Real-time Traffic Light Recognition Based on Smart Phone Platforms

Wei Liu, Shuang Li, Jin Lv, Bing Yu, Ting Zhou, Huai Yuan, and Hong Zhao

Abstract—**Traffic light recognition is of great significance for driver assistance or autonomous driving. In this paper, a traffic light recognition system based on smart phone platforms is proposed. First, an ellipsoid geometry threshold model in HSL color space is built to extract interesting color regions. These regions are further screened with a post-processing step to obtain candidate regions which satisfy both color and brightness conditions. Second, a new kernel function is proposed to effectively combine two heterogeneous features, HOG and LBP, which is used to describe the candidate regions of traffic light. A Kernel Extreme Learning Machine (K-ELM) is designed to validate these candidate regions and simultaneously recognize the phase and type of traffic lights. Furthermore, a spatial-temporal analysis framework based on a finite state machine is introduced to enhance the reliability of the recognition of the phase and type of traffic light. Finally, a prototype of the proposed system is implemented on a Samsung Note3 smart phone. To achieve a real-time computational performance of the proposed K-ELM, a CPU-GPU fusion based approach is adopted to accelerate the execution. Experimental results on different road environments show that the proposed system can recognize traffic lights accurately and rapidly.**

Index Terms—**Traffic light recognition, smart phone, geometry threshold model, kernel extreme learning machine, finite state machine.**

I. Introduction

TYPICAL traffic scenes contain a lot of traffic information, such as road signs, road markings, traffic lights, etc. Usually, it is not easy for the drivers to keep attention to the various presenting traffic information. The distraction, visual fatigue and understanding errors of the drivers can lead to severe traffic accidents. Especially, as the traffic lights are used to direct the pedestrians and vehicles to pass the intersections orderly and safely, it is of great importance to recognize and understand them accurately. Therefore, many research institutions are striving to recognize the traffic lights using in-car cameras to assist the driver to understand driving conditions. This function is critical to driving assistance or even autonomous driving [1-8]. For example, in order to drive safely through road intersections, Google's self-driving car has mounted a camera positioned near the rear-view mirror for traffic light recognition [8]. In recent years, with the increase of computation power, the application of smart phones in the driving assistance gradually becomes a hot research field [9-12]. Compared with the commercial driving assistance systems which use dedicated hardware, the driving assistance application based on smart phone has several advantages such as low cost, easier usability and upgradability. Several interesting work for traffic light recognition on smart phone platforms has been reported. For instance, the authors of [9] present a mobile vision system to detect pedestrian light in live video streams to help pedestrians with visual impairment cross roads. In [10], a real-time red traffic light recognition method is proposed on mobile platforms. The method consists of real-time traffic lights localization, circular regions detection and traffic lights recognition.

Due to the ego movement of the vehicle as well as the variety of outdoor conditions, accurate traffic light recognition is still faced with various challenges [5], [6], [9]:

1) Varying unknown environment.
2) The interference of other light sources, such as billboards, street lamps, etc.
3) The impact of different weather and illumination conditions.
4) The change of viewing angles and sizes of the traffic lights due to the ego motion of the vehicle.
5) Various appearances of traffic lights, e.g. with or without the countdown timer.
6) The existence of different types of traffic lights which indicate different meanings, such as the traffic light with a round lamp, the one with an arrow lamp, etc.
7) The functions of autofocus and automatic white balance of on-board cameras or smart phones which may result in color cast or blur.
8) The requirement of the real-time processing behavior of the traffic light recognition algorithm.

In order to solve the above problems, we present a traffic light recognition system on smart phone platforms. Different from [9], the smart phone is fixed on the front windshield of the ego vehicle with a bracket. The system recognizes the traffic light, including its phase (red or green) and type (round, straight arrow, etc.) information, and reminds the driver to follow the

Manuscript received xx, xxxx, revised xx, xxxx. This work was supported by National Natural Science Foundation of China (Grant No. 61273239) and the Fundamental Research Funds for the Central Universities of China (Grant No.120418001).

W. Liu, H. Yuan and H. Zhao are with the Research Academy, Northeastern University, Shenyang 110179, China.(e-mail: lwei@neusoft.com).

S. Li, J. Lv,B. Yu and T. Zhou are with the Advanced Automotive Electronics Technology Research Center, Neusoft Corporation, Shenyang 110179, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2515338, IEEE Transactions on Circuits and Systems for Video Technology

indications of traffic lights.

The system consists of three stages: candidate region extraction, recognition and spatial-temporal analysis. In the stage of candidate region extraction, an ellipsoid geometry threshold model in HSL color space is built to extract interesting color regions, which can resolve the incorrect segmentation problem in the existing linear color threshold method and avoid the problem of color cast to a certain extent. Meanwhile, these regions are further screened with a post-processing step and the candidate regions which simultaneously satisfy both color and brightness conditions are obtained. In the stage of recognition, a new nonlinear kernel function is proposed to effectively combine two heterogeneous features (HOG and LBP), and a Kernel Extreme Learning Machine (K-ELM) is designed to verify if a candidate region is a traffic light or not, and simultaneously recognize the phase and type of traffic lights. In the stage of spatial-temporal analysis, a multi-frame recognition framework based on finite state machine is introduced to further increase the reliability of recognition over a period of time.

Besides, this system has been implemented on a smart phone platform. For real time performances, some additional work is also done, including a quick lookup table based color candidate region extraction, and a CPU-GPU based acceleration of the K-ELM execution.

The remainder of this paper is organized as follows. The related work is presented in Section II. The system framework is presented in Section III. The details of the proposed system are described: candidate region extraction (Section IV), recognition in single images (Section V), spatial-temporal analysis (Section VI) and the system implementation on smart phone platforms (Section VII). In Section VIII, experimental results are provided, in comparison with the state-of-the-art methods. Finally, the conclusions and future work are made in Section IX.

II. Related work

Traffic lights are very different across the world. A typical traffic light consists of three lamps arranged vertically. The colors of the lamps are red, yellow, and green from top to bottom, and they are either round or arrow-shaped (see Fig.1). The color of the active lamp represents the phase of the traffic light, i.e. red, green and indicates the passable condition of the corresponding lane. In recent years, there have been traffic lights with count-down timers. Of these traffic lights, the yellow lamp in the middle is replaced with a count-down timer indicating the time remaining. Currently most of the existing traffic lights recognition approaches are composed of two main processes: generation and verification of the traffic light candidate regions.

A. Generation of Traffic Light Candidate Regions

According to the CIE standard, each color of traffic lights is usually defined in a specific area in CIE chromaticity diagram. Color feature is therefore widely used to detect traffic lights. There are some common color spaces, such as RGB [5], [6], [9], [13-15], HSI [16], [17], HSV [3], [18-20]. The RGB color space is not robust against the change of illuminations. Comparing with RGB color space, HSI and HSV color spaces are insensitive to luminance fluctuation, and similar to the color perception of human. Thus many researchers use these color spaces as a measure for distinguishing predefined colors of traffic lights [3], [16]. Besides the color spaces mentioned above, other color spaces such as YCbCr [21] and CIELab [22] are also used to extract traffic light candidate regions.

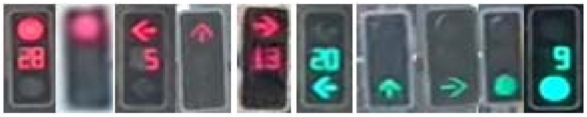


Fig.1. Vertical type traffic lights.

Although the color spaces used are different, most of the existing methods determine whether a pixel is an interesting color pixel or not by linear color threshold method [1], [3], [4], [14], [15]. That is to say, given the predefined range thresholds $T_{\min}$ and $T_{\max}$ in a certain color channel, a pixel is an interesting color pixel if its corresponding component value T_v satisfies $T_{\min} < T_v < T_{\max}$. The color threshold based segmentation method has a common trade-off problem: an increased false positive rate due to the wide color threshold range or a decreased true positive rate due to a narrow color threshold range [23]. Especially, this situation could get worse if color cast appears due to the white balance problem or the severe exposure of camera to external illumination. Consequently, a robust color segmentation method is required that considers various illumination conditions.

Color feature alone is inadequate to generate traffic light candidate regions because of the existence of some interference such as billboards and trees. In addition, complex scenes and cluttered backgrounds may cause many false positives. It is necessary to use additional features to distinguish traffic lights and overcome interference effects. Thus many researchers integrate some specific features to eliminate the interference regions, such as the geometrical feature, the brightness feature, etc. In [1] and [24] the authors use the geometrical features (aspect ratio, area, and pixel density) to obtain precise candidate regions of lamps. In [25], the color, brightness, and structural features are employed individually to obtain a set of traffic light candidate locations. The authors of [26] propose to eliminate some interference regions by detecting the circular edge of lamp candidate region.

In addition, several interesting works have been reported for detection on the candidate regions. For instance, in [27], the use of inter-component difference information for effective color edge detection is proposed. In [28], a novel framework for saliency detection, which first models the background with deep learning architecture and then separates salient objects from the background, is proposed.

In recent years, in order to shorten the computation cost and reduce the risk of getting incorrect candidate regions, some techniques adopt additional sensors such as the GPS and pre-existing maps containing traffic light locations. For example, in [8] the authors propose a method to predict the positions of traffic light with a prior map. The predicted positions are then projected into the image frame using a camera model, and serve as traffic light candidate regions. To improve the recognition accuracy, in [29], an on-board GPS sensor is employed to identify the traffic light candidate regions.

