Vehicle detection algorithm based on lightweight YOLOX

Cong Xiong
Chongqing University of Science and Technology
Anning Yu (anning865@163.com)
Chongqing University of Science and Technology
Senhao Yuan
Chongqing University of Science and Technology
Xinghua Gao
Chongqing University of Science and Technology

Research Article

Keywords: Vehicle detection, BIT-Vehicle dataset, YOLOX, Deep learning

Posted Date: September 13th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2036635/v1

Abstract

Nowadays, accurate and fast vehicle detection technology is of great significance for the construction of intelligent transportation systems in the context of the era of big data. This paper proposes an improved lightweight YOLOX real-time vehicle detection algorithm. Compared with the original network, the detection speed and accuracy of the new algorithm have been improved with fewer parameters. First, referring to the GhostNet, we make a lightweight design of the backbone extraction network, which significantly reduces the network parameters, training cost, and inference time. Furthermore, by introducing the α-CIoU loss function, the regression accuracy of the bounding box(bbox) is improved while the convergence speed of the model is also accelerated. The experimental results show that the mAP of the improved algorithm on the BIT-Vehicle dataset can reach up to 99.21% with 41.2% fewer network parameters and 12.7% higher FPS than the original network and demonstrate the effectiveness of our proposed method.

1 Introduction

ITS (intelligent transportation system) [1] is the future direction of the transportation system, which integrates advanced information technology, communication technology, sensing technology, control technology, and computer technology into the transportation system to effectively monitor the transportation system, improve the transportation system's efficiency, and provide a guarantee for the safe operation of the transportation system. However, extracting information effectively from massive multimedia data is a great challenge. How to use computers to automatically process valid video and image information from thousands of cameras is the top priority in realizing an intelligent transportation system.

Computer vision technology can effectively understand video data and extract useful information, which can detect motor vehicle attributes, such as color, type, vehicle brand, license plate information, etc., and help traffic departments grasp real-time road conditions. For example, by using this information, the supervisory department can accurately identify the motor vehicle models on the road, which helps in dangerous vehicle monitoring, such as muck trucks or hazardous chemical vehicles. Moreover, accurate identification and positioning of these specific vehicles can help prevent traffic accidents or crime.

To obtain useful information, many researchers have used different methods to achieve vehicle detection and classification. Navastara et al. [2] used Histogram of Oriented Gradients(HOG) and Local Binary Patterns (LBP) to extract the features of vehicle, and then input them into Hierarchical Multi-SVM (HMSVM) to distinguish vehicle categories. Wei et al. [3] propose a two-step detection algorithm based on combining the features of Harr and HOG, which has higher detection accuracy and time efficiency than traditional methods. Guo et al. [4] adopted HOG method to extract vehicle type features in images, and then used Support Vector Machine (SVM) to classify these features to achieve vehicle detection.

Traditional vehicle detection methods need to manually design feature extraction methods based on experience, and this process is complicated. In addition, most of the extracted features are edge features, which cannot effectively reflect the semantic information of vehicles. With the development of deep learning, these traditional methods have been gradually replaced by deep learning techniques. Deep learning has been widely used in image processing due to its solid fitting ability and has made significant progress in recent years. The researchers have applied the object detection algorithm of deep learning to the field of vehicle detection. Object detection can be divided into two-stage detection algorithms and single-stage detection algorithms. Two-stage detection algorithms such as R-CNN [5], Fast R-CNN [6], and Fast R-CNN [7] need to generate candidate frames of the target area first and then classify and regress the candidate frames. Single-stage detection algorithms such as YOLO [8] and SSD [9] can directly predict the class and location of the object from the extracted features. In contrast, single-stage detection algorithms are fast but less accurate.

At present, many scholars apply target detection algorithms to vehicle detection. Yang et al.[10] proposed a pedestrian and vehicle detection algorithm based on the improved YOLOv2 [11]. The authors analyze the labels of the dataset to set a priori box that is more in line with pedestrians and vehicles and combine multi-scale training to improve the detection accuracy. Yang et al. [12] improved Mask R-CNN to detect pedestrians and vehicles, and built a real-time vehicle identification system. Wang et al. [13] proposed a soft-weighted method that fused RetinaNet [14] and Cascade-RCNN [15], and the results proved that this ensemble model has an excellent detection ability for overlapping objects. Alireza et al. [16] corrected the wrong labeling and unclear labels in the BITvehicle dataset and verified the higher accuracy after fixing these problems on Tiny-YOLOv3. Wang et al. [17] used Faster R-CNN in NVIDIA Jetson TK1 to realize real-time detection of vehicle type. The above detection algorithms are based on the anchor-based algorithm, which requires manual setting of the size of the prior boxes. Zheng et al. [18] introduced a version of YOLO: YOLOX, which used anchor-free algorithm and outperformed the previous versions of YOLO in both detection speed and accuracy on the coco dataset. There is currently less work associated with using YOLOX to detect vehicles.

Existing models have achieved good results in vehicle type detection, but they do not consider both detection accuracy and inference speed. This paper proposes an improved YOLOX-S detection model. The main work is as follows:

(1) Lightweight optimizations are made for the two CSPnet modules in the feature extraction network, which can improve the model accuracy and reduce the amount and complexity of model parameters.

(2) To obtain more accurate bbox regression, a new loss function, α-CIoU, is introduced. By adjusting the power of IoU and the penalty term, the YOLO detector can be more flexible to achieve different levels of the bbox regression accuracy.

2 Methodology

In recent years, various image classification methods have achieved very high accuracy on ImageNet, but the number of parameters is huge, such as Vision Transformer [19] and model soup [20]. In addition to accuracy, computational complexity is also an important indicator for evaluating models. Too complex neural networks cannot be deployed on convenient mobile devices. So more lightweight models have been proposed, such as ShuffleNetv2 [21], which summarizes four light network design guidelines. This paper aims to design a lightweight and easy-to-deploy model, so the four guidelines of ShuffleNetv2 are used as a reference when creating the network.

