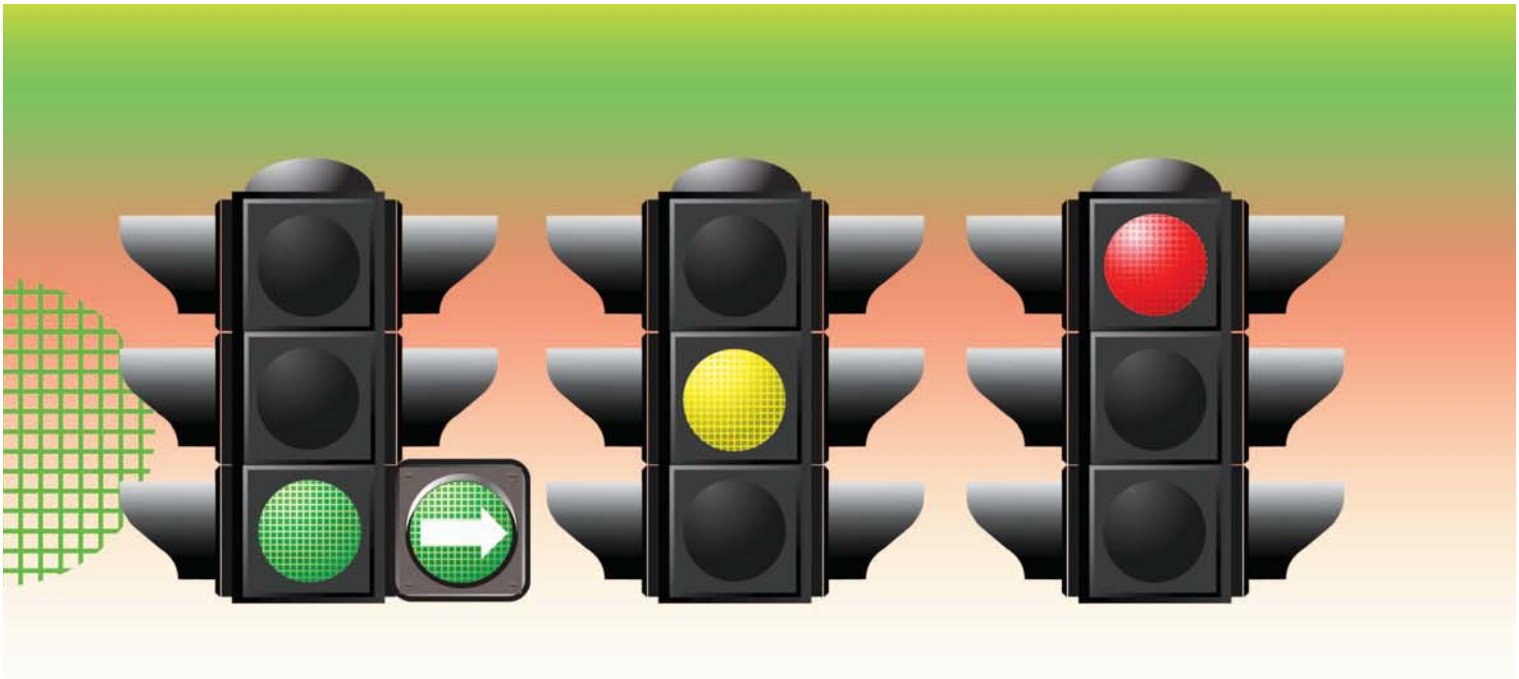


Accurate and Reliable Detection of Traffic Lights Using Multiclass Learning and Multiobject Tracking

Zhilu Chen and Xinming Huang
Senior Member, IEEE
The authors are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA
E-mail: xhuang@wpi.edu

***Abstract*—**Automatic detection of traffic lights has great importance to road safety. This paper presents a novel approach that combines computer vision and machine learning techniques for accurate detection and classification of different types of traffic lights, including green and red lights both in circular and arrow forms. Initially, color extraction and blob detection are employed to locate the candidates. Subsequently, a pretrained PCA network is used as a multiclass classifier to obtain frame-by-frame results. Furthermore, an online multiobject tracking technique is applied to overcome occasional misses and a forecasting method is used to filter out

Digital Object Identifier 10.1109/MITS.2016.2605381
Date of publication: 25 October 2016

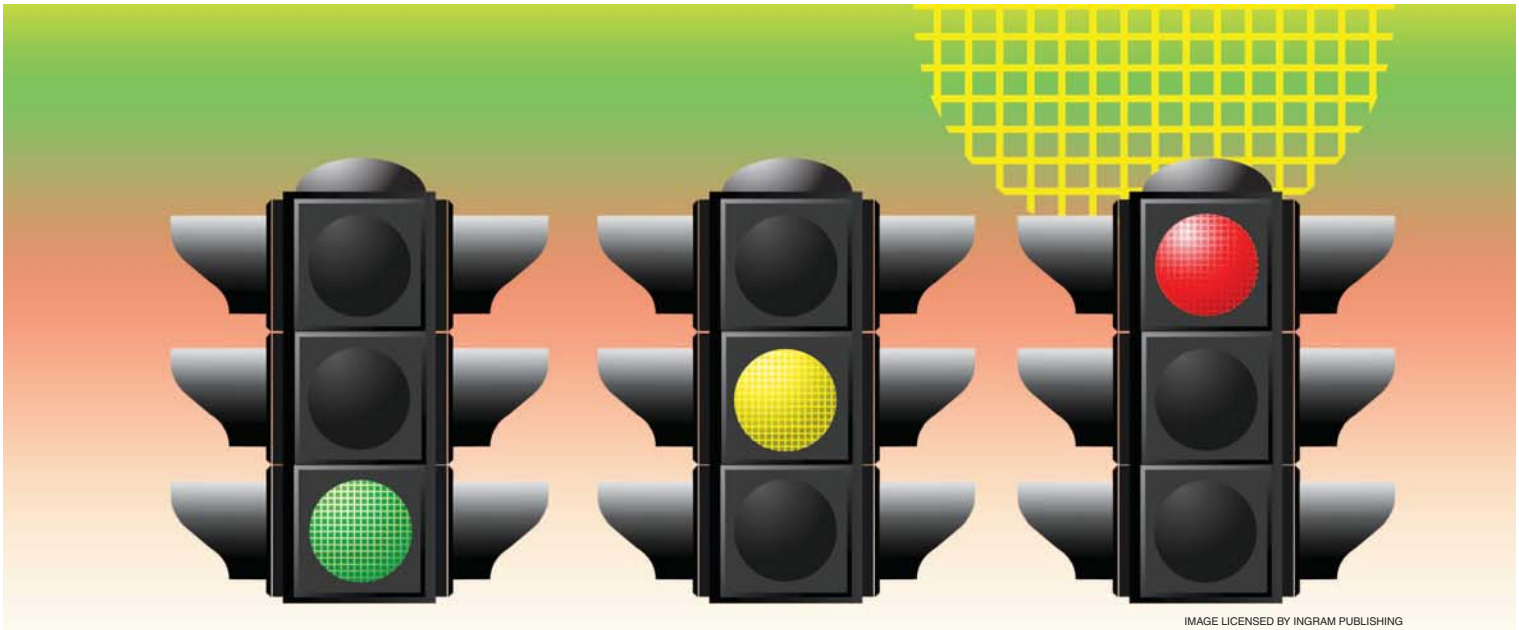


IMAGE LICENSED BY INGRAM PUBLISHING

false positives. Several additional optimization techniques are employed to improve the detector performance and handle the traffic light transitions. When evaluated using the test video sequences, the proposed system can successfully detect the traffic lights on the scene with high accuracy and stable results. Considering hardware acceleration, the proposed technique is ready to be integrated into advanced driver assistance systems or self-driving vehicles. We build our own data set of traffic lights from recorded driving videos, including circular lights and arrow lights in different directions. Our experimental data set is available at http://computing.wpi.edu/Data set.html.

I. Introduction

Automatic detection of traffic lights is an essential feature of an advanced driver assistance system or self-driving vehicle. Today it is a critically important road safety issue that many traffic accidents occurred at intersection are caused by drivers running red lights. Recent data from Insurance Institute of Highway Safety (IIHS) show that in the year of 2012 on US roads, red-light-running crashes caused about 133,000 injuries and 683 deaths [1]. Introduction of automatic traffic light detection system, especially red light detection, has important social and economic impacts.

In addition to detecting traffic lights, it is also important to recognize the lights as they appear in circular or as directional arrow lights. For example, a red left arrow light and a green circular light can appear at the same time. Without recognition, the detection systems can get confused because valuable information has been lost. There are few papers in the literature that combine detection and recognition of traffic lights together.

