Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00238-0

Letter

Vision-Based Smart Parking Occupancy Detection Using Deep Learning Object Detection Models

Jeevitha K [1, ✉], Mujahid Ghouse Mohamed [1], Gokulavasan S [1], Karthikeyan P [1], Amudaishwariyan V [1]

1. *Artificial Intelligence & Data Science, Sri Manakula Vinayagar Engineering College, Puducherry, IND*

Received: April 19, 2026 | Review began: June 14, 2026 | Review ended: July 27, 2026 | Published: August 23, 2026

Abstract

Finding parking in urban areas is becoming increasingly challenging as cities continue to grow. Inefficient parking utilization, excessively high infrastructure costs, and the limited ability of traditional sensor-based monitoring systems to scale make finding a solution difficult. The proposed parking occupancy detection framework is vision-based and uses the YOLOv8 object detection model to provide real-time identification of occupied and vacant spaces. The model was trained and tested using a dataset that consisted of a set of about 6,850 annotated images collected from the PKLot benchmark dataset as well as images collected from real-time parking lot monitoring on a college campus in the USA. A complete analysis of YOLOv8 models was performed by examining data augmentation techniques, robustness of YOLOv8 under different environmental conditions, and generalizability of YOLOv8 across different datasets. Results from several experiments suggested that the YOLOv8m model produced the best balance between accurate detection and computational efficiency with a mean Average Precision (mAP@0.5) of 99.4% and was capable of providing real-time inferences. Moreover, YOLOv8 demonstrated robust performance in low-light, blurry, and noisy environments when combined with different synergistic combinations of data augmentation techniques. The proposed parking occupancy detection framework represents an accurate, scalable, and cost-effective alternative to legacy parking monitoring solutions, making it appropriate for implementation in smart city and edge computing applications.

Categories: AI/ML-based decision support systems, Computer Vision, IoT Applications

Keywords: smart parking, yolov8, object detection, edge inference, parking occupancy detection

Introduction

Parking challenges in urban areas have intensified with rising vehicle ownership, where drivers waste up to 17 hours yearly circling for spaces amid severe congestion and fuel inefficiency, consuming a large share of city-center traffic and causing substantial global economic losses plus significant CO_2 emissions from wasted fuel. Traditional sensor-based systems falter under weather exposure and high per-slot costs exceeding a few hundred currency units [1]. Existing convolutional neural network (CNN)-based vision methods, including recent YOLOv8 approaches, report very high benchmark accuracy yet still show 10-20% performance drops in realistic conditions due to occlusions, lighting changes, and domain shifts. Vision systems using single overhead cameras, as adopted in this work, address these limitations by monitoring multiple slots per camera with contextual awareness of shadows and partial occlusions, reducing per-slot hardware expenditure by roughly 70-74% compared to traditional sensors. Thus, providing parking systems with smart search, personalized recommendations, and contextually aware interactions has become an important concern for both researchers and application developers.

Recent computer vision-based approaches, particularly CNN-driven methods such as YOLOv8, report high benchmark accuracy on curated datasets; however, real-world performance frequently degrades by 10-20% due to occlusions, illumination variations, camera viewpoint changes, and domain shifts. Vision systems often utilize the single overhead cameras as adopted in this work and mitigate these challenges by enabling multi-slot monitoring with contextual awareness of shadows and partial occlusions, achieving approximately 70-74% reduction in per-slot hardware costs compared to traditional sensor-based installations [2]. Moreover, these conventional parking management systems often rely only on manual monitoring or per-slot sensor deployment, both of which are costly, difficult to scale, and highly sensitive to environmental disturbances. Significant research gaps remain, and as most existing works rely on single datasets, they lack systematic degradation analysis and provide minimal evaluation of cross-dataset generalization performance [3].

Despite this rapid progress, many parking detection studies emphasize peak mAP values on clean datasets while offering limited analysis of how data augmentation strategies, model scaling, and deployment constraints affect robustness in realistic environments [4]. Classical feature-based methods such as LBP-SVM consistently underperform modern deep learning detectors in cross-view and large-scale scenarios, underscoring the necessity for carefully engineered CNN pipelines suitable for smart-city integration [5]. Recent research has examined how to detect parking occupancy using the embedded artificial intelligence (AI) camera platform with lightweight CNNs executed on low-power hardware such as the Raspberry Pi. These models were deployed in conjunction with OpenCV or other on-device inference engines for real-time performance in the smart city ecosystem [6].

The following contributions were made to the solutions to these problems:

1. Benchmarking YOLOv8 variants based on 6,850 images annotated by PKLot and real-world campus datasets to determine the effect of accuracy and latency on each of the YOLOv8 variants.

2. The application of controlled synthetic degradations on YOLOv8 and the development of robust augmentation strategies that provided an improvement of 9-13% to the performance of YOLOv8 as demonstrated in controlled synthetic degradation tests.

3. The study of cross-dataset transfer of YOLOv8, and when a YOLOv8 trained on the PKLot dataset was tested against the unseen campus dataset, the performance dropped by 10.1%.

4. The deployment of YOLOv8 was optimized for performance using ONNX-based inference technology, increasing throughput on an RTX-class GPU from 67 FPS to over 100 FPS while reducing costs by 70-74% relative to sensor-based systems.

Materials And Methods

Related work

Traditional Parking Detection

Parking detection systems based on traditional technology utilized various physical (i.e. analog) sensors to determine whether or not a vehicle is parked in a specific location, including ultrasonic and magnetic sensors. These sensors had an accuracy range while deployed in a controlled environment of between 85% and 92%. While this level of accuracy allowed for acceptable detection capabilities, the costs associated with the installation and maintenance of these systems proved to be excessive because of the extensive requirement for all cabling to run to each parking location, necessary excavations of each parking space, the time required for calibrating parking detection systems, and the regular need to replace faulty sensors. In addition, the parking detection systems based on traditional technology are susceptible to environmental variables such as rain, sand, temperature, and electromagnetic interference, so deployment in urban settings becomes difficult on a mass scale [7].

Prior to the use of deep learning techniques, early camera-based systems used traditional methods (such as background subtraction, edge detection, and manually created features) to determine if a parking spot had been occupied. These methods were susceptible to changes in illumination, shadows and obstructions (such as trees and buildings), and seasonal changes, and many times needed custom calibrations on-site to maintain consistent reliable performance over time, ultimately restricting their usage in many