2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

Vehicle License Plate Recognition Based on Class-specific ERs and SaE-ELM

Chao Gou, Kunfeng Wang, Bo Li, and Fei-yue Wang

Abstract— **In this paper, an effective approach to vehicle license plate recognition based on Extremal Regions (ERs) and Self-adaptive Evolutionary Extreme Learning Machine (SaE-ELM) is proposed. In the license plate detection step, some computations including morphological operations, various filters, different contours and validations are sequentially performed to extract some image regions as candidate license plates. Then, accurate character segmentation is achieved through a proper selection of ERs. In the character recognition step, the HOG (histogram of oriented gradients) feature vector in each character region is extracted, and then the characters are recognized using an offline trained pattern classifier of SaE-ELM. Experimental results show that our approach works quite well in complex traffic environments.**

I. Introduction

In recent years, intelligent transportation systems (ITS) have been widely used to tackle the growing urban problems, especially traffic congestion and accidents. ITS can roughly be categorized into intelligent infrastructure systems and intelligent vehicle systems. With the rapid development of computer vision technology, more and more vision-based systems are applied in ITS. For instance, computer vision systems for vehicle license plate recognition (VLPR) are used as a core of intelligent infrastructure systems like electronic payment systems and freeway management systems [1].

In general, VLPR from traffic images or videos consists of three processing steps: license plate region detection, character segmentation, and character recognition.

In order to recognize vehicle license plates, the license plate regions should first be extracted from a still traffic image. Accurate detection of license plate regions is essential to carry on other steps of VLPR. There are two major methods for locating vehicle license plates: one method is based on color information [2], [3], while another one is based on textures or edges of the license plates [4], [5]. In the methods of the first type, many detect the license plates based on the H component of the HIS color model. In addition, the color combination of a license plate and its characters is specific, and this combination occurs almost only in the license plate region [3]. However, the methods that use color combination to localize license plates may become invalid when there are regions in the image whose color information is similar to that of the license plate. Moreover, license plate detection based on color information is sensitive to adverse illumination conditions and camera settings. On the other hand, texture-based methods use high edge-density areas where color transition occurs dramatically. These methods can detect license plate regions in relatively simple environments, but can easily be affected by noises and are computationally complex when there are many edges in the image.

In the second step, the license plate is segmented to extract the isolated characters. In [6], the extracted license plate is rescaled to a template size, while in the template all the character positions are known. This method is incapable of dealing with any shift in the extracted license plates. Considering that the characters and license plate backgrounds have different colors, some methods [3], [7] project the binary extracted license plate vertically to determine the starting and the ending positions of characters, and then project the extracted characters horizontally to extract each character alone. The projection method is common and simple, but is dependent of their accurate positions. It is obvious that this method needs prior knowledge on the character number and is sensitive to noises.

Finally, it is difficult to recognize license plate characters because different camera zooms lead to different character sizes. Moreover, the extracted characters are very small and may be similar in their shapes, such as the pairs of S-5, C-G, and D-0. A large number of character recognition methods have been proposed, including neural networks [6], [7], a support vector machine (SVM) in [8], character templates in [9], [10] and so on. The conventional neural networks and SVM methods have a high correct rate when used for character recognition, but they need a long time to train the model and the recognition procedure is also time-consuming. The character templates method is unable to deal with ambiguity of the characters.

In this paper, an effective approach to vehicle license plate recognition is proposed, based on Extremal Regions (ERs) and Self-adaptive Evolutionary Extreme Learning Machine (SaE-ELM) [11].The flowchart of the method is illustrated in Fig.1. Firstly, top-hat transformation is adopted to preprocess the input image which helps restrain background noises, followed by Sobel filter to find the vertical edges and morphological operations is used to remove blank spaces between each vertical edge line. Then the coarse LP detection

This work is supported partly by the National Natural Science Foundation of China under Grant 61304200, MIIT Project of the Internet of Things Development Fund: R&D of Real-time Image Recognition Techniques and Application Systems for Social Security, and the Early Career Development Award of SKLMCCS under Grant 2F13Q02.

