Car Detector Based on YOLOv5 for Parking Management

Duy-Linh Nguyen[0000−0001−6184−4133], Xuan-Thuy Vo[0000−0002−7411−0697], Adri Priadana[0000−0002−1553−7631], and Kang-Hyun Jo[0000−0002−4937−7082]

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, acejo@ulsan.ac.kr

Abstract. Nowadays, YOLOv5 is one of the most widely used object detection network architectures in real-time systems for traffic management and regulation. To 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. 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 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 % and 97.4 % of mAP@0.5 on the Car Parking Lot Dataset and the Pontifical Catholic University of Parana+ Dataset, respectively.

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 [15], 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, the management and development of tools to support parking lots are essential. To construct smart parking lots, researchers propose many methods based on geomagnetic [25], ultrasonic [16], infrared [2], and wireless techniques [21]. 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 [5, 12, 13]. Following that trend, this paper proposes a car detector to support smart parking management. This work explores lightweight network architectures and redesigned modules inside of the YOLOv5 network 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 parking management and other related fields of computer vision.
2 - The proposed detector performs better than other detectors on the Car Parking Lot Dataset and the Pontifical Catholic University of Parana+ 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), Histograms of Oriented Gradients (HOG), and Haar-like features [18, 19, 22]. Then, the feature vectors go through classifiers like the Support Vector Machine (SVM) and Adaboost [6, 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 [1, 24, 3] fuse the modern CNNs and SVM networks to achieve high spatial resolution in vehicle count detection and counting. Research in [11] 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 [10] proposes a new feature-matching method and a spatial context analysis for pedestrian-vehicle discrimination. An improved YOLOv5 network architecture is designed

