

# Comparison of SVMs in Number Plate Recognition

Lihong Zheng, Xiangjian He and Tom Hintz

University of Technology, Sydney, Department of Computer Systems,  
{lzheng, sean, hintz}@it.uts.edu.au

**Abstract.** High accuracy and high speed are two key issues to consider in automatic number plate recognition (ANPR). In this paper, we construct a recognition method based on Support Vector Machines (SVMs) for ANPR. Firstly, we briefly review some knowledge of SVMs. Then, the number plate recognition algorithm is proposed. The algorithm starts from a collection of samples of characters. The characters in the number plates are divided into two kinds, namely digits and letters. Each character is recognized by an SVM, which is trained by some known samples in advance. In order to improve recognition accuracy, two approaches of SVMs are applied and compared. Experimental results based on two algorithms of SVMs are given. From the experimental results, we can make the conclusion that ‘one against one’ method based on RBF kernel is better than others such as inductive learning-based or ‘one against all’ method for automatic number plate recognition.

## 1 Introduction

Number recognition is playing an important role in image processing field. For example, there are thousands of containers and trucks need to be registered every day at container terminals and depots. Normally, this registration will be done manually. However, this is not only prone to error but also slow to meet the increasing volume of containers and trucks. Hence, an automatic, fast and precise number recognition process is required.

The fundamental issues in number plate recognition are the requirements of high accuracy and high recognition speed. Since last two decades, various commercial ANPR products (Zheng, He and Li 2005) around the world are available, such as SeeCar in Israel, VECON in Hongkong, LPR in USA, the ANPR in UK, IMPS in Singapore, and the CARINA in Hungary (Zheng 2005). Even though there have been so many successful ANPR systems, there are still several problems for character recognition of number plates. The following three problems are the most critical. Firstly, the recognition system must be able to handle various sizes, fonts, spaces and alignments of the characters in the number plates. Secondly, the recognition system

must be robust to changes in illumination and colors used. Thirdly, the recognition system must be able to distinguish the obscured characters in real-life images due to rust, mud, peeling paint, and fading color. To resolve the problems above, an effective method must have a general adaptability to different conditions. It should have good tolerance for noise and classify and recognize the characters in number plate accurately and credibly.

In order to improve the performance of recognition, an algorithm on number recognition was proposed in (Aksoy, Cagil and Turker 2000) based on RULES-3 induction theory. This algorithm trains character samples and obtains the rules that are used to recognize the numbers on number plates. One advantage of using this method is that the recognition speed is much quicker in number recognition. But it is not robust to image rotation, translation and scaling. However, it cannot distinguish digits 6 and 9 without additional observation.

In order to improve the recognition performance, we propose another algorithm to number recognition (Zheng and He 2006). This technique uses a Support Vector Machine (SVM) to train character samples and obtain the rules that are used to recognize the numbers on number plates. SVM (Cristianini 2000; Vapnik 1999) is forcefully competing with many methods for pattern classification. An SVM is a supervised learning technique first discussed by Vapnik (Vapnik 1999). SVM takes Statistical Learning Theory (SLT) as its theoretical foundation, and the structural risk minimization as its optimal object to realize the best generalization. They are based on some simple ideas and provide a clear intuition of what learning from examples is all about. More importantly, they possess the feature of high performance in practical applications. From 1960s to present, SVMs become more and more important in the field of pattern recognition.

The organization of this paper is as follows. We first introduce some basic knowledge of SVMs in Section 2. In Section 3, multi-class classifier model and ‘one against all’ and ‘one against one’ strategy are briefly introduced. The algorithm of number plate recognition is done in Section 4. The experimental results for number recognition are demonstrated in Section 5. We conclude in Section 6.

