Journal of Ambient Intelligence and Humanized Computing (2022) 13:271–281
https://doi.org/10.1007/s12652-021-02900-y

ORIGINAL RESEARCH

Simultaneous detection and tracking using deep learning and integrated channel feature for ambint traffic light recognition

Ke Wang[1] · Xinwei Tang[1] · Shulian Zhao[2] · Yuchen Zhou[3]

Received: 5 August 2020 / Accepted: 9 January 2021 / Published online: 30 January 2021

Abstract

Perceiving the information about ambient traffic lights is an inevitable task for autonomous vehicles. To deal with the issue, this work develops an accurate and fast traffic light recognition strategy for autonomous vehicles by an onboard camera. In this paper, deep learning based detection and object tracking is synthesized to determine the position and color of traffic lights. First, the mechanism of simultaneous detection and tracking is founded, wherein the video reading module, convolutional neural network (CNN) module, integrated channel feature tracking (ICFT) module are run simultaneously. Then, the respective modules of detection and tracking are introduced. CNN model is designed and trained to obtain the position of traffic lights utilized as initial information for tracking. ICFT is applied to continually track the traffic light targets and determine the light color. Finally, the effectiveness of the presented method is validated via comparing with the state of art. Experiments results indicate that the proposed technique can improve the accuracy and speed of recognition. Our contributions are: (1) Establish a mechanism for simultaneous detection and tracking of traffic lights; (2) Carefully design the CNN architecture and ICFT features; (3)The precision and recall rates on traffic lights recognition reached 0.962 and 0.909, respectively, and the recognition speed reached 21.4FPS (GPU: Nvidia Titan Xp).

Keywords Traffic light recognition · Autonomous vehicle · Deep learning · Intelligent transportation

1 Introduction

Development of autonomous vehicles is one of the most prevalent research hotspots in the recent decade (Wang et al. 2014; Chen et al. 2019; Wang et al. 2020a). Traffic lights recognition is a critical technology in autonomous vehicles, which is able to obtain information on the status, color, and number of signal lights, and the lanes controlled by each light (Fairfield and Urmson 2011; Possatti et al. 2019; De Charette and Nashashibi 2009b). Although traffic lights are designed with various techniques, nowadays, there are still some challenges in identifying traffic lights. These challenges include: (1) in a complex and changing traffic environment, the recognition requires superior robustness (Chiang et al. 2011; Wang et al. 2020b; 2) to ensure the safety of the vehicle during the driving, the operation of the algorithm must be real-time (Greenhalgh and Mirmehdi 2012).

In the initial phase of developing traffic light recognition system, the feature-based methods (Saini et al. 2017; Lee et al. 2018; Cai et al. 2012; Diaz-Cabrera et al. 2015; Hosseinyalamdary and Yilmaz 2017; Wang and Xiong 2016) are widely adopted. For example, an ellipsoid geometry threshold model in HSV color space is built to extract interesting color regions. Besides, a kernel function is proposed to combine two heterogeneous features which are used to describe the candidate regions of traffic light. (Liu et al. 2016) But it can not perform well when it occurs

✉ Ke Wang
kw@cqu.edu.cn

Xinwei Tang
txw@cqu.edu.cn

Shulian Zhao
zhaosl10@mails.tsinghua.edu.cn

Yuchen Zhou
zyckjj@sina.com

[1] School of Automobile Engineering, State Key Lab of Mechanical Transmission, Chongqing University, Chongqing 400044, China

[2] School of Vehicle and Mobility, Tsinghua University, Beijing 401122, China

[3] School of Automobile Engineering, Chongqing University, Chongqing 400044, China

to the diverse weather with various brightness (Wang et al. 2021). Furthermore, the adaptive background suppression filter is implemented to predict the location for traffic lights (Shi et al. 2015). This method highlights the traffic light candidate regions while suppressing the undesired backgrounds. Besides, several features such as the aspect ratio, area, location, and context of traffic lights are tried (Li et al. 2017; Kim et al. 2013; De Charette and Nashashibi 2009a). The contribution of related references is to design and use various new features on one or more conditions to improve the accuracy of traffic lights detection. But their common challenge is that this feature design based on the researcher's prior knowledge cannot cope with complex and diverse realistic scenarios.

In recent years, deep learning methods which can provide models imitating neural decision-making are applied to deal with classification and object detection (Jensen et al. 2016). For example, in some works, various deep neural network algorithms are trained as efficient classifiers based on the cumulative training data. The contribution of designing and using neural networks is to greatly improve the accuracy of traffic light recognition in dynamic scenes. Because neural networks establish an implicit function to describe various characteristics of traffic lights.(Lee and Kim 2019; Bach et al. 2018; John et al. 2015; Chen and Huang 2016) Recent studies reveal that a combination of the image information and deep learning is a promising way to promote the performance of recognition (Wang and Zhou 2018; Wang et al. 2019; Hirabayashi et al. 2019), wherein prior feature is exploited to generate region of interest (ROI), and neural network is utilized to determine the state or color of traffic lights. John et al. (2014) used image processing techniques to extract the texture, color, and shape features of the candidate area hereafter the identification of the traffic light state is made by an artificial neural network using Multilayer-Perceptron (MLP). In these works, preprocessing slightly reduces the amount of calculation and saves processing time. Still, the common problem of deep learning methods is the excessive calculation that slows down the speed of processing and instability in video detection.

In order to reduce computational redundancy, achieve the requirement of real-time for autonomous vehicles, some researchers are exploring the pattern of informing drivers the position, status, and remaining time of traffic lights through Vehicle-to-roadside-Infrastructure (V2I) or GPS in the last several years. For example, Hirabayashi et al. (2019) uses current location and finds traffic lights on the road. Ci et al. (2019) studies the effect of V2I on traffic flow at signalized intersections. But the large-scale introduction of V2I requires a large investment in infrastructure, which will not be possible in the short term. Therefore, it is still meaningful to study onboard traffic light recognition algorithms.

