Real-time Signal Light Detection based on Yolov5 for Railway

Wentao Liu, Zhangyu Wang, Bin Zhou*, Songyue Yang, Ziren Gong
Beihang University Transportation Science and Engineering Beijing, China

*Corresponding author: zyliuwt@buaa.edu.cn

Abstract. To improve the safety and efficiency of train operation, autonomous driving train have developed rapidly in recent years. Among them, the signal detection is one of the most basic functions. However, due to the small size of signal light and the complicated of the railway environment, the signal detection is still a huge problem. The existing methods, such as the approach based on Hough circle transformation, are hard to meet the practical application requirements. In this paper, a real time railway signal lights detection based on Yolov5 is introduced. And a lot of experiments were conducted to prove the effectiveness of the proposed method. The experimental results show that the proposed method achieved 0.972 for both average recall rate and average accuracy rate. Besides, the detection speed of the proposed method reached astonishing 100FPS. Overall, the detection speed and accuracy both meet the practical application requirements.

Keywords: Autonomous driving; Perception; Signal light detection; Deep learning.

1. Introduction
In recent years, more and more cities have begun to build railways to ease the urban traffic congestion. However, with the increasing number of subways, the risk of traffic accidents from railways is also growing. In order to keep the safety of subways operating, the automatic driving system to assist railway drivers is needed.

Signal lights detection is an important part in the assisted driving perception system. The main purpose of signal lights detection is to detect them correctly and quickly during the train operation. At present, signal lights are mainly observed by the drivers, it may lead to serious accidents due to the driver’s inattention.

Railway signal lights detection is similar to intersection signal lights detection in vehicle automatic driving. Both of them are red and green in color and round in shape. Therefore, we can learn from the way of intersection signal lights perception to perceive railway signal lights. Due to the color characteristic of signal lights, we generally use CV to extract signal lights. The traditional way depends on its color and shape, using feature matching, color matching and other methods to extract signal lights.

In recent years, deep learning has attracted most attentions. Along with the development of hardware, compute power required by deep learning has been reached. More and more researchers are trying to solve problems with deep learning. It has been used in various fields, such as autonomous driving, medicine and finance, etc. Railway signal lights detection is a standard small object detection problem in deep learning. Many predecessors have proposed numerous object detection networks for small that.
Therefore, this paper proposes a railway signal lights detection method based on Yolov5 object detection model. According to the experiment results, the effect of signal light detection is better and can be applied in engineering. The following Chapter 2 introduces the related work of other scholars in signal light detection. Chapter 3 introduces the Yolov5 network model, Chapter 4 conducts experiments and shows the experiment results, and Chapter 5 is final conclusion.

2. Related works
In the field of traffic signal lights detection, predecessors have proposed lots of methods, which can be summarized into three categories: 1). Extracting and classifying signal lights based on traditional CV [2]; 2). Extracting the ROI of the signal lights based on traditional CV, and classify it through the classification neural network [6]; 3). Extracting and classifying traffic lights based on neural network [3].

The extraction of signal lights based on traditional CV is mainly based on color and shape. Literature [1] extracts signal lights area through traditional image segmentation. Position and color of signal light are determined according to segmentation result. Literature [19] used another multi-scale, multi-stage detector based on multi-channel features to extract and classify traffic lights.

This kind of method usually has better detection results in specific environment, but the robustness is always poor.

The second method is used more currently. Such as:

The literature [9] extracts the signal lights through traditional CV. First extracting a single channel, turning image into a grayscale image, then getting the ROI of signal lights. Finally taking ROI into the classification neural network author built.

And finally get the signal classification result. Literature [7] builds the extraction network MBBNet and the classification network Pre-RTTLD. He re-input signal light features area into the Pre-RTTLD, and finally gets the detection results.

Compared to the first method, the second method has a little improvement in robustness, especially the algorithm that relies on neural network to extract signal lights. But the detection speed is slow and cannot be real-time.

The third method which was proposed in few years is studied by most researchers. Such as:

Literature [3] Constructs a neural network using focal regression loss function and res-net101 feature extraction backbone. At the same time, in order to fit the actual situation, only the upper part of image is taken for detection, which corresponds to the characteristic that traffic lights are generally located above the images.

Literature [8] uses Facebook's open source RetinaNet network. Using BOSH's traffic signal dataset to train, and achieving good results.