Based on our survey, there are very few data sets available for traffic lights. The Traffic Lights Recognition (TLR) public benchmarks [2] contain image sequences with traffic lights and ground truth. However, the images in the data set do not have high resolution, and the number of physical traffic lights is limited due to the fact that the image sequences are converted from a short video. In addition, this data set only contains circular traffic lights, which is not always the case in real applications. Therefore, we opt to build our own data set for traffic light detection, including circular lights and arrow lights in all three directions. Our data set of traffic lights can be used by many other researchers in computer vision and machine learning.

In this paper, we propose a new method that combines computer vision and machine learning techniques. Color extraction and blob detection are used to locate the candidates, followed by the PCA network (PCANet) [3] classifiers. The PCANet classifier consists of a PCANet and a linear Support Vector Machine (SVM). Our experimental results suggest the proposed method is highly effective for detecting both green and red traffic lights of many types.

Despite of the effectiveness of PCANet and many outstanding achievements made by computer vision researchers, object detection from an image still makes frequent errors, which may cause huge problems in the real-world critical applications such as Advanced Driver Assistance Systems (ADAS). Traditional frame-by-frame detection methods ignore the inter-frame information in the video. Since the objects in a video are normally in continuous motion, their identities and trajectories are valuable information that can improve the frame-based detection results. Unlike a pure tracking problem that tracks a marked object from the first frame, tracking-by-detection algorithms involves frame-by-frame detection, inter-frame tracking and data association. In addition, multiobject tracking (MOT) algorithms can be employed to distinguish different objects and keep track of their identities and trajectories. When it becomes a multiclass problem such as recognizing different types of traffic lights, additional procedure such as a voting scheme is often applied. In addition, the method needs to address the situation

that traffic light status can change suddenly during the detection process.

The rest of the paper is organized as follows. Section II provides a summary of related work. Section III describes our data collection and experimental setup. In Section IV, we propose a method that combines computer vision and machine learning techniques for traffic light detection using PCANet. In Section IV-C, we propose a MOT-based method that stabilizes the detection and improves the recognition results. Performance evaluations is presented in Section V, followed by some discussion in Section VI and conclusions in Section VII.

II. Related Work

There are several existing works on traffic light detection. Spot light detection [4], [5] is a method based on the fact that a traffic light is much brighter than the lamp holder usually in black color. A morphological top-hat operator is used to extract the bright areas from gray-scale images, followed by a number of filtering and validating steps. In [6], an interactive multiple-model filter is used in conjunction with the spot light detection. More information is used to improve its performance, such as status switching probability, estimated position and size. Fast radial symmetry transform is a fast variation of the circular Hough transform, which can be used to detect circular traffic lights as demonstrated in [7].

Several other methods also combine the vehicle GPS information. A geometry-based filtering method is proposed to detect traffic lights using mobile devices at low computational cost [8]. The GPS coordinates of all traffic lights are presumably available, and a camera projection model is used. Mapping traffic light locations is introduced in [9] by using tracking, back-projection and triangulation. Google also presented a mapping and detection method in [10] which is capable of recognizing different types of traffic lights. It predicts when traffic lights should become visible with the help of GPS data, followed by classifying possible candidates. Geometric constraints and temporal filtering are then applied during the detection. The inter-frame information is also helpful for detecting traffic lights. A method that uses Hidden Markov Model to improve the accuracy and stability of the results is demonstrated in [11]. The state transition probability of traffic lights is considered and information from several previous frames is used. Reference [12] introduces a traffic light detector based on template matching. The assumption is that the two off lamps in the traffic light holder are similar to each other and neither of them look similar with the surrounding background. In our earlier work [13], this method is extended by applying machine learning techniques and adding additional information during the process. However, only red lights are considered and its approach is different from that in this paper in many ways.

Deep learning [14], [15] is a class of machine learning algorithms that has many layers to extract hidden features. Unlike hand-crafted features such as Histograms of Oriented Gradients (HOG) features [16], it learns features from training data. PCANet is a simple, yet effective deep learning network proposed by [3]. Principal Component Analysis (PCA) is employed to learn the filter banks. It can be used to extract features of faces, hand written digits and object images. It has been tested on several data sets and delivers surprisingly good results [3]. Using PCANet in traffic light detection or other similar applications has not been researched thus far.

Integration of detection and tracking has been used in a few ADAS-related works. The trajectory of traffic light is used to validate the theoretical result in [6]. Kalman filter is employed to predict the traffic sign positions. It claims that tracking algorithm is able to improve the overall system reliability [17], [18]. Utilizing accumulated classifier decisions from a tracked speed limit sign, a majority voting scheme is proven to be very robust against accidental misclassifications [19].

Meanwhile, multiobject tracking (MOT) is aimed at tracking every individual object while maintaining their identities. MOT does not address the detection problem. Even if an object is mistakenly identified, MOT continues to track that object [6]. Multiobject tracking builds the trajectories of the objects and then associates the tracking results with the detection results. There are two categories of tracking-by-detection methods: batch methods and online methods. Batch methods [20], [21] usually requires the information of the whole sequence to be available before association, which is not applicable to real-time traffic light detection. Online methods [22], [23] do not have such requirements. They utilize the information accumulated up to the current frame and make the estimations, therefore online methods can be applied to real-time applications. It shares some similarities with time series analysis and forecasting [24], whose goal is to predict the future state or value based on the past and current observations. The comparison of our approach and related work is discussed in VI.

III. Data Collection and Experimental Setup

In this paper, we focus on the detection of red/green traffic lights and the recognition of their types. The amber lights can be detected using similar techniques, but we do not consider amber lights here due to lack of data. The recognition of arrow lights requires that the input frames must be high resolution images. Otherwise all lights are just colored dots or balls in the frame, and it is impossible to recognize them.

We mount a smartphone behind the front windshield and record videos when driving on the road. Several hours of videos are recorded around the city of Worcester, Massachusetts, USA, during both summer and winter seasons. Subsequently, we select a subset of video frames to build

the data set since most of the frames do not contain traffic lights. In addition, passing an intersection only takes a few seconds in case of the green lights. At red lights, the frames are almost identical as the vehicle is stopped. Thus the length of selected video for each intersection is very short. Several minutes of traffic-light-free frames are retained in our data set for assessment of false positives. Each image has a resolution of 1920 × 1080 pixels. To validate the proposed approach and to avoid overlapping of training and test data, the data collected in the summer is used for training and the data collected in the winter is used for testing. Our traffic light data set is made available online at http://computing.wpi.edu/Data set.html.

A. Training Data

All the training samples are taken from the data collected during the summer. Input data to the classifier are obtained from the candidate selection procedure described in A, and the classifier output goes to the tracking algorithm for further processing. Thus evaluation of the classifier is independent to the candidate selection or the post-processing (tracking). The classifier is trained to distinguish true and false traffic lights, and to recognize the types of the traffic lights. OpenCV [25] is used for SVM training, which chooses the optimal parameters by performing 10-fold cross-validation.

The positive samples, which contain the traffic lights, are manually labeled and extracted from the data set images. The negative samples, such as segments of trees and vehicle tail lights, are obtained by applying the candidate selection procedure over the traffic-light-free images. The green lights and red lights are classified separately. For green lights, there are three types base on their aspect ratios. The first type is called Green ROI-1, which contains one green light in each image and its aspect ratio is approximately 1:1. The second type is called Green ROI-3. It contains the traffic light holder area which has one green light and two off lights, and its aspect ratio is approximately 1:3. The third type is called Green ROI-4. It contains the traffic light holder area which has one green round light, one green arrow light, and two off lights, and its aspect ratio is approximately 1:4.

Each type of sample images has several classes. The Green ROI-1 and Green ROI-3 both has five classes including negative samples as shown in Fig. 1 and Fig. 2. These 5 classes from top to bottom are Green Negative (GN-1; GN-3), Green Arrow Left (GAL-1; GAL-3), Green Arrow Right (GAR-1; GAR-3), Green Arrow Forward (GAF-1; GAF-3) and Green Circular (GC-1; GC-3).

The Green ROI-4 also has five classes including negative samples as shown in Fig. 3. The five classes from top to bottom are Green Negative (GN-4), Green Circular and Green Arrow Left (GCGAL-4), Green Circular and Green Arrow Right (GCGAR-4), Green Arrow Forward and Left

FIG 1 Examples of 5 classes of Green ROI-1.