The authors (Kunfeng Wang is the corresponding author) are with Qingdao Academy of Intelligent Industries, Qingdao 266109, China, and also with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (phone: 86-10-82544791; e-mail: gouchao.cas@gmail.com, kunfeng.wang@ia.ac.cn).

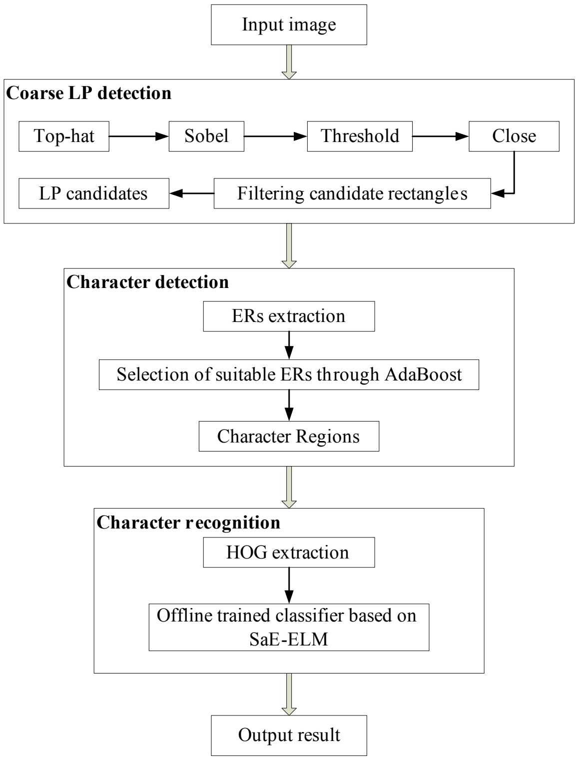


Fig. 1. The flowchart of the proposed approach

is achieved by filtering different sizes of rectangular regions though geometrical validations. Secondly, suitable ERs [12] are selected as character regions though a Real AdaBoost classifier with decision trees. Accurate character segmentation and LP location are achieved by geometrical attributes of characters in standard license plates. Thirdly, features of the character regions, namely histogram of oriented gradients (HOG) [13] descriptors, are extracted from the input image and the characters are recognized using the offline trained classifier based on SaE-ELM.

The remainder of this paper is arranged as follows. Section II explains coarse license plate detection, ERs extraction, and character segmentation. License plate recognition based on HOG features and SaE-ELM is presented in Section III. Experimental results for natural traffic images are discussed in Section IV. The conclusion is drawn in Section V.

II. License Plate Detection and Character Segmentation

A. Coarse Localization of License Plates

License plate localization aims at distinguishing license plates from other regions of the image. The proposed framework for coarse localization of license plates is shown in Fig.1. We first perform top-hat transformation in the input image, which is a mathematical morphological transformation. Specifically, top-hat is a non-linear filter [14], consisting of multiple functions for restraining noises, extracting features, segmenting images, etc. The top-hat transformation equation is shown in (1).

$$Top\_hat(A) = A - (A \circ B) \tag{1}$$

For Aggregate A and structure B, $A \circ B$ is defined as an open arithmetic in mathematical morphology, which helps to remove small objects and to smooth edges of the input image. The open arithmetic equation is shown in (2).

$$A \circ B = (A \Theta B) \oplus B \tag{2}$$

$A \Theta B$ and $A \oplus B$ are both morphological arithmetic, called erosion and dilation, respectively. Erosion is used to eliminate isolated noises and its equation is shown in (3). Dilation is used to combine the object with the background points and its equation is shown in (4).

