# A lightweight model for multi-traffic object detection based on deep learning under complex traffic conditions

Guoqiang Chen[1*] and Yanan Cheng[1]

[1] School of Mechanical and Power Engineering, Henan Polytechnic University, 2001 Century Avenue, Jiaozuo, Henan, China
[*] Corresponding author. E-mail: chengq@hpu.edu.cn

Received: July 18, 2021; Accepted: Sept. 17, 2021

The object detection is extremely important in autonomous driving environment awareness. Besides vehicle and pedestrian detection, traffic signs and lights are important objects. The paper presents how to achieve precise results in multi-traffic object detection while minimizing the model size. A deep learning network YOLOv5s-Ghost-SE-DW is proposed based on the YOLOv5s. The proposed network can detect all traffic objects including traffic signs and lights. First, the convolution layer is replaced by Ghost module to reduce the parameter and model size. Second, in order to improve accuracy and real-time performance, the attention mechanism SELayer is embedded to fuse more spatial features. Third, the DW convolution is used to extract features and further reduce the parameter number. The effect of different modules on the whole network is verified by ablation experiments. The YOLOv5s-Ghost-SE-DW yields a model size of 5.22MB while achieving 15.58 FPS real-time performance on CPU. The FPS increases by 27.5%.

**Keywords:** ghost module; attention mechanism; DW convolution; real-time object detection; lightweight network; complex traffic conditions



http://dx.doi.org/10.6180/jase.202206_25(3).0019

## 1. Introduction

Artificial intelligence has been a hot topic in recent years. Autonomous driving, as an important application of artificial intelligence, has been facing many challenges that scientists try to solve [1]. At present, with the combination of computer vision and deep learning, the autonomous driving environment awareness has been greatly promoted, which has become one of the mainstream methods in the object detection field. Autonomous driving is faced with a complex traffic condition, in which vehicles need accurately detect and identify every object within a limited time. The traditional object detection is divided into three steps: region selection, feature extraction and classification. Region selection is the basic step. The object may appear anywhere in the image and the object size is uncertain, so different scales and aspect ratio sliding windows are used to traverse the entire image initially. The feature quality directly affects the classification accuracy. Feature extraction is the most critical step, because it directly extracts the local image information in each window.

It is very difficult to design a robust feature due to the object size, the illumination change, the background diversity and so on. The classifier is mainly used to classify the extracted feature. However, the disadvantages of traditional object detection method are obvious. There are two main problems. First, the region selection strategy based on sliding window has no pertinence, with high time complexity and windows redundancy. Second, the hand-crafted feature is not very robust to the variety of image changes.

With the development of computer hardware and algorithms, large-scale datasets provide a benchmark for different algorithms. The deep learning technology has become the best algorithm in the field of computer vision and finally become the best method in all perceptual tasks. The

convolutional neural network has made excellent achievements in object detection and led computer vision to a new era.

## 2. Related work

### 2.1. Related work to autonomous driving

Autonomous driving covers a very wide range. At present, there are also a lot of vehicle artificial intelligence technologies. Mekki et al. [2] proposed the evolutionary game-based vehicular cloud access algorithm (EG-VCA) and the Q-learning-based vehicular cloud access algorithm. These two algorithms allow each vehicle to automatically select the way and avoid the use of a centralised controller. The experiment results showed that these two algorithms achieved good convergence. Kamalesh et al. [3] proposed a real-time pothole detection and warning system by combing Internet of Things technology. The experimental results show that the reporting success rate reaches 100%.

### 2.2. Related work to object detection

Object detection is an important application in autonomous driving. When an autonomous vehicle faces a complex traffic environment, it is important for the autonomous vehicle to detect objects and avoid obstacles timely. Dow et al. [4] combined deep learning classifier and zebra-crossing recognition techniques to improve pedestrian safety and reduce accidents at intersections. The results revealed that the proposed algorithm achieved good results on real-time performance. Object detection algorithms are mainly divided into One-Stage and Two-Stage based on deep learning. In the object detection algorithm based on Two-Stage, the convolutional neural network is used to extract features, and then Region Proposal Network [5] is used to generate candidate regions instead of sliding windows. Moreover, the classification of candidate frame regions and the preliminary prediction of object location are completed.

