ISSN: 2586-7652 (Print) Vol. 03, No. 02, March 2020
ISSN: 2635-7607 (Online)
International Journal of Advanced Engineering
Source: http://ictaes.org
Manuscript received: July 26, 2020; Revised: August 30, 2020; Accepted: September 13, 2020

# Fine Tuning YOLOv4 for Vehicle Detection

Sandesh Shrestha[1], Rabindra Phoju[2]

[1]*Department of Computer Engineering, Khwopa Engineering College, Nepal*
[2]*Department of Electronics and Communication Engineering, Khwopa Engineering College, Nepal*

*shrestha.sandesh@khec.edu.np[1], rabinesn@khec.edu.np[2]*

***Abstract***

*In order to cope with the traffic congestion problem at highways, real time vehicle detection could prove to be a big deal. After detecting the type of automobiles according to their weights, an automatic system could be developed to automate the routing the vehicles as per the category. Such system could replace the mundane human effort. Additionally, real time vehicle detection could be applied for automated parking system, and tollway ticketing and many more. One of the fastest and accurate state of the art object detection system is YOLOv4 which is able to classify 80 different objects. But for the application specific purpose, it is required to modify and re-train the network on custom dataset. This project aims to make the YOLOv4 algorithm detect only three different vehicle classes. In order to show the feasibility and ampleness of the system, this research presents exploratory results of the vehicle detection following this architecture and few experiences on convenient executions.*
***Keywords:*** *YOLOv4, Object Detection, Computer Vision, Convolutional Neural Network, Vehicle Detection.*

## 1. Introduction

Object Detection is a sub-field of computer vision that is changing the manner in which we distinguish objects. Object detection and is a significant and challenging task in numerous computer vision applications, for example, surveillance, vehicle detection and navigation. Object detection varies from image classification- the former identifies a specific object and draws a bounding box around it rather than just solving the classification problem. There can be various bouncing boxes in a picture, identifying different objects in the picture.

There are two different techniques to object detection: 1.) Machine learning-based technique, and 2.) Deep learning-based technique. For the machine learning approach, it is a necessary to define features, then using techniques for classification. Viola, Jones at al proposed the Viola-Jones algorithm which uses boosted cascade of simple Haar features [1]. Scale-invariant feature transform (SIFT) [2] and Histogram of oriented gradients (HOG) [3] are also the techniques for feature extraction. For classification, Support Vector Machines (SVM) are

Corresponding Author : Rabindra Phoju
Author's affiliation : Department of Electronics and Communication Engineering, Khwopa Engineering College
Email: rabinesn@khec.edu.np
Copyright © ICT-AES

deployed. For the deep learning approach which perform end-to-end object detection, it is not necessary to define features.

Convolutional Neural networks (CNN) is a class of deep neural networks, which find application mainly in image visualization. Zhao et al provided a review of object detection employing a deep learning approach [4]. Region-based CNN algorithms (RCNN, Fast R-CNN, Faster R-CNN), Single Shot MultiBox Detector (SSD), YOLO are some deep learning-based approaches to object detection. A standard CNN is not suitable for object detection, due to the variable length of the output layer, as there can be multiple occurrences of the same object in an image. One way to approach the matter of multiple occurrences of an object in a picture is that the division of the image into completely different regions of interest and employing a CNN for classification of an object's presence in that region. But the matter with this approach is that the objects of interest tend to possess totally different spatial locations within the image. This demands for high computation due to a large number of regions of interest. R-CNN algorithm, proposed by Girschick et al [5] solves the problem of a large number of regions of interest. They proposed a method of extracting 2000 regions from the image, called as region proposals, using selective search, and classifying in those region proposals only instead of a large number of regions. Fast R-CNN improvised the previous RCNN algorithm by generating a convolutional feature map by directly feeding the image to the CNN [6]. Different region proposals are known from the feature map. This significantly reduced the time to test an image compared to the R-CNN. Additionally, by removing the selective search and permitting the network to find out the region proposals, Shaoqing et al developed the Fast R-CNN. They named it Faster R-CNN, because it takes even less time than the previous Fast R-CNN for checking the image [7]. A typical object detector architecture consists of four modules- the input, the backbone, the head, and the neck. The backbone is the pre-trained network which takes the input image performs feature extraction. The head predicts the objects' categories and bounding boxes. The neck serves the aim of accelerating robustness by grouping feature maps from intermediate stages. These algorithms can be trained and further optimized for the problem of detecting vehicles. In this project, YOLOv4 is used for object detection and transfer learning was applied for detecting vehicles of only three different classes. YOLOv4 provides higher accuracy and faster results so as to implement real-time vehicle detection.

