Article

The One-Stage Detector Algorithm Based on Background Prediction and Group Normalization for Vehicle Detection

Fei Lu 1,2, Fei Xie 1,2,*, Shibin Shen 1,2, Jiquan Yang 1,2, Jing Zhao 3, Rui Sun 4,5, and Lei Huang 6

1 School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, China; 181802025@stu.njnu.edu.cn (F.L.); 63018@njnu.edu.cn (S.S.); 63047@njnu.edu.cn (J.Y.)
2 Nanjing Institute of Intelligent High-End Equipment Industry Company Limited, Nanjing 210042, China
3 College of Automation & College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; zhaojing@njupt.edu.cn
4 College of Civil Aviation, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China; rui.sun@nuaa.edu.cn
5 The Center for Transport Studies, Imperial College, London SW7 2AZ, UK
6 School of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China; huanglei@njfu.edu.cn
* Correspondence: xiefei@njnu.edu.cn

Received: 17 July 2020; Accepted: 10 August 2020; Published: 25 August 2020

Abstract: Vehicle detection in intelligent transportation systems (ITS) is a very important and challenging task in traffic monitoring. The difficulty of this task is to accurately locate and classify relatively small vehicles in complex scenes. To solve these problems, this paper proposes a modified one-stage detector based on background prediction and group normalization to realize real-time and accurate detection of traffic vehicles. The one-stage detector firstly adds a module to adjust the width and height of anchors and predict the target background, which avoids the problem of the target vehicle missing detection or wrong detection due to the influence of the complicated environments. Then, group normalization and the loss function based on weight attenuation can improve the one-stage detector performance in the training process. The experimental results on traffic monitoring datasets indicate that the improved one-stage detector is superior to the other neural network models in terms of precision at 95.78%.

Keywords: vehicle detection; one-stage detector; group normalization; intelligent transportation

1. Introduction

Target detection technology based on deep learning is widely used in unmanned driving, intelligent monitoring, intelligent transportation and other fields. For example, with the traffic violation of statutes increased, monitoring and accurately detecting vehicles in videos has become an important research topic of city traffic management. The target detection algorithm based on deep learning in traffic attracts the attention of many researchers, but for complex traffic monitoring environments for different scales and types of vehicles, quick and accurate target detection has been by far one of the most challenging tasks [1,2].

In recent years, researchers at home and abroad have carried out extensive and in-depth research on target detection methods based on deep learning. The R-CNN (Region-CNN) model proposed by Girshick et al. [3] generates a large number of candidate boxes by selective search algorithms [4], and then candidate boxes are used as the input to the convolutional neural network for detection. However, the process of generating candidate boxes is time consuming, and there is a large amount of...
overlap between the candidate boxes. The improved Fast R-CNN algorithm [5] adds target classification and regression into the neural network, reducing the operation time. Girshick et al. put forward the concept of anchor boxes in Faster R-CNN [6] and realized the generation of candidate boxes through convolutional neural network, thus effectively reducing the time of target detection. However, all of the above algorithms generate candidate targets through candidate boxes, and then use convolutional neural network for processing. Although the detection accuracy is high, it cannot meet real-time requirements. In other words, all of the above methods are two-stage detectors. In order to solve the real-time problem of target detection, Redmon et al. proposed the YOLO (You Only Look Once) algorithm [7] and introduced a one-stage detector, which for the first time treated target detection as a regression problem, and directly predicted the category and position of targets through the convolution neural network without generating candidate boxes. Although it improves the detection speed, the accuracy is also low. Subsequently, Redmon proposed an improved YOLO v2 algorithm [8] and applied all normalization layers in the convolutional layer [9], which improved the recall rate and positioning ability of the target. Liu et al. proposed the SSD algorithm [10], which was faster and more accurate than the previous YOLO algorithm. However, due to the extraction of features through low-level convolution, there was a problem of insufficient feature extraction. Later, based on YOLO v2, Redmon proposed the YOLO v3 algorithm [11], which adopted the Darknet53 network structure and predicted with multi-scale features to improve the detection accuracy of small targets. Meanwhile, it also improved the detection accuracy while maintaining real-time performance. On the basis of YOLO v3 algorithm, Bochkovskiy et al. proposed YOLO v4 algorithm [12], which added CSPDarknet53, the Mish activation function and dropblock into the backbone network. The detection performance of objects is better than YOLO v3.

