Traffic signal auto detection on CARLA for object detection training

Team member: Minhwan Oh (329006983)

YouTube link: https://youtu.be/GfEp5DsPx5w
GitHub link: https://github.com/Bluetea724/Carla_Traffic_Lights_Auto_Detection
Dataset and weights file for Yolov3 : https://drive.google.com/drive/folders/1Inju5YEVRLnQ6I-fPgB81yhJ2TZkl6Su?usp=sharing
(I can't upload image files for dataset and weight file in GitHub because both sizes are over 25MB, so I post google drive instead)

Introduction: Traffic light detection is one of an essential part of autonomous driving for a dataset because high accuracy of object detection for autonomous driving is possible with a proper dataset not only cars, pedestrians, bicycles, but also traffic signs and traffic lights [1]. For this project, I use Carla [2], which is an open-source simulator for autonomous driving. It can remove any other noises and can implement a virtual world that is very close to the real world. The main ideas for this project are shaped-based and color-based methods.

Regarding shaped-based detection, it could use a prior map for detection [3], and cascade classifiers [4], Hough transform. Finally, I decided to use using Hough transform using OpenCV [5] since traffic signals are typically circle. Region of interest (ROI) was also considered [6,7,8] for an additional filter for traffic light detection. Still, I decided not to use it because the view of simulation for the experiment could not be appropriate for using ROI. Instead, a color algorithm was added due to improving accuracy and class classification [9,10].

After detecting traffic signals, this program automatically makes labeling based on the center of detected objects as a Yolo's dataset [11]. Therefore, this project can test as unsupervised learning with shaped and color-based algorithm, and supervised learning with Yolo [11].

Environment:
- OS version: Ubuntu 18.04.4
- CARLA version: 0.9.6
- Python version: 2.7
- Yolo version: Yolov3 tiny version, Darknet
- OpenCV version: 3.4.2
- CARLA Map: Town5
- CARLA Weather: Cloudy Sunset
- Basic code: Manual_control.py in CARLA

Main Idea: Assumes that traffic lights should be close to circle if we look at traffic lights straight forward. Using HoughCircles [5] function, if there are some circles whose radius of image pixel is between 3 and 10, then it becomes a traffic light candidate. If the candidate data is matched with a color condition, it will be Green or Yellow traffic light.

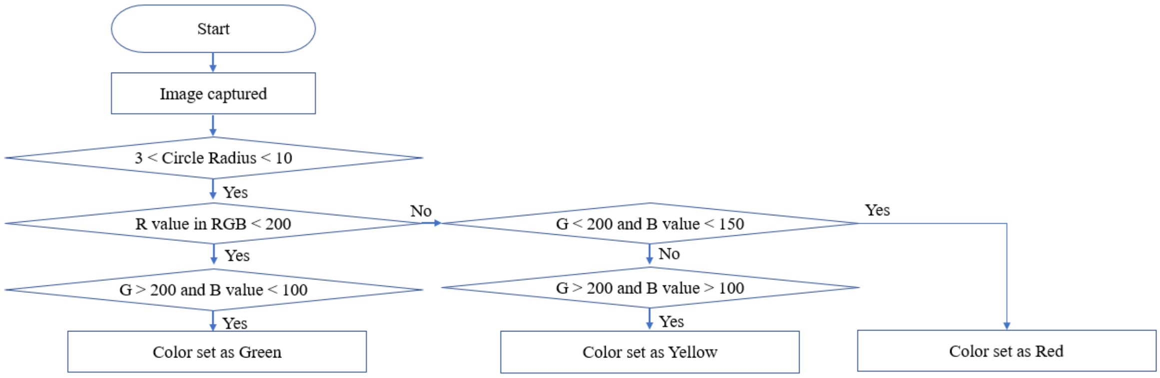

Figure 1. Flow chart for traffic lights detection

During image capture and circle radius check, handling houghcircles function in OpenCV is a huge challenge because if the houghcircles function is too robust to detect many shapes like circles, it can't get a correct result. Therefore, finding a proper canny edge and center detection parameters were significant. I tried at least over 300 times tests because I needed to find which OpenCV functions and settings were best for this case, and had to find optimal parameters canny edge, center detection parameters, and circle radius.

Regrading auto labeling, I used the center of circles' information and radius data from HoughCircles function. The label's area is proportional to the radius.