B. Verification of Traffic Light Candidate Regions

Although the simple geometric verification can remove some non-traffic light candidate regions, some interference regions which are similar to traffic lights still exist, for example the car tail light. Therefore, it needs a further verification to check the candidate regions. The most common verification methods are template matching and machine learning. The former method uses some templates of traffic lights, which are predefined by a priori knowledge, to verify the candidate regions, as in [2] and [4]. The advantage of this method is its simplicity, while the disadvantages lie in its strict demands of the accuracy of the candidate regions' boundaries and its low robustness. Compared with the former method, the latter method has the advantages of high accuracy and robustness. For instance, in order to verify the candidate regions of traffic lights, Adaboost algorithm and Haar-like feature are adopted in [3]. In [23], the authors employ a support vector machine (SVM) with the HOG feature for the traffic light verification. In [29], a convolutional neural network is used to recognize the phase of the traffic lights, i.e. red or green under normal illumination conditions. In [30], the H component in HSI space is classified with BP-neural network to verify the traffic light candidate regions. In [31] the authors employ a support vector machine (SVM) with the local binary patterns (LBP) feature for traffic light verification. Recently, a new classification method in [32] based on neural networks and mid-level features shows promising results. Nevertheless, due to the complexity and the real-time capability, this method is still a challenge when applied on mobile platforms.

To resolve the issues of low accuracy and reliability in the single frame based traffic light recognition algorithm, some researchers expand the recognition method into video streaming (multi-frame). For example, the CAMSHIFT algorithm is applied to track the candidate regions of traffic lights in [3]. In [6], the authors use a motion estimation method to track the traffic light in the video streaming, and feedback the current phase of traffic light. In [24], a video sequence based decision scheme is proposed. It avoids the temporary inconsistency in the verification of candidate regions. In [6], the authors present an HMM to find the optimal state sequence associated with the given observation sequence, aiming to obtain the best performance in the determination of the traffic light phases. In [33], an Interacting Multiple Model filter is used to track the traffic light through the time and to increase traffic light recognition performances. These methods described above improve both the reliability and the precision of traffic light recognition.

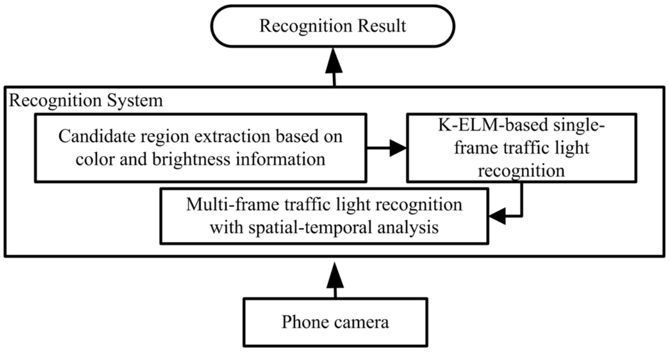

Recognition Result
Recognition System
Candidate region extraction based on color and brightness information
K-ELM-based single-frame traffic light recognition
Multi-frame traffic light recognition with spatial-temporal analysis
Phone camera

Fig.2. The overview of the proposed traffic light recognition system.

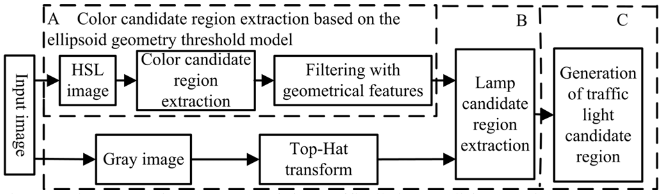

A Color candidate region extraction based on the ellipsoid geometry threshold model
B
C
Input image
HSL image
Color candidate region extraction
Filtering with geometrical features
Lamp candidate region extraction
Generation of traffic light candidate region
Gray image
Top-Hat transform

Fig.3. The extraction process of the traffic light candidate region.

III. TRAFFIC LIGHT RECGONITION SYSTEM

A traffic light recognition system based on smart phone platforms is proposed to recognize the vertical traffic lights in urban environment. We are only interested in recognizing the red and green traffic light, since the yellow traffic light represents a transient phase between the green light and the red light, and it serves as a warning signal. It is even absent for the traffic light with countdown timer. Fig.2 describes the proposed recognition system.

The recognition system in this paper includes three stages: candidate region extraction based on color and brightness information, K-ELM-based single-frame traffic light recognition, and multi-frame traffic light recognition with spatial-temporal analysis. The details of the proposed system are presented in the following sections respectively.

IV. EXTRACTION OF TRAFFIC LIGHT CANDIDATE REGIONS

Since the traffic light is brighter than most of background and has special color, the recognition system proposed in this paper combines a color segmentation with a bright region extraction algorithm to generate the traffic light candidate regions. The system has three stages: the extraction of color candidate region, the extraction of lamp candidate region, and the generation of the traffic light candidate region. The extraction process is shown in Fig.3.

A. Color Candidate Region Extraction Based on Ellipsoid Geometry Threshold Model

In this work, the HSL color space is adopted to extract the traffic light candidate regions. The space is better matched to visual perception, with less correlated color channel [34]. In [24] the authors have pointed out that the "color appearances" of the traffic lights are concentrated around several predetermined specific colors so that they can be well described by Gaussian distributions. Therefore, similar to [24], we model the color features of the traffic lights as 1D Gaussian distributions.

Firstly, we model the hue, saturation and lightness according to 1D Gaussian distributions. The value of the color channel k at each pixel defined as C_k , $k=1,2,3$, $C_k \sim N(\mu_k,\sigma_k{}^2)$. μ_k and $\sigma_k{}^2$ are the mean and variance of color channel k .

Then, the interesting pixels of red and green traffic light candidate regions are generated by the following equations:

$$b_r = \begin{cases} 1, if\left(H_r \in \left(H_{r1}^l, H_{r1}^h\right) \cup S_r \in \left(S_{r1}^l, S_{r1}^h\right) \cup L_r \in \left(L_{r1}^l, L_{r1}^h\right)\right) \\ or\left(H_r \in \left(H_{r2}^l, H_{r2}^h\right) \cup S_r \in \left(S_{r2}^l, S_{r2}^h\right) \cup L_r \in \left(L_{r2}^l, L_{r2}^h\right)\right) \\ 0, else \end{cases} \quad (1)$$

$$b_g = \begin{cases} 1, if\left(H_g \in \left(H_g^l, H_g^h\right) \cup S_g \in \left(S_g^l, S_g^h\right) \cup L_g \in \left(L_g^l, L_g^h\right)\right) \\ 0, else \end{cases} \quad (2)$$

Here, the value range in color channel k is $(\mu_k - \lambda \cdot \sigma_k, \mu_k + \lambda \cdot \sigma_k)$, and $\lambda = 3$.

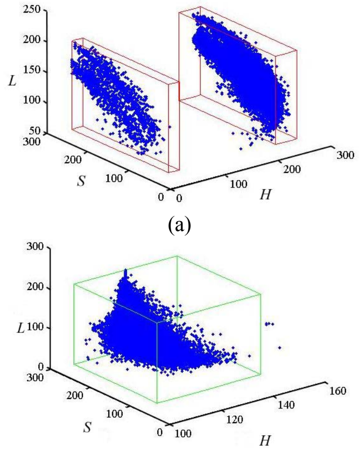
In order to learn parameters μ_k and σ_k , the training images with red, green traffic lights are collected respectively and the traffic lights regions are labeled manually. With all the pixels in the manually labeled traffic light regions, the parameters μ_k and σ_k can be estimated. As the samples are collected from different weather and illumination conditions, the parameters can adapt to different environments. With the ranges determined above, three cubes (two for red and one for green) can be determined, and the centers of the cubes denotes $(H_{ri}, S_{ri}, L_{ri}), i=1,2$ and (H_{g0}, S_{g0}, L_{g0}) respectively. In Fig.4, the statistical distributions of the pixels in manually labeled traffic lights regions are shown for (a) the red pixels and (b) the green pixels.

From the Fig.4 results, one can see that the pixels are gathering in the compact regions around the centers of the cubes, instead of filling them. It is also clear, from Fig.4, that the colors of the pixels in the corners of the cubes are very different from the colors of those at the center. Since a cube without corners resemble an ellipsoid, an ellipsoid geometry threshold model is proposed in this paper. The model of red pixels (H_r, S_r, L_r) and green pixels (H_g, S_g, L_g) are expressed as (3) and (4) respectively:

$$\frac{(H_r - H_{ri})^2}{h_{ri}^2} + \frac{(S_r - S_{ri})^2}{s_{ri}^2} + \frac{(L_r - L_{ri})^2}{l_{ri}^2} \le 1, i=1,2 \quad (3)$$

$$\frac{\left(H_g - H_{g0}\right)^2}{h_g^2} + \frac{\left(S_g - S_{g0}\right)^2}{s_g^2} + \frac{\left(L_g - L_{g0}\right)^2}{l_g^2} \le 1 \quad (4)$$

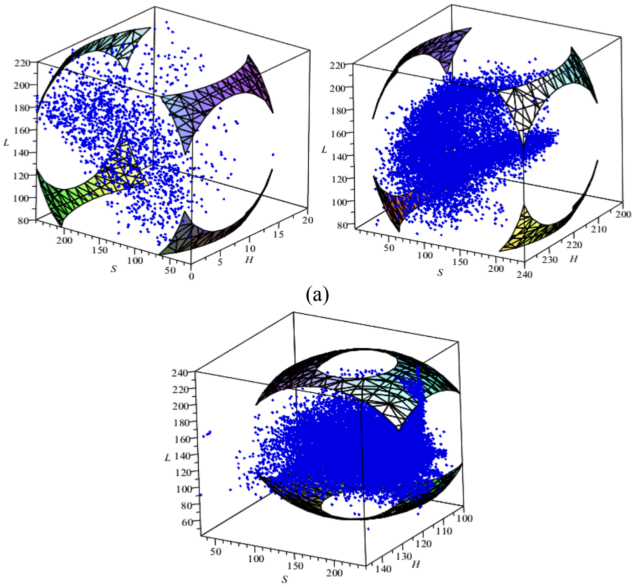
where $H_r \in \left(H_{ri}^l, H_{ri}^h\right)$, $S_r \in \left(S_{ri}^l, S_{ri}^h\right)$, $L_r \in \left(L_{ri}^l, L_{ri}^h\right)$, $H_g \in \left(H_g^l, H_g^h\right)$, $S_g \in \left(S_g^l, S_g^h\right)$ and $L_g \in \left(L_g^l, L_g^h\right)$. The centers of the ellipsoids are coincident with the centers of the cubes expressed as (3) and (4). The other parameters of the ellipsoids can be calculated as follows:



(a)

(b)

Fig.4. Statistical distributions of the pixels in manually labeled traffic light regions. (a) Shows the statistical distribution of red pixels, (b) shows the statistical distribution of green pixels.