However, due to the complex and varied environmental background and the shield between vehicles, there are still many problems in the practical application of vehicle detection, and the deep learning model for vehicle detection in complex environments is still a challenging subject. This paper proposes the one-stage detector algorithm based on background prediction and group normalization for vehicle detection algorithms, which can improve the accuracy of the vehicle detection in complex environments.

The contribution of this paper is as follows:

1. Adding a network module to adjust the width and height of anchor boxes and predict target backgrounds, which firstly detects the environment background to prevent the vehicle from being affected by the environmental background. Therefore, this method can reduce the error of vehicle wrong detection or missing detection and improve the accuracy of vehicle detection.
2. Using group normalization instead of batch normalization, which solves the problem that the performance of batch normalization is getting worse with the decrease of batch size. At the same time, a weight attenuation term is added based on the traditional cross-entropy loss function to solve the problem that the positive samples cannot be effectively trained.

This paper is organized as follows. In Section 2, the existing work related to target detection is introduced. The third section mainly proposes the one-stage detector algorithm based on background prediction and group normalization for vehicle detection. In Section 4, the datasets and analysis of tests are introduced. Finally, Section 5 provides the conclusions for this research work and recommendations for future work.

2. Related Work

Deep learning target detection algorithms have been widely applied to vehicle detection of urban traffic monitoring. Meanwhile, improved algorithms for specific problems have been proposed and achieved remarkable results.

The region-based vehicle detection algorithm firstly produces the candidate box in the image and then classifies the vehicle in the candidate box. Yi Zhou et al. [13] proposed a unified rapid vehicle
The detection framework (DAVE), which effectively combined vehicle detection and attribute marking. DAVE is composed of two convolution neural networks: a vehicle candidate box extracted network and a vehicle inference learning network, to predict the angle of view, the properties of color and type. The improved network can effectively detect the traffic monitoring vehicles, but DAVE has a poor handling effect on small vehicles with occlusion and slow detection speed. Wenming Cao et al. [14] proposed a fast deep neural network with knowledge guidance training and prediction of regions of interest, which can significantly reduce overall computational complexity and improve vehicle detection performance. Compared with the traditional SSD algorithm, the detection speed of this method is significantly improved, but the detection accuracy is not significantly improved.

The vehicle detection algorithm based on regression uses a one-stage neural network to directly predict the position and category of vehicles in the image and realize real-time detection. Zhiming Luo et al. [15] proposed a traffic camera data set (MIO-TCD), including 11 traffic object classes, to evaluate traffic vehicle detection algorithms. In order to solve the problem that convolution features are mesoscale sensitive in vehicle detection tasks, Xiaowei Hu et al. [16] proposed a scale-insensitive convolutional neural network (SINet) for rapid detection of vehicles with large-scale changes. SINet as a one-stage detector can improve accuracy and speed in KITTI datasets, and the detection performance of SINet for small scale vehicles is still to be improved due to the large number of highly overlapping, fuzzy, and small vehicles in practical application scenarios.