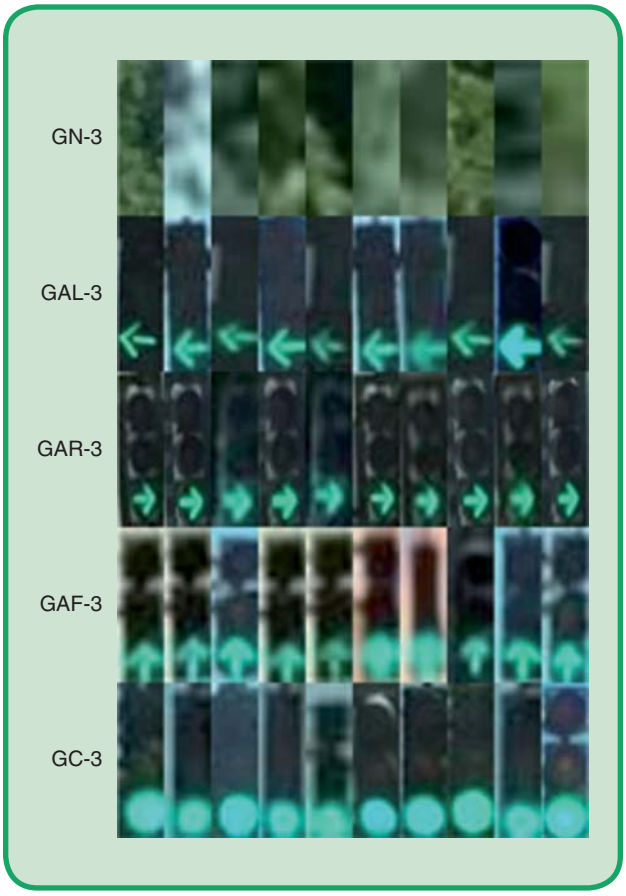


FIG 2 Examples of 5 classes of Green ROI-3.

(GAFL-4) and Green Arrow Forward and Right (GAFR-4). The Green Negative samples are obtained from traffic-lights-free videos by using the color extraction method discussed in Section IV-A.

For red lights, there are two types of sample images base on their aspect ratios. The first type is called Red ROI-1 as shown in Fig. 4. It contain one red light in each image and its aspect ratio is approximately 1:1. The other type is called Red ROI-3 as shown in Fig. 5. It contains the traffic light holder which contains one red light and two off lights, and its aspect ratio is approximately 1:3. Each type of sample

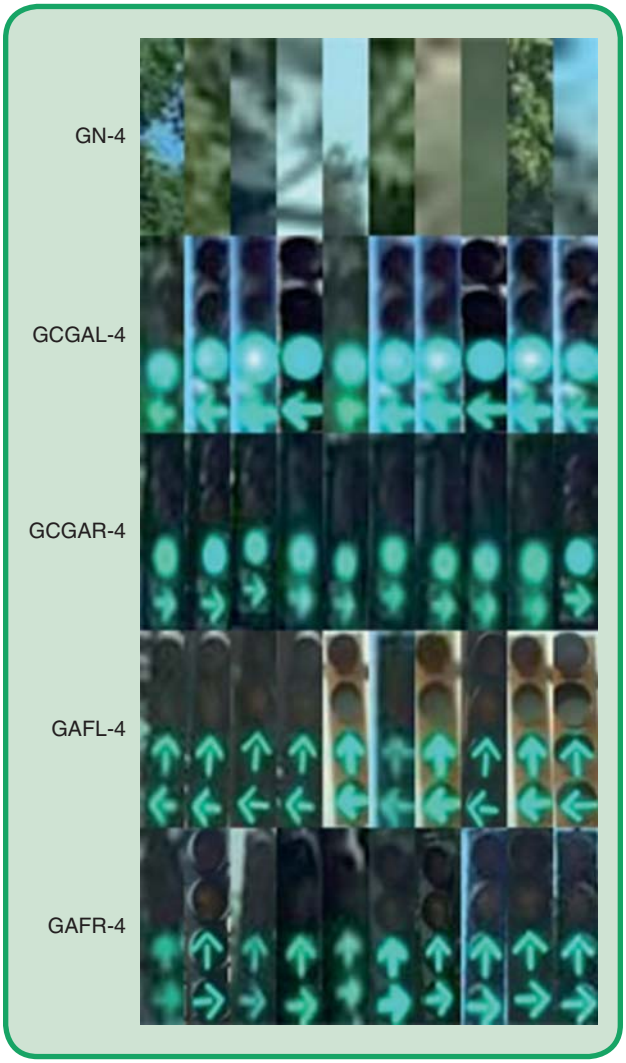


FIG 3 Examples of 5 classes of Green ROI-4.

FIG 4 Examples of 3 classes of Red ROI-1.

images has three classes: Red Negative (RN-1; RN-3), Red Arrow Left (RAL-1; RAL-3) and Red Circular (RC-1; RC-3). The Red Negative samples are obtained from traffic-lights-free videos by using the color extraction method mentioned in IV-A. The red light do not have ROI-4 data because the red light is on top followed by an amber light and one or two green lights at the bottom. If the red light is on, the amber and green lights beneath must be off. These three lights

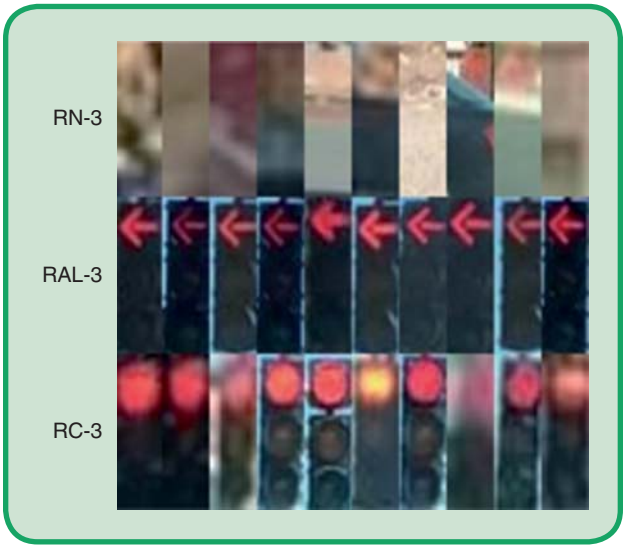


FIG 5 Examples of 3 classes of Red ROI-3.

Table 1. Number of training samples of Green ROI-*n* and Red ROI-*n*.

Class	*n* = 1	*n* = 3	*n* = 4
GN-*n*	13218	13218	13213
GAL-*n*	1485	835	–
GAR-*n*	1717	617	–
GAF-*n*	2489	1018	–
GC-*n*	3909	3662	–
GCGAL-*n*	–	–	369
GCGAR-*n*	–	–	281
GAFL-*n*	–	–	749
GAFR-*n*	–	–	1005
RN-*n*	7788	7619	–
RAL-*n*	1214	1235	–
RC-*n*	4768	5035	–

are in ROI-3 vertical setting, regardless of the status of the 4th light at the very bottom.

Table 1 shows the number of training samples of Green ROI-*n* and Red ROI-*n*, where *n* is 1, 3 or 4.

Features of a traffic light itself may not be as rich as other objects such as a human or a car. For example, a circular light is just a colored blob that looks similar to other objects in the same color. Therefore, it is difficult to distinguish the true traffic lights from other false candidates solely based on color analysis. The ROI-3 and ROI-4 samples are images of the holders, which provide additional information for detection and classification. The approach of combing all these information together is explained in IV-B2.

Table 2. Information of 23 test sequences.

Seq ID	Frames	Traffic Lights	Types of Traffic Lights	Description
1	91	182	Green circular×2.	Lights in all frames.
2	90	180	Green circular×2.	Lights in all frames.
3	61	147	Green arrow left×3.	Lights in all frames.
4	48	144	Green circular×3.	Lights in all frames.
5	156	312	Red circular×2.	Lights in all frames.
6	156	211	Green circular×2.	Lights at start, then move out.
7	214	428	Green circular×2.	Lights in all frames.
8	76	152	Red circular×2.	Lights in all frames.
9	245	305	Green circular×2.	Lights at start, then move out.
10	174	177	Green circular×2.	Lights at start, then move out.
11	91	348	Red circular×3; green arrow left; green arrow right; green arrow forward; green circular.	Red lights at start, then green lights.
12	56	280	Red arrow left; green arrow right×2; green arrow forward×2.	Lights in all frames.
13	82	70	Green circular×2.	Lights at start, then move out.
14	259	518	Green circular×2.	Lights in all frames.
15	65	325	Red arrow left; green arrow right×2; green arrow forward×2.	Lights in all frames.
16	185	242	Green circular×2.	Lights at start, then move out.
17	93	186	Red circular×2.	Lights in all frames.
18	630	0	None.	No traffic lights.
19	580	0	None.	No traffic lights.
20	416	0	None.	No traffic lights.
21	550	0	None.	No traffic lights.
22	759	0	None.	No traffic lights.
23	3035	0	None.	No traffic lights.
Total	8112	4207	–	–

B. Test Data

All test images are taken from the data set that we collected in the winter. The ground truths are manually labeled and are used for validating the results. In our proposed method, tracking technique is used to further improve the performance. However, traffic lights can move out of the image or change states during the tracking process. Therefore the test sequences need to cover many possible scenarios for all types of lights. Detailed information of the test sequences is shown in Table 2.