Experiments:

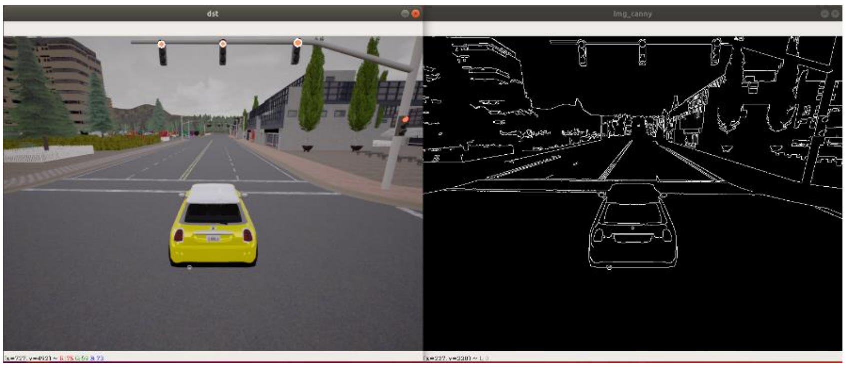

Figure 2. Detecting traffic lights (Left), Finding edge information (Right)

To check whether houghcircles function's parameters are appropriate, I used canny function in OpenCV, which detects the edge of shape [5]. Houghcircles function uses canny function inside, but it is very difficult whether the edge parameter is right for circle detection. Using the canny function, I could find an appropriate edge and center detection parameter above figure. After finding circles as traffic lights candidates, they were checked by color-based algorithms to verify whether it is a correct traffic light or not. And then, it draws white color circles on traffic lights in Figure 2.

Results:

Correct labels	993
Wrong labels	14
Precision rate	98.6%

Table 1. Test result for data collection

During a data collection test, the precision rate per image file was very high as 98.6% that only 13 out of 1003 images had wrong circle information. However, a recall rate was low because all traffic signals that were not captured like circles were considered as not traffic light.

Figure 3. Traffic lights recognition case1

As seen in Figure3, red traffic lights were not recognized because they were too small to be detected. Also, Figure 2. One red light in the left picture was not recognized.

Figure 4. Traffic lights recognition case2

I analyzed why there were wrong detection cases. One of the wrong cases was it detected traffic signal as traffic light because its size was similar as a traffic light and it was close to trees, so it was detected as green color. The success rate of Yolov3-tiny using collected data is approximately 87%.

The reason the success rate of Yolov3-tiny was lower than auto-detection was recall coverage. Because of supervised learning, it can detect wide coverage of traffic lights. Therefore, as you can see in uploaded YouTube video, it can detect additional traffic lights even though it didn't detect in automatic detection.

Merits: Since its precision rate is high, labeling workload for a Yolo dataset could reduce a lot. If it saves image files, that means at least one traffic signal is included in the data. And, it could also be used as traffic signal detection to change some parameters. Additionally, this algorithm can be used as an unsupervised learning algorithm.

Check with proposal goal:

Goal in Proposal and Literature Survey	Final result
You only look once (Yolo) is a state-of-the-art, real-time object detection system. The strong points of Yolo are fast and accurate. Currently, if Carla user wants to use Yolo, they have to capture and label themselves. Due to environment's limitation, it can't train during Carla simulation. Therefore, after capturing data, it will train, then test it.	It could capture and label based on shaped and color-algorithm, then could train successfully. And, it could also use it as Yolo dataset.

Table 2. Target and result comparison

References:

[1] A. Dominguez-Sanchez, S. Orts-Escolano, J. Garcia-Rodriguez and M. Cazorla, "A New Dataset and Performance Evaluation of a Region-based CNN for Urban Object Detection," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018, pp. 1-8.

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. ´ CARLA: An open urban driving simulator. In Conference on Robot Learning (CoRL), 2017.

[3] N. Fairfield, and C. Umson, "Traffic Light Mapping and Detection", Google, in IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 5421–5426.

[4] F. Lindner, U. Kressel, S. Kaelberer, "Robust recognition of traffic signals", in: Proceedings of IEEE Intelligent Vehicles
Symposium, June 14–17, 2004, pp. 49–53.

[5] G. Bradski and A. Kaehler, Learning OpenCV: Comput. Vision with the OpenCV Library. Sebastopol, CA, USA: O'Reilly Media, 2008.

[6] T. M. Hoang, S. H. Nam and K. R. Park, "Enhanced Detection and Recognition of Road Markings Based on Adaptive Region of Interest and Deep Learning," in IEEE Access, vol. 7, pp. 109817-109832, 2019.

[7] R. Nabati and H. Qi, "RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles," 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 2019, pp. 3093-3097.

[8] D. Wang, C. Devin, Q. Cai, F. Yu and T. Darrell, "Deep Object-Centric Policies for Autonomous Driving," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 8853-8859.

[9] G. Siogkas , E. Skodras , E. Dermatas, "Traffic Lights detection in Adverse Conditions using Color", Symmetry andSpatiotemporal Information, Conference: in International Conference on Computer Vision Theory and Applications (VISAPP), 2012.

[10] Diaz-Cabrera et al, "Suspended Traffic Lights detection and distance estimation using color features", 15th IEEE Conference on Intelligent Transportation Systems, 2012.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). jun 2016, pp. 779–788, IEEE.