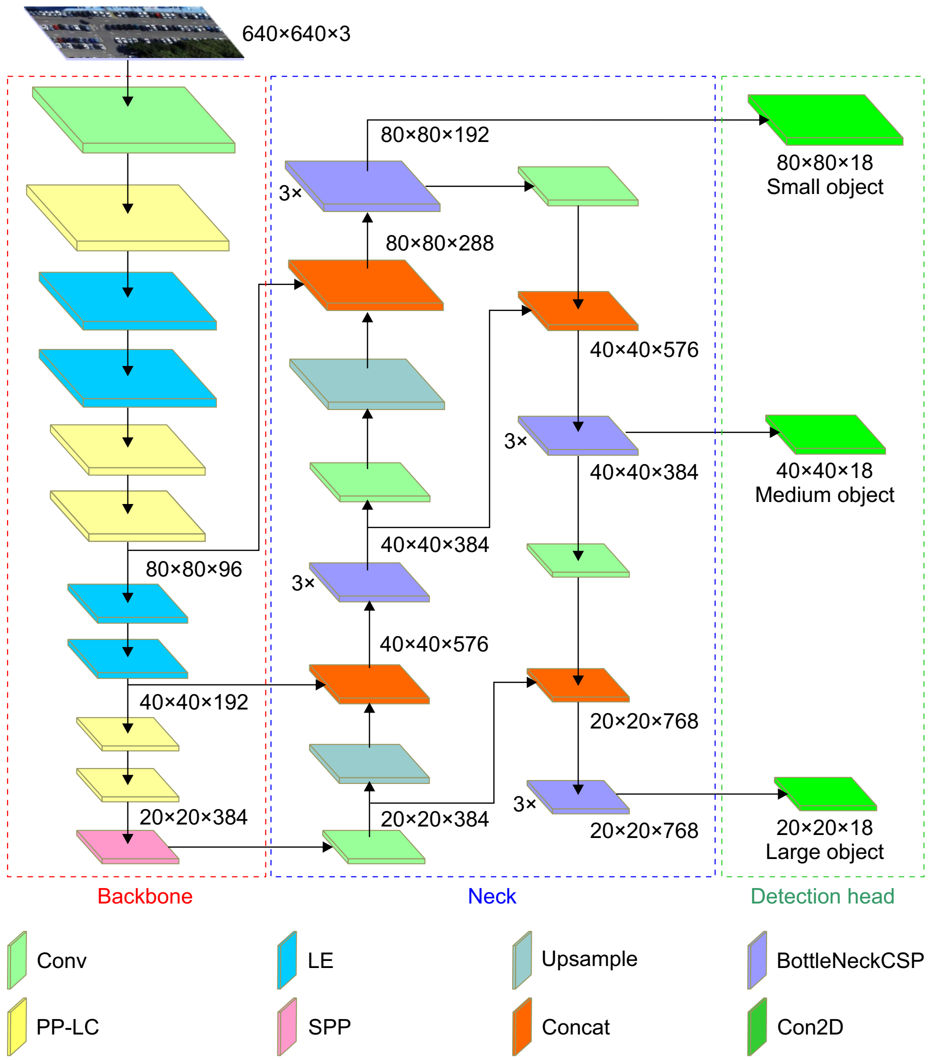

Fig. 1. The architecture of proposed car detection network. LE: LiteEfficientNet, PP-LC: PP-LCNet, SPP: Spatial Pyramid Pooling, CSP: Cross Stage Partial.

by [7] for vehicle detection and classification in Unmanned Aerial Vehicle (UAV) imagery and [23] for real-world imagery. Another study in [20] 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 [9] 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 (Con2D) with kernel size of 1×1 followed by a batch normalization (BN) and a ReLU activation function as shown in Fig. 2 (a). Subsequent blocks in the backbone module are also

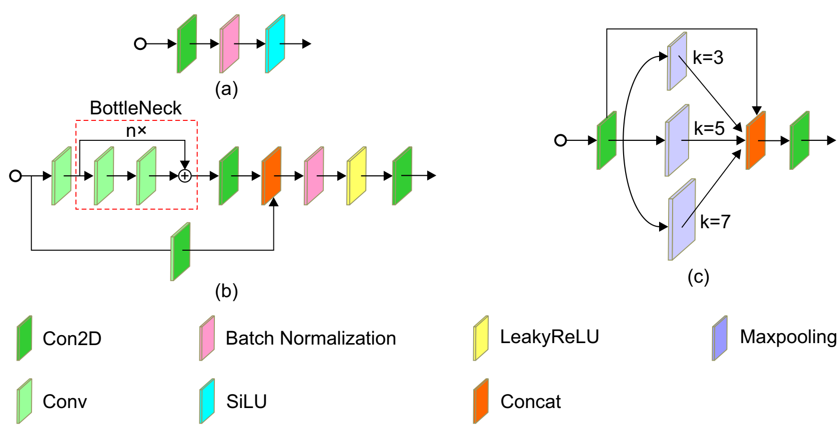

Fig. 2. The architecture of Conv (a), CSPBottleNeck (b), and SPP (c) blocks.

redesigned based on inspiration from lightweight network architectures such as PP-LCNet [4] and EfficientNet [17]. The design of the PP-LCNet (PP-LC) 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 Con2D). In between these layers, the BN and the hardswish

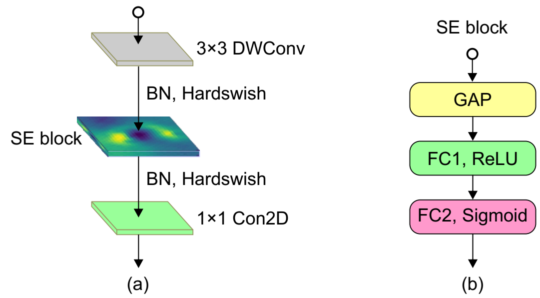

Fig. 3. The architecture of PP-LCNet (a) and SE (b) blocks.

activation function 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 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

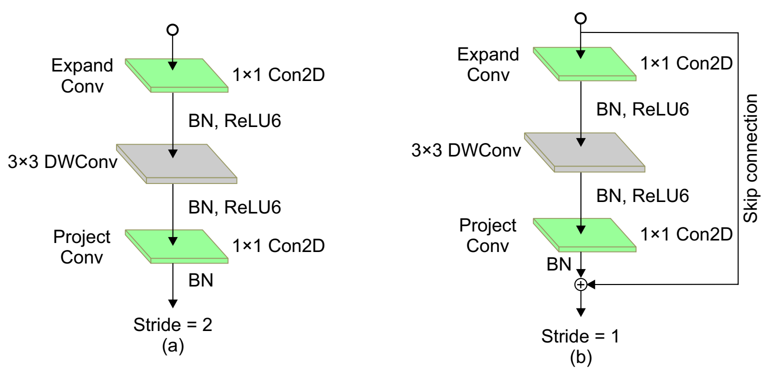

Fig. 4. The two types of LiteEfficientNet (LE) architecture, stride = 2 (a) and stride = 1 (b)

is LiteEfficientNet (LE). 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 Con2D), a depth-wise convolution layer (3×3 DWConv), and ends with a project convolution layer (1×1 Con2D). 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. This block extracts the feature maps on 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 following the original YOLOv5. This 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)	$20 \times 20 \times 21$	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 to compute the bounding box regression. The object confidence score loss $\mathcal{L}_{obj}$ and the classification loss $\mathcal{L}_{cls}$ using Binary Cross Entropy loss to calculate. λ_{box}, λ_{obj}, and λ_{cls} are balancing parameters.

