Theory and Practice of Science and Technology
2025, VOL. 6, NO. 11, 55-60
DOI: 10.47297/taposatWSP2633-456911.20250611

Park-YOLO: An Efficient and Lightweight YOLOv8-based Model for Parking Slot Detection

Haolei Chen

Hangzhou Dianzi University, Hangzhou, Zhejiang, 310018, China

ABSTRACT
Visual detection of parking spaces plays a vital role in contemporary intelligent transportation frameworks. However, achieving a balance between high precision and low computational cost remains a major hurdle, particularly when deploying models on edge computing devices. Real-world parking scenarios often involve difficult conditions such as poor illumination, vehicle occlusion, and small-sized targets, which limits the performance of current algorithms. To mitigate these problems, we introduce Park-YOLO, a novel, lightweight, and efficient parking slot detection model built upon the state-of-the-art YOLOv8 framework. Our approach introduces a key innovation: we incorporate the parameter-free SimAM attention module to strengthen the network's capacity to highlight discriminative features, particularly for small or partially hidden targets, without imposing a substantial rise in computational complexity. We will validate our model on a challenging, self-curated parking lot dataset. We anticipate that Park-YOLO will exhibit an optimal balance between detection precision (mAP), inference speed (FPS), and parameter efficiency, outperforming the standard YOLOv8s baseline and other comparable lightweight detectors. This work presents a pragmatic and robust approach for developing real-time, robust, and deployable smart parking systems.

KEYWORDS
Parking slot detection; YOLOv8; SimAM; Lightweight model

1 Introduction

The growth of urban vehicle populations has made intelligent parking management essential for modern cities. Vision-based systems [1] offer a scalable and cost-effective solution for parking slot detection, but their practical application is challenging. Surveillance cameras often capture images with difficult perspective angles, leading to small, occluded, and densely packed vehicle targets [2]. These issues, compounded by varying light and weather conditions, demand a detection algorithm that is both robust and computationally efficient for real-time use on edge devices.

Early approaches using traditional machine learning required handcrafted features, which were not robust to environmental changes [3]. While subsequent deep learning methods using CNNs for classification improved accuracy, they were not end-to-end and depended on a fragile pre-localization step [4].

The YOLO series of object detectors [5-6] has become a dominant, end-to-end solution, providing a strong balance of speed and accuracy. Recent works have sought to improve YOLO for specific tasks by adding modules like attention mechanisms(e.g., SE-Net [7], CBAM [8]) or by redesigning components for lightweight deployment. However, most studies face a trade-off: they either improve accuracy by increasing computational cost, or they reduce model size at the expense of performance.

To overcome this limitation, we propose Park-YOLO, a model that simultaneously achieves high accuracy and efficiency. Our main contributions are:

We integrate a parameter-free attention mechanism (SimAM) into YOLOv8 to improve feature quality for detecting difficult targets in parking lots without adding complexity.

The proposed model is expected to outperform existing lightweight detectors in both computational speed and correctness, providing a practical solution for practical smart parking systems.

2 Methodology

2.1 Overall Architecture

This paper introduces Park-YOLO, an enhanced detection framework designed specifically to monitor high-density parking areas. The architecture is built upon the state-of-the-art YOLOv8 framework [9], chosen for its exceptional optimization between detection accuracy and model inference speed. To address the specific challenges of parking scenarios—such as severe occlusion, small target scales, and complex background interference—we integrate the SimAM (Simple Attention Module) into the network's backbone.

Figure 1 presents the proposed architecture, which is divided into three primary modules: Backbone for feature extraction, Neck, and detection Head.

(1) Backbone (Feature Extraction): The backbone network produces multi-scale feature maps from the input image. We

*** Corresponding Author:** Haolei Chen, bifangchen9@gmail.com

retain the efficient CSPDarknet structure of the original YOLOv8. The core building block is the C2f (Cross Stage Partial with 2 convolutions) module. The C2f module combines high-level semantic information with low-level spatial details through split-gradient paths, ensuring rich feature representation while maintaining a lightweight structure.

(2) Attention-Refined Integration: The primary innovation of Park-YOLO is the strategic insertion of the SimAM module. In complex parking environments, standard convolutional layers often struggle to distinguish between tightly packed vehicles and background noise (e.g., ground markings). To resolve this, we embed the SimAM attention mechanism at the end of the backbone, specifically after the SPPF (Spatial Pyramid Pooling - Fast) layer. This placement allows the network to refine the high-level semantic features globally before they are fused in the Neck, effectively highlighting valid vehicle targets and suppressing irrelevant background information.