2.1 YOLOX algorithm

The original intention of YOLO design is for faster inference speed. The real bboxes in the previous generations of YOLO are based on regression of the priori boxes, and the setting of the priori boxes cause the generated pre-selected boxes to perform IoU with the real boxes during the training phase, which will take up a lot of memory space and time cost. In order to speed up the calculation, YOLOX adopts the anchor-free method, and selects the positive sample anchor frame through the SimOTA method, which greatly reduces the number of candidate frames and speeds up the inference speed. The current detection heads of YOLO series may lack the expressive ability. Therefore, the target classification and bbox regression information of outputs are separately through the decoupled head which not only improves the detection accuracy, but also speeds up the convergence speed of the network.

The YOLOX network can be divided into three parts: Backbone, Neck, and Head. Assuming that the input image has a size of 416×416 after the resize operation, three different sizes of feature maps will be generated: 52×52, 26× 26, and 13×13 with downsampling in backbone, and these three feature maps can detect objects of different sizes. Furthermore, the three feature layers are decoupled after FPN and PANet.

2.2 Improverd method

2.2.1 The improved network structure of YOLOX-S

Aiming at the problem that the original YOLOX-S has a large amount of network parameters and computation, which is not conducive to the deployment of terminal equipment, we have carried out a lightweight design for the backbone extraction network. The structure of the improved YOLOX-S is shown in Fig. 1.

We propose two feature extraction modules, which reduce the parameters for feature extraction and increase information fusion of the network model. Since the focus layer makes it more difficult to deploy the network on edge computing devices, it is replaced by a convolutional layer. Then the neck part is used to strengthen network feature extraction and multi-scale feature fusion, and the structure of FPN and PANet are also used in our work.

MobileNet [22] proposed the concept of depthwise separable convolution. First, feature extraction is performed by 3×3 group convolution and then 1×1 convolution is used to change the channel. Compared

with ordinary convolution, parameters can be greatly reduced. However, the introduced 1×1 convolution will still generate a certain amount of calculation. The equation for the calculation amount of convolution is as follows:

$$num = b * h * w * c * k * k \qquad (1)$$

Among them, b represents batch-size, h and w represent the height and width of the feature map separately, c represents the channel number, and k represents the kernel-size. It can be seen from the Eq. (1) that when b and c are relatively large, it will still bring a large amount of calculation. In addition, GhostNet [23] points out the output feature maps exist redundance after convolution, and there is a significant similarity between most of the feature maps, as shown in Fig. 2. When the feature maps of 16 channels are generated by convolution, it can be seen two similar feature map pair examples are annotated with same color boxes.

Therefore, GhostNet proposes Ghost Module(GM) to avoid generating these redundant feature maps, so that the amount of model parameters can be reduced while ensuring good detection accuracy. GM is the basic unit of GhostNet network, and its main function is to replace ordinary convolution. First, the channels of the original feature maps were compressed by 1×1 convolution, then GM utilizes cheap linear operations to get more feature maps. In addition, the identity mapping [24] and linear transformations are preserved in GM. After that the output feature maps has the same shape as the input feature maps, as shown in Fig. 3.

1) CSPGM module: The YOLOX network structure uses a lot of standard convolution, which brings about the problem of large computation. Therefore, this article introduces the GM, which is capable of generating feature maps with fewer amounts of parameters and calculations through efficient operation. So, referring to the GhostNet, two improved network modules are proposed: CSPGhost Module (CSPGM) and Changed Ghost Module (CGM).

The CSPGM structure is shown in Fig. 4. First, feature maps of input layer is divided into two parts, and the left part goes through the GM stacked to continue feature extraction, and then merges them through the cross-stage, which reduces the amount of computation. At the same time, the accuracy of the model can be guaranteed.

2) CGM module: In order to further fuse multi-scale global overall information and local detailed information, we use the CGM module to enhance feature extraction, which is helpful for local cross-channel information interaction and improves the accuracy of the model. The CGM structure is shown in Fig. 5, which divides the input feature maps into two branches. The left branch adjusts the output channel to half of the input through 1×1 convolution, then extracts features through 3×3 depthwise convolution, and last passes through lightweight effective channel attention module (ECA) [25], which can obtain cross-channel interaction information by increasing a few parameters and improve the network's attention to channel information. The right branch is derived from the splitting of input feature map and finally the two branches are spliced as shown in Fig. 5.

The feature maps of these two improved methods, in which input channels and output channels are equal, conform to the G1 guideline proposed by ShuffleNetv2 Among them, the Channel-split in CGM divide the feature maps into two groups, which does not increase the number of groups during convolution and conforms to the G2 guideline. The G4 guideline points out that although element-wise operators such as ReLU and Add have small FLOPs, they require a large MAC. Therefore, operations such as Add should be avoided as much as possible when designing a network.

2.2.3 The improved loss function

Bbox regression is a mainstream technique in object detection, which uses a rectangular bbox to predict the location of the target object in the image, aiming to refine predicted bboxes location. And bbox regression uses the overlap area between the predicted bboxes and the ground truth bboxes as the loss function. When the overlap doesn't exist, the gradient of loss function will disappear, which affects the model convergence speed and detection accuracy. The original model used GIoU as a positioning loss function for bbox.

