Deep Traffic Light Detection for Self-driving Cars from a Large-scale Dataset

Jinkyu Kim[1,†], Hyunggi Cho[2], Myung Hwangbo[2], Jaehyung Choi[2],
John Canny[1], and Youngwook Paul Kwon[2,3]

***Abstract*— Traffic lights perception problem is one of the key challenges for autonomous vehicle controllers in urban areas. While a number of approaches for traffic light detection have been proposed, these methods often require a prior knowledge of map and/or show high false positive rates. Recent successes suggest that deep neural networks will be widely used in self-driving cars, but current public datasets do not provide sufficient amount of labels for training such large deep neural networks. In this paper, we developed a two-step computational method that can detect traffic lights from images in a real-time manner. The first step exploits a deep neural object detection architecture to fine true traffic light candidates. In the second step, a point-based reward system is used to eliminate false traffic lights out of the candidates. To evaluate the proposed approach, we collected a human-annotated large-scale traffic lights dataset (over 60 hours). We also designed a real-world experiment with an instrumented self-driving vehicle and observed that the proposed method was able to handle false traffic lights substantially better compared with the baseline considered.**

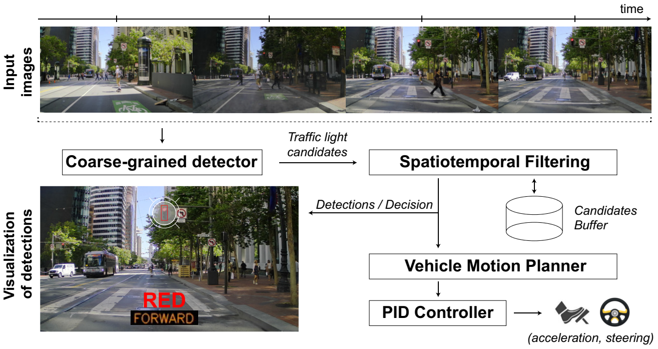

Fig. 1: Our model detects traffic lights, *e.g.*, a red circle, from an input raw image at each timestep. Our model consists of two major steps: (1) coarse-grained detector that utilizes deep neural object detection architecture and is tuned to discover as many true traffic lights as possible. (2) the spatiotemporal filtering step that eliminates false traffic lights with respect to features extracted from both spatial and temporal domains. The output of our proposed detector is then fed into the vehicle motion planner and the PID controller that computes corresponding acceleration and steering angle commands.

I. Introduction

Self-driving vehicle control has recently made remarkable progress. These controllers involve a variety of sophisticated algorithms for perception, behavioral/motion planner, and dynamics controllers. Despite their recent success in some driving scenarios (*i.e.*, highway driving), there still remain new challenges for urban driving that involves more complex driving scenarios that need interaction with traffic controls, vehicles, pedestrians, etc. Especially, traffic lights pose a challenging (computer vision) problem when subject to varying lighting, view distances, and weather conditions. Though its importance for automated driving in urban areas, conventional approaches showed insufficient reliability and robustness enough to be used in autonomous systems in the urban environment without utilizing prior knowledge.

Recent successes suggest that deep neural networks will be widely used in self-driving cars, especially for a perception part. Behrendt *et al.* [1] utilized the "You Only Look Once" (YOLO) network architecture followed by a small classification convolutional network to detect traffic lights. For obtaining a reliable and robust performance from such a large deep neural network, a large amount of dataset is strongly required to provide a large variation in environmental conditions. However, the current publicly available datasets show lacks such variation. For example, the VIVA challenge [2] for traffic lights only provide 44 minutes of data and the Bosch Small Traffic Lights Dataset [1] provides only about 5,000 images (less than 3 hours), which is insufficient from the view of the conventional way of training such large deep neural networks. We, therefore, create a large dataset, which provides over 60 hours of driving images that cover diverse driving conditions (*i.e.*, lighting and weather). Thus, we argue this dataset will be ideal for further traffic light detection studies.