The third method is faster than the first two, and some can be deployed on the control machine for real-time detection. For the latest model, the detection accuracy greatly exceeds the first two methods.

3. Methodology

3.1. System Structure
This paper adopts a similar method with Literature [8], using the Yolov5 network to extract and classify railway signal lights. As shown in Figure 1, We put the RGB images whose sizes are all the same 1280×720 into the neural network, where images will go through backbone and neck parts of the neural network. Then output the results which include target's position information, category information and confidence through three different sizes of detection head. The detections were carried out in feature maps of 52×52, 26×26 and 13×13 sizes.
3.2. Signal light detection network

Yolov5 is a typical one-stage object detection network, composed of three components modules, Backbone, Neck, and Head.

3.2.1. Backbone. The main function of Backbone is to extract feature map of image. It inherits Yolov3 [11] network and adds modifications, as shown in Figure 2. Yolov3 uses the darknet53 network, Yolov5 keeps partial structure and modifies the residual network. Learning from the network structure of CSPNet, it mixes convolutional layers and X residual components.

Introduction of Backbone's modules:
1. CBL: It is composed of Conv+Bn+Leaky_relu activation function, which can realizes feature map extraction of images.
2. Res unit: Learning from the residual structure in the Resnet model and using residual components make network deeper.
3. CSP1_X: Learning from the structure of CSPNet, it concat three convolutional layers and X Res units modules. CSP1_X is mainly to maintain or even improve the ability of CNN while reducing the amount of calculation by 20%.
4. SPP: Multi-scale fusion is carried out using the maximum pooling of 1×1, 5×5, 9×9, 13×13.
5. Concat: Tensor splicing and expanding dimension, which is equivalent to increasing the number of categories of features, and description of each type of feature has not changes.

3.2.2. Neck. The main purpose of the Neck layer is to get a Feature Pyramid Network (FPN)[14], so that multi-scale objects can be better detected in the entire image. At the same time, the difference between CSP2_X from Neck and CSP1 in Backbone is that the residual module is replaced with ordinary convolution module in CSP2_X. Network performs upsampling twice in Neck, and got 26×26 and 52×52 feature maps respectively. Yolov5 model detects objects of different sizes in different sizes feature maps. We use 13×13 feature maps to detect large objects, because it has a larger receptive field for each grid, which is more suitable for predicting large targets, while 52×52 is suitable for small object detection because it feels small field.

The Neck part also learns from PANet to improve accuracy of small object detection. First of all, PANet is an FPN network. For transferring shallow features to the top layer P5, model must go through dozens or even hundreds of layers. Obviously, after such multi-layers passed, the shallow features has
lost most information. As shown by the red dotted line in Figure 4, a complete backbone structure is required before shallow features enter the next stage.

![Figure 3 Network structure of Neck](image1.png)

**Figure 3** Network structure of Neck

The idea of PANet is to connect the shallow features directly to the $P_2$ layer, and then go through the Neck part to connected to the top layer $N_9$, and the total number of layers passed is less than 10. Therefore, PANet can better retain the shallow information, and the shallow information usually contains more small object feature information, which is good for detection of small object. The green dotted line in the figure 4 symbolizes the optimized path of PANet.

3.2.3. **Head.** For the outputing in head part, because the output has three sizes feature maps, each grid in any of maps is assigned three prediction anchor. each anchor predicts 5 tuples and the one-hot vector including the number of object categories. For signal light detection model, the number of object categories is 2, so the total size is: $3 \times (4 + 1 + 2)$. 3 represents three priori boxes, 4 represents the object rectangle position, 1 represents the object confidence, and 2 represents the total object categories (2 one-hot vector categories). The output size of each map is (the first two bits are the feature map size, see the Neck section above):

- $[13, 13, 3 \times (4 + 1 + 2)]$
- $[26, 26, 3 \times (4 + 1 + 2)]$
- $[52, 52, 3 \times (4 + 1 + 2)]$

3.3. **Loss function**

The location loss function of signal light detection network is GiouLoss [15]. First calculating the minimum area of two boxes, and then calculating the size of the closed area that does not belong to two boxes, and last calculating the IOU. Finally, subtracting this part from IOU to get GIOU. This method
converges much faster than the original method of simply calculating the IOU area, and training speed can be accelerated.

