Information Fusion 18 (2014) 78–85

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

License plate detection based on multistage information fusion

Zhenjie Yao *, Weidong Yi

School of Electronic and Communication Engineering, University of the Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 4 April 2012
Received in revised form 5 March 2013
Accepted 20 May 2013
Available online 1 June 2013

Keywords:
License plate detection
Adaboost
Multistage information fusion
HSI
SVM

ABSTRACT

Adaboost detector has been successfully used in object detection. In this paper, we propose a new License Plate (LP) detection technique based on multistage information fusion, which is adopted to reduce high false alarm rate in the conventional Adaboost detector. The proposed multistage information fusion system is composed of an enhanced Adaboost detector, a color checking module and an SVM detector, where the latter two stages further check whether the image patch that gets through the Adaboost detector is an LP. Test results of the dataset that consists of 950 real-world images show that the fusion reduces the false alarm rate. The proposed Fusion detector outperforms the conventional Adaboost detector throughout the ROC (Receiver Operating Characteristic) curve. The AUC (Area Under the Curve) of the best Fusion detector reaches 0.9081; however, the AUC of the best Adaboost detector is only 0.8441, which shows that the modification on feature extraction and the multistage information fusion significantly improve the LP detection performance.

1. Introduction

A license plate recognition (LPR) system is an important part of intelligent transportation systems (ITSs). It is used in a wide range of applications including parking charges, pay-per-use road usage and traffic law enforcement, etc. In general, a license plate recognition system is divided into three steps: (1) detecting the license plate; (2) extracting the characters from the license plate one by one; and (3) recognizing the characters from (2). Among the above three steps, License Plate (LP) detection is the most difficult step. If an LP is precisely detected, further processing becomes rather simple. In a practical context, LP detection in an inconstant environment is difficult. The variable factors include illumination variance, complex background, unpredictable weather and tilts caused by viewing angle. These problems are particularly intolerable when the system is used in an outdoor environment.

In the past few decades, LP detection has generated much research. There are three major categories of methods for LP detection: The first category is statistical or morphological operations of low level visual features, such as color, edge, etc. [1–5]. This is the most frequently used LP detection method in the early LPR systems. Visual attention model presents an appropriate framework to integrate the low level features [6]. The second category utilizes middle level features for LP detection, in which shape [7] and symmetry [8] are taken into consideration. Texture analysis by Gabor filter [9], wavelet transform [10] and Hough transform [11] also give impressive results in LP detection. The last category is the methods based on artificial intelligence (AI), including neural networks (NN) [12,13], Adaboost [14], etc. Support vector machine (SVM) is a powerful texture classifier [15]; however, due to its heavy computation requirements, SVM has not been used in LP detection directly. In reference [16], the authors first estimated LP region by motion information, then verified the result by SVM. In an AI based LP detection method, the classifier scans the whole image patch by patch, and determines whether it is an LP. Most of these works perform well in simple and constant backgrounds; but they suffer performance degradation if the background is complex or the environment (such as illumination, climate, viewing angle, distance) changes. Simply put, the challenges of LP detection are still far from being solved.

Since Adaboost has had great success in face detection [17], Dlagnekov tried to apply it to detect license plates and achieved a detection rate of 95.6% with a false alarm rate of 5.7% [14]. Pan et al. tried to detect Chinese license plates using Adaboost [18], whose test results on 200 frames yielded a detection rate of 81% with a false alarm rate of 6.5%. Their results are with a high false alarm rate, not as good as expected. From all the above we can conclude that, the Adaboost LP detectors are far from practical use.

This paper presents a multistage information fusion detector for LP detection, in which information fusion is applied to improve the conventional Adaboost LP detector. The direct contributions of this work are improved feature extraction and multistage information fusion of detectors. The feature extraction of [14,18] for LP detection does not look into the details of LPs. In our implementation, the Harr-like features are restricted in the scale of characters and

* Corresponding author.
E-mail address: yaozhenjie@gmail.com (Z. Yao).

1566-2535/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.inffus.2013.05.008

strokes, which are more meaningful for LP detection. Besides modifications and restraints on feature extraction, a multistage information fusion detector, which is composed of an enhanced Adaboost detector, a color checking module and an SVM detector, reduces the false alarm rate effectively. Tests on 950 real-world car images in complex background show encouraging results, which proves that the modifications on feature extraction and the multistage information fusion do improve the performance of LP detection.

The rest of the paper is organized as follows: Section 2 describes details of a license plate detection algorithm based on Adaboost. In Section 3, a multistage information fusion procedure composed of an Adaboost detector, a color checking module and an SVM detector is presented. Test results on a dataset consisting of real-world car images are shown and discussed in Section 4. Finally, we conclude this paper and look forward to future work in Section 5.

2. Adaboost detector – training and detection

This section provides a detailed description of Adaboost detector. Section 2.1 presents the Harr-like feature extraction for LP detection. Section 2.2 establishes a method of training the Adaboost LP detector offline. And Section 2.3 outlines the online work procedure of the Adaboost LP detector.

