Vehicle Detector Based on YOLOv5 Architecture for Traffic Management and Control Systems

Duy-Linh Nguyen, Xuan-Thuy Vo, Adri Priadana, and Kang-Hyun Jo
Department of Electrical, Electronic and Computer Engineering,
University of Ulsan,
Ulsan, Korea
ndlinh301@mail.ulsan.ac.kr, xthuy@islab.ulsan.ac.kr, priadana@mail.ulsan.ac.kr, and acejo@ulsan.ac.kr

***Abstract*—Vehicle detection is an important module in traffic management and control systems. These systems require compactness, mobility, and high accuracy when deployed in a real-time context. Based on the YOLOv5 network architecture, this paper proposes several improvements to increase the performance and speed of the network when applied to vehicle detection. The research aims to redesign the backbone and neck modules with lightweight convolutional network architectures such as EfficientNet, PP-LCNet, and MobileNet. In addition, the Squeeze-and-Excitation attention architecture is also used inside the above-mentioned architectures to help the network focus on salient information during feature extraction. The network is trained and evaluated on a modified and normalized dataset of the UA-DETRAC dataset. As a result, the proposed network achieves 58.1% of mAP@0.5 and 40.1% of mAP@0.5:0.95 with just over ten million network parameters. This result outperforms other methods and is comparable to the lightweight architectures of the YOLOv5 family.**

***Index Terms*—Convolutional Neural Network (CNN), Lightweight architecture, Vehicle detection, YOLOv5.**

I. Introduction

The rapid development of vehicles in both quantity and type requires supporting tools for traffic management and control, especially in intelligent traffic systems. The goal of vehicle detectors is to provide assistant information for vehicle traffic counting, speed measurement, traffic accident detection, and traffic coordination [1]. For a long time, sensor-based methods have been widely applied to collect sequent information for traffic analysis and processing. These methods use specialized detectors which are high implementation and maintenance costs, such as laser detectors, radar detectors, induction loop detectors, etc. In addition, the performance is heavily influenced by environmental factors [2]. Nowadays, closed circuit television (CCTV) and surveillance camera systems have been deployed in almost all traffic systems. Its flexibility and convenience have spurred vision-based methods to be developed to meet real-world requirements and solve problems existing in traditional methods. In that trend, this paper proposes techniques to improve the YOLOv5 network architecture for vehicle detection. The techniques exploit the lightweight convolutional neural network (CNN) architectures and optimization the network parameters for low-computing devices in real-time scenarios.

The core contributions of this paper are as follows:

- Proposed a vehicle detector based on the YOLOv5 network with lightweight architectures to optimize the network parameters and computational complexity.
- Built an image dataset for vehicle detection tasks from a set of large videos of the UA-DETRAC dataset. This dataset was trained and evaluated with all variants of the YOLOv5 network family and then compare to the proposed network.

The remaining parts of the paper are organized as follows: Section II introduces several technologies relative to vehicle detection. Section III explains the detail of improvements in the proposed method. Section IV reports and analyzes the experimental results. Finally, Section V concludes the issue and presents the direction of future works.

II. Related Works

The related methods of vehicle detection will be introduced in this section. These techniques can be considered with traditional-based and CNN-based methods.

A. Traditional-based methods

These methods are mainly based on manually defined feature patterns from standard feature extractors. Several common feature extractors were the Haar-like feature [3], the histogram of oriented gradients (HOG) feature [4], and the local binary pattern (LBP) [5]. Besides, to improve detection accuracy, other studies have fused feature extraction and classification methods together. The work in [6], used a combination of HOG and LBP methods to generate feature vectors. Then it applied the support vector machine (SVM) method to learn and classify the media. Similarly, the authors in [7] exploited the Haar-like and HOG methods for feature extraction and SVM for vehicle classification. In other studies, to reduce the computational complexity of the SVM method, the AdaBoost classifier and its variants were used for classifying vehicles on the features extracted by the Haar-like method [8]. The detection accuracy of traditional methods depends largely on prior knowledge. But in real-life settings, vehicles appear in a variety of shapes, colors, and distortions. Therefore, vehicle detection is limited and difficult to implement in real-time applications.

