

# Wavelets and Support Vector Machines for Texture Classification

Kashif Mahmood Rajpoot  
*Faculty of Computer Science & Engineering,  
Ghulam Ishaq Khan Institute,  
Topi, PAKISTAN.*  
[kmr@giki.edu.pk](mailto:kmr@giki.edu.pk)

Nasir Mahmood Rajpoot  
*Department of Computer Science,  
University of Warwick,  
Coventry, UK.*  
[nasir@dcs.warwick.ac.uk](mailto:nasir@dcs.warwick.ac.uk)

## Abstract

*We present a novel texture classification algorithm using 2-D discrete wavelet transform (DWT) and support vector machines (SVM). The DWT is used to generate feature images from individual wavelet subbands, and a local energy function is computed corresponding to each pixel of the feature images. This feature vector is first used for training and later on for testing the SVM classifier. The experimental setup consists of images from the Brodatz and MIT VisTeX texture databases and a combination of some images therein. The proposed method produces promising classification results for both single and multiple class texture analysis problems.*

## 1. Introduction

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. Accurate results have been achieved traditionally through various schemes but only working under certain assumptions or limitations.

With the increasing popularity of digital libraries, image and multimedia databases, texture analysis has become a focus area of research since image regions can be described by their textural properties. It can also be extremely useful in numerous other related areas, for example: content-based image retrieval, remote sensing, image segmentation and interpretation, object recognition, and industrial applications.

Several approaches for texture analysis have been proposed by researchers working in this area during last three decades or so. The texture classification

problem is normally divided into two steps of (i) feature extraction, and (ii) classification. Statistical, model-based and/or filtering (signal processing) based approaches are perhaps the most commonly used feature extraction techniques [1]. It is not easy to predict or even experimentally validate the superiority of one method over the other. It is, however, commonly believed that the filtering based approach often produces relatively high accuracy results for majority of the test images from various classes (aerial, natural, animal, etc.). A large variety of filtering techniques have been used for texture analysis including: wavelet decomposition, Gabor filtering, discrete cosine transform (DCT), ring/wedge filtering, and multi-channel filtering. Randen et al. [1] provide a comprehensive study of filtering methods for texture analysis.

In this work, our choice of feature extraction and classifier is Discrete Wavelet Transform (DWT) and Support Vector Machines (SVM), respectively. There are a number of wavelet filters and adaptations of the conventionally used dyadic wavelet transform (e.g., wavelet frame, wavelet packet) which may be helpful in capturing distinct textural characteristics. In addition, wavelets offer an inherent multiresolution analysis which can help in representing discriminant textural features at various scales. The application of SVM as a classifier to the problem of texture analysis is relatively new [2, 3] and perhaps not fully explored. Kim et al. [2] bypassed the feature extraction stage and left it to be performed internally by the SVM. The misclassification error for most of their test images was around 15-20%, which is high noting the potential of SVM classifier. Li et al. [3] employ the DWFT (Discrete Wavelet Frame Transform) and present their work for a single-texture classification problem with a total of 30 texture classes. Their technique for feature extraction by computing the energy distribution limits their work to only single-texture case and is not applicable to multi-texture classification problem (where there are more than one texture classes in a

single image). Also, instead of conducting parameter selection for SVM, they have used a classifier fusion scheme by training 3 classifiers for each distinct texture class. This makes their method computationally intensive both at the training and the testing stages. Besides, this may not ensure the best possible results of SVM since optimal parameters are never searched.

Our work, in contrast, uses a DWT subband filtering based local energy function for the feature extraction stage and a manual parameter selection for SVM. This makes our method less computationally demanding (for classifier testing) and essentially applicable to both single-texture and multi-texture classification cases. We present our results for both the cases. Next section gives an introductory overview of the SVM. Section 3 describes the feature extraction and classification stages followed by the experimental results in Section 4. We conclude with a summary of observations made from this study and future work directions in Section 5.

## 2. SVM – An introductory overview

In the context of supervised classification, *machine learning* and *pattern recognition* is the extraction of regularity or some sort of structure from a collection of data. Neural networks (NN) and Bayesian classifiers are the typical examples to learn such organization from the given data observations. Support Vector Machines (SVM) is a relatively new classifier and is based on strong foundations from the broad area of statistical learning theory [4]. Since its inception in early 90s, it has found applications in a wide range of pattern recognition problems, to name a few: handwritten character recognition, image classification, financial time series prediction, face detection, bioinformatics, biomedical signal analysis, medical diagnostics, and data mining.