This paper, proposes a novel traffic lights recognition strategy. First, the multi-thread program is built, wherein the video reading, CNN model, ICFT is settled. Then, the respective module of detection and tracking are cooperated to search the traffic light targets and determine the light color. Finally, the performance of the presented traffic lights recognition method is validated in experiments. The results indicate that the presented method is a promising choice for traffic lights recognition.

Three original innovations and contributions are underlined in this article: (1) a composite mechanism of traffic light recognition is constructed to jointly utilize both detection and tracking information. To the best of our knowledge, this is a novel attempt to combine deep learning and object tracking methods in traffic lights recognition of autonomous vehicles. (2) The architecture of CNN and the features in ICFT are well-designed and suitable for traffic light recognition. (3) Compared with traditional image processing methods or a single deep learning algorithm, the proposed strategy is of better recognition accuracy and speed for the traffic light.

The following content is arranged in this layout: the part of the method detail is explained in Sect. 2. Section 3 describes the results and analysis of various experiments performed on the dataset. Finally, the conclusion and future work are summarized in Sect. 4.

2 Methods

In this section, the details of the method are given that including the mechanism, CNN model, and IFCT. The constructions and mathematical formulations of these three parts are expounded carefully.

2.1 Mechanism of simultaneous detection and tracking

In this paper, an innovative mechanism of simultaneous detection and tracking is created. There are three threads in the mechanism: Reading, Detection and Tracking.

Figure 1 demonstrates the main process of the mechanism. The output of the detection thread is utilized as auxiliary information to update and correct the initial information for the tracking thread.

As for the description in time scale, which is shown in Fig. 2. The recognition process can be recognized as a cycle without a fixed period. Once the tracking module starts, the frames of the image would be quickly proceed based on the initial information, and outputs are saved. After each detection, Inter-frame Buffer discriminates the newly appearing or disappearing target, filtering out the influence of the mutation caused by false detection, and completing the update of

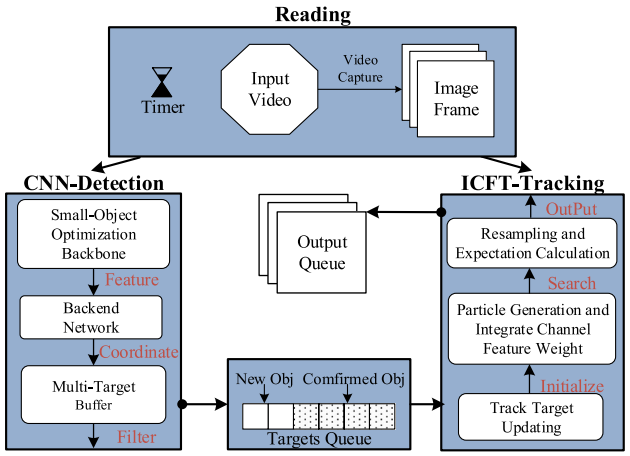

Fig. 1 The mechanism of traffic light recognition algorithm. The reading thread reads every frame from the input video with a speed of 100 frames per second. The detection thread produces the updated coordinate information of targets which is needed by tracking thread. The tracking also runs on the newest image captured by the reading thread

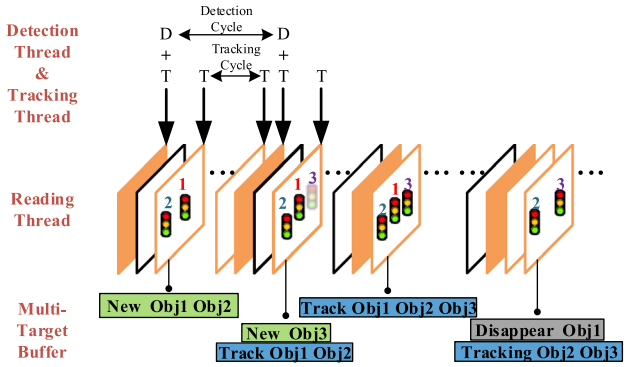

Fig. 2 The process in time scale. Each picture is read from the input video, and we take the sequence of pictures that we keep reading as the timeline. When the first frame of the video is read, the detection thread starts running immediately. The tracking thread won't start until the first target is found

the target position and quantity. Besides, the final candidate area given by Inter-frame Buffer is corrected using MSER (Maximally Stable Extreme Regions) to acquire a more accurate initial information for the tracking thread.

2.2 Deep learning based detection

A deep learning based method is implemented in the detection part.The CNN model is founded which consists of two main parts: backbone and backend. Table 1 shows the brief structure of the backbone network with some main layers. The backbone is composed of five residual network blocks. Different from the sequential networks such as GoogleNet and VGG19, residual networks can better solve the overfitting problem of deep neural networks. In terms of the number of network layers, in order to increase the calculation speed, the network depth is strictly limited. Compared with Faster R-CNN (152 layers), the proposed network model has only 58 layers. Before the data enter each block, the feature map is processed by a convolutional layer with a stride of 2, the size is reduced to a quarter, and the number of filters is doubled. The softmax is used as the activation function.

In the backend network, some networks only use single-scale feature. Many network models employ feature maps of different sizes to detect targets, such as SSD. However, SSD does not reuse low-level high-resolution feature maps, that is, does not make full use of the spatial information in the low-level feature maps, which is very important for the detection of small objects. Therefore, we add the feature maps obtained by the last residual networks to the previous feature map. Through such a connection, the feature maps used in each layer of prediction are fused with different resolutions and different strength of semantic features.

