A system for licence plate recognition using a hierarchically combined classifier

Lihong Zheng*

School of Computing and Mathematics,
Charles Sturt University,
Locked Bag 588,
Wagga Wagga, New South Wales 2678, Australia
E-mail: lzheng@csu.edu.au
*Corresponding author

Xiangjian He, Qiang Wu and Bijan Samali

Faculty of Engineering and Information Technology,
University of Technology Sydney,
P.O. Box 123,
Broadway, NSW 2007, Australia
E-mail: Xiangjian.He@uts.edu.au
E-mail: Qiang.Wu@uts.edu.au
E-mail: Bijan.Samali@uts.edu.au

Abstract: In a real time, automatic licence plate recognition system, licence detection, character segmentation and character recognition are three important components. All these three components generally require high accuracy and fast recognition speed to process. In this paper, general processing steps for license plate recognition (LPR) are addressed. After three types of combined classifiers are introduced and compared, a hierarchically combined classifier is designed based on an inductive learning-based method and an support vector machine (SVM)-based classification. This approach employs the inductive learning-based method to roughly divide all classes into smaller groups. Then, the SVM approach is used for character classification in individual groups. Having obtained a collection of samples of characters in advance from licence plates after licence detection and character segmentation steps, some known samples are available for training. After the training process, 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 next fast training and testing processes based on SVMs. The experimental results show that the hierarchically combined classifier is better than either the inductive learning-based classification or the SVM-based classification with a lower error rate and a faster processing speed.

Keywords: licence plate recognition; class tree; hierarchically combined classifier.

Reference to this paper should be made as follows: Zheng, L., He, X., Wu, Q. and Samali, B. (2011) 'A system for licence plate recognition using a hierarchically combined classifier', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 10, No. 2, pp.189–202.

Biographical notes: Lihong Zheng received her PhD in Computing Sciences in 2008. Currently, she is a Lecturer in the School of Computing and Mathematics at Charles Sturt University. Her previous research interest was on automation, robotics and artificial intelligence. Her current research interests are computer vision, image processing, pattern recognition and machine learning. She has had one patent and more than 20 publications in these years.

Xiangjian He is a Professor in the University of Technology, Sydney (UTS). He got his PhD in Computing Sciences in 1999. He is a Senior Member of IEEE. Also he is a Deputy Director of Research Centre for Innovation in IT Services and Applications (iNEXT) at UTS. His research interests are computer vision, image processing, motion tracking and computer and network security. In recent years, he has had over 200 refereed publications in books, journals and conferences.

Qiang Wu received his PhD degree in 2004 from the University of Technology, Sydney (UTS) and MEng and BEng in Communication and Electronic System from Harbin Institute of Technology, China in 1998 and 1996, respectively. Currently, he is a Senior Lecturer in the School of Computing and Communications of UTS and also an Active Researcher in Computer Vision Research Group of iNEXT Research Centre in UTS. He has been working in the field of computer vision and image processing for several years. His research interests include image processing, object detection, biometrics and pattern recognition.

Bijan Samali is currently the Head of the School of Civil and Environmental Engineering at the University of Technology, Sydney and has a Personal Chair in Structural Engineering at UTS. He is also the Director of Centre for Built Infrastructure Research at UTS. He has published over 300 technical papers in engineering journals and conference proceedings. His main research interests lie in the general area of structural dynamics including wind and earthquake engineering with special emphasis on structural control, dynamic measurement and analysis of buildings and bridges, damage detection and structural health monitoring, and use of smart materials in engineering applications.

This paper is a revised and expanded version of a paper entitled [A hierarchically combined classifier for licence plate recognition] presented at [8th IEEE International Conference on Computer and Information Technology, 2008 (CIT 2008), Sydney, July 2008].

1 Introduction

A licence plate is a unique identification of a vehicle. The fundamental issues in real-time licence plate recognition are the accuracy and the recognition speed. Till now, lots of commercial automatic licence plate recognition (ALPR) products are available around the world. The existing ALPR systems include Safe-T-Cam (Csiro), SeeCar in Israel (Htsol), VECON in Hongkong (Asiavision), LPR in USA (Lpr), the ALPR in UK (Ivsuk), IMPS in Singapore (Spg) and the CARINA in Hungary (Arh). In these systems, we see various approaches. One approach applies an image pre-processing to obtain clear images (with little noise) of the licence plates. Then, the licence plate images are used as input to commercial optical character recognition (OCR) software to recognise the characters in them. Other approaches include template matching methods and learning-based methods.

