A Robust Model for Traffic Signs Recognition Based on Support Vector Machines

Min Shi
Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway
minshi@idi.ntnu.no,

Haifeng Wu
Department of Engineering Cybernetic, Norwegian University of Science and Technology, Trondheim, Norway
haifeng@stud.ntnu.no

Hasan Fleyeh
Department of Computer Engineering, Dalarna University, Borlänge, Sweden
hfl@du.se

Abstract

Road and traffic sign recognition has been of great interest for many years. This paper presents an approach to recognize Swedish road and traffic signs by using support vector machines. We focus on recognizing seven categories of traffic sign shapes and five categories of speed limit signs. Two kinds of features, binary image and Zernike moments, are used for representing the data to the SVM for training and test. We compare and analyze the performances of the SVM recognition model using different feature representations and different kernels and SVM types through recognizing 350 traffic sign shapes and 250 speed limit signs. Experiments have shown excellent results, which have achieved 100% accuracy on sign shapes classification and 99% accuracy on speed limit signs classification.

1. Introduction

Intelligent Transportation System (ITS) integrates various technologies with traditional transportation infrastructure and vehicles, which is able to improve transportation safety and efficiency. Traffic signs on roads offer basic source of information for ITS. Automatic road and traffic sign detection and recognition, as important subtasks of ITS, collect the real-time traffic data for processing at a central facility, which have been of substantial interest for many years in the ITS research field.

Traffic signs are designed to be easily detected and recognized by humans according to their color and shapes. The colors of traffic signs are usually different from natural environments, which make them readily detectable by humans. The shapes of traffic signs, however, provide more meaningful information than colors because humans are able to recognize traffic signs without color information. In many traffic recognition systems, therefore, to classify the shapes of traffic sign is one of their main tasks.

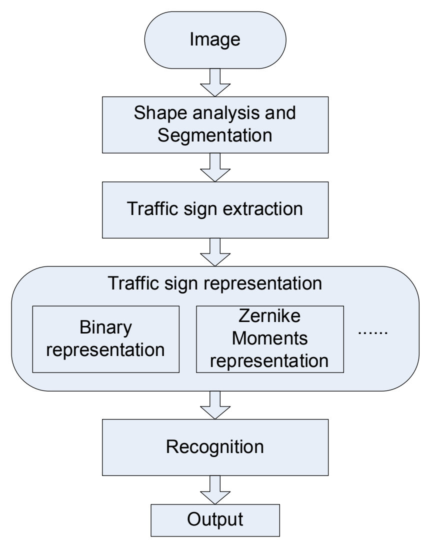


Figure 1. Block diagram of a traffic sign recognition system.

The block diagram of our recognition system is shown in figure 1. All the images of traffic signs recognized by our system are captured from various natural environments, including faded signs, damaged signs, rotated or translated signs, bad weather condition like rain, fog, snow and so on. The shapes of every traffic sign are extracted from these images. The extracted traffic signs are scaled to the same dimension and saved in binary image files. The features of a

978-0-7695-3119-9/08 $25.00 © 2008 IEEE
DOI 10.1109/CISP.2008.307

binary image are represented and input into a recognition model. The recognition result is output in the end. Before implementing an online recognition task, this recognition model has to be trained offline.

The traffic sign detection and extraction work have been done by H. Fleyeh [1]. A shadow and highlight invariant colour segmentation algorithm were used to extract traffic signs from those captured pictures. This method has shown high robustness under variance light conditions. Figure 2. shows an example of traffic signs extraction.

Figure 2. Traffic signs extraction

Figure 3. Sign shapes for recognition

Figure 4. Speed limit signs for recognition

In this work we have developed a shape-based recognition model using support vector machines (SVM). We choose Swedish traffic signs as our case study. Our tasks focus on recognizing seven categories of traffic signs (figure 3) and five speed limit signs (figure 4). We ignore some other categories, such as rectangle giving information signs, because comparing with those categories these seven categories of traffic signs are more important and more difficult to be classified by computers. All of these traffic signs are recognized only based on sign shapes. In other words, the color properties of signs are ignored during recognition.

For ITS the recognition model must be able to recognize signs invariant to rotation and translation. The traffic signs, ideally, should be perpendicular to the trajectory of the vehicle. However, to consider possible rotation and translation of traffic signs in natural environment, besides using direct binary representation, we also employ Zernike moments representation to represent the feature values of traffic signs. Zernike moments are chosen because they have some invariance properties for pattern recognition [2].