## 2. Review on YOLO

Joseph Redmon et al shaped a brand-new algorithmic program in 2015 known as You Only Look Once (YOLO) [8]. In this algorithm, a single convolutional network predicts the classes and bounding boxes for the whole image in one run of the algorithm, unlike the region-based classification algorithms. The input image is split into S x S grid, and every grid is liable for the detection of an object. In each grid, m bounding boxes are taken, and for each bounding box, an offset value for bounding box and class probability is provided by the network. The bounding boxes having a class probability above a particular threshold are chosen and are further used to locate the object in the image.

In their paper for YOLOv3, Redmon et al showed a huge improvement over YOLOv2 [10] [9]. In various techniques, YOLOv3 improvised over its predecessor. Firstly, by utilizing dimension clusters as anchor boxes, it improved the bounding box prediction. The network then used logistic regression to predict an abjectness score for each bounding box. A threshold is set for each classifier, and classes with higher scores than the threshold are assigned to the bounding box. Secondly, for better class prediction, independent logistic classifiers replaced the SoftMax layer. Then, each box predicts the classes. The bounding box may have multiclass classification. Also, the number of convolutional layers is increased to 53, using the Darknet-53 framework, which is the backbone behind YOLOv3. The network is a blend of YOLOv2, Darknet-19, which consists of 3 x 3 and 1 x 1

filters along with shortcut connections. Finally, it makes predictions across three totally different scales, that is comparable to the Feature Pyramid Network (FPN) [11]. It adds up some convolutional layers from the base feature extractor. Three bounding boxes are predicted at each scale. This ultimately provides better features from early in the network, and the predictions at the last layers are benefited from the early computations. Lecheng Ouyang et al enforced vehicle target detection based upon YOLOv3 in complicated scenes and showed benefits over conventional target detection algorithms in terms of accuracy and speed [12]. They showed in which manner YOLOv3 can be used for vehicle detection, and it offers an accuracy of 89.16% at 21fps on the VOC dataset. YOLOv3 is very quick, which runs in 22ms at 320 x 320 with 28.2 mAP. In spite of being 3 times faster than RetinaNet [13], its accuracy is lower.

Bochkovskiy et al put forward the YOLOv4 algorithmic program with important changes from the previous version, and far higher accuracy [14]. Published in April 2020 it's the most recent, most advanced iteration of YOLO. They designed a deeper and complicated network, wherever they used Dense Block to achieve higher accuracy. It consists of multiple convolutional layers, with batch normalization and ReLU. Further, a Dense Net is created which consists of multiple Dense Blocks with transition layers in between [15]. Then, the input feature maps of Dense Blocks are separated into 2 components by Cross-Stage Partial connections (CSP) [16]. One half goes on to the input to succeeding transition layer, and therefore the different half can undergo the Dense Block. This reduces the computational needs as only one half goes through the Dense Block. For the backbone of the feature extraction, CSPDarknet-53 has been used, which uses the CSP connections beside Darknet-53 from the previous YOLOv3. Spatial Pyramid Pooling (SPP) is employed as the neck over CSPDarknet-53 because it expands the receptive field, differentiates the foremost important feature and doesn't reduce the speed [17]. Here they used Path Aggregation Network (PANet) in place of the Feature Pyramid Network (FPN) in YOLOv3 [18]. For the head, the original YOLOv3 network has been used. Apart from the architecture, it consists of a training strategy called as the Bag of Freebies to induce higher accuracy without the additional cost to hardware. With these, better performance can be obtained for free. Another set of methods that provide higher results at low cost, however not fully free, were titled the Bag of Specials. There are some extra enhancements on design for a robust training on a single GPU just like the introduction of latest data augmentation strategies like CutMix, Mosaic, etc., selection of hyper-parameters while applying general algorithms, and modifications to existing methods to make their design suitable for better detection and training. The YOLOv4 creators have selected CSP darknet 53 backbone, SPP additional module, PANet path aggregation neck, Greedy NMS and YOLOv3 (anchor-based) head as the architecture of YOLOv4.

