Accepted Manuscript

Vietnam Journal of Computer Science

Article Title:	Car Detection for Smart Parking Systems Based on Improved YOLOv5
Author(s):	Duy-Linh Nguyen, Xuan-Thuy Vo, Adri Priadana, Kang-Hyun Jo
DOI:	10.1142/S2196888823500185
Received:	04 September 2023
Accepted:	16 November 2023
To be cited as:	Duy-Linh Nguyen *et al.*, Car Detection for Smart Parking Systems Based on Improved YOLOv5, *Vietnam Journal of Computer Science*, doi: 10.1142/S2196888823500185
Link to final version:	https://doi.org/10.1142/S2196888823500185

World Scientific
www.worldscientific.com

Vietnam Journal of Computer Science

Car Detection for Smart Parking Systems Based on Improved YOLOv5

Duy-Linh Nguyen, Xuan-Thuy Vo, Adri Priadana, and Kang-Hyun Jo*
Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea
ndlinh301@mail.ulsan.ac.kr, xthuy@islab.ulsan.ac.kr, priadana@mail.ulsan.ac.kr, and acejo@ulsan.ac.kr

Received (Day Month Year)
Revised (Day Month Year)

Nowadays, YOLOv5 is one of the most popular object detection network architectures used in real-time and industrial systems. Traffic management and regulation are typical applications. To take advantage of the YOLOv5 network and develop a parking management tool, this paper proposes a car detection network based on redesigning the YOLOv5 network architecture. This research focuses on network parameter optimization using lightweight modules from EfficientNet and PP-LCNet architectures. On the other hand, this work also presents an aerial view dataset for car detection tasks in the parking, named the Aerial View Parking Lot (AVPL). The proposed network is trained and evaluated on two benchmark datasets which are the Car Parking Lot Dataset and the Pontifical Catholic University of Parana+ Dataset and one proposed dataset. The experiments are reported on mAP@0.5 and mAP@0.5:0.95 measurement units. As a result, this network achieves the best performances at 95.8 %, 97.4 %, and 97.0% of mAP@0.5 on the Car Parking Lot Dataset, the Pontifical Catholic University of Parana+ Dataset, and the proposed AVPL dataset, respectively. A set of demonstration videos and the proposed dataset are available here: https://bit.ly/3YUoSwi.

Keywords: Convolutional neural network (CNN); EfficientNet; PP-LCNet; Parking management; YOLOv5.

1. Introduction

Along with the rapid development of modern and smart cities, the number of vehicles in general and cars in particular has also increased in both quantity and type. According to a report by the Statista website [1], there are currently about one and a half million cars in the world and it is predicted that in 2023, the number of cars sold will reach nearly 69.9 million. This number will increase further in the coming years. Therefore, developing tools to support parking lot management is essential. To construct smart parking lots, researchers propose many methods based on geomagnetic [2], ultrasonic [3], infrared [4], and wireless techniques [5]. In the geomagnetic method, geomagnetic sensors installed around the parking lot are used to collect

*corresponding author.

and transmit information about cars and the environment to the processing center. The ultrasonic method utilizes ultrasonic wave signals to predict the outline and boundaries of the cars through a grid map. Similarly, the infrared method applies infrared waves to estimate car distance and communicate between devices through electronic circuits. The wireless technique designs the wireless network nodes and develops them based on the electronic chips to detect the vehicle and environmental information in the parking. Generally, these approaches mainly rely on the operation of sensors designed and installed in the parking lot. Although these designs achieve high accuracy, they require large investment, labor, and maintenance costs, especially when deployed in large-scale parking lots. Exploiting the benefits of convolutional neural networks (CNNs) in the field of computer vision, several researchers have designed networks to detect empty or occupied parking spaces using conventional cameras with quite good accuracy [6,7,8]. Following that trend, this paper proposes a car detector to support smart parking systems. This work explores lightweight network architectures and redesigned modules inside of the YOLOv5 network [9] to balance network parameters, detection accuracy, and computational complexity. It ensures deployment in real-time systems with the lowest deployment cost. The main contributions of this paper are shown below:

1 - Proposes an improved YOLOv5 architecture for car detection that can be applied to smart parking systems and other related fields of computer vision.

2 - Provides an Aerial View Parking Lot (AVPL) dataset for car detection tasks in the parking.

3 - The proposed detector performs better than other detectors on the Car Parking Lot dataset, the Pontifical Catholic University of Parana+ dataset, and the proposed AVPL dataset.

