > 0 \)
- **sigmoid kernel function:** \( K(X_i, X_j) = \tanh(\gamma X_i^T X_j + r) \)

Here, \( \gamma \), \( r \), and \( d \) are kernel parameters.

4. Experimental results

This research proposed a rotation-invariant and scale-invariant texture classification method by using GLCM and SVMs. To verify the performance of the proposed method, the study used Brodatz textures (Brodatz, 2010) which are the most commonly used texture data set in the generic texture interpretation research literature. This imagery provides a variety of classes and allows comparison with other research.

4.1 Samples preparing

The experiments were conducted with 20 monochrome texture images, each of size 640 x 640, obtained from Brodatz textures, as shown in Figure 1. To evaluate the scale-invariant and rotation-invariant properties of the proposed GLCM features, six clusters of images are generated from each of the Brodatz image: original image, enlarged image by scale factor 2, reduced image by scale factor 0.5, enlarged image by scale factor 2 in 5 orientations from 15° to 90° (15°, 30°, 45°, 60°, and 90°), and reduced image by scale factor 0.5 in 5 orientations from 15° to 90° (15°, 30°, 45°, 60°, and 90°). 5200 x 200 texture image regions are randomly selected (overlapping or nonoverlapping) for each of these 20 Brodatz texture images in six clusters. 3 of these image regions are used for training and other 2 of these image regions are used for testing when using SVM classifier. Hence, there are totally 5x20x6 = 600 image samples which are prepared for further analysis.

4.2 Experiments

The proposed GLCM method has been implemented using Matlab 7.1 on a personal computer with Microsoft Windows XP Professional Version 2002 Service Pack3, AMD Athlon(tm) 64 Processor 3200+ 1.99 GHz, and RAM 704MB. The SVM classifications were carried out using BSVM tool (Hsu and Lin, 2006) as BSVM can solve SVMs for the solution of large classification problems.

As mentioned in Section 2, four orientation parameter of GLCM (\( \theta = 0°, 45°, 90°, 135° \)) are considered in the paper. The interpixel distance \( d \) of GLCM was set to 10 pixels in this study. To realize the effects of the orientation parameter \( \theta \) on texture classification and to evaluate the performance of the proposed approach, this research conducted 7 experiments, as explained below. Note that among the 5 200 x 200 image regions, the first 3 images are used for training and the other two are used for testing for the SVM classifier except for Experiment 2 and Experiment 7.

**Experiment 1:**
The orientation parameter $\theta$ in co-occurrence matrices was set to $\theta=0^\circ$, $45^\circ$, $90^\circ$, $135^\circ$. For each of the 4 orientations, 22 GLCM features are calculated and they are viewed as a feature vector. Thus there are totally $4 \times 600=2400$ samples. Size of the training feature vector obtained in this experiment is $(3/5) \times 2400 = 1440$.

Experiment 2:
All samples are the same as Experiment 1, except for the training and test samples. In this experiment, the first three image regions are used for testing and the other two are used for training for the SVM classifier. Size of the training feature vector obtained in this experiment is $(2/5) \times 2400 = 960$.

Experiment 3:
For the GLCM orientation parameter, only $\theta=0^\circ$ is considered. The total number of samples in this experiment is 600. Size of the training feature vector obtained in this experiment is $(3/5) \times 600 = 360$.

Experiment 4:
For the GLCM orientation parameter, only $\theta=45^\circ$ is considered. The total number of samples in this experiment is 600. Size of the training feature vector obtained in this experiment is $(3/5) \times 600 = 360$.

Experiment 5:
For the GLCM orientation parameter, only $\theta=90^\circ$ is considered. The total number of samples in this experiment is 600. Size of the training feature vector obtained in this experiment is $(3/5) \times 600 = 360$.

Experiment 6:
For the GLCM orientation parameter, only $\theta=135^\circ$ is considered. The total number of samples in this experiment is 600. Size of the training feature vector obtained in this experiment is $(3/5) \times 600 = 360$.

Experiment 7:
The training sample dataset will set to be the same as the experiment with the highest classification accuracy in Experiments 1 to 6, and the test dataset is the same as Experiment 2 since it has the most test samples.