Template matching methods must have character images as templates stored in the memory. Then, the input image is compared with the templates for recognition. Learning-based methods extract the knowledge (or features) of characters from training samples. Then, the input image is recognised based on the feature matching.

Zheng et al. (2004) compared several types of classifiers and found that support vector machine (SVM) approach had the highest accuracy for printed text and handwriting text identification in noisy document images. Zhao et al. (2000) made the same conclusion after comparing several classifiers for recognising handwritten numbers. Hence, SVMs have considerable potentials for classification. We have also concluded that inductive learning-based methods as shown in [Aksoy00] are simple and fast, and SVM-based methods as shown in Cristianini and Shawe-Taylor (2000) and Vapnik (1995) achieve comparable higher accuracy. SVM-based methods 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. Furthermore, SVM was originally designed for bi-class classification. SVM is also with complex computation burden. Hence, in this paper, a new approach is designed to combine the inductive learning-based approach together with SVM-based approach in order to improve the performance of licence plate recognition. A hierarchically combined classifier is constructed to recognise characters of licence plates. This approach is discussed in detail in Section 3. A tree structure for multi-layer classification is demonstrated to simplify and speed up the recognition computation.

The organisation of this paper is as follows. First, general licence plate recognition process is described in Section 2. We then introduce a hierarchically combined classification model in Section 3. Section 4 presents the detailed algorithms for character recognition. The experimental results for licence plate recognition are demonstrated in Section 5. Finally, conclusion and further work are presented in Section 6.

2 Licence plate recognition system

Licence plate recognition system usually consists of three major parts: licence plate detection, character segmentation and character recognition.

2.1 Licence plate detection

Licence plate detection is one of important steps in a LPR system. The quality of a licence plate detector influences the accuracy of licence plate recognition. On the other hand, many factors can affect the accuracy and efficiency of licence place detection. Most existing licence plate detection algorithms are restricted by various controlled conditions such as fixed backgrounds, known colour or designated ranges of the distance between cameras and vehicles. In this paper, we use an AdaBoost learning algorithm as shown in Zhang et al. (2007) based on both global statistical features and local Haar-like features to construct a cascaded classifier for licence plate detection.

As shown in Zhang et al. (2007), the basic idea of the detection algorithm was to use a variable scanning window moving around an input vehicle image. At each position, the image area covered by the scanning window was classified using a pre-trained classifier as either a licence plate area (a positive decision) or a non-licence plate area (a negative decision).

The algorithm adapted for licence plate detection is as shown in Zhang et al. (2007), a six-layer cascaded classifier is constructed to increase the detection speed, in which the first two layers are based on global features and the last four layers are based on local Haar-like features. A positive result from the first classifier triggers the evaluation of a second classifier. A positive result from the second classifier triggers a third classifier, etc. A negative outcome at any layer leads to the immediate rejection of the image region (block). In other words, those image regions that are not rejected by the upper layer classifier will be processed by the next layer classifier. If the classifier at any layer rejects a selected image region, no further processing will be performed for this region. The reason of using this method for detection before recognition is that the algorithm can be achieved in real time and accurately, and it can be more flexibly applied for applications under complex environmental conditions. As an illustration, Figure 1 shows nine licence plates detected using the algorithms introduced in this section.

2.2 Character segmentation

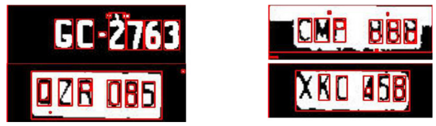
Once the licence plate area is located, the next step is to segment the characters individually. In recent years, there have been many algorithms developed for character segmentation on licence plates. These algorithms include the works shown in Zhang and Zhang (2003) based on Hough transform, and in Zhang et al. (2006) based on horizontal and vertical projections. However, licence plate images captured in real time usually contain heavy noise and are with complex backgrounds. Sometimes, licence plate boundaries are connected to inside characters, characters are connected to each other, or characters are broken (Jia et al., 2007). It brings difficulties for proper character segmentation due to image noise, plate frame, rivet, space mark, plate rotation and illumination variance (Zhang and Zhang, 2003). Improperly segmented characters will result in misrecognised characters. Hence, it is still a problem for accurate and real-time character segmentation.