2.1. Harr-like feature extraction

In Ref. [17], Viola and Jones proposed four types of Harr-like features for an Adaboost detector to detect faces. Lienhart and Maydt [19] extended the detector by rotating the Harr-like feature. Meanwhile, they extracted another type of feature called center-surround feature. All the types of features mentioned above can be calculated fast by accumulation image, more details about accumulation image could be found in Ref. [17]. The features extracted by [14,18] for LP detection are horizontal and vertical deviations, and all of the features are extracted in the scale of LP. In other words, only coarse Harr-like features are extracted; the feature extractors do not look into details.

The Harr-like features are modified to accommodate our application of license plate detection. The 4-section (diagonal) feature is removed; center-surround feature is added. There are 5 types of feature; all of which are shown in Table 1. Each feature is the sum of pixels which lie within the black sections subtracted by the sum of pixels in the white sections. Harr-like features in our scheme are restricted in the scale of character and stroke, since these Harr-like features are more meaningful for LP detection.

According to the LP standard in China [20], the LPs have an aspect ratio of about 3. All LPs can be scaled to the same size as shown in Fig. 1. The only constraint on the LP size is the aspect ratio of 3, which means that an LP is of size $H * 3H$. LP with smaller H is in low resolution, which would result in information loss; whereas, LP with larger H is in high resolution, which contains meaningless details for LP detection and induces complex computation. As the tradeoff decision, H is set to 20. All the training samples are scaled to $20 * 60$. Consequently, the trained detector could only identify LP of size $20 * 60$.

Table 1
Harr-like features.

Features					
Feature type	1	2	3	4	5
Horizontal range	1–3	1–3	2–7	2–7	5–7
Vertical range	2–14	2–14	1–3	1–3	12–14
Total number	23,595	22,737	14,715	12,753	141

As mentioned above, the feature extraction focuses on stroke scale and character scale. The constraints on scale significantly reduce the computational cost. Through simple measurement we can draw the following conclusions. When an LP is scaled to $20 * 60$, the size of a horizontal stroke is about $2 * 6$, the size of a vertical stroke is about $13 * 2$, and the size of a character is about $13 * 6$. The horizontal and vertical range (in pixel) of white sections are shown in Table 1. For each type of feature, the size of the white section can be any value within an interval given in the table.

Given the basic resolution of $20 * 60$, there are 73,941 features in the exhaustive feature set. The number of each type of feature is also shown in Table 1. It is worth mentioning that the features of type 5 (whose white section is surrounded by black) are extracted in character scale only, and the center section (white color) should be a single character. So the number of center-surround feature is 141, much less than the other types of features.

2.2. Adaboost offline training

LP images (the positive samples) for training are cropped manually; whereas nonLP images (the negative samples) are cropped automatically and randomly from the whole image. There are totally 1800 negative samples and 200 positive samples in the training dataset. Fig. 1a shows some LP images, and Fig. 1b gives some nonLP images. All the images are scaled to $20 * 60$. Given the training images, 73,941 features are extracted from each sample. As shown later, each feature forms a weak-classifier for the Adaboost detector.

The offline training procedure is presented as Algorithm 1.

Algorithm 1. Adaboost offline training procedure

1. Given sample x_i with label y_i, $w_{t,i}$ is the weight of ith sample in tth training round, m and l are the number of positive and negative samples, respectively.
2. Initialize sample weights $w_{1,i} = \frac{1}{2m}$ if $y_i = 1$; $w_{1,i} = \frac{1}{2l}$ if $y_i = 0$.
3. For $t = 1, \ldots, T$

   (1) Normalize the sample weight $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$.

   (2) For each feature j, train a weak-classifier h_j, determine threshold

   θ_j and bias p_j by minimizing weighted error $\varepsilon_j = \sum_i w_{t,i} |h_j(x_i) - y_i|$.

   where $h_j(x_i) = sgn(p_i \cdot (x_i - \theta_i))$, $p_i = \pm 1$,
   $sgn(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$.

   (3) Get weak-classifier h_t with minimum error ε_t.

   (4) Update sample weights by $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$, in which

$$e_i = \begin{cases} 0 & x_i \text{ is correctly classified} \\ 1 & \text{otherwise} \end{cases}, \quad \text{and} \quad \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}.$$

4. The final strong classifier is

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geqslant 0.5 \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

   where $\alpha_t = log\left(\frac{1}{\beta_t}\right)$.

(a) LP images (positive samples) (b) nonLP images (negative samples)

Fig. 1. Examples of training data.

As shown in Algorithm 1, the Adaboost training is to find a series of weak-classifiers and combine them into a strong classifier. The pursuit of "strong" is done by a sample weight update mechanism. The basic idea of the sample weight update is decreasing the weights of correctly classified samples while increasing the weights of misclassified samples. Such a weight update mechanism ensures that the subsequent weak-classifiers pay more attention to the misclassified samples. In other words, subsequent weak-classifiers compensate the weak points of their predecessors; so they can do better on the misclassified samples. The misclassification rate drops as new weak-classifiers get involved, which explains why we can get strong classifier in the end.