The distribution of the remaining parts in the paper is as follows: Section 2 presents the car detection-based methods. Section 3 explains the proposed architecture in detail. Section 4 introduces the experimental setup and analyzes the experimental results. Section 5 summarizes the issue and future work orientation.

2. Related works

2.1. *Traditional machine learning-based methods*

The car detection process of traditional machine learning-based techniques is divided into two stages, manual feature extraction and classification. First, feature extractors generate feature vectors using classical methods such as Scale-invariant Feature Transform (SIFT) [10], Histograms of Oriented Gradients (HOG) [11], and Haar-like features [12]. Then, the feature vectors go through classifiers like the Support Vector Machine (SVM) and Adaboost [13,14] to obtain the target classification result. The traditional feature extraction methods rely heavily on prior knowledge. However, in the practical application, there are many objective confounding factors including weather, exposure, distortion, etc. Therefore, the applicability of these techniques on real-time systems is limited due to low accuracy.

2.2. *CNN-based methods*

Parking lot images obtained from drones or overhead cameras contain many small-sized cars. In order to detect these objects well, many studies have focused on the small object detection topic using a combination of CNN and traditional methods or one-stage detectors. The authors in [15,16,17] fuse the modern CNNs and SVM networks to achieve high spatial resolution in vehicle count detection and counting. Research in [18] develops a network based on the YOLOv3 network architecture in which the backbone network is combined between ResNet and DarkNet to solve object vision in drone images. The work in [19] proposes a new feature-matching method and a spatial context analysis for pedestrian-vehicle discrimination. An improved YOLOv5 network architecture is designed by [20] for vehicle detection and classification in Unmanned Aerial Vehicle (UAV) imagery and [21] for real-world imagery. Another study in [22] provides a one-stage detector (SF-SSD) with a new spatial cognition algorithm for car detection in UAV imagery. The advantage of modern machine learning methods is high detection and classification accuracy, especially for small-sized objects. However, they require the network to have a high-level feature extraction and fusion, and a certain complexity to ensure operation in real-world conditions.

3. Methodology

The proposed car detection network is shown in Fig. 1. This network is an improved YOLOv5 architecture including three main parts: backbone, neck, and detection head.

3.1. *Proposed network architecture*

Basically, the structure of the proposed network follows the design of the YOLOv5 network architecture with many changes inside the backbone and neck modules. Specifically, the Focus module is replaced by a simple block called Conv. This block is constructed with a standard convolution layer (Conv2D) with kernel size of 1×1 followed by a batch normalization (BN) and a ReLU activation function as shown in Fig. 2 (a). This replacement greatly reduces computational complexity but still ensures feature extraction at the initial stage. Subsequent blocks in the backbone module are also redesigned based on inspiration from lightweight network architectures such as PP-LCNet [23] and EfficientNet [24]. The design of the PP-LCNet layer is described in detail in Fig. 3 (a). It consists of a depthwise convolution layer (3×3 DWConv), an attention block (SE block), and ends with a standard convolution layer (1×1 Conv2D). In between these layers, the BN and the hardswish activation function [25] are used. The SE block is an attention mechanism based on a global average pooling (GAP) layer, a fully connected layer (FC1) followed by a rectified linear unit activation function (ReLU), and a second fully connected layer (FC2) followed by a sigmoid activation function as Fig. 3 (b). This method uses lightweight