The final loss function is:

\[
\begin{align*}
\lambda_{iou} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \lambda_{iou} t_{iou} + \\
\lambda_{cls} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \lambda_{cls} (C_i - \hat{C}_i)^2 + \\
- \sum_{i=0}^{S^2} \sum_{j=0}^{B} \lambda_{cls} \sum_{c \in \text{classes}} \lambda_c \left[ \hat{p}_c(c) \log(p_c(c)) + (1 - \hat{p}_c(c)) \log(1 - \hat{p}_c(c)) \right] \end{align*}
\]

\( S \) stands for 13,26,52, that is the size of grid. \( B=3 \), which is the number of anchor boxes for each grid.

The first line represents the location loss, \( t_{iou} \) represents that if the overlapping between bounding box predict and the GT object more than other GT object, then it is included in the location loss function, but if the overlap is less than threshold, it is not included in the loss function, \( t_{noob} \) just like that.

\( \lambda_{iou} \) iou_normalizer (location loss)
\( \lambda_{cls} \) cls_normalizer (target confidence loss)
\( C \) is the confidence of the predicted box, and \( \hat{C} \) is the confidence of the GT. \( \hat{C} \) is always 1.
\( \lambda_c \) classes_multiplers (classification loss) use standard cross entropy loss function 
\( \hat{p}_c \) is the GT true classification function, and \( p_c \) is the predicted box classification function.

4. Experiment

4.1. Training set & test set preparation

We need to prepare the training dataset and change part of the code of Yolov5 to start training. We place two cameras at the head and end of the train to collect data, and pick images including signal lights we need from thousands of images. Relying on that, we have about 3000 images including signal lights with corresponding labels, all of which are 1280×720 in size and red or green lights in colors.

Because the number of categories is small relatively, we do not need to label the background. The final dataset categories are shown in table 1.

| Number of images | Number of signal lights |
|------------------|-------------------------|
| Red light        | 2460                    | 3409                    |
| Green light      | 361                     | 676                     |

Figure 5 Examples of training set images

To verify the result of Yolov5, 1600 images of signal lights in different environments were prepared. Contains all the situations that can be encountered during subway operation, Including red or green lights in the outdoor daytime, outdoor night and tunnel, as well as weather factors such as rain, fog, and light factors such as cloudy, dim and camera lens pollution.
Table 2. Image statistics of test set

|             | G:250 | R:250 |
|-------------|-------|-------|
| Outdoor daytime |       |       |
| Outdoor night   | G:250 | R:250 |
| Tunnul         | G:300 | R:300 |

4.2. Data enhancement

4.2.1. Adaptive anchor frame. In generally, signal light object is small target, but some signal lights are larger when trains ran over them. So the choice of anchor size will also have a great impact on the result of object detection. Compared with the previous versions of Yolo, Yolov5 innovatively uses the self-adaptive anchor method. Original Yolo’s anchor size is calculated based on the dataset label, but Yolov5 can set anchor size as a parameter for training. So that the size of anchor and the size of signal lights can be optimally matched, we can also manually adjust the size of anchor, such as narrow it at all scales, which is conducive to small signal lights detection.

4.2.2. Mosaic data enhancement. Mosaic data enhancement uses 4 images, randomly zoomed, cropped and arranged for splicing, which enriches the dataset greatly. The advantage is that the background of detected object can be enriched. And surrounding environment have great influence for small object detection. Enriching dataset is equivalent to increasing the training set data to improve the detection accuracy.

![Figure 6 Example of Mosaic data enhancement](image)

4.3. Training parameter settings

In order to keep detection speed, we choose yolov5s with the simplest and fastest network structure, and train models of 300 epochs and 450 epochs for comparison. The two models use same learning rate 1e-5, and the loss function is shown in section 3. We use official pre-trained yolov5s model on the coco dataset as pre-trained model. The training environment is python3.8 and pytoch1.6. Training runs on two 2080Tis with a total of 24GB video memory. Batchsize is set to 128, finally loss value has converged, and loss values of the two models are very close.

4.4. Test Results

4.4.1. Yolov5 test results. Testing model effects of 300 epochs and 450 epochs respectively, and select the better one for final deployment.