IV. Proposed Method of Traffic Light Detection and Recognition

Fig. 6 shows the flowchart of our proposed method of traffic light detection and recognition, which consists of three stages. Firstly, color extraction and candidates selection are performed over the input image. Secondly, to determine whether the selected candidates are traffic lights and what types of lights, they are processed by PCANet and SVM. Finally, tracking and forecasting techniques are applied to improve the performance and stabilize the final output.

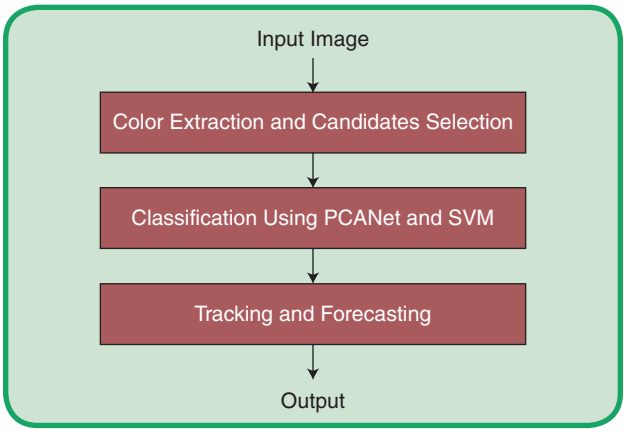


FIG 6 Flowchart of the proposed method of traffic light detection and recognition.

A. Locating Candidates Based on Color Extraction

To locate the traffic lights, color extraction is applied to locate the Region of Interest (ROI), i.e., the candidates. The images are converted to hue, saturation, and value (HSV) color space. Comparing with RGB color space, HSV color space is more robust against illumination variation and is more

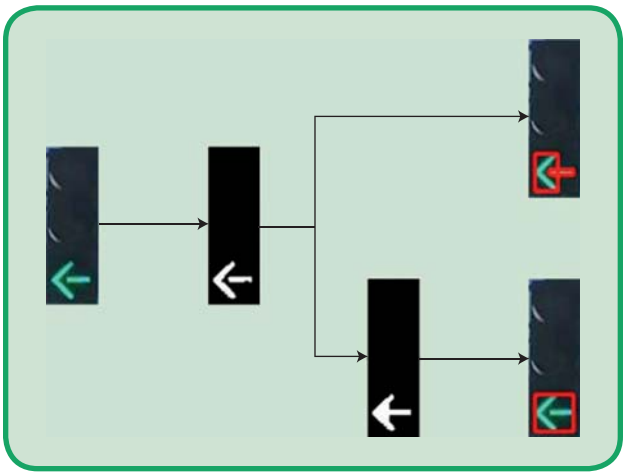


FIG 7 Color extraction, blob detection and closing operation.

FIG 8 A sample frame from our traffic light data set.

suitable for segmentation [26]. The desired color is extracted from an image mainly based on the hue values, which results a binary image. Suppose the HSV value of the ith pixel in an image is

$$HSV_i = \{h_i, s_i, v_i\} \tag{1}$$

In order to extract green pixels, we set the color thresholds based on the empirical data:

$$40 \leq h_i \leq 90 \tag{2}$$
$$60 \leq s_i \leq 255 \tag{3}$$
$$110 \leq v_i \leq 255 \tag{4}$$

In order to extract red pixels, besides (3) and (4), one of the following conditions must hold:

$$165 \leq h_i \leq 180 \tag{5}$$
$$0 \leq h_i \leq 20 \tag{6}$$

These values are adjustable and similar settings can be found in [11]. Note that the threshold values that we choose work well in OpenCV [25] and may need proper conversion in order to work with other libraries.

Blob detection can be implemented using flood-fill or contour following. The blobs can be considered as the potential candidates. However, it is possible that an arrow light may be labeled as two different regions, because the head and tail of an arrow are sometimes separated with a gap between them. When the traffic lights are closer to the camera, it is more likely that the gaps can be clearly seen and thus affect the result of blob extraction. To solve this problem, the closing operation is performed on the binary image obtained from color extraction. Closing operation is a typical morphological operation in image processing. It applies a dilation followed by an erosion, which eliminates gaps and holes on the binary image. Therefore, the arrow light can be detected as a whole and the candidates after closing is more reliable than the original candidates. Fig. 7 shows the original result of color extraction and blob detection (top right), and the result with closing operation (bottom right).

The side-effect of the closing operation is that it might connect a green light with other green objects in the background such as trees. When the traffic lights are far away from camera, this problem is more likely to occur because the black borders of traffic light holders are thin. However, when the traffic lights are far away, the gaps are more likely to be filled by the halo of the lights, or become invisible due to the limitation of image resolution. Therefore, the original candidates are more reliable than those after closing. It is difficult to determine whether the morphological closing operation should be applied. Therefore, we choose to keep both the original candidates and the candidates after closing operation. In case overlapped candidates are identified through the classification, the candidate with aspect ratio closest to one is selected.

The objective of eliminating false positives is considered in the latter part of the proposed method. Fig. 8 shows an example of the road images. In this image, there are four green traffic lights, but 895 green candidates can be extracted using the method mentioned above. This requires our classifier to be very strong to filter out the negative candidates while retain positive ones. However, even if the classifier is able to filter out 99% of the negative candidates, there are still about 9 false positives remaining in this image, which is an unacceptable result. Therefore, prefiltering and post-validation steps are necessary in addition to the classifier itself. For red traffic lights, the number of candidates is much smaller than that of the green traffic lights. For example, there are 19 red candidates in Fig. 8 from the color extraction.

In many previous works [4], [5], [8], [12], a variety of morphological filtering techniques were applied to eliminate some candidates for the purpose of reducing false positives. However, any filtering has a possibility of missing the true

traffic lights, because the traffic lights are not always clear due to their size and the obscure background in an image. Thus only aspect ratio check is performed in the proposed method and all blobs that pass the check are kept as candidates. The aspect ratio ar is defined as

$$ar = w/h \tag{7}$$

where w is the width and h is the height of the candidate. In order to pass the aspect ratio check, the following inequality must hold:

$$2/3 \leq ar \leq 3/2 \tag{8}$$

The aspect ratio check is to reduce the number of candidates. In Fig. 8, the number of green candidates is reduced to 51 and the number of red candidates in reduced to 9 after the aspect ratio check.

B. Classification

1) PCANet

The PCANet classifier is applied to determine whether a candidate is a traffic light or not. PCANet classifier consists of a PCA network and a multiclass SVM. The structure of PCANet is simple, including a number of PCA stages followed by an output stage. The number of PCA stages can be variant, but the typical value is 2, making it so called two-stage PCANet. As shown in [3], a two-stage PCANet outperforms the single stage PCANet in most cases, but further increase of the number of stages does not necessarily provide better performance, according to their empirical experience. Therefore, two-stage PCANet is used in our proposed method.

The structure of PCANet is to emulate that of a traditional convolutional neural network [27]. The convolution filter bank is chosen to be PCA filters. The nonlinear layer is the binary hashing (quantization). The pooling layer is the block-wise histogram of binary vectors. There are two parts in the PCA stage—patch mean removal and PCA filters convolution. For each pixel of the input image, there are a patch of pixels in the same size of the filter. Their mean are then removed from each patch, followed by PCA filter convolution. The PCA filters are obtained by unsupervised learning during the training process. The number of PCA filters can be variant. The impact of the number of PCA filters is discussed in [3]. Generally speaking, more PCA filters lead to better performance. In this paper, we choose 8 filters for both PCA stages and we find it is sufficient to deliver good performance.

The output stage consists of binary hashing and block-wise histogram. The output of PCA stages are converted to binary values, with positive value to one and else to zero. Thus a binary vector is obtained for each patch, and the length of this vector is fixed. This binary vector is then converted to decimal value. The block-wise histogram of these decimal values forms the output features. The SVM is then fed with the features from PCANet. Fig. 9 shows the structure of a two-stage PCANet. The number of filters in stage 1 is m and in stage 2 is n.