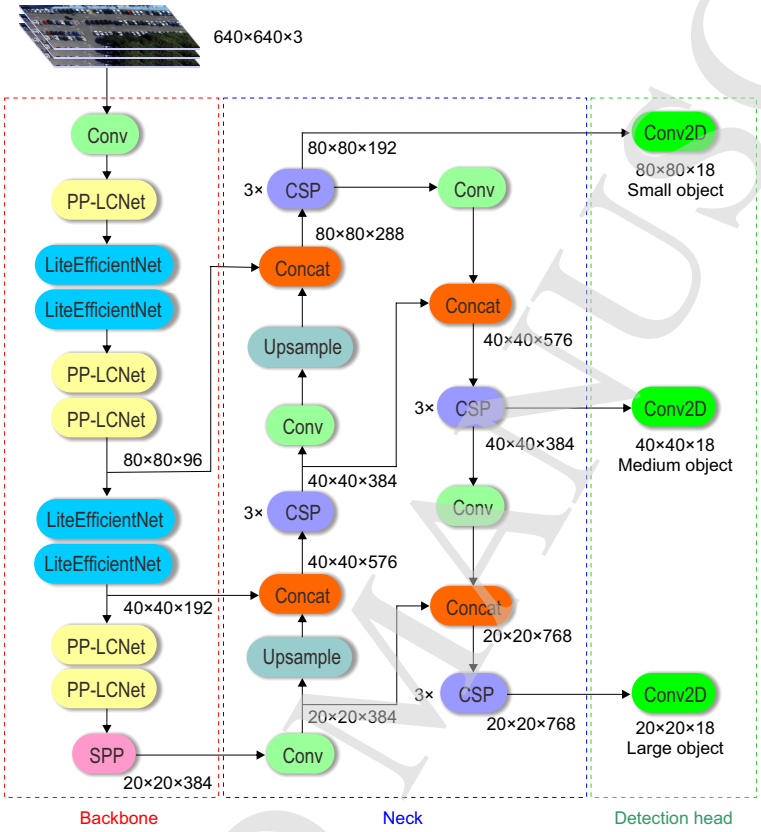

Fig. 1. The architecture of proposed car detector.

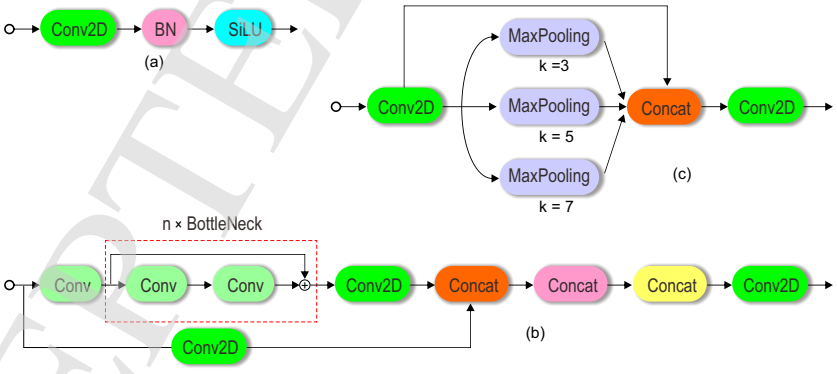

Fig. 2. The architecture of Conv (a), BottleNeck Cross Stage Partial (CSP) (b), and Spatial Pyramid Pooling (SPP) (c) blocks.

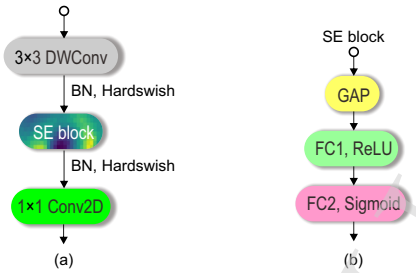

Fig. 3. The architecture of PP-LCNet (a) and SE (b) blocks.

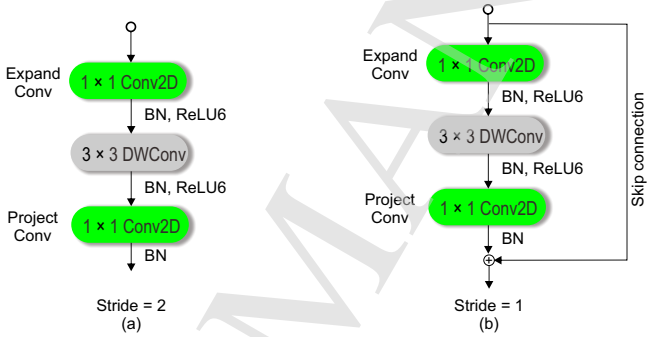

Fig. 4. The two types of LiteEfficientNet architecture, stride = 2 (a) and stride = 1 (b)

convolution layers that save a lot of network parameters. In addition, the attention mechanism helps the network focus on learning important information about the object on each feature map level. The next block is LiteEfficientNet. This block is very simple and is divided into two types corresponding to two stride levels (stride = 1 or stride = 2). In the first type with stride = 2, the LiteEfficientNet block uses an extended convolution layer (1×1 Conv2D), a depth-wise convolution layer (3×3 DWConv), and ends with a project convolution layer (1×1 Conv2D). For the second type with stride = 1, the LiteEfficientNet block is exactly designed the same as the first type and added a skip connection to merge the current and original feature maps with the addition operation. These blocks still apply the lightweight architectures and add one more skip connection to extract the feature maps following the channel dimension. The combined use of PP-LCNet and LiteEfficientNet blocks ensures that feature extraction is both spatial and channel dimensions of each feature map level. The detail of the LiteEfficientNet block is shown in Fig. 4. The last block in the backbone module is the Spatial Pyramid Pooling (SPP) block. This work re-applies the architecture of SPP in the YOLOv5 as Fig. 2 (c). However, to minimize the network parameters, the max pooling kernel sizes are reduced from 5×5, 9×9, and 13×13 to 3×3, 5×5, and 7×7, respectively.