All the 73,941 features mentioned above are fed into the training procedure of Adaboost. In total, there are 300 weak-classifiers to be trained. Table 2 lists the distribution of top 10, top 30, top 100 and top 300 features in corresponding columns. As shown in Table 2, type 1 is always the most frequently used feature, which coincides with our intuition that LP has rich vertical texture. Type 2 is not as frequently used as type 1; probably because type 2 is more complex than type 1. Simple features tend to be more frequently used. Though type 5 is infrequently used, it is relatively frequent if we consider that only 141 features of type 5 joined in the competition.

Fig. 2 shows the classification error rate of the strong and weak-classifiers on the training set in different training rounds. With the increase of training rounds, the overall error rate declines sharply. The overall error rate drops to 0 when 40 weak-classifiers are adopted. It is interesting to note that the error rates of most weak-classifiers range from 0.25 to 0.3. But the weak-classifiers did combine to form a strong classifier. In [14,18], the authors determined the final number of weak-classifiers by the results from the training dataset, which means 40 weak-classifiers are sufficient for our LP detection application (in Fig. 2, all the 2000 training examples are correctly classified by combining 40 weak-classifiers); however, it may not be the case. We trained more weak-classifiers than the requirements of the training dataset in case more weak-classifiers are needed in practice. As the experimental results in Section 4, taking more weak-classifiers into consideration would achieve better performance in practice. The most probable explanation is that there are differences between the prior distribution of training data and practical data.

Table 2
Feature statistic.

Feature type	Top 10	Top 30	Top 100	Top 300
1	7	16	55	149
2	0	2	4	22
3	1	8	24	77
4	2	3	14	44
5	0	1	3	8

2.3. Adaboost online LP detection

The online detection is illustrated in Fig. 3. As mentioned in subSection 2.1, the Adaboost detector works on image patches of size $20 * 60$, but LP size in the real-world images may range from $10 * 30$ to $100 * 300$. Since it is difficult to resize the detector, we resize image into different size, which leads to an image pyramid. To detect LPs in different scales, a search window of size $20 * 60$ slides over each layer of the pyramid. The detector exhausts all image patches in each scale, checking whether it is an LP.

3. LP checking by multistage information fusion

As shown by results in [14,18], the conventional Adaboost detector in real LP detection application achieves good performance; however, there is still room to reduce the high false alarm rate. It is reasonable to presume that additional checks would be helpful to reduce the false alarms, and it is found to be the case for the application shown in this work.

Color checking [4,21] has been applied to LP detection, and have achieved encouraging performance. In Ref. [16], the SVM algorithm is adopted to verify license plate regions detected by motion information. Here we adopt color checking and SVM as additional checks, which lead to a multistage information fusion detector. Experimental results show that most false alarms of Adaboost detector can be removed by the additional checks.

As shown in Fig. 4, the multistage information fusion system is composed of an enhanced Adaboost detector, a color checking module and an SVM detector. Each image patch that gets through the Adaboost detector is rechecked by the color checking module. Similarly, the SVM detector checks the patches that get through the color checking module. Patches that get through all the three checks are the detected LPs. The multistage information fusion ensures that the color checking module and the SVM detector are applied to the image patch only when it gets through the Adaboost detector; therefore, the fusion would slightly increase the computation time than using only Adaboost detector.

3.1. HSI color checking

According to the latest LP standard [20], there are 3 major types of LP in China: white character with blue background, black character with yellow background and black character with white background. Color checking is to find out whether there is any of the 3 types of the color composition in an LP candidate.

Deb and Kang-Hyun developed an LP detection algorithm in HSI color space [21]. In our work, the color checking is done in HSI color space too. The HSI color space is an intuitive and perceptually relevant approximation of human being's color perception, in which color is expressed as a 3-dimension vector: hue (H), saturation (S), and intensity (I). Further details about the HSI color space can be found in [22].

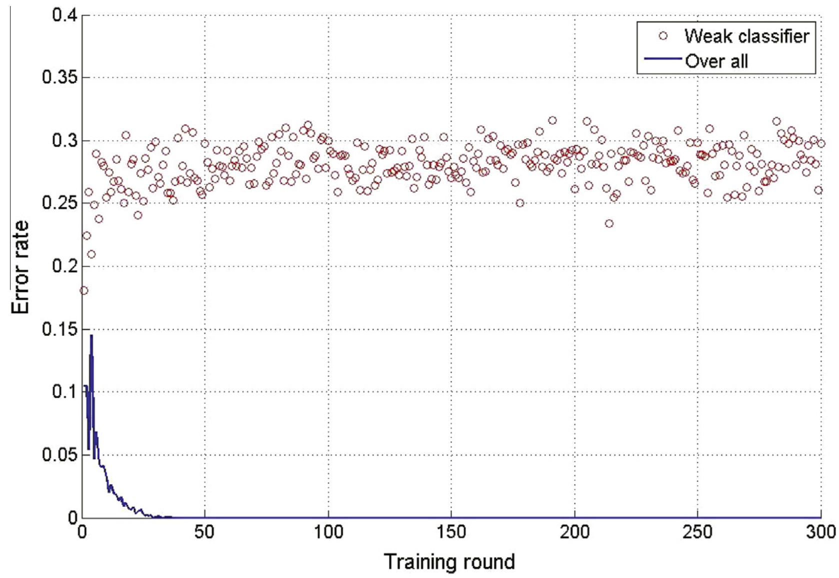


Fig. 2. Learning curve of Adaboost.