In our system, blob extraction (Horn, 1986) algorithm is used for quick and accurate character segmentation.

Blob extraction (Horn, 1986) is an image segmentation technique that categorises the pixels in an image as belonging to one of many discrete regions. Blob extraction performs on binary images through globally thresholding image pixel values. It consists of processes of region labelling, connected-component labelling and blob discovery. It first scans an image and groups its pixels into components based on pixel connectivity. All pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all components have been determined, each pixel is labelled according to the component it was assigned to. Figure 2 shows detected blobs in red boxes.

Figure 1 Detected licence plates

Figure 2 Detected blobs of licence plate samples (see online version for colours)



The detailed process for blob extraction is described below:

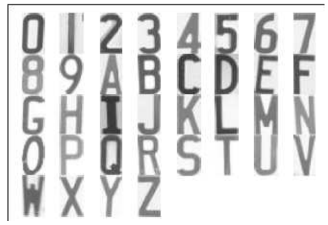
1. Scan the binary image from left to right and from top to bottom.
2. For every pixel, check the neighbouring pixels in northeast, north, northwest and west directions in the 8-connectivity neighbour structure for its region criterion such that the pixel intensity value is 1 on the binary image.
   a. If none of the neighbours meet the criterion then assign the pixel to the region with the current value of region counter. Then, increment the region counter.
   b. If only one neighbour meets the criterion assign the pixel to the region of this neighbour.
   c. If multiple neighbours meet the criterion and they are all members of the same region, assign the pixel to the region of these neighbours.
   d. If multiple neighbours meet the criterion but they are members of various regions, assign the pixel to one of the regions that these neighbours belong and record that all of these regions must be combined into one single region and assigned a new, single region value.
3. The pixels with the same region value is grouped together to form a component and is assigned a unique label.

Based on the labelled component in an image, we can extract the regions or blobs of which each has a unique label. The top, bottom, right and left positions, and the width, length and area of each blob can be determined or computed. Then, we select the blobs as the candidate characters based on the height-to-width ratio values and the estimated area of a character. The estimated average size of character and space between characters can also be used as selection criteria. Obviously, if the licence plate boundary is connected to some characters inside the licence plate or some characters are connected to each other, these characters may not be recorded as candidate characters because of the sizes of the corresponding components or regions. The information such as the character amount on a licence plate, the space between segmented characters, and the averaged width and height of a licence plate are also be considered to improve the segmentation accuracy. Figure 3 shows some segmented characters from captured licence plates.

2.3 Character recognition

There have been various commercial systems for licence plate recognition around the world. In these systems, either OCR-based methods or learning-based methods are applied.

Figure 3 The segmented characters on licence plates



OCR has shown its advantage in recognising printed document or text where the background has no or very little noise. However, licence plate images captured in real time usually contain heavy noise and are with complex backgrounds. Therefore, OCR-based approaches put more efforts on removing the noise and try to improve the quality of characters segmented from licence plate images.

A learning-based algorithm on licence plate recognition was proposed in Aksoy et al. (2000) based on RULES-3 induction theory. One advantage of using this method is that the recognition speed is much quicker and it is robust to image rotation and translation. But it is not robust to image scaling. Kim et al. (2000) proposed another learning-based method using a SVM-based character recogniser for licence plate recognition. The recognition rate of Kim's module was about 97.2%. Learning-based approach such as those shown in Zheng et al. (2006a,b) and Jin and Qin (2003) extracts optimal features of characters to improve the recognition accuracy.

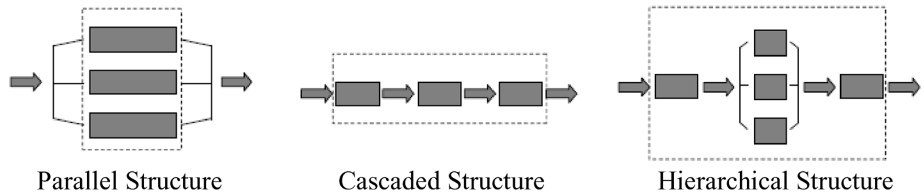
3 Combined classifier design

Each classifier has its own application areas 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 initialisations due to the random inherence in the training procedure. For example, SVMs may lead to different classification accuracies or matching rates when different values of parameters are set. It can also be seen that no single classifier can be of high accuracy for dealing all problems. Instead of selecting a best classifier and discarding the others, we combine a group of classifiers together with their outputs to take advantages of individual classifiers. Through combination of classifiers, we can tolerate the failure of individual classifiers, and hence increase training efficiency and improve the overall classification accuracy.