Table 1 The Backbone of the Network

	Type	Filters	Size	Output
	Convolutional	32	3×3	512×288
	Convolutional	32	3×3 / 2	256×144
	Convolutional	16	1×1	
1×	Convolutional	32	3×3	
	Residual			256×144
	Convolutional	64	3×3 / 2	128×72
	Convolutional	32	1×1	
1×	Convolutional	64	3×3	
	Residual			128×72
	Convolutional	128	3×3 / 2	64×36
	Convolutional	64	1×1	
4×	Convolutional	128	3×3	
	Residual			64×36
	Convolutional	256	3×3 / 2	32×18
	Convolutional	128	1×1	
4×	Convolutional	256	3×3	
	Residual			32×18
	Convolutional	512	3×3 / 2	16×9
	Convolutional	256	1×1	
2×	Convolutional	512	3×3	
	Residual			16×9
	Avgpool		Global	
	Connected		1000	
	Softmax			

Convolutional: convolutional layer, residual: residual network, *Num*×: repetition times of the structure, last three lines represent the pooling method, the output connection scale, and the activation function used by the network

In the way of connection, the Add function is adopted rather than the Concatenate layer as usual. At the same time, since this method only adds additional cross-layer connections to the original network, practically no additional time and calculations are required. The calculation amount of the Concatenate layer is twice that of the Add layer.

According to the explanation above the entire CNN is illustrated in Fig. 3.

2.3 Integrated channel feature tracking

In the proposed method, an integrated channel feature is used as an object model to compute the weight. The steps of the individual tracking algorithm will be specifically described below:

2.3.1 Target model description

This method uses the integrated channel feature function as the description of the target model. The integrated feature includes HSV and LBP, of which the calculation method is introduced later.

2.3.2 Particle sample set and particle initialization

The position and size of the traffic light target in the video is represented by a rectangular box, so the state space $s_t^{(n)}$ of the particle sample of the traffic light at t time is constructed by four parameters of the rectangle:

$$s_t^{(n)} = \left[x_t^{(n)}, y_t^{(n)}, h_t^{(n)}, w_t^{(n)}, a_t^{(n)}\right] \tag{1}$$

Where $n \in \{1, 2, \ldots, N\}$ and N is the number of random particles, $x_t^{(n)}$ and $y_t^{(n)}$ denote the center coordinate of the rectangular box, $h_t^{(n)}$ and $w_t^{(n)}$ determine the height and width of the rectangular box, $a_t^{(n)}$ is the corresponding scale factor. In particle initialization, a random particle set with N particles of which each state vector obeying a Gaussian distribution is generated.

The range of traffic lights in the image area can be estimated and restricted by the transition model. A second-order auto-regressive dynamics model is adopted. The particle sample set is propagated through the system state transition equation to obtain a new particle sample set:

$$s_t^{(n)} = A_1 s_{t-1}^{(n)} + A_2 s_{t-2}^{(n)} + BW \tag{2}$$

$$W \tilde{} N(0, \Lambda) \tag{3}$$

Where A_1, A_2, m is the Auto-regressive coefficients, and taking $A_1 = 2.0, A_2 = -1.0, B = 1.0$. $N(0, \Lambda)$ denotes the Gaussian distribution with zero mean and covariance $\Lambda = \mathrm{diag}\left(\sigma_x^2, \sigma_y^2, \sigma_w^2, \sigma_h^2, \sigma_a^2\right)$. Here, $\sigma_x^2 = 2.0, \sigma_y^2 = 2.0, \sigma_w^2 = 1.0, \sigma_h^2 = 1.0, \sigma_a^2 = 0.01$.

2.3.3 Weight calculation

First, the histograms of the Hue and Saturation channels of the target image and particle samples are computed separately.

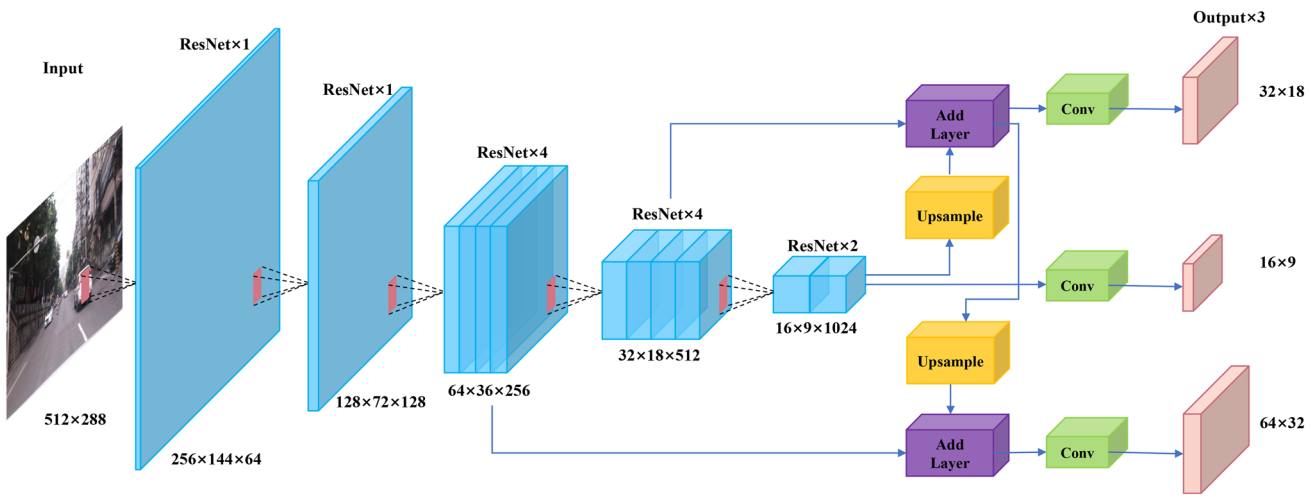

Fig. 3 The structure of the detection network. The feature map is extracted in the backbone while detection is finished in the backend of the network. The number of network layers and the number of filters in the backbone are set properly to improve the computational efficiency without excessively reducing the recognition accuracy. Subsequently, the connection of multi-scale feature maps is created in the backend network to enhance the performance of small object detection without substantially increasing the calculation amount of the original model