## 2 Principles of SVMs

In 2000, SVM was defined by Cristianini & Taylor (Cristianini and Shawe-Taylor 2000) as ‘a system for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalization theory and exploiting optimization theory’. An SVM is a pattern recognizer that classifies data without making any assumptions about the underlying process by which the observations were granted. The SVMs use hyperplanes to separate the different classes. Many hyperplanes are fitted to separate the classes, but there is only one optimal separating hyperplane. The optimal one is expected to generalize well in comparison to the others. The optimal hyperplane is determined only by support vectors, which are ideally distributed near class boundaries. The hyperplane is constructed so as to maximize a measure of the ‘margin’ between classes. A new data sample is classified by the SVM according to the decision boundary defined by the hyperplane.

An SVM corresponds to a linear method in a very high dimensional feature space. The feature space is nonlinearly related to the input space. Classification is achieved by realizing a linear or non-linear separation surface in the feature space (Vapnik 1999).

We briefly describe general knowledge of SVMs as follows (Zheng and He 2006).

Given a two-class classification problem, separating hyperplanes can be defined as:

$$H_{\tilde{w},b} : \tilde{w}^T \tilde{x} + b = 0,$$

where  $w$  is a normal vector, the input is denoted by  $x$  and  $b$  is an offset. SVM tries to find the optimal hyperplane via maximizing the margin between the positive input vectors,  $\{x_i \text{ when } y_i=+1, \text{ for } i=1, \dots, n\}$ , and negative input vectors,  $\{x_i \text{ when } y_i=-1, \text{ for } i=1, \dots, n\}$ .

In the linear case, this is equivalent to maximize  $2/\|\tilde{w}\|$  ( $\|\cdot\|$  is norm of  $\tilde{w}$ ) that is regarded as a canonical representation of the separating hyperplane, i.e.,

$$\begin{cases} \min & \frac{1}{2} \|\tilde{w}\|^2 \\ \text{s.t.} & y_i (\langle \tilde{w}, \tilde{x}_i \rangle + b) \geq 1, \quad \forall i \end{cases} \quad (1)$$

Here  $\tilde{w}$  can be solved as follows by applying the Lagrangian multiplier  $\alpha$ .

$$\tilde{w} = \sum_{i=1}^n \alpha_i y_i \phi(x_i)$$

where  $\alpha_i \geq 0$ , ( $i = 1, 2, \dots, n$ ), is the Lagrangian multiplier, and  $\phi$  is the kernel function.

For a new input, its classified label is according to the result of:

$$f_{H_{\tilde{w},b}}(x) = \text{sgn}(\tilde{w}^T \phi(x) + b) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i K(x, x_i) + b\right),$$

where  $K(x, x_i) = \phi(x)^T \phi(x_i)$ .

In the case that the set is not linearly separable or does not satisfy the inequality constraint  $y_i (\langle \tilde{w}, x_i \rangle + b) \geq 1$ , for all  $i$ , a slack and nonnegative variable  $\xi$  is added into Eq. 1 as shown by

$$\begin{cases} \min & \frac{1}{2} \|\tilde{w}\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} & y_i (\tilde{w}^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, \quad i = 1, \dots, n \end{cases} \quad (2)$$

The term  $\sum_{i=1}^n \xi_i$  is an upper bound on the number of misclassification in the training set. It indicates the distance that the training point from the optimal hyperplane and the amount of violation of the constraints. Furthermore,  $C$  is the penalty term for misclassifications.  $C$  controls the trade-off between maximizing the margin and minimizing the training error, and between a better generalization and an efficient computation.

### 3 Multi-class Model of SVMs

Among many classification methods, SVM has demonstrated superior performance. It has been successfully utilized in handwritten numeral recognition. However, SVM was originally designed for binary classification, and its extension to solve multi-class problems is not straightforward. The popular methods for applying SVM to multi-class problems decompose a multi-class problem into many binary-class problems and incorporate many binary-class SVMs.

Two main approaches have been suggested for applying SVMs for multi-class classification (Foody and Mathur 2004). In each approach, the underlying basis has been to reduce the multi-class problem to a set of binary problems, and to enable the use of basic SVM.