Collecting a large-scale dataset is only part of a story. Reliable traffic light detector strongly requires low false negative (or discovering as many true traffic lights as possible) and false positive (or eliminating false traffic lights) rates, while maintaining a high detection accuracy. Here, we propose a new computational method for traffic light detection, which consists of two major steps: (1) coarse-grained traffic light detection and (2) spatiotemporal filtering of the detected traffic lights. The first step considers individual images and collects traffic light candidates using a deep neural object detection architecture. The focus of this step is to reduce false

[1] Jinkyu Kim and John Canny are with the Department of Electrical Engineering and Computer Sciences, UC Berkeley, CA 94720, USA.

[2] Hyunggi Cho, Myung Hwangbo, Jaehyung Choi, and Youngwook Paul Kwon are with Phantom AI Inc., Burlingame, CA 94010, USA.

[3] Youngwook Paul Kwon is with the Department of Mechanical Engineering Engineering, UC Berkeley, CA 94720, USA.

[†] Work was done while the author was at Phantom AI Inc. for internship.
Correspondence: `paulkwon@phantom.ai`

negatives (FNs) or to discover as many true traffic lights as possible. The second step is then to eliminate false positives (FPs) by considering spatial and temporal characteristics of traffic lights. To distinguish true and false traffic lights, we propose a point-based reward system where each detected traffic lights earn rewards and the final decision is made based on these rewards. To demonstrate the effectiveness of applying the proposed method to self-driving vehicles, we test with an instrumented vehicle and successfully drive 6 kilometers on city streets in the San Francisco Bay Area, California, USA.

Our contributions can be summarized as follows:

1) We propose a new computational method for accurately detecting traffic lights from a raw input image in a real-time manner.
2) We generated a large-scale traffic lights dataset with over 71,771 images (over 60 hours) with human annotated bounding boxes.
3) We demonstrate the effectiveness of applying our proposed approach by conducting a real-world experiment (driving over 6 kilometers including 17 intersections with traffic lights) with an instrumented vehicle.

II. Related Work

A number of approaches have been proposed for traffic light detection and classification for autonomous vehicles and/or for driver assistance systems to navigate in urban areas. Most of these approaches utilized a supervised learning approach with human-designated features. This literature is too wide to survey here. For a thorough review of this literature, see [3].

These approaches usually depend on strong assumptions: (1) they are based on recognizing human-designated features, which generally require demanding parameter tuning for a balanced performance. (2) Some require the detailed maps that provide prior knowledge about the specific locations of all installed traffic lights, which but demand high costs in building such a map. Furthermore, other issues may include: (i) color-tone shifting due to changes in atmospheric conditions and nearby light sources. (ii) Occlusion by other objects. (iii) High false positive rates caused by brake lights, reflections, and pedestrian crossing lights. and (iv) Inconsistent traffic light lamps due to dirt, defects, over-saturation of the camera (especially during night-time).

Recent approaches suggest that deep neural networks can be successfully used for the traffic light detection task. Weber *et al*. [4] utilized a 7-layer convolutional neural network to predict the multi-class probability map followed by bounding box regression. Behrendt *et al*. [1] used the "You Only Look Once" (YOLO) network architecture to detect traffic lights, and utilized a tiny convolutional neural network to classify the categories of each detected traffic lights. They also provide a dataset, called the Bosch Small Traffic Lights Dataset, which provides approximately 5,000 images (2.8 hours of driving) and 8,334 annotations. Despite its potential, training these deep neural networks requires a large amount of annotated dataset to train a reliable detector that can address challenges in traffic light detection task. Though there exist some other open-sources of traffic light annotations, these datasets are still insufficient for training deep neural networks in terms of diversity of scenes, quality of annotations, and their limited volume. For example, the VIVA challenge dataset [2] only provides approximately 40 minutes of scenes, while the Bosch [1] Small Traffic Lights dataset provides less than 3 hours (5,000 images). We, therefore, collect our own dataset, which provides over 60 hours (over 71,771 images) of driving images that cover diverse driving conditions (*i.e*., day vs. night and sunny vs. raining). Thus, we argue this dataset will be ideal for traffic light detection studies.