The neck module in the proposed network utilizes the Path Aggregation Network (PAN) architecture with Conv and BottleNeck Cross Stage Partial (CSP) blocks following the original YOLOv5. Fig. 2 (b) shows details of the CSP block. The PAN module combines the current feature maps with previous feature maps by concatenation operations. It generates the output with three multi-scale feature maps that are enriched information. These serve as three inputs for the detection heads.
The detection head module also leverages the construction of three detection heads from the YOLOv5. Three feature map scales of the PAN neck go through three convolution operations to conduct prediction on three object scales: small, medium, and large. Each detection head uses three anchor sizes that describe in Table 1.

Table 1. Detection heads and anchors sizes.

Heads	Input	Anchor sizes	Ouput	Object
1	$80 \times 80 \times 129$	(10, 13), (16, 30), (33, 23)	$80 \times 80 \times 18$	Small
2	$40 \times 40 \times 384$	(30, 61), (62, 45), (59, 119)	$40 \times 40 \times 18$	Medium
3	$20 \times 20 \times 768$	(116, 90), (156, 198), (373, 326)	$20 \times 20 \times 18$	Large

3.2. *Loss function*

The definition of the loss function is shown as follows:

$$\mathcal{L} = \lambda_{box}\mathcal{L}_{box} + \lambda_{obj}\mathcal{L}_{obj} + \lambda_{cls}\mathcal{L}_{cls}, \tag{1}$$

where $\mathcal{L}_{box}$ uses CIoU loss [26] to compute the bounding box regression as shown in Eq. 2. The object confidence score loss $\mathcal{L}_{obj}$ and the classification loss $\mathcal{L}_{cls}$ using Binary Cross Entropy loss [27] to calculate as presented in Eq. 5 and Eq. 6, respectively. λ_{box}, λ_{obj}, and λ_{cls} are balancing parameters.
The bounding box regression loss:

$$\mathcal{L}_{box} = \frac{1}{N_{pos}} \sum_{(x,y)} 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha\upsilon, \tag{2}$$

in which:

- N_{pos} is the total number of cells containing an object,
- (x, y) is coordinate of each cell,
- IoU is the Intersection Over Union of the predicted and ground-truth boxes,
- b and b^{gt} are the central points of the predicted bounding box and ground-truth bounding box, respectively,
- ρ is the Euclidean distance, and c is the diagonal length of the smallest enclosing bounding box covering the two boxes,
- α is a positive trade-off parameter:

$$\alpha = \frac{\upsilon}{(1 - IoU) + \upsilon}, \tag{3}$$

- υ measures the consistency of the aspect ratio:

$$\upsilon = \frac{4}{\pi}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2, \tag{4}$$

- where w^{gt}, h^{gt} and w, h are the dimensions (w: width, h: height) of the ground-truth box and predicted box, respectively.

The object confidence score loss:

$$\mathcal{L}_{obj} = \frac{1}{N_{pos}} \sum_{(x,y)} \sum_{c=1}^{class=1} C_c log(p(C_c)) + (1 - C_c) log(1 - p(C_c)), \tag{5}$$

in which:

- C_c is the confident score of an object belonging to class c,
- $p(C_c)$ is the predicted probability of an object belonging to class c.

The classification loss:

$$\mathcal{L}_{cls} = \frac{1}{N_{pos}} \sum_{(x,y)} \sum_{c=1}^{class=1} y_c log(p(y_c)) + (1 - y_c) log(1 - p(y_c)), \tag{6}$$

in which:

- y_c is the ground-truth label of class c,
- $p(y_c)$ is the predicted probability of class c.