For classifier combination (Kittler et al., 1998), basically there are three types of structures or models to use. These types are parallel type, cascaded type and hierarchical type. Figure 4 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.

Figure 4 Three types of combined systems



In the cascaded structured model, the classifiers are organised 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 results. In this model, if one classifier fails, all the subsequent classifiers cannot work well. The overall classification accuracy cannot be guaranteed.

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 its own effects on the final decision. The output of the classifier that gives the best accuracy is selected as the output according to optimisation criteria. Therefore, the overall performance is improved.

No matter which of three models shown in Figure 4 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 (Chang et al., 2004) results in the accuracy of higher than 95% and better overall performance than the individual classifiers. More examples can be found in Wang et al. (2005) and Yang and Nakagawa (2004).

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 high enough to efficiently classify 10 Arabic numbers and 26 letters on licence plates. As a consequence, inductive learning-based method is suggested to be used only 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 5 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 higher 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.

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 6.

Based on the extracted rules of an inductive learning-based classifier, firstly, the characters on a licence plate are divided into two groups, namely digitals and letters.

Then, each group is further separated into nine smaller collections denoted by C_i ($i = 1, 2, \ldots, 9$). 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*.

Figure 5 An experimental trend curve showing the relationship between accuracy and number of classes

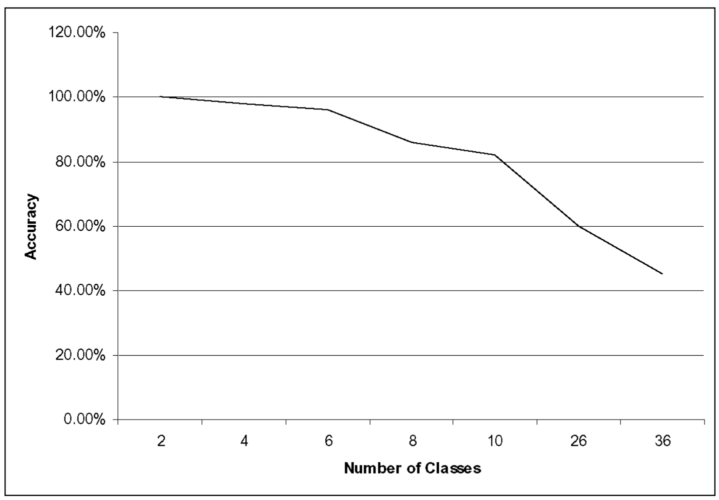
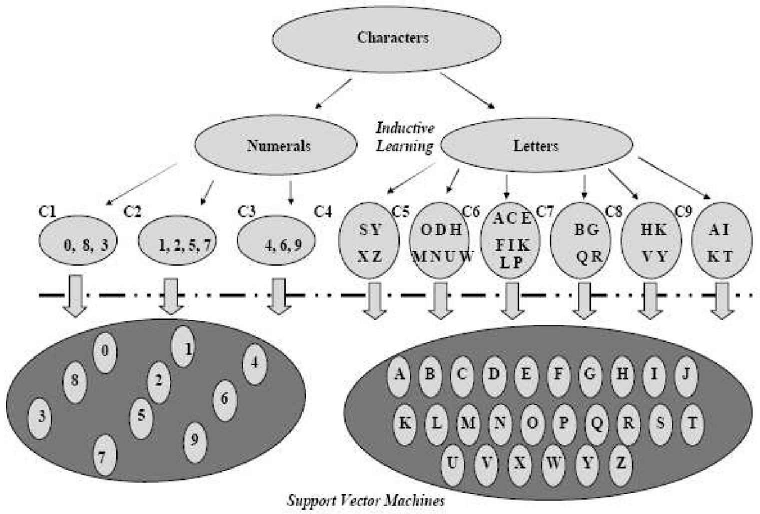