$$X = A \Theta B = X : B + X \subset A \tag{3}$$

$$X = A \oplus B = X : (-B + X) \cap A \neq \varnothing \tag{4}$$

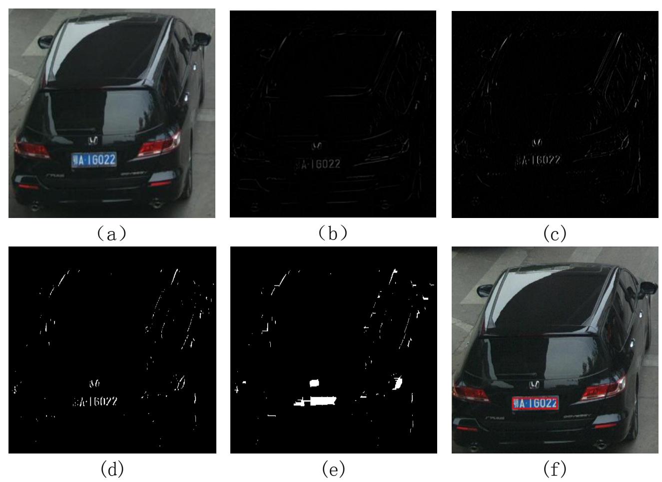


Fig. 2. Some intermediate results of coarse localization. (a) Origin image (local view). (b) Top-hat. (c)Sobel. (d) Binaryzation. (e)Close. (f) Coarse location

As shown in Fig.2 (b), after top-hat transformation, an important feature of the license plate region is the the existence of dense vertical edges. According to Zheng *et al.* [4], the license plate can easily be isolated if the vertical edges are extracted from the car image. In light of that, the Sobel filter is used to get vertical edges as shown in Fig.2 (c). In addition, Gaussian blur in 5×5 grid is applied to remove noises from the camera and other ambient factors. As shown in Fig.2 (d), similar to [15], a threshold filter based on the Otsu method is needed to obtain the binary image. Then the blank spaces between each vertical edge line are removed though a close morphological operation as shown in Fig.2 (e). Once extracting the bounding rectangles of external contours, a filtering method based on the aspect ratio and area is applied to get the regions that may contain license plates. The coarse localization of license plates is achieved, and the result is several positions of candidate license plates (see Fig.2 (f) and Fig.4 (a)).

B. Extraction of Character Regions

1) Extremal regions: A color image can be considered as a mapping $I : D \subset M \times N \rightarrow V$ where V indicates three channels represented by $\{0,...,255\}^3$ A channel C of the image I is a mapping $C : D \rightarrow P$ where P is a totally ordered set and $f_c : V \rightarrow P$ is a projection of pixel values to an ordered set. Region R of an image I is a contiguous subset of D. The outer boundary ∂R of R is a set of pixels adjacent to R(4-connected pixels) but do not belong to R. Extremal Region (ER) is a region whose outer boundary pixels have strictly higher values than the region itself, i.e. $\forall m \in R, n \in \partial R : C(m) \leq t < C(n)$, where t denotes the threshold of the ER. As presented in [12], an ER r at threshold t is a union of one or more ERs R_{t-1} at threshold $t-1$ and pixels of value t formulated in (5).

$$r = (\bigcup u \in R_{t-1}) \cup (\bigcup m \in D : C(m) = t) \tag{5}$$

2) Character detection: Maximally Stable Extremal Region (MSER) [16], whose size remains unchanged over a range of thresholds, is a special case of ER. Some methods [10], [17], [18] have been proposed to use the MSER detector to detect and segment license plate characters. However, these methods are not sufficiently robust against low image contrast and low illumination in complex traffic environments. In light of that, Matas and Zimmermann [19] proposed to drop the stability requirement of MSERs and select class-specific ERs for object detection and categorization. Neumann *et al.* [12] proposed to select suitable ERs for scene text detection though a sequential classifier trained for character detection.