(a)

(b)

Fig.5. Ellipsoid geometry threshold models. (a) Ellipsoid geometry threshold model of red color. (b) Ellipsoid geometry threshold model of green color.

$$\begin{cases} (2h_{ri})^2 = 4\left(H_{ri}^h - H_{ri}^l\right)^2 \\ (2s_{ri})^2 = \left(S_{ri}^h - S_{ri}^l\right)^2 + \left(L_{ri}^h - L_{ri}^l\right)^2 \\ (2l_{ri})^2 = \left(S_{ri}^h - S_{ri}^l\right)^2 + \left(H_{ri}^h - H_{ri}^l\right)^2 \end{cases} \quad (5)$$

$$\begin{cases} \left(2h_g\right)^2 = 4\left(H_g^h - H_g^l\right)^2 \\ \left(2s_g\right)^2 = \left(S_g^h - S_g^l\right)^2 + \left(L_g^h - L_g^l\right)^2 \\ \left(2l_g\right)^2 = \left(S_g^h - S_g^l\right)^2 + \left(H_g^h - H_g^l\right)^2 \end{cases} \quad (6)$$

Fig.5 shows the visualization of the ellipsoid geometry threshold models and the statistical distributions of the pixels in manually labeled traffic light regions. Compared with the cubes built by the traditional linear color threshold method, the proposed ellipsoid geometry model can eliminate large amount of the color pixels which are uninteresting to our algorithm. Also, more than 99% of the interesting color pixels from the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2515338, IEEE Transactions on Circuits and Systems for Video Technology

training images are contained within the ellipsoids defined by the above parameters. More test images with traffic lights have also been collected and the statistic of these images reveals a similar conclusion.

Compared to the traditional methods, the proposed ellipsoid geometry threshold model has the following advantages:

1) It resolves the incorrect segmentation problem in the traditional methods and improves the accuracy of the candidate color pixel extraction;
2) It avoids the problem of traffic light color cast to a certain extent;
3) It saves the processing time for the subsequent color candidate region extraction and verification process, as it filters out some uninteresting color regions.

After extracting the interesting color pixels, we conduct an 8 connected-component labeling to generate lamp candidate regions. Similar to [1], several geometrical features of each candidate region are computed: the pixel density, the aspect ratio and the area. By using these geometrical features some color regions that unlikely belong to lamp candidate regions are eliminated. As shown in Fig. 7(d), the vehicle's right tail light is eliminated as it doesn't satisfy the aspect ratio constraint.

B. Extraction of Lamp Candidate Regions

Due to the influence of complex background, weather, illumination conditions and other light sources, some interference regions still exist in the obtained color candidate regions. Considering that an active lamp is brighter than the surrounding local region, this characteristic can be used as a post-processing step to extract the lamp candidate region while removing the interference regions. The extraction process is as follows.

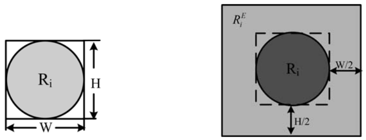
Firstly, an expanded region R_i^E is built for the color candidate regions $R_i, i = 1, 2, ..., N$. As shown in Fig.6, the height and width of the minimum enclosing rectangle of the color candidate region are denoted as H and W respectively. The region between the boundaries of R_i^E and R_i is denoted as R_i'.

Then, the original color image of the expanded region R_i^E is converted to a gray-scale image. The top-hat transform is applied to eliminate the influence of uneven illumination. Here, a square structuring element whose width is 11 pixels is chosen for the top-hat filter.

Finally, a color candidate region R_i is labeled as a lamp candidate region if the region satisfies the following conditions:

$$\begin{cases} N_i < M \\ \sigma_{R_i} > \sigma_{R_i'} \end{cases} \tag{7}$$

where $N_i = N'_{R_i^E} - N_{R_i}$, N_{R_i} represent the numbers of pixels in the region R_i, $N'_{R_i^E}$ represents the number of pixels in the expanded region R_i^E whose color is the same as the ones in the region R_i. M is a threshold and $M = WH/4$. σ_{R_i} and $\sigma_{R_i'}$ represent the average gray values of region R_i and region R_i' respectively. They are calculated as follows:



(a) (b)

Fig.6. Sketch image of regions. (a)Color candidate region, (b) Expanded region.

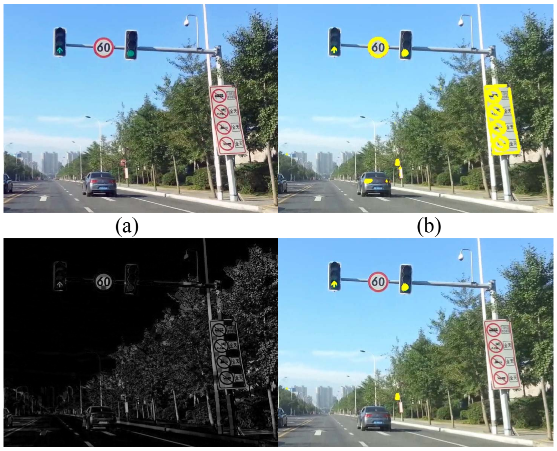

(a) (b) (c) (d)

Fig.7 Results of extraction of lamp candidate regions. (a) Original color image, (b) color candidate regions (marked in yellow), (c) image after top-hat transform, and (d) lamp candidate regions (marked in yellow).

$$\sigma_{R_i} = \frac{\sum_{(x,y)\in R_i} f(x,y)}{N_{R_i}} \tag{8}$$

$$\sigma_{R_i'} = \frac{\sum_{(x,y)\in R_i^E} f(x,y) - \sum_{(x,y)\in R_i} f(x,y)}{N_{R_i^E} - N_{R_i}} \tag{9}$$

where $f(x,y)$ represents the gray value of pixel (x,y), $N_{R_i^E}$ represents the total number of pixels in the expanded region R_i^E.

Fig.7 shows some results of the lamp candidate regions. In Fig. 7(d), it can be seen that some regions which don't satisfy the brightness condition are removed.

It is noticeable that for the purpose of illustration, in Fig.7(c) the top-hat transform result of the whole image is provided. However, in the real-world application, considering the computational performance, the top-hat transform is applied only to the expanded region R_i^E.

C. Generation of Traffic Light Candidate Regions

As we know, a typical traffic light consists of three lamps arranged vertically with equal sizes and the order of the three lamps is fixed as red, yellow(or count-down timer), and green from top to bottom, as shown in Fig.1. Considering a backboard is often around a traffic light in most traffic scenes, this structural information can also be used to select real red and green traffic lights from the candidates. Here, according to the relationship (the relative positions and size ratios) between active lamps and the backboard, we can generate the traffic light candidate regions.

It should be noted that, for the traffic light with count-down timer, since the active lamp and the count-down timer have the same color and are extremely close to each other, these two regions might be probably extracted as only one single region. This region would be easily treated as one single lamp, and it will lead to the generation of a wrong traffic light candidate region. To solve this problem, we generate the traffic light candidate region R_i^{TL} according to the aspect ratio A_i ($A_i = H/W$) of the obtained lamp candidate region, as shown in (10) and (11).

$$\begin{cases} X_L = \max(1, x_l - K) \\ X_R = \min(c, x_r + K) & \textit{if } R_i \textit{ is a red lamp} \\ Y_T = \max(1, y_t - K) & \textit{candidate region} \\ Y_B = \min(r, y_t + 7K) \end{cases} \tag{10}$$

$$\begin{cases} X_L = \max(1, x_l - K) \\ X_R = \min(c, x_r + K) & \textit{if } R_i \textit{ is a green lamp} \\ Y_T = \max(1, y_b - 7K) & \textit{candidate region} \\ Y_B = \min(r, y_b + K) \end{cases} \tag{11}$$

where (X_L, Y_T) and (X_R, Y_B) respectively denote the left-top and right-bottom vertices of the traffic light candidate region R_i^{TL}, (x_l, y_t) and (x_r, y_b) respectively represent the left-top and right-bottom vertices of the minimum enclosing rectangle of the lamp candidate region R_i, and r, c represent the height and width of the whole image respectively. Here K can be determined as follows:

$$\begin{cases} K = W / 2, & \textit{if } A_i \geq 1.5 \\ K = (W + H) / 4, & \textit{else} \end{cases} \tag{12}$$

V. RECOGNITION OF THE TRAFFIC LIGHT IN A SINGLE IMAGE

After the above procedures of traffic light candidate region extraction, there also exists the influence of interfering light sources, such as car tail lights. In order to verify whether a candidate region is a traffic light or a background and simultaneously recognize the type of the traffic light (round, straight arrow, left-turn arrow, right-turn arrow etc.), in this section, a new nonlinear kernel function is proposed to effectively combine two heterogeneous features, HOG and LBP, which is used to describe the traffic lights. Also, a Kernel Extreme Learning Machine (K-ELM) is designed to recognize the candidate region.