The structure of one-stage detectors is shown in Figure 1, which consists of an input terminal, a backbone network module, a characteristic enhancer module and a prediction module. As the backbone network of feature extractor, VGG16, ResneXt-101 [17], and Darknet53 can achieve the feature extraction of the input image. In order to better fuse features, the Neck module was at the back of the backbone network, which can use FPN [18], PANet [19] and Bi-FPN algorithms. Because different targets have different characteristics, using the characteristics of shallow can distinguish some simple targets, and using the characteristics of deep can distinguish the complex target. Then the characteristics of different outputs can better deal with large targets and small targets. Finally, RPN, RetinaNet and FCOS were used in the prediction module to predict the position and category of images. The multi-scale prediction of the one-stage detector was be carried out on the feature map with the size of 19 × 19, 38 × 38 and 76 × 76. However, before the feature map outputs the prediction results, the network carried out the feature fusion operation first, and splintered the features with high semantic, low resolution and low semantic, high resolution together so that the features with high resolution also contain rich semantic information. For the three output feature graphs, each pixel predicts three boxes on the feature graph, and each box predicts the center coordinates x, y, h, w and the object confidence. Finally, the non-maximum suppression (NMS) algorithm can select the predicted bounding box as the final detection box.

Figure 1. The architecture of one-stage detector.

3. The Structure of One-Stage Detector Based on Background Prediction and Group Normalization

The architecture of the one-stage detector algorithm based on background prediction and group normalization for the vehicle detection algorithm is shown in Figure 2. The backbone network adopts CSPDarknet53 structure, which contains five CSP (Cross Stage Paritital) modules. It can enhance the learning ability of the convoluted neural network, reduce the amount of computation.
4. The One-Stage Detector Algorithm Based on Background Prediction and Group Normalization for Vehicle Detection Algorithm

4.1. The Prediction Branch with Adjusting Anchor and Predicting Background

The modified one-stage detector algorithm follows the backbone network of a one-stage detector, adds a network module to adjust the width and height of anchor boxes and predicts the target backgrounds. A threshold value was set after the output result of the branch network, and a background predicted value was generated according to the result of classification and the threshold value set. When the probability that the sample is predicted as background is greater than the threshold value, the object prediction is 0, otherwise it is 1. The background predicted value was mapped to the last layer (the prediction layer), and the predicted value was 1 in the sample and participated in the final training and testing. The anchor box part of the figure is the anchor boxes adjustment and background prediction module, which is connected with the feature fusion layer of three scales, respectively.

Figure 3 shows that convolution module parameters are added after the 76 × 76 characteristic graph. The convolution module was composed of three convolution layers whose convolution kernel size was 5 × 5, 3 × 3 and 1 × 1, respectively, and the step size was 1. The parameters of the network layer after adding feature map with scales of 38 × 38 and 19 × 19 and are similar to Figure 2. Finally, the binary classification results of width and height of anchor boxes and target background were produced on feature maps of three scales.
After the above operation, three feature graphs of size H × W × 18 are obtained, in which (H, W) is the height and width of the feature graph, whose values are respectively (76,76), (38,38) and (19,19). Eighteen is the number of channels in the feature graph, which can be written as 3 × (4 + 2). Number 3 means that each pixel in the feature graph predicts three boxes and number 4 + 2 represent the x-coordinate, y-coordinate, height and width of each box and the score of the sample belonging to the object and background. Finally, using the target background score value and the set threshold value of each sample to calculate the object predicted value and correcting the anchor boxes according to the width and height offset value, an output was produced from the feature map. In the final prediction stage, the prediction layer in the above figure used the background predicted value and the corrected anchor boxes as predictions.

4.2. The One-Stage Detector Training Based on Group Normalization

Batch Normalization (BN) is commonly used in target detection, but it has certain limitations. BN makes normalization in the dimension of the batch; however, this dimension is variable. In the process of training datasets, sliding was used to calculate the mean and variance of data and directly call the training dataset, however when the training dataset and testing dataset distribution is different, this can lead to errors. Because the input image data is large in target detection, batch size can only be set as a small value in order to save memory, and a small batch size will lead to an inaccurate mean value and variance, which could degrade BN performance. In this paper, group normalization (GN) instead of BN normalize the channel dimension, and the formula is:

\[
S_i = \{k|k_N = i_N, \left[ \frac{k}{C/G} \right] = \left[ \frac{i}{C/G} \right] \}
\]

where G represents the number of groups and is a predefined super parameter; C/G is the number of channels in each group; \( \left[ \frac{k}{C/G} \right] = \left[ \frac{i}{C/G} \right] \) represents indexes i and k in the same set of channels, each set of channels is stored sequentially along the C axis, GN along the (H, W) and along a set C/G channel to calculate the mean and variance; H and W are the height and width axes of space.