Then, Bhattacharyya coefficient is used to calculate the likelihood between two histograms:

$$\beta\left(p_{s_t^{(n)}}, q_0\right) = \frac{1}{2}\sum_{H,\,S}\sum_{u=1}^{m}\sqrt{p_{s_t^{(n)}}^{(u)} q_0^{(u)}} \tag{4}$$

Where $p_{s_t^{(n)}}^{(u)}$ denotes each histogram bin of one particle sample, q_0 is each histogram bin of target, and m is the number of histogram bins. Each histogram value $c_t^{(n)}$ for the particle sample set $s_t^{(n)}$ is calculated by Bhattacharyya coefficient:

$$c_t^{(n)} = f_c \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\left[1-\beta\left(p_{s_t^{(n)}}, q_0\right)\right]}{2\sigma^2}} \tag{5}$$

$$f_c = \frac{1}{\sum_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\left[1-\beta\left(p_{s_t^{(n)}}, q_0\right)\right]}{2\sigma^2}}} \tag{6}$$

Where f_c is the normalization coefficient as well as the following f_h.

Second, the LBP histogram is calculated. Then calculate the histogram of each cell, that is, the frequency of each number (assuming the decimal number LBP value). Similarly, the $h_t^{(n)}$ is calculated:

$$h_t^{(n)} = f_h \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\left[1-\beta\left(j_{s_t^{(n)}}, k_0\right)\right]}{2\sigma^2}} \tag{7}$$

Where $j_{s_t^{(n)}}$ is the LBP histogram of each particle samples, k_0 is the LBP histogram of the target.

Also, the distance weight is calculated:

$$r_t^{(n)} = \left(x_t^{(n)} - x_0\right)^2 + \left(y_t^{(n)} - y_0\right)^2 \tag{8}$$

$$R_t^{(n)} = e^{-\frac{r_t^{(n)2}/2\sigma^2}{\sigma\sqrt{2\pi}}} \tag{9}$$

$$\sigma = \frac{1}{3} w_t^{(n)2} \tag{10}$$

Where x_0, y_0 is the coordination of the target center in the image, $r_t^{(n)}$ is the distance between each particle and target center, and $R_t^{(n)}$ indicates the distance weight.

Then the distance weight is assigned to every feature weight:

$$C_t^{(n)} = c_t^{(n)} R_t^{(n)} \tag{11}$$

$$H_t^{(n)} = h_t^{(n)} R_t^{(n)} \tag{12}$$

Therefore, the Integrated Channel Feature Based Weight can be obtained:

$$I_t^{(n)} = \sqrt{C_t^{(n)} H_t^{(n)}} \tag{13}$$

The average of the particle sample set based on the weight is estimated as the output of the object tracking:

$$E\left(s_t^{(n)}\right) = \sum_{n=1}^{N} I_t^{(n)} s_t^{(n)} \tag{14}$$

2.3.4 Re-sampling

First, the particles are sorted according to the weight size, and then a new set of particles is re-sampled according to the discrete probability distribution rules obtained after sorting. The newly generated particles are given equal initialization weights. To maintain particle diversity, Gaussian noise is added to the re-sampling process.

2.4 Inter-frame buffer

The inter-frame buffer is demonstrated in Fig. 4. It is assumed that after the *kth* detection, a certain target is found, and the distance of which between all the targets tracked in the previous frame is compared with the threshold for determining whether it is a new target. If it is a new target, the new target will not be tracked in this cycle right away.

Then in the $(k+1)th$ detection, if the target still appears, the target will be tracked, that is, the new target enters. But if the target does not appear for the $(k+1)th$ test, it will not enter the tracking.

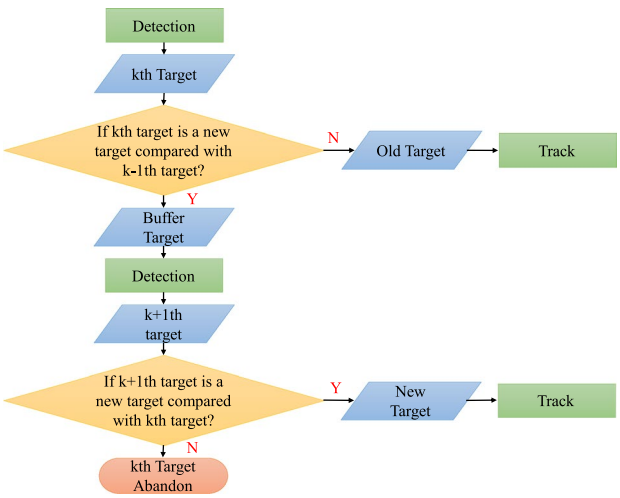

Fig. 4 The process of inter-frame buffer

After the new target obtained, MSER is carried out. MSER performs a binarization process on an image that has been processed into a gray-scale image. The coordination of darker traffic light cases set in other backgrounds can be corrected by MSER as shown in Fig. 5.

3 Results and discussion

First, the detection network is trained and four derived networks are compared afterward. Meanwhile, the performance comparison between the common single-channel feature and the proposed integrated channel feature is also implemented. Finally, the overall algorithm is tested.

3.1 Datasets

3.1.1 Berkeley deep drive 100K

44,932 images that traffic lights appear in diverse transportation and weather conditions are obtained from the 80 thousand annotation files in the BDD100K. The resolution of training images is 1280×720 pixels and the frame rate of which is 30FPS.

3.1.2 Local urban dataset

A local urban dataset is established to evaluate the algorithm we developed. The data is captured at Jiangbei District, Chongqing, China. The video acquisition device is the Logitech C922 HD Camera. The camera is fixed at the top of the front windshield of an electric vehicle. All the videos are 720p with a frame rate of 30FPS. After editing and filtering, 22 videos are finally reserved. Meanwhile, these sequences of videos are split into 1770 images.