Figure 6 A hierarchically combined structure of classifiers



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

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

4 Classification algorithms

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

4.1 Coarse classification layer

The classifiers which employ the inductive learning-based method is used for coarse classification of characters in licence plates. In the state of New South Wale, Australia, there are some basic types of licence plates. Six characters contained in a licence 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 licence 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 (Zheng et al., 2006b) for extracting rules is summarised 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_cth feature mask (Aksoy et al., 2000) 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 ($i = 1, 2, \ldots, N_f$), 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 are 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 Zheng et al. (2006b), two rule sets for classifiers *R*1 and *R*2, respectively, are learned. Using the rules created for *R*1, the ten digits (zero to nine) are categorised into three collections. Similarly, *R*2 categorise the 26 letters into six collections. These nine collections will be used for finer classification in the next layer.

4.2 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.

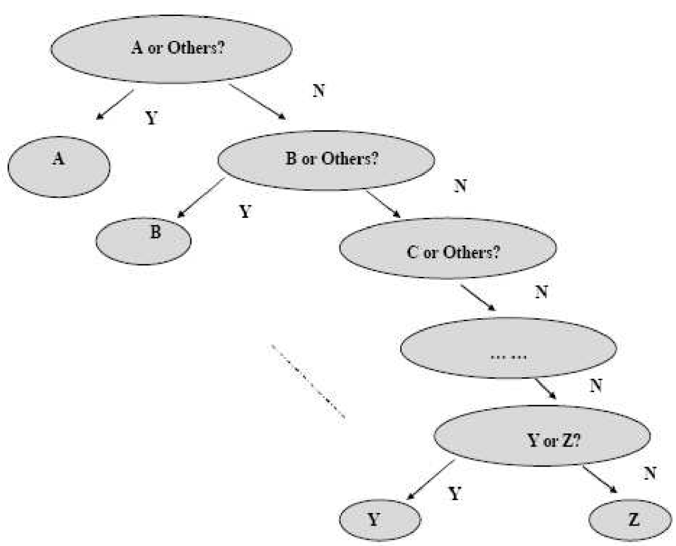
One may find that there are overlaps between the letter's collections and no overlaps between digit's collections. Due to this reason, multiple outcomes may be obtained for certain letters such as '*A*'. Therefore, additional step is proposed as a max-win strategy to decide the outcomes obtained for such letter and select the optimal output as the classification for the letter.

4.3 Classification tree

Commonly, an SVM-based method applies two approaches for multi-class classification. They are 'one against all' (OVA) and 'one against one' (OVO) methods (Dong et al., 2005; Foody et al., 2004). 'OVA' 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. 'OVO' 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 (CT). The CT is constructed into the layers of binary classifiers. Each layer has a binary classifier to separate candidate into two groups. Figure 7 shows the tree classification approach. This CT is proposed to speed up the recognition time and also the training time when using SVM-based classifiers.

Figure 7 CT structure diagram



Let us take the letter recognition of a car licence 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 those of letter *Z* as negative samples.

This approach is similar to 'OVA' method. But the difference of this approach from OVA is that this approach uses a hierarchical approach as shown in Figure 4. The classifier used to recognise letter '*A*' has 25 letters other than letter '*A*' as it's negative labelled samples. If a classifier is designed for recognising letter '*Y*', on the other hand, this classifier has only samples of letter '*Y*' as its positive labelled samples and the samples of letter '*Z*' as it's negative labelled 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, CT saves more time for candidate recognition. Once each SVM has been trained, these different SVMs are put in different layers of the CT. For a given candidate character input, it takes on average only 19-folds of test time for every classifier used in 'OVA' approach to find its recognition result matching the classification accuracy of 'OVA'. 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 CT. On the contrary, original 'OVA' method recognises a candidate through finding the maximum values of 26 classifiers for letter recognition. It needs 26-folds of test (recognition) time. 'OVO' requests even much time in training process and testing process. Therefore, the CT provides a much faster way than 'OVA' and 'OVO' methods.

The comparison results are shown in Table 1. The matching rates of using the 'OVO', 'OVA' and the CT are 63%, 80.7% and 82.3%, respectively. In the table, *Ttrain* stands for the training time of a classifier. *Ttest* stands for the testing time of a classifier. The CT improves the recognition accuracy.