We analyze four SVM kernels and two types of SVM through training and testing the SVM model to recognize 350 sign shapes and 250 speed limit signs. Surprisingly, the basic linear kernel performs better than other kernels. Through incorporating linear kernel with either one of the two types of SVM, this model have achieved 100% accuracy on sign shape classification and 99% accuracy on speed limit signs classification. The rest of this paper is organized as follows. Section 2 describes the feature representation of traffic sign. Section 3 presents the theory of SVM. The experimental results are illustrated in section 4. Finally, conclusion is given in section 5.

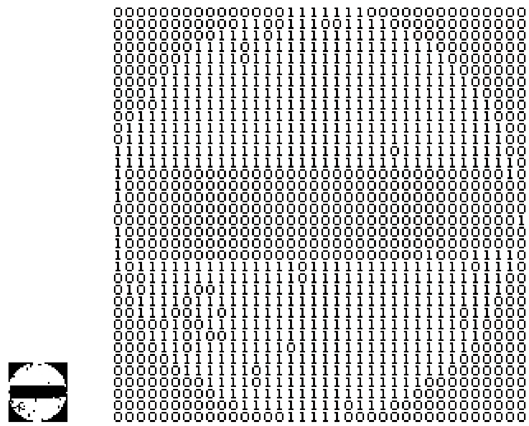
2. Feature representation of traffic sign

Each traffic sign used for recognition is represented by an N dimensional feature (i.e. attributes) vector. We have experimented with two different feature selection methods, namely, direct binary representation and Zernike moments representation.

2.1. Direct binary representation

Direct binary representation is the most simplest and straightforward method to represent a binary image. A binary image only has two possible colors, either black or white, for each pixel, where back pixel denotes value 0 and white pixel denotes value 1.

Each binary image used in this work has size 36×36 pixels, therefore there are totally 1296 attributes for one input vector, each attribute has value either 0 or 1. Figure 5 shows an example of binary representation for a no entry sign.



(a) Binary image (b) Binary representation

Figure 5. Binary representation for a no entry sign

2.2. Zernike moments representation

Moments have been widely used in computer vision applications especially in pattern recognition [3, 4]. Moments extract a set of features from an image, which will be used for performing patterns recognition tasks. A collection of moments can be computed to capture the global features of an image and used as a feature vector for pattern classification and recognition. However the number of moments that can be computed is infinite, so it is important to efficiently compute a finite subset of the moments that for discriminating between patterns. Depending on the specific application and type of patterns, different moments may be more useful than others. In this work Zernike moments are chosen to represent traffic signs.

Zernike moments, proposed by Teague, consist of a set of complex polynomials that form a complete orthogonal set over the interior of the unit circle [5]. They have been widely used in image analysis, reconstruction and recognition [2, 6]. The motivation that we choose Zernike moments to represent features of traffic signs is that Zernike moments have some important properties for pattern recognition, such as rotation invariance and noise rubustness.

The two-dimensional Zernike moments of order p with repetition q of an image intensity function $f(x, y)$ are defined as [2],

$$Z_{pq} = \frac{p+1}{\pi} \iint_{x^2+y^2 \le 1} f(x, y) V_{pq}^*(x, y) dxdy \qquad (1)$$

For a discrete image, if $f(x, y)$ is the current pixel then equation 1 can be written as,

$$Z_{pq} = \frac{p+1}{\pi} \sum_x \sum_y f(x, y) V_{pq}^*(x, y) \qquad (2)$$

where $x^2 + y^2 \le 1$, $V_{pq}^*(x, y)$ is the circular Zernike polynomials in an unit circle defined as follows,

$$V_{pq}(x, y) = R_{pq}(r_{xy}) e^{jq\theta_{xy}} \qquad (3)$$

where $r_{xy} = \sqrt{x^2 + y^2}$, $\theta_{xy} = \tan^{-1}(y/x)$, and the real-valued radial polynomial, $R_{pq}(r)$, is given as,

$$R_{pq}(r) = \sum_{k=0}^{(p-|q|/2)} (-1)^k \frac{(p-k)!}{k!(\frac{p+|q|}{2}-k)!(\frac{p-|q|}{2}-k)!} r^{p-2k} \qquad (4)$$

where $0 \le |q| \le p$ and $p - |q|$ is even.