Considering that each channel of the image is processed separately, the number of ERs for a megapixel image can easily reach the level of 1×10^6. Hence, a specific classifier is needed to select suitable ERs for character detection. Referring to Neumann *et al.* [12], we use the incrementally computed descriptors as features for the classifier. The incrementally computable descriptors of ERs are computed by sequentially increasing threshold t from 0 to 255, and include vectors as below:

- **Bounding box** *(x,y,width,height)*: coordinate of top-right corner, width and height of the region.
- **Perimeter** p: length of the region boundary.
- **Area** a: number of pixels in the region.
- **Euler number** e: a topological feature based on changes of 2×2 pixel in the binary image which indicates the difference between the number of connected components and number of holes.
- **Horizontal crossings** h_i: a feature about transitions between pixels belonging to and not belonging to the region in a given row i of the region.

In this paper, we also compute the incrementally computable descriptors (compactness $\sqrt{a}/p$, number of holes $1-e$, horizontal crossings feature h and aspect ratio w/h) for each region r and use the AdaBoost classifier presented in [12] to estimate the class-conditional probability $p(r|character)$. Then the local maximum of the class-conditional probability is selected only if the local maximum of the probability is greater than a threshold p_{min} and the difference between local maximum and local minimum is over a limit $\triangle_{min}$.

Fig. 3. Selected ERs for each channel. (a) Origin license plate. (b) Blue. (c) Green. (d) Red. (e) Intensity. (f) Intensity gradient.

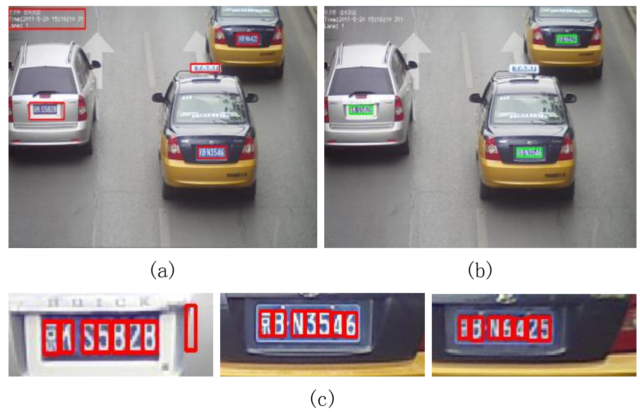


Fig. 4. Coarse-to-fine license plate detection. (a) Coarse detection. (b) Accurate detection. (c) Characters segmentation though bounding boxes of selected ERs

According to the character detector aforementioned, we can select a set of ERs as character regions in each channel for each candidate license plate region (see Fig.3). Only the ERs with probabilities $p(r|character)$ more than 0.75 are selected. Another filter method based on the standard license plate character bounding boxs aspect ratio and area is applied to get the accurate character regions (see Fig.4 (c)). In our study, we set the threshold value of height-width ratio to 1-10, and threshold value of area to 80-800. Additionally, p_{min} is set to 0.5 and $\triangle_{min}$ is set to 0.2.

3) Accurate Localization of License Plates: Considering the geometrical relationship among characters in standard Chinese license plates, we can use some geometrical parameters to remove the disturbing regions and infer the locations of missed characters. The length-width ratios of license plate and characters both stay within a certain range. In addition, the number of adjacent ERs (corresponding to characters) allows a maximum of 7. In this study, a candidate license plate region is identified only if the number of characters detected in this region is between 5 and 10. As shown in Fig.4 (b) and Fig.4 (c), coarse-to-fine license plate localization is achieved, and meanwhile the character regions are segmented accurately.

III. License Plate Recognition

A. Self-adaptive Evolutionary Extreme Learning Machine

Feedforward neural networks (FNN) have been widely used in many fields due to their tremendous approximation capabilities. Gradient-descent (e.g., the backpropagation algorithm), standard optimization (e.g., support vector machines) and least-square(e.g., radial basis function) are three major methods for training FNN [20]. But these methods have a slow learning speed, which has become a bottleneck in practical applications. Extreme learning machine (ELM) was proposed by Huang *et al.* in [21] as a new learning algorithm for single-hidden layer feedforward neural network (SLFN). This learning scheme, termed as ELM, can achieve a speed that is thousands of times faster than conventional learning methods [21].The essence of ELM is that the hidden layer of SLFNs need not be tuned.

