

© [2009] IEEE. Reprinted, with permission, from L. Zheng., X. He., Q.Wu., W. Jia., B. Samali and M. Palaniswami. 2008, A hierarchically combined classifier for license plate recognition, IEEE 8th International Conference on Computer and Information Technology (CIT2008), pp 372-377. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Technology, Sydney's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

A Hierarchically Combined Classifier for License Plate Recognition

L. Zheng¹, X. He¹, Q. Wu¹, W. Jia¹, B. Samali¹ and M. Palaniswami²

¹*Faculty of Engineer and Information Technology, University of Technology, Sydney
PO Box 123, Broadway NSW 2007 AUSTRALIA*

E-mails: {lzheng, sean, wuq, wejia}@it.uts.edu.au, bijian@eng.uts.edu.au

²*Department of Electrical and Electronic Engineering,
The University of Melbourne, Victoria, 3010 AUSTRALIA
E-mail: swami@ee.unimelb.edu.au*

Abstract

High accuracy and fast recognition speed are two requirements for real-time and automatic license plate recognition system. In this paper, we propose a hierarchically combined classifier based on an Inductive Learning Based Method and an SVM-based classification. This approach employs the inductive learning based method to roughly divide all classes into smaller groups. Then the SVM method is used for character classification in individual groups. Both start from a collection of samples of characters from license plates. After a training process using some known samples in advance, the inductive learning rules are extracted for rough classification and the parameters used for SVM-based classification are obtained. Then, a classification tree is constructed for further fast training and testing processes for SVM-based classification. Experimental results for the proposed approach are given. From the experimental results, we can make the conclusion that the hierarchically combined classifier is better than either the inductive learning based classification or the SVM-based classification in terms of error rates and processing speeds.

1. Introduction

A license plate is a unique identification of a vehicle. The fundamental issues in real time license plate recognition are the accuracy and the recognition speed. Various commercial ALPR (Automatic License Plate Recognition) products are running around the world, which include Safe-T-Cam [1], SeeCar in Israel [2], VECON in Hongkong [3], LPR in USA [4], the ALPR in UK [5], IMPS in Singapore [6], and the

CARINA in Hungary [7]. In these systems, we see various approaches. One approach applies an image pre-processing to obtain clear images (with little noise) of the license plates. Then the license plate images are used as input to commercial OCR software to recognize the characters in them. Other approaches include template matching method and learning-based method. Template matching method must have character images as templates stored in memory. While learning-based methods extract the knowledge of characters from training samples.

A learning-based algorithm on license plate recognition was proposed based on RULES-3 [8] induction theory. One advantage of using this method is that the recognition speed is much quicker in number recognition and it is robust to image rotation and translation. But it is not robust to image scaling. Kim [9] proposed another learning-based method called SVM-based character recognizer for license plate recognition. The recognition rate of Kim's module was about 97.2%.

Zheng [10] compared several types of classifiers and found that SVM approach had the highest accuracy for printed text and handwriting identification in noisy document images. Zhao [11] made the same conclusion after comparing several classifiers for recognizing handwritten numbers. Hence, SVMs have considerable potentials for classification. We have also concluded that inductive learning based method as shown in [8] is simple and fast, and Support Vector machines as shown in [13,14] achieve comparable higher accuracy. Support Vector Machines are a good choice when dealing with noisy data. However, SVM has its limitation, such as huge amount of training samples needed and long training time required. SVMs are also with complex computation burden.

In this paper, a new approach is designed to combine the above-mentioned two classifiers together in order to improve the performance of license plate recognition. A hierarchically combined classification system based on the inductive learning based method and a SVM-based multi-class classifier is constructed to recognize characters of license plate. This approach is discussed in detail in Section 2. A tree-structure for multi-layer classification is demonstrated to simplify and speed up the recognition computation.

The organization of this paper is as follows. We first introduce hierarchically combined classification model in Section 2. Section 3 presents the detailed algorithms for character recognition. The experimental results for license plate recognition are demonstrated in Section 4. Finally, conclusion and further work are presented in Section 5.

2. Combined Classifier Design

Each classifier has its own region where it performs the best. In a practical pattern recognition system, there may be different feature sets, different training sets, and different training sessions. Different classifiers give different results according to different initializations due to the random inherence in the training procedure. For example, SVMs may get different matching rate for setting different values of parameters. It can also be seen that no such single classifier can achieve accuracy dealing for all problems. Instead of selecting a best classifier and discarding the others, one can combine a group of classifiers, whose outputs may be combined, to take advantage of individual classifiers. Through combination of classifiers, we can tolerate the failure of individual classifiers, therefore increase training efficiency and improve the overall classification accuracy.