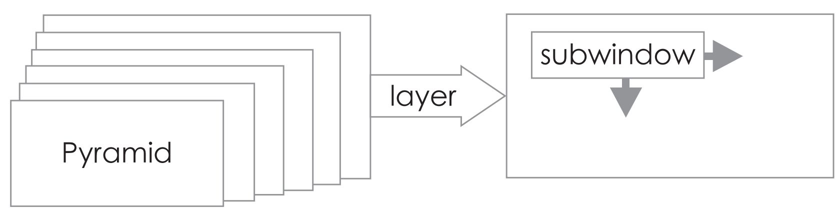


Fig. 3. Online detection.

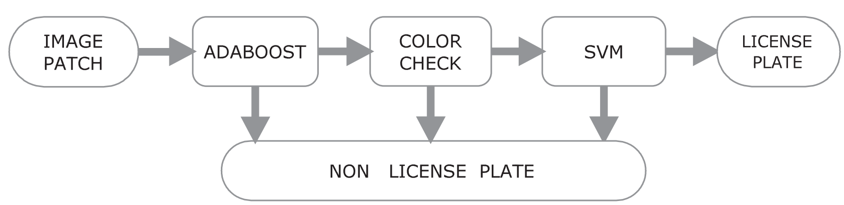


Fig. 4. Multistage fusion for LP check.

The Colors of LPs may be affected by illumination or fade; for example, a blue LP may look purple or green in particular illumination. The HSI range of each object color is listed in Table 3, they are empirically determined by statistics of the training dataset mentioned in Section 2.2.

The color checking reduces the false alarms rate by identifying LP candidates as nonLPs if there is no typical LP color composition. First, the color composition of the LP candidate from the Adaboost detector will be analyzed. To be specific, the proportion of the 4 colors in Table 3 is calculated. For example, given an LP candidate, if the number of both blue and white pixels is larger than 5% of the total number of pixels in the candidate region, the LP candidate will get through the color checking module and further be checked by the SVM detector. Otherwise, if none of the three types of color compositions is found, the LP candidate is recognized as a nonLP, and no further checking is needed.

It can be found that tightening of the HSI range of each color will lead to false rejection; enlarging of the HSI range will lead to false alarm. On one hand, the Adaboost detector at the beginning can filter out most of the potential false alarms; on the other hand, the SVM detector at the end can remove the false alarms that induced by color checking. Considering the 2 companion detectors, we decide to enlarge the HSI range of each color slightly to avoid missing detection.

Table 3
The HSI range of each color on LPs.

Color	H	S	I
Blue	0.45–0.66	0.1–1	0.4–0.95
Yellow	0.08–0.19	0.1–1	0.2–0.95
White	0–1	0–0.2	0.8–1
Black	0–1	0–1	0–0.18

3.2. SVM detector

SVM is a machine learning algorithm that searches maximum-margin hyperplane for data classification [23,24]. In the LP detection application, the SVM detector works as a binary texture classifier with 1 for a positive (LP) patch and -1 for a negative (nonLP) patch.

The classical SVM is a linear binary classifier. The decision function of a linear SVM is defined as:

$$f(x) = sgn\left(\sum_{i=1}^{N}\alpha_i y_i (x_i, x) + b\right). \tag{2}$$

Generally, datasets from real-world are not linearly separable. A nonlinear kernel function is needed to map the original data to data in high dimension space, where the data is linearly separable. Here we adopt SVM using Gaussian Radial Basis Function (RBF) kernel. The nonlinear decision function is

$$f(x) = sgn\left(\sum_{i=1}^{N}\alpha_i y_i K(x_i, x) + b\right). \tag{3}$$

in which $K(x_i, x)$ is the Gaussian RBF, which is defined by

$$K(x, x') = \exp\left(-\frac{\|x - x'\|}{\sigma^2}\right). \tag{4}$$

The parameters α and b are solved by optimization under the principle of maximum-margin. Details of SVM can be found in [23,24].

In this application, the SVM detector shares the same training dataset with Adaboost, which is described in Section 2.1. First, the horizontal (Fig. 5b) and vertical (Fig. 5c) jumping pixels (pixels with strong contrast with its neighbors) are extracted. The size of statistical sub-window is $6 * 10$, which overlaps with each other by half. Given the fact that the size of an image from training dataset is $20 * 60$, the total number of sub-windows is 50. For each sub-window, we count the number of horizontal (Fig. 5b) and vertical (Fig. 5c) jumping pixels in it, respectively. The dimensionality of the features is 100. Such a simple feature extraction gives impressive performance; 3-fold cross validation on the training dataset achieved an accuracy of 97.8%. It is worth noting that, the feature is the number of jumping pixels, which is apparently independent with the LP color composition.