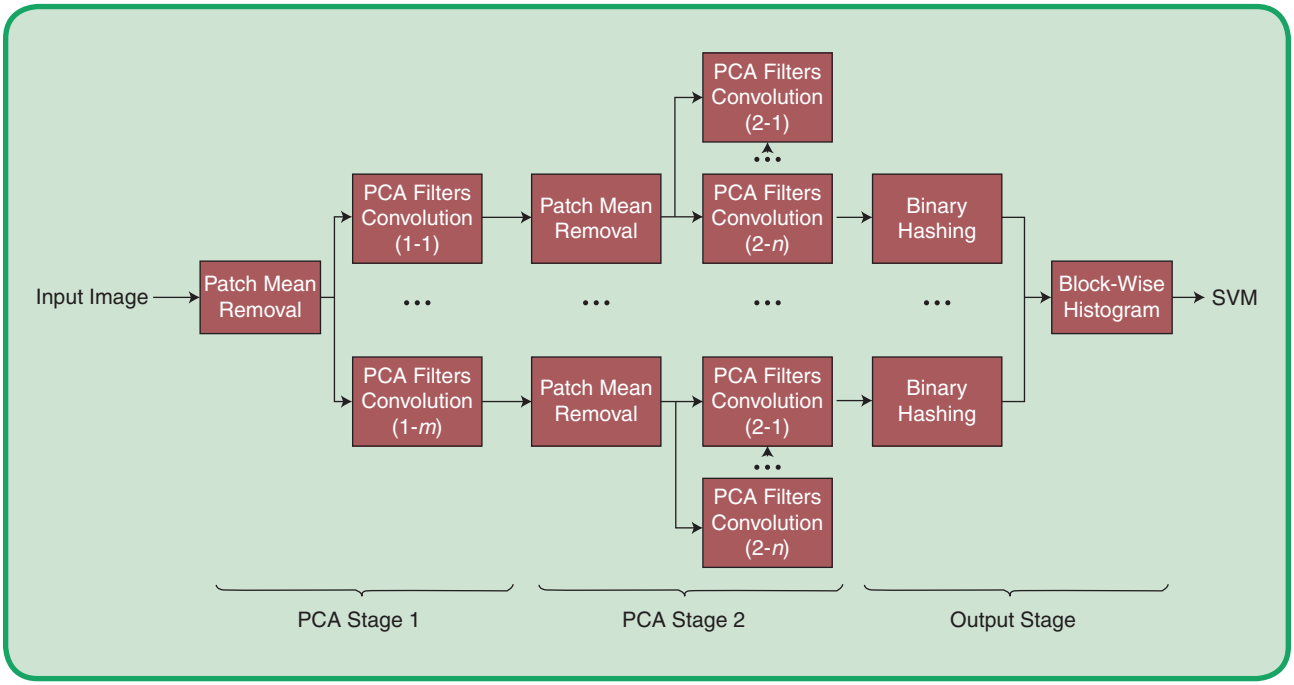


FIG 9 The structure of two-stage PCANet.

2) Recognizing Green Traffic Lights Using PCANet

As mentioned in IV-A, due to a large number of green objects in an image such as trees, street signs, and green vehicles, the classifier must be strong enough to eliminate the potential false positives while maintain a high detection rate. Using the green areas as candidates is not sufficient. For example, a fragment of tree leaves may occasionally look similar to the green lights in some frames, which causes false positive "flashing" in the video of detection results.

To solve this problem, a validation step is applied to the system. It is assumed that the traffic lights always appear in a holder. The traffic light holder contains three or four lamps that are vertically aligned in our collected data. Note that horizontal traffic lights are also often used and can be processed using the same method if the data set is available. In addition, these lamps have certain combinations. The traffic light holder area thus contains important information that can help us detect the traffic lights. In a vertical traffic light holder, the bottom one is always a green lamp. Therefore, the position of potential traffic light holder can be located according to the green area. The aspect ratio of the green area is approximately 1:1 and the green area is called as ROI-1. The traffic holder area with three lamps is called ROI-3 and the traffic holder area with four lamps is called ROI-4. Suppose the rectangular bounding box of ROI-1 is $R_{\text{ROI}-1}$ where

$$R_{\text{ROI}-1} = \{x_{\text{ROI}-1}, y_{\text{ROI}-1}, w_{\text{ROI}-1}, h_{\text{ROI}-1}\} \tag{9}$$

Similarly there are bounding boxes $R_{\text{ROI}-3}$ for ROI-3 and $R_{\text{ROI}-4}$ for ROI-4 where

$$R_{\text{ROI}-3} = \{x_{\text{ROI}-3}, y_{\text{ROI}-3}, w_{\text{ROI}-3}, h_{\text{ROI}-3}\} \tag{10}$$

$$R_{\text{ROI}-4} = \{x_{\text{ROI}-4}, y_{\text{ROI}-4}, w_{\text{ROI}-4}, h_{\text{ROI}-4}\} \tag{11}$$

The variables $x_{\text{ROI}-i}, y_{\text{ROI}-i}$ are the coordinates of the top-left corner of the bounding box $R_{\text{ROI}-i}, w_{\text{ROI}-i}$ is its width and $h_{\text{ROI}-i}$ is its height. The $R_{\text{ROI}-3}$ can be obtained based on $R_{\text{ROI}-1}$ as follows, where the coefficients are determined empirically based on the assumption that the lights are vertically aligned and the green light is the lowest light:

$$x_{\text{ROI}-3} = x_{\text{ROI}-1} - 0.1 \times w_{\text{ROI}-1} \tag{12}$$

$$y_{\text{ROI}-3} = y_{\text{ROI}-1} - 2.5 \times h_{\text{ROI}-1} \tag{13}$$

$$w_{\text{ROI}-3} = 1.2 \times w_{\text{ROI}-1} \tag{14}$$

$$h_{\text{ROI}-3} = 3.6 \times h_{\text{ROI}-1} \tag{15}$$

In the case of horizontally aligned lights, these coefficient should be changed accordingly. Similarly, the $R_{\text{ROI}-4}$ can be obtained based on $R_{\text{ROI}-1}$ as follows:

$$x_{\text{ROI}-4} = x_{\text{ROI}-1} - 0.1 \times w_{\text{ROI}-1} \tag{16}$$

$$y_{\text{ROI}-4} = y_{\text{ROI}-1} - 3.9 \times h_{\text{ROI}-1} \tag{17}$$

$$w_{\text{ROI}-4} = 1.2 \times w_{\text{ROI}-1} \tag{18}$$

$$h_{\text{ROI}-4} = 5.1 \times h_{\text{ROI}-1} \tag{19}$$

All samples of ROI-1 are resized to 10 × 10 pixels, all samples of ROI-3 to 10 × 33 pixels and all samples of ROI-4 to 10 × 43 pixels. Three PCANet classifiers are trained separately for ROI-1, ROI-3 and ROI-4. Each classifier is able to perform multiclass classification, such as distinguishing left arrows, right arrows, circular lights and negative samples.

In order to combine the results of these three classifiers, several methods are evaluated using the test data set. An intuitive solution is the voting strategy. The results of ROI-1, ROI-3 and ROI-4 are voted to several classes and the class that has the most votes is selected as the final result. However, this method is not accurate. The ROI-3 may contain partial area of a traffic light holder if it is actually a four-light holder. The ROI-4 may contain background if it is actually a three-light holder. Therefore, the positive results of ROI-3 and ROI-4 are both considered as possible regions. If any positive results of ROI-1 overlap with these regions, it is considered a true positive green light. This is a more plausible approach because the two cases mentioned above do contain the traffic light holders that are the possible regions. Although the class types determined by ROI-3 and ROI-4 may be inaccurate, the ROI-1 is capable of providing an accurate result.

3) Recognizing Red Traffic Lights using PCANet

Red traffic lights are recognized in a similar way as to green lights. The bounding boxes of Red ROI-1 and ROI-3 are expressed the same as that of the green lights shown in (9) and (10). Assuming the lights are vertically aligned and the red light is the top light, the $R_{\text{ROI}-3}$ can be obtained based on $R_{\text{ROI}-1}$ using Equation 12, 14, 15 and

$$y_{\text{ROI}-3} = y_{\text{ROI}-1} - 0.1 \times h_{\text{ROI}-1} \tag{20}$$

C. Stabilizing the Detection and Recognition Output

1) The Problem of Frame-by-Frame Detection

Frame-by-frame detection is important, but not sufficient to render stable output. The reasons are twofold. One aspect is that no detector can perform perfectly under all possible scenarios. Another aspect is that the input data sometimes are not of good quality. For example, vehicle vibrations may cause cameras to lose focus, making the frames vague. An arrow red traffic light in such situation may look identical to a circular red light and can hardly be recognized even by human eyes, which is shown on the

image in the center of Fig. 10. However, the arrow light is clear in other frames. If the detector recognizes this arrow light in previous frames and keeps track of it, a correct estimation can be provided in the vague frame even if the detector gives an incorrect result. In addition, there may be multiple lights in a frame, so multiple lights need to be distinguished and not confused with each other.