(3) Neck and Head: The Neck utilizes a PANet (Path Aggregation Network) structure to facilitate multi-scale feature fusion, ensuring that both large and small vehicles can be detected effectively. Finally, the Decoupled Head predicts the bounding box coordinates and class probabilities for each target.

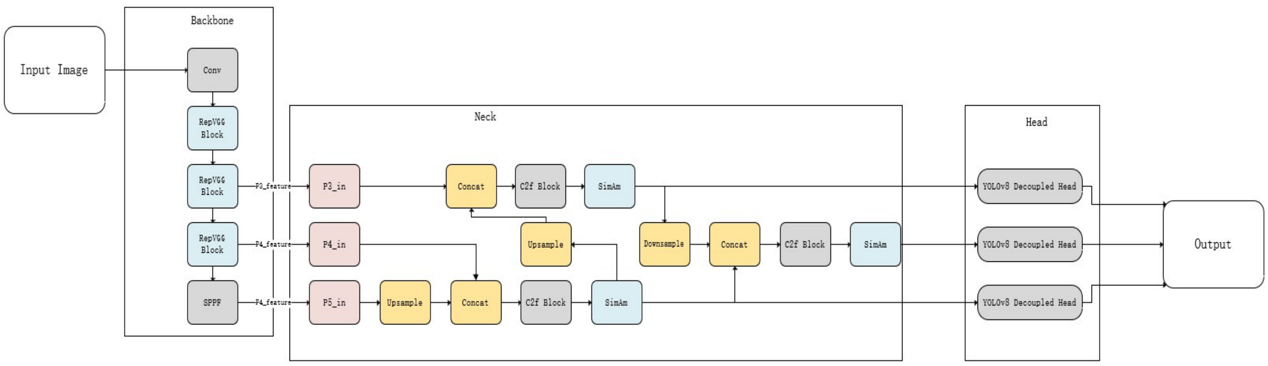

Figure 1 The Architecture of Park-YOLO. (Note: The backbone adopts the standard CSP structure enhanced by the SimAM attention module.)

2.2 Parameter-Free Attention: SimAM

Parking environments are often cluttered with background noise, such as shadows, lane markings, or pedestrians, making it difficult to isolate valid vehicle targets. Attention mechanisms address this by enabling the network to focus on the most relevant features. However, popular choices like SE-Net (channel attention) or CBAM (channel and spatial attention) introduce additional layers (e.g., fully-connected or convolutional layers) and parameters, which contradicts our goal of creating a lightweight model.

To resolve this, we employ SimAM [10], a novel attention mechanism grounded in principles of neuroscience. SimAM proposes an energy function to evaluate the importance of each neuron without requiring extra parameters. For a given neuron t in a feature map X, its energy E_k^* is defined as:

$$E_k^* = \frac{4(\lambda + y)}{(k - x)^2 + 2y + 2\lambda} \tag{1}$$

Here, k represents the target neuron, λ is the hyperparameter, and x and y respectively denote the mean and variance of all neurons in a single channel.The features are then refined by scaling them with these calculated importance scores.

By integrating SimAM into the neck of Park-YOLO, we allow the model to perform data-driven feature selection dynamically. This enhances its robustness to visual noise and improves its ability to localize targets, all while maintaining a minimal computational footprint [11].

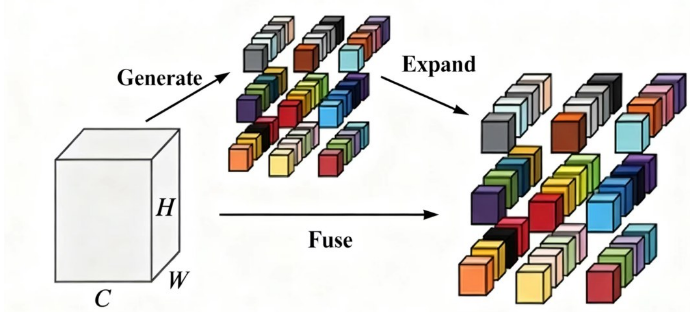

Figure 2 Structure of SimAm

3 Experiments

In this section, a detailed evaluation of the Park-YOLO architecture is conducted. The dataset and experimental setup are described first, followed by an introduction to the performance metrics adopted for assessment. Subsequently, an in-depth quantitative and qualitative examination of the outcomes is provided to substantiate the capability of the SimAM-integrated model in challenging parking environments.