In this paper, we use a learning algorithm named Self-adaptive Evolutionary Extreme Learning Machine (SaE-ELM) [11] for a SLFN to recognize the plate numbers. In SaE-ELM, the network hidden node parameters are optimized by the self-adaptive differential evolution algorithm [22], whose trial vector generation strategies and their associated control parameters are self-adapted in a strategy pool by learning from their previous experiences in generating promising solutions, and the network output weights are computed using the Moore-Penrose (MP) generalized inverse [11].

For a training set consisting of N arbitrary distinct samples $S=\{(x_j,t_j)|x_j\in R^n, t_j\in R^m, j=1,2,3,...,N\}$, the SaE-ELM algorithm with $\tilde{N}$ hidden nodes includes following steps.

Step1. Initialization

Initialize the population of the first generation including all the hidden node parameters to a set of NP vectors as (6):

$$\theta_{k,G}=[w^T_{1,(k,G)},...,w^T_{\tilde{N},(k,G)},b_{1,(k,G)},...,b_{\tilde{N},(k,G)}] \tag{6}$$

where w_i is N-dimensional weight vector connecting ith hidden node and input neurons, and b_i is the bias. Both w_i and b_i are generated randomly. In addition, $k=1,2,3,...,NP$ and G denotes the generation number.

Step2. Calculate the output weight

For using $\tilde{N}$ hidden nodes to approximate N samples, the output weight β can be analytically determined by finding a least-square solution as follows.

$$H_{k,G}=\begin{bmatrix} g(w_{1,(k,G)}\cdot x_1+b_{1,(k,G)}) & \cdots & g(w_{\tilde{N},(k,G)}\cdot x_1+b_{\tilde{N},(k,G)}) \\ \vdots & \ddots & \vdots \\ g(w_{1,(k,G)}\cdot x_N+b_{1,(k,G)}) & \cdots & g(w_{\tilde{N},(k,G)}\cdot x_N+b_{\tilde{N},(k,G)}) \end{bmatrix}_{N\times\tilde{N}} \tag{7}$$

$$\beta_{k,G}=H^{\dagger}_{k,G}T \tag{8}$$

where $g(\cdot)$ is an activation function, $H_{k,G}$ denotes the hidden layer output matrix of the neural network, $H^{\dagger}_{k,G}$ is the Moore-Penrose generalized inverse of $H_{k,G}$, T is the label matrix.

In this step, the next generation $\theta_{k,G+1}$ is determined by the root mean square error (RMSE) which is calculated using the following equations:

$$RMSE_{k,G}=\sqrt{\frac{\sum_{i=1}^{N}\|\sum_{j=1}^{\tilde{N}}\beta_j g(w_{j,(k,G)}\cdot x_i+b_{j,(k,G)})-t_i\|}{m\cdot N}} \tag{9}$$

$$\theta_{k,G+1}=\begin{cases} u_{k,G+1}, & \text{if } RMSE_{\theta_{k,G}}-RMSE_{u_{k,G+1}}>\lambda\cdot RMSE_{\theta_{k,G}} \\ u_{k,G+1}, & \text{if } |RMSE_{\theta_{k,G}}-RMSE_{u_{k,G+1}}|<\lambda\cdot RMSE_{\theta_{k,G}} \\ & \text{and } ||\beta_{u_{k,G+1}}||<||\beta_{u_{k,G}}|| \\ \theta_{k,G}, & \text{else} \end{cases} \tag{10}$$

where m is the number of classes for categorization, $u_{k,G+1}$ is the trial vector at the $G+1$th generation for differential evolution and λ is the preset small positive tolerance rate.