III. Deep Traffic Light Detection

Here, we propose a method that accurately and reliably detects traffic lights from a stream of images captured by a front-view dash-cam attached to the windshield. As we depicted in Figure 2, the proposed method contains two major steps: (1) coarse-grained traffic light detector (Section III-B) and (2) spatiotemporal filtering (Section III-D) of the traffic lights candidates. In the first step (coarse-grained detector), traffic light candidates from each image are collected by utilizing a deep neural object detection architecture. The main focus of this step is to discover the true traffic lights as many as possible (*i.e*., reducing the number of false negatives). Thus, it is possible that the traffic light candidate collection may contain false positives. In the second step (spatiotemporal filtering), we eliminate such erroneously detected traffic lights by simultaneously considering other traffic lights over time and space. To distinguish between true and false traffic lights, we use a point-based reward system where each detected traffic lights earn rewards with respect to features extracted from both spatial and temporal domains.

A. Preprocessing

We use an input image that is resized to 288×512×3 with bilinear interpolation algorithm, hence to reduce computational burdens for a real-time detection. For the images with different aspect ratios, we cropped the height to match the ratio. Following a common practice in image classification tasks, we subtracted the mean RGB value to achieve zero-centered inputs, which are originally in different scales. Note that our dataset contains images where the camera gains are automatically calibrated to obtain high-quality images. During the testing process, we also used a cropped image in the center part of the image, where traffic lights are commonly observed in that area. Thus, a batch of two images (*i.e*., whole and cropped images) are fed into our detector.

B. Coarse-grained Traffic Light Detection

Traffic light detector strongly requires showing reliable performance in real-time and working for both small (*i.e*., 3x9 pixels) and large objects with low false positive and low false negative rates, while maintaining a high detection accuracy. For example, a false red traffic light will lead the autonomous vehicle to abruptly stop while driving, while