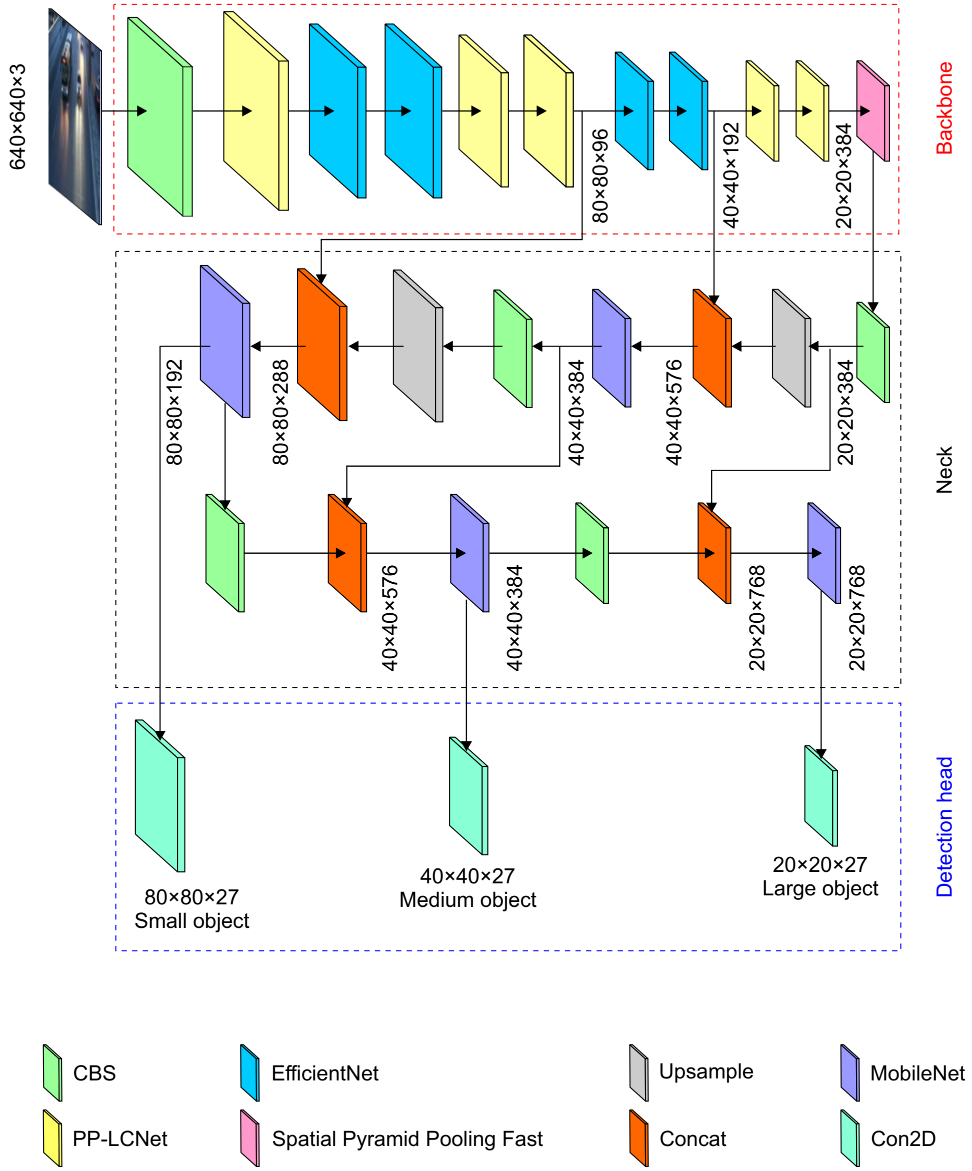


Fig. 1. The overall proposed network architecture.

B. *CNN-based methods*

Unlike traditional methods, CNN-based methods directly extract features and learn them through the training phase. Vehicle detection applications are widely deployed with both single-stage and two-stage detectors. The work in [9] used feature fusion techniques in CNN to connect high-level features and low-level features and detect different sizes of highway vehicles on multi-scales. The study in [10] applied YOLOv2 network architecture for detecting vehicles and the Kanade-Lucas-Tomasi tracking technique for counting the number of vehicles. Later, the YOLOv5 network and several dataset augmentation techniques were utilized by [11] for real-time vehicle detection. In general, CNN-based methods have solved the pre-feature extraction problem and greatly improved the ability to detect vehicles in different contexts.