The convolutional neural network is also called the backbone network, and the common backbone network includes Alexnet [6], VGG [7], Resnet [8] and so on. At present, the most advanced object detection algorithm includes Faster R-CNN [9], PVANet [10], R-FCN [11], MR-CNN [12], Cascade R-CNN [13], Soft-NMS [14] and so on. Among them, the following networks achieve better results by optimizing each component of Faster R-CNN. PVANet uses better backbone network Inception [15]. HyperNet [16], and residual modules to find more robust features and accelerate performance. MR-CNN uses a more accurate Region Proposal Network to transform the region before inputting, pays more attention to its context information and completes better screening and recommendation of the region. Cascade R-CNN uses a more complete ROI classification and cascade regression as a resampling mechanism to gradually increase the value of IOU to achieve better detection results. Soft-NMS improves the detection accuracy rate by post-processing samples, modifying the rules of deleting detection boxes in NMS, merging and filtering candidate boxes.

Compared with Two-Stage algorithms, the convolution neural network is also used to extract features without Region Proposal Network in One-Stage algorithms. Faster detection speed is achieved by inputting the features into the regression network to directly return the class probability and object bounding box. Therefore, a One-Stage detection framework is proposed based on regression. But at the same time, the speed is given priority while detection accuracy is lost. The YOLO [17–20] is one of the most classic One-Stage object detection algorithms. The whole algorithm adopts the end-to-end network to evaluate the whole image by regression, so that the real-time performance can be guaranteed reliably. The YOLO algorithm divides the input image into S×S grids, and regresses each grid to output the category and bounding box of the object area corresponding to the current grid. The framework based on regression is a new object detection framework. Since the YOLO algorithm divides the input image into S×S grids, the positioning accuracy of the algorithm was not as high as that based on Region Proposal Network. SSD [21] combines high detection accuracy of Two-Stage algorithms with high speed of One-Stage algorithms and applies Region Proposal Network. SSD adds the anchor mechanism of Faster-RCNN on the basis of YOLO and multi-scale region features to regress the object location and bounding box. In terms of accuracy, it achieves precise accurate rate as good as region proposal, while ensuring real-time performance. Although, the position of the bounding box predicted by SSD is not very ideal for small objects, it is a relatively advanced One-Stage object detection algorithm to perform well in real-time performance without sacrificing too much detection accuracy.

In summary, significant progress has been achieved in object detection. However, the existing model still has some defects in detection accuracy and speed, especially under the multi-scale, high light, rainy, shadow and other complex environment conditions. Consequently, from the perspective of generalization and real-time performance, to further improve the detection accuracy and speed, it is extremely important to combine the lightweight module and the convolutional neural network to reduce the model size and improve real-time performance. Accordingly, the objective of the study is to explore a more lightweight and