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+) [8]. 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.

4.2 Experimental setup

The proposed network is conducted on the Pytorch framework and the Python programming language. 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 10^3 and ends at 10^5. The momentum set at 0.8 and then increased 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 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 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 and the four versions of YOLOv5 architectures (l, m, s, n). Then, it compares the results obtained with the results in [7, 20] 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). 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 retrained YOLOv5m network, nearly 1.5 times lower than the retrained YOLOv5l network, and quite lower than other experiments in [7] 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

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 [7]	N/A	167.0	205.0	94.5	57.9	138.2
YOLOv5l [7]	N/A	90.6	108.0	95.0	59.2	72.1
YOLOv5m [7]	N/A	41.1	48.0	94.6	57.8	40.4
Modified YOLOv5 [7]	N/A	44.0	57.7	94.9	61.1	50.5
SSD [20]	N/A	N/A	N/A	68.7	N/A	N/A
YOLO9000 [20]	N/A	N/A	N/A	20.9	N/A	N/A
YOLOv3 [20]	N/A	N/A	N/A	85.3	N/A	N/A
YOLOv4 [20]	N/A	N/A	N/A	87.81	N/A	N/A
SA+CF+CRT [20]	N/A	N/A	N/A	89.8	N/A	N/A
SF-SSD [20]	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**

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 $\uparrow$) and lower than the retrained YOLOv5l network (4.5 ms $\downarrow$). The comparison results are shown in Table 3 and several qualitative results are shown in Fig. 5.

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 [20]	N/A	N/A	N/A	32.6	N/A	N/A
YOLO9000 [20]	N/A	N/A	N/A	12.3	N/A	N/A
YOLOv3 [20]	N/A	N/A	N/A	95.0	N/A	N/A
YOLOv4 [20]	N/A	N/A	N/A	94.1	N/A	N/A
SA+CF+CRT [20]	N/A	N/A	N/A	92.9	N/A	N/A
SF-SSD [20]	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 parking management 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, car density, occlusion, shadow, objects similarity,

Fig. 5. The qualitative results and several mistakes of the proposed network on the validation set of the CarPK and PUCPR+ datasets with IoU threshold = 0.5 and confidence score = 0.5. Yellow circles denote the wrong detection areas.

and the distance from the camera to the cars. Several mistaken cases are listed in Fig. 5 with yellow circles.

4.4 Ablation study

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% $\uparrow$) but decreased in mAP@0.5:0.95 (0.8% $\downarrow$) 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% $\uparrow$ of mAP@0.5 and mAP@0.5:0.95) over the SPPF block when they generate the same GFLOPs and network parameters.

Table 4. 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 parking management 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 network achieves 95.8 % of mAP@0.5 and 63.1 % of mAP@0.5:0.95 and better performance results when compared to recent works. The optimization of network parameters, speed, and detection accuracy provides the ability to deploy on real-time systems. In the future, the neck and detection head modules will be developed to detect smaller vehicles and implement on larger datasets.