## 4. Methodology

Total of 4972 images containing different types of vehicles in Kathmandu valley of different lighting conditions were extracted from the CCTV camera footage. All of the vehicles in the collected images were properly labeled in YOLOv4 format. The dataset was split into training set and validation set in 80:20 proportion. For testing, the real-time videos during six different conditions i.e. day, night, evening, fog and rain were used and for calculating mAP on the test dataset, 50 images of each conditions were used for evaluation. The complete flowchart of our methodology is shown in Figure 1.

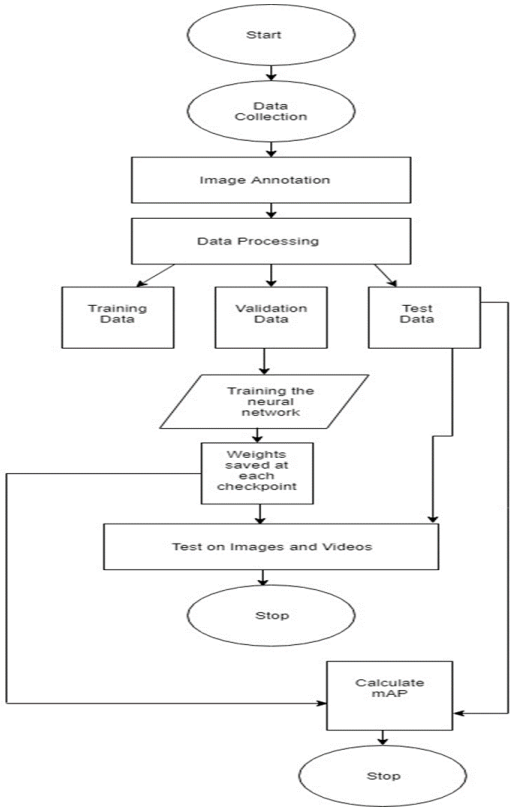



**Figure 1**. Flowchart of the proposed system

## 3. Proposed refinement in YOLOv4

We applied transfer learning technique to the new state-of-the-art object detection algorithm YOLOv4, to specifically suit the needs of vehicle detection, using some custom dataset. Transfer learning is basically used to learn new tasks from the previously learned tasks. It requires comparatively less training data than training from the scratch and the learning process can be more accurate. In transfer learning technique, the weights of the convolutional layers are fixed to that of the previously trained model and the weights of dense layers are updated by feeding new data. Figure 2 shows the basic concept of transfer learning. In our system, using transfer learning on the pretrained YOLOv4 model trained on COCO dataset which originally can detect 80 different classes, three vehicle categories (light weight vehicles, medium weight vehicles and heavy weight vehicles) were detected.