For classifier combination [15], basically there are three types of structures or models to use. These types are parallel type, cascaded type and hierarchical type. Figure 1 gives their diagram illustrations. They have different features and characteristics.

In the parallel structured model, different kinds of classifiers are grouped and work simultaneously and outputs of these classifiers are combined and considered to obtain the final result. The final result is the one obtained by the classifier who gives the highest accuracy. Parallel model can obtain higher classification accuracy than each classifier but it may have higher computational complexity because each classifier pursues in its own way for classification.

In the cascaded structured model, the classifiers are organized in form of cascade, where the input to the classifier in a succeeding layer is the output of the classifier in the layer above. The output of latter layer is obviously influenced by the former layer's result. In this model, if one classifier fails, all the subsequent classifiers can not work well. The overall classification accuracy cannot be guaranteed.

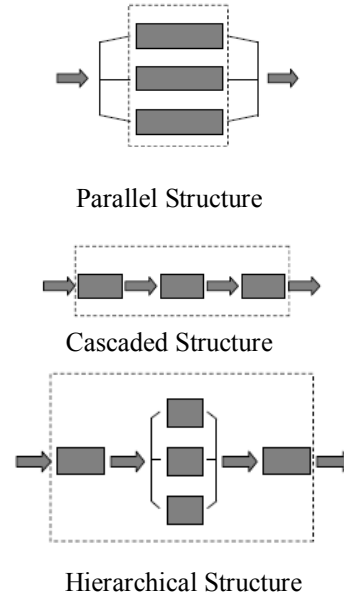


Figure 1. Three Types of Combined Systems

The hierarchical model is a mixture of the previous two types. Different classifiers are used for the same set of samples. Each classifier may have different effects on the final decision. The output of the classifier that gives the best accuracy is selected as the output according to optimization criteria. Therefore the overall performance is hence improved.

No matter which of three models shown in Figure 1 is selected to combine the classifiers, the aim is to provide a more accurate and creditable output. There are many examples showing that a combined classifier can improve the classification accuracy. For example, Chang's combined classifier [16] results in the accuracy of higher than 95% and better overall performance than the individual classifiers. More examples can be found in [17, 18].

In the following, our approach for a hierarchically combined model is presented in detail.

Our combined classifier is to take advantages of two classification methods, namely Inductive Learning Based Method and SVM-based Method for character recognition. The classifiers based on these two methods are layered in a hierarchical structure.

Inductive learning based method is simple and fast. But the classification accuracy of inductive learning based method is not very high to efficiently classify 10 Arabic numerals and 26 letters on license plates. As a consequence, inductive learning based method is suggested to be used as a coarse classifier to divide the 36 different classes (for 10 digits and 26 letters) into several groups before further and finer classification.

On the other hand, SVM takes approach of appropriate kernel design and relevant efficient training algorithms, and hence SVM-based method is a powerful classification method. Due to its original design as a bi-class classifier, SVM has high classification accuracy for bi-class problems. But SVM-based method has lower accuracy in dealing with multi-class problems than bi-class problems. The experimental trend curve between accuracy and number of classes in Figure 2 shows that a higher accuracy can be obtained if the number of classes is less. In other words, the smaller classes to be classified in a multi-class problem, the more exact recognition rate can be obtained using SVM-based method and thereby less time is taken in training and testing processes. We use SVM-based method in our combined classification model for fine classification after the coarse classification using the inductive learning based method.

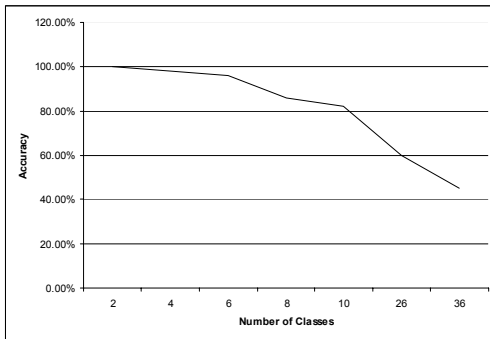


Figure 2. An Experimental Trend Curve Showing the Relationship between Accuracy and Number of Classes.

Our combined classifier consists of two layers. The first layer uses the inductive learning based method, and the second layer applies the SVM-based method. They are combined in a hierarchical structure which is shown in Figure 3.

Based on the extracted rules of an Inductive Learning based classifier, firstly, the characters on a license plate are divided into two groups, namely digitals and letters.