III. Proposed Methodology

Fig. 1 shows the overall proposed network architecture. This network is an improvement from YOLOv5 architecture [12] comprised of three modules: backbone, neck, and detection head.

A. *Backbone*

During the study on the YOLOv5 architecture, this work found that the Focus and Cross Stage Partial Network Bottleneck (CSP) blocks present many advantages in feature extraction but cause computational complexity and a lot of network parameters. Therefore, the first step is replacing the Focus module with another more straightforward one that still ensures effective feature extraction, called CBS. This module is designed with a 1×1 convolution layer (1×1 Con2D) followed by a batch normalization (BN) and a SiLU activation function. Fig. 2 shows the structure of the CSB module.

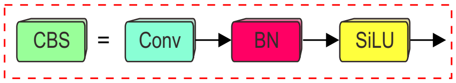


Fig. 2. The structure of the CBS block.

The next step is removing all CSP modules and choosing the combination of two lightweight modules proposed from the idea of PP-LCNet [13] and EfficientNet [14]. The structure of lightweight PP-LCNet is shown in Fig. 3. This module is organized by a 3 × 3 depthwise convolution layer (3 × 3 DWCon), an attention block (SE block), and a 1 × 1 convolution layer. Interspersed between these layers is a batch normalization (BN) and a Hardswich activation function (HS). The SE attention block is inspired by the original SE attention mechanism [15] and consists of a global average pooling layer, a fully connected (FC1) layer followed by a rectified linear unit (ReLU) activation function, and a second FC (FC2) followed by a sigmoid activation function. This design takes advantage of lightweight architectures to save a large number of network parameters in the backbone. On the other hand, the SE attention mechanism increases the network's ability to focus on outstanding features. Therefore, this design ensures the quality of feature extraction at each feature map level.

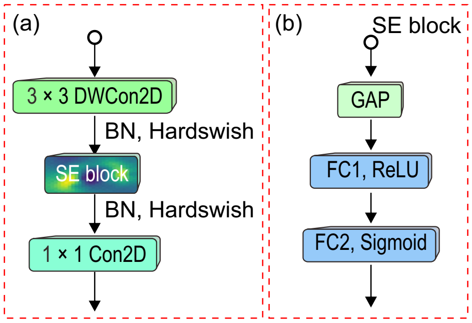


Fig. 3. The structure of PP-LC (a) module and SE (b) block.

The structure of the lightweight EfficientNet module is shown in Fig. 4. This module is designed quite simply with only convolution and depthwise layers, interspersed with a BN and a ReLU6 activation function (for stride 2 case (Fig. 4 (b)). Same design for the stride 1 case (Fig. 4 (a)) but adds a skip connection that is started from the input and aggregated with the feature map of the main branch through the addition operation. Its feature extraction process focuses on the channel dimension. The integration of lightweight PP-LCNet and EfficientNet architectures covers spatial and channel dimension feature extraction across the entire backbone.

The final block in the backbone module is the Spatial Pyramid Pooling Fast (SPPF). This block is a variant of Spatial Pyramid Pooling (SPP) widely used in early generations of the YOLOv5 architecture. Different from the original SPP block, the SPPF block is designed with three max pooling layers with the same kernel size $(k = 5)$ arranged side by side. The output of each max pooling layer is aggregated with the output of the first SBS by a concatenation operation followed by another CBS layer. This block has a role as a bridge between the backbone and neck modules. The structure of the SPPF block is shown in Fig. 5.

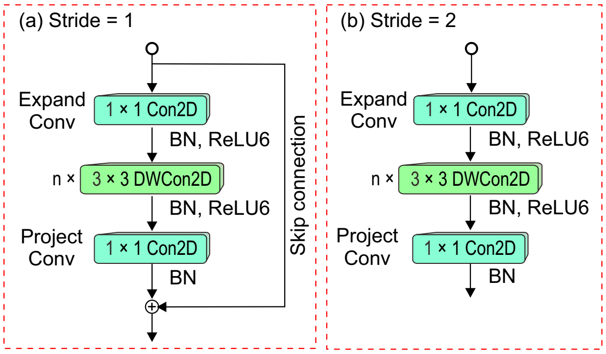


Fig. 4. The structure of lightweight EfficientNet block.