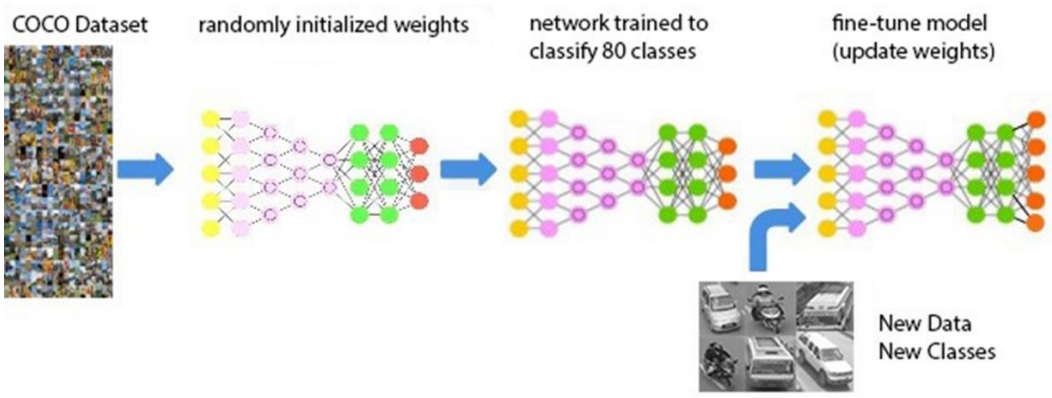



**Figure 2**. Transfer Learning

## 5. Experimentation

The system was trained with the custom dataset using transfer learning with the pretrained model initially trained on COCO dataset for 6000 iterations and the steps were 4800, 5400. The batch size was set to 64 and subdivisions were 16. The image size was set to 416x416 pixels in RGB mode. The momentum and decay were 0.949 and 0.0005 with the learning rate of 0.001. The weights of the model were saved at the end of each 1000 iterations. The training loss at the end of 6000th iteration was 0.3737 and the mean average precision (mAP) value was obtained to be 99.2% as obtained in the training chart shown in Figure 3.