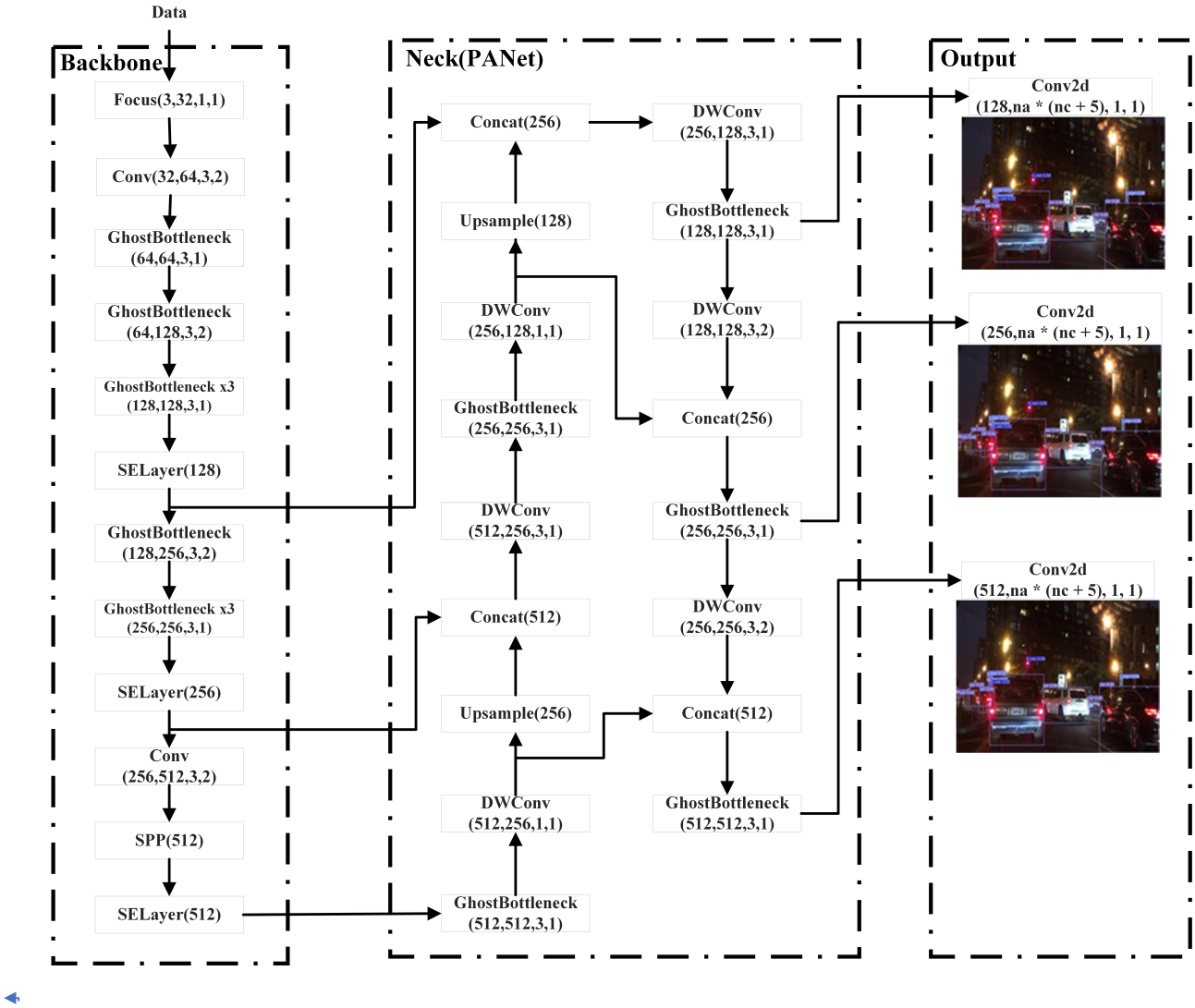


**Fig. 1.** Whole network structure

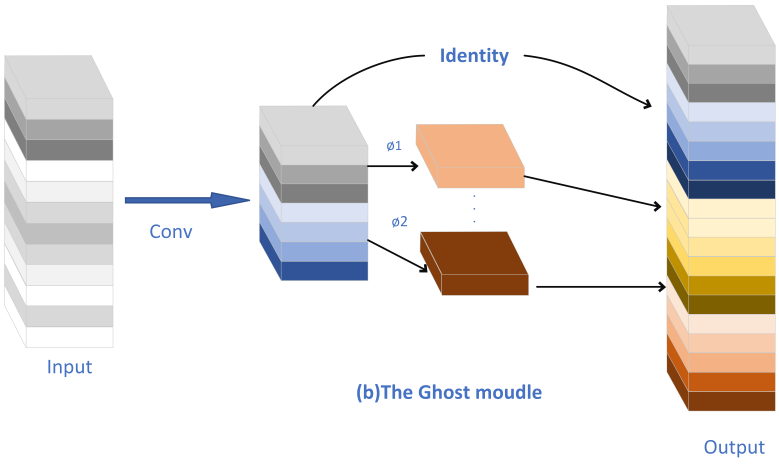


**Fig. 2.** Ghost module

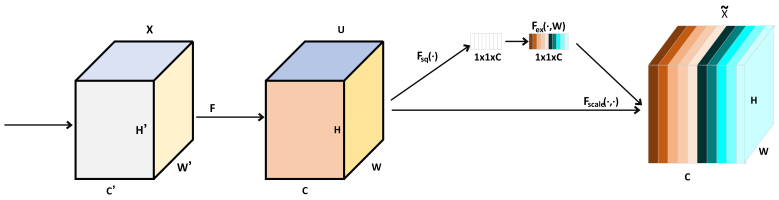


**Fig. 3.** SELayer module

accurate method for multi-traffic object detection.