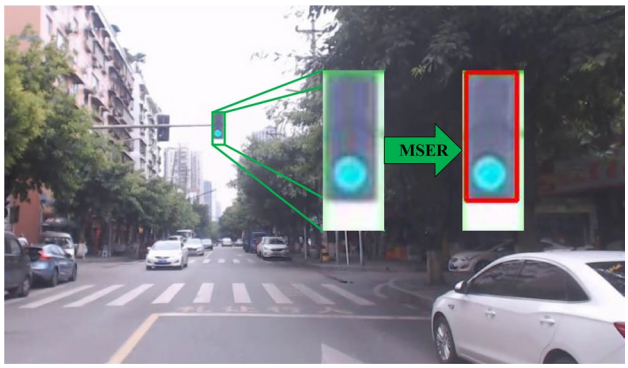

Fig. 5 MSER correction

3.2 Evaluations of detection network

The computer runs in the whole experiment is equipped with Nvidia Titan XP with 12 GB memory and the resolution of training input images is 521×288. LUD is uesd in the test of network models trained before.

The details of detection results compared with YOLOv3 are listed in Table 2. Our network model and YOLOv3 have little difference in the number of TP and FN, so the recall rate is similar. But the number of FP is reduced by 15%. This means that our network has stronger anti-interference ability. One step closer, the accuracy of our network has increased by 2.3 percentage points, and the F1 value is also better. At the same time, the results indicate that our network has a faster calculation speed, and its operating speed has increased by 23.8%. Through comparison and analysis, our network is greatly increasing the calculation speed, at the same time, it still maintains the recognition performance level of the existing excellent network models, and has stronger stability.

To further prove the optimality of our network, four self-derived networks (MF, BR, MU, MFBR) are introduced to be compared with our network in the experiment. These networks are of some differences in their structure and they are given in Table 3. The presence or absence of the first ResNet determines the initial size of the resolution of the three feature maps. The feature map sizes of BR and MFBR are 4 times larger than those of other models. The larger the size of the feature map, the more conducive to the recognition of small-sized targets, but the fewer global features obtained by the receptive field. Setting more filters in the network can get more features. For example, the number of filters for MF and MFBR is twice that of other models, and the number of features they obtain is also twice that of other models. With more features, the more accurate the model's description of the target, but obviously the amount of calculation is also greater. MU increases the multiple of up-sampling, which

Table 2 Details of the detection results

Network	TP	FP	FN	FPS	Precision	Recall	F1
YOLOv3	196	40	38	14.47	0.830	0.838	0.834
Ours	197	34	37	17.92	0.853	0.842	0.847

$Precision = TP/(TP + FP)$, $Recall = TP/(TP + FN)$, where TP represents number of true-positives, FP represents number of false-positives and FN represents number of false-negatives. $F_1 = 2 \times Precision \times Recall/(Precision + Recall)$

Table 3 Parameters of Models

Network	First ResNet	Filters times	Up-sampling times	Connection layer	Resolutions of feature maps
MF	I/A	2	2	Concat	16×9, 32×18, 64×36
BR	N/A	1	2	Concat	32×18, 64×36, 128×72
MU	I/A	1	4	Concat	16×9, 32×18, 128×72
MFBR	N/A	2	2	Concat	32×18, 64×36, 128×72
Ours	I/A	1	2	Add	16×9, 32×18, 64×36

MF more filters, *BR* bigger resolution, *MU* more up-sampling. MF has more filters in network. Resolution of three feature map in BR is bigger. The first up-sampling time of MU is 4, therefore the third feature map is bigger. MFBR possesses the characteristics of both MF and BR

Table 4 Details of detection results via self-derived networks

Network	TP	FP	FN	FPS	Precision	Recall	F1
MF	196	36	38	15.75	0.845	0.838	0.841
BR	188	26	46	16.08	0.878	0.803	0.839
MU	175	27	59	18.98	0.866	0.748	0.803
MFBR	194	23	40	12.94	0.894	0.829	0.860
Ours	197	34	37	17.92	0.853	0.842	0.847

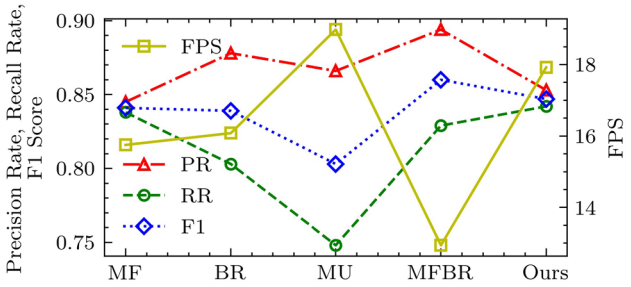

Fig. 6 Comparison of the detection performance. In the daytime, the conditions of captured images are favorable and the similar objects of a traffic light are few, so the model with a better recall rate is preferred. Furthermore, the detection speed is always the indicator that we care about. According to the comprehensive comparison, our network is the optimal network that is suitable for the stage of detection for the overall algorithm

increases the area of the feature map (the largest one) used to identify the smallest target, which is more beneficial to the recognition of small targets. There are two ways to join the two feature maps in the connection layer: Add is more efficient and Concat retains more information.

Each network introduced above is trained on the identical device and settings. The details of detection results are shown in Table 4 and Fig. 6. About the precision rate, all the derived networks are higher than YOLOv3. Especially, MFBR is 6.4 percentage points higher than YOLOv3. What is opposite, the recall rate of most models is the lower or equal compared to YOLOv3. Only our network is a little bit higher than YOLOv3. Compared with the F1 score which can measure the accuracy of the two-classification model, MFBR and our network are better. Referred to the item of FPS, MU and our network are much faster than YOLOv3 and increase by 31.2% and 23.8% respectively. Although the precision rate and F1 score of MFBR are both best, the speed is too slow and cannot reach real-time detection. In Fig. 7, there are a couple of detection results of our network in the test.

3.3 Feature channel tracking comparison

The Intersection over Union (IoU) value between the tracking result and the ground truth after tracking a certain number of times is applied to characterize the accuracy. The Success Rate Map and Accuracy Map of different tracking features are displayed in Fig. 8. The detailed data is shown in Table 5 and the advantages are bolded.