The goal of multiobject tracking is to recover the complete tracks of multiple objects and to give estimation of their current states. There are two categories of multiobject tracking methods: batch methods and online methods. The batch methods require the detection results of the entire sequence before analyzing the identity and constructing the trajectory of each object, which makes it impractical for real-time applications. The online methods are based on information that is available up to the current frame, which can provide results in real-time. Traffic light detection is a time-critical application that needs to give immediate feedback to the driver or controller, therefore multiobject tracking must be done using the online method. The online methods track objects from previous frames, and associate the tracking result with detection result of the current frame.

2) Tracking and Data Association

Here we propose an intuitive approach which is optimized for the traffic light detection application. For video camera at 30 frames per second (FPS), the motion of the lights between the adjacent frames are of small values. Therefore, an object in the next frame should be found near its location in the previous frame. Since color is an important feature of traffic lights, mean shift method is employed to locate the traffic light based on its previous position. Given a traffic light in the previous frame, the mean shift procedure calculates the histogram in the hue channel of the HSV color space, and then calculates histogram back-projection in the current frame in order to locate the light.

There are other tracking methods such as particle filter, which is proven to work for multiple people tracking [22]. We do not adopt it for two reasons. One is that traffic lights are small objects in a high resolution image which has 1920×1080 pixels. This makes it difficult for the particles to locate the traffic lights accurately and may need a large number of particles, which is computationally expensive. The other reason is that the weights of each particle cannot be evaluated effectively. The assumption that the detection confidence of each particle is higher when it gets closer to the actual position of the light is not true. The lights are so small in the image and a small deviation may lose the target completely. In addition, our detector is trained based on images of complete traffic lights, thus it cannot distinguish partial lights from backgrounds nor give higher confidence values for them.

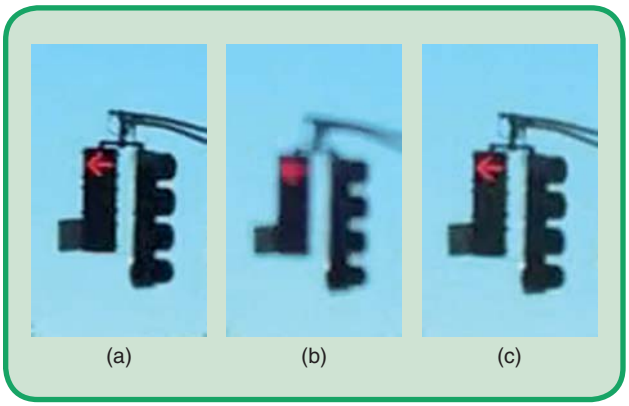


FIG 10 An arrow light in three consecutive frames. The middle one is vague and look similar to a circular light. A detector often fails on such vague frame.

For data association, [22] employs greedy data association and observes similar result compared with the Hungarian algorithm [28]. In our approach, the tracking result is simply associated with the detection result when they overlap. The reason is that the traffic lights are motionless in adjacent frames and mean shift performs well in locating them. In addition, unlike people detection, traffic lights do not intersect with each other and there is no need to consider the object identities switch problem, which makes it easier to associate the tracking and detection results. Once the association is established, the detected regions are used for mean shift tracking in the next frame, instead of using the regions found by mean shift itself. It solves the scale problem of mean shift and the detected regions are considered more accurate than the tracking result.

Building trajectories of the objects can overcome occasional misses, but still cannot filter out false positives. For example, if a rear light of a car is misclassified as a red traffic light in several frames, its trajectory is very likely to be built by multiobject tracking algorithms. However, the time series data for each object can be obtained from online multiobject tracking. Since the time series data consist of classification results over time, they can be used to generate the final output using forecasting and time series analysis.

3) Forecasting

Given the previous detection or recognition result of a target, the estimation of its current state is the final output. Such process is called forecasting and time series analysis. Multiobject tracking algorithms focus on building the trajectories and pay little attention to filtering out false positives. The idea is that the accumulated classification results of a false object often have different patterns compared to that of a true object, which can be used to filter out false positives. It is based on an assumption that the detector has the ability to distinguish the true positives and false positives to some extent, at least better then random

guessing. Otherwise, it is impossible to filter out the false positives. Some methods can be used to address the false positives problem. In [22], a tracker is only initialized in certain regions of the image, and is deactivated or terminated when there is no associated detection in a certain number of frames. Tracklet confidence is introduced in [29], which is influence by factors such as length, occlusion and affinity between tracking and detection.

In this paper, we employ a simple forecasting technique after online multiobject tracking, aiming at stabilizing the imperfect output of traffic light detection and recognition. For each object, there is a binary time series where 1 denotes that the detection result is true and 0 otherwise. The simple moving average (SMA) of the time series is then calculated. Let n be the window size of the SMA, b_i be the value of the time series in the ith frame, and S_m be the SMA value in frame m, the formula is

$$S_m = \frac{b_{m-(n-1)} + b_{m-(n-2)} + \cdots + b_{m-1} + b_m}{n} \tag{21}$$

or alternatively

$$S_m = S_{m-1} - \frac{b_{m-n}}{n} + \frac{b_m}{n} \tag{22}$$

It can be interpreted that the S_m is propagated from S_{m-1} while replacing the oldest value with the newest value in the sliding window. The S_m can be used to determine whether the object is considered positive, and a threshold t is set to determine the final output $\hat{b}_m$ as

$$\hat{b}_m = \begin{cases} 1 & S_m \geq t \\ 0 & S_m < t \end{cases} \tag{23}$$

When $\hat{b}_m$ is positive, a majority voting scheme is used to determine the type of the traffic light. The history labels of this particular light are voted to corresponding bins, and the one bin which has the most votes tells the type of the traffic light.

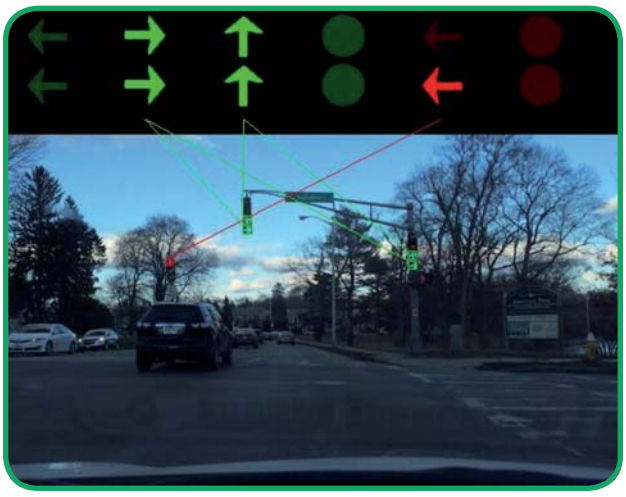


FIG 11 All traffic lights are detected and recognized correctly in the frame.

4) Minimizing Delays

Forecasting and time series analysis usually have delays. As the window size m grows, the delays become more severe. The delays at the head of a trajectory helps avoid picking up false positives, because false positives are expected to be occasional and inconsistent. However, slowly picking up true positives produces misses or false negatives. On the other hand, the delays at the tail of a trajectory helps avoid dropping off true positives, because true positives are expected to be consistent with minimal and temporal errors. However, slowly dropping off false positives produces erroneous output and increase the total number of false positives in the sequence. The delays must be balanced so that their side effects are minimized while their useful functionalities are not compromised.

At the head of trajectories, a dynamic threshold and modified moving average are employed. Suppose in frame m, the moving average $\hat{S}_m$ is modified as

$$\hat{S}_m = \begin{cases} \dfrac{b_{m-(n-1)} + b_{m-(n-2)} + \cdots + b_{m-1} + b_m}{n} & m \geq n \\ \dfrac{b_1 + b_2 + \cdots + b_{m-1} + b_m}{m} & m < n \end{cases} \tag{24}$$

and set the threshold $\hat{t}_m$ with a positive constant value α as

$$\hat{t}_m = \begin{cases} t & m \geq n \\ t + \alpha(1 - \dfrac{m}{n}) & m < n \end{cases} \tag{25}$$

At the beginning, the threshold is high and it drops slowly when more frames are available. The output from the first n frames is suppressed because of insufficient information to make a reliable decision. In a video at 30 FPS, 5 frames correspond to about 167 ms. According to [30], the reaction time of human is over a second. So such delays are acceptable. As a result, a true object with high confident is picked up quickly and the false positives can still be filtered out.