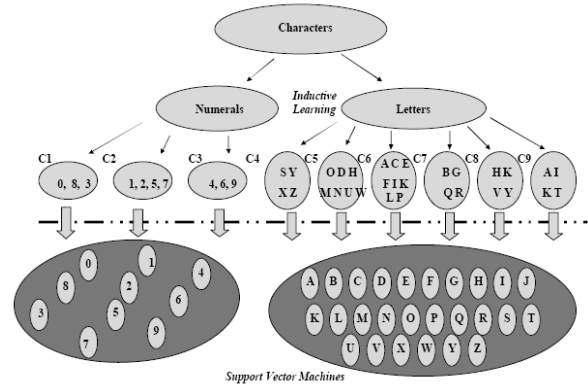


Figure 3. A Hierarchically Combined Structure of Classifiers

Then, each group is further separated into nine smaller collections denoted by C_i ($i = 1, 2, \dots, 9$). In this approach, Collection 1 contains digital 0, 3 and 8, Collection 2 contains 1, 2, 5 and 7, and Collection 3 contains 4, 6 and 9. Similarly, six collections for letters are formed and named as Collection 4 for S, X, Y and Z, Collection 5 for O, D, H, M, N, U and W, Collection 6 for A, C, E, F, I, K, L and P, Collection 7 for B, G, Q and R, Collection 8 for H, K, V and Y, and Collection 9 for A, I, K and T.

After the small collections are formed, each SVM-based classifier at layer two recognize the characters one after another within the collection that the classifier is responsible for.

Finally, all characters in all collections are recognized and the classification results are brought together. The recognition of a license plate is hence accomplished.

3. Classification Algorithms

This section discusses in detail how the hierarchically combined classifier works.

3.1 Coarse classification layer

The classifiers which employ the inductive learning based method is used for coarse classification of characters in license plates. In the state of New South Wales, Australia, there are some basic types of license plates. Six characters contained in a license plate are commonly grouped into the sequence of two letters, two digitals and two letters; three letters and three digitals; or three letters, two digitals and one letter. Once the characters are segmented from license plates, two groups of characters are constructed followed the position in different types. One is for digits and another is for letters. Therefore, two classifiers, R1 and R2, are

created, each applying the inductive learning based method, for numerals and letters respectively at layer 1.

The algorithm [21] for extracting rules is summarized as follows.

Step 1. Take the edge map of a sample character.

Step 2. Set iteration count N_c to 0.

Step 3. If $N_c < N_f$ then $N_c = N_c + 1$; ELSE go to Step 5.

Step 4. Find the appearance frequency of N_c -th feature mask [21] in the sample. Go to Step 3.

Step 5. Form the rules (the frequency ranges of edge masks) for this sample character. If there are no more unclassified samples, then STOP; ELSE go to Step 1.

Suppose we have N_f different feature masks M_i ($i = 1, 2, \dots, N_f$), let us denote the frequency of M_i appeared in the candidate character by C_i and the frequency range of M_i for k -th sample character by $[L_{ik}, H_{ik}]$, where L_{ik} and H_{ik} are two non-negative integers such that $L_{ik} \leq H_{ik}$. Each particular character will have its specific L_{ik} and H_{ik} values for each feature masks. For example, the first feature mask's frequency range block of character A is [30 50]. It means that the frequency values of character A's samples is bigger than 30 and smaller than 50 for the first feature mask. Using these particular range blocks we can divide characters into different groups in first stage. Through training process, rule set is generated from the samples in the training set.

After training process presented in [21], two rule sets for classifiers R1 and R2 respectively are learned. Using the rules created for R1, the ten digits (zero to nine) are categorized into three collections. Similarly, R2 categorize the 26 letters into six collections. These nine collections will be used for finer classification in the next layer.

3.2 Fine classification layer

In the second layer, nine SVMs are trained using the corresponding digits or letters' samples for the nine collections respectively. Once training process is complete, these nine SVMs are ready for digital number and letter classification. Because the number of characters in each collection is much smaller than the number of all characters (which is 36), SVM-based classification for recognition of characters in the collections result in more accurate results than simply applying a SVM for recognition of characters without going through the coarse layer.

One may find that there are overlaps between the letter's collections and no overlaps between digit's collections. Due to this reason, multiple outcomes may be obtained for certain letters such as 'A'. Therefore,

additional step is proposed as a max-win strategy to decide the outcomes obtained for such a letter and select the optimal output as the classification for the letter.

3.3 Classification Tree

Commonly, an SVM-based method applies two approaches for multi-class classification. They are "One against all" and "One against one" methods [19, 20]. "One against all" uses a set of binary classifiers, each trained to separate one class from the rest. There are n classifiers for n decision functions corresponding to n classes. "One against one" uses one classifier for each pair of classes. This method requires $n(n-1)/2$ classifiers or SVMs to be applied for all pairs of classes.