According to the experimental results, among these groups of channel features, the Average Error, Average IOU, and AUC of the single-channel are not as excellent as integrated channel features. Furthermore, in our algorithm, HSV+LBP reaches 52.5 FPS which is much quicker than other integrated groups. Overall, the performance of the integrated channel feature is satisfactory, and the accuracy and stability are better than the single-channel feature.

3.4 Entire algorithm performance evaluation

The five test videos in the evaluation are shown in Fig. 10. Table 6 reveals the result of this test. In the experiment, the algorithm processes a total of 14103 frames of the image during the experiment, which takes 660.2 seconds, and the actual average running frame rate is 21.4 FPS. Comparing with the performance of the YOLOv3 in Table 2 (14.47FPS),

Fig. 7 Samples of traffic light detection result. The detected traffic lights are marked by red rectangles

the complete algorithm can process 47.9% more images in the same amount of time.

Referring to the precision rate and recall rate, the precision rate ranges from 0.937 to 0.973 with an average of 0.962; the recall rate ranges from 0.834 to 0.953 with an average of 0.909. By comparison, both performances are more superior to the result of previous evaluation on YOLOv3—precision rate increased by 15.9%, and the recall rate increased by 8.5%. As revealed in the experimental results, there is a significant improvement in traffic light recognition supported by the proposed algorithm.

In the entire algorithm experiment, the algorithm we proposed still has certain limitations. In Fig. 9, there are two typical defects in the experiment: (a) The rightmost traffic light in the bottom row is missed; (b) Although the traffic light is found, its box position is interfered with by the countdown indicator next to the light. Higher Precision rate means accurate recognition and few false detections, but the recall rate of the proposed method is relatively low, that is, there is a case of missed detection. In addition, when the target pixel area is very small, the image composed of half of the black countdown indicator and a red number is similar to the traffic light, resulting in inaccurate positioning of the traffic light (Fig. 10).

For the five video test results in Table 6, we conducted a statistical significance test to determine if the average performance data of the proposed method in the experiment is significantly improved compared to YOLOv3 (results in Table 2). The test process of FPS is shown below. The

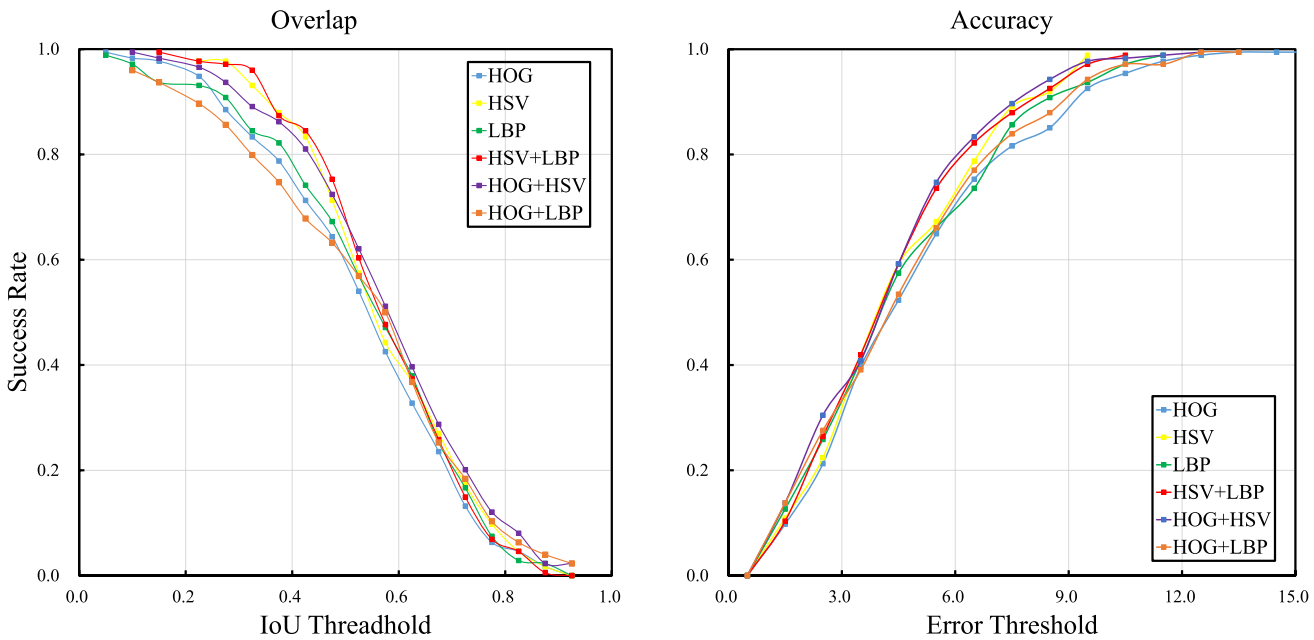

Fig. 8 Performance of tracking

Table 5 Details of tracking test results

Channels	FPS	Average error	Average IOU	AUC
HOG	10.4	4.954	0.528	0.307
HSV	200.8	4.480	0.565	0.342
LBP	59.8	4.667	0.539	0.321
HOG+HSV	10.7	4.268	0.573	0.348
HOG+LBP	7.8	4.711	0.530	0.314
HSV+LBP(Ours)	52.5	4.393	0.567	0.344

Fig. 9 Mistakes in detection result

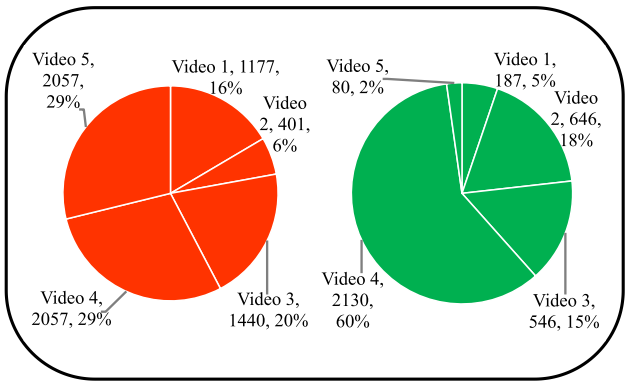

Fig. 10 Proportions of light's color in videos. The test video contains 10712 targets, of which red and green lights are 7132 and 3589 respectively