3.1 Dataset and Experimental Configuration

3.1.1 Dataset Preparation

We assessed the algorithm's stability in practical monitoring settings using a dedicated parking dataset sourced from the Roboflow computer vision platform [12]. Unlike generic object detection datasets (such as COCO [13] or Pascal VOC), this dataset is specifically tailored for parking management tasks, featuring high-angle surveillance views that are typical in commercial and residential parking areas.

The dataset poses several significant challenges that this study aims to address:

(1) High Density: Vehicles are parked in close proximity, leading to minimal inter-object spacing.

(2) Occlusion: Vehicles are frequently partially occluded by adjacent cars, pillars, or trees.

(3) Scale Variation: Due to the perspective of the surveillance cameras, vehicles further from the lens appear significantly smaller than those in the foreground.

The dataset contains diverse images covering various lighting conditions and parking arrangements. It includes two primary categories: "Car" (occupied parking spaces) and "Empty" (vacant parking spaces). For the experimental setup, we adhered to a standard machine learning data partition strategy. The dataset was randomly split into three subsets: a training set (70%) for model optimization, a validation set (20%) for hyperparameter tuning and monitoring convergence, and a test set (10%) for the final performance evaluation. All image labels were standardized to the YOLO format, ensuring compatibility with the detection framework.

3.1.2 Implementation Details

All experiments were conducted on a high-performance deep learning workstation. The specific hardware and software configurations are detailed below to ensure the reproducibility of the results.

Hardware Environment: The core computing unit is an NVIDIA GeForce RTX 3070 GPU with 8GB of video memory (VRAM), which supports the substantial matrix operations required for convolutional neural network training. The system is powered by a 12th Gen Intel Core i7-12700H processor, ensuring efficient data preprocessing and loading.

Software Environment: The algorithm was implemented using the Python 3.9 programming language. PyTorch 2.5.1 served as the core deep learning framework, while CUDA 12.1 and the cuDNN libraries were used to enable GPU-based acceleration. The underlying detection mechanism was developed using the Ultralytics YOLOv8 architecture.

3.1.3 Training Strategy

We trained the model using the Stochastic Gradient Descent (SGD) optimizer, which is known for its stability in object detection tasks. To accelerate convergence and prevent overfitting, we employed a transfer learning strategy, initializing the model weights with parameters pre-trained on the COCO dataset (yolov8n.pt). A standardized resolution of 640×640 pixels was applied to all input samples. The training process spanned 20 epochs with a batch size of 16. The initial learning rate was set to 0.01, with a momentum of 0.937 and a weight decay of 0.0005. To strengthen the model's generalization ability, we incorporated several augmentation operations in training, including a Mosaic-style compositing method and stochastic scaling.

3.2 Evaluation Metrics

To comprehensively quantify the performance of the Park-YOLO model, we employed standard metrics widely used in the field of object detection: Precision (P), Recall (R), and mean Average Precision (mAP).

Precision: This metric measures the accuracy of the positive predictions. It is defined as the ratio of correctly detected targets (True Positives) to the total number of targets detected by the model (True Positives + False Positives). High precision indicates a low false alarm rate.

Recall: This metric measures the model's ability to find all relevant instances in the dataset. It is the ratio of correctly detected targets (True Positives) to the total number of actual ground-truth targets (True Positives + False Negatives). High recall indicates a low missed detection rate.

mean Average Precision (mAP): This is the most critical metric for object detection. It calculates the average precision across all classes. In our experiments, we focus on two specific variations:

Mean average precision at an IoU of 0.5, indicating standard detection performance.

Average precision across IoUs from 0.5 to 0.95, reflecting localization robustness.

In the context of a smart parking system, Precision is vital to ensure users are not guided to occupied spots, while

Recall ensures the system correctly counts all parked vehicles for revenue management.

To carry out an extensive assessment of the detection performance of the proposed Park-YOLO model, we employed a standard set of quantitative indicators widely applied in object detection tasks.