The paper is organized as follows. The related work in multi-traffic object detection is reviewed in Section 2. The whole network structure and different modules are given in Section 3. The ablation experiments and results analysis are showed in Section 4. Finally, the paper is summarized in Section 5.

## 3. Methodology

The original idea is to reduce the network model size while keeping detection accuracy. The lightweight module and the attention mechanism SELayer [22] were used to reduce the model size and keep the detection accuracy.

### 3.1. Motivation

Building a smaller, more efficient and accurate network model has become a hotspot topic. At present, the lightweight module is mainly designed for mobile devices. It was deployed to the mobile device through reasonable connections with some existing modules, such as dense connection [23], residual module and group convolution. However, the mobile device requires high real-time performance and detection accuracy. In order to achieve this target, it is necessary to reduce the whole network parameter number reasonably rather than delete the convolution layers blindly. The main goal is to reduce the model size and achieve good detection speed and accuracy. The Ghost module in GhostNet generates the same feature map number with fewer parameters and reduces the model size, but the detection accuracy is decreased. In order to improve the detection accuracy, the attention mechanism SELayer was used to build the relationship between the feature channel and the feature space. It can learn the importance of each channel automatically and suppress the information that is not important to the current task. So the attention mechanism SELayer is embedded reasonably.

### 3.2. The structure of YOLOv5s-Ghost-SE-DW

The whole network structure is shown in Fig. 1. Based on the YOLOv5s, the new modules are reasonably embedded. The Ghost module and the attention mechanism SELayer is added to the backbone network respectively. The Ghost module of 1×1 is used to replace BottleneckCSP. The Ghost module of 2×2 is used to replace the convolution layer for down-sampling and feature extraction. After extracting features from images, the attention mechanism SELayer is used to obtain the importance of each feature channel. For a color image with three channels, the features are first extracted from the backbone network, and then the features are inputted into the PANet network. In order to further reduce the parameter number, the DW convolution is used instead of the ordinary convolution. Finally, the input feature maps are processed by up-sampling and down-sampling to predict the category and bounding box. The four-number set, for example, Focus (3,32,1,1), in Fig. 1 represents the input channel, the output channel, the convolution size and the step size; A number set, for example, SELayer(128), represents the output channel. Up-sampling uses the nearest interpolation and double sampling. × N means the module is stacked N times.

#### *3.2.1. Ghost module embedding*

To reduce the network model size, the usual method is model compression and pruning. The model is compressed to make the model carry fewer parameters, so as to solve the memory and speed problem. However, the lightweight module focuses on the convolution method to design more efficient networks. By optimizing and improving the convolution calculation in the network, the network parameter number can be reduced without losing the network performance. Inspired by the lightweight network GhostNet, the original backbone network BottleneckCSP module in the YOLOv5s was replaced by the Ghost module with the step size 1 and the step size 2 respectively. The Ghost module is shown in Fig. 2. The Ghost module mainly uses fewer parameters to generate the same feature map numbers. It requires less computation cost and memory than the ordinary convolution layer. Integrating into the existing convolution neural network can reduce the computation cost and the parameter number. In the Ghost module, the feature maps are convoluted to get the $H' \times W' \times m$ feature maps, and then the m channel feature maps are mapped

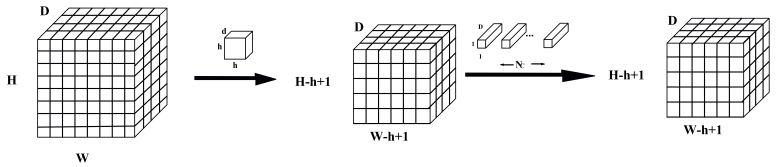


**Fig. 4.** DW convolution

linearly. The GhostBottleneck consists of two stacked Ghost modules. The first Ghost module is used as an extension layer to increase the channel number. The second Ghost module reduces the channel number.