A. Feature Extraction of Traffic Light Candidate Regions

HOG and LBP are two heterogeneous features with complementary information. The combination of the two features, can extract contour and texture information simultaneously and has obtained effective results in the applications such as pedestrian detection, face recognition, etc. [35]. Thus, we use HOG-LBP feature to describe the traffic light candidate region. In [36], HOG and LBP are directly concatenated to form a feature vector, while the contributions of each feature are not considered, and the descriptive ability of the features is not fully exploited. Inspired by [37], a new nonlinear kernel function is proposed to combine the two heterogeneous features:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{(1-\beta)\left\|x_i^{HOG} - x_j^{HOG}\right\|^2 + \beta\left\|x_i^{LBP} - x_j^{LBP}\right\|^2}{\gamma}) \tag{13}$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ represents the proposed kernel function; $\mathbf{x}_i$ is the feature vector of sample i; $\mathbf{x}_i = \left[x_i^{HOG}, x_i^{LBP}\right]$; x_i^{HOG}, x_i^{LBP} represent the feature vectors of HOG and LBP respectively. β is a combination coefficient, which determines the contribution of each feature, and $\beta \in [0,1]$. By (13), the HOG feature and the LBP can be combined with different β .It is worth noting that the method of combinative HOG-LBP features in [35] is only a special case of the proposed method which is equal to $\beta = 0.5$. More details can be seen from the experimental results in section VIII.

In this paper, a traffic light candidate region is firstly converted into gray-scale, and is then scaled to a size of 20×40 pixels which is used to extract the features of HOG and LBP. For HOG feature, the block size is 10, the cell size is 5, and orientation bin number is 9; for LBP feature, we extract 58-dimensional uniform patterns and 1-dimensional non-uniform pattern per block, and the feature vectors of all the blocks are concatenated as the LBP feature of the candidate region. For each traffic light candidate region, the dimension of the feature vector is 1995. In order to reduce the computation burden, the BW method [38] is adopted to reduce feature dimensions. The strategy of BW is to select the features with large between-category distances and small within-category distances. Considering the algorithmic acceleration based on OpenCL (to be described in section VII.B), the dimension of the feature vector is reduced to 256, and then it is input to the K-ELM with the proposed kernel function.

B. Recognition of Traffic Light Candidate Regions

ELM is a machine learning method with fast training speed and suitable for multi-category classification task [39]. Many research results show that ELM produces comparable or better classification accuracies with implementation complexity compared to artificial neural networks and support vector machines [40]. Furthermore, it has been pointed out that K-ELM achieves good generalization performance, meanwhile there is no randomness in assigning connection weights between input and hidden layer and the number of hidden nodes does not need to be given [41-42]. Therefore we select the K-ELM to verify whether a candidate region is a traffic light or the background and recognize the phase and type of the traffic light. The output function of K-ELM can be written compactly as:

$$\begin{aligned} f(x) &= h(x)H^T(\frac{I}{\lambda} + HH^T)^{-1}T \\ &= \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T (\frac{I}{\lambda} + \Omega_{ELM})^{-1}T \end{aligned} \tag{14}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2515338, IEEE Transactions on Circuits and Systems for Video Technology

where $f(x)$ is the output of K-ELM; N is the number of training samples; $x_i(i=1,2,\cdots,N)$ expresses the feature vector of training samples; x is the feature vector of a traffic light candidate region, i.e. the input to the K-ELM classifier; Ω_{ELM} is the kernel matrix for the classifier; $\Omega_{ELM}=HH^T:\Omega_{ELM\,ij}=h(x_i)h(x_j)=K(x_i,x_j)$; $K(x_i,x_j)$ represents the proposed kernel function in (13); $T=[t_1,t_2,\cdots,t_i,\cdots,t_N]^T$ is the target vector; $t_i=[t_{i1},t_{i2},\cdots,t_{im}]$ is the output vector of the i th training sample, and λ is the regularization coefficient.

In this paper, two K-ELM classifiers are trained according to the color information of the candidate regions: one for the recognition of green traffic lights, the other for the recognition of red traffic lights. For each color in the traffic lights, the output of K-ELM is designed as 6 classes, i.e. $m=6$, which correspond to non-traffic lights and five different types of traffic lights that are round, straight arrow, left-turn arrow, right-turn arrow and unknown type. The unknown type indicates the traffic light whose type cannot be determined. Once a candidate region is verified by the corresponding K-ELM classifier, it is regarded as a traffic light. The phase of this traffic light is determined to be the color of the candidate region, and its type is recognized by the K-ELM's output vector via the maximum operation.

VI. RECOGNITION OF THE TRAFFIC LIGHT BASED ON SPATIAL-TEMPORAL ANALYSIS

A. Traffic Light Phase Recognition Using a Finite State Machine

The recognition of the traffic light in single images was described in sections IV and V. The traffic lights in the image were extracted and recognized, and the phase and type of recognized traffic light can also be given. However, some false positive recognition results might exist. In order to increase the reliability of recognition over a period of time, in this section, the traffic light recognition is extended from single images to multi-frame images due to the following reasons.

1) The number of phases of the traffic light is limited (red or green), and each phase will last for a certain period of time. In fact, there exists an optional yellow or count-down timer phase, but in our system we ignore this phase.
2) For each traffic light, it can only be at one particular phase at one time, namely either red or green light is switched on.

The above rules can be applied to improve the recognition performance of traffic light in a single frame. In this section, we introduce an information queue S_i, which allows a verification by multi-frame spatial-temporal analysis. The information queue S_i is used to record the recognition results of the recent Q_{size} times of the ith recognized traffic light, the recognition results consist of phase, type and location. The Q_{size} denotes the size of the information queue. The maintenance process of the information queue is shown in Fig.8.

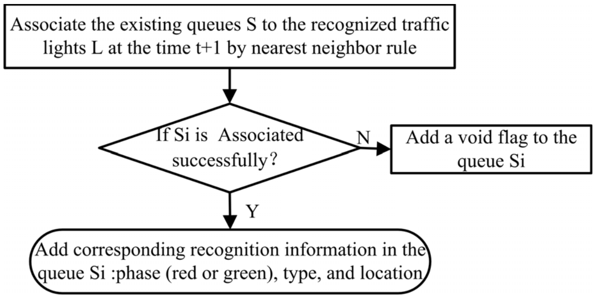

Associate the existing queues S to the recognized traffic lights L at the time t+1 by nearest neighbor rule
If Si is Associated successfully?
N
Add a void flag to the queue Si
Y
Add corresponding recognition information in the queue Si :phase (red or green), type, and location

Fig.8. The maintenance process of the information queue.

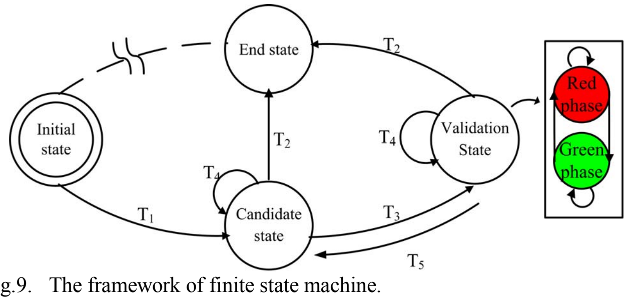


Fig.9. The framework of finite state machine.

First, the existing information queues $S=\{S_1,S_2,\cdots,S_i,\cdots S_K\}$ and the recognized traffic lights $L=\{L_1,L_2,\cdots,L_j,\cdots L_N\}$ at time $t+1$ are associated by the nearest neighbor rule. Then, according to the association results, the information queues will be updated: If S_i is associated to L_j, a new piece of information corresponding to L_j is added; otherwise, a void flag is pushed.

Here, in order to perform the association, the traffic lights represented by S need to be tracked. Then the updated tracked locations are associated to L by the nearest neighbor rule. In this paper, the LK algorithm [43] is adopted for tracking the traffic lights. All tracking points are chosen at the positions where obvious features could be extracted, such as the corners of the traffic light backboard, etc. It should be noted that, for each of the unassociated recognized traffic lights, a new queue is established.

After establishing the information queues, a spatial-temporal analysis framework based on a finite state machine is introduced to enhance the reliability of the recognition of traffic lights. For each recognized traffic light, four states exist in its life cycle: $\{\text{Initial state,Candidate state ,Validation state,End state}\}$. The finite state machine is able to describe clearly the transitions between these states (shown in Fig.9) and the required conditions of these transitions. In this paper, only the traffic lights at validation state have phase recognition results. Thus some occasional single-frame false positive recognition results can be reduced.

This finite state machine can be interpreted as follows:

For a new recognized light L_j which has not been associated with any existing queues, a new queue is established and its state information is initialized. At this moment, the light L_j is at the initial state. This state is a temporary state. Once the initialization is complete, it transits to the candidate state. This process corresponds to the state transition process T_1 in Fig.10.

For a traffic light L_j at candidate state or validation state, it will enter the end state when there are N_V consecutive void flags in the queue, which means the traffic light is not successfully associated continuously. The queue will be deleted afterwards. This process corresponds to T_2.

For a traffic light L_j at candidate state, it will turn into the validation state if the validation condition is met. This process corresponds to T_3 . Otherwise, it will maintain the current candidate state, which corresponds to T_4 . The validation condition is as follow: the number of the most frequent appearing phase N_{s} in the recent Q_{size} times in the information queue should not be less than the preset threshold $Q_{\min}$.

For a traffic light L_j at validation state, the output phase after the multi-frame spatial-temporal analysis is:

$$phase = \begin{cases} Green & if\ N_s = N_g \\ Red & if\ N_s = N_r \\ Unknown & \text{otherwise} \end{cases} \tag{15}$$

where, $N_s = \max(N_r, N_g)$, N_r, N_g represent the number of phases as red and green in the recent Q_{size} times respectively.

When the information queue of one validated traffic light no longer meets the validation condition, its state turns from the validation state to the candidate state. This process corresponds to T_5.

B. Type Recognition of the Traffic Light

After the recognition of the phase of the traffic light, a simple voting approach is adopted to determine the type of the traffic light.

$$\begin{gathered} Type = \begin{cases} k^*, & if\ \sum_{t=1}^{T} L_t^{k^*} > Q_{\min} \\ \text{Unknown}, & otherwise \end{cases} \\ k^* = \arg\max_k \sum_{t=1}^{T} L_t^k \\ L_t^k = \begin{cases} 1, & if\ C_t = k \\ 0, & otherwise \end{cases} \end{gathered} \tag{16}$$

where $k \in \{1,2,3,4,5\}$ represents the type of the traffic light to be round, straight arrow, left-turn arrow, right-turn arrow and unknown type respectively. C_t represents the type of the traffic light at time t.