GIoU introduces the minimum circumscribed rectangle of the predicted bboxes and the ground truth bbox as a penalty term. However, GIoU will degenerate to IoU when two boxes belong to the containment relationship. In order to solve the shortcomings of GIoU, this paper introduces the α-CIoU [26] loss function, which retains all the properties of the CIoU [27–29], while paying more attention to high IoU goals, and creates more space for optimizing all levels of goals, achieving different levels of detection frame regression accuracy. The formulas are as follows:

$$L_{\mathrm{CIoU}} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \beta v \qquad (2)$$

$$L_{\alpha-\mathrm{CIoU}} = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(b, b^{gt})}{c^{2\alpha}} + (\beta v)^{\alpha} \quad (3)$$

In Eq. (2), b and b^{gt} represent the center points of the detection bbox and the ground truth bbox, respectively, ρ represents the Euclidean distance between the two center points, and c represents the diagonal distance of the smallest closure region that can contain both the predicted bbox and the ground-truth bbox. v and β are respectively:

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \qquad (4)$$

$$\beta = \frac{v}{(1 - IoU) + v} \qquad (5)$$

It can be seen from Eq. (3) that when α > 1, the loss will decrease with the increase of IoU, which improves the model convergence speed and is more conducive to the target with larger IoU.

3 Experiments

3.1 Dataset

We use a public dataset made by the Beijing Institute of Technology: BITvehicle [30], which includes 9850 images, and most images have only one or two vehicles. These images are divided into Bus, Microbus, Minivan, Sedan, SUV, and Truck. The number of each label category and corresponding example picture are shown in Fig. 6 and Fig. 7, respectively.

3.2 Experimental environment & hyperparameter setting

The platform for our experiments is shown in Table 1:

Operating System	Ubuntu 18.04 LTS
CPU	Intel(R) Xeon(R) E5-2620
GPU	NVIDIA GeFore TITAN X
Memory	16G
Framework	Tensorflow, CUDA10.1

We divided the images into the training set and testing set in a ratio of 9:1, resulting in 8865 and 985 images, respectively. We adopt SGD as the optimizer, with a weight decay of 5e-4 and momentum of 0.937 as default. At the same time, the CosineAnnealing method was used to update learning rate. Due to the constraint of the memory, the batch size was set at 32.

3.3 Experimental environment & hyperparameter setting

This experiment uses Parameters, Size (MB), FPS, and mAP@0.5 as model metrics. Parameters can be used to measure the complexity of the model, and FPS is used to measure the real-time inference speed

of the model. Also, mAP@0.5 means the average AP of each category when the IoU is set to 0.5, and AP is the mean precision value on the precision-recall curve. Their calculation formulars are shown in Eqs. (6)–(8):

$$P = \frac{TP}{TP + FP} \quad (6)$$

$$R = \frac{TP}{TP + FN} \quad (7)$$

$$AP = \int_0^1 P(R)\, dR \quad (8)$$

3.4 Result

This paper introduces different innovative strategies for YOLOX-S. In order to explore the impact of these strategies on the model, the above-mentioned methods are used for ablation experiments on the previously divided BIT vehicle dataset, as shown in Table 2.

Table 2
Test results after different improvement strategy combinations

Improvement stategy	Different YOLOX-S algorithm models					
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6
Alpha-CIoU		√				√
CSPGM			√		√	√
CGM				√	√	√

The above experimental performances are shown in Table 3. Obviously, it can be seen that the accuracy of the α-IoU model is improved by 0.67%, and the FPS is slightly decreased, but this decrease is acceptable from the perspective of the improvement effect, indicating that the loss function can more accurately distinguish various types of vehicles. With the addition of CSPGM, the number of model parameters is slightly reduced, but the mAP is increased by 0.42, which indicates that the CSPGM structure has stronger feature extraction capabilities. As shown in Table 5, the amount of parameters is greatly reduced with adding CGM to the model, and the accuracy is almost unchanged, which shows that the CGM structure integrates the original feature information well. It can be clearly seen that adding all

the innovation points to the model at the same time performed well, improving both accuracy and speed.

Table 3
Comparison of overall detection performance.

Model	Input_size	Params(M)	Size(M)	mAP@0.5(%)	FPS
Experiment 1	416×416	8.96	34.7	98.22	56.5
Experiment 2	416×416	8.96	34.7	98.99	56.4
Experiment 3	416×416	8.28	32.1	98.64	58.4
Experiment 4	416×416	5.94	27.1	98.28	63.6
Experiment 5	416×416	5.26	20.6	98.54	65.7
Experiment 6	416×416	5.26	20.6	99.21	65.2

The P-R curves obtained by the model in this paper for detecting various types of vehicles are shown in Fig. 8, where the abscissa is the recall rate (Recall), the ordinate is the accuracy rate (Precision), and the shaded area is the detection accuracy (AP) of this type of defect.

This paper focuses on the realization of the light weight vehicle type detection algorithm, which provides the possibility for practical application. To demonstrate the advantages of the algorithm, we compared it with several object detection algorithms including CenterNet, YOLOv4-tiny, YOLOv5-S, and the original YOLOX-S algorithm. All experiments' training and test sets are the same, and the comparison results are shown in Table 4.

Table 4
Comparison with other detection algorithms

	mAP(%)	FPS	Size(MB)	AP(%)					
				Bus	Microbus	Minivan	Sedan	SUV	Truck
CenterNet	95.5	29.5	125.2	99.1	96.4	89.7	98.9	92.5	98.0
YOLOv4-tiny	93.7	95.3	22.6	99.4	91.6	90.4	86.1	98.3	92.9
YOLOv5-S	96.9	53.6	27.5	99.7	96.8	95.37	99.1	95.8	95.0
YOLOX-S	98.2	56.5	34.7	99.9	96.9	96.4	99.2	97.6	99.2
ours	99.2	65.2	20.6	100	99.1	99.5	99.2	97.6	99.9

As shown in Table 4, YOLOv4-tiny has the fastest detection speed but the lowest detection accuracy. CenterNet is an anchor-free detection algorithm with the largest weight file and the slowest detection speed. The mAP of YOLOv5-S is 98.2%, maintaining good accuracy and speed. Our proposed method

based on YOLOX-S has better detection accuracy almost in all categories and inference speed also faster than the other algorithm. It shows that it still has better performance under the condition of hardware constraints.