#### 3.2.2. *SELayer embedding*

In order to maintain the network detection accuracy, in addition to adding Ghost moule, the attention mechanism SELayer is also added to the backbone network. Essentially, the convolution is feature fusion on a local area that includes space and channel. However, the convolution does not consider the relationship between the feature channel and the feature space because the attention mechanism SELayer takes it into account. In the SELayer module, the spatial dimension is compressed first by squeeze operation to get the 1×1×C feature maps. That is to say, each feature map is pooled globally and averaged into a value. Then, the importance of each channel is predicted by using the full connection layer to get the importance of different channels by excitation operation. Finally, a weighting operation is performed to multiply the activation values of each learned channel to promote important features and suppress unimportant features. The SELayer module is shown in Fig. 3

#### 3.2.3. *PANet modification*

To further reduce the parameter number in the PANet network, the ordinary convolution layers were replaced by the DW convolution layer. The DW convolution is divided into two steps, the Depthwise convolution and the Pointwise convolution. The Depthwise convolution operates in a two-dimensional plane, and the convolution kernel number corresponds to the depth. Each channel of the input layer is convolved independently. The feature information of different channels is not effectively utilized in the same spatial location. So the Pointwise convolution, with the kernel size of 1×1××M (where M represents the depth of the upper output), is used to combine the previous feature maps in the depth direction. The DW convolution is shown in Fig. 4.

For an ordinary 2D convolution (where stride=1 and padding=0), the input size is H×W×D, $N_c$ represents the convolution kernel number, the kernel size is h× h × D, and h is an even number. For a random image with three channels, the ordinary convolution parameters are $P_{conv}$ and the DW convolution parameters are $P_{DWconv}$.

$$P_{\text{conv}} = (H-h+1)\times(W-h+1)\times N_C \times h \times h \times D \quad (1)$$

$$P_{DWconv} = (H-h+1)\times(W-h+1)\times N_C \times h \times h \times D + N_C \times 1 \times 1 \times D \times (H-h+1)\times(W-h+1) \quad (2)$$

$$\frac{P_{DW\,\text{conv}}}{P_{conv}} = \frac{1}{N_c} + \frac{1}{h^2} \quad (3)$$

During convolution, there are many output channels, and the $N_c$ number is much greater than h. Therefore, for ordinary 2D convolution, it takes $h^2$ times longer than DW convolution, and the DW convolution reduces not only the training time but also parameters number.

## 4. Experiment

### 4.1. Experimental settings

All experiments in this study are based on the pytorch deep learning framework and are trained on a single NVIDIA GTX2080Ti. Other configurations include Ubuntu 18.04, CUDA 10.1, deep learning acceleration library CUDNN v7.6 and Python interface environment.

### 4.2. Experimental datasets

For autonomous driving environment awareness, the large-scale datasets provide the foundation for training and benchmarks for different algorithms. KITTI [24] and Cityscapes [25] datasets are widely used in the autonomous driving field, such as vision range finding, 3D object detection and 3D tracking. However, the dataset mainly lays the foundation for algorithms, ignoring the problem of long-distance driving. BDD100K is one of the largest autonomous driving video datasets, which has a diversity of geography, environment and weather [26]. Compared with other datasets, it also has more comprehensive labels for object occlusion and truncation. In addition to pedestrian and vehicle detection, the datasets have been improved

**Table 1.** Comparison of different datasets

| Conditions / Name | KITTI | Cityscapes | BDD100K |
|---|---|---|---|
| #Sequence | 22 | 50 | 100,000 |
| #Images | 14,999 | 5000(+2000) | 120,000,000 |
| Multiple Cities | No | Yes | Yes |
| Multiple Weathers | No | No | Yes |
| Multiple Times of Day | No | No | Yes |
| Multiple Scene types | Yes | No | Yes |

**Table 2.** Dataset training and test samples distribution

| Datasets | Train | Valid | Test |
|---|---|---|---|
| BDD100K | 69863 | 10000 | 2000 |

to simulate the external environment in real-word conditions. It is divided into ten categories: Bus, Light, Sign, Person, Bike, Truck, Motor, Car, Train and Rider. In order to achieve more precise classification, we classify the object in the entire datasets into 13 categories and divided the Sign into tl_red, tl_yellow, tl_green, tl_none and t_sign. Table 1 shows the comparison between different datasets, and Table 2 shows the number of the modified datasets.