Only position information of signal lights needs to be detected, no IOU threshold is required, so we do not set the threshold limit. Finally, we get recall rate, accuracy rate and average running speed of models.
**Table 3. Yolov5s (300epoch) test results**

| Successful detection | False detection | Recall | Precision | speed |
|-----------------------|-----------------|--------|-----------|-------|
| Outdoor daytime       | G:244 R:241     | G:3    | R:8       | G:0.976 R:0.964 G:0.988 R:0.968 |
| Outdoor night         | G:246 R:244     | G:5    | R:12      | G:0.984 R:0.976 G:0.980 R:0.953 |
| Tunnul                | G:286 R:294     | G:11   | R:5       | G:0.953 R:0.980 G:0.963 R:0.983 |

**Table 4. Yolov5s (450epoch) test results**

| Successful detection | False detection | Recall | Precision | speed |
|-----------------------|-----------------|--------|-----------|-------|
| Outdoor daytime       | G:236 R:235     | G:4    | R:8       | G:0.944 R:0.940 G:0.983 R:0.967 |
| Outdoor night         | G:249 R:236     | G:4    | R:10      | G:0.993 R:0.944 G:0.984 R:0.959 |
| Tunnul                | G:284 R:286     | G:18   | R:3       | G:0.947 R:0.953 G:0.940 R:0.990 |

It is obvious that detection result of the 450 epochs model is worse than 300 epochs. Possible reason for that is the signal lights training dataset are insufficient. Because of the similar signal lights environment, images repeatability is too high, numerous images maybe oversample, which eventually leads to overfitting at model of 450 epochs. However, if part of images in the testset are similar to the training set, such as the green lights of outdoor night, the test accuracy is greatly improved.

From the above two tables, we can also see the outstanding speed of Yolov5, which can maintain a processing speed of 100FPS with high recall and accuracy.

**4.4.2. Detection results of other signal detection algorithms.** In order to compare Yolov5 with other signal lights detection algorithms, we use another currently commonly used object detection network CenterNet [16] for signal lights detection. Using same training set and same learning rate as Yolov5, training 200 epochs on same computer, the final result is as follows:

**Table 5. CenterNet test results**

| Successful detection | False detection | Recall | Precision | speed |
|-----------------------|-----------------|--------|-----------|-------|
| Outdoor daytime       | G:236 R:245     | G:2    | R:6       | G:0.944 R:0.980 G:0.991 R:0.967 |
| Outdoor night         | G:246 R:248     | G:2    | R:8       | G:0.984 R:0.992 G:0.999 R:0.969 |
| Tunnul                | G:293 R:295     | G:1    | R:2       | G:0.976 R:0.983 G:0.997 R:0.993 |

It can be seen that recall and accuracy of CenterNet are higher than those of Yolov5, but it runs slowly and cannot meet the real-time requirements.

We finally tested the traditional CV-based signal lights detection algorithms, using a method similar to Literature [23], first extracting the signal ROI, and then getting result through the classification network Mobile-Netv2. The final result is as follows:
Table 6. Traditional CV + classification neural network detection results

|                  | Successful detection | False detection | Recall | Precision | speed |
|------------------|----------------------|-----------------|--------|-----------|-------|
| Outdoor daytime  | G:226                | G:18            | G:0.904| G:0.926   | 33FPS |
|                  | R:224                | R:20            | R:0.896| R:0.918   |       |
| Outdoor night    | G:222                | G:23            | G:0.888| G:0.906   |       |
|                  | R:227                | R:20            | R:0.908| R:0.908   |       |
| Tunnul           | G:265                | G:30            | G:0.883| G:0.898   |       |
|                  | R:276                | R:15            | R:0.920| R:0.948   |       |

Based on the results of algorithm, we found that traditional CV can basically extract the position of signal lights in most scenarios. But when light conditions are terrible or signal lights are bright, there will still be unrecognized issues, and detection speed is slow, with an average speed of 33FPS.

4.5. Algorithm effect comparison

According to detection results of the above three algorithms, we can sort out the following three tables:

(To simplified tables, we abbreviate the traditional CV-based signal lights detection algorithms mentioned above as Vision-based heuristic method)

(Successful detection: the signal light is detected correctly in the image)

(Correct detection: the detected object is signal light)

Table 7. Algorithm effect based on the color of signal lights

|                  | Successful detection | Correct detection | Recall | Precision |
|------------------|----------------------|-------------------|--------|-----------|
| Vision-based heuristic | G:713/800           | G:713/784         | G:0.891| G:0.909   |
|                   | R:272/800           | R:727/782         | R:0.909| R:0.930   |
| CenterNet         | G:775/800           | G:775/780         | G:0.969| G:0.994   |
|                   | R:788/800           | R:788/804         | R:0.985| R:0.980   |
| Yolov5s           | G:779/800           | G:779/804         | G:0.974| G:0.969   |
|                   | R:776/800           | R:776/795         | R:0.970| R:0.976   |

It can be seen that the result of red signal lights is better than that of green signal lights. This is related to the color of signal lights and the intensity of different colors affected by the surrounding environments of lights. The red lights have a long wavelength and strong penetrability, which is less affected by external factors. It also has a lot to do with the comparison of the number of red and green lights in training set.