4 Conclusion

In this paper, we have proposed a lightweight network Ghost-YOLOX based on YOLOX-S algorithm model. On one hand, we have improved the network structure of YOLOX-S which includes two modules referring to GhostNet and proposed SPPFblock. On the other hand, α-CIoU is introduced to improve the convergence speed of the model and the regression accuracy of the bboxes. The experiments' result show that the mAP value and inference speed of the Ghost-YOLOX algorithm in BIT dataset are 0.99% and 12.7% higher than the original YOLOX-algorithm model. Meanwhile, the amount of parameters has been reduced by 41.2%. At traffic intersections, the algorithm can accurately identify the vehicles, but the types of identification are limited. In the future we will pay more attention to expand the number of samples in the dataset, including enriching road scenes.

Declarations

Ethical Approval

not applicable

Competing interests

not applicable

Authors' contributions

1. **Yu Anning is responsible for providing experimental platform and paper review;**
2. Xiong Cong is responsible for writing the paper and completing the experiment;
3. Senhao Yuan is responsible for the drawing of Figure 1, Figure 3-Figure 7;
4. Xinghua Gao is responsible for the production of all tables.

Funding

1. Chongqing Municipal Education Commission Science and Technology Research Project(KJQN202001523);
2. 2021 Cooperation Project between Chongqing Municipal Undergraduate Universities and Chinese Academy of Sciences(HZ2021015);
3. Chongqing University of Science and Technology Youth Science Fund Project(CKRC2019042);

Availability of data and materials

This article uses a public dataset: BIT Vehicle Dataset.

References

1. Qi L. Research on intelligent transportation system technologies and applications[C]//2008 Workshop on Power Electronics and Intelligent Transportation System. IEEE, 2008: 529-531.
2. Navastara, D. A., et al. "Vehicle Classification Based on CCTV Video Recording Using Histogram of Oriented Gradients, Local Binary Patterns, and Hierarchical Multi-SVM." *IOP Conference Series: Materials Science and Engineering*. Vol. 1077. No. 1. IOP Publishing, 2021.
3. Wei, Yun, et al. "Multi-vehicle detection algorithm through combining Harr and HOG features." *Mathematics and Computers in Simulation* 155 (2019): 130-145.
4. Guo E, Bai L, Zhang Y, et al. Vehicle detection based on superpixel and improved hog in aerial images[C]//International Conference on Image and Graphics. Springer, Cham, 2017: 362-373.
5. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
6. Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
7. Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
8. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
9. Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.
10. Yang Z, Li J, Li H. Real-time pedestrian and vehicle detection for autonomous driving[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 179-184.
11. Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.
12. Yang S, Chong X, Li X, et al. Intelligent Intersection Vehicle and Pedestrian Detection Based on Convolutional Neural Network[J]. Journal of Sensors, 2022, 2022.
13. Wang H, Yu Y, Cai Y, et al. Soft-weighted-average ensemble vehicle detection method based on single-stage and two-stage deep learning models[J]. IEEE Transactions on Intelligent Vehicles, 2020, 6(1): 100-109.
14. Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
15. Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6154-6162.

16. Taheri Tajar A, Ramazani A, Mansoorizadeh M. A lightweight Tiny-YOLOv3 vehicle detection approach[J]. Journal of Real-Time Image Processing, 2021, 18(6): 2389-2401.
17. Wang X, Zhang W, Wu X, et al. Real-time vehicle type classification with deep convolutional neural networks[J]. Journal of Real-Time Image Processing, 2019, 16(1): 5-14.
18. Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. arXiv preprint arXiv:2107.08430, 2021.
19. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
20. Wortsman M, Ilharco G, Gadre S Y, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time[J]. arXiv preprint arXiv:2203.05482, 2022.
21. Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 116-131.
22. Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.
23. Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1580-1589.
24. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
25. Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks,"2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11531-11539
26. He J, Erfani S, Ma X, et al. $\alpha $-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression[J]. Advances in Neural Information Processing Systems, 2021, 34.
27. Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 12993-13000.
28. Zheng Z, Wang P, Ren D, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation[J]. IEEE Transactions on Cybernetics, 2021.
29. Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.
30. Dong Z, Wu Y, Pei M, et al. Vehicle type classification using a semisupervised convolutional neural network[J]. IEEE transactions on intelligent transportation systems, 2015, 16(4): 2247-2256.