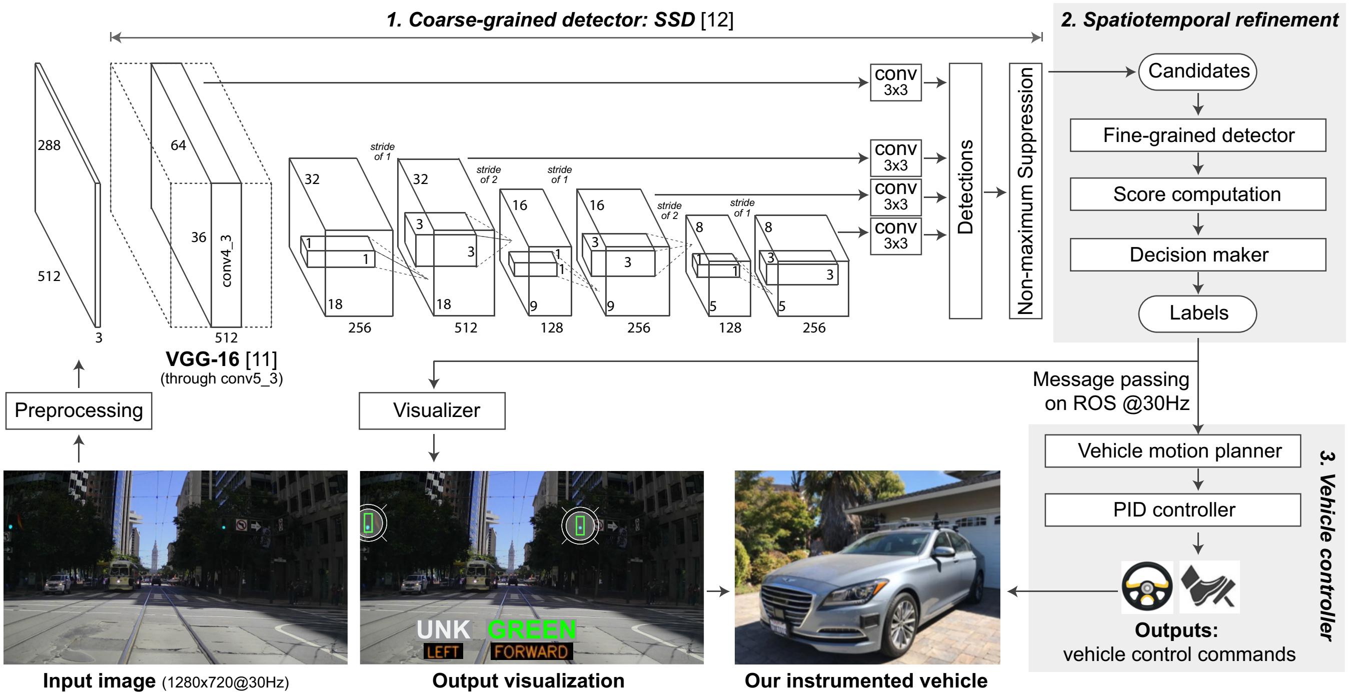

Fig. 2: An overview of our proposed model. It can be understood in three parts: (i) a coarse-grained detector that utilizes the deep neural perception network architecture called SSD (Single-Shot multi-box Detector [5]), (ii) a spatiotemporal filtering, and (iii) a vehicle controller. To demonstrate the feasibility of applying our model to self-driving cars, we use an instrumented autonomous vehicle which uses the output of our model as an input to control its dynamics.

a missed red light will cause the vehicle to go through an intersection originally with red lights in its course of driving.

In this coarse-grained traffic light detection step, we focus to reduce false negative (FN) rates or to collect as many true traffic lights as possible. We utilize the Single-Shot multi-box Detector (SSD) [5] that has been shown to be an effective tool for an object detection task. Note that we use the SSD architecture that has shown improved detection accuracy in other benchmarks than YOLO network architecture, which was utilized in the existing work by Behrendt *et al.* [1]. More modern architecture, such as Mask R-CNN [6], may provide better detection accuracy, but we leave this comparison for future work. The SSD model is based on a convolutional network and takes the whole image as an input and predicts a fixed-size collection of bounding boxes and corresponding confident scores for the presence of object instances in those boxes. The final detections are then produced followed by a non-maximum suppression step – all detection boxes are sorted on the basis of their predicted scores, and the detections with maximum score is then selected, while other detections with a significant overlap are suppressed. As we described in Figure 2, we use a standard VGG-16 network architecture [7] as a base convolutional network, which is pre-trained on ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [8]. Auxiliary structures – convolutional predictor and the additional convolutional feature extractor – are used following the work by Liu *et al.* [5].

1) Training objective: The loss function $\mathcal{L}$ (= $\mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{conf}}$) is a weighted sum of two types of loss: (1) the localization loss $\mathcal{L}_{\text{loc}}$ measures a Smooth L1 loss between the predicted and the ground-truth bounding box in a feature space. (2) The confidence loss $\mathcal{L}_{\text{conf}}$ is a softmax loss over multiple classes confidences. For more rigorous details, refer to [5]

2) Data augmentation: To train a robust detector to various object sizes, we use random cropping (the size of each sampled image is [0.5, 1] of the original image size with fixed aspect ratio) and flipping to yield consistent improvement. Following [5], we also sample an image so that the minimum jaccard overlap with the objects is $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Note that each sampled image is then resized to a fixed size followed by photometric distortions with respect to brightness, contrast, and saturation.

C. Characterizing Traffic Lights

According to our analysis, the traffic lights appearing in the detection pipeline possess the following characteristics:

(C.1) As the confidence of a traffic light candidate decreases, so does the possibility of this being a true traffic light.

(C.2) The possibility of a traffic light candidate being true increases if the traffic light candidate is detected again in next timestep at almost the same location.