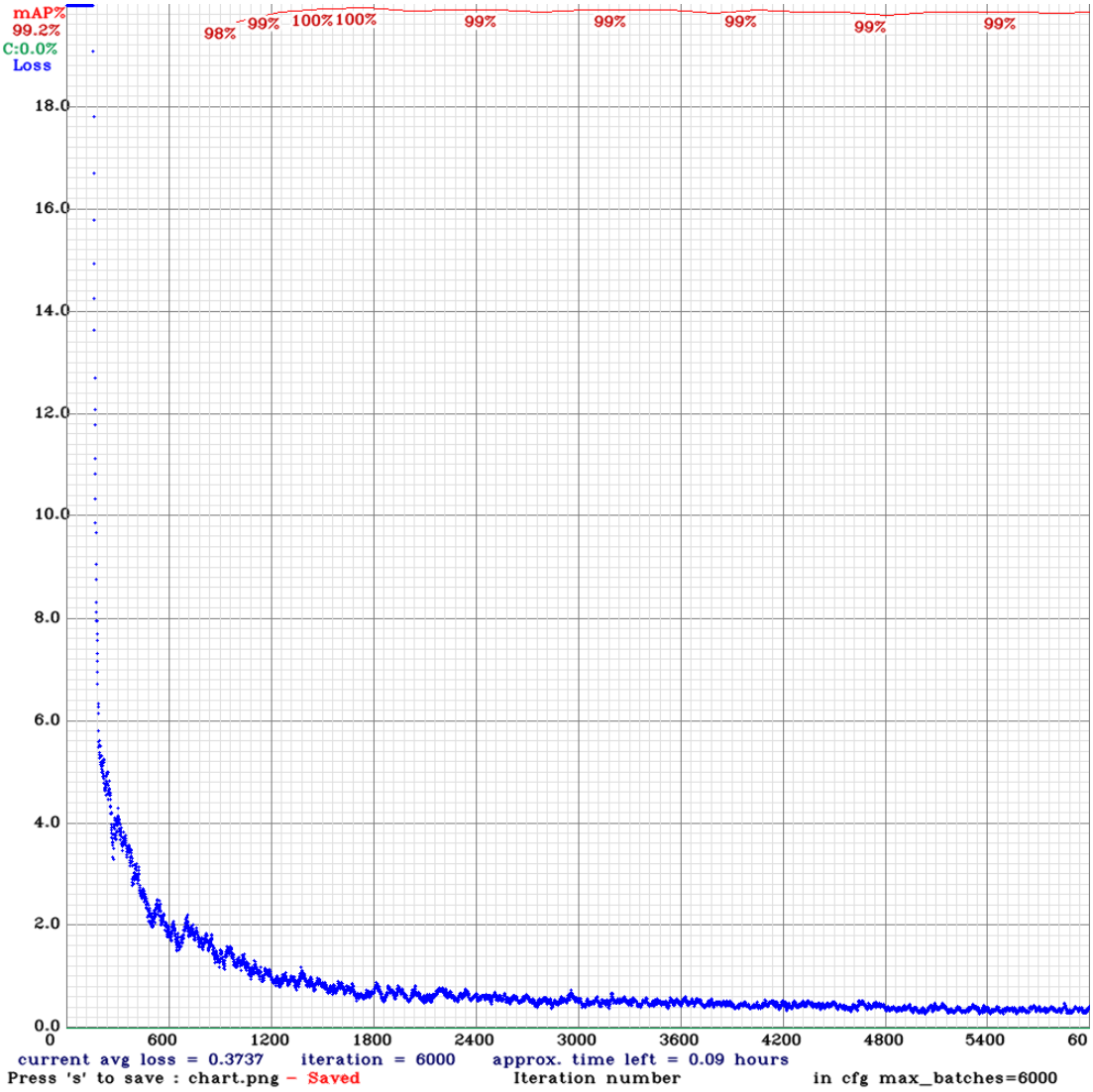



**Figure 3**. Training Chart

For all experiments, hardware configuration of Intel Core i7-7770k, 4.3 GHz CPU, Asus Prime Z270-Aand Asus ROG-SYRIX-GTX1080TI Motherboard with 16 GB RAM for both CPU and GPU was used.

## 5. Results and Comparisons

The time required to complete the training process with the above-mentioned configurations was 8 hours 37 minutes 32 seconds and consumed 10369 MB GPU memory. While testing, the system ran at 33 milliseconds for 1080p image and 47 FPS for 1080p video stream consuming 8886 MB GPU memory. Testing was carried

out for the videos at 5 different conditions- day, night, evening, fog and rain. The mAP at 50% threshold for all the conditions is shown in Figure 4 and the ground truth vs detection results on the test data are shown in Figure 5.

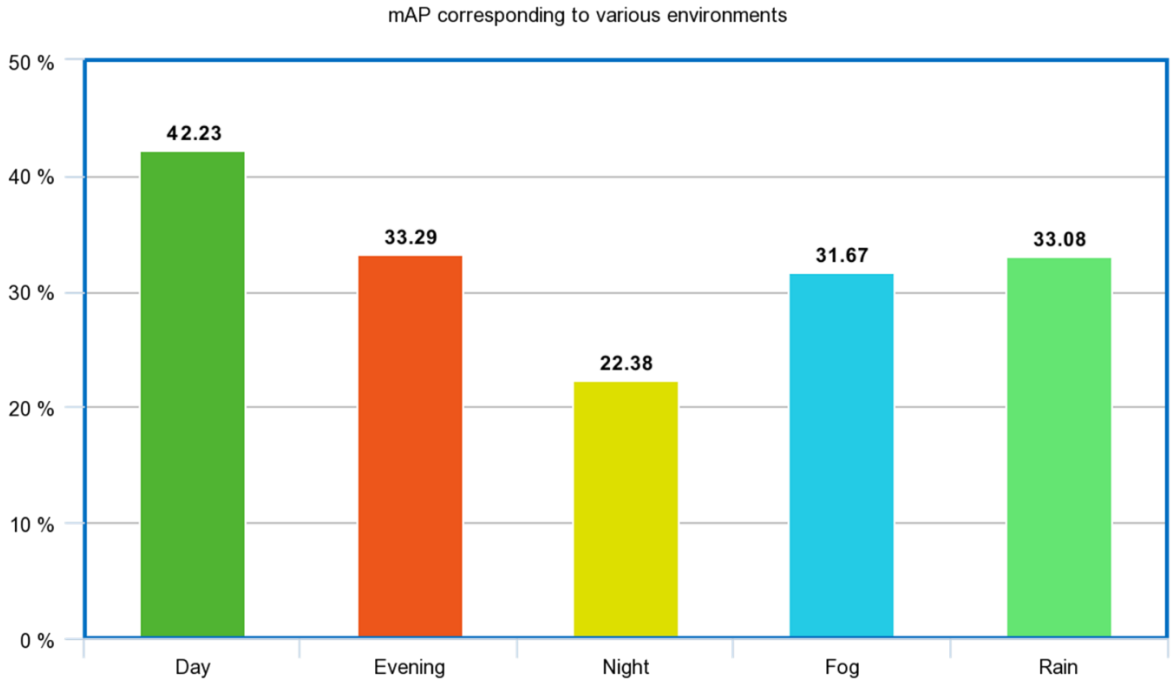



**Figure 4**. mAP[0.50] corresponding to various environments

The detection results during the day time has the highest mean average precision at 50% threshold of 42.23 and that of night has the least mAP [0.5] of 22.38. The mAP[0.50] at other 3 conditions are didn't deviate much. The mAP[0.5] for evening, fog and rain conditions are 33.29, 22.38 and 31.67 respectively. Considering the fact that the system was tested at such extreme and varying light conditions, the performance of the proposed system is quite satisfactory compared to the original YOLOv4's results on COCO dataset that is mAP [0.5] of 65.7.