At the tail of trajectories, the object that no longer exists need to be dropped quickly. Traffic lights may change their states or move out of image during the tracking process. The transition of state is sudden. It usually has at most 1 frame that shows both lights are on or both are off, indicating the transition is taking place. This particular frame does not exist in many cases, so it is not reliable to tell when the transition occurs. However, traffic lights are motionless in adjacent frames and the last valid position of a currently off light is still useful. When transition happens, it can be determined if a detected light belongs to the same traffic light holder with a different colored light. Subsequently, the transition is identified and the expired information is dropped. On the other hand, when positive detections of a light around the edge of an image for a few consecutive frames are lost, the object is dropped to avoid erroneous output. Occlusion is not

considered in this paper, because it is not safe to predict the state of the light without actually seeing it completely.

V. Performance Evaluation

A. Detection and Recognition

Fig. 11 shows an example frame with detected traffic lights. Here two metrics named *precision* and *recall* are used, where

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \quad (26)$$

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \quad (27)$$

The *true positives* (TP) are samples that belong to this class, and are recognized as this class correctly. The *false positives* (FP) are samples not belong to this class, but are incorrectly recognized as this class. The *false negatives* (FN) are samples that belong to this class, but are recognized as the other classes erroneously.

The *true positives* here must be detected and recognized correctly. A detected but misclassified light does not provide correct identity of the actual light, which is a *false negative*. Meanwhile, it provides a false identity of another type of light, which is a *false positive*. Therefore, a detected but misclassified light is considered both a *false positive* and a *false negative*. For example, if a red left arrow light is detected but recognized as a red circular light, then the number of *false negatives* and the number of *false positives* are both incremented by 1. Table 3 shows the results of the test sequences with different configurations, such as using HOG or PCANet, with or without tracking. It is clear that the PCANet outperforms HOG and tracking technique improves the performance. The results are not perfect due to the lack of more training data and/or the occasional quality issue of captured video as shown in Fig. 10.

B. False Positives Evaluation

The number of false positives is evaluated over several traffic-light-free sequences

Table 3. Test result of 17 sequences that contain traffic lights.

	HOG				HOG + Tracking				PCANet				PCANet + Tracking			
Seq. ID	TP	FN	FP	Precision	Recall	TP	FN	FP	Precision	Recall	TP	FN	FP	Precision	Recall	TP	FN	FP	Precision	Recall
1	182	0	9	95.3%	100%	162	12	13	92.6%	93.1%	182	0	6	96.8%	100%	162	12	6	96.4%	93.1%
2	179	1	13	93.2%	99.4%	171	1	4	97.7%	99.4%	180	0	13	93.3%	100%	172	0	15	92.0%	100%
3	143	4	48	74.9%	97.3%	135	4	8	94.4%	97.1%	145	2	3	98.0%	98.6%	135	4	0	100%	97.1%
4	140	4	10	93.3%	97.2%	132	0	0	100%	100%	139	5	3	97.9%	96.5%	132	0	0	100%	100%
5	102	210	0	100%	32.7%	154	150	0	100%	50.7%	298	14	0	100%	95.5%	304	0	0	100%	100%
6	211	0	51	80.5%	100%	186	17	41	81.9%	91.6%	211	0	42	83.4%	100%	186	17	32	85.3%	91.6%
7	411	17	15	96.5%	96.0%	420	0	11	97.4%	100%	428	0	6	98.6%	100%	420	0	0	100%	100%
8	136	16	6	95.8%	89.5%	420	0	11	97.4%	100%	428	0	6	98.6%	100%	144	0	0	100%	100%
9	302	3	374	44.7%	99.0%	297	0	128	69.9%	100%	303	2	99	75.4%	99.3%	297	0	37	88.9%	100%
10	168	9	14	92.3%	94.9%	169	0	10	94.4%	100%	140	37	6	95.9%	79.1%	160	9	5	97.0%	94.7%
11	325	23	18	94.8%	93.4%	306	30	22	93.3%	91.1%	329	19	2	99.4%	94.5%	314	22	3	99.1%	93.5%
12	218	62	33	86.9%	77.9%	232	28	11	95.5%	89.2%	211	69	33	86.5%	75.4%	201	59	29	87.4%	77.3%
13	67	3	5	93.1%	95.7%	54	8	17	76.1%	87.1%	66	4	1	98.5%	94.3%	54	8	17	76.1%	87.1%
14	485	33	83	85.4%	93.6%	510	0	144	78.0%	100%	493	25	34	93.5%	95.2%	510	0	13	97.5%	100%
15	282	43	21	93.1%	86.8%	295	10	7	97.7%	96.7%	280	45	0	100%	86.2%	271	34	0	100%	88.9%
16	231	11	44	84.0%	95.4%	230	4	35	86.8%	98.3%	201	41	19	91.4%	83.1%	220	14	16	93.2%	94.0%
17	186	0	144	56.4%	100%	178	0	110	61.8%	100%	186	0	12	93.9%	100%	178	0	1	99.4%	100%
Total	3586	439	879	80.3%	89.1%	3612	253	548	86.8%	93.45%	3752	273	276	93.1%	93.2%	3698	167	168	95.7%	95.7%

Table 4. Number of false positives in traffic-light-free sequences.

	HOG		HOG + Tracking		PCANet		PCANet + Tracking	
Seq. ID	No.	No. per Frame	No.	No. per Frame	No.	No. per Frame	No.	No. per Frame
18	150	0.2381	12	0.0190	39	0.0619	0	0
19	45	0.0776	35	0.0603	56	0.0966	26	0.0448
20	11	0.0264	0	0	18	0.0433	12	0.0288
21	127	0.2309	23	0.0418	37	0.0673	9	0.0164
22	280	0.3689	125	0.1647	40	0.0527	6	0.0079
23	179	0.0590	85	0.0280	105	0.0346	80	0.0264
Total	792	0.1327	280	0.0469	295	0.0494	133	0.0223

Table 5. Results of several recent works on traffic lights detection.

Paper	Year	Method	Light Types	Image Size	Timing	Performance
Our approach	2016	PCANet; multiobject tracking	Green circular; red circular; Green arrow; red arrow	1920 × 1080	3 Hz	Precision 95.7%; recall 95.7%
[6]	2014	Spot light detection; adaptive template matching; Multiple model filter; single object tracking	Green circular; red circular; amber circular	–	–	Average accuracy 97.6%; False alarms ignored in detection
[11]	2014	Image processing; hidden markov models	Green circular; red circular; amber circular	648 × 488	25 frames per second	Overall detection rate 98.33% and 91.34% in different scenarios
[7]	2014	Fast radial symmetry transform	red circular; amber circular	240 × 320	Most time consuming part ~1.82 s	Precision 84.93%; recall 87.32%
[8]	2013	Filtering scheme with GPS information	Green circular; red circular	720 × 480	15.7 ms per frame	Precision 88.2%; recall 81.5%
[9]	2011	Traffic light mapping and localization using GPS information; Several probabilistic stages	Green circular; red circular; amber circular	1.3 megapixel	Real-time; 15Hz frame input	Accuracy: 91.7%
[10]	2011	Traffic light mapping and localization using GPS information; Onboard perception system	Green circular; red circular; amber circular; Green arrow; red arrow; amber arrow	2040 × 1080	4 Hz	Precision 99%; recall 62%

as shown in Table 4. Again, PCANet outperforms HOG and tracking technique improves the performance. The number of false positives is rapidly increased if there are misrecognized objects. A single misrecognized object produces 30 false positives in one second, if the video frame rate is 30 FPS.

The false positives are not eliminated completely in our proposed method simply because the trade-off between precision and recall. Eliminating more false positives may cause more false negatives, making precision increase and recall decrease, or vice versa. Reference [10] argues that false-positive green lights are dangerous and should be eliminated as much as possible, yielding 99% precision and 62% recall. While such argument is reasonable for practical applications, we do not perform such adjustments in this paper. Instead, we demonstrate highly accurate and well-balanced precision and recall results to validate our

proposed approach as well as the performance improvements by the introduction of PCANet and tracking.

VI. Discussion

A. Comparison with Related Work

Table 5 compares several recent papers on traffic light detection and recognition. However, it is difficult to compare them directly, because different testing data and different evaluation metrics were used. There are benchmarks for object detection and image classification like ImageNet [31], but no benchmark has yet been created for multiclass traffic light detection and classification. Researchers use their own collected data in their respective papers. Some papers [8]-[10] utilize the information other than images, such as GPS data and prior knowledge of traffic light locations. Some focus on a specific type of traffic lights, while others try to solve multiple colors and types at the same time. These factors make it difficult for us to compare their performance appropriately.

On the other hand, the efficiency is also hard to compare. The image sizes in these papers vary. In general, traffic lights can be seen even if they are still far away using higher resolution cameras. Instead, a far away traffic light may appear only as a few pixels in a lower resolution image. A higher resolution camera can provide clear images of traffic lights when they are further away. So the system may detect the traffic light slightly earlier, which provides the driver additional time to respond. However, large image size leads to higher computational cost and longer processing time. Another factor is that different hardware platforms were used in their implementations, such as desktop computers and on-board systems. Additional hardware modules may also be involved such as GPS and inertial measurement unit (IMU) [9].