To calculate the Zernike moments of a traffic image, first, a minimal circle that contains the sign object was defined; and then regarding the circle as a unit circle, the pixel coordinates of the traffic sign were mapped in the unit circle; finally, Zernike moments are calculated by using the coordinates achieved from the previous step.

3. SVM classification

The traffic signs are recognized in this stage according to their feature representation using SVM.

SVM are a kind of machine learning methods based on mathematical foundations of statistical learning theory, which was proposed first by Vapnik in 1992 [7]. SVM use a hypothesis space of linear functions in a high dimensional feature space, and then perform pattern recognition tasks by building decision boundaries that optimally separate the data into categories in the hypothesis space. The basic training principle of SVM is to analyze and find an optimal hyperplane such that the expected classification error for unseen test samples is minimized, i.e. good generalization. The function of the hyperplane is expressed as:

$$f(\mathrm{x}) = \sum_{i=1}^{\ell} \alpha_j y_j K(x_i, x) + b \qquad (5)$$

where x is the input vector to be classified, ℓ is the number of training samples, and $K()$ is known as kernel function. A kernel constructs an implicit mapping from the input space into a feature space, and then a linear machine is trained in the feature space to classify input vectors.

In this work we explore the performance of SVM using four different kernels:

- linear: $K(x_i, x_j) = x_i^T x_j$
- polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- radial basis function (RBF):
  $K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0$
- sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

where $\gamma, r,$ and d are kernel parameters.

For classification problems, the optimal hyperplane could not be able to separate the input vectors completely, so different classification types have been proposed for SVM [8-10]. In this work we discussed two types of SVM classification, C -support vector classification (C -SVC) and $\nu-$ support vector classification (ν -SVC).

Given training vector $x_i \in R^n$, C -SVC solves the following primal problem for binary classification $y_i \in \{-1,1\}$:

$$\mathit{Minimize}_{\xi, \mathrm{w}, b} \quad \frac{1}{2}\left\langle \mathrm{w}^{\mathrm{T}} \mathrm{w} \right\rangle + C \sum_{i=1}^{\ell} \xi_i$$

$$\mathit{Subject\ to} \quad y_i(\mathrm{w}^{\mathrm{T}} \phi(\mathrm{x}_i) + b) + \xi_i \geq 1$$
$$\xi_i \geq 0, i = 1, \ldots, \ell$$

where w is the optimal hyperplane, slack variable ξ_i allows some data to be misclassified, and C is an priori constant, which gives a trade-off between maximum margin and classification error.

The ν -SVC uses a parameter ν that is able to control the number of support vectors and errors. Its primal form is:

$$\mathit{Minimize}_{\xi, \mathrm{w}, b} \quad \frac{1}{2}\left\langle \mathrm{w}^{\mathrm{T}} \mathrm{w} \right\rangle - \nu\rho + \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i$$

$$\mathit{Subject\ to} \quad y_i(\mathrm{w}^{\mathrm{T}} \phi(\mathrm{x}_i) + b) + \xi_i \geq \rho$$
$$\xi_i \geq 0, i = 1, \ldots, \ell, \rho \geq 0$$

where the parameter $\nu \in (0,1]$ is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors.

The SVM are well known to have a good generalization and overcome the curse of dimension in both computation and generalization [11]. The later characteristic is highly beneficial to perform pattern recognition tasks without data preprocessing, such as traffic sign recognition using direct binary representation. The goal of analyzing different kernels and types of SVM is to find the best model of SVM to perform classification tasks in our traffic recognition system.

4. Experimental results

The traffic sign recognition model has been developed based on the LIBSVM library and implemented in C++ language [12]. The library source codes have been modified and some methods have been encapsulated so that the library can be reuse and extended more easily.