Table 8. Algorithm effect based on signal lights environment

|                  | Successful detection | Correct detection | Recall | Precision |
|------------------|----------------------|-------------------|--------|-----------|
| Vision-based heuristic | Day:450/500         | Day:450/488       | Day:0.900| Day:0.922 |
|                   | Night:449/500       | Night:449/492     | Night:0.898| Night:0.913 |
|                   | Tunnel:541/600      | Tunnel:541/586    | Tunnel:0.902| Tunnel:0.923 |
| CenterNet         | Day:481/500         | Day:481/489       | Day:0.962| Day:0.984 |
|                   | Night:494/500       | Night:494/504     | Night:0.988| Night:0.980 |
|                   | Tunnel:588/600      | Tunnel:588/591    | Tunnel:0.980| Tunnel:0.995 |
| Yolov5s           | Day:485/500         | Day:485/496       | Day:0.970| Day:0.978 |
|                   | Night:490/500       | Night:490/507     | Night:0.980| Night:0.966 |
|                   | Tunnel:580/600      | Tunnel:580/596    | Tunnel:0.967| Tunnel:0.973 |
As can be seen from the above table, since the two deep learning model use same training sets, the final result is also relatively similar, and the best results are both showed at outdoors night. That also corresponds to the largest proportion of outdoor night images in training set.

Table 9. The average detection effect of the three algorithms

| Algorithm              | Successful detection | Correct detection | Recall | Precision | Velocity |
|------------------------|----------------------|-------------------|--------|-----------|----------|
| Vision-based heuristic | 1440/1600            | 1440/1566         | 0.900  | 0.920     | 33FPS    |
| CenterNet              | 1563/1600            | 1563/1584         | 0.977  | 0.987     | 40FPS    |
| Yolov5s                | 1555/1600            | 1555/1599         | 0.972  | 0.972     | 100FPS   |

It can be seen from the table that CenterNet and Yolov5s have little difference in recall rate, but the accuracy of CenterNet is little higher than Yolov5s. In terms of speed, Yolov5s is extremely faster than CenterNet and traditional CV, and basically realizes real-time detection on mobile platforms.

4.6. Algorithm evaluation and conclusion

As can be seen from the above table, the detection speed and result of object detection based on neural network are much better than the traditional CV algorithm. Yolov5 is extremely fast meanwhile its detection accuracy satisfies the requirements. And it has better performance in all scenarios. However, the method relying on neural network has data requirements. In the case of insufficient data, training issues such as over-fitting will occur, resulting in poor detecting results.

4.6.1. Robustness. Traditional CV algorithms are slightly less robust, and the perception result is poor when encountering severe light scattering, far away signal lights or entering and exiting railway station. By contrast, deep learning algorithms have good performance in various environments, and the robustness meets requirements for deployment.

4.6.2. Detection result and algorithm complexity. For deep learning model, the network structure of CenterNet is simpler than Yolov5, which is easy to understand and improve. Yolov5 uses a lot of tricks to achieve a faster speed, which makes network more complicated. Comparing the above table, we can find that detection result of Yolov5 is worse than CenterNet, but the difference between two models is small, and both can meet the deployment requirements.

Another important metric for signal lights detection is detection distance. Both Yolov5 and CenterNet have good detection capabilities for long-distance signal lights. But traditional CV is poor in this aspect. When signal light is less than 10×10 pixels, traditional CV algorithm is basically unable to detect, but the deep learning method still can detect successfully.

4.6.3. Deployment. For some special environments, if you cannot collect enough image datasets, deep learning algorithms is unable to use. At this time, we can use traditional CV solutions for detection. In general, in addition to the above robustness and detection effects, the deployment of applications needs to consider the operating speed. Yolov5 has obvious advantages in running speed. In general, mobile devices have poor computing power, all kinds of algorithms will be 2-3 times slower than deployed on mobile devices. From this point of view, only Yolov5 can meet the real-time requirements.