Step3. Mutation and crossover

The trial vector generation strategy for each target vector in the current generation is chosen according to the self-adapting probabilities from a candidate pool, which is constructed by four strategies, i.e., DE/rand/1, DE/rand/2, DE/rand-to-best/2, and DE/current-to-rand/1. Refer to [11] for more information. For each target vector, a set of crossover rate CR and control parameters F are randomly generated based on the Gaussian distributions with mean 0.3 and standard deviation 0.1, mean 0.6 and standard deviation 0.3, respectively.

Step4. Evaluation

According to (10), all the vectors $u_{k,G+1}$ are evaluated to get the suitable output weight which results in better generalization performance for the SLFN.

The essence of SaE-ELM is to choose a trial vector generation strategy based on the self-adapting probabilities, while crossover rates CR can be gradually adjusted according to the previous experiences. Step 3 and step 4 are repeated to obtain the best parameters for the SLFN. Only the number of populations NP needs to be assigned manually, and in this study, NP is set to 10. Besides, the tolerance rate λ is set to 0.015.

B. Recognition

When we localize the license plate, we also segment the character regions based on suitable ERs. But different character regions have different sizes (with width in pixel from 4 to 24, height from 18 to 43). In the experiment, we rescale each character region into a consistent size of 20×40. Then we extract the HOG feature in each character region. HOG describes distributions of edges at different parts of an image, and is very effective in representing the license plate characters.

Chinese license plate characters are composed of 31 Chinese characters, 10 digits and 24 alphabets (excluding letters I and O). As a result, we trained the classification model using 1435 labeled characters with 500 hidden nodes and 65 outputs based on the SaE-ELM algorithm. On a standard PC, the training time is 118 seconds, and we get a recognition accuracy of 98.6% for 522 testing characters in 0.12 seconds.

IV. Experiment and Results

A. Dataset and Evaluation Criteria

Since there is no commonly available Chinese license plate dataset, we build a dataset to verify the effectiveness of the proposed method. Our dataset consists of 1007 images taken from a variety of traffic scenes. The resolutions of these images are of three kinds: 1936×2592, 1280×736 and 720×280. The images are separated into four subsets, as listed in Table I. This dataset is very challenging due to various illumination conditions, complex backgrounds, and blurred license plates. In order to evaluate the algorithm performance, ground truth is acquired by manually labeling the positions and contents of the license plates. It is noted that 20 pixels is typically the lowest height of identifiable characters in most recognition methods [1]. As a result, in this paper, the significant license plates regions is with height from 18 to 43 and width more than 90 in pixel.

TABLE I
Image Database Compositions

Datasets	Taken time	Resolution	Images
Subset 1	Day	$1936 \times 2592, 1280 \times 736$	272
Subset 2	Day	720×280	273
Subset 3	Night	1936×2592	241
Subset 4	Night	720×280	221

In this paper, we consider blue (B), green (G), red (R), intensity (I), intensity gradient (∇) and their combination of channels to detect ERs. Similar to the PASCAL detection criterion, it is correct only if the overlap of the detected and ground truth bounding box is above 0.5. According to the experimental results (see Table II) for image subset 2 (a single license plate exists in each image, which is taken in daytime), the combination of G channel and R channel is adopted in our experiment because it achieves the best trade-off between processing time and detection performance.

We estimate the overall accuracy by calculating separately the percentage of license plates correctly detected and characters correctly recognized [1]. Hence, three indicators, namely LDR (license plate detection rate), CRR (character recognition rate), and OVR (overall recognition rate, the percentage of license plates correctly detected and with all the characters correctly recognized), are used to evaluate the proposed approach.

B. Experimental results

All the experiments are performed on a standard PC with 2.66-GHz Intel Core 2 Quad CPU and 2-GB RAM. The experimental results and comparison with methods [10] for each image subset are listed in Table III. Our method achieves better performance for MSER is a special case of ER. Moreover, the ERs are extracted in two channels while the MSER based method [10] in gray level. For the recognition step, template-based method [10] which classified the characters by the Euclidean distance get worse performance because the accumulated error of 180 dimensions affects a lot.