4. Experiments

4.1. *Datasets*

The proposed network is trained and evaluated on two benchmark datasets, the Car Parking Lot Dataset (CarPK) and the Pontifical Catholic University of Parana+ Dataset (PUCPR+) [28] and one proposed dataset. The CarPK dataset contains 89,777 cars collected from the Phantom 3 Professional drone. The images were taken from four parking lots with an approximate height of 40 meters. The CarPK dataset is divided into 988 images for training and 459 images for validation phases. The PUCPR+ dataset is selected from a part of the PUCPR dataset consisting of 16,456 cars. The PUCPR+ dataset provides 100 images for training and 25 images for validation. These are image datasets for car counting in different parking lots. The cars in the image are annotated by bounding boxes with top-left and bottom-right angles and stored as text files (*.txt files). To accommodate the training and evaluation processes, this experiment converts the entire format of the annotation files to the YOLOv5 format. The proposed dataset is a combination of the Drone Car Counting Dataset YOLO [29] and the Aerial View Car Detection for Yolov5 [30] from the Kaggle website, named the Aerial View Parking Lot (AVPL). This dataset contains 339 drone-view images including 314 images for training and 25 images for evaluation. The annotation files follow the YOLO format.

4.2. *Experimental setup*

The proposed network is conducted on the Pytorch framework (version 1.11.0) and the Python programming language (version 3.7.1). This network is trained on a Testla V100 32GB GPU and evaluated on a GeForce GTX 1080Ti 11GB GPU. The optimizer is Adam optimization. The learning rate is initialized at 10^{-5} and ends at 10^{-3}. The momentum sets at 0.8 and then increases to 0.937. The training process goes through 300 epochs with a batch size of 64. The balance parameters are set as follows: λ_{box}=0.05, λ_{obj}=1, and λ_{cls}=0.5. To increase training scenarios and avoid the over-fitting issue, this experiment applies data augmentation methods such as mosaic, translate, scale, and flip. For the inference process, other arguments are set like an image size of 1024×1024, a batch size of 32, a confidence threshold = 0.5, and an IoU threshold = 0.5. The speed testing results are reported in milliseconds (ms).

4.3. *Experimental results*

The performance of the proposed network is evaluated lying on the comparison results with the retrained YOLOv5 networks from scratch and the recent research on the two above benchmark datasets. Specifically, this work conducts the training and evaluation of the proposed network with the four versions of YOLOv5 architectures (l, m, s, n). Then, it compares the results obtained with the results in [20,22] on the CarPK dataset and the results in [20] on the PUCPR+ dataset. As a result, the proposed network achieves 95.8% of mean Average Precision with an IoU threshold of 0.5 (mAP@0.5) and 63.1% of mAP with ten IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95) on the CarPK dataset. This result shows the superior ability of the proposed network compared to other networks. While the speed (inferent time) is only 1.7 ms higher than the retrained YOLOv5m network, nearly 1.5 times lower than the retrained YOLOv5l network, and quite lower than other experiments in [20] from 2.3 (YOLOv5m) to 7.9 (YOLOv5m) times. Besides, the weight of the network (22.7 MB) and the computational complexity (23.9 GFLOPs) are only half of the retrained YOLOv5m architecture. The comparison results on the CarPK validation set are presented in Table 2. For the PUCPR+ dataset, the proposed network achieves 97.4% of mAP@0.5 and 58.0% of mAP@0.5:0.95. This result is outstanding compared to other competitors and is only 0.3% of mAP@0.5 and 2.5% of mAP@0.5:095 lower than retrained YOLOv5m, respectively. However, the proposed network has a speed of 17.9 ms, which is only slightly higher than the retrained YOLOv5m network (2.3 ms ↑) and lower than the retrained YOLOv5l network (4.5 ms ↓). The comparison results on the PUCPR+ dataset are shown in Table 3. For the AVPL dataset case, this work only compares the performance of the proposed network and four versions of the YOLOv5 architecture. The results in Table 4 show that the proposed network reaches 97.0% of mAP@0.5 and 76.2% of mAP@0.5:0.95. This result is leading compared to the YOLOv5 series except to YOLOv5m (0.2% ↓) at mAP@0.5:0.95, while the computational complexity and

Table 2. Comparison result of proposed car detection network with other networks and retrained YOLOv5 on CarPK validation set. The symbol " † " denotes the retrained networks. N/A means not-available values.