VII. SYSTEM IMPLEMENTATION ON SMART PHONE PLATFORMS

In our research, this recognition system is implemented on a Samsung Note3 smart phone. The Note3 is equipped with a quad-core Krait 400-architecture CPU at up to 2.3GHz per core, an Adreno 330 GPU with a frequency of 450MHz and 3G RAM.

With limited computing resources on smart phone platforms, more efficient solutions need to be explored to achieve real time performances.

A. Quick Extraction on Color Candidate Regions

As the image sequence captured by Samsung Note3 Smartphone is in the YUV (YUV4:2:0) color space, it needs to be converted into HSL space. Then one can apply the proposed ellipsoid geometry threshold model to judge if a pixel is an interesting color pixels or not. However, the process of color space conversion and judgment will cause a sharp increase of computation cost. To reduce computation load on the device, we combine both color space conversion and interesting color pixel judgment process in a lookup table (LUT). The storage structure of the LUT is $C[Y][U][V] = C_V$, where $C_V \in \{0,1,2\}$ represents the pixel is green, red and uninteresting color pixel respectively. The size of the LUT is $256 \times 256 \times 256$ .Therefore, by simply looking up the conversion table, a given pixel can be judged quickly if it is an interesting color pixel or not.

B. Acceleration of K-ELM Algorithm Using OpenCL

From formula (13) it can be seen that the output $f(x)$ of K-ELM consists of two parts: one part is $A_{N\times m} = (\frac{I}{\lambda} + \Omega_{ELM})^{-1} T$, which is only related to the training samples and can be calculated off-line, thus it does not consume any online computation resource. The other part is $B_{1\times N} = [K(x, x_1), \cdots, K(x, x_N)]$, which is related to the feature vector of the candidate region and the feature vectors of the training samples. This part needs to be calculated online with the proposed kernel function in (12), thus it is time-consuming. Considering that the computation of each dimension ofB_{1xN} is independent, it is suitable for parallel optimization. In order to reduce the computation time, a CPU-GPU fusion based approach is adopted to accelerate the proposed K-ELM algorithm using OpenCL. For the recognition of a given candidate region, its acceleration process is shown in Fig.10.

First, the calculation and dimension-reduction of the candidate region's feature vector x is performed on CPU. To facilitate the computation of GPU, considering the suggestion of [37] and the number of the GPU's processing elements, the dimension of the feature vector is reduced to 256.

Then, the needed data for calculating $f(x)$ is copied from the CPU memory to the global memory of GPU. The data includes the feature vector x , the off-line calculated feature vectors $X_{256\times N} = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}$ of training samples and $A_{N\times m}$. Here, $X_{256\times N}$ and $A_{N\times m}$ are treated as constant matrix and copied only once at the initialization. Next, $f(x)$ is calculated on GPU. To calculate $B_{1\times N} = [K(x, x_1), \cdots, K(x, x_N)]$, the global execution space of GPU is divided into n work-groups,

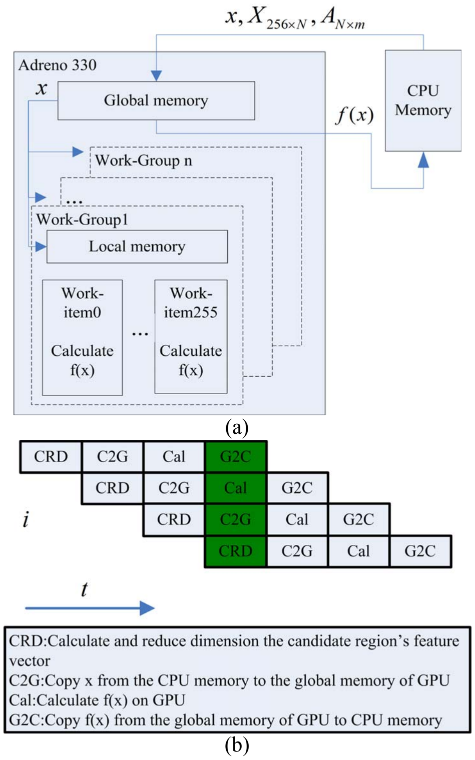

$x, X_{256\times N}, A_{N\times m}$
Adreno 330
x
Global memory
$f(x)$
CPU Memory
Work-Group n
...
Work-Group1
Local memory
Work-item0
Calculate f(x)
...
Work-item255
Calculate f(x)
(a)
CRD C2G Cal G2C
CRD C2G Cal G2C
CRD C2G Cal G2C
CRD C2G Cal G2C
i
t
CRD:Calculate and reduce dimension the candidate region's feature vector
C2G:Copy x from the CPU memory to the global memory of GPU
Cal:Calculate f(x) on GPU
G2C:Copy f(x) from the global memory of GPU to CPU memory
(b)

Fig.10. The acceleration process of the K-ELM.(a) describes the proposed acceleration approach for one candidate region.(b) shows the parallel process of multiple candidate regions.

$n=\left\lceil \frac{N}{M} \right\rceil$. Here, $\lceil\ \rceil$ indicates the ceiling operation, M indicates the number of work-items in each work-group, and totally $M\times n$ work-items are generated. In this paper, $M=256$, which corresponds to the dimension of the feature vector. For each work-item, one $K(x,x_i)$ operation is executed. In the process of calculating matrix B, as the feature vector x of the candidate region is used by all the work-items, it is copied from global memory to the shared local memory of each work-group. Also, all the work-items are programmed to perform coalesced access to the global memory so that the memory access is accelerated.

Finally, the output $f(x)$ is copied to the CPU memory.

Within the framework of OpenCL, a pipe-line scheme is also designed to parallel process multiple candidate regions, as shown in Fig.10 (b). It can be seen that the calculation and copy of the feature vectors of these candidate regions are pipe-lined, as well as the calculation and copy of the corresponding outputs. These operations will hide the time used for copying data between the memories of CPU and GPU.

The proposed acceleration approach has been implemented and evaluated on Samsung Note 3 smart phone. From the test results, it can be seen that the proposed acceleration approach achieves 0.75 ms per candidate region, which is 5 times faster than the only CPU version of implementation. This meets the requirement of the system's real-time behavior.

TABLE I
GROUND TRUTH STATISTICS OF THE TEST VIDEOS

	#	Ground Truth
Red lights	Total number	11543
	Number of Round light	4474
	Number of turn-left	3255
	Number of right-turn	1378
	Number of straight	1542
	Number of unknown	894
Green lights	Total number	10960
	Number of Round light	4186
	Number of turn-left	2797
	Number of right-turn	1246
	Number of straight	1833
	Number of unknown	898

VIII. EXPRIMENTAL RESULTS

In this section, we first analyze the contribution of the nonlinear kernel function that is used to combine two heterogeneous features. And then we compare quantitatively the proposed K-ELM recognition method with state-of-the-art methods. Furthermore, we verify the effectiveness of the proposed system via the experimental results of spatial-temporal analysis. Finally, the runtime performance of the system is evaluated as well.

A. Experimental Data

In order to analyze the contribution of the heterogeneous feature combination method and validate the performance of K-ELM, two datasets are built up for the training and testing purposes. For the training dataset, two subsets are collected, one for each color of the traffic lights. In each subset, we collect 5000samples of traffic lights and 5000samples of non-traffic lights. Of each subset, 50% of the samples serve for training and 50% are used as a validation dataset to determine the optimal parameters of K-ELM (see section VIII.B).

Similar to the training dataset, a test dataset is collected. The test dataset also has two subsets, one for each color of the traffic lights. In each subset, we collect 4000 samples of traffic lights and 4000 samples of non-traffic lights. It is noticeable that the test samples are collected from two sources: the images captured with the vehicle-mounted Samsung Note 3 phone and the images from the Internet.

To quantitatively evaluate the effectiveness of the proposed system, 150 test videos have been collected from different illumination conditions (morning, daytime and twilight) and different weather conditions (sunny, cloudy, rainy) by Samsung Note 3 phone installed on the test vehicle. Each video contains at least 100 frames of images with a resolution of 1280×720. The collected videos contain several types of traffic lights, as shown in Fig.1, including 1) round lights with a count-down timer, 2) round lights without a count-down timer, and 3) various types of arrow lights (left-turn, right-turn, straight).

A ground truth label was made manually, storing the phases of all visible traffic lights. Furthermore, the types of traffic lights which can be visually distinguished by human are also given. For those traffic lights whose types are visually indistinguishable, they are labeled as unknown types. In TABLE I, the statistics of the videos are presented.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2515338, IEEE Transactions on Circuits and Systems for Video Technology

B. Contribution Analysis of the Nonlinear Kernel Function

In the proposed heterogeneous feature combination method, the coefficient β acts as an important role and directly affects the description power of the feature combination. To demonstrate the contribution of parameter β, with the training dataset introduced in Section VIII.A, a family of K-ELM classifiers with various β values is trained and evaluated on the validation dataset. Considering both recall and precision, $F-measure$ is adopted as the evaluation criteria.

$$F = 2*\frac{RE*PR}{RE+PR} \qquad (17)$$

where, $RE = \frac{TP}{TP+FN}$, $PR = \frac{TP}{TP+FP}$.