GN is the mean and variance of each group in the direction of the channel, and has nothing to do with batch size, so it is not constrained by batch size. With the reduction of batch size, GN’s performance is stable, while BN’s performance gets worse and worse.

4.3. Target Detection Loss Function Based on Weight Attenuation

In the process of calculating loss, all predicted bounding boxes in the model by one-stage detector is divided into positive samples (IoU > 0.5) or negative samples (IoU < 0.4). In general, the proportion of the target is much smaller than that of the background, so the positive and negative samples are easily distinguished, and most of them are negative samples that are easy to classify. At this time, the loss
function is relatively slow and may not be optimized in the iterative process of a large number of samples. In this paper, loss function adds a weight attenuation to solve the problem of an unbalanced number of positive and negative samples, which is modified based on standard cross-entropy loss.

The cross-entropy loss function is:

\[
CE = -y \log y' - (1 - y) \log (1 - y') = \begin{cases} 
-\log y' & y = 1 \\
-\log(1 - y') & y = 0 
\end{cases}
\]

(2)

where \( y' \) is the predicted probability, which is between 0 and 1; and \( y \) is the true label. It can be seen that in the case of cross-entropy loss, the output probability of the positive sample is higher, the loss is smaller, and the output probability of the negative sample is smaller. The modified formula is as follows:

\[
FL = \begin{cases} 
(1 - y')^\gamma [-y \log y' - (1 - y) \log(1 - y')] \\
y'^\gamma [-y \log y' - (1 - y) \log(1 - y')] 
\end{cases}
\]

\[
= \begin{cases} 
-(1 - y')^\gamma \log y' & y = 1 \\
-y'^\gamma \log(1 - y') & y = 0 
\end{cases}
\]

(3)

In the formula, \( \gamma \) is the adjustment parameter, and the value is set as 2, which is used to adjust the rate of weight reduction of simple samples, making the model pay more attention to the difficulty in sample classification during the training process. The improved loss function is:

\[
L = \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} t_{ij}^{\text{obj}} [-t_{x_i} \log t_{x_i} - (1 - t_{x_i}) \log(1 - \hat{t}_{x_i})] + \\
\lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} t_{ij}^{\text{obj}} [-t_{y_i} \log t_{y_i} - (1 - t_{y_i}) \log(1 - \hat{t}_{y_i})] + \\
\lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} t_{ij}^{\text{obj}} ((t_{w_i} - \hat{t}_{w_i})^2 + (t_{h_i} - \hat{t}_{h_i})^2) + \\
\lambda_{\text{obj}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} t_{ij}^{\text{obj}} [-c_i \log c_i - (1 - c_i) \log(1 - \hat{c}_i)] + \\
\lambda_{\text{noobj}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} t_{ij}^{\text{noobj}} [-c_i (1 - \hat{c}_i)^\gamma \log c_i - (1 - c_i) (\hat{c}_i)^\gamma \log(1 - \hat{c}_i)] + \\
\lambda_{\text{class}} \sum_{i=0}^{s^2} \sum_{c \in \text{classes}} t_{ij}^{\text{obj}} [-p_i(c) \log p_i(c) - (1 - p_i(c)) \log(1 - \hat{p}_i(c))] 
\]

(4)