Models	Parameter	Weight (MB)	GFLOPs	mAP@0.5	mAP@0.5:0.95	Inf. time (ms)
YOLOv5l†	46,631,350	93.7	114.2	95.3	62.3	26.4
YOLOv5m†	21,056,406	42.4	50.4	94.4	61.5	15.9
YOLOv5s†	7,022,326	14.3	15.8	95.6	62.7	8.7
YOLOv5n†	1,765,270	3.7	4.2	93.9	57.8	6.3
YOLOv5x [20]	N/A	167.0	205.0	94.5	57.9	138.2
YOLOv5l [20]	N/A	90.6	108.0	95.0	59.2	72.1
YOLOv5m [20]	N/A	41.1	48.0	94.6	57.8	40.4
Modified YOLOv5 [20]	N/A	44.0	57.7	94.9	61.1	50.5
SSD [22]	N/A	N/A	N/A	68.7	N/A	N/A
YOLO9000 [22]	N/A	N/A	N/A	20.9	N/A	N/A
YOLOv3 [22]	N/A	N/A	N/A	85.3	N/A	N/A
YOLOv4 [22]	N/A	N/A	N/A	87.81	N/A	N/A
SA+CF+CRT [22]	N/A	N/A	N/A	89.8	N/A	N/A
SF-SSD [22]	N/A	N/A	N/A	90.1	N/A	N/A
Our	**11,188,534**	**22.7**	**23.9**	**95.8**	**63.1**	**17.6**

network weight are half of YOLOv5m. For speed evaluation, the proposed network is better than the large-size models YOLOv5l (36.2 ms ↓) and the medium-size model YOLOv5m (3.2 ms ↓) in the same testing conditions. Several qualitative results on each dataset are shown in Fig. 5. Besides, this experiment also compares the performance of the proposed method and YOLOv5m as the results are shown in Fig. 6. These results prove that the proposed method is better than YOLOv5m when detecting the car in a dark color, overlapping, and crowded condition.

Table 3. Comparison result of proposed car detection network with other networks and retrained YOLOv5 on PUCPR+ validation set. The symbol "†" denotes the retrained networks. N/A means not-available values.

Models	Parameter	Weight (MB)	GFLOPs	mAP@0.5	mAP@0.5:0.95	Inf. time (ms)
YOLOv5l†	46,631,350	93.7	114.2	96.4	53.8	22.4
YOLOv5m†	21,056,406	42.4	50.4	97.7	60.5	15.6
YOLOv5s†	7,022,326	14.3	15.8	84.6	38.9	7.4
YOLOv5n†	1,765,270	3.7	4.2	89.7	41.6	5.9
SSD [22]	N/A	N/A	N/A	32.6	N/A	N/A
YOLO9000 [22]	N/A	N/A	N/A	12.3	N/A	N/A
YOLOv3 [22]	N/A	N/A	N/A	95.0	N/A	N/A
YOLOv4 [22]	N/A	N/A	N/A	94.1	N/A	N/A
SA+CF+CRT [22]	N/A	N/A	N/A	92.9	N/A	N/A
SF-SSD [22]	N/A	N/A	N/A	90.8	N/A	N/A
Our	**11,188,534**	**22.7**	**23.9**	**97.4**	**58.0**	**17.9**

From the mentioned results, the proposed network has a balance in performance, speed, and network parameters. Therefore, it can be implemented in smart parking systems on low-computing and embedded devices. However, the process of testing this network also revealed some disadvantages. Since the car detection network is mainly based on the signal obtained from the drone-view or floor-view camera, it is influenced by a number of environmental factors, including illumination, weather,

Fig. 5. The qualitative results of the proposed network on the validation set of the CarPK, PUCPR+, and AVPL datasets with IoU threshold = 0.5 and confidence score = 0.5.

Table 4. Comparison result of proposed car detection network with retrained YOLOv5 on the AVPL validation set.

Models	Parameter	Weight (MB)	GFLOPs	mAP@0.5	mAP@0.5:0.95	Inf. time (ms)
YOLOv5l	46,631,350	93.7	114.2	96.0	76.1	65.7
YOLOv5m	21,056,406	42.4	50.4	96.4	76.4	32.7
YOLOv5s	7,022,326	14.3	15.8	95.9	75.7	13.3
YOLOv5n	1,765,270	3.7	4.2	96.3	74.3	4.6
Our	**11,188,534**	**22.7**	**23.9**	**97.0**	**76.2**	**29.5**