Figures

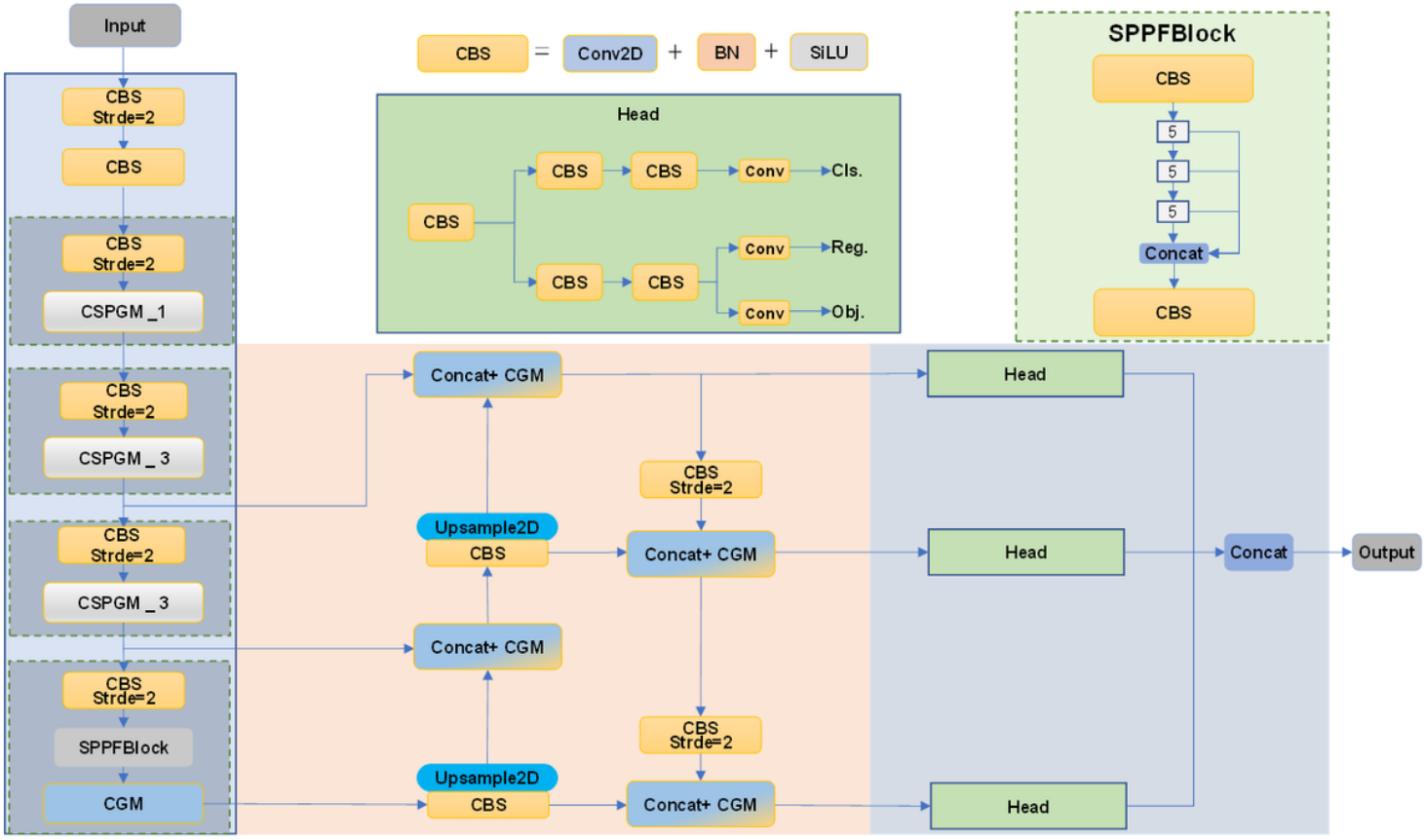

Figure 1

Improved YOLOX-S network structure

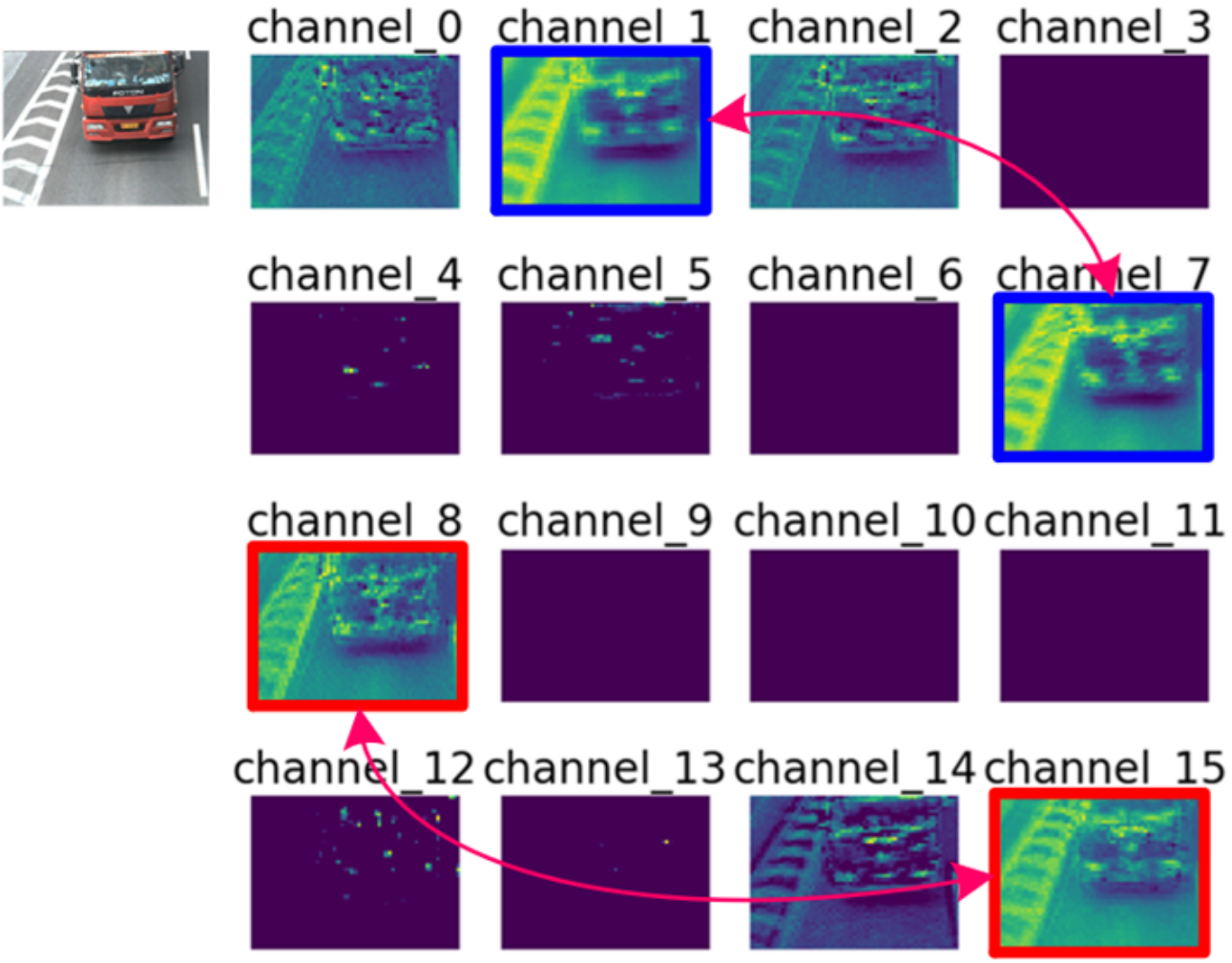

Figure 2

Redundant feature maps