SVM has become, in practice, the classifier of choice of numerous researchers and practitioners for several real-world classification problems. This is because SVM is capable of generalizing well (predicting the unseen or unknown samples with a good degree of accuracy) as compared to many traditional classifiers (NN, etc.) It offers several advantages which are typically not found in other classifiers:

- Computationally much less intensive (esp. in comparison to NN)
- Performs well in higher dimensional spaces (a factor which limits many efficient classifiers)
- Lack of training data is often not a severe problem

- Based on minimizing an estimate of test error rather than the training error (*structural risk minimization*)
- Robust with noisy data (noise can severely degrade the performance of NN)
- Does not suffer as much from the curse of dimensionality and prevents overfitting

It is the first author's observation that research activity in pattern recognition areas, at the national level, has missed out so far on this powerful classification method. Therefore, an introductory overview of SVM is presented in the remainder of this section.

### 2.1 Introduction to support vector machines

A binary class supervised classification problem is usually formulated in the following way: given  $n$  training samples  $(\langle x_i \rangle, y_i)$  where

$\langle x_i \rangle = (x_{i1}, x_{i2}, \dots, x_{im})$  is an input feature vector and  $y_i \in \{-1, +1\}$  is the target label, the task of the

discriminant function or a classifier is to learn the patterns in the training samples in such a way that at a later stage it can predict reliably a  $y_i$  for an unknown

$x_i$ . SVM is fundamentally developed for such binary classification case and is extendable for multi-class situation. Like other linear classifiers, it attempts to evaluate a linear decision boundary (assuming that the data is linearly separable) or a linear hyperplane between the 2-classes (Figure 1a). Theoretically, when the data is linearly separable, there exist possibly an infinite number of hyperplanes (Figure 1b) which can correctly classify the training data. SVM, unlike other classifiers of its kind, strives to find out an optimal hyperplane (Figure 1c).

It is commonly believed that points belonging to the two data classes often lie in such a way that there is always some 'margin' between them. SVM attempts to maximize this margin ( $2\gamma$  in Figure 1c) by considering it as a quadratic programming problem, see [4, 5] for mathematical formulation and derivation of the solution.

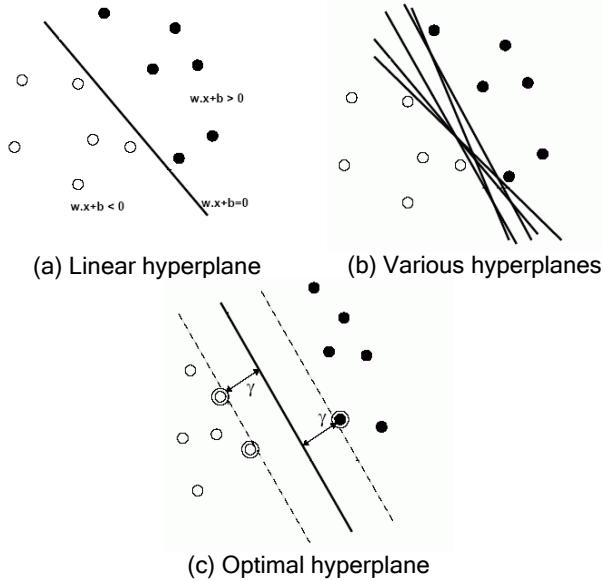


Figure 1: Evaluation of an optimal hyperplane

## 2.2 Statistical learning theory – Key to generalization for SVM

SVM foundations are strongly rooted in the *statistical learning theory* (SLT): an area of study which addresses the issue of how to control the generalization ability of a classifier. Vapnik and Chervonenkis [4] developed this theory for creating classifiers or discriminant functions that can generalize well. Normally, classifiers like NN try to minimize the training error (called *empirical error*) but, statistically, there is no guarantee that such a classifier (with zero empirical error) will perform well on the test data. SLT, in contrast, provides an estimate of the test error (called *risk*):

$$Er \leq Ee + c \sqrt{\frac{V(F)}{N}}$$

where  $Er$  is the risk,  $Ee$  is the empirical error,  $c$  is a constant,  $V(F)$  is the VC-dimension (the Vapnik-Chervonenkis dimension, denoting the maximum number of samples that can be shattered by a set of functions) and  $N$  is the number of training samples. This equation describes the basis of the *structural risk minimization* principle. SVM, in practice, is based on the minimization of this risk estimate which often leads to efficient classifiers.