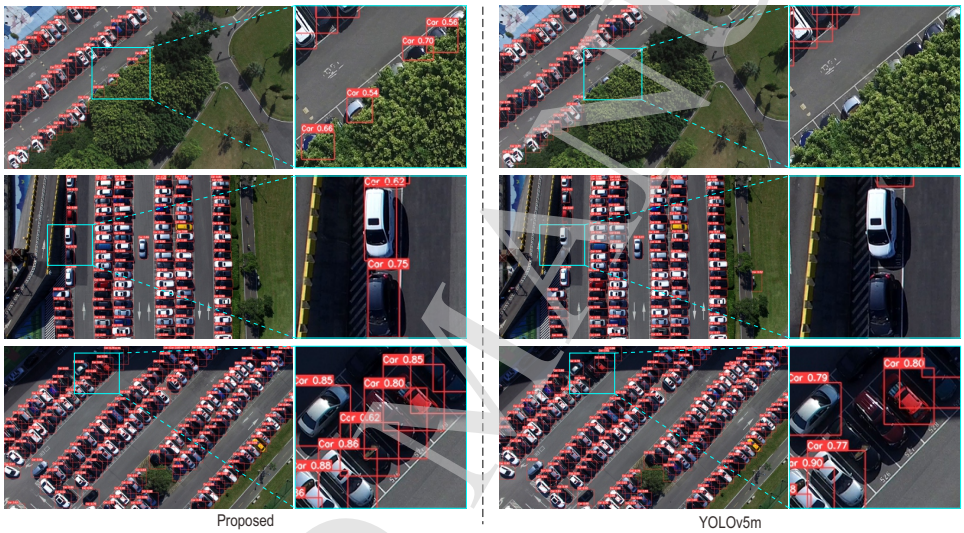

Fig. 6. The comparison result between the proposed method and YOLOv5m on the CarPK dataset with IoU threshold and confidence are set to 0.5.

car density, occlusion, shadow, object similarity, and the distance from the camera to the cars.

4.4. *Ablation studies*

The experiment conducted several ablation studies to inspect the importance of each block in the proposed backbones. The blocks are replaced in turn, trained on the CarPK training set, and evaluated on the CarPK validation set as shown in Table 4. The results from this table show that the PP-LCNet block increases the network performance at mAP@ 0.5 (1.1% ↑) but decreases in mAP@0.5:0.95 (0.8% ↓) when compared to the LiteEfficientNet block. Combining these two blocks gives a perfect result along with the starting Conv and the ending SPP blocks. Besides, it also shows the superiority of the SPP block (0.4% ↑ of mAP@0.5 and mAP@0.5:0.95) over the SPPF block when they generate the same GFLOPs and network parameters.

Table 5. Ablation studies with different types of backbones on the CarPK validation set.

Blocks	**Proposed backbones**			
Conv	✓	✓	✓	✓
PP-LCNet		✓	✓	✓
LiteEfficientNet	✓		✓	✓
SPPF			✓	
SPP	✓	✓		✓
Parameter	10,728,766	9,780,850	11,188,534	11,188,534
Weight (MB)	21.9	19.9	22.7	22.7
GFLOPs	20.8	18.5	23.9	23.9
mAP@0.5	95.1	94.3	95.4	**95.8**
mAP@0.5:0.95	58.2	59.3	62.7	**63.1**

5. Conclusion

This paper introduces an improved YOLOv5 architecture for car detection in smart parking systems. The proposed network contains three main modules: backbone, neck, and detection head. The backbone module is redesigned using lightweight architectures: PP-LCNet and LiteEfficientNet. The neck and detection head reuse the structure from the original YOLOv5m with several minor modifications. The network achieves the best results at 95.8%, 97.4%, and 97.0% of mAP@0.5 and better performances when compared to recent works. The optimization of network parameters, speed, and detection accuracy provides the ability to deploy the model on real-time systems. In the future, the neck and detection head modules will be developed to detect smaller vehicles and implement them on larger datasets. Moreover, the improved method also will be compared to the latest YOLOv8 version to inspect the efficiency and novelty of the proposed architecture.