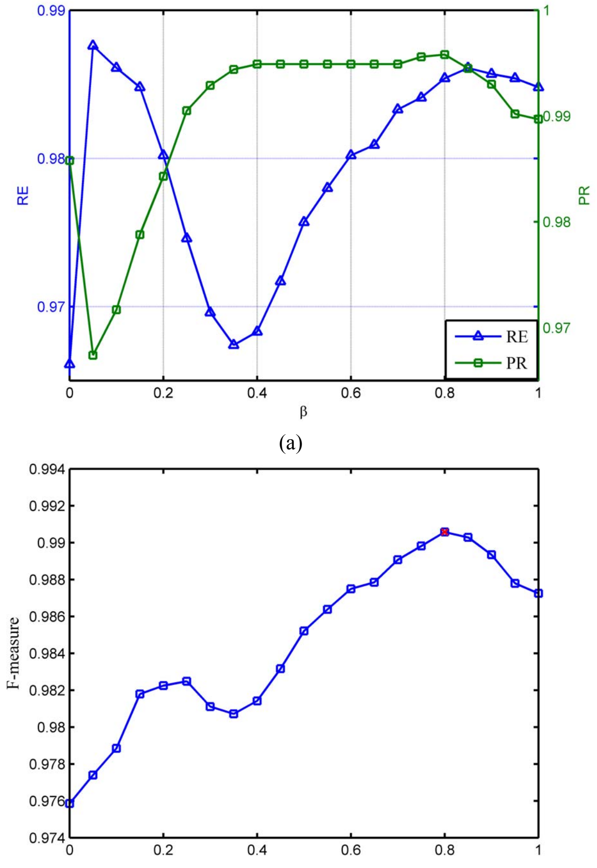
Here, RE and PR represent recall and precision respectively. TP is the total number of correctly recognized traffic light samples, FN is the total number of missed traffic light samples, $(TP+FN)$ indicates the total number of traffic light samples in the ground truth. FP is the total number of misrecognized non-traffic light samples, and $(TP+FP)$ indicates the total number of recognized traffic lights.

It should be noted that the proposed K-ELM outputs both the phase and the type of the traffic lights simultaneously. Here, in order to conveniently demonstrate the contribution of the parameter β, only the recall and precision of the phase information are considered. This is done by treating the output of K-ELM as a binary classification output – traffic light and non-traffic light, regardless of the type information.

Besides the parameter β, there are two other parameters in K-ELM: the kernel parameter γ and the regularization parameter λ. The choice of the values of these two parameters can also affect the performance of the classifier. To analyze the contribution of β especially, we take the HOG - LBP features in [35] as the baseline of comparison, which corresponds to the case with $\beta = 0.5$. In this case, the optimal values of γ and λ are firstly determined by applying multiple experiments with a grid search strategy on validation dataset. Then, the variation of F-measure with respect to the various values of β is analyzed using the determined optimal values of γ and λ. In this paper, the search range is defined as $\{2^{-10}, 2^{-9} \ldots, 2^{4}\}$ for γ, $\{2^{-5}, 2^{-4}, \ldots, 2^{10}\}$ for λ. The optimal values of parameter γ and λ on the validation dataset are determined and $\gamma = 1, \lambda = 16$.

Then, the variation of F-measure with respect to the various

values of β (from 0 to 1) is analyzed using the determined optimal values of γ and λ.The analysis results on the validation dataset of red lights are shown in Fig.11. From the results, it can be seen that different values of β have different effects on the red light recognition, when using the optimal values of parameters γ and λ. In Fig.11(b), there exist several feature combinations with different β values whose descriptive power is superior to that of $\beta = 0.5$. This exhibits the contribution of parameter β.The feature combination with $\beta = 0.8$ has the highest score of F.Therefore, for red lights, $\beta = 0.8$ is chosen as the combination coefficient in this paper. Similarly, the optimal combination coefficient β can also be obtained for green lights.

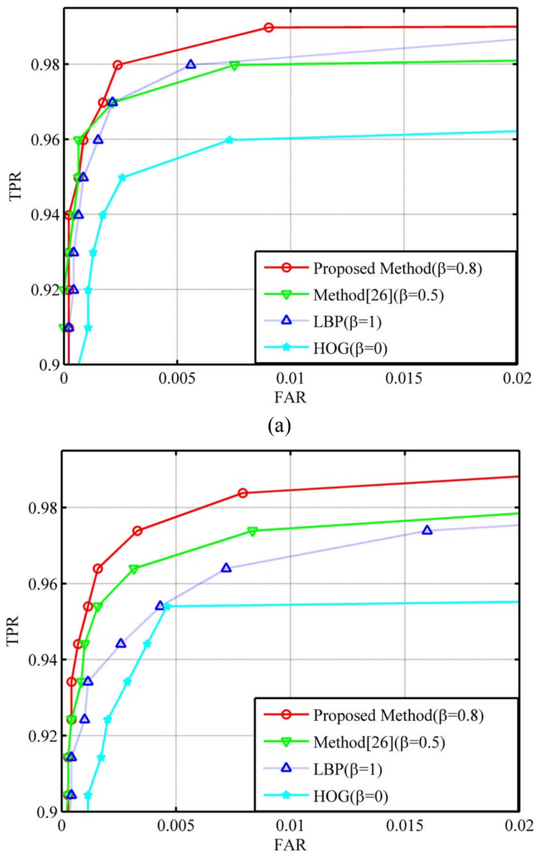


(a)

(b)

Fig.11. Evaluation results of red traffic light recognition with various β values. (a) Shows the curves of RE and PR with different β values. (b) Shows the curve of $F-measure$ with various β values, and the red square marker indicates the point with the maximal $F-measure$ value .

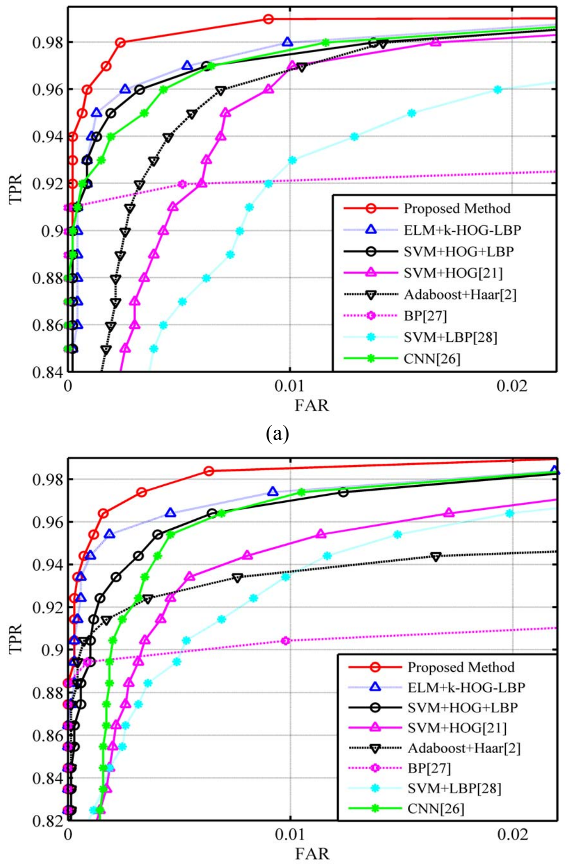
Furthermore, in order to further validate the effectiveness of the proposed heterogeneous feature combination method, four different feature combinations are tested and compared on the test dataset, with the ROC curves shown in Fig.12.The horizontal axis shows the False Positive Rate (FPR), and the vertical axis shows the True Positive Rate (TPR). Four selected parameters $\beta = 0$, $\beta = 1$, $\beta = 0.5$ and $\beta = 0.8$ respectively represent the HOG, LBP, HOG+LBP [35]and the proposed HOG-LBP combination feature. From these results, one can see that the proposed feature combination method outperforms the single feature and the feature combination method of [36]. This shows the effectiveness of the heterogeneous features with the proposed combination method.



(a)

(b)

Fig.12.Comparisons of different feature combinations. (a) Shows the results on test dataset of red light recognition. (b) Shows the results on test dataset of green light recognition.



(a)

(b)

Fig.13. ROC curves of different recognition methods (a) The ROC curve of red light recognition, (b) The ROC curve of green light recognition.

C. Quantitative Comparison between K-ELM and Other Methods

A traffic light contains both phase and type information. Firstly, we evaluate the contribution of the proposed K-ELM on the phase recognition performance. The proposed K-ELM is compared with other state-of-the-art methods: (i) Adaboost + Haar-like [2]; (ii) BP network [30]; (iii) SVM + LBP [31]; (iv) SVM + HOG [23]; (v) CNN [29]. Here, in comparison, the result of SVM + HOG + LBP is also given. To perform a quantitative evaluation, we test different methods on test datasets. The ROC curves are shown in Fig.13.

Here, the evaluation result of another method is also provided: an ELM classifier with the proposed heterogeneous feature combination, labeled "ELM + k-HOG-LBP". To ensure a fairer comparison, for all the methods, the same training dataset and test dataset (as mentioned above) are used, and their optimal parameters are also determined on the validation dataset. From the results, one can see that the proposed K-ELM method performs much better than the state-of-the-art methods.

It has been mentioned previously that the proposed K-ELM outputs both the phase and the type of the traffic light simultaneously, while in the state-of-the-art methods only the phase recognition result is provided. These methods perform a binary classification by outputting the traffic light as positive and non-traffic light as negative. For the sake of comparison, only the output of the phase information is considered. This is done by treating the outputs of all types of traffic lights as positive and the outputs of non-traffic light as negative.

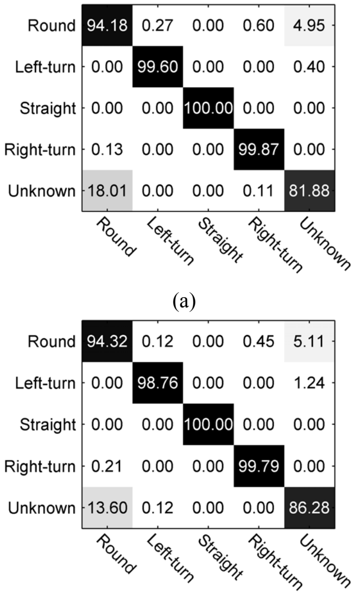
Furthermore, we evaluate the contribution of the proposed K-ELM on type recognition performance. To our best knowledge, this work is the first prototype of traffic light type recognition in the literature. In order to compare, several other methods are also implemented by us, including:(i) linear SVM classifier; (ii) BP network with multiclass output;(iii) ELM with multiclass output;(iv)nearest neighbor(NN). In the SVM case, we have followed the one-versus-one vote scheme for multiclass classification. To perform a fair comparison, for all the methods, the same heterogeneous feature combination is adopted, and two classifiers are trained per method to respectively recognize the types of red and green lights.