3.3. Post processing

As shown in Fig. 6a (enlarged to make the multiple detections distinguishable, in which LPs detected by the multistage fusion detector are marked with red boxes), a single LP may be detected several times, the different detections of the same LP may differ in scale or position (neighbor patches have large overlaps between each other). To handle the problem of multiple detections of the same LP, LPs with overlaps with each other are put into the same group. In Fig. 6a, all the red boxes are in the same group. The mean of detections in the same group is the final detected LP, which is shown in Fig. 6b.

i. *About the order of the 3 stages.* In fact, the serial fusion is equivalent to logic AND, an image patch is recognized as an LP if and only if it is recognized as an LP by all the 3 stages of the fusion. The logic AND is commutative, which indicates that the fusion order is not important for the detection performance. Considering both the computation speed and detection performance, Adaboost is placed first, then comes color checking module. SVM is put in the end because it is the most time consuming one.

ii. *About the contributions of the 3 stages.* To find out the contribution of each stage, the Receiver Operating Characteristic (ROC) curves of each stage is shown in Fig. 7. As shown in the figure, an Adaboost detector followed by a color checking module do better than a simple Adaboost detector; meanwhile, an LP detection system with SVM detector at the end does better than that composed of Adaboost and color checking module. Simply put, all the 3 stages make their unique contributions, none of them are neglectable.

4. Experimental results

In this section, the detection method is verified by a dataset composed of real-world car images. A surveillance camera was installed at the left side of an exit of our campus. 950 images were captured during 11 h with constantly changing context. The resolution of each image is $576 * 720$, and 858 of the 950 images in the dataset include a car with a license plate. The other 92 images in the dataset contain no license plate, among which some include very small sections of LP, some include a new car without an LP, and some include pedestrians or bicycles. It is worth mentioning that our test dataset is more complex than other datasets. Reference [14] detected LP with black character and white background only, while reference [18] detected LP with white character and blue background only. Our dataset consists of all 3 major types of Chinese LP. The images are collected in a real-world situation; all the cars passed by without any extra restraint.

For ease of presentation, the multistage information fusion detector is named Fusion detector, and the conventional Adaboost is named Adaboost detector.

Fig. 8 shows some cases in which the Fusion detector performs well. LPs detected by the Fusion detector are marked by red boxes. The Fusion detector can detect an LP from the complex background; whether the LP is small (Fig. 8a), yellow (Fig. 8b), or white (Fig. 8c), and even if it suffers from fade (Fig. 8d), tilts (Fig. 8e), or color deviation (Fig. 8f). We find that the Fusion detector is robust to context changes; it could detect the LPs even under an adverse environment.

Image patches with texture and color similar to LPs are easy to be mistaken for LPs. For example, (1) blue jeans of a bicycler; (2) vehicle logo; and (3) other characters on the car. All these image patches have similar texture as LPs, and will lead to false detections, which unfortunately, can't be avoided by our scheme completely.

Fig. 9 presents the ROC curves of both detectors obtained from the dataset. Both detectors take a variable number of features into consideration. Because it is difficult to count the total number of negative samples in an image, the x-axis of the ROC curves is the number of false alarms instead of the rate of false alarm. Each curve is an interpolation result of 21 real test points obtained by modifying the threshold of the Adaboost detector. Fig. 9a presents the ROC curves of the Adaboost detector with 30, 50, 100, 150, 200, 300 features, respectively. Fig. 9b presents the ROC curves of the Fusion detector with the corresponding number of features. By analyzing and comparing the two figures, we have found out the following valuable points:

(a) LP image (b) Horizonal jumping pixels (c) Vertical jumping pixels

Fig. 5. Feature extraction for SVM.

(a) Multiple detections of the same LP (enlarged to make the detections distinguishable). (b) Mean location of multiple detections.

Fig. 6. LP detected for multiple times.

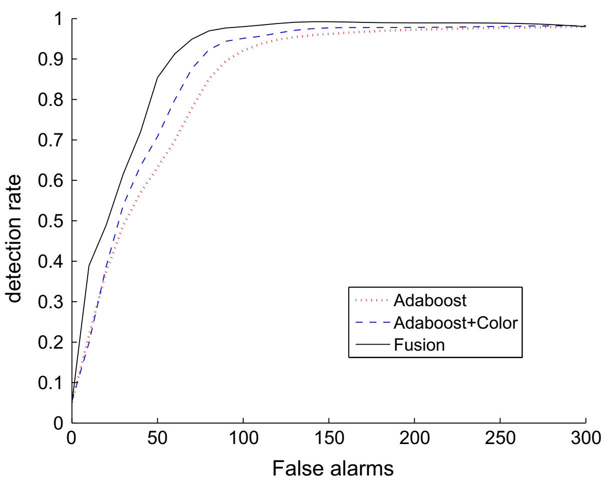


Fig. 7. ROC comparison of each stage. 300 weak-classifiers adopted.