B. Limitation and Plausibility

This paper presents a prototype system that can effectively detect several common types of traffic lights in a vertical aligned setting. We would like to emphasize that the proposed system is extendable. The ROI selection can be modified for other types of traffic lights such as horizontally aligned lights. The multiclass classification can be trained if sufficient data are provided. We feel confident that the proposed system can be extended to detect all type of traffic lights and even for other pattern recognition tasks with some modification.

Different light condition, color distortion, motion blur and variance of scenes may compromise the system performance in the real world. Thus the robustness of the trained model is a key factor in addition to detection accuracy. The robustness of our trained models can be improved by training with more data collected under all kinds of conditions using different cameras. Researchers in machine learning are often focused on investigating better algorithms, but sometimes getting more data beats a clever algorithm [32]. However, detecting traffic lights in severe weather or night condition may require different algorithms or even additional sensors and little research has been done on such topics. This will be part of our future work as more data become available.

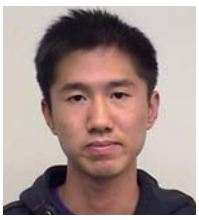
The processing time depends on the image size as well as the number of candidates in an image. The image size in our data set is 1920 × 1080, which is considerably larger than most of the other papers in Section VI-A. Our implementation is currently a single-thread version running at approximately 3 Hz on a CPU. Our implementation can be accelerated by using multiple CPU threads, GPUs or FPGA hardware. Previously we have successfully employed GPU to accelerate a traffic sign detection system in [33] and a fast deep learning system in [34]. Using hardware is another option to accelerate such systems. The most time consuming part is the PCANet classification part, which has been accelerated on an FPGA in our latest work [35].

Since the proposed system is based on a camera sensor, its reliability is directly affected by the video quality. There are many factors that can affect the output images, such as the camera sensors, configurations, post-processing procedures, and etc. An example of the data quality problem is shown in Fig. 10. Therefore, the proposed method is not expected to work at night. The traffic lights at night appear in different ways depending on the camera and its configurations. There may be halo effect around the lights, or the lights turn to be white at center and only have thin colored rings at the edge. A solution on one camera may not suitable for another camera. Therefore we decide not to investigate the problem at night.

VII. Conclusions

In this paper, we propose a system that can detect multiple types of green and red traffic lights accurately and reliably. Color extraction and blob detection are applied to locate the candidates with proper optimization. A classification and validation method using PCANet is then used for frame-by-frame detection. Multiobject tracking method and forecasting technique are employed to improve accuracy and produce stable results. As an additional contribution, we build a traffic light data set from the videos captured by a camera mounted behind the windshield. This data set has been released to the public for computer vision and machine learning research and is available online at http://computing.wpi.edu/Dataset.html.

About the Authors



Zhilu Chen received the B.E. degree in microelectronics from Xi'an Jiaotong University, Xi'an, China, in 2011, and the M.S. degree in electrical and computer engineering from Worcester Polytechnic Institute (WPI), Worcester, MA in 2013. He is currently pursuing

the Ph.D. degree from the Electrical and Computer Engineering Department, Worcester Polytechnic Institute, Worcester, MA. His research interests are computer vision, machine learning and GPU acceleration for Advanced Driver Assistance Systems.

Xinming Huang (M'01–SM'09) received the Ph.D. degree in electrical engineering from Virginia Tech, Blacksburg, VA in 2001. He is currently a Professor in the Department of Electrical and Computer Engineering at Worcester Polytechnic Institute (WPI), Worcester, MA. Previously he was a Member of Technical Staff with Bell Labs of Lucent Technologies, from 2001 to 2003. His research interests are in the areas of integrated circuits and embedded systems, with emphasis on reconfigurable computing, wireless communications, information security, computer vision, and machine learning.

References

[1] Red light running. Insurance Institute of Highway Safety [Online]. Available: http://www.iihs.org/iihs/topics/t/red-light-running/topicoverview

[2] Traffic lights recognition: Public benchmarks [Online]. Available: http://www.lara.prd.fr/benchmarks/trafficlightsrecognition

[3] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. (2014). Pcanet: A simple deep learning baseline for image classification? [Online]. Available: arXiv preprint arXiv:1404.3606.

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

[5] R. de Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2009, pp. 333–338.

[6] 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 *Proc. 17th Int. Conf. Information Fusion*, 2014, pp. 1–7.

[7] S. Sooksatra and T. Kondo, "Red traffic light detection using fast radial symmetry transform," in *Proc. 11th Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2014, pp. 1–6.

[8] T.-P. Sung and H.-M. Tsai, "Real-time traffic light recognition on mobile devices with geometry-based filtering," in *Proc. 7th Int. Conf. Distributed Smart Cameras*, 2013, pp. 1–7.

[9] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011, pp. 5784–5791.

[10] N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011, pp. 5421–5426.

[11] A. Gomez, F. Alencar, P. Prado, F. Osorio, and D. Wolf, "Traffic lights detection and state estimation using hidden markov models," in *Proc. IEEE Intelligent Vehicles Symp.*, 2014, pp. 750–755.

[12] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. D. Stefano, "Traffic sign detection via interest region extraction," *Pattern Recognit.,* vol. 48, no. 4, pp. 1039–1049, 2015.

[13] Z. Chen, Q. Shi, and X. Huang, "Automatic detection of traffic lights using support vector machine," in *Proc. IEEE Intelligent Vehicles Symp.*, 2015, pp. 37–40.

[14] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.,* vol. 18, no. 7, pp. 1527–1554, July 2006.

[15] I. Arel, D. Rose, and T. Karnowski, "Deep machine learning: a new frontier in artificial intelligence research," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, Nov. 2010.

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

[17] S. Lafuente-Arroyo, S. Maldonado-Bascon, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road sign tracking with a predictive filter solution," in *Proc. IEEE 32nd Annu. Conf. Industrial Electronics*, 2006, pp. 3314–3319.

[18] S. Lafuente-Arroyo, S. Maldonado-Bascon, P. Gil-Jimenez, J. Acevedo-Rodriguez, and R. Lopez-Sastre, "A tracking system for automated inventory of road signs," in *Proc. IEEE Intelligent Vehicles Symp.*, 2007, pp. 166–171.

[19] C. Keller, C. Sprunk, C. Bahlmann, J. Giebel, and G. Baratoff, "Real-time recognition of U.S. speed signs," in *Proc. IEEE Intelligent Vehicles Symp.*, 2008, pp. 518–523.

[20] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[21] W. Brendel, M. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 1273–1280.

[22] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 33, no. 9, pp. 1820–1833, Sept. 2011.

[23] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 1815–1821.

[24] D. C. Montgomery, L. A. Johnson, and J. S. Gardiner, *Forecasting and Time Series Analysis*. New York: McGraw-Hill, 1990.

[25] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.

[26] H. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognit.,* vol. 34, no. 12, pp. 2259–2281, 2001.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. New York: Curran Associates, 2012, pp. 1097–1105.

[28] H. W. Kuhn, "The Hungarian method for the assignment problem," in *50 Years of Integer Programming 1958-2008*. Berlin, Germany: Springer, 2010, pp. 29–47.

[29] S.-H. Bae and K.-J. Yoon, "Robust online multiobject tracking based on tracklet confidence and online discriminative appearance learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 1218–1225.

[30] K. Basak, S. N. Hetu, ZheminLi, C. L. Azevedo, H. Loganathan, T. Toledo, R. Xu, Y. Xu, L.-S. Peh, and M. Ben-Akiva, "Modeling reaction time within a traffic simulation model," in *Proc. 16th Int. IEEE Conf. Intelligent Transportation Systems*, 2013, pp. 302–309.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision,* vol. 115, no. 3, pp. 211–252, 2015.

[32] P. Domingos, "A few useful things to know about machine learning," *ACM Commun.*, vol. 55, no. 10, pp. 78–87, Oct. 2012.

[33] Z. Chen, X. Huang, Z. Ni, and H. He, "A GPU-based real-time traffic sign detection and recognition system," in *Proc. IEEE Symp. Computational Intelligence in Vehicles and Transportation Systems*, 2014, pp. 1–5.

[34] Z. Chen, J. Wang, H. He, and X. Huang, "A fast deep learning system using GPU," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2014, pp. 1552–1555.

[35] Y. Zhou, W. Wang, and X. Huang, "FPGA design for pcanet deep learning network," in *Proc. IEEE 23rd Annu. Int. Symp. Field-Programmable Custom Computing Machines*, 2015, pp. 232–232.

ITS