In the formula, \( s^2 \) is all grid units of output feature graph; \( B \) is the number of predicted bounding box for each grid; \( t_{ij}^{\text{obj}}, t_{ij}^{\text{noobj}} \) is used to determine whether the \( j \)-th boundary box in the \( i \)-th grid is responsible for object prediction; \( t_{ij}^{\text{obj}} \) determines whether the center of the object falls on the \( j \)-th grid; \( \hat{t}_{x_i}, \hat{t}_{y_i}, \hat{t}_{w_i}, \hat{t}_{h_i} \) is the relative position of the predicted bounding box; \( t_{x_i}, t_{y_i}, \hat{t}_{w_i}, \hat{t}_{h_i} \) is the position parameter of the real box; \( c_i \) is the confidence of the true bounding box; \( \hat{c}_i \) is the confidence of the predicted bounding box; \( p_i(c) \) is the class probability of the true bounding box; \( \hat{p}_i(c) \) is the class probability of the predicted boundary box; \( \lambda_{\text{coord}} \) is the weight of coordinate loss in the total loss whose value is set as 5; \( \lambda_{\text{obj}} \) is the weight of positive samples in the confidence loss whose value is set as 1; \( \lambda_{\text{noobj}} \) is the weight of negative samples in confidence loss whose value is set as 0.5; and \( \lambda_{\text{class}} \) is the weight of category loss in the total loss whose value is set as 1.
5. Tests and Results Analysis

5.1. Dataset

To evaluate the performance of the improved network and the proposed method, the dataset used in this paper was from surveillance videos in some areas of Zhenjiang, Jiangsu province, China. In the Figure 4, these video datasets contain real image data from urban, rural and highway scenes, with up to 30 cars per image and varying degrees of occlusion. In this paper, the experimental dataset consisted mainly of 30,000 images from the video, of which 24,000 images are the training and verification dataset, and 6000 images are the testing dataset. Each test image is labeled by LabelImg. All vehicle categories were cars, buses and trucks. The monitoring period included 24 h a day, sunny days, cloudy days and rainy days, etc., which are representative to some extent.

![Figure 4. The training curve of loss function based on weight attenuation.](image)

5.2. Experiment and Analysis

The hardware environment of the experiment was Intel Core i7-9900K CPU, NVIDIA Titan RTX GPU. In the training stage of this paper, 50,000 iterations were carried out, momentum configuration was 0.9, weight attenuation configuration was 0.0005, batch size was set as 16, learning rate was initially 0.001, and learning rate was reduced to 0.0001 when the network iteration was 35,000. The learning rate was reduced 0.00001 when the network iteration was 45,000. In order to facilitate the comparison of experimental results and other algorithm results in this paper, the commonly used calculation formulas of accuracy is as follows.

\[
\text{precision} = \frac{TP}{TP + FP} \quad (5)
\]

\[
\text{recall} = \frac{TP}{TP + FN} \quad (6)
\]

where TP is the number of correctly divided positive samples; FP is the number of wrongly divided into positive samples; FN is the number of samples wrongly divided into negative samples; TN is the number of correctly divided negative samples.

The P-R (Precision-Recall) curve takes recall rate and precision rate as horizontal and vertical coordinates, and the area enclosed by the P-R curve is the Average Precision (AP). AP_{50} is the AP measurement at IOU threshold of 0.5; AP_{75} is the AP measurement at IOU threshold of 0.75; AP_{S} is the AP measurement value of target frame with pixel area less than 32^2; AP_{M} is the AP measurement of...
The calculation formula of detection speed is as follows:

$$FPS = \frac{\text{frame}}{\text{second}}$$

(7)

where frame is the number of video frames and second is the unit of time.

In this paper, the pre-training model yolov3.conv.137 was as the initial parameter of the network during training, which can greatly shorten the training time. Figure 4 shows a comparison of the downward trend of the improved loss function as the number of iterations increased on traffic monitoring datasets with the method proposed in this paper. As can be seen from Figure 4, with increasing training times when the iteration exceeds 30,000, the loss value tends to be stable and finally drops to about 0.1.