The first approach, called 'one against all' (Foody and Mathur 2004; Dong, Suen and Krzyzak 2005), uses a set of binary classifiers, each trained to separate one class from the rest. For a given input  $x_i$ , there are  $k$  decision functions.  $x_i$  is classified to be in the one of  $k$  classes that gives the largest decision value.

The second approach is called 'one against one'. In this approach, a series of classifiers are applied to each pair of classes, and only the label of the most commonly computed class is kept for each case. The application of this method requires  $k(k-1)/2$  classifiers or machines be applied to each pair of classes, and a strategy to handle instances in which an equal number of votes are derived for more than one class for a case. Once all  $k(k-1)/2$  classifiers have been undertaken, the max-win strategy is followed.

The multi-class model can be described as follows.

Given  $n$  training data

$$\Omega = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \mid x_i \in R^n, (i = 1, 2, \dots, n)\}, \text{ and } y_i \in \{1, 2, 3, \dots, k\},$$

where  $k$  is the number of classes. The classification function is as:

$$\left\{ \begin{array}{l} \min_{w^i, b^i, \xi^i} \quad \frac{1}{2} \|w_j^i\|^2 + C \sum_{j=1}^n \xi_j^i \\ s.t. \quad (w^i)^T \phi(x_i) + b^i \geq 1 - \xi_j^i, \text{ if } y_j = i, \\ (w^i)^T \phi(x_i) + b^i \leq -1 + \xi_j^i, \text{ if } y_j \neq i, \\ \xi_j^i \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

where  $K(x, x_i) = \phi(x)^T \phi(x_i)$

In APRN,  $k$  is 36, which includes 10 for digits and 26 for letters. The above formula implies the following 36 decision functions for all 36 digits and letters:

$$\begin{aligned} &(w^1)^T \phi(x) + b^1, \\ &\dots \\ &(w^{36})^T \phi(x) + b^{36}. \end{aligned}$$

An  $x$  is classified to be the digit or letter  $a$  if its decision function gives the maximum value in the SVM for  $a$ , i.e.,

$$\text{Class of } x \equiv \arg \max_{i=1, \dots, 36} ((w^i)^T \phi(x) + b^i).$$



Fig.1. The number plate samples

#### 4 Number Plate Classifier Design

The car number plate at the New South Wales state of Australia has up to six characters as shown in Fig. 1. Usually, the number plate consists of two main sections. The upper section contains main information of the number plate, and the lower part is for the name of the state. The upper part is more important, and is separated into two groups of characters. The first group usually consists of three or four letters of A to Z and the second group consists of three or two digits of 0 to 9. In order to speed up the process, two sets of SVMs are designed according to these two groups of characters. One set of SVMs is designed for recognizing digital numbers and the other one is designed for letters. The details of our algorithms are described as follows. For comparison, the ‘one against all’ and ‘one against one’ methods are both adopted.

In the first approach using ‘one against all’ method, for recognizing the digits in a number plate, ten SVMs are designed for the ten digits from 0 to 9. Each SVM has one digital number sample as one label and all or some of the other samples are as another label. After training, each SVM gets its own values of parameters. The decision value of the testing sample will be calculated based on the values of parameters obtained. The final recognition result will be achieved according to the class that gives the maximum decision value. The procedure for recognizing the letters in a number plate is the same as that for digits except that the total number of SVMs is 26 for 26 letters.

In the second approach using ‘one against one’ method, SVM has one digital number sample as one label and any one of the other samples is taken as another label. Therefore, 45 SVMs are designed for the ten digits from 0 to 9, and 325 SVMs are for letter A to Z.

We summarize the SVM based algorithm for number recognition in this paper as follows. In order to recognize a number plate, we go through the following steps.