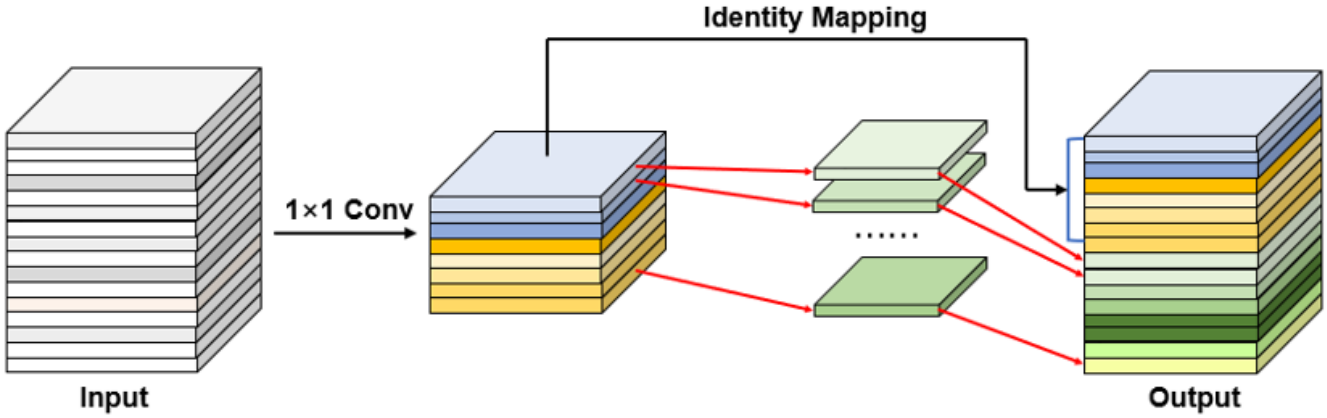

Figure 3

The Ghost Module (GM)

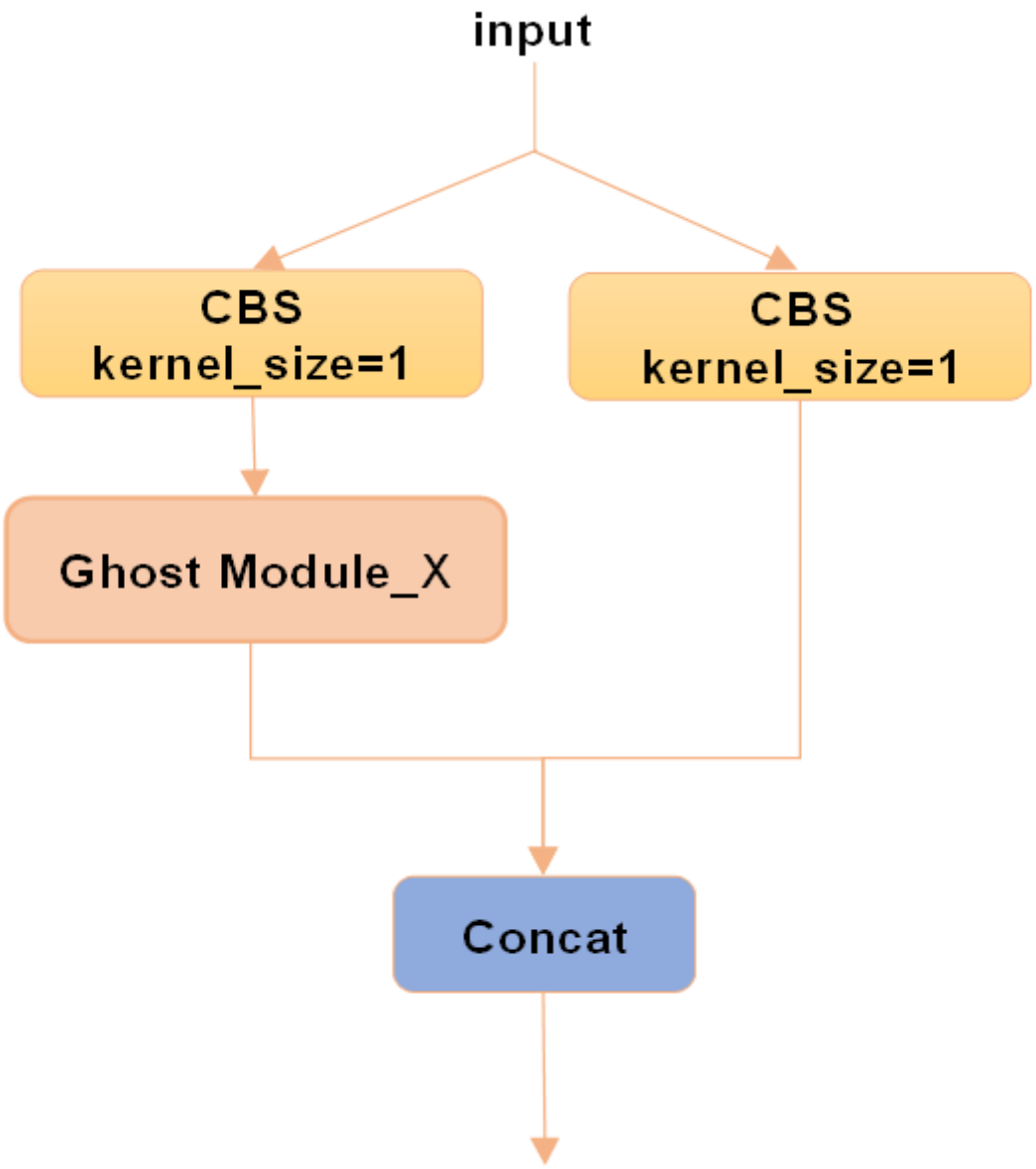

Figure 4

The CSPGhost Module (CSPGM)