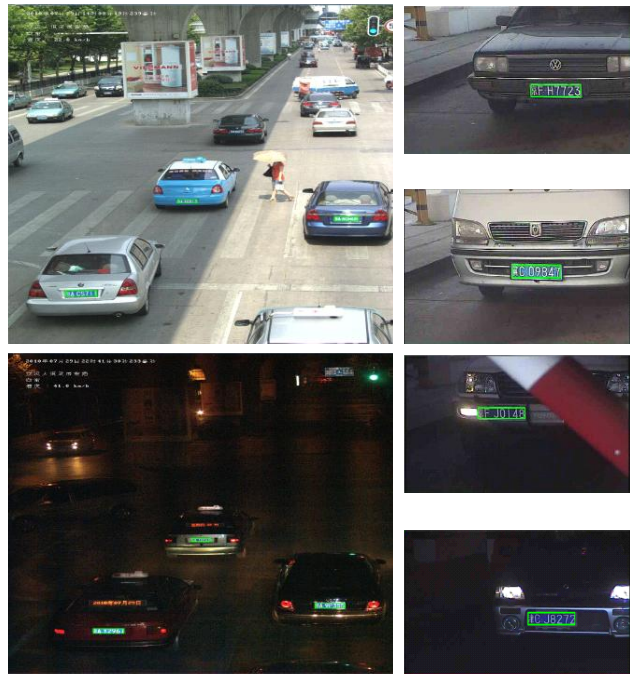


Fig. 5. Examples of location results from data set 1 to 4

TABLE II
License Plate Detection Results for Image Subset 2

Channel	Recall(%)	Precision(%)	Average Time(s)
B	91.2	99.6	0.49
G	94.5	100	0.46
R	95.2	99.6	0.46
I	93.8	99.6	0.47
∇	1.1	100	0.49
G&R	**96.7**	**100**	**0.90**
R&L	96.0	99.2	0.92
G&R&I	97.1	98.1	1.33

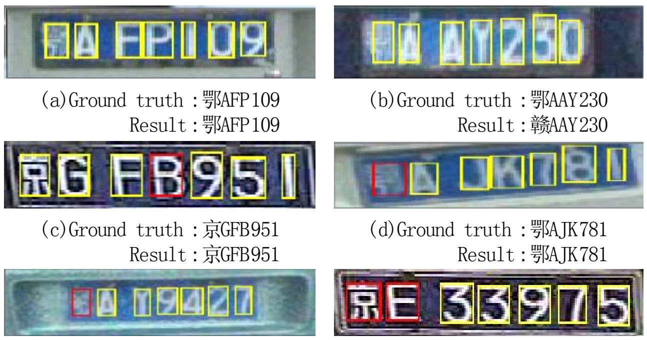
Some images with location results of each set obtained by the proposed method are shown in Fig.5. Our method can detect multi-plates in each image and recognize complex Chinese characters. We present examples of the final character segmentation and recognition results in Fig.6. It can be seen that most of the characters can be correctly recognized despite some license plates are skewed and have low resolution. However, some Chinese characters may be wrongly recognized because they have many strokes. But this can be solved by adding more robust training samples or extracting more distinguishable features from character regions.

V. Conclusions

In this paper, an effective approach to license plate detection and recognition is proposed, based on class-specific ERs and SaE-ELM. Firstly, top-hat transformation, various

TABLE III
EXPERIMENTAL RESULTS OF DATABASE AND COMPARISON

Datasets	Our Approach			MSER and Templates Based [10]		
	LDR	**CRR**	**OVR**	**LDR**	**CRR**	**OVR**
Subset 1	93.5%(404/432)	97.3%(2751/2828)	91.0%	89.6%(387/432)	93.0%(2518/2709)	83.3%
Subset 2	96.7%(264/273)	97.4%(1800/1848)	94.2%	91.2%(249/273)	93.8%(1635/1743)	85.5%
Subset 3	92.6%(313/338)	95.5%(2093/2191)	88.4%	86.4%(292/338)	91.6%(1872/2044)	79.1%
Subset 4	96.4%(213/221)	97.0%(1446/1491)	93.5%	91.0%(201/221)	93.2%(1311/1407)	84.8%