- Step 1. Pre-process the image of number plate.
- Step 2. Segment the image into several parts of which each contains only a single character.
- Step 3. Normalize each letter or digit on the number plate.
- Step 4. Extract the feature vector of each normalized candidate
- Step 5. Recognizes the single character (a digit or a letter) by the set of SVMs trained in advance.
- Step 6. If there are no more unclassified samples, then STOP. Otherwise, go to Step 5.
- Step 7. Add these test samples into their corresponding database for further training.
- Step 8. Recognize number plate by bringing all characters used together.

When a number plate region is located and extracted, the histogram projection methods are applied for character segmentation. The number plate is segmented and the sub-images containing individual characters (digits and letters) forming the number plate are obtained. In the pre-processing step, each sub-image of a character is normalized into a certain size which is 20 pixels in width and 36 pixels in length. Then the sub-image is binarized into range of  $[-1, +1]$  for enhancing the character from background. The support vectors are calculated directly from the binarized sub-images. The high dimensional feature vectors are stored into two kinds of database, one is for digital numbers, and the other is for letters. The above feature vectors are used to train SVMs with RBF kernel (see Section 5). In our experiments, 720 dimensional feature vectors are input into SVMs, which have been trained successfully. Then, which character that a given candidate should be can be obtained in according to the outputs of SVMs.

When all digits and letters on a number plate are recognized (or classified), the recognition of the number plate is complete.

### 5 Experimental Results

Support vector machines in our experiments are trained using algorithms as shown in (Gunn 1997). Based on the approach we described above, we did experiments for digital numbers of 0 to 9 and letters of A to Z. In our database, there are average 768 training samples for character which are segmented from real images of number plates. Figure 2 presents some of example of characters in number plates. We selected randomly one third of them for training and the rest samples were used for testing.



Fig.2. Segmented characters

The experimental results are based on two methods, namely ‘one against all’ and ‘one against one’. Two kernel functions that are linear kernel and RBF kernel are used and shown below.

Linear:  $K(x, x_i) = \langle x \cdot x_i \rangle$

RBF:  $K(x, x_i) = \exp(-\|x - x_i\|^2 / 2\sigma^2)$

Tables 5.1 and 5.2 show a comparison of using the two methods. Also, we estimate the matching rate using different kernel parameters  $\sigma$  and cost parameters  $C$ . Matching rate = Number of recognized characters correctly/Number of all testing characters.

Table 5.1 The experimental results of characters (Digits and Letters) of number plate (One against all)

	Digital Numbers		Letters	
	RBF kernel ( $C=10^{-2} \sim 10^4$ , $\sigma=1 \sim 1.0$ )	Linear Kernel ( $C=10^{-2} \sim 10^4$ )	RBF kernel ( $C=10^{-2} \sim 10^4$ , $\sigma=1 \sim 1.0$ )	Linear Kernel ( $C=10^{-2} \sim 10^4$ )
Matching rate	83.8%	75.7%	80.7%	65.3%
Percentage of SV	98%	14.3%	98.8%	11.5%
Training Time(seconds)	0.46	0.5	6.1	7.2
Testing time	0.6	1.88	0.31	0.56

We also report the training time, testing time and the percentage of support vectors in the tables. All the experiments are performed on a Pentium 4 PC with 2.0GHz CPU. The training time and testing time increase with the number of training sam-

ples. However, the classification accuracy does not change much. For further comparison, we also give the experimental results as shown in Table 5.3 obtained from well-known database iris and UCI (UCI).