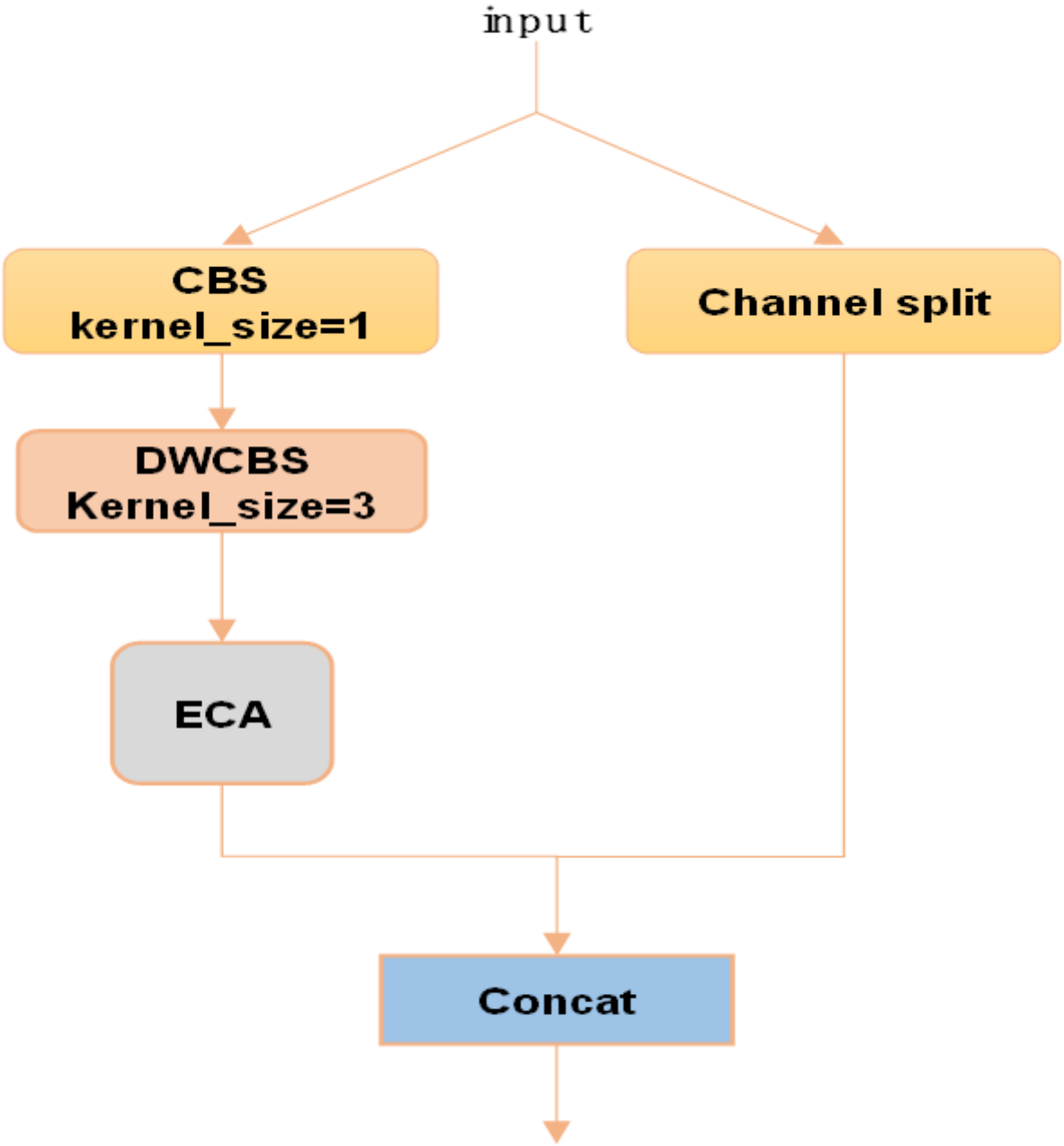

Figure 5

The Changed Ghost Module (CGM)