In the following, we present an efficient method for multi-class classifier design using a tree structure, namely classification tree. The classification tree is constructed into the layers of binary classifiers. Each layer has a binary classifier to separate candidate into two groups. Figure 4 the tree classification approach. This classification tree is proposed to speed up the recognition time and also the training time when using SVM-based classifiers.

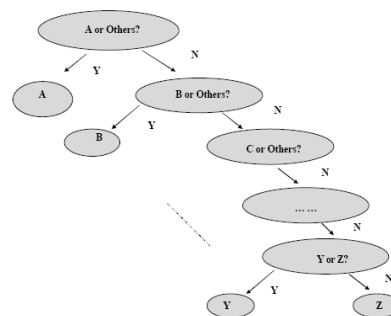


Figure 4 Classification Tree Structure Diagram

Let us take the letter recognition of a car license plate as an example. During the training process, 26 classifiers (SVMs) are built up using one character as a positive sample and all others as negative samples. The SVMs in the higher level nodes have higher priorities than those in the lower level nodes. For instance, SVM which distinguish letter A from other letters has higher priority than SVM which identifies letter B. From the top to the bottom, the training data set decreases in size. In other words, the last classifier which separates letter Y from Z needs only the training samples of letter Y as positive samples and letter Z's as negative samples.

This approach is similar to "One against All (OVA)" method. But the difference between this

approach from OVA is that this approach uses a hierarchical approach as shown in Figure 4, The classifier used to recognize letter ‘A’ has twenty-five letters other than letter ‘A’ as its negative labeled samples. If a classifier is designed for recognizing letter ‘Y’, on the other hand, this classifier has only samples of letter ‘Y’ as its positive labeled samples and letter ‘Z’ as its negative labeled samples. It is clear to see that the higher level classifier takes longer training time due to more training samples while lower level classifier takes less time. The computational burden is greatly released for lower level classifiers.

In addition, classification tree saves more time for candidate recognition. Once each SVM has been trained, these different SVMs are put in different layers of the classification tree. For a given candidate character input, it takes on average only nineteen-folds of test time for every classifier used in ‘‘One against all’’ approach to find its recognition result matching the classification accuracy of ‘‘One against all’’. For instance, if one needs to tell if the candidate is letter ‘A’, the classification result is obtained only by classifier in the first layer. On the other hand, if one wants to classify a letter ‘Z’, it must go through 25 classifiers in 25 layers of the classification tree. On the contrary, original ‘‘One against all’’ method recognizes a candidate through finding the maximum values of 26 classifiers for letter recognition. It needs twenty-six-folds of test (recognition) time. ‘‘One against one’’ requests even much more time in training process and testing process. Therefore, the classification tree provides a much faster way than ‘‘One against all’’ and ‘‘One against one’’ methods.

Table 1. Comparison of the Classification Tree (CT) with OVO and OVA.

	OVO	OVA	CT
Matching Rate	63%	80.7%	82.3%
Training Time (/T _{train})	325	26	19
Testing Time (/T _{test})	325	26	19

The comparison results are shown in Table 1. The matching rates of using the ‘One against One’ (OVO), ‘One against All’ (OVA) and the Classification Tree are 63%, 80.7% and 82.3% respectively. In the table, T_{train} stands for the training time of a classifier. T_{test} stands for the testing time of a classifier. The classification tree improves the recognition accuracy.

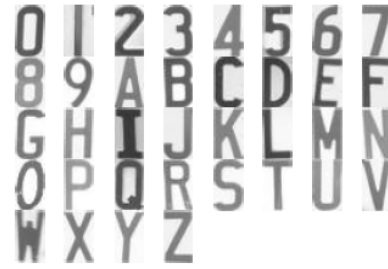


Figure 5. The segmented characters

4. Experiment Setup and Results

The experiments in this section are set up to show that the hierarchically combined classifier outperforms the individual classifiers. The experiments are based on the database in which the license plate images are segmented from the car license plates in NSW in real-time as shown in Figure 5. The benchmark of every experiment is based on error rate defined as

$$\text{Error rate} = \frac{\text{The number of characters recognized incorrectly}}{\text{The number of testing characters}}$$

The lower error rate, the better performance a classifier has. On the other hand, the lower error rate, the better classifier is.