Figures 5–7 show the comparison diagram of the detection effect of the proposed method and the original algorithm in different traffic monitoring environments. As can be seen from Figure 5, both algorithms can accurately detect the target vehicles on the highway, since there are only vehicles such as cars and trucks and no interference from other vehicles. Slight differences detected by the two methods are indicated by arrows in the figure. In Figure 6, the environment became more complex on the urban environment, and all kinds of traffic signs may have affected vehicle monitoring. In Figure 6a, the model identified the traffic sign as the car, resulting in vehicle wrong detection. In Figure 6b, the proposed one-stage algorithm successfully detected the road vehicle to solve the problem of partial wrong detection of non-vehicle targets. The detection differences in the figure are directly pointed out by arrows. As shown in Figure 7, there were many vehicles on rural roads, which made it more difficult to detect vehicles. In Figure 7a, due to the occlusion between vehicles and some pedestrians riding motorcycles or battery vehicles, there was interference with vehicle detection, leading to the detection of these non-vehicle targets. The model regarded pedestrians riding electric vehicles in the figure as vehicles, leading to wrong detection, and the model could not detect some vehicles in the surveillance video in the corner or far away from the surveillance perspective. In Figure 7b, because the branch based on background prediction was added, the background was detected in the complex environment, avoiding the object to be affected by the environment and resulting in wrong detection or missing detection. The vehicles that blocked each other did not affect vehicle detection and non-vehicle targets were not mistakenly detected. The model correctly recognized pedestrians riding electric vehicles and detected some vehicles in the corner or far away from the surveillance perspective. The differences detected by the two methods are indicated directly in the figure by different colored arrows. Therefore, the model training performance is improved based on group normalization and improved loss function in the vehicle training process.

Figure 5. Comparison of improved one-stage detector and one-stage detector in highway environment.
Figure 6. Comparison of improved one-stage detector and one-stage detector in urban environment.

Figure 7. Comparison of improved one-stage detector and one-stage detector in rural environment.

Then, the paper compares the P-R curves of different algorithms, as shown in Figure 8. It can be seen that the recall rate and accuracy rate of SSD and SINet were both lower than that of YOLO v4 and the improved algorithm in this paper. While under the same precision rate, the recall rate of the improved algorithm was higher than the YOLO v4 algorithm in this paper. The one-stage detector algorithm based on background prediction and group normalization for the vehicle detection model showed that the omission rate and error rate of the model were lower than that of the YOLO v4 model, which had a better detection effect.

Figure 8. P-R curves for different models.
Table 1 shows the performance comparison between the proposed model and other important target detection models. All models were trained with traffic monitoring training datasets. Among them, the detection accuracy of the proposed model in this paper was the highest, but its detection speed was only slightly lower than that of YOLO v4. This is because the network structure of this model is more complex than the latter, but it also fully meets the requirements of real-time detection.

| Model  | FPS | AP | AP$_{50}$ | AP$_{75}$ | AP$_S$ | AP$_M$ | AP$_L$ | mAP |
|--------|-----|----|----------|----------|--------|--------|--------|-----|
| SSD    | 32.4| 81.3| 74.3     | 68.3     | 66.3   | 73.2   | 80.4   | 86.93|
| SINet  | 31.5| 86.4| 80.2     | 76.4     | 75.2   | 80.1   | 87.2   | 88.5 |
| YOLO v4| 42.3| 92.8| 90.5     | 84.2     | 82.3   | 87.2   | 92.6   | 93.56|
| Ours   | 41.5| 95.3| 91.2     | 87.3     | 84.3   | 88.2   | 96.9   | 95.78|