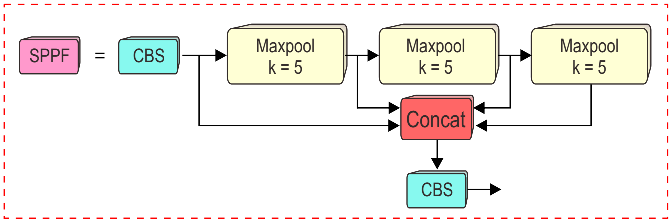


Fig. 5. The structure of SPPF block.

B. Neck

The neck module of the proposed network architecture still reuses the Path Aggregation Network (PAN) [16] mechanism. This mechanism combines the current feature maps with previous feature maps through upsampling and concatenation operations. Inside, this work replaces all CSP blocks with the lightweight MobileNet structure as shown in Fig. 6. The MobileNet block is built similarly to the structure of the mentioned lightweight EfficientNet block with two cases for stride 1 and stride 2. Differently, MobileNet blocks add an SE attention mechanism after the 3 × 3 DWCon2D and change the ReLU6 activation functions with the Hardswish activation functions. The output of the neck module is three aggregated feature maps corresponding to the three scale levels of the object to be detected: large (20 × 20 × 768), medium (40 × 40 × 384), and small (80 × 80 × 192).

C. Detection head

From the three levels of feature maps generated by the neck module, this work utilizes the structure of the detector heads in the YOLOv5 network. These three output feature maps go through three convolution operations to produce three

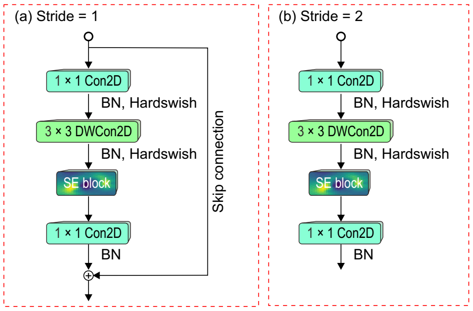


Fig. 6. The structure of lightweight MobileNet block.

detectors with dimensions $80 \times 80 \times 27$, $40 \times 40 \times 27$, and $20 \times 20 \times 27$ for small, medium, and large object sizes, respectively. Each detector head uses three anchors of different sizes. The details of the detection heads are shown in Table I. The last parameter of the feature map dimension is the prediction coefficient calculated as follows:

$$C = (5 + C_n) \times A = (5 + 4) \times 3 = 27, \tag{1}$$

where C is the detection coefficient, C_n is the number of classes, and A is the number of anchors.

TABLE I
THE DETAIL OF EACH DETECTION HEAD.

Heads	**Input**	**Anchor sizes**	**Ouput**	**Object**
1	80 × 80 × 129	(10, 13), (16, 30), (33, 23)	80 × 80 × 27	Small
2	40 × 40 × 384	(30, 61), (62, 45), (59, 119)	40 × 40 × 27	Medium
3	20 × 20 × 768	(116, 90), (156, 198), (373, 326)	20 × 20 × 27	Large

TABLE II
THE DETECTION RESULTS OF THE PROPOSED NETWORK WITH EACH CLASS ON THE UA-DETRAC VALIDATION SET.

Class	**Labels**	**Labels**	**P**	**R**	**mAP@0.5:0.95**
all	68,064	63.8	55.8	58.1	40.1
others	1,691	32.6	60.2	26.1	15.9
car	55,462	74.1	66.9	71.0	53.4
van	3,795	52.1	49.6	54.8	39.8
bus	7,116	73.5	75.6	71.4	51.3

D. Loss function

The loss function in this paper is defined as follows:

$$L = \lambda_{box} L_{box} + \lambda_{obj} L_{obj} + \lambda_{cls} L_{cls}, \tag{2}$$

where L_{box} is the bounding box regression loss which uses CIoU loss [17] to compute. L_{obj} and L_{cls} are the object confidence score loss and the classification loss, respectively. They use Binary Cross Entropy loss [18] to calculate. λ_{box}, λ_{obj}, and λ_{cls} are applied to control the balancing of overall loss.