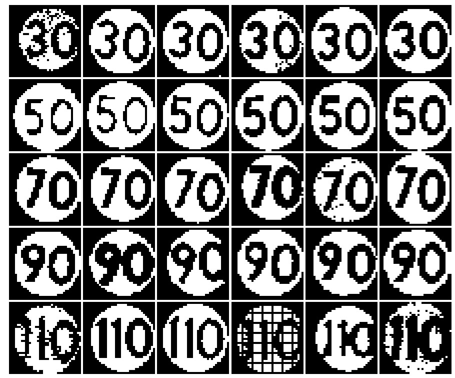


Figure 6. Part samples of speed limit signs

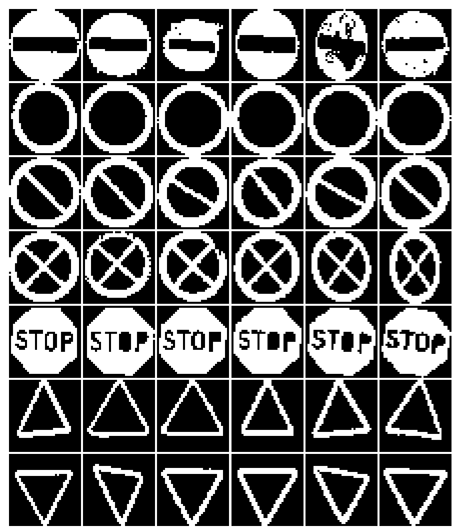


Figure 7. Part samples of sign shapes

The experimental data used for training and test the SVM recognition model were selected from a binary image database of traffic sign, which consisted of 600 data samples. The data samples contained five categories of speed limit signs (figure 6) and seven categories of sign shapes (figure 7). Each category had 50 samples that were divided randomly into two data sets, 30 for training and 20 for test, in every experiment.

The experiments were divided into two parts. The first part was to train and test the SVM recognition model using two features representation of traffic sign, direct binary representation and Zernike moments representation. The second part analyzed the SVM model using different kernels and SVM types. The following subsections describe the experimental set up in detail.

4.1. Classification using different features representation

The data represented by binary representation contained seven categories of sign shapes and five categories for speed limit signs. Zernike moments, however, can only represent six categories for road sign shapes, because they have the property of rotation invariance, which means it does not distinguish between upward triangles and downward triangles.

Ten pairs of training/test data sets were created for every classification task. Each pair of data set was selected randomly without repetition from the data samples. These ten pairs of data sets were inputted into the SVM model for training and test. Linear kernel and C -SVM type were chosen as the SVM model parameters for this experiment, where the parameter C of C -SVM type was set to 1. Table 1 and table 2 shows the average results on classification of traffic sign shapes and speed limit signs.

Table 1. Average performances on classification of traffic sign shapes using different features

Features	Train	Test
Binary	100%	100%
Zernike moments	100%	98.33%

Table 2. Average performances on classification of speed limit signs using different features

Features	Train	Test
Binary	100%	99%
Zernike moments	99.13%	85%

As seen in the above tables, the performances for sign shapes classification using binary representation were identical and extremely high, achieving 100% accuracy, on all the pairs of data sets. None of these instances was misclassified for both training and test. While for the classification of speed limit signs using binary representation, the average performances were 100% accuracy on training, but 99% accuracy on test. Normally those traffic sign images that were misclassified have very poor image quality. Table 3 shows two instances of the poor quality images. They were not the only two instances of poor quality included in our data sets, but the two instances that were misclassified most frequently. In the above experiments, the first instance was misclassified frequently as speed limit sign 70 and the second one was misclassified frequently as speed limit 30.

Table 3. Two instances of traffic sign images that were misclassified using binary representation

Road Sign Images	Misclassified As
	Speed Limit Sign 70
	Speed Limit Sign 30

Intuitively, the number 30 is more similar to the number 50 than to the number 70. In the first case, however, it was misclassified frequently as the number 70. One of the reasons could be there were much pepper noises around the number 30 and some instances of speed limit signs 70 had the similar feature, as shown in figure 8. In the other case, the number 50 was damaged badly, which caused trouble to classification. Despite the noises and damages, the performances of SVM model still achieved 100% accuracy on the classification of speed limit signs using binary representation for all pairs of training sets.

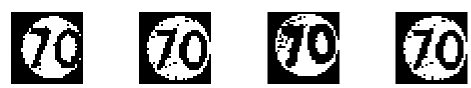


Figure 8. Some instances of speed limit signs 70 have the feature of pepper noise

Comparing the performances of SVM model using different features, obviously the SVM recognition model using Zernike moments representation did not work as well as that using binary representation. Zernike moments extract the features from binary images, which reduced the dimension of the input vectors. The choice of Zernike moments parameters is a key factor for efficiently computing and discriminating between patterns. The low order moments represent the global shape of a pattern and the higher order the details. However, higher order increases computational cost.

In the above experiments, the (p,q) of Zernike moments were defined from (5, 1) to (12, 12), which contained 40 attributes for each input vector. Using these Zernike moments representation some sign shapes were a little difficult to be distinguished from others. For speed limit signs, especially, the similar shapes enabled themselves to be misclassified more frequently when the SVM model were trained and tested using Zernike moments representation.