6. Conclusions

This paper proposes a modified one-stage detector algorithm based on background prediction and group normalization for vehicle detection algorithm in traffic monitoring. The method increases the branch by adjusting anchor size and target background prediction. In the complex traffic environment, the convolutional network can distinguish the target vehicle from the non-target vehicle to avoid the problem of non-target vehicle wrong detection. In addition, group normalization instead of batch normalization can improve the performance of target detection, which is not limited by batch size. Finally, the cross-entropy loss function based on weight attenuation improves the training accuracy of the network. Experiments on traffic monitoring datasets show that the proposed one-stage detector can achieve an accuracy of approximately 95% and keep real-time performance at the same time, which is superior to the SSD, SINet, YOLO v4 models in vehicle detection accuracy. In the future, the proposed scheme will be applied to traffic management. This is necessary for the automatic detection of vehicles and pedestrian targets on urban roads and surrounding areas, which will effectively assist traffic management departments to analyze the running status of vehicles and pedestrians. In this way, more effective transportation schemes and early warning measures are developed to promote the construction of intelligent transportation systems.

Author Contributions: Conceptualization, F.L. and F.X.; methodology, F.L. and F.X.; software, F.L. and S.S.; validation, J.Z. and R.S.; formal analysis, F.L.; data curation, F.X. and S.S.; writing—original draft preparation, F.L.; writing—review and editing, F.X.; supervision, J.Y.; project administration, L.H.; funding acquisition, F.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the National Key Research and Development Program of China (Grant No. 2017YFB1103200), the National Natural Science Foundation of China (Grant No. 61601228, 41974033, 61803208), the Natural Science Foundation of Jiangsu Province (BK20161021, BK20180726).

Acknowledgments: The authors gratefully acknowledge the helpful comments and suggestions of the reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, J.; Xu, X.; Yin, D.; Ma, Z.; Shen, L. A Space Mapping Based 0–1 Linear Model for Onboard Conflict Resolution of Heterogeneous Unmanned Aerial Vehicles. *IEEE Trans. Veh. Technol.* 2019, 68, 7455–7465. [CrossRef]

2. Anala, M.R.; Makker, M.; Ashok, A. Anomaly Detection in Surveillance Videos. In Proceedings of the 2019 26th International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW), Hyderabad, India, 17 December 2019; pp. 93–98. [CrossRef]

3. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–29 June 2014; pp. 580–587. [CrossRef]
4. Buzcu, I.; Alatan, A.A. Fisher-selective search for object detection. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3633–3637. [CrossRef]

5. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]

6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: New York, NY, USA, 2017; Volume 39, pp. 1137–1149. [CrossRef]

7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]

8. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]

9. Wu, S. L1-Norm Batch Normalization for Efficient Training of Deep Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 2019, 30, 2043–2051. [CrossRef] [PubMed]

10. Liu, W.; Anguelov, D.; Erhan, D. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Las Vegas, NV, USA, 27–30 June 2016; pp. 21–37.

11. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.

12. Bochkovskiy, A.; Wang, C. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. Available online: https://arxiv.org/abs/2004.10934 (accessed on 23 April 2020).

13. Zhou, Y.; Liu, L.; Shao, L.; Mellor, M. Fast Automatic Vehicle Annotation for Urban Traffic Surveillance. *IEEE Trans. Intell. Transp. Syst.* 2018, 19, 1973–1984. [CrossRef]

14. Cao, W.; Yuan, J.; He, Z.; Zhang, Z.; He, Z. Fast Deep Neural Networks with Knowledge Guided Training and Predicted Regions of Interests for Real-Time Video Object Detection. *IEEE Access* 2018, 6, 8990–8999. [CrossRef]

15. Luo, Z. MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization. *IEEE Trans. Image Process.* 2018, 27, 5129–5141. [CrossRef]

16. Hu, X. SINet: A Scale-Insensitive Convolutional Neural Network for Fast Vehicle Detection. *IEEE Trans. Intell. Transp. Syst.* 2019, 20, 1010–1019. [CrossRef]

17. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [CrossRef]

18. Zhao, Y.; Han, R.; Rao, Y. A New Feature Pyramid Network for Object Detection. In Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China, 14–15 September 2019; pp. 428–431. [CrossRef]

19. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).