i. For each curve, with the increase of false alarms, the detection rate of the detector increases.
ii. Approximately speaking, as more features are included, both detectors tend to get improved results. The detectors with 30 features achieved the worst performance; whereas the detectors with 300 features achieved the best performance. By increasing features, we have increased the constraints on LP detection, thereby reducing the false alarms.
iii. In both figures, detectors with 200 and 300 weak-classifiers achieved similar performance. Recall the learning curve in Fig. 2, the error rate of Adaboost goes to 0 within about 40 rounds, which means 40 weak-classifiers are enough to classify the training dataset correctly. More weak-classifiers would not improve the performance on training dataset. However, in Fig. 9, it is found that as more weak-classifiers are included, the performance of the detectors do improve obviously. As stated before, detectors with 200 and 300 weak-classifiers achieve similar performance, and so we draw the conclusion that more features would lead to limited improvements; 300 features are enough for the test

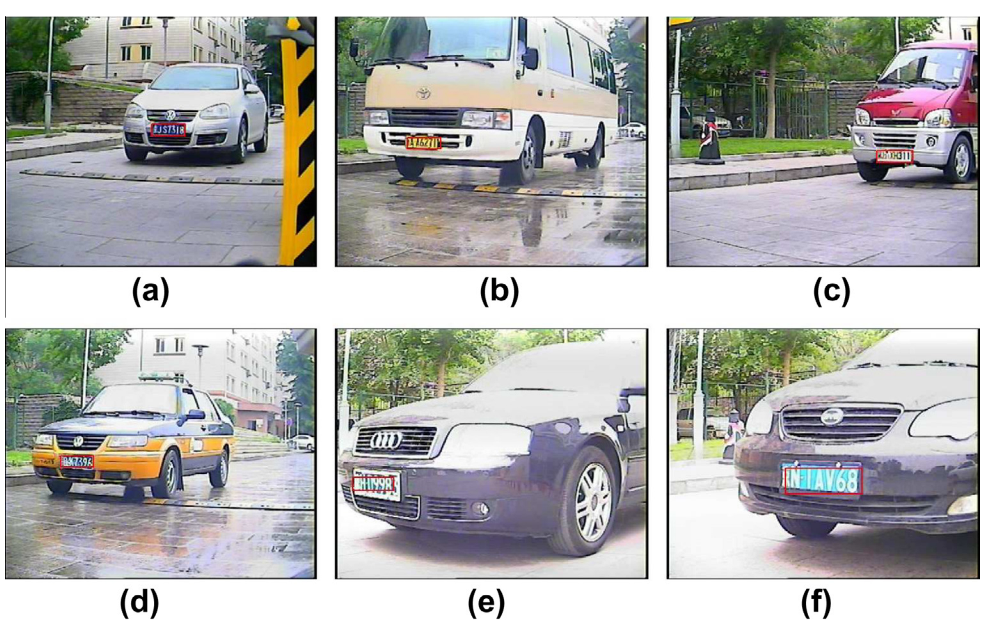
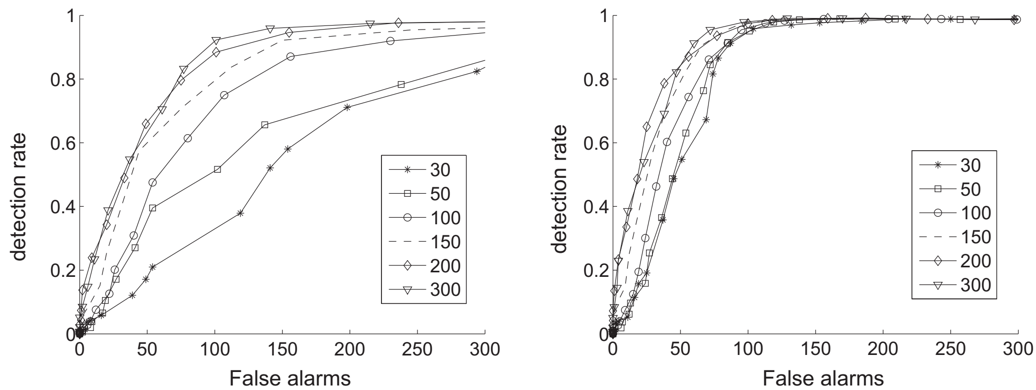


Fig. 8. Some images in which LP can be successfully detected. Detected LP regions are marked by red boxes.



(a) ROC curves of the Adaboost detector (b) ROC curves of the Fusion detector

Fig. 9. ROC curves of the two methods. Each method adopts various numbers of weak-classifiers.

dataset. Notice that all the above analysis is based on a particular test dataset. Practical situations differ from each other, so the number of weak-classifiers should be determined accordingly.

iv. Additionally, we found that the Adaboost detector is more sensitive to the number of features than the Fusion detector. As shown in Fig. 9a, the performance of the Adaboost detectors with different number of features is significantly different from each other (the ROC curves are far from each other, and it is easy to distinguish every curve); which indicates that the Adaboost detector is very sensitive to the number of features. However, as shown in Fig. 9b, the Fusion detectors with different number of features get similar results (the ROC curves are much closer to each other than the ROC curves in Fig. 9a; they are difficult to distinguish); which manifests that the Fusion detector is less sensitive to the number of features.