4.2. Classification using different kernels and SVM types

The experiments performed in the first part chose the simplest linear kernel and C-SVM type to train and test the SVM model. This part focused on the analysis of performances using different kernels and SVM types.

The SVM model were trained respectively using four basic kernels, linear, polynomial, RBF and sigmoid, and two kinds of SVM type, C-SVM and ν-SVM. Four group experiments were performed; each group employed the same pair of training/test data set. Other parameters of the SVM model were given as follows: $C=1$, $\nu=0.5$, $\gamma=1/n$, $r=0$, $d=3$, where n is the number of attributes for an input vector.

Table 4 and table 5 show the experimental results that trained and tested the SVM model using binary representation for the traffic sign shapes classification and speed limit signs classification.

Table 6 and table 7 show the experimental results that trained and tested the SVM model using Zernike moments representation for traffic sign shapes classification and speed limit signs classification. Lots of experiments revealed that linear kernel combined with C-SVM had better performances than others. One of the reasons could be that linear kernel normally has good performances when the number of input attributes is big. The leading cause, however, is that the accuracy of a SVM model highly depends on the selection of the model parameters.

There are no any parameters in linear kernel, but three, one, two parameters respectively in polynomial, RBF and sigmoid kernels. In addition, C and ν are two important parameters of SVM types. To analyze the performances of selecting different value of parameters, the above pair of training/test data set of speed limit signs using Zernike moments representation were chosen to train and test the SVM model in the following experiments.

Table 4. The performances of traffic sign shapes classification using different kernels and SVM types with binary representation

SVM Type	Kernel	Train	Test
C-SVM	Linear	100%	100%
	Polynomial	100%	97.86%
	RBF	100%	100%
	Sigmoid	100%	99.29%
ν-SVM	Linear	100%	100%
	Polynomial	100%	97.86%
	RBF	100%	100%
	Sigmoid	99.52%	100%

Table 5. The performances of speed limit signs classification using different kernels and SVM types with binary representation

SVM Type	Kernel	Train	Test
C-SVM	Linear	100%	98%
	Polynomial	100%	96%
	RBF	100%	97%
	Sigmoid	99.33%	97%
ν-SVM	Linear	100%	98%
	Polynomial	100%	96%
	RBF	100%	97%
	Sigmoid	100%	98%

Table 6. The performances of traffic sign shapes classification using different kernels and SVM types with ZM representation

SVM Type	Kernel	Train	Test
C-SVM	Linear	100%	100%
	Polynomial	87.22%	85.83%
	RBF	98.89%	99.17%
	Sigmoid	96.67%	99.17%
ν-SVM	Linear	98.33%	99.17%
	Polynomial	98.33%	97.5%
	RBF	98.33%	99.17%
	Sigmoid	98.33%	99.17%

Table 7. The performances of speed limit signs classification using different kernels and SVM types with ZM representation

SVM Type	Kernel	Train	Test
C-SVM	Linear	100%	82%
	Polynomial	70%	56%
	RBF	89.33%	72%
	Sigmoid	74%	68%
ν-SVM	Linear	93.33%	78%
	Polynomial	93.33%	85%
	RBF	94%	79%
	Sigmoid	93.33%	76%

The parameter C defines the upper bound of α in C-SVM model; it is a trade-off between maximum margin and classification error. Figure 9 shows the performances of the SVM model using linear kernel and C-SVM type with different parameter C. There is no kernel parameter in linear kernel, so the parameter C is the only variable in this model. A higher value of the parameter C allows α having a large value. The accuracy of classification was improved by increasing the value of α. However, it was unuseful to define an excessive upper bound of α.

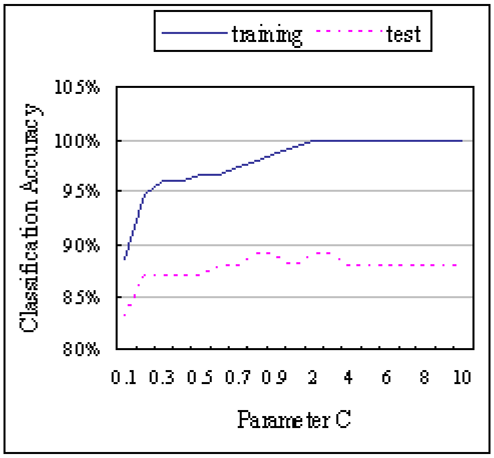


Figure 9. The performances of the SVM model using linear kernel and different parameter C

The ν-SVM uses another parameter ν to control the number of margin error and support vectors. Figure 10 and figure 11 shows the performances of SVM model using linear kernel and ν-SVM type with different parameter ν. The number of support vectors was increased by increasing the value of the parameter ν. However, since the number of margin error was increased also, the classification accuracy of training and test was decreased on the whole.