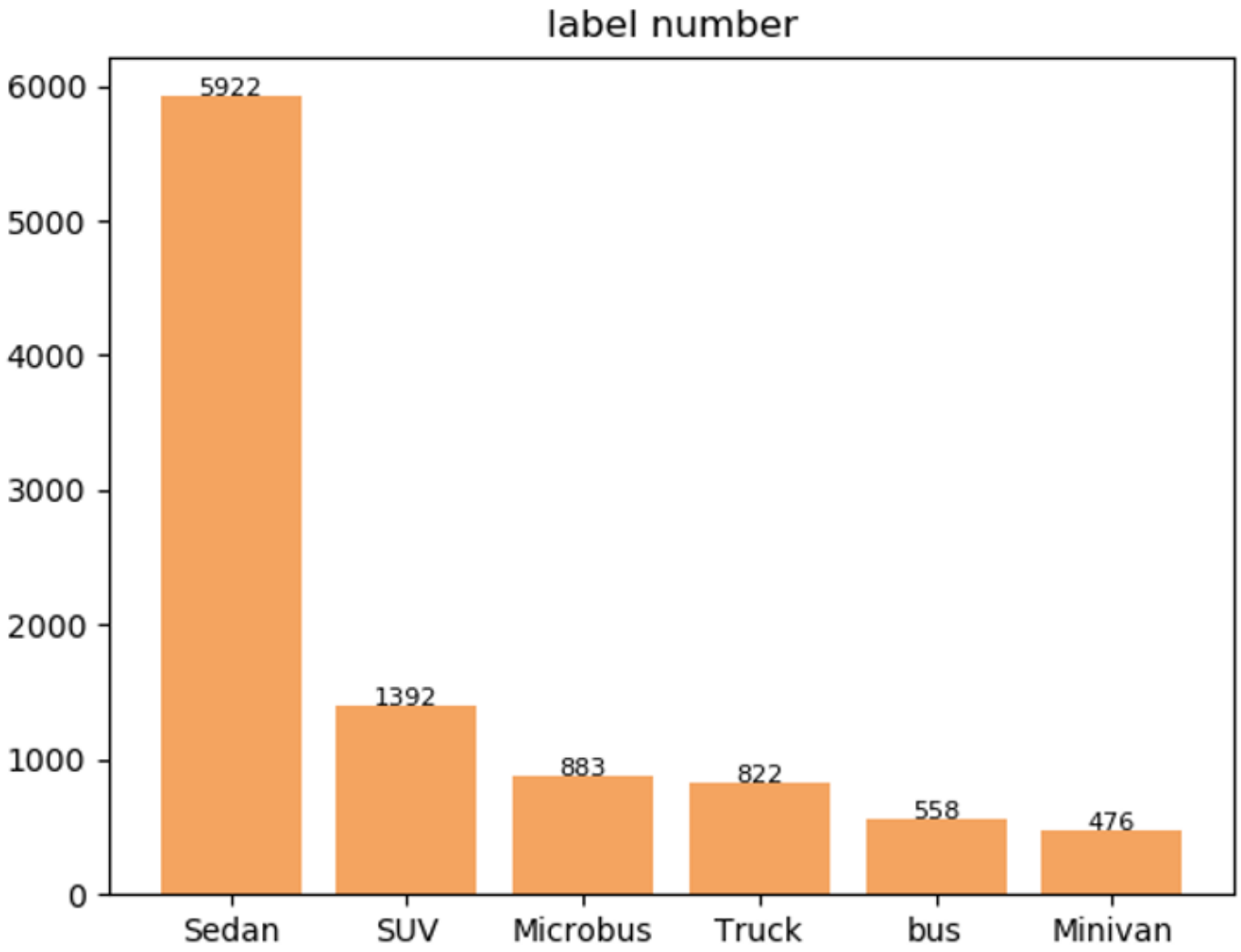

Figure 6

The number of labels

Figure 7

Some examples of the dataset

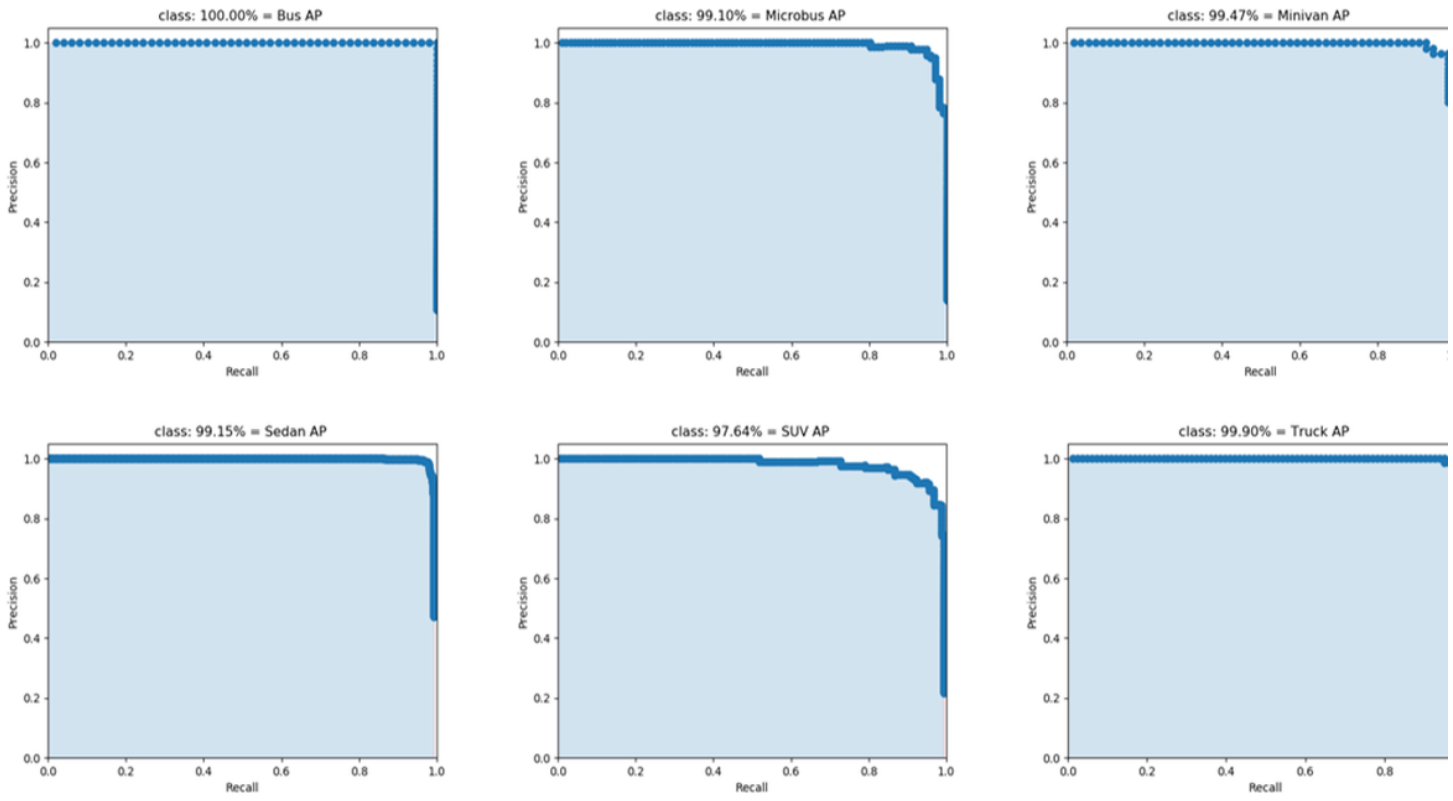

Figure 8

P-R curve