Table 2 Average Error Rate of Collection 1 to Collection 9

Cases	Average Error Rate	Testing Time (s)
Collection 1	10.8%	0.02
Collection 2	14.5%	0.04
Collection 3	17.0%	0.02
Collection 4	12.5%	0.04
Collection 5	1.0 %	0.1
Collection 6	12.5%	0.1
Collection 7	33%	0.03
Collection 8	33%	0.04
Collection 9	16.7%	0.03
Average	17.7%	0.06

Considering the three collections for digits, the SVM-based multi-class classifiers give the error rates of 10.7%, 14.5% and 17.0% respectively as shown in Table 2.

Considering the six collections of letters, the average error rates are 12.5%, 1.0 %, 29.1 %, 33%,

33% and 16.7 % respectively. The overall error rate is 17.7% on average.

In our data sample sets, there are some deformations contained in these images. In addition, there are less samples images for those characters. From the results showing the error rates of others, we believe that more training samples added, the less error rate is achieved.

5. Conclusions

The inductive learning method is simple and quick. It does class recognition through rule induction. It needs not store the training samples in memory. A disadvantage of inductive learning based method is that the feature masks are dependent on the edge information. So, this method shows pitfalls in pursuing higher classification accuracy especially for dealing with multi-class classification. SVM has the high classification accuracy but it takes more samples and more time for training. Furthermore, the classification accuracy is reduced when SVM originally designed for bi-class classification is used for multi-class classification.

The proposed classifier combining the inductive learning with SVM-based classifiers shows the improved classification accuracy compared with the inductive learning based and SVM-based methods. Furthermore, the classification tree saves the training and testing time for a lot when using proposed combined classifier. The experimental results support the above claim.

6. References

- [1] <http://vision.cmit.csiro.au/project/stc/>.
- [2] <http://www.htsol.com/AboutUs.html>.
- [3] <http://www.asiavision.com.hk/>.
- [4] <http://www.perceptics.com/license-plate-reader.html>.
- [5] <http://www.ivsuk.com/anpr.asp>.
- [6] <http://www.singaporegateway.com/optasia/imps>.
- [7] <http://www.arhungary.hu/>.
- [8] M. S. Aksoy, G. Cagil, A. K. Turker, "Number-plate recognition using inductive learning", *Robotics and Autonomous Systems*, Elsevier, Vol.33, pp.149-153, 2000.
- [9] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-based approach for license plate recognition," *Neural Networks for Signal Processing X*, 2000. *Proceedings of the 2000 IEEE Signal Processing Society Workshop*, vol. 2, pp. 614 - 623, 2000.
- [10] Y. Zheng, H. Li, and D. Doermann, "Machine printed text and handwriting identification in noisy document images," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, vol. 26, pp. 337 - 353, Mar. 2004.
- [11] B. Zhao, Y. Liu, and S.-W. Xia, "Support vector machine and its application in handwritten numeral recognition," *Proceedings of 15th International Conference on Pattern Recognition*, vol. 2, pp. 720 - 723, Sept 2000.
- [12] L. Zheng and X. He, "Number Plate Recognition Based on Support Vector Machines", *Proceeding of IEEE AVSS 2006 conference*. ISBN-13: 978-0-7695-2688-1, 2006.
- [13] Cristianini, N. and Shawe-Taylor, J., "An introduction to support vector machines and other kernel-based learning methods", Cambridge University Press, 2000.
- [14] Vapnik, V. N., "The Nature of statistical learning theory", New York: Springer, 1995.
- [15] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. 20, pp. 226-239, 1998.
- [16] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," in *IEEE Transactions on Intelligent Transportation Systems*, 2004, pp. 42-53.
- [17] L.-S. Wang, Y.-T. Xu and L.-S. Zhao, "A kind of hybrid classification algorithm based on rough set and support vector machine," in *2005 International Conference on Machine Learning and Cybernetics*, pp. 1676-1679, 2005.
- [18] Y. Yang and M. Nakagawa, "Improving the structuring search space method for accelerating large set character recognition," in *Ninth International Workshop on Frontiers in Handwriting Recognition, IWFHR-9 2004*, pp. 251-256, 2004.
- [19] Foody, G.M.; Mathur, A., "A relative evaluation of multiclass image classification by support vector machines", *IEEE Transactions on Geoscience and Remote Sensing*, Volume 42(6), pp.1335-1343, 2004.
- [20] Jian-xiong Dong, Ching Y. Suen and Adam Krzyzak, "Algorithms of fast SVM evaluation based on subspace projection", *2005 IEEE International Joint Conference on Neural Networks*, Vol. 2(31), pp.865-870, 2005.
- [21] L. Zheng X. He, Q. Wu and T. Hintz, "Learning-based Number Recognition on Spiral Architecture", *Proceeding of IEEE ICARCV2006*. Singapore, pp.897-901, 2006.