There is one parameter γ in RBF kernel; the parameter γ is normally given a very small value. In all the above experiments, it was initialized as $\gamma = 1/n$, where n was the attributes number of an input vector. Figure 12 illustrates the performances with different values of γ. The training classification was improved by increasing the value of γ. The generalization of SVM model, however, was decreased if the value of γ exceeds a certain value. In the above experiments the number of attributes using Zernike moments representation for an input vector was 40, so the initialized value of γ was 0.025. Obviously, for this experiment the best choice of γ was approximate to 0.1. Still, the default value $\gamma = 1/n$ was not a bad choice in this case.

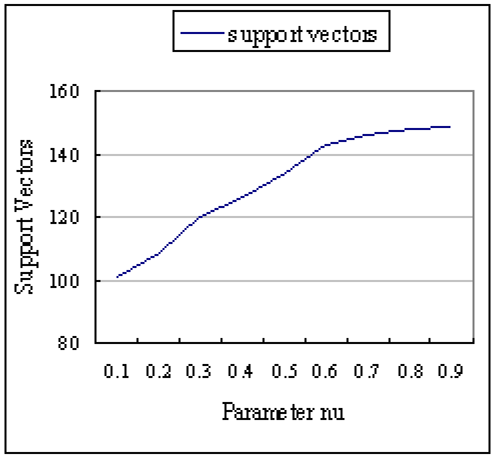


Figure 10. The Support Vectors of the SVM model using linear kernel and different parameter ν

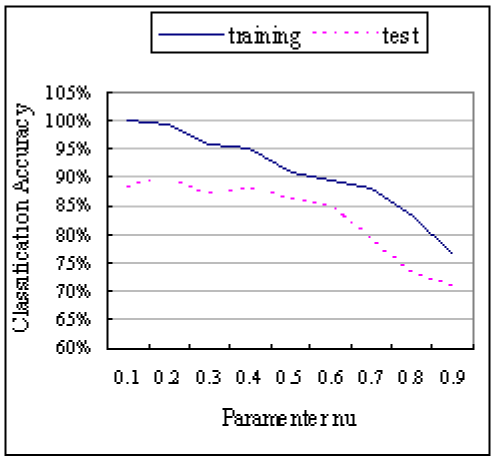


Figure 11. The performances of the SVM model using linear kernel and different parameter ν

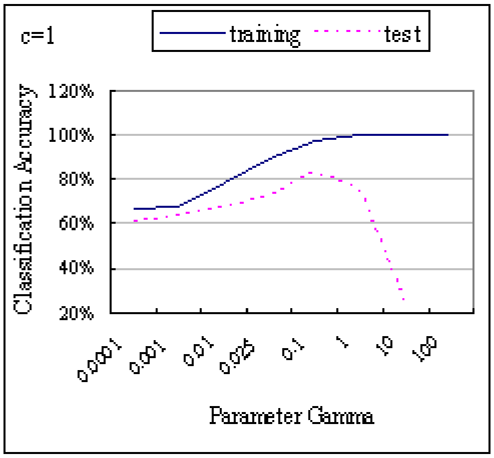


Figure 12. The performances of the SVM model using RBF kernel and C-SVM with different parameter γ

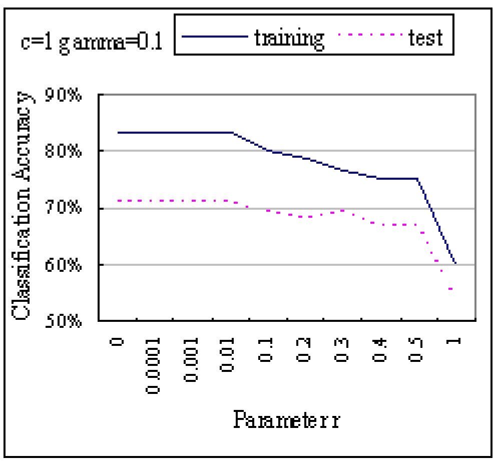


Figure 13. The performances of the SVM model using sigmoid kernel and C-SVM with different parameter r

Besides the parameter γ, sigmoid kernel has another parameter r. The parameter r is a coefficient presented in sigmoid kernel and polynomial kernel. It is not a very important parameter and normally is initialized as zero. Figure 13 shows the performances with different values of r. As we can see that the accuracy of classification was decreased by increasing value of r.