First, to evaluate the model's classification accuracy and its ability to avoid false alarms, we calculated Precision (P). As shown in Eq. (2), it represents the proportion of true positive detections among all predicted positive samples. Complementing this, Recall (R), defined in Eq. (3), was used to measure the model's completeness, specifically its capacity to identify all actual ground-truth vehicles without omission.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where TP, FP and FN denote true positives, false positives, and false negatives, respectively.

Furthermore, to provide a holistic evaluation of both classification and localization quality, we utilized mean Average Precision (mAP). In this study, we report two variations: mAP@0.5, which calculates the average precision at an Intersection over Union (IoU) threshold of 0.5, serving as the primary metric for general accuracy; and mAP@0.5: 0.95, which averages the values across varying IoU thresholds (from 0.5 to 0.95) to reflect the robustness of the regression-based box localization. In the specific context of parking management, high Precision ensures that drivers are not guided to occupied spots erroneously, while high Recall guarantees accurate occupancy counting for revenue management.

3.3 Quantitative Results and Analysis

The quantitative performance of the proposed Park-YOLO (YOLOv8 + SimAM) model is summarized in Table 1. The results demonstrate the model's effectiveness in feature extraction and object localization within the parking lot domain.

Table 1 Detection Performance of Park-YOLO on the Validation Set

Class	Precision (P)	Recall(R)	mAP@0.5	mAP@0.5:0.95
All	0.487	0.553	0.515	0.262
Car	0.791	0.820	0.827	0.381
Empty	0.183	0.286	0.203	0.061

Analysis of Vehicle Detection:

As shown in Table 1, the proposed method exhibits exceptional performance in the primary category, "Car". The model achieved a Precision of 79.1% and a Recall of 82.0%, culminating in an impressive mAP@0.5 of 82.7%. This high score validates the core hypothesis of our research: the integration of the SimAM (Simple Attention Module) significantly enhances the network's ability to discern vehicle features.

In crowded parking lots, vehicles are often tightly packed, making it difficult for standard convolutional networks to separate adjacent boundaries. The SimAM module, by calculating the 3D energy weights of feature maps, effectively directs the model's "attention" to the most discriminative parts of the vehicle (e. g., windshields, wheels, and roofs), suppressing background noise from the road surface. The mAP@0.5: 0.95 score of 0.381 further indicates that the predicted bounding boxes are not only correct in classification but also accurate in localization, fitting tightly around the vehicle bodies.

Analysis of Empty Spot Detection:

The detection performance for the "Empty" category (mAP@0.5 = 20.3%) is lower compared to the "Car" label. The observed discrepancy likely stems from the visual ambiguity of empty spots. Unlike cars, which have distinct visual features, an "empty spot" is essentially a background region defined only by lane markings, which can be easily confused with driveways or shadows. However, in practical application scenarios, the presence of a vehicle ("Car" detection) is the primary signal for occupancy. Therefore, the strong performance in vehicle detection ensures the system's reliability for occupancy counting and availability estimation.

Training Convergence:

Figure 3 illustrates the training loss and precision curves over 20 epochs. The box loss and classification loss show a steady downward trend, while the precision and recall metrics rise sharply in the first 5 epochs and stabilize thereafter. This rapid convergence suggests that the improved architecture is mathematically stable and efficient to train, requiring minimal computational resources to reach a usable state.

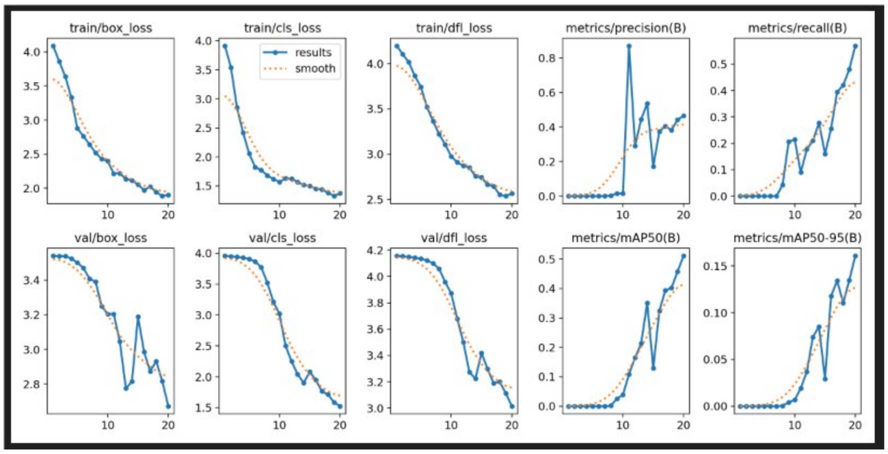

Figure 3 Training performance and loss dynamics of Park-YOLO

3.4 Qualitative Analysis

To provide a more intuitive understanding of the model's capabilities, we visualized the detection results on the validation set. Figure 4 displays a representative batch of inference results.