IV. EXPERIMENTS

A. Dataset

UA-DETRACT [19] is a large dataset for multi-object detection and multi-object tracking. The dataset consists of 10 hours of video obtained from Cannon EOS 550D cameras in 24 locations in Beijing and Tianjin cities (China) under different weather conditions. Videos are recorded with a frame rate of 25 frames per second and a resolution of 960×540 pixels. In total, more than 140,000 frames with 8,250 vehicles and 1.21 million bounding boxes were annotated by hand. This dataset is divided into four classes including car, bus, van, and others. Recognizing the contextual duplication in sequential video frames, this experiment extracted and used only 8,222 images for training and 5,621 images for evaluation. This is to reduce the burden and save time for model training and evaluation while still ensuring vehicle detection accuracy.

B. Experimental setup

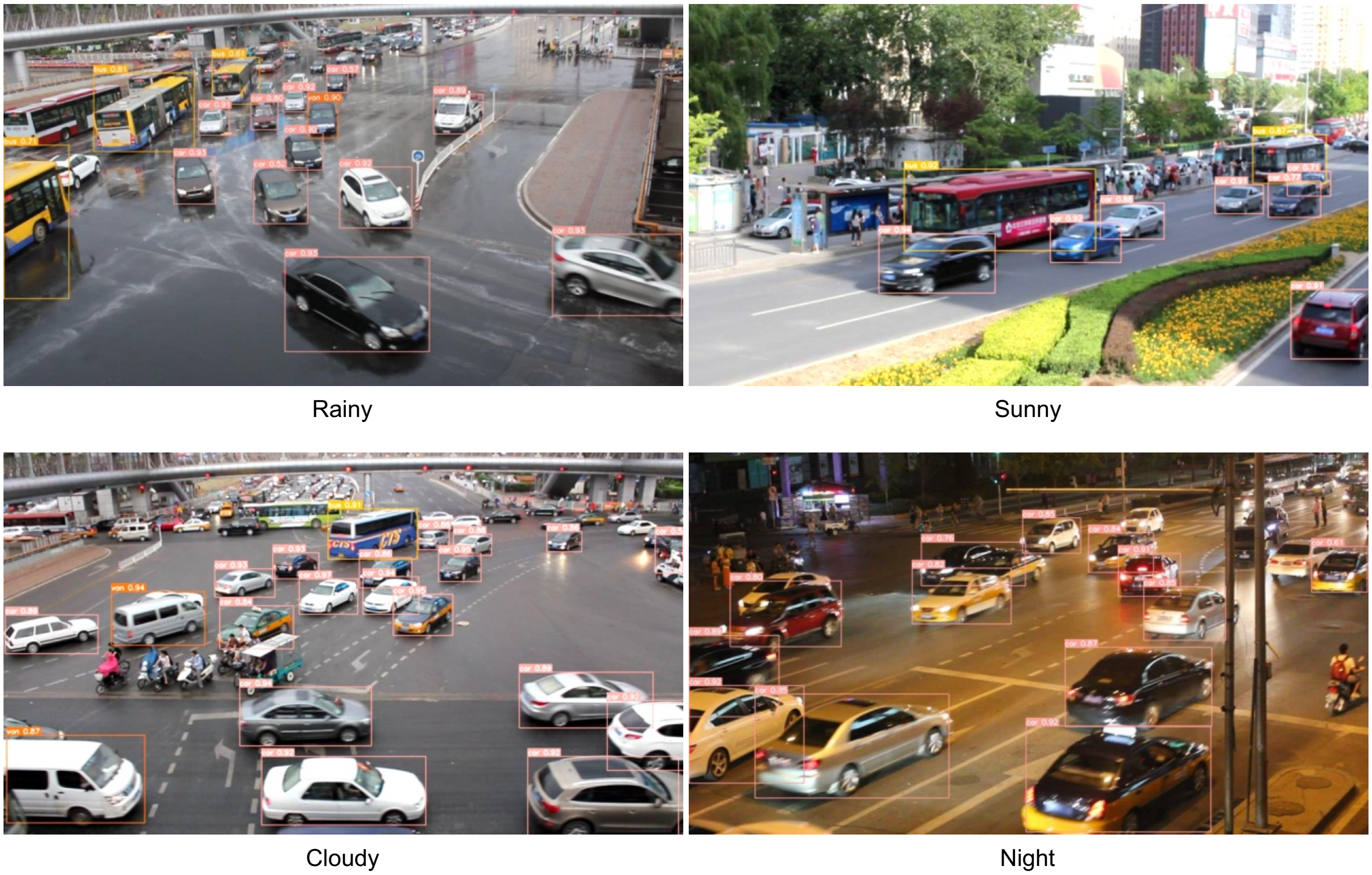
The proposed network architecture is implemented using the Python programming language on top of the Pytorch framework. This model was trained on a Testla V100 32GB GPU and evaluated on a GeForce GTX 1080Ti 11GB GPU. The input image size is 640×640 pixels. The learning rate is set from 10^3 and increases to 10^5. Similarly, momentum is also assigned from 0.8 then gradually increases to 0.937. The optimization method is Adam optimization. The training process takes 300 epochs with a batch size of 64. The balancing parameters are set as $\lambda_{box} = 0.05$, λ_{obj}=1, and $\lambda_{cls} = 0.5$. Several data augmentation methods are applied such as flip, translate, mosaic, and scale. The inference time (ms) is performed and reported with the same training input image size, batch size of 32, confidence threshold and IoU threshold are set to 0.5.