Table 6 Details of evaluation results

Video	pFrames	pTime	Fps	Precision	Recall
1	1990	106.8s	18.6	0.959	0.953
2	2263	115.3s	19.6	0.937	0.834
3	2841	132.2s	21.5	0.962	0.888
4	4381	190.5s	23.0	0.973	0.920
5	2628	115.4s	22.8	0.954	0.922
Total	14103	660.2s	21.4	0.962	0.909

pFrames denotes the number of processed frames, pTime means the processing time

significance level α is set to 0.05. The hypotheses for the significance test are as follows:

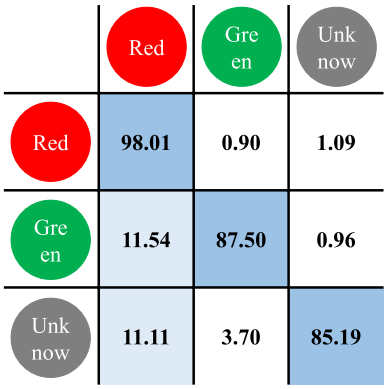

Fig. 11 Confusion matrix of color classification

H_0: The FPS of the proposed method is not higher than that of YOLOV3.

H_1: The FPS of the proposed method is higher than that of YOLOV3.

For the test of a single normal population mean, when the standard deviation is unknown, the T-test is used:

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2} = 1.975 \tag{15}$$

$$T = \frac{\overline{X} - \mu_0}{S}\sqrt{n} = 7.846 \tag{16}$$

$$\text{Rejection interval}: \{t > t_{1-\alpha}(n-1) = 2.132\} \tag{17}$$

Where S is the sample standard deviation, T is the test statistic. Since T is in the rejection interval, H_0 is rejected, that is, H_1 is accepted. The significance test results of Precision and Recall are also the same. Therefore, it can be considered that the improvement of the proposed method compared to YOLOv3 is not accidental.

The color of traffic lights is told by the hue feature. The corresponding confusion matrix is shown in Fig. 11. From the result, the recognition accuracy of red lights is higher than that of green lights because the difference between red and background color is more significant than green especially referring to blue sky and green trees.

Figure 12 reveals the time consummation of the detection thread and the tracking thread and total process. The average detecting time on a single frame of the detection thread is not much different from the time consumption in the previous experiment, which is basically above and below 0.056s. However, the average tracking time on a single frame of the tracking thread is much shorter, which takes about 0.019s. The shorter the time required to process the task, the lower the computational complexity of the algorithm. In Fig. 11, both Track Thread and Detection Thread can achieve traffic

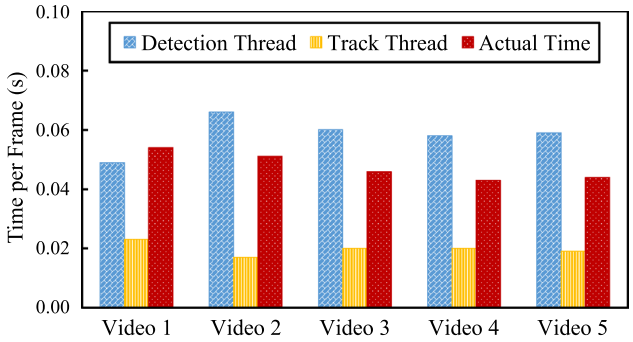

Fig. 12 Processing time of every thread

light targets However, the average time for Detection Thread to process each frame of pictures is 2.6 times that of Track Thread. That is to say, the computational complexity of the detection network model is 2.6 times that of the tracking model. When compared with other deep learning methods, the FPS of YOLOv3 is 14.47, the FPS of the proposed method is 21.4. Therefore, the complexity of our method is 47.9% lower than that of YOLOv3 and the proposed algorithm can meet the requirement of the real-time application.

4 Conclusion

To enhance the usability of the traffic light recognition system in autonomous vehicles, this article employs CNN and ICFT to determine the coordinates and color for traffic lights. This paper improves the recognition accuracy and processing speed by combining detection and tracking. Experiment results first estimate the optimality of the presented CNN models and ICFT, which indicates that the Recall (0.842) and FPS (0.853) of the modified model are close to those of YOLOv3 (0.838 and 0.830) but FPS (17.92) is higher than 14.47. Additionally, IFCT is proved to achieve better performance of 4.393 Average Error, 0.567 Average IOU, and 0.344 AUC than single-channel feature tracking. The overall test further demonstrates the superiority of the proposed method, which means the proposed traffic lights recognition method could be adaptive to autonomous vehicles and achieve better performance.

Future work focuses on three perspectives: (1) Apply the related traffic light recognition system of this article into the system-on-chip and deploy on a real vehicle; (2) Communicate the traffic light information via 5G. By doing this, the efficiency and safety of autonomous vehicles in the network can be promoted by sharing the information; (3) Employ more advanced algorithms to improve the adaptability of CNN in different places. Reinforcement learning (RL) is a promising method to train the network in the way of unsupervised learning.

Funding This research was funded by National Natural Science Foundation of China (Grant No. 51605054), National Key Research and Development Program of China (SQ2020YFF0410766), Natural Science Foundation of Chongqing (cstc2020jcyj-msxmX0575), Chongqing Technology Innovation and application development project (cstc2020jscx-msxmX0109 and cstc2019jscx-fxydX0063), Fundamental Research Funds for the Central Universities (2020CDJ-LHZZ-042).