TABLEII shows the type recognition rates obtained on the test datasets. As can be seen, the proposed K-ELM-based type recognition method outperforms all the competing methods, providing type recognition rates equal to 92.59% and 93.81% for the green light dataset and the red light dataset. The corresponding confusion matrixes are shown in Fig.14 (a) and Fig.14 (b) respectively. From the results, one can see that the recognition accuracy of arrow lights (straight, left-turn, and right-turn) is higher than that of round lights. This is because the arrow lights' clear edges and pointing directions highly benefit the recognition. As for the false recognition rate, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2515338, IEEE Transactions on Circuits and Systems for Video Technology

TABLE II
TYPE RECOGNITION RATES OF DIFFERENT METHODS

	Green light	Red light
SVM	89.72%	90.95%
BP	89.33%	90.11%
NN	86.19%	87.21%
CNN	89.25%	90.45%
ELM	91.33%	92.42%
K-ELM	92.59%	93.81%



	Round	Left-turn	Straight	Right-turn	Unknown
Round	94.18	0.27	0.00	0.60	4.95
Left-turn	0.00	99.60	0.00	0.00	0.40
Straight	0.00	0.00	100.00	0.00	0.00
Right-turn	0.13	0.00	0.00	99.87	0.00
Unknown	18.01	0.00	0.00	0.11	81.88

(a)

	Round	Left-turn	Straight	Right-turn	Unknown
Round	94.32	0.12	0.00	0.45	5.11
Left-turn	0.00	98.76	0.00	0.00	1.24
Straight	0.00	0.00	100.00	0.00	0.00
Right-turn	0.21	0.00	0.00	99.79	0.00
Unknown	13.60	0.12	0.00	0.00	86.28

(b)

Fig. 14. Confusion matrix. (a) Confusion matrix for type estimation of red lights. (b)Confusion matrix for type estimation of green lights.

round lights are easily confused with left-turn, right-turn and unknown types of lights. This is due to the fact that with such image resolution, it is very difficult to distinguish the types of the traffic lights at a far distance.

D. Spatial-Temporal Analysis

We also quantitatively evaluate the effectiveness of the proposed system via the experiments results of spatial-temporal analysis on test videos. In this paper, the parameters of the information queue are set as $Q_{size} = 7$ and $Q_{\min} = 4$, due to the reason that the output delay is expected to be as small as possible while a reliable output is guaranteed. As the average processing speed is 20frames per second (to be described in section VII.E) with $Q_{\min} = 4$, the output delay is limited within 0.5 seconds, which is smaller than the common reaction time of drivers. So the influence of the output delay can be neglected. If Q_{size} is set too large, too many outdated light recognition results will be stored in the queue. This not only influences the recognition performance, but also consumes more memory. Therefore, in this paper, $Q_{size} = 7$.

TABLE III shows the phase recognition results of traffic lights after the spatial-temporal analysis. Here, in order to demonstrate the improvement by the proposed spatial-temporal analysis, the single-frame recognition results are also given for a comparison. In TABLE III, TPR represents the true positive rate, also known as recognition rate or Recall. FPR represents the false positive rate, and gives the percentage of misrecognized non-traffic lights in the total number of traffic lights recognized by the system. In the case of FPR measurement, the smaller the value is, the better the accuracy is. As can be seen from the results, the accuracy of the phase recognition of traffic lights is effectively improved with the proposed spatial-temporal analysis.

TABLE III
RECOGNITION RESULTS OF TRAFFIC LIGHT PHASE

Phase	TP	FP	FN	TPR	FPR
RL-SF	11366	83	177	98.47%	0.64%
RL-STA	11434	2	109	99.06%	0.017%
GL-SF	10737	91	223	97.97%	0.66%
GL-STA	10807	3	153	98.60%	0.028%

RL-SF: Red traffic light(single frame), RL-STA: Red traffic light (spatial-temporal analysis), GL-SF: Green traffic light(single frame), GL-STA: Green traffic light (spatial-temporal analysis)

TABLE IV
RECOGNITION RESULTS OF TRAFFIC LIGHT TYPE

Type	TP	FP	FN	TPR	FPR
RL-SF	10300	110	239	96.72%	1.1%
RL-STA	10355	23	271	97.23%	0.22%
GL-SF	9677	123	262	96.17%	1.3%
GL-STA	9736	38	288	96.76%	0.39%

RL-SF: Red traffic light (single frame), RL-STA: Red traffic light (spatial-temporal analysis), GL-SF: Green traffic light (single frame), GL-STA: Green traffic light (spatial-temporal analysis)

TABLE IV shows the type recognition results of traffic lights after the spatial-temporal analysis. The single-frame recognition results are also given for a comparison. Here, the statistics of the unknown type traffic lights are not included. As can be seen from the results, the accuracy of the type recognition of traffic lights is markedly improved with the proposed spatial-temporal analysis.

Fig.15 shows some recognition results of typical traffic scenes. To provide a clearer view of the results, only the parts of images around the traffic lights are shown in the figure. It can be seen that the proposed system can recognize traffic lights of different types correctly.

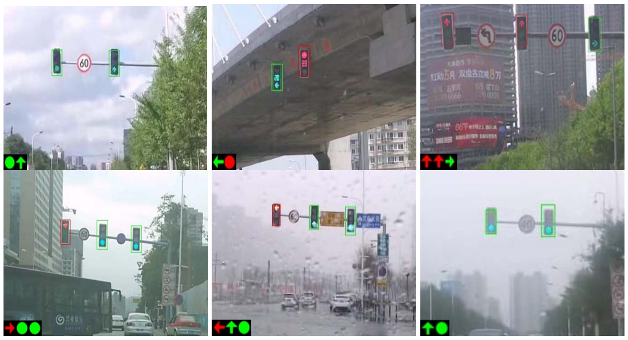


Fig.15. Typical test results of the traffic light recognition system on a mobile platform.

E. Runtime Performance Evaluation

We evaluate the processing time of the proposed system on the Samsung Note3 platform. The processing time is given in TABLE V. From TABLE V, one can see that the processing

TABLE V
PROCESSING TIME OF THE PROPOSED SYSTEM

Process	Time (ms)
Extraction of Color candidate region	11.97
Extraction of traffic light candidate region	5.28
Traffic light recognition(with OpenCL/without OpenCL)	20.44/63.2
Spatial-temporal analysis	9.21
Total(with OpenCL/without OpenCL)	46.90/89.66

time of the proposed system with OpenCL acceleration is about 47ms/f, which means our system can achieve about 20 fps.

IX. CONCLUSION AND FUTURE WORK

In this paper, a traffic light recognition system based on smart phone platforms is proposed and several contributions have been made.

1) To avoid the influences of the camera color cast, the complex background, weather, illumination conditions, an ellipsoid geometry threshold model in HSL color space is built to extract interesting color regions. Meanwhile, a post-processing step is applied to obtain candidate regions of traffic lights.
2) A new kernel function is proposed to effectively combine two heterogeneous features (HOG and LBP), and a Kernel Extreme Learning Machine (K-ELM) is designed to verify if a candidate region is a traffic light or not.
3) To further increase the reliability of recognition over a period of time, a spatial-temporal analysis framework based on finite state machine is introduced to recognize the phase and type of traffic lights.
4) A CPU-GPU fusion based approach is adopted to accelerate the execution of the proposed K-ELM so that a computational performance which is 5 times faster than the only CPU version of implementation can be achieved.

The test results of real scenes show that the proposed system can simultaneously accurately recognize the phase and type of traffic lights compared to the existing methods. Besides, the response of the system is rapid and a feedback can be given in less than a second. It's also worth pointing out that the recognition of traffic lights (especially the recognition of arrow lights) is not useful unless the results are associated with the lane information. There may be multiple lights in the cross road which has many lanes in the same direction. In such scenario, each traffic light indicates the traffic situation of its corresponding lane. Therefore, the recognition results of traffic lights must be associated with the lanes. In the future, we will try to fuse the recognition results with GPS navigation information. By considering both the trajectory planning and the current location information, the recognition results will be reasonably interpreted and utilized.

REFERENCES

[1] C. Yu, C. Huang, and Y. Lang, “Traffic light detection during day and night conditions by a camera,” in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, 2010, pp. 821–824.

[2] R. De Charette and F. Nashashibi, “Traffic light recognition using image processing compared to learning processes,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 333–338.

[3] J. Gong, Y. Jiang, G. Xiong, C. Guan, G. Tao, and H. Chen, “The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 431–435.

[4] R. De Charette and F. Nashashibi, “Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates,” in *Intelligent Vehicles Symposium, 2009 IEEE*, 2009, pp. 358–363.

[5] M. Omachi and S. Omachi, “Traffic light detection with color and edge information,” in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, 2009, pp. 284–287.

[6] A. E. Gomez, F. Alencar, P. V. Prado, F. S. Osorio, D. F. Wolf, and others, “Traffic lights detection and state estimation using Hidden Markov Models,” in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, 2014, pp. 750–755.

[7] J. Baber, J. Kolodko, T. Noel, M. Parent, and L. Vlacic, “Cooperative autonomous driving: intelligent vehicles sharing city roads,” *Robot. Autom. Mag. IEEE*, vol. 12, no. 1, pp. 44–49, 2005.

[8] N. Fairfield and C. Urmson, “Traffic light mapping and detection,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 5421–5426.