For comparative analysis, Fig. 10 shows several ROC curves of both detectors. With the increase of false alarms, the detection rate of the Fusion detector increases more rapidly than that of the Adaboost detector. If both detectors use the same number (30, 100, 300) of features, the Fusion detector outperforms the Adaboost detector throughout the curve. Further more, the Fusion detector with 30 features achieves similar performance with the Adaboost detector with 300 features. The Fusion detector achieves much better performance than the Adaboost detector, even with fewer features. All the facts above demonstrate that the color checking module and the SVM detector can remove most false alarms of the Adaboost detector, which leads to refined detection performance.

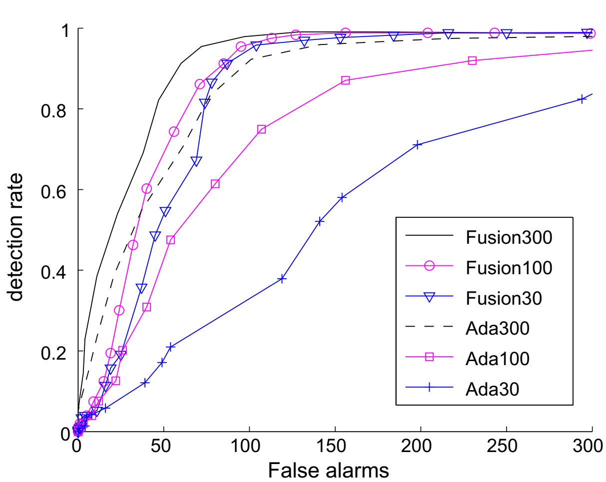


Fig. 10. ROC comparison of two detectors. Each method adopts various numbers of weak-classifiers.

Table 4
LP detection performance comparison between Adaboost detector and fusion detector with 300 features.

Detector	False alarm numbers					
	30	60	100	150	200	260
Adaboost (%)	47.95	69.05	91.92	96.19	97.25	97.99
Fusion (%)	60.76	90.85	97.96	99.24	98.97	98.62

Table 5
AUC of each ROC curve.

Detector	Number of features					
	30	50	100	150	200	300
Adaboost	0.4978	0.5955	0.7208	0.7918	0.8367	0.8441
Fusion	0.8187	0.8318	0.8555	0.8876	0.9081	0.9054

For the sake of clarity, Table 4 lists some selected points on the ROC curves. Both detectors are with 300 features. The data in the table are obtained by interpolation. Based on Table 4 we can draw the same conclusion as the ROC curves in Fig. 10; the Fusion detector outperforms the Adaboost detector throughout the points. For example, when both detectors report 60 false alarms, the Fusion detector achieved a detection rate of 90.85%, while the Adaboost detector achieved 69.05% only.

The AUC (Area Under the Curve) is a measurement of detection accuracy, which is defined as the ratio of the area under the ROC curve to the area of the whole figure. The larger the AUC is, the better performance the detector achieves. Table 5 shows AUC of all the ROC curves mentioned previously. For both detection methods, the AUC increases as the number of features increases. The Fusion detector has larger AUC when both detectors use the same number of features. The AUC of the Adaboost detector ranges from about 0.5 to 0.85; however, the AUC of the Fusion detector ranges from 0.8 to 0.91. The AUC of the best Fusion detector reached 0.9081; while the AUC of the best Adaboost detector is only 0.8441. Fusion detector with 100 features achieved an AUC of 0.8555, which is larger than AUC of the best Adaboost detector. Once again, we find the benefits of multistage information fusion. As we have expected, the color checking and SVM detector removed most false alarms of the Adaboost detector.

5. Conclusions

Conventional Adaboost detector is enhanced by multistage information fusion for LP detection. Major modifications include: First, the Harr-like feature extraction is modified and restricted in the scale of character and stroke to accommodate for LP detection. Second, multistage information fusion is adopted to reduce false alarm rate of the conventional Adaboost LP detector. The whole multistage information fusion system is composed of an enhanced Adaboost detector, a color checking module and an SVM detector, the latter two of which further check whether the image patch that gets through the Adaboost detector is a license plate.

Test results, derived from a dataset that consists of 950 real-world car images with complex background, show that the Fusion detector decreases the false alarm rate effectively. No matter how many features get involved, the Fusion detector outperforms the conventional Adaboost detector throughout the ROC curve. For example, while the false alarm number of both detectors is 60, the best Fusion detector detects 90.85% license plates; the best Adaboost detector detects only 69.05%. Comparison of the AUC of each ROC curve leads to the same conclusion. Specifically, the AUC of the best Fusion detector reach 0.9081, while the AUC of the best Adaboost detector is only 0.8441. The Fusion detector is capable to detect various types of LPs. All the above indicates that modification on feature extraction and fusion did remove most false alarms of the Adaboost detector; and improve the detection performance.

What makes the results are convincing is that the three stages are complementary. They check each input patch from different, almost independent points of view. Specifically speaking, the Adaboost detector identifies LP by the regional contrast clues (the Harr-like features); color checking module identifies LP by color clues; while SVM identifies LP by the global distribution of jumping pixels. The multistage information fusion combines them into a more comprehensive detector, which significantly improves the LP detection performance.