**Table 5.2.** The experimental results of characters (Digits and Letters) of number plate (One against one)

	Digital Numbers		Letters
	RBF kernel ( $C=10^{-2} \sim 10^4$ , $\sigma=.1 \sim 1.0$ )	Linear Kernel ( $C=10^{-2} \sim 10^4$ )	RBF kernel ( $C=10^{-2} \sim 10^3$ , $\sigma=.5 \sim 1.0$ )
Matching rate	68.8%	70.6%	63%
Percentage of SV	98%	14.3%	98.8%
Training Time(seconds)	0.02	0.01	4.3
Testing time	1.88	0.5	0.5

**Table 5.3.** The experimental results of iris and UCI database (RBF) (One against all)

Cases	Test matching rate $C=(500 \sim 10^3)$ $\sigma=.5 \sim 1$	Percentage of SV	Training Time(s)	Testing time
Iris	97.8%	16%	0.1	0.01
UCI	89.86%	29.3%	63.8	0.4

## 6 Discussion and Conclusions

The major advantages of SVMs are that each SVM is a maximal margin hyperplane in a feature space built using a Kernel function, and each SVM is based on firm statistical and mathematical foundations concerning generalization and optimization theory. The training for SVMs is relatively easy. From the experimental results, it is obvious that SVMs based on RBF kernel function perform better due to its properties described in above section. The algorithm based on ‘one against all’ gets higher matching rate than method of one again one.

Due to noise contained in the image of real number plates, the recognition rate is lower than what obtained in some standard database such as *iris* (Gunn 1997) and UCI (UCI). But the following conclusion still holds. In ‘one against one method’, each classifier must give a label to a candidate no matter if it is correct or not. Therefore, in many cases, error label information is given and data are mistrained. The parameters after training have lower credit. On the contrary, however, ‘one against all’ method shows better performance.

For the failed cases in our experiment, we notice that the amounts of every character’s samples are not evenly in our database. For example, character ‘A’ owned much more samples than other characters. Characters ‘H’ and ‘L’ have smaller num-



ber of samples in our database. The parameters obtained through training are less powerful than others which were trained using a big amount of samples. Another reason is that the images of these characters are much more blurred or distorted than the training samples. These characters are misclassified into other similar classes. However, compared with earlier results using inductive Rule3 (Zheng, He, Wu and Hintz 2006) where the recognition accuracy rate is 71%, accuracy rates obtained using SVM is competitive and better.

Having said all above, SVMs can be applied in number plate recognition successfully especially for heavier noisy characters. Since SVM has the highest classification accuracy as a binary classifier, for further improvement of matching rate, we should combine some other classifiers together to make the number of characters in a group as small as possible. Therefore, the overall matching rate will be definitely higher than other methods for number plate recognition.

## References

- Aksoy, M. S., Cagil, G. and Turker, A. K. (2000) Number-plate recognition using inductive learning. *Robotics and Autonomous Systems*, Elsevier, Vol.33, pp.149-153.
- Cristianini, N. and Shawe-Taylor, J. (2000) An introduction to support vector machines and other kernel-based learning methods. *Cambridge University Press*.
- Dong, J., Suen, C.Y. and Krzyzak, A. (2005) Algorithms of fast SVM evaluation based on subspace projection. *2005 IEEE International Joint Conference on Neural Networks*, Vol. 2(31), pp.865-870.
- Foody, G.M. and Mathur, A. (2004) A relative evaluation of multiclass image classification by support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, Vol.42(6), pp.1335-1343.
- Gunn, S. R. (1997) Support vector machines for classification and regression. *Technical report*. Image Speech and Intelligent Systems Research Group, University of Southampton. UCI [www.kernel machines\data/UCL.html](http://www.kernel.machines\data/UCL.html)
- Vapnik, V. N. (1999) The nature of statistical learning theory. New York: *Springer*.
- Zheng, L., He, X. and Li, Y. (2005) A comparison of methods for character recognition of car number plates. *Proc. of International Conference on Computer Vision (VISION'05)*, Las Vegas, pp.33-39.
- Zheng, L. and He, X. (2006) Number plate recognition based on support vector machines. *Proceeding of IEEE AVSS 2006 conference*. ISBN-13: 978-0-7695-2688-1.
- Zheng, L., He, X., Wu, Q. and Hintz, T. (2006) Learning-based number recognition on Spiral Architecture. *Proceeding of IEEE ICARCV2006*. Singapore, pp.897-901.