## 2.3 Kernel function – The kernel-trick

The concepts above are presented for a linear classification case. These are generalizable to a non-linear case where a mapping function  $\phi(x_i)$  is used to map the input space into a higher dimensional feature space such that the non-linear hyperplane becomes linear (Fig 2). To avoid the increased computational complexity and curse of dimensionality, a *kernel-trick* or kernel function  $K(x_i, x_j)$  is employed which, in essence, computes an equivalent kernel value in the input space such that no explicit mapping is required [6].

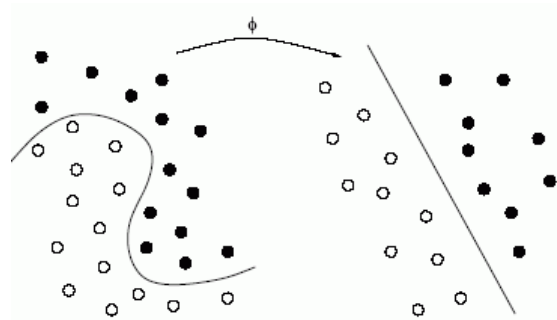


Figure 2: Feature mapping

A few popular kernels are:

Linear:-  $K(x_i, x_j) = \langle x_i, x_j \rangle$

Gaussian:-  $e^{-\gamma \|x_i - x_j\|^2}$

Polynomial:-  $(\gamma \langle x_i, x_j \rangle + a)^d$

## 2.4 Further links

A friendly guide for practitioners to use SVM is [7]. Vert [6] gives an excellent introduction to SVM for beginners with no background in this area. A more detailed tutorial is contained in [5], which is oriented for researchers working in pattern recognition. A standard text on the subject issues by the architect of SVM is [4]. Current research issues and links can be found in [8, 9].

A large number of software implementations of SVM with distinctive features have been developed by researchers in the machine learning area. A nice and efficient one is the LIBSVM [10] developed in C++, whereas its MATLAB variant OSU SVM [11] is also available.

### 3. Our approach

This paper considers the problem of texture classification only for a gray-level case which is conventionally tackled in two stages of feature extraction and classification.

#### 3.1 DWT feature extraction

A large variety of feature extraction methods exist which are based upon signal processing (or filtering) techniques (see [1] for a comprehensive survey of such techniques). Wavelet filtering is one such method that can be successfully used for feature extraction in texture analysis. The idea of using the wavelets for feature extraction in texture classification context is not entirely new and researchers have been using it for over a decade or so in one or the other form. The inherent capability to do so was highlighted by Mallat’s work [12, 13] on multiresolution analysis using the wavelet transform.

We have used a simple Discrete Wavelet Transform (DWT) representation which can also capture the small differences in the rotation or scale that might be desired for some applications. Depending upon the need, other representation options can be a Discrete Wavelet Frame Transform (DWFT) [14], a Discrete Wavelet Packet Transform (DWPT) [15, 16], or scale, rotation, and translation invariant wavelets [17].

We used Daubechies’ 8-tap filters and a 3-level decomposition of DWT to compute the wavelet subbands. As shown in Figure 3, feature images are generated by reverse filtering each of the individual subbands except the lowpass subband at the coarsest resolution. A local energy function is then computed corresponding to each pixel in each of the feature images. This yields a 9-dimensional feature vector  $\vec{x} = [x_1, x_2, \dots, x_9]$  which is supposed to contain the major properties of each of the texture classes. This type of feature extraction enables our method to be applicable in both single-texture and multi-texture situations, as is not the case in [3] where the energy distribution for each subband is computed. In this work, we omitted any extensive search for different decomposition levels or wavelet filters (for such a study, see [18]) as our focus is to study a method based upon wavelets and SVM that can be used in various situations.

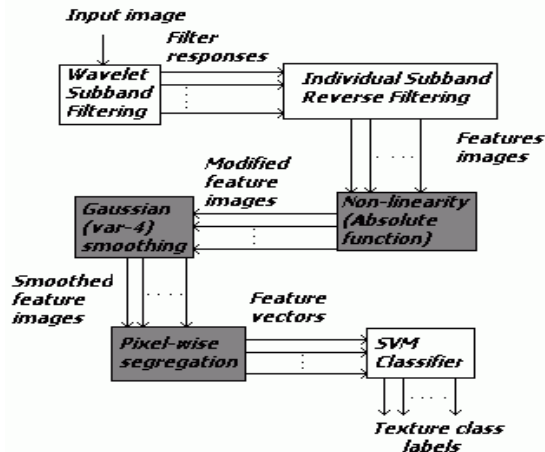


Figure 3: Texture classification method setup (local energy function computed in shaded blocks)

#### 3.2 SVM classification

As shown in Figure 3, the local energy based feature vector is fed into the SVM classifier initially for training (to learn the pattern) from known examples and for predicting the labels of unknown samples once the training is complete. As described above, SVM is principally a binary classifier. We use an all-at-once decomposition scheme; other possible options are one-against-others and one-against-one.