C. Experimental results

To evaluate the performance of the proposed network, this experiment compares 5 variants of retrained YOLOv5 network architecture (n, s, m, l, x) from scratch and the other networks in [2] on the UA-DETRAC dataset. The detailed result of the proposed network on each class and the comparison results are shown in Table II and Table III, respectively. As a result, the proposed network achieves 58.1% of mAP@0.5 (mean average precision with an IoU threshold of 0.5) and 40.1% of mAP0.5:0.95 (mean Average Precision with an IoU threshold of 0.5 to 0.95). For mAP@0.5, the performance of the proposed network is superior to the networks in [2] (6.6%↑ to 7.8% ↑) and two retrained YOLOv5 networks (YOLOv5n (6.0% ↑), YOLOv5s (3.6% ↑), YOLOv5l (0.8% ↑)) and is comparable to the other retrained YOLOv5 networks (YOLOv5m (2.5% ↓), YOLOv5x (1.7% ↓)). With mAP@0.5:0.95, the performance of the proposed network outperforms the lightweight YOLOv5 networks (n, s) and is lower than the large-scale YOLOv5 networks from 3.0% (YOLOv5l) to 4.8% (YOLOv5m, YOLOv5x). In terms of speed (inference time), the proposed network is also faster than large-scale YOLOv5 networks (YOLOv5l (3.9 ms ↑)),

TABLE III
COMPARISON RESULT OF PROPOSED NETWORK WITH OTHER NETWORKS ON THE UA-DETRAC VALIDATION SET.

Models	Parameter	Weight (MB)	GFLOPs	mAP@0.5	mAP@0.5:0.95	Inf. time (ms)
YOLOv5x	86,193,601	173.1	204.0	59.8	44.9	17.7
YOLOv5l	46,124,433	92.8	107.8	57.3	43.1	10.0
YOLOv5m	20,865,057	42.2	48.0	60.6	44.9	5.7
YOLOv5s	7,020,913	14.4	15.8	54.5	39.0	2.5
YOLOv5n	1,764,577	3.8	4.2	52.1	37.1	1.3
YOLOv5s-GIoU [2]	N/A	13.7	N/A	50.3	N/A	N/A
YOLOv5s-CIoU [2]	N/A	13.7	N/A	50.5	N/A	N/A
YOLOv5-NAM-GIoU [2]	N/A	13.9	N/A	51.2	N/A	N/A
YOLOv5-NAM-CIoU [2]	N/A	13.9	N/A	51.5	N/A	N/A
Our	**10,215,169**	**20.8**	**18.4**	**58.1**	**40.1**	**6.1**