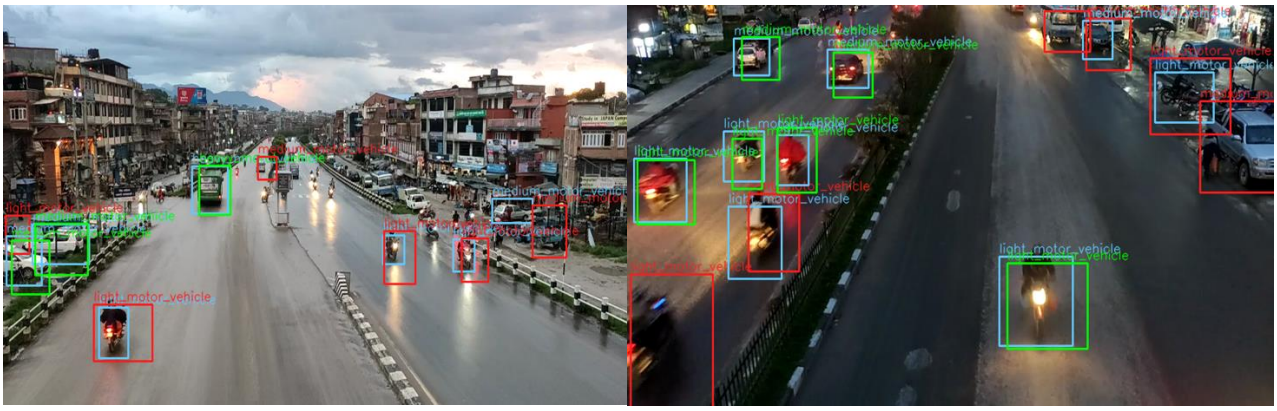


**Figure 5**. Ground truth vs detection results on the test data

## 6. Conclusion

In this paper, we modified the new state-of-the-art object detection algorithm YOLOv4, to specifically suit the needs of vehicle detection. Wherever attainable we have got through an experiment, verified the results and conferred the same within the paper. The system is able to make detections with good accuracy at different

lighting conditions and even in extreme environments. The results provide a promising scope of application of assorted techniques and tweaks to optimize YOLOv4 for alternative specific things. Further, tracking of the detected vehicles could be implemented to enhance the reliability of the system. This system can be used for automating toll-ticketing systems, parking management and also for traffic congestion management.

## References

[1] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.

[2] Lowe, David G. (1999). Object recognition from local scale-invariant features. Proc. 7th International Conference on Computer Vision (ICCV'99) (Corfu, Greece): 1150-1157. doi:10.1109/ICCV.1999.790410.

[3] Navneet Dalal, Bill Triggs. (2005), Histograms of Oriented Gradients for Human Detection. International Conference on Computer Vision & Pattern Recognition (CVPR '05), Jun, San Diego, United States. pp.886–893, ff10.1109/CVPR.2005.177ff. ffinria00548512f

[4] Zhao, Zhong-Qiu & Zheng, Peng & Xu, Shou-Tao & Wu, Xindong. (2019). Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems. PP. 1-21. 10.1109/TNNLS.2018.2876865

[5] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[6] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[7] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

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

[9] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[10] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[11] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[12] Ouyang, L., & Wang, H. (2019, July). Vehicle target detection in complex scenes based on YOLOv3 algorithm. In IOP Conference Series: Materials Science and Engineering (Vol. 569, No. 5, p. 052018). IOP Publishing.

[13] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, (2017), "Focal Loss for Dense Object Detection," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.

[14] Bochkovskiy, A., Wang C., Mark Liao, H. (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection". arXiv:2004.10934v1 [cs.CV] 23 Apr 2020.

[15] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[16] Wang, C., Liao, H.M., Yeh, I., Wu, Y., Chen, P., & Hsieh, J. (2019). CSPNet: A New Backbone that can Enhance Learning Capability of CNN. ArXiv, abs/1911.11929.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. (2015), Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37(9):1904–1916.

[18] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. (2018), Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768.