4.3 Results:
The displacement parameter $(d)$ of the GLCM was set to 10 pixels in this research. 4 orientation parameters ($\theta=0^\circ$, $45^\circ$, $90^\circ$, $135^\circ$) were considered. The research utilized BSVM 2.01 provided by Dr. Lin (http://www.csie.ntu.edu.tw/~cjlin/bsvm/) as SVM analysis tool. As the texture classification problem is almost non-linearly separable, we used SVM's with linear, polynomial, RBF, and sigmoid function as kernel function. Grid Parameter Search algorithm provided by Dr. Lin was used to search the better setting of cost parameter in Eq. (42) and Gamma parameter in Eq. (46) for RBF kernel. The parameters setting of other kernels were obtained by a pre-test result. The classification accuracy is shown in Table 1. It can be found that the RBF kernel function of SVM performs better than other three kernels. The orientation parameter $\theta$ in co-occurrence has significant influence on the classification accuracy, and it has the highest texture classification accuracy as $\theta=0^\circ$.

Among the experiments, Experiment 3 has the highest recognition rate (99.17%) as the SVM kernel function is RBF. The training dataset of Experiment 3 and the test dataset of Experiment 2 are selected to conduct experiment 7. When using RBF kernel function, the classification accuracy achieved to 99.72%. It verifies that the proposed GLCM textural features are scale-invariant and orientation-invariant.

All misclassified image samples in Experiments 1 to 7 were chosen for confusion analysis. The confusion matrix of the SVM classifier using RBF kernel is shown in Table 2. From the confusion matrix, it can be found that image class 15 (Brodatz textural image D91) is the most difficult to be accurately recognized.
5. Conclusions

Traditional texture analyses are often sensitive to translation, scaling, and rotation. The study overcame this problem by extracting some invariant features from the gray-level co-occurrence matrix. In this study, a texture image classification system is introduced. In feature extraction stage of this proposed system, the GLCM method is used, and SVM classifier is used in classification stage. Finally 22 textural features obtained from co-occurrence matrix are given to inputs of SVM classifier. Experiments were conducted with 20 monochrome texture images acquired from Brodatz texture images. Six types of image samples including resized and rotational images are used to evaluate the rotation-invariant and scale-invariant properties of the GLCM features. As shown in Table 1, the correctness performance of classification processing is very good for SVM, especially for the RBF kernel function in orientation parameter $\theta = 0^\circ$. The highest recognition rate is 99.17%, respectively. The classification accuracy compared to previous methods is promising.

The classification performances of this study show the advantages of the proposed approach: it is scale-invariant and orientation-invariant and with high recognition rate. The proposed method offers advantage in texture image classification. Future research directions are toward increasing the variety and number of texture images and implementing a real-time intelligent texture classification system.

References


# Table 1. Classification accuracy of 20 texture images (%)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientation (θ)</td>
<td>0°, 45°, 90°, 135°</td>
<td>0°, 45°, 90°, 135°</td>
<td>0°</td>
<td>45°</td>
<td>90°</td>
<td>135°</td>
<td></td>
</tr>
<tr>
<td>linear</td>
<td>97.19</td>
<td>95.69</td>
<td>98.33</td>
<td>90.42</td>
<td>90.83</td>
<td>88.75</td>
<td>93.55</td>
</tr>
<tr>
<td>polynomial</td>
<td>97.92</td>
<td>95.21</td>
<td>98.33</td>
<td>72.92</td>
<td>90.83</td>
<td>90.00</td>
<td>90.87</td>
</tr>
<tr>
<td>RBF</td>
<td>97.71</td>
<td>95.76</td>
<td>99.17</td>
<td>92.50</td>
<td>90.42</td>
<td>90.00</td>
<td>95.04</td>
</tr>
<tr>
<td>sigmoid</td>
<td>92.29</td>
<td>90.97</td>
<td>92.50</td>
<td>91.67</td>
<td>88.83</td>
<td>87.50</td>
<td>90.63</td>
</tr>
<tr>
<td>average</td>
<td>96.28</td>
<td>94.41</td>
<td>97.08</td>
<td>86.88</td>
<td>90.23</td>
<td>89.06</td>
<td>92.52</td>
</tr>
</tbody>
</table>
Table 2. Confusion matrix of the 20 texture images

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>sum</td>
<td></td>
<td>8</td>
<td>15</td>
<td>7</td>
<td>14</td>
<td>6</td>
<td>4</td>
<td>13</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>7</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
<td>7</td>
<td>154</td>
</tr>
</tbody>
</table>
Figure 1.20 Brodatz texture images. From left to right and top to bottom: D1, D2, D5, D13, D35, D38, D40, D41, D46, D47, D60, D68, D83, D91, D99, D101, D102, D104, D105, D107

Figure 2. Separation of two classes by SVM

Class 1

Margin $1/|w|$

Class 2

$w \cdot X + b = 0$

Total Margin $2/|w|$

$w \cdot X + b = -1$

Support vectors

$w \cdot X + b = 1$