Rainy

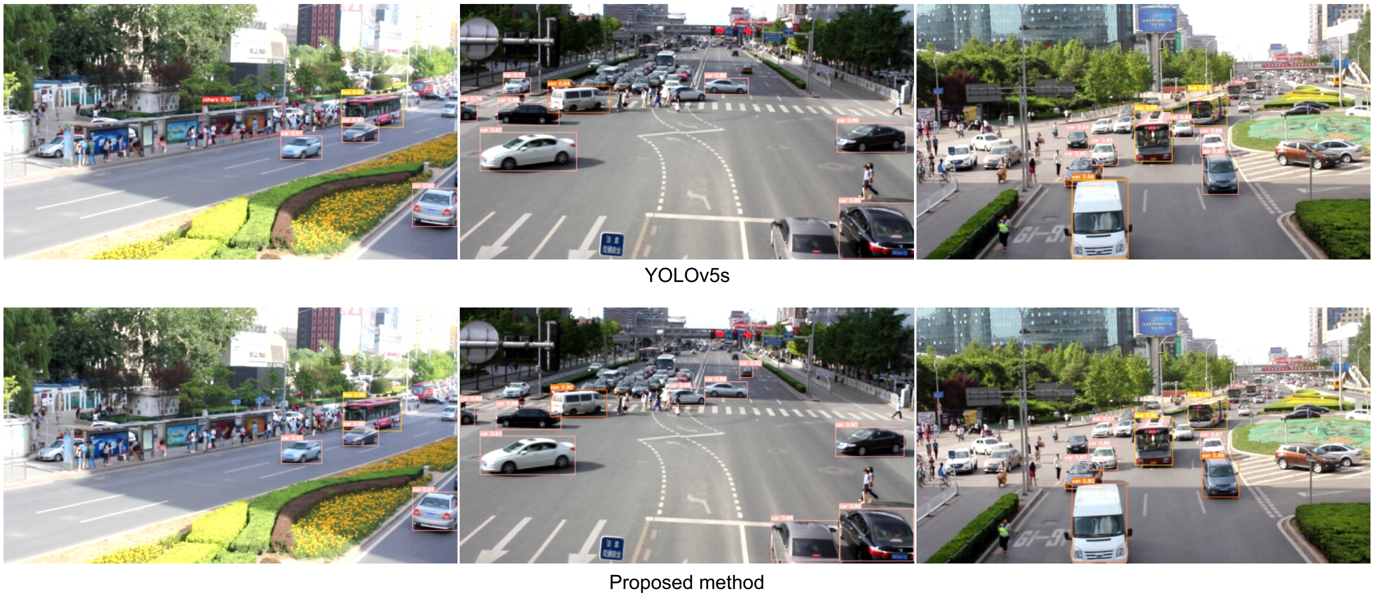
Sunny

Cloudy

Night

Fig. 7. The qualitative result on UA-DETRAC validation set.

(YOLOv5x (11.6 ms ↑)) and approximates YOLOv5m (0.4 ms ↓) but the network parameter and computational complexity (GFLOPs) are less than half. Fig. 7 presents several qualitative results on the UA-DETRAC dataset with different scenes (cloudy, night, sunny, and rainy). With just over 10 million network parameters, the proposed network can be considered for comparison with the YOLOv5s architecture which was widely used in mobile and embedded devices. The visualization results in Fig. 8 show that the proposed vehicle detection network is better than the YOLOv5s network architecture in many different contexts. From the above results, it is easy to see the balance in detection speed, computational complexity, and network parameters of the proposed network. This allows this model can be deployed in real-time applications for vehicle detection. Through the experimental process, the proposed network has revealed several weaknesses due to the impact of environmental factors and actual context. The factors that reduce the network's detectability include weather conditions, vehicle density, vehicle overlap, vehicle moving speed, vehicle direction, and distance from the vehicle to the camera. In addition, the quality and camera angle are also crucial factors affecting the quality of the proposed method.



YOLOv5s

Proposed method

Fig. 8. The comparison result between YOLOv5s and proposed method on UA-DETRAC validation set.

D. Ablation studies

This work conducts several ablation studies to evaluate the effect of each block in the proposed network. The blocks are replaced one by one and then trained and evaluated on the UA-DETRAC dataset. The results obtained are shown in Table IV. This result demonstrates that the combination of the lightweight PP-LCNet and EfficientNet networks in the backbone module increases mAP by more than 2% while slightly increasing the network parameters and computational complexity. Similarly, the integration of lightweight PP-LCNet and EfficientNet in the backbone module, MobileNet in the neck module, and SPP block increases mAP by nearly 2%, and other factors are almost unchanged. Finally, replacing the SPP block with the SPPF block achieves the best results. Therefore, this work selects the last architecture to train, evaluate, and report on vehicle detection capabilities.