(C.3) If multiple traffic lights of the same category (*i.e.*, red, yellow, and green) are detected in a scene, then they are usually true traffic lights differently located (*i.e.*, multiple traffic lights are installed at an intersection).

(C.4) Traffic lights shall be located following the governmental guideline (*i.e.*, at least one of the signal faces shall be located at an intersection mounted on the mast arm.), hence the possibility of a traffic light candidate being true increases as its location gets close to the usual.

As is evident above, examining traffic lights individually is not sufficient, and multiple traffic light candidates over space and time should be considered simultaneously.

D. Spatiotemporal Filtering

1) Fine-grained detector: Recall from Section III-A, we rescaled images by 40% to reduce the computational burdens for a real-time system. We observe that the classification performance of our coarse-grained detector slightly decreases as we have smaller traffic lights (*i.e.*, seen from a farther distance). Thus, we utilize an additional small classification network, called fine-grained detector, that has a high-resolution input. All bounding boxes from the coarse-grained detector are cropped and rescaled to 100×100 pixels, they are then fed into the fine-grained detector. For training, we collect image patches that are cropped and rescaled from the ground-truth dataset. Overall, we collect 24,991 and 6,248 patches for training and validation, respectively.

2) Score function: According to our traffic light characterization (see C.1–C.4), we need to examine multiple traffic light candidates simultaneously for accurate traffic light status recognition. In addition, we set the confidence threshold value of the coarse-grained detector so as to minimize the number of FNs (*i.e.*, the true traffic lights that are erroneously left undetected). Consequently, it is likely that the traffic lights detected in the previous step contain false traffic lights that further need to be filtered out. In this spatiotemporal filtering step, we seek to resolve these issues using a point-based reward system where each detected traffic lights earn points with respect to the following characterizations:

(S.1) Each traffic light candidate has its own score for being true, and its score is accumulated in the next timestep if detected again under our matching criterion (*i.e.*, euclidean distance between centers of each candidate).

(S.2) Every traffic light candidates from coarse-grained detector earn a reward R at each timestep.

(S.3) Scores are discounted by a pre-specified discount rate γ at each timestep.

Concretely, the score function $s_j(t)$ for a candidate j is defined as follows:

$$s_j(t) = \min\big(S_{\text{MAX}}, Rc_j(t) + \gamma s_j(t-1)\big) \qquad (1)$$

where $c_j(t) \in [0,1]$ is the confidence value computed by the coarse-grained detector. A maximum score is set to S_{MAX}.

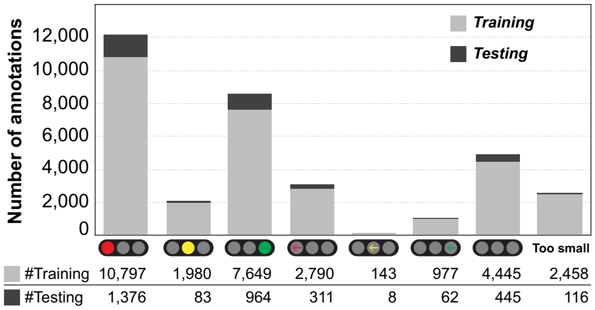

Fig. 3: Traffic lights annotation statistics.

3) Decision: The output of this step is a tuple of the current traffic light status. For each type k of traffic light signals (*i.e.*, k ∈{turning left, going forward, and turning right}) and each traffic light status (*i.e.*, unknown, red, yellow, and green), we accumulate scores over traffic light candidates and output the status of the maximum score.

$$o_k(t) = argmax_{i\in\{\text{red, yellow, green, unknown}\}} \sum_j \mathbb{1}(i,j)s_j(t) \qquad (2)$$

where $\mathbb{1}(i,j)$ is an indicator function that is 1 if j-th candidate has the same status as i, otherwise 0.