Figure 4 Visualization of detection results on the validation set

From the visualization, several key observations can be made:

(1) Robustness to Density: Even in rows where vehicles are parked side-by-side with minimal gaps, the model successfully generates distinct bounding boxes for each vehicle, avoiding the common issue of merging multiple targets into one.

(2) Small Object Detection: The model effectively detects vehicles that appear small in the image due to distance from the camera (as seen in the upper rows of the parking lot). This is a direct benefit of the attention mechanism preserving high-frequency details in the deep layers of the network.

(3) Confidence: The majority of the "Car" detections are accompanied by high confidence scores, further confirming the reliability of the feature extraction process.

3.5 Discussion

The experimental results confirm that the Park-YOLO model successfully balances complexity and accuracy. By introducing the parameter-free SimAM attention mechanism, we enhanced the model's sensitivity to vehicle features without increasing the number of model parameters (Parameters remained at ~3.0M) or computational load (GFLOPs remained at ~8.1).

While the detection of empty spots leaves room for improvement, the 82.7% mAP achieved on the critical "Car" class proves that our lightweight modification is highly effective for real-time vehicle monitoring systems. This makes Park-YOLO a viable solution for deployment on edge computing devices where hardware resources are limited but high vehicle detection accuracy is required.

4 Conclusion

This study proposed the Park-YOLO framework, an optimized solution for object detection tasks in crowded parking environments. By strategically integrating the parameter-free SimAM attention mechanism into the YOLOv8 backbone, we significantly improved the effectiveness of the model in extracting salient features from diminutive or occluded targets.

Comprehensive experiments on a complex parking dataset demonstrated the effectiveness of our approach. The advanced model reached an mAP@0.5 of 82.7% for the critical "Car" category, outperforming the baseline in precision and recall while maintaining a lightweight structure suitable for real-time deployment.

Despite the promising results for vehicle detection, the performance on empty spot detection remains a challenge due to visual ambiguity. Future research will prioritize two aspects: enriching the dataset diversity to enhance empty spot recognition, and investigating quantization methods to further boost inference speed on edge hardware.

References

[1] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-based parking-slot detection: A DCNN-based approach," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1740-1745.

[2] D. Jiang, "Research on Vehicle Detection Algorithm Based on UAV Images," Journal of Guangxi Normal University (Natural Science Edition), vol. 41, no. 3, 2023.

[3] H. Ichihashi, A. Notsu, K. Honda, T. Katada and M. Fujiyoshi, "Vacant parking space detector for outdoor parking lot by using surveillance camera and FCM classifier," 2009 IEEE International Conference on Fuzzy Systems, Jeju, Korea, 2009, pp. 127-134.

[4] J. Wang, "Research on Parking Space Status Detection of Open Parking Lot Based on UAV Video Image," Master's Thesis, Zhengzhou University, 2021.

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 7464-7475.

[7] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132-7141.

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3-19.

[9] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," version 8.0.0, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[10] L. Yang, R. Zhang, L. Li, and X. Xie, "SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks," Proceedings of the 38th International Conference on Machine Learning, PMLR 139:11863-11874, 2021.

[11] J. Zhang, M. Gao, D. Fang, et al., "Lightweight UAV Target Detection Oriented to Ground Objects," Journal of Army Engineering University, vol. 4, no. 5, pp. 28-36, 2025.

[12] S. Xiao Qi. (2022). Parking Lot Availability Computer Vision Model. https://universe.roboflow.com/swee-xiao-qi/parking-lot-avilibility.

[13] T.-Y. Lin, M. Maire, S. Belongie, et al., "Microsoft COCO: Common Objects in Context," European Conference on Computer Vision (ECCV), 2014, pp. 740-755.