References

Bach M, Stumper D, Dietmayer K (2018) Deep convolutional traffic light recognition for automated driving. In: 2018 21st international conference on intelligent transportation systems (ITSC), IEEE, pp 851–858

Cai Z, Li Y, Gu M (2012) Real-time recognition system of traffic light in urban environment. In: 2012 IEEE symposium on computational intelligence for security and defence applications, IEEE, pp 1–6

Chen Z, Huang X (2016) Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking. IEEE Intell Transp Syst Mag 8(4):28–42

Chen J, Wang K, Bao H, Chen T (2019) A design of cooperative overtaking based on complex lane detection and collision risk estimation. IEEE Access 7:87951–87959

Chiang CC, Ho MC, Liao HS, Pratama A, Syu WC (2011) Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts. Int J Innov Comput Inf Control 7(12):6919–6934

Ci Y, Wu L, Zhao J, Sun Y, Zhang G (2019) V2i-based car-following modeling and simulation of signalized intersection. Phys A Stat Mech Appl 525:672–679

De Charette R, Nashashibi F (2009a) Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In: 2009 IEEE intelligent vehicles symposium, IEEE, pp 358–363

De Charette R, Nashashibi F (2009b) Traffic light recognition using image processing compared to learning processes. In: 2009 IEEE/RSJ international conference on intelligent robots and systems, IEEE, pp 333–338

Diaz-Cabrera M, Cerri P, Medici P (2015) Robust real-time traffic light detection and distance estimation using a single camera. Exp Syst Appl 42(8):3911–3923

Fairfield N, Urmson C (2011) Traffic light mapping and detection. In: 2011 IEEE international conference on robotics and automation, IEEE, pp 5421–5426

Greenhalgh J, Mirmehdi M (2012) Real-time detection and recognition of road traffic signs. IEEE Trans Intell Transp Syst 13(4):1498–1506

Hirabayashi M, Sujiwo A, Monrroy A, Kato S, Edahiro M (2019) Traffic light recognition using high-definition map features. Robot Auton Syst 111:62–72

Hosseinyalamdary S, Yilmaz A (2017) A Bayesian approach to traffic light detection and mapping. ISPRS J Photogr Remote Sensing 125:184–192

Jensen MB, Philipsen MP, Møgelmose A, Moeslund TB, Trivedi MM (2016) Vision for looking at traffic lights: Issues, survey, and perspectives. IEEE Trans Intell Transp Syst 17(7):1800–1815

John V, Yoneda K, Qi B, Liu Z, Mita S (2014) Traffic light recognition in varying illumination using deep learning and saliency map. In: 17th international IEEE conference on intelligent transportation systems (ITSC), IEEE, pp 2286–2291

John V, Yoneda K, Liu Z, Mita S (2015) Saliency map generation by the convolutional neural network for real-time traffic light

detection using template matching. IEEE Trans Comput Imaging 1(3):159–173
Kim HK, Shin YN, Sg Kuk, Park JH, Jung HY (2013) Night-time traffic light detection based on svm with geometric moment features. Int J Comput Inf Eng 7(4):472–475
Lee E, Kim D (2019) Accurate traffic light detection using deep neural network with focal regression loss. Image Vis Comput 87:24–36
Lee SH, Kim JH, Lim YJ, Lim J (2018) Traffic light detection and recognition based on haar-like features. In: 2018 international conference on electronics, information, and communication (ICEIC), IEEE, pp 1–4
Li X, Ma H, Wang X, Zhang X (2017) Traffic light recognition for complex scene with fusion detections. IEEE Trans Intell Transp Syst 19(1):199–208
Liu W, Li S, Lv J, Yu B, Zhou T, Yuan H, Zhao H (2016) Real-time traffic light recognition based on smartphone platforms. IEEE Trans Circuits Syst Video Technol 27(5):1118–1131
Possatti LC, Guidolini R, Cardoso VB, Berriel RF, Paixão TM, Badue C, De Souza AF, Oliveira-Santos T (2019) Traffic light recognition using deep learning and prior maps for autonomous cars. In: 2019 international joint conference on neural networks (IJCNN), IEEE, pp 1–8
Saini S, Nikhil S, Konda KR, Bharadwaj HS, Ganeshan N (2017) An efficient vision-based traffic light detection and state recognition for autonomous vehicles. In: 2017 IEEE intelligent vehicles symposium (IV), IEEE, pp 606–611
Shi Z, Zou Z, Zhang C (2015) Real-time traffic light detection with adaptive background suppression filter. IEEE Trans Intell Transp Syst 17(3):690–700
Wang K, Xiong Z (2016) Visual enhancement method for intelligent vehicle's safety based on brightness guide filtering algorithm thinking of the high tribological and attenuation effects. J Balk Tribol Assoc 22(2A):2021–2031
Wang JG, Zhou LB (2018) Traffic light recognition with high dynamic range imaging and deep learning. IEEE Trans Intell Transp Syst 20(4):1341–1352
Wang K, Huang Z, Zhong Z (2014) Simultaneous multi-vehicle detection and tracking framework with pavement constraints based on machine learning and particle filter algorithm. Chin J Mech Eng 27(6):1169–1177
Wang K, Huang X, Chen J, Cao C, Xiong Z, Chen L (2019) Forward and backward visual fusion approach to motion estimation with high robustness and low cost. Remote Sensing 11(18):2139
Wang K, Li G, Chen J, Long Y, Chen T, Chen L, Xia Q (2020a) The adaptability and challenges of autonomous vehicles to pedestrians in urban China. Accid Anal Prev 145:105692. https://doi.org/10.1016/j.aap.2020.105692
Wang K, Zhang S, Chen J, Ren F, Xiao L (2020b) A feature-supervised generative adversarial network for environmental monitoring during hazy days. Sci Total Environ 748:141445. https://doi.org/10.1016/j.scitotenv.2020.141445
Wang k, Ma S, Chen J, Lu J (2021) Approaches challenges and applications for deep visual odometry toward to complicated and emerging areas. IEEE Trans Cogn Dev Syst. https://doi.org/10.1109/TCDS.2020.3038898