Besides the conclusion above, it is also found that more weak-classifiers are needed for the Adaboost detector in practical use. Because there are differences between training data and practical data, the number of weak-classifiers should be determined according to the real-world data, rather than determined by the training dataset only.

Although the Fusion detector is designed for detecting Chinese LP in our implementation, it can be extended for detecting LP in other countries after modifications of the color checking module.

Acknowledgments

The authors thank the anonymous reviewers, their comments and suggestions helped to improve the quality of the paper. We also want to thank Mingye Wang for his patient language editing. This work was supported by grants partly from the National Science and Technology Major Project (No. 2010ZX03006-001-02), and partly from One Hundred Talents Project of The Chinese Academy of Sciences (No. 99M2008M02).

References

[1] C. Anagnostopoulos, I. Anagnostopoulos, I. Psoroulas, V. Loumos, E. Kayafas, License plate recognition from still images and video sequences: a survey, IEEE Transactions on Intelligent Transportation Systems 9 (3) (2008) 377–391.

[2] D. Zheng, Y. Zhao, J. Wang, An efficient method of license plate location, Pattern Recognition Letters 26 (15) (2005) 2431–2438.

[3] S. Wang, H. Lee, Detection and recognition of license plate characters with different appearances, 2003 IEEE Proceedings on Intelligent Transportation Systems 2 (2003) 979–984.

[4] X. Shi, W. Zhao, Y. Shen, Automatic license plate recognition system based on color image processing, Lecture Notes on Computer Science 3483 (2005) 1159–1168.

[5] B. Hongliang, L. Changping, A hybrid license plate extraction method based on edge statistics and morphology, in: Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, 2004, pp. 831–834.

[6] Z. Yao, W. Yi, License plate location based on improved visual attention model, in: International Conference on Machine Vision (ICMV), 2010, pp. 562–566.

[7] C. Oz, F. Ercal, A practical license plate recognition system for real-time environments, in: Proc. IWANN, Springer, 2005, pp. 881–888.

[8] D. Kim, S. Chien, Automatic car license plate extraction using modified generalized symmetry transform and image warping, in: IEEE International Symposium on Industrial Electronics. ISIE 2001, vol. 3, 2001.

[9] F. Kahraman, B. Kurt, M. Gokmen, License plate character segmentation based on the Gabor transform and vector quantization, Lecture Notes in Computer Science (2003) 381–388.

[10] C. Hsieh, Y. Juan, K. Hung, Multiple license plate detection for complex background, in: Proceedings of the 19th International Conference on Advanced Information Networking and Applications, 2005, pp. 389–392.

[11] C. Nguyen, M. Ardabilian, L. Chen, Real-time license plate localization based on a new scale and rotation invariant texture descriptor, in: 11th International IEEE Conference on Intelligent Transportation Systems, 2008, pp. 956–961.

[12] M. Chacon, A. Zimmerman, License plate location based on a dynamic PCNN scheme, in: Proc. Int. Joint Conf. Neural Netw., vol. 2, 2003, pp. 1195–1200.

[13] Y. Chen, C. Han, K. Fan, B. Jeng, C. Wang, The application of convolutional neural network on face and license plate detection, in: Proc. 18th Conf on Computer Vision Graphics and Image Processing, CVGIP, 2005, pp. 31–37.

[14] L. Dlagnekov, License Plate Detection Using AdaBoost, Computer Science & Engineering Dpt., San Diego, La Jolla. <http://www.cse.ucsd.edu/classes/fa04/cse252c/projects/louka.pdf>.

[15] K. Kim, K. Jung, S. Park, H. Kim, Support vector machines for texture classification, IEEE Transactions on Pattern Analysis and Machine Intelligence (2002) 1542–1550.

[16] I. Tsai, J. Wu, J. Hsieh, Y. Chen, Recognition of vehicle license plates from a video sequence, IAENG International Journal of Computer Science 36 (1) (2009) 26–33.

[17] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001, vol. 1, IEEE, 2001, pp. I–511.

[18] P. Shizhu, S. Weiqun, W. Lingqun, Vehicle license rapid location based on Adaboost, Computer Engineering 12 (2006).

[19] R. Lienhart, J. Maydt, An extended set of haar-like features for rapid object detection, 2002 International Conference on Image Processing. 2002. Proceedings, vol. 1, IEEE, 2002, pp. I–900.

[20] GA36-2007, License plate of motor vehicle of the People's Republic of China, 2007.

[21] K. Deb, K. Jo, Hsi color based vehicle license plate detection, in: International Conference on Control, Automation and Systems, 2008. ICCAS 2008, IEEE, 2008, pp. 687–691.

[22] A. Koschan, M. Abidi, Digital Color Image Processing, Wiley Interscience, 2008.

[23] X. Zhang, Pattern Recognition, Tsinghua Publication House, 2010.

[24] S. Canu, Y. Grandvalet, V. Guigue, A. Rakotomamonjy, Svm and kernel methods matlab toolbox, Perception Systems et Information, INSA de Rouen, Rouen, France (2005).