(a)Ground truth：鄂AFP109 Result：鄂AFP109
(b)Ground truth：鄂AAY230 Result：赣AAY230
(c)Ground truth：京GFB951 Result：京GFB951
(d)Ground truth：鄂AJK781 Result：鄂AJK781
(e)Ground truth：鄂AY9427 Result：津A19427
(f)Ground truth：京E33975 Result：京E33975

Fig. 6. Examples of character segmentation and recognition results (yellow bounding boxes are the detected ERs, and red bounding boxes are the inferred locations of character regions).

filters, different contours and validations are applied to achieve coarse license plate detection. Then, class-specific ERs are selected as character regions through a classifier with decision trees. Accurate character segmentation and coarse-to-fine license plate localization are achieved using geometrical attributes of standard license plates and characters. Finally, the characters are recognized using the offline trained classifier on the basis of SaE-ELM. Experimental results show that the proposed approach achieves a promising performance in detecting and recognizing Chinese license plates. Moreover, it is effective to deal with complex illumination and weather conditions during 24 hours one day. In spite of these successes, the proposed approach also has some limitations in detecting partly-occluded license plates and recognizing blurred characters. In the future, we will focus attention on building more abstract and useful representations for license plates by using deep architectures.

REFERENCES

[1] C. Anagnostopoulos, "License plate recognition: A brief tutorial," *Intelligent Transportation Systems Magazine, IEEE*, vol. 6, no. 1, pp. 59–67, 2014.

[2] K. Deb, V. V. Gubarev, and K.-H. Jo, *Vehicle license plate detection algorithm based on color space and geometrical properties*. Springer, 2009, pp. 555–564.

[3] X. Shi, W. Zhao, and Y. Shen, *Automatic license plate recognition system based on color image processing*. Springer, 2005, pp. 1159–1168.

[4] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2431–2438, 2005.

[5] H. Mahini, S. Kasaei, and F. Dorri, "An efficient features-based license plate localization method," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2. IEEE, pp. 841–844.

[6] I. Paliy, V. Turchenko, V. Koval, A. Sachenko, and G. Markowsky, "Approach to recognition of license plate numbers using neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 4, 2004, pp. 2965–2970.

[7] B. Shan, "Vehicle license plate recognition based on text-line construction and multilevel rbf neural network," *Journal of Computers*, vol. 6, no. 2, 2011.

[8] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 830–845, 2011.

[9] M. Yu and Y. D. Kim, "An approach to korean license plate recognition based on vertical edge matching," in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, vol. 4. IEEE, pp. 2975–2980.

[10] B. Li, B. Tian, Q. Yao, and K. Wang, "A vehicle license plate recognition system based on analysis of maximally stable extremal regions," in *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*. IEEE, pp. 399–404.

[11] J. Cao, Z. Lin, and G.-B. Huang, "Self-adaptive evolutionary extreme learning machine," *Neural processing letters*, vol. 36, no. 3, pp. 285–305, 2012.

[12] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 3538–3545.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, pp. 886–893.

[14] J.-W. Hsieh, S.-H. Yu, and Y.-S. Chen, "Morphology-based license plate detection from complex scenes," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3. IEEE, pp. 176–179.

[15] D. L. Baggio, *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing Ltd, 2012.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[17] W. Wang, Q. Jiang, X. Zhou, and W. Wan, "Car license plate detection based on mser," in *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on*. IEEE, pp. 3973–3976.

[18] B. Li, B. Tian, Y. Li, and D. Wen, "Component-based license plate detection using conditional random field model," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1690–1699, 2013.

[19] J. Matas and K. Zimmermann, *A new class of learnable detectors for categorisation*. Springer, 2005, pp. 541–550.

[20] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[21] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[22] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 398–417, 2009.