A Gaussian kernel function was used due to its superiority over other kernels for most of the applications. It always pays off using optimal parameter values for the respective kernel. For the problem at hand, a manual search found the parameter values  $C = 1, \gamma = 1 \times 10^{-3}$  for the Gaussian kernel which were used in all of the following experiments to perform the training.

### 4. Experimental setup & results

In order to demonstrate the potential of our proposed texture classification method, we experimented with its application to a number of scenarios. We obtained images from Brodatz album [19], MIT’s VisTex album [20], and a few aerial and synthetic samples [21], and combined them in different ways. Experimental results for three scenarios and their discussion is presented in the following sections.

#### 4.1 Supervised segmentation

In supervised classification, it is often the case that the training and test data are taken as non-overlapping data subsets. However, in some real-world situations (e.g., content-based image retrieval or interactive

segmentation), it may not be feasible to do so. In such cases, no data other than the test image itself is available to collect both training examples and test samples. It may also be the case that the ground truth (which is commonly used to validate the quantified performance of the classifier’s predictions) is not available. In such situations, the training data is collected as a limited subset of existing textures in the test image and once the classifier is trained, it is asked to predict the labels for whole of the image. Figures 4 and 5 show supervised segmentation results for binary-texture and multi-texture images, respectively. In Figure 4, 15% of the image was taken as training data while almost 25% of the image was considered as training data in Figure 5.

#### 4.2 Single texture classification

We experimented with images containing only one type of texture, in order to test the discrimination power of the SVM classifier. Such an exercise may not be very useful in segmentation type of problems but can be helpful when the task is to determine what texture class a particular image belongs to. In this case, nearly 8% of the proportion from each texture class was taken as the training subset. Table 1 gives simulation results for images from the Brodatz and VisTex databases.

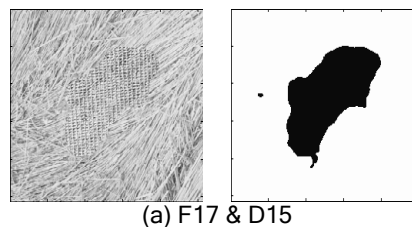
#### 4.3 Multi-texture classification

In contrast to the single texture classification, the task of multi-texture classification is relatively more challenging. It is a typical example of texture classification where the original texture classes may or may not be known *a priori*. The problem is to separate different regions in the image based on their unique textural characteristics. Similar to the single-texture case, we used nearly 8% of each textural region as the training subset and the multi-texture image is segmented with the trained SVM classifier. Note that the training subsets are taken from the original texture class and not the test images. We experimented with 2-class, 8-class, and 10-class problems, shown respectively in 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> columns of Figure 6. It turns out that the misclassification error grows with the increase in number of classes.

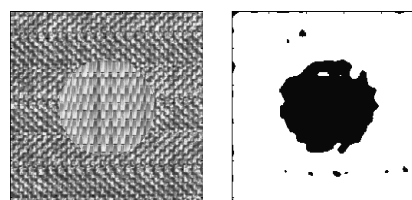
### 5. Summary & conclusions

A wavelet feature extraction based SVM texture classification method is presented. A range of experiments for testing the potential of the method are reported. Our results demonstrate that wavelets and

SVM can join hands to give a promising robust solution to both single- and multi-class texture classification problems. It shows that our method can be successfully utilized for different applications of texture analysis. Our future work will look into using automatic parameter selection for SVM, and employing DWFT and DWPT for feature extraction.