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. Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., Zuair, M.: Deep learning approach for car detection in uav imagery. Remote Sensing **9**, 1–15 (03 2017). https://doi.org/10.3390/rs9040312
2. Chen, H.C., Huang, C.J., Lu, K.H.: Design of a non-processor obu device for parking system based on infrared communication. In: 2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW). pp. 297–298 (2017). https://doi.org/10.1109/ICCE-China.2017.7991113
3. Chen, S., Zhang, S., Shang, J., Chen, B., Zheng, N.: Brain-inspired cognitive model with attention for self-driving cars. IEEE Transactions on Cognitive and Developmental Systems **11**(1), 13–25 (2019). https://doi.org/10.1109/TCDS.2017.2717451
4. Cui, C., Gao, T., Wei, S., Du, Y., Guo, R., Dong, S., Lu, B., Zhou, Y., Lv, X., Liu, Q., Hu, X., Yu, D., Ma, Y.: Pp-lcnet: A lightweight cpu convolutional neural network (2021)
5. Ding, X., Yang, R.: Vehicle and parking space detection based on improved yolo network model. Journal of Physics: Conference Series **1325**, 012084 (10 2019). https://doi.org/10.1088/1742-6596/1325/1/012084
6. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) Computational Learning Theory. pp. 23–37. Springer Berlin Heidelberg, Berlin, Heidelberg (1995)
7. Hamzenejadi, M.H., Mohseni, H.: Real-time vehicle detection and classification in uav imagery using improved yolov5. In: 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE). pp. 231–236 (2022). https://doi.org/10.1109/ICCKE57176.2022.9960099
8. Hsieh, M., Lin, Y., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal network. CoRR **abs/1707.05972** (2017), http://arxiv.org/abs/1707.05972
9. Jocher, G., et al.: ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements (Oct 2020). https://doi.org/10.5281/zenodo.4154370, https://doi.org/10.5281/zenodo.4154370
10. Liang, X., Zhang, J., Zhuo, L., Li, Y., Tian, Q.: 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 pp. 1758–1770 (2019)
11. Liu, M., Wang, X., Zhou, A., Fu, X., Ma, Y., Piao, C.: Uav-yolo: Small object detection on unmanned aerial vehicle perspective. Sensors **20**(8) (2020). https://doi.org/10.3390/s20082238, https://www.mdpi.com/1424-8220/20/8/2238
12. Martín Nieto, R., García-Martín, , Hauptmann, A.G., Martínez, J.M.: Automatic vacant parking places management system using multicamera vehicle detection. IEEE Transactions on Intelligent Transportation Systems **20**(3), 1069–1080 (2019). https://doi.org/10.1109/TITS.2018.2838128

13. Mettupally, S.N.R., Menon, V.: A smart eco-system for parking detection using deep learning and big data analytics. In: 2019 SoutheastCon. pp. 1–4 (2019). https://doi.org/10.1109/SoutheastCon42311.2019.9020502
14. Mitra, V., Wang, C.J., Banerjee, S.: Text classification: A least square support vector machine approach. Applied Soft Computing **7**, 908–914 (06 2007). https://doi.org/10.1016/j.asoc.2006.04.002
15. 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/
16. Shao, Y., Chen, P., Tongtong, C.: A grid projection method based on ultrasonic sensor for parking space detection. pp. 3378–3381 (07 2018). https://doi.org/10.1109/IGARSS.2018.8519022
17. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks (2020)
18. Viola, P., Jones, M.: 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). https://doi.org/10.1109/CVPR.2001.990517
19. XU Zihao, HUANG Weiquan, W.Y.: Multi-class vehicle detection in surveillance video based on deep learning. Journal of Computer Applications **39**(3), 700 (2019)
20. Yu, J., Gao, H., Sun, J., Zhou, D., Ju, Z.: Spatial cognition-driven deep learning for car detection in unmanned aerial vehicle imagery. IEEE Transactions on Cognitive and Developmental Systems **14**(4), 1574–1583 (2022). https://doi.org/10.1109/TCDS.2021.3124764
21. Yuan, C., Qian, L.: Design of intelligent parking lot system based on wireless network. In: 2017 29th Chinese Control And Decision Conference (CCDC). pp. 3596–3601 (2017). https://doi.org/10.1109/CCDC.2017.7979129
22. Zhang, S., Wang, X.: Human detection and object tracking based on histograms of oriented gradients. In: 2013 Ninth International Conference on Natural Computation (ICNC). pp. 1349–1353 (2013). https://doi.org/10.1109/ICNC.2013.6818189
23. Zhang, Y., Guo, Z., Wu, J., Tian, Y., Tang, H., Guo, X.: Real-time vehicle detection based on improved yolo v5. Sustainability **14**(19) (2022). https://doi.org/10.3390/su141912274, https://www.mdpi.com/2071-1050/14/19/12274
24. Zhao, F., Kong, Q., Zeng, Y., Xu, B.: A brain-inspired visual fear responses model for uav emergent obstacle dodging. IEEE Transactions on Cognitive and Developmental Systems **12**(1), 124–132 (2020). https://doi.org/10.1109/TCDS.2019.2939024
25. Zhou, F., Li, Q.: 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. pp. 268–271 (2014). https://doi.org/10.1109/DCABES.2014.58