IV. Traffic Light Detection Dataset

In order to effectively train and evaluate a deep neural perception approach, we have collected a large-scale traffic lights dataset. Our dataset contains RGB color images captured by a dashcam mounted behind the front mirror of the vehicle. Each image has the resolution of 1280 × 720 pixels. We provide the dataset statistics in Table I. Our dataset is composed of over 60 hours of driving taken in diverse driving conditions, *e.g.*, day/night, city/residential ares, etc. We have collected 71,771 images mainly in the San Francisco Bay Area in California, USA. To avoid high similarity between images, we sample images at every 3 seconds. Overall, 34,604 are labeled, the minimum size of labeled traffic lights is approximately 3 (width)×9 (height) pixels. We also introduce a training and a test set, containing 64,607 and 7,164 images, respectively. In Figure 3, we illustrate the distribution of the different traffic light states, which have eight categories: off, too small to annotate, green (circle),

TABLE I: Dataset details with the comparison to other publicly available datasets: the VIVA Challenge for traffic lights [2] and the Bosch Small Traffic Lights dataset [1].

	VIVA [2]		Bosch [1]		Ours	
	Training	Testing	Training	Testing	Training	Testing
#Images	20,526	22,481	5,093	8,334	64,607	7,164
FPS	16	16	1/2	15.6	1/3	1/3
#Annotations	54,161	64,170	10,756	13,493	31,239	3,365
#Hours	≈ 22min	≈ 22min	≈ 2.8 hours	≈ 8.9min	≈ 53 hours	≈ 6 hours
Image Res.	1280×960		1280×720		1280×720	
Location	San Diego, USA		The SF Bay Area, USA		The SF Bay Area, USA	

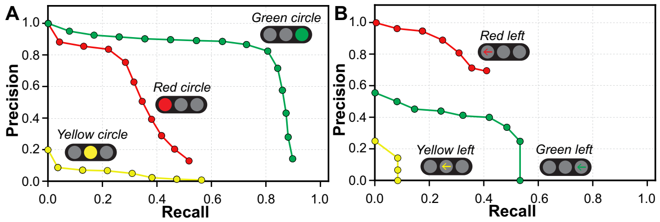

Fig. 4: Performance evaluation of our coarse-grained detector in terms of two widely-used metrics: precision and recall. For red, green, and yellow circles, see A. For others, see B.

red (circle), yellow (circle), green (left-turn), red (left-turn), and yellow (left-turn).

V. Result and Discussion

A. Training and Evaluation Details

For training a coarse-grained detection model, we use the stochastic gradient algorithm (SGD). Unless stated, we use default hyper-parameters following the work by Liu [5]. Our model took less than 3 days to train on three NVIDIA Titan X Pascal GPUs. Our implementation is based on a deep learning framework called Caffe [9].

B. Quantitative Analysis

As shown in Figure 4, we first measured the classification performance of our coarse-grained traffic light detector in terms of recall and precision. We tested with different threshold values in [0, 1]. Note that a good classifier will have higher precision and recall values. The coarse-grained detector seeks to minimize the number of FNs (or undetected true traffic lights) to maintain high recall – suggesting that few true traffic lights went undetected by using this method. In the cases tested, maintaining a high recall increases the number of FPs (or detected false traffic lights), which support the need to use the refinement step. The numbers of training example for the yellow circle and left lights are smaller than other colors, and we observe the classifier shows poor classification performance for the yellow circle and left lights. It would also worth exploring the use of other types of more expressive neural networks, which may give a performance improvement over our network configuration [10]. However, exploration of other architectures would be out of our scope.

Recall from Section III-D, we use a fine-grained detector that further examines the traffic light candidates by using an additional small neural network with high-resolution inputs. Table II shows the classification performance with and without the fine-grained detector. In the cases tested except for two classes (*i.e.*, red circle and red left), using a fine-grained detector resulted in higher classification performance in F-measure values. The F-measure is 3.44 - 17.03% higher as compared to the coarse-grained detector only.