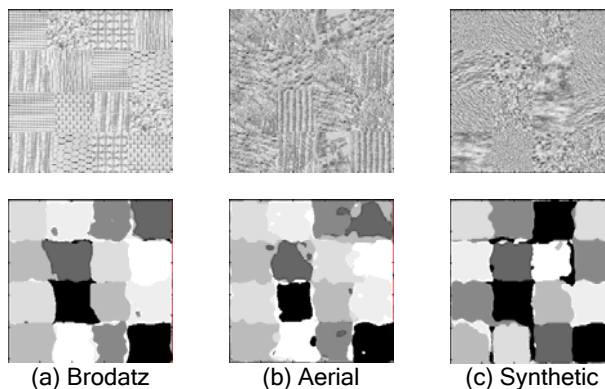


(a) F17 & D15



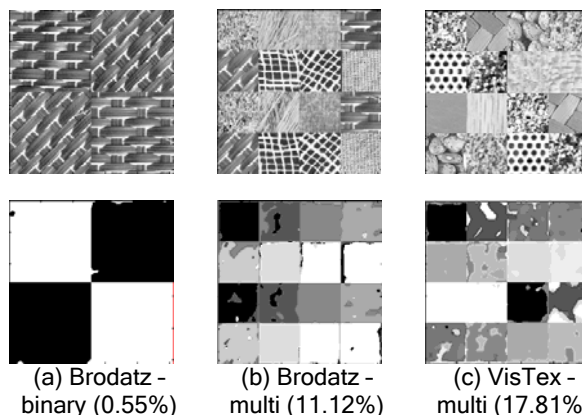
(b) D17 & D55

Figure 4: Supervised binary segmentation



(a) Brodatz (b) Aerial (c) Synthetic

Figure 5: Supervised multi segmentation



(a) Brodatz - binary (0.55%) (b) Brodatz - multi (11.12%) (c) VisTex - multi (17.81%)

Figure 6: Multi-texture classification (and misclassification error rate)

**Table 1: Single texture classification**

Texture class	Error rate (%)
D9	8.16
D15	13.20
D19	1.34
D65	13.59
D65R	11.01
D103	3.63
D103R	4.27
F17	5.62
<b>Average:</b>	<b>7.60</b>

(a) Brodatz

Texture class	Error rate (%)
Beachsand-1.2.7	6.78
Fabric.0000	28.07
Food.0000	24.90
Grass-1.1.1	26.81
Hexholes-1.5.2	5.68
Leaves.0003	23.05
Pigskin-1.1.11	5.89
Raffia-1.1.10	8.58
Water.0001	0.56
Woodgrain-1.1.9	6.19
<b>Average:</b>	<b>13.65</b>

(b) VisTex

## 6. References

- [1] T. Randen, J.H. Husoy, "Filtering for Texture Classification: A Comparative Study," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291-310, 1999.
- [2] K.I. Kim, K. Jung, S.H. Park, H.J. Kim, "Support Vector Machines for Texture Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1542-1550, 2002.
- [3] S. Li, J.T. Kwok, H. Zhu, Y. Wang, "Texture Classification using the Support Vector Machines," *Pattern Recognition*, vol. 36, no. 12, pp. 2883-2893, 2003.
- [4] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.
- [5] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 955-974, 1998.
- [6] J.P. Vert, "Introduction to Support Vector Machines and Applications to Computational Biology," *Seminar Report*, Kyoto University, Japan.
- [7] C.W Hsu, C.-C. Chang, C.-J. Lin, "A Practical Guide to Support Vector Classification," *Technical Report*, Department of Computer Science & Information Engineering, National Taiwan University, Taiwan.
- [8] <http://www.support-vector.net/>, Links & resources
- [9] <http://www.svms.org/>, Research Issues
- [10] C.W Hsu, C.-C. Chang, C.-J. Lin, "LIBSVM : a library for support vector machines," *Technical Report*, Department of Computer Science and Engineering, National Taiwan University, Taiwan.
- [11] J. Ma, Y. Zhao, S. Ahalt, "OSU SVM Classifier Matlab Toolbox," Ohio State University, USA.
- [12] R. Gonzalez, R. Woods, *Digital Image Processing*, Chapter 7, 2<sup>nd</sup> Edition (Pearson), 2002.
- [13] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, 1989.
- [14] M. Unser, "Texture Classification and Segmentation using Wavelet Frames," *IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1549-1560, 1995.
- [15] A. Laine, J. Fan, "Texture Classification by Wavelet Packet Signatures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1186-1190, 1993.
- [16] N. Rajpoot, "Local Discriminant Wavelet Packet Basis for Texture Classification," *Proceedings of SPIE Wavelets X*, San Diego, California, USA, August 2003.
- [17] V. Manian, R. Vasquez, "Scaled and Rotated Texture Classification using a Class of Basis Functions," *Pattern Recognition*, vol. 31, no. 12, pp. 1937-1948, 1998.
- [18] S. Livens, P. Scheunders, G.V. de Wouwer, D.V. Dyck, "Wavelets for Texture Analysis," *Proceedings of the 6<sup>th</sup> IEE International Conference*, 1997.
- [19] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1966.
- [20] *MIT Vision and Modeling Group*, 1998.
- [21] J.M.H. du Buf, M. Spann, M. Kardan, "Texture feature performance for image segmentation," *Pattern Recognition*, 23, pp. 291-309, 1990.
- [22] R. Haralick, K. Shangmugam, L. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 610-621, 1973.