TABLE IV
ABLATION STUDIES WITH DIFFERENT TYPES OF PROPOSED NETWORK ON THE UA-DETRAC VALIDATION SET.

Blocks	**Proposed networks**			
Conv	✓	✓	✓	✓
PP-LCNet		✓	✓	✓
EfficientNet	✓	✓	✓	✓
MobileNet	✓		✓	✓
SPPF				✓
SPP	✓	✓	✓	
Parameter	9,736,527	10,191,617	10,215,169	10,215,169
Weight (MB)	19.9	22.8	20.8	20.8
GFLOPs	19.1	23.9	18.4	18.4
mAP@0.5	52.8	54.7	56.2	**58.1**
mAP@0.5:0.95	38.0	40.3	39.3	**40.1**

V. CONCLUSION AND FUTURE WORK

This paper proposes a method to improve the YOLOv5 object detection network for vehicle detection. Research focuses mainly on redesigning backbone and neck modules using a combination of lightweight architectures such as PP-LCNet, EfficientNet, and MonileNet. In addition, this work also provides an image dataset for vehicle detection extracted from the large UA-DETRAC dataset. With the optimization of network parameters, computational complexity, and inference speed, the proposed network has the potential to be applied to mobile and embedded devices. In the future, the vehicle detection network will be further developed with attention and context feature enhancement mechanisms for small-sized vehicles far from the camera.

ACKNOWLEDGEMENT

This result was supported by the ”Regional Innovation Strategy (RIS)” through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003).

REFERENCES

[1] C. Meng, H. Bao, and M. Yan, “Vehicle detection: A review,” *Journal of Physics: Conference Series*, vol. 1634, p. 012107, 09 2020.

[2] J. Wang, Y. Dong, S. Zhao, and Z. Zhang, “A high-precision vehicle detection and tracking method based on the attention mechanism,” *Sensors*, vol. 23, no. 2, 2023.

[3] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–I, 2001.

[4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, pp. 886–893 vol. 1, 2005.

[5] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[6] M.-T. Pei, J.-J. Shen, M. Yang, and Y. Jia, "Vehicle detection method in complex illumination environment," vol. 36, pp. 393–398, 04 2016.

[7] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "A hybrid vehicle detection method based on viola-jones and hog + svm from uav images," *Sensors*, vol. 16, no. 8, 2016.

[8] D.-Y. Huang, C.-H. Chen, T.-Y. Chen, W.-C. Hu, and K.-W. Feng, "Vehicle detection and inter-vehicle distance estimation using single-lens video camera on urban/suburb roads," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 250–259, 2017.

[9] L. Chen, F. Ye, Y. Ruan, H. Fan, and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network," *EURASIP Journal on Image and Video Processing*, vol. 2018, 10 2018.

[10] A. Gomaa, T. Minematsu, M. Abdelwahab, M. Abo-Zahhad, and R. Taniguchi, "Faster cnn-based vehicle detection and counting strategy for fixed camera scenes," *Multimedia Tools and Applications*, vol. 81, pp. 25443–25471, July 2022. Publisher Copyright: © 2022, The Author(s).

[11] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-time vehicle detection based on improved yolo v5," *Sustainability*, vol. 14, no. 19, 2022.

[12] G. Jocher and et al., "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020.

[13] C. Cui, T. Gao, S. Wei, Y. Du, R. Guo, S. Dong, B. Lu, Y. Zhou, X. Lv, Q. Liu, X. Hu, D. Yu, and Y. Ma, "Pp-lcnet: A lightweight CPU convolutional neural network," *CoRR*, vol. abs/2109.15099, 2021.

[14] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019.

[15] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *CoRR*, vol. abs/1709.01507, 2017.

[16] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *CoRR*, vol. abs/1803.01534, 2018.

[17] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *CoRR*, vol. abs/2005.03572, 2020.

[18] R. Rubinstein and D. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Information Science and Statistics, Springer New York, 2011.

[19] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, 2020.