[9] J. Roters, X. Jiang, and K. Rothaus, “Recognition of traffic lights in live video streams on mobile devices,” *Circuits Syst. Video Technol. IEEE Trans. On*, vol. 21, no. 10, pp. 1497–1511, 2011.

[10] Y.-T. Chiu, D.-Y. Chen, and J.-W. Hsieh, “Real-time traffic light detection on resource-limited mobile platform,” in *Consumer Electronics-Taiwan (ICCE-TW), 2014 IEEE International Conference on*, 2014, pp. 211–212.

[11] A. Acharya, J. Lee, and A. Chen, “Real Time Car Detection and Tracking in Mobile Devices,” in *Connected Vehicles and Expo (ICCVE), 2012 International Conference on*, 2012, pp. 239–240.

[12] C.-W. Tang, K.-T. Feng, P.-H. Tseng, C.-H. Chen, and J.-W. Guo, “A pitch-aided lane tracking algorithm for driver assistance system with insufficient observations,” in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 2012, pp. 3261–3266.

[13] Y. K. Kim, K. W. Kim, and X. Yang, “Real time traffic light recognition system for color vision deficiencies,” in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, 2007, pp. 76–81.

[14] M. Omachi and S. Omachi, “Detection of traffic light using structural information,” in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, 2010, pp. 809–812.

[15] M. Diaz-Cabrera, P. Cerri, and J. Sanchez-Medina, “Suspended traffic lights detection and distance estimation using color features,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012, pp. 1315–1320.

[16] H. Tae-Hyun, J. In-Hak, and C. Seong-Ik, “Detection of traffic lights for vision-based car navigation system,” in *Advances in Image and Video Technology*, Springer, 2006, pp. 682–691.

[17] Y. Jie, C. Xiaomin, G. Pengfei, and X. Zhonglong, “A new traffic light detection and recognition algorithm for electronic travel aid,” in *Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on*, 2013, pp. 644–648.

[18] J. Choi, B. T. Ahn, and I. S. Kweon, “Crosswalk and traffic light detection via integral framework,” in *Frontiers of Computer Vision,(FCV), 2013 19th Korea-Japan Joint Workshop on*, 2013, pp. 309–312.

[19] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, “Traffic Light Mapping, Localization, and State Detection for Autonomous Vehicles,” in *International Conference on Robotics and Automation*, 2011.

[20] Z. Li-tian, F. Meng-Yin, Y. Yi, and W. Mei-ling, “A framework of traffic lights detection, tracking and recognition based on motion models,” in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 2298–2303.

[21] H.-K. Kim, J. H. Park, and H.-Y. Jung, “Effective traffic lights recognition method for real time driving assistance system in the daytime,” *World Acad. Sci. Eng. Technol. 59th*, 2011.

[22] S. Sooksatra and T. Kondo, “Red traffic light detection using fast radial symmetry transform,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference on*, 2014, pp. 1–6.

[23] C. Jang, C. Kim, D. Kim, M. Lee, and M. Sunwoo, “Multiple exposure images based traffic light recognition,” in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, 2014, pp. 1313–1318.

[24] Y. Shen, U. Ozguner, K. Redmill, and J. Liu, “A robust video based traffic light detection algorithm for intelligent vehicles,” in *Intelligent Vehicles Symposium, 2009 IEEE*, 2009, pp. 521–526.

[25] Y. Zhang, J. Xue, G. Zhang, Y. Zhang, and N. Zheng, “A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles,” in *Control Conference (CCC), 2014 33rd Chinese*, 2014, pp. 4924–4929.

[26] F. Lindner, U. Kressel, and S. Kaelberer, “Robust recognition of traffic signals,” in *Intelligent Vehicles Symposium, 2004 IEEE*, 2004, pp. 49–53.

[27] J. Ren, J. Jiang, D. Wang, and S. S. Ipson, “Fusion of intensity and inter-component chromatic difference for effective and robust colour edge detection,” *IET Image Process.*, vol. 4, no. 4, pp. 294–301, 2010.

[28] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, “Background Prior-Based Salient Object Detection via Deep Reconstruction Residual,” *Circuits Syst. Video Technol. IEEE Trans. On*, vol. 25, no. 8, pp. 1309–1321, Aug. 2015.

[29] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita, “Traffic light recognition in varying illumination using deep learning and saliency map,” in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 2286–2291.

[30] W. Hong-jiang, R. Na, Z. Wen-Qiang, Z. Ting-ting, C. Lun-feng, Z. Rong-xue, and T. Feng, “Research on Unmanned Vehicle Traffic Signal Recognition Technology,” in *Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on*, 2010, vol. 2, pp. 298–301.

[31] C.-C. Chiang, M.-C. Ho, H.-S. Liao, A. Pratama, and W.-C. Syu, “Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts,” *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 12, pp. 6919–6934, 2011.

[32] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, “Effective and Efficient Midlevel Visual Elements-Oriented Land-Use Classification Using VHR Remote Sensing Images,” *Geosci. Remote Sens. IEEE Trans. On*, vol. 53, no. 8, pp. 4238–4249, 2015.

[33] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, “Tracking both pose and status of a traffic light via an Interacting Multiple Model filter,” in *Information Fusion (FUSION), 2014 17th International Conference on*, 2014, pp. 1–7.

[34] R. Pan, W. Gao, and J. Liu, “Color clustering analysis of yarn-dyed fabric in HSL color space,” in *Software Engineering, 2009. WCSE’09. WRI World Congress on*, 2009, vol. 2, pp. 273–278.

[35] W.-J. Park, D. Kim, S. Suryanto, C.-G. Lyuh, T. M. Roh, and S.-J. Ko, “Fast human detection using selective block-based HOG-LBP,” in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 601–604.

[36] X. Wang, T. X. Han, and S. Yan, “An HOG-LBP human detector with partial occlusion handling,” in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 32–39.

[37] Y.-X. Li, S. Ji, S. Kumar, J. Ye, and Z.-H. Zhou, “Drosophila gene expression pattern annotation through multi-instance multi-label learning,” *Comput. Biol. Bioinforma. IEEEACM Trans. On*, vol. 9, no. 1, pp. 98–112, 2012.

[38] Z.-L. Sun, H. Wang, W.-S. Lau, G. Seet, and D. Wang, “Application of BW-ELM model on traffic sign recognition,” *Neurocomputing*, vol. 128, pp. 153–159, 2014.

[39] G.-B. Huang, D. H. Wang, and Y. Lan, “Extreme learning machines: a survey,” *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 107–122, 2011.

[40] S. S. Baboo and S. Sasikala, “Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis,” in *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*, 2010, pp. 748–757.

[41] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, “A fast and accurate online sequential learning algorithm for feedforward networks,” *Neural Netw. IEEE Trans. On*, vol. 17, no. 6, pp. 1411–1423, 2006.

[42] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *Syst. Man Cybern. Part B Cybern. IEEE Trans. On*, vol. 42, no. 2, pp. 513–529, 2012.

[43] B. Peasley and S. Birchfield, “Replacing projective data association with Lucas-Kanade for KinectFusion,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 638–645.

Wei Liu received the M.S. and the Ph.D. degrees in Control Theory and Control Engineering from Northeastern University, China in 2001 and in 2005, respectively. He is currently a professor-level senior engineer with Research academy, Northeastern University, China. He is also the director of Intelligent Vision Laboratory of Neusoft Corporation, China. His research interests include computer vision, image processing and pattern recognition with applications to intelligent video surveillance and advanced driver assistance systems.

Shuang Li received the Master's degrees in Applied Mathematics from Northeastern University, China in 2014.

She is currently a software engineer with Neusoft Corporation, China. Her research interests include computer vision, image processing and pattern recognition.

Jin lv received the B.S. degree in electronic and information engineering from Shenyang University of Technology, China, in 2008 and the M.S. degree in pattern recognition and intelligent system from Northeastern University, China in 2010. She is a Research Engineer with the Advanced Automotive Electronics Technology Research Center, Neusoft Corporation, China. Her research interests include image processing, computer vision, and machine learning.

Bing Yu received the B.S. degree from Shanghai Jiao Tong University, China in 2010, and the M.S. degree in Automation from the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), France in 2012.He is currently a research engineer with the Advanced Automotive Electronics Technology Research Center in Neusoft Corporation, China. His research interests include image processing, computer vision and machine learning.

Ting Zhou received the B.S. degree in Automation from Northeastern University, China, in 2011 and the M.S. degree in Control Engineering from Northeastern University, China in 2013.

She is a Research Engineer with the Advanced Automotive Electronics Technology Research Center, Neusoft Corporation, China. Her research interests include machine learning, computer vision, and image semantic segmentation.

Huai Yuan received the B.S. degree in Computer Software from Nankai University, China in 1983, and the M.S. degree in Computer Software from Northeastern University, China in 1986.

He is currently an associate professor of Northeastern University, China. He is also the director of Advanced Automotive Electronics Technology Research Center in Neusoft Corporation, China. His research interests include computer vision, image processing and intelligent vehicles.

Hong Zhao received the M.S. degree and Ph.D. degrees in Computer Science from Northeastern University, China in 1984 and 1991, respectively.

Since 1994, he has been a professor with Northeastern University, China. He is currently the director of National Engineering Research Center for Digital Medical Imaging Device, China. His research interests include computer multimedia systems, distributed computer systems, image processing, and computer vision.

Footnotes:

W. Liu, H. Yuan and H. Zhao are with the Research Academy, Northeastern University, Shenyang 110179, China.(e-mail: lwei@neusoft.com).

S. Li, J. Lv, B. Yu and T. Zhou are with the Advanced Automotive Electronics Technology Research Center, Neusoft Corporation, Shenyang 110179, China.

Manuscript received xx, xxxx, revised xx, xxxx. This work was supported by National Natural Science Foundation of China (Grant No. 61273239) and the Fundamental Research Funds for the Central Universities of China (Grant No.120418001).