The epochs are 300. The learning rate is 1×10-2. The batch size is 32. The image input size is 640×640 and the training phase is reduced by random gradient with a momentum term of 0.937. These experiments show that our improved algorithm achieves good results on the BDD100K dataset. Furthermore, the model size is about one third of the original model.

### 4.3. Ablation experiments

To clearly show the impact of different modules, the results were compared through ablation experiments. Table 3 compares different modules from four aspects: model size, mAP (%)0.50, mAP (%)0.50:0.95 and FPS. The Ghost module reduces the model size without decreasing the detection accuracy significantly from Table 3. By adding the attention mechanism SELayer and DW convolution, the model size is decreased from 9.64MB to 5.22MB; and mAP (%) 0.50 and mAP (%) 0.50:0.95 are reduced by 6.9% and 3.25% respectively.

### 4.4. The comparison of real-time performance

The reference point of measuring real-time performance is FPS, which depends on the test images and video resolutions. With GeForce GTX 2060, we measured the total time to test 2000 images on the BDD100K datasets. The ghost module was mainly designed for mobile devices, and we also use CPU to adopt the same operation. The average value of testing time by 30 times is shown in Table 4. On GPU, the overall test time of the YOLOv5s-Ghost-SE-DW is longer than the YOLOv5s. On CPU, compared with the YOLOv5s, the time is reduced by 22.95 seconds. The FPS reaches 15.58. The real-time performance is increased by 27.5%.

The entire network is modified based on YOLOv5s. The different modules were added to verify the hypothesis through ablation experiments. Experiments show that our network performs well on the improved BDD100K datasets. The model size is 5.22MB and one third of the original model. The average time to test a single image on the CPU is 0.0629s. The entire network model is small, which reduces the deployment cost. Our model still achieves good detection results in the face of complex traffic conditions including multi-scale, multi-object, rainy, night, snowy, high light and so on shown in Fig. 5.

## 5. Conclusions

In this study, The YOLOv5s-Ghost-SE-DW algorithm is proposed based on YOLOv5s. First, in order to reduce the model size, the Ghost module is introduced to replace the original module to extract features and reduce the parameter number. Second, the attention mechanism SELayer is embedded reasonably to fuse more features in space and extract multi-scale spatial information to improve the detection accuracy. Third, the DW convolution is proposed to further reduce the model size. Finally, the different module is verified by ablation experiments on the improved BDD100K datasets. Through training on the improved BDD100K datasets, the YOLOv5s-Ghost-SE-DW algorithm performs better and can accurately identify the whole object in complex traffic conditions.

## Acknowledgements

This work is supported by Fundamental Research Funds for the Universities of Henan Province (No.NSFRF200401), the Key Technology R&D Program of Henan Province of China (No. 212102210045, No. 182102310706) and National Natural Science Foundation of China (No. U1304525).

**Table 3.** Performance results of different modules on the BDD100K dataset

| | YOLOv5s | | | YOLOv5s-Ghost-SE-DW |
|---|---|---|---|---|
| Ghost module | | ✓ | ✓ | ✓ |
| Ghost module | | | ✓ | ✓ |
| DWconv | | | | ✓ |
| Model Size (MB) | 14.1 | 9.64 | 9.72 | 5.22 |
| mAP (%)0.50 | 0.4657 | 0.4053 | 0.41 | 0.396 |
| mAP (%)0.50:0.95 | 0.2312 | 0.1919 | 0.1939 | 0.1987 |
| FPS | 65.77 | 75.38 | 69.5 | 34.87 |

**Table 4.** BDD100K dataset performance on GPU and CPU

| | | YOLOv5s | YOLOv5s-Ghost-SE-DW |
|---|---|---|---|
| GPU | Average test time(s) | 65.92 | 91.67 |
| | FPS | 65.77 | 34.87 |
| CPU | Average test time(s) | 186.15 | 163.20 |
| | FPS | 12.22 | 15.58 |