There are three kernel parameters in polynomial kernel, γ, r and d. Figure 14 shows the performances with different values of d. Normally smaller values of d are better choices. On the whole, the generalization of the SVM model was decreased with the increase of d.

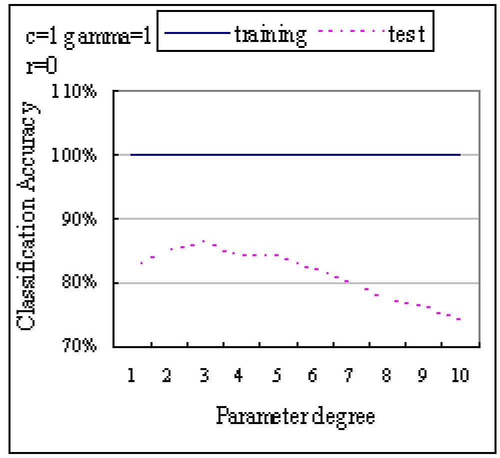


Figure 14. The performances of the SVM model using polynomial kernel and C-SVM with different parameter d

Actually, the change of one parameter will affect the performances of other parameters if a SVM model has more than one model parameters. The more the number of model parameters the bigger the search space to find optimal parameters for SVM. In this work, the SVM model using the linear kernel combined with either C-SVM or ν-SVM has already shown remarkable results.

5. Conclusion

This paper presents a robust model to recognize traffic signs using SVM. We focus on classifying seven categories of traffic sign shapes and five categories speed limit signs by using this model.

Two kinds of features were used for representing each sign. Despite that Zernike moments have many beneficial properties for pattern recognition [2], direct binary representation outperformed Zernike moments in our case study. The dimension of input vector using binary representation can be unconsidered because SVM have overcome the curse of dimension in both computation and generalization.

Four different kernels and two SVM types were analyzed in this work. The main purpose is to find an optimal SVM recognition model for our traffic recognition system. The experimental results have shown that the linear kernel works more efficiently than other kernels, no matter it combined with either C-SVM or ν-SVM. This resultant recognition system has shown excellent results for recognizing a wide variety of traffic signs including noises, damages, rotation and translations.

References

[1] H. Fleyeh, "Shadow And Highlight Invariant Colour Segmentation Algorithm For Traffic Signs," in IEEE Conference on Cybernetics and Intelligent Systems. Thailand, 2006.

[2] A. Khotanzad and Y. H. Hong, "Invariant Image Recognition by Zernike Moments," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 489-497, 1990.

[3] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IEEE Transactions on Information Theory*, vol. 8, pp. 179 - 187 1962.

[4] R. J. Prokop and A. P. Reeves, "A Survey of Moment-Based Techniques For Unoccluded Object Representation and Recognition," *CVGIP - Graphical Models and Image Processing*, vol. 54, pp. 438-460, 1992.

[5] M. Teague, "Image analysis via the general theory of moments," *J. Opt Soc. Am.*, vol. 70 (8), pp. 920–930, 1980.

[6] A. Padilla-Vivanco, A. Martínez-Ramírez, and F. Granados-Agustín, Digital image reconstruction by using Zernike moments.

[7] N. Cristianini and J. Shawe-Taylor, *AN INTRODUCTION TO SUPPORT VECTOR MACHINES*, vol. Preface: Cambridge University Press, 2000.

[8] C. Cortes and V. Vapnik, "Support-Vector Network," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[9] V. Vapnik, *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[10] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New Support Vector Algorithms," *Neural Computation*, vol. 12, pp. 1207-1245, 2000.

[11] N. Cristianini and J. Shawe-Taylor, *AN INTRODUCTION TO SUPPORT VECTOR MACHINES*, vol. Chapter 4: Cambridge University Press, 2000.

[12] C.-C. Chang and C.-J. Lin, LIBSVM: a Library for Support Vector Machines.