C. Real-world Experiments

To demonstrate the feasibility of applying our traffic light detector for a real self-driving car (see Figure 5 (A)), we utilize an instrumented vehicle equipped with the following specifications:

(V.1) Vehicle: Hyundai Genesis G80
(V.2) Sensors: 2×Velodyne LiDAR sensors, 4×Radar sensors, and 1×video camera (resolution: 1280x720 pixels, frame rate: 10Hz, field of view (FOV): 60 degrees).
(V.3) PC: Intel i7 Quad-core processor, 16GB DDR3 memory, a 1TB SSD, a Titan X Pascal GPU, and Linux OS.

We use the Robot Operating System (ROS) for synchronizing the sensor data and for the message passing of perception, motion planning, and control nodes. At each timestep, the sensory data is consumed from raw sensors (camera, LiDAR, and Radar) and processed by a collection of ROS nodes that all communicate with each other. We use the PID controller for our control node, hence the final output control commands to the throttle, brake, and steering wheel are provided through our drive-by-wire units. We depict major steps in Figure 2.

We test our proposed traffic light detector with an instrumented vehicle on public roads in Bay Area, California, USA. As shown in Figure 5 (C-E), the test runs were performed in an unseen pre-specified testing route (over 6 kilometers), and the test scenario comprised the following features: (1) The vehicle traversed 17 traffic light-controlled intersections where the vehicle will follow the rules to stop on red and go on green. (2) Different lighting conditions (day vs. night), weather (rainy vs. sunny), and different road traffic congestion levels are tested. We build a visualization to show which traffic lights are detected and its the final decision. We provide examples from our visualizer during the on-road driving test. Our real-world experiments support that the proposed traffic light detector can be successfully operated in a real-time manner.

VI. Conclusion

We described a traffic light detection model for self-driving vehicles by incorporating a state-of-the-art deep neural object detection architecture and a spatiotemporal filtering with a point-based reward system. We showed that (i) incorporation of a spatiotemporal filtering improves traffic light detection performance by reducing false positive rates,

TABLE II: The effect of using fine-grained detector (see Section III-D) is evaluated in terms of precision, recall, and F-measure. Scores are reported in percentage (%).

Classes	*without* Fine-grained detector			*with* Fine-grained detector		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Red circle	70.20	29.70	41.74	68.00	29.50	41.15
Yellow circle	2.20	40.00	4.17	6.00	30.90	10.05
Green circle	37.60	87.80	52.65	60.80	81.60	69.68
Red left	84.20	27.70	41.69	62.40	29.00	39.60
Yellow left	14.30	8.30	10.50	66.70	16.70	26.71
Green left	32.30	50.00	39.25	36.40	51.60	42.69