(a) MuLti-object (b) Multi-scale (c) Rainy

(d) Night (e) General environmrnt (f) Complex environment

(g) High light & Night (h) Night & Snowy (i) Snowy

**Fig. 5.** Performance of our model under complex traffic conditions

## NOMENCLATURE

Alexnet Alex Networks
Cascade R-CNN Cascade Region-Convolution Neural Networks
Faster R-CNN Faster Region-Convolution Neural Networks
MR-CNN Multi-Region Convolution Neural Networks
PVANet Performance Vs Accuracy Networks
R-FCN Region-Fully Convolutional Networks
Resnet Residual Network
SELayer Squeeze and Excitation Layer
Soft-NMS Soft-Non Maximum Suppression
SSD Single Shot Detector
VGG Visual Geometry Group
YOLOv1 You Only Look Once v1
YOLOv2 You Only Look Once v2
YOLOv3 You Only Look Once v3
YOLOv4 You Only Look Once v4

## References


[1] S. K. Mishra and S. Das. "A Review on Vision Based Control of Autonomous Vehicles Using Artificial Intelligence Techniques". In: *2019 International Conference on Information Technology (ICIT)*. IEEE. 2019, 500–504.

[2] T. Mekki, I. Jabri, A. Rachedi, and M. B. Jemaa, (2019) *"Vehicular cloud networking: evolutionary game with reinforcement learning-based access approach"* **International Journal of Bio-Inspired Computation *13***(1): 45–58.

[3] M. Kamalesh, B. Chokkalingam, J. Arumugam, G. Sengottaiyan, S. Subramani, M. A. Shah, et al., (2021) *"An Intelligent Real Time Pothole Detection and Warning System for Automobile Applications Based on IoT Technology"* **Journal of Applied Science and Engineering *24***(1): 77–81.

[4] C.-R. Dow, H.-H. Ngo, L.-H. Lee, P.-Y. Lai, K.-C. Wang, and V.-T. Bui, (2020) *"A crosswalk pedestrian recognition system by using deep learning and zebra-crossing recognition techniques"* **Software: Practice and Experience *50***(5): 630–644.

[5] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, (2015) *"What makes for effective detection proposals?"* **IEEE transactions on pattern analysis and machine intelligence *38***(4): 814–830.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, (2012) *"Imagenet classification with deep convolutional neural networks"* **Advances in neural information processing systems *25***: 1097–1105.

[7] K. Simonyan and A. Zisserman, (2014) *"Very deep convolutional networks for large-scale image recognition"* **arXiv preprint arXiv:1409.1556**:

[8] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778.

[9] S. Ren, K. He, R. Girshick, and J. Sun, (2016) *"Faster R-CNN: towards real-time object detection with region proposal networks"* **IEEE transactions on pattern analysis and machine intelligence *39***(6): 1137–1149.

[10] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, (2016) *"Pvanet: Deep but lightweight neural networks for real-time object detection"* **arXiv preprint arXiv:1608.08021**:

[11] J. Dai, Y. Li, K. He, and J. Sun. "R-fcn: Object detection via region-based fully convolutional networks". In: *Advances in neural information processing systems*. 2016, 379–387.

[12] S. Gidaris and N. Komodakis. "Object detection via a multi-region and semantic segmentation-aware cnn model". In: *Proceedings of the IEEE international conference on computer vision*. 2015, 1134–1142.

[13] Z. Cai and N. Vasconcelos. "Cascade r-cnn: Delving into high quality object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 6154–6162.

[14] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. "Soft-NMS-improving object detection with one line of code". In: *Proceedings of the IEEE international conference on computer vision*. 2017, 5561–5569.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, 1–9.

[16] T. Kong, A. Yao, Y. Chen, and F. Sun. "Hypernet: Towards accurate region proposal generation and joint object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 845–853.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 779–788.

[18] J. Redmon and A. Farhadi. "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 7263–7271.

[19] J. Redmon and A. Farhadi, (2018) *"Yolov3: An incremental improvement"* **arXiv preprint arXiv:1804.02767**:

[20] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, (2020) *"Yolov4: Optimal speed and accuracy of object detection"* **arXiv preprint arXiv:2004.10934**:

**[21]** W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, 21–37.

**[22]** J. Hu, L. Shen, and G. Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 7132–7141.

**[23]** G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 4700–4708.

**[24]** A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, 3354–3361.

**[25]** M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 3213–3223.

**[26]** F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. "Bdd100k: A diverse driving dataset for heterogeneous multitask learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, 2636–2645.