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. Scotiabank, Number of cars sold worldwide from 2010 to 2022, with a 2023 forecast (in million units) https://www.statista.com/statistics/200002/international-car-sales-since-1990/, note = Accessed: Jan. 01, 2023. [Online]. Available: https://www.statista.com/statistics/200002/international-car-sales-since-1990/.
2. F. Zhou and Q. Li, Parking guidance system based on zigbee and geomagnetic sensor technology, in *2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 2014, pp. 268–271.
3. Y. Shao, P. Chen and C. Tongtong, A grid projection method based on ultrasonic sensor for parking space detection07 2018, pp. 3378–3381.
4. H.-C. Chen, C.-J. Huang and K.-H. Lu, Design of a non-processor obu device for park-

ing system based on infrared communication, in *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2017, pp. 297–298.
5. C. Yuan and L. Qian, Design of intelligent parking lot system based on wireless network, in *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 3596–3601.
6. X. Ding and R. Yang, Vehicle and parking space detection based on improved yolo network model, *Journal of Physics: Conference Series* **1325** (10 2019) p. 012084.
7. R. Martín Nieto, García-Martín, A. G. Hauptmann and J. M. Martínez, Automatic vacant parking places management system using multicamera vehicle detection, *IEEE Transactions on Intelligent Transportation Systems* **20**(3) (2019) 1069–1080.
8. S. N. R. Mettupally and V. Menon, A smart eco-system for parking detection using deep learning and big data analytics, in *2019 SoutheastCon*, 2019, pp. 1–4.
9. G. Jocher and et al., ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements (October 2020).
10. W. Y. XU Zihao, HUANG Weiquan, Multi-class vehicle detection in surveillance video based on deep learning, *Journal of Computer Applications* **39**(3) (2019) 700–705.
11. S. Zhang and X. Wang, Human detection and object tracking based on histograms of oriented gradients, in *2013 Ninth International Conference on Natural Computation (ICNC)*, 2013, pp. 1349–1353.
12. 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*, **1**2001, pp. I–I.
13. Y. Freund and R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in *Computational Learning Theory*, ed. P. Vitányi (Springer Berlin Heidelberg, Berlin, Heidelberg, 1995), pp. 23–37.
14. V. Mitra, C.-J. Wang and S. Banerjee, Text classification: A least square support vector machine approach, *Applied Soft Computing* **7** (06 2007) 908–914.
15. N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan and M. Zuair, Deep learning approach for car detection in uav imagery, *Remote Sensing* **9** (03 2017) 1–15.
16. F. Zhao, Q. Kong, Y. Zeng and B. Xu, A brain-inspired visual fear responses model for uav emergent obstacle dodging, *IEEE Transactions on Cognitive and Developmental Systems* **12**(1) (2020) 124–132.
17. S. Chen, S. Zhang, J. Shang, B. Chen and N. Zheng, Brain-inspired cognitive model with attention for self-driving cars, *IEEE Transactions on Cognitive and Developmental Systems* **11**(1) (2019) 13–25.
18. M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma and C. Piao, Uav-yolo: Small object detection on unmanned aerial vehicle perspective, *Sensors* **20**(8) (2020).
19. X. Liang, J. Zhang, L. Zhuo, Y. Li and Q. Tian, Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis, *IEEE Transactions on Circuits and Systems for Video Technology* (2019) 1758–1770.
20. M. H. Hamzenejadi and H. Mohseni, Real-time vehicle detection and classification in uav imagery using improved yolov5, in *2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2022, pp. 231–236.
21. Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang and X. Guo, Real-time vehicle detection based on improved yolo v5, *Sustainability* **14**(19) (2022).
22. J. Yu, H. Gao, J. Sun, D. Zhou and Z. Ju, Spatial cognition-driven deep learning for car detection in unmanned aerial vehicle imagery, *IEEE Transactions on Cognitive and Developmental Systems* **14**(4) (2022) 1574–1583.
23. 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* **abs/2109.15099** (2021).
24. M. Tan and Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, *CoRR* **abs/1905.11946** (2019).
25. A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le and H. Adam, Searching for mobilenetv3 (2019).
26. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye and D. Ren, Distance-iou loss: Faster and better learning for bounding box regression, *CoRR* **abs/1911.08287** (2019).
27. U. Ruby and V. Yendapalli, Binary cross entropy with deep learning technique for image classification, *International Journal of Advanced Trends in Computer Science and Engineering* **9** (10 2020).
28. M. Hsieh, Y. Lin and W. H. Hsu, Drone-based object counting by spatially regularized regional proposal network, *CoRR* **abs/1707.05972** (2017).
29. Hemateja, Drone car counting dataset yolo (2021), Accessed: Aug. 21, 2023. [Online]. Available: https://www.kaggle.com/datasets/ahemateja19bec1025/drone-car-counting-dataset-yolo.
30. Braunge, Aerial view car detection for yolov5 (2023), Accessed: Aug. 21, 2023. [Online]. Available: https://www.kaggle.com/datasets/braunge/aerial-view-car-detection-for-yolov5.