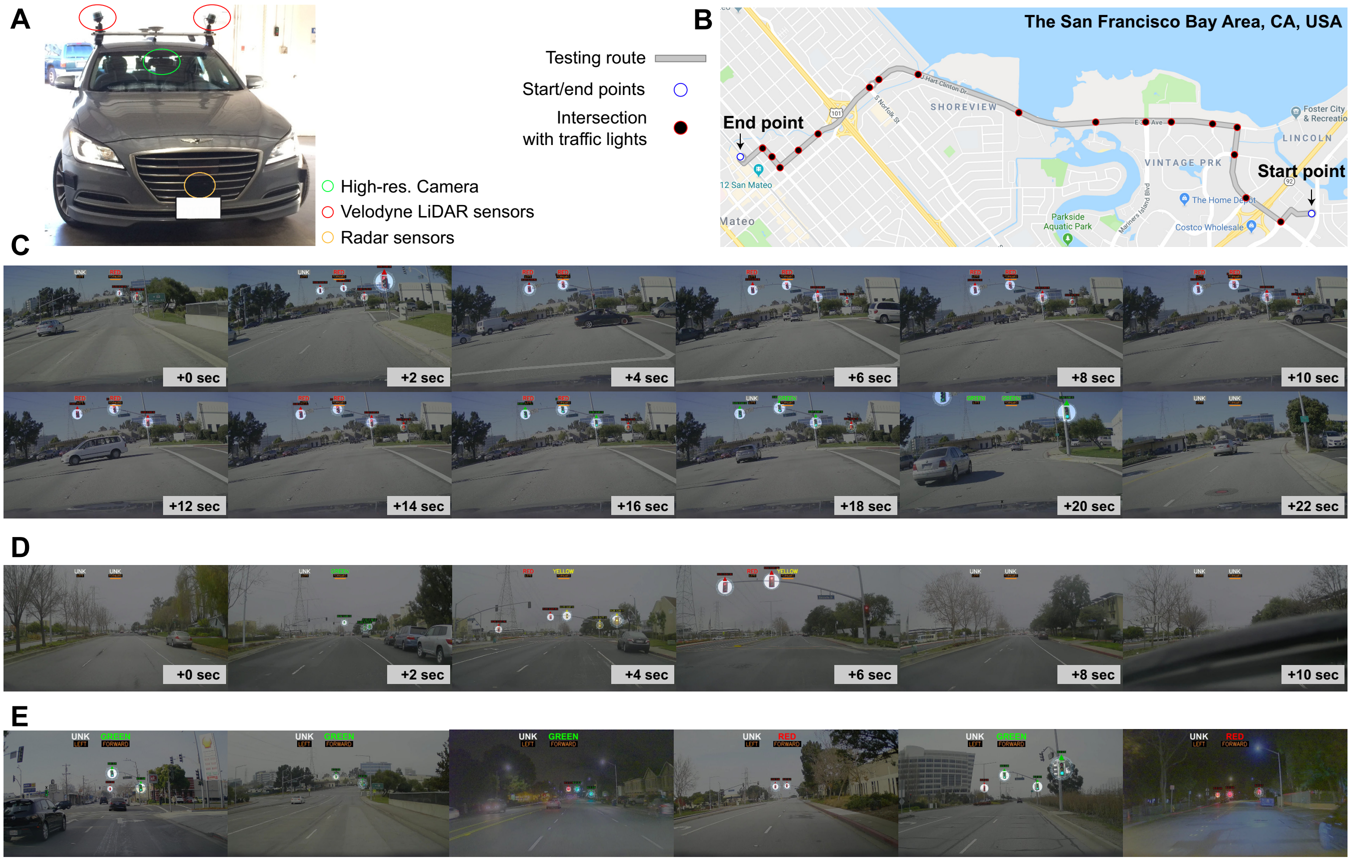

Fig. 5: (A) Our instrumented vehicle with sensor layout for real-world evaluation of the proposed traffic light detection pipeline. (B) Our testing route (in the San Francisco Bay Area, CA, USA) for the real-world experiment. This route is over 6 kilometers including 17 intersections with traffic lights installed. *Map credit*: Google Maps. (C-D) Visualizations for traffic light detection over time. Unseen consecutive input images are sampled at every 2 seconds (see bottom-right). Our final decisions are depicted on the top of each figure, while detected traffic lights are highlighted by a white circle. (E) Additional examples of detected traffic lights during the test scenario.

(ii) our model can be operated in a real-time manner and can be applied to real-world self-driving cars, (iii) our large-scale traffic lights dataset provides a diverse variations and allows us to train and evaluate a large deep neural network.

ACKNOWLEDGMENT

The authors would like to thank Jihyun Yoon and Chankyu Lee for their helpful comments and the anonymous reviewers of this paper. This work was supported in part by Samsung Scholarship.

REFERENCES

[1] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1370–1377.

[2] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Learning based traffic light detection: Evaluation on challenging dataset," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[3] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: Issues, survey, and perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, 2016.

[4] M. Weber, P. Wolf, and J. M. Zöllner, "Deeptlr: A single deep convolutional network for detection and classification of traffic lights," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 342–348.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 21–37.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[8] 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," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[10] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.