5. Conclusion

This paper proposes a new signal lights detection method, namely the railway signal light detection based on Yolov5 neural network. We first construct a dataset consisting of subway scenes with signal lights, and then start training to get the Yolov5 model. After that we use the test set to confirm model detection effect. We found that the signal lights detection model trained by Yolov5s has an average recall rate and accuracy of 0.972, while running speed can reach 100FPS. At the same time, Because of high robustness, the model can basically adapt to railway signal lights detection tasks in all environments.
Acknowledgments
This work is supported by the National Natural Science Foundation of China under Grant #52072020 and the Beijing Natural Science Foundation under Grant #L191002. Thanks are due to Beijing AI for Rail Technology Co.Ltd. who provides us with data support. The authors would also like to thank the insightful and constructive comments from anonymous reviewers.

This work is supported by the Beijing Natural Science Foundation under Grant #L191002 and the National Natural Science Foundation of China under Grant #52072020.

References
[1] Hassan, N., Ming, K. W., & Wah, C. K. (2020, February). A Comparative Study on HSV-based and Deep Learning-based Object Detection Algorithms for Pedestrian Traffic Light Signal Recognition. In 2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS) (pp. 71-76). IEEE.
[2] Alam, A., & Jaffery, Z. A. (2019). A vision-based system for traffic light detection. In Applications of Artificial Intelligence Techniques in Engineering (pp. 333-343). Springer, Singapore.
[3] Lee, E., & Kim, D. (2019). Accurate traffic light detection using deep neural network with focal regression loss. Image and Vision Computing, 87, 24-36.
[4] Pon, A., Adrienko, O., Harakeh, A., & Waslander, S. L. (2018, May). A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection. In 2018 15th Conference on Computer and Robot Vision (CRV) (pp. 102-109). IEEE.
[5] Le, Q. N. N., Bhattacharyya, A., Chembakasseril, M. T., & Hartanto, R. (2020). Real-time Sign Detection and Recognition for Self-driving Mini Rovers based on Template Matching and Hierarchical Decision Structure. In ICAART (1) (pp. 208-215).
[6] Le, Q. N. N., Bhattacharyya, A., Chembakasseril, M. T., & Hartanto, R. (2020). Real-time Sign Detection and Recognition for Self-driving Mini Rovers based on Template Matching and Hierarchical Decision Structure. In ICAART (1) (pp. 208-215).
[7] Ouyang, Z., Niu, J., Ren, T., Li, Y., Cui, J., & Wu, J. (2020). MBBNet: An edge IoT computing-based traffic light detection solution for autonomous bus. Journal of Systems Architecture, 109, 101835.
[8] Aneesh, A. N., Shine, L., Pradeep, R., & Sajith, V. (2019, July). Real-time Traffic Light Detection and Recognition based on Deep RetinaNet for Self Driving Cars. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT) (Vol. 1, pp. 1554-1557). IEEE.
[9] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
[10] Diaz, M., Cerri, P., Pirlo, G., Ferrer, M. A., & Impedovo, D. (2015, September). A survey on traffic light detection. In International Conference on Image Analysis and Processing (pp. 201-208). Springer, Cham.
[11] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
[12] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
[13] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).
[14] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).
[15] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized
intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 658-666).

[16] Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. arXiv preprint arXiv:1904.07850.

[17] Ouyang, Z., Niu, J., Liu, Y., & Guizani, M. (2019). Deep CNN-based real-time traffic light detector for self-driving vehicles. IEEE transactions on Mobile Computing, 19(2), 300-313.

[18] Lee, E., & Kim, D. (2019). Accurate traffic light detection using deep neural network with focal regression loss. Image and Vision Computing, 87, 24-36.

[19] Du, X., Li, Y., Guo, Y., & Xiong, H. (2017, July). Vision-based traffic light detection for intelligent vehicles. In 2017 4th International Conference on Information Science and Control Engineering (ICISCE) (pp. 1323-1326). IEEE.

[20] Ouyang, Z., Niu, J., Liu, Y., & Guizani, M. (2019). Deep CNN-based real-time traffic light detector for self-driving vehicles. IEEE transactions on Mobile Computing, 19(2), 300-313.