Table 1 Comparison of the CT with OVO and OVA

	OVO	*OVA*	*CT*
Matching rate	63%	80.7%	82.3%
Ratio of Training time to Ttrain	325	26	19
Ratio of Testing time to Ttest	325	26	19

5 Experiment set-up 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 licence plate images are segmented from the car licence plates in

New South Wales State of Australia in real time as shown in Figure 3. The benchmark of every experiment is based on error rate defined as

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

The lower the error rate is, the better the classifier performs.

5.1 Multi-class classification on the collections

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 sample 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.

Table 2 Average error rate of collections 1–9

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	0.10
Collection 6	12.5	0.10
Collection 7	33.0	0.03
Collection 8	33.0	0.04
Collection 9	16.7	0.03
Average	17.7	0.06

6 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 CT saves the training and testing time a lot when using the combined classifier proposed. The experimental results support the above claim.

References

Aksoy, M.S., Cagil, G. and Turker, A.K. (2000) 'Number-plate recognition using inductive learning', *Robotics and Autonomous Systems*, Vol. 33, pp.149–153.

Chang, S-L., Chen, L-S., Chung, Y-C. and Chen, S-W. (2004) 'Automatic license plate recognition', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, pp.42–53

Cristianini, N. and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge: Cambridge University Press.

Dong, J-X., 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, No. 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, No. 6, pp.1335–1343.

Horn, B.K.P. (1986) *Robot Vision*. Cambridge: The MIT Press, pp.69–71, ISBN 0-262-08159-8.

Jia, X. et al. (2007) 'A novel algorithm for character segmentation of degraded license plate based on prior knowledge', *Proceedings of the IEEE International Conference on Automation and Logistics*, pp.249–253.

Jin, L-W. and Qin, J-Z. (2003) 'Car plate number characters recognition using Gabor orientation features and neural networks', *IEEE International Conference Neural Networks and Signal Processing*, pp.1628–1631.

Kim, K.K., Kim, K.I., Kim, J.B. and Kim, H.J. (2000) '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.

Kittler, J., Hatef, M., Duin, R.P.W. and Matas, J. (1998) 'On combining classifiers', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, pp.226–239.

Vapnik, V.N. (1995) *The Nature of Statistical Learning Theory*. New York: Springer.

Wang, L-S., Xu, Y-T. and Zhao, L-S. (2005) 'A kind of hybrid classification algorithm based on rough set and support vector machine', *2005 International Conference on Machine Learning and Cybernetics*, pp.1676–1679.

Yang, Y. and Nakagawa, M. (2004) 'Improving the structuring search space method for accelerating large set character recognition', *Ninth International Workshop on Frontiers in Handwriting Recognition, IWFHR-9 2004*, pp.251–256.

Zhang, J., Fan, X. and Huang, C. (2006) 'Research on characters segmentation and characters recognition in intelligent license plate recognition system', *Proceedings of the 25th Chinese Control Conference*, pp.1753–1755.

Zhang, H., Jia, W., He, X. and Wu, Q. (2007) 'Learning-based license plate detection in vehicle image database', *Int. J. Intelligent Information and Database Systems (IJIIDS)*, Vol. 1, No. 2, pp.228–243, ISSN: 1751-5858.

Zhang, Y. and Zhang, C. (2003) A new algorithm for character segmentation of license plate, *IEEE Intelligent Vehicles Symposium*, pp.106–109.

Zhao, B., Liu, Y. and Xia, S-W. (2000) 'Support vector machine and its application in handwritten numeral recognition', *Proceedings of 15th International Conference on Pattern Recognition*, Vol. 2, pp.720–723.

Zheng, L. and He, X. (2006a) '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. (2006b) 'Learning-based number recognition on spiral architecture', *Proceeding of IEEE ICARCV2006*, Singapore, pp.897–901.

Zheng, Y., Li, H. and Doermann, D. (2004) 'Machine printed text and handwriting identification in noisy document images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, pp.337–353.

Websites

Csiro_http://vision.cmit.csiro.au/project/stc/.
Htsol_http://www.htsol.com/AboutUs.html.
Asiavision_http://www.asiavision.com.hk/.
Lpr_http://www.perceptics.com/license-plate-reader.html.
Ivsuk5_http://www.ivsuk.com/anpr.asp.
Spg_http://www.singaporegateway.com/optasia/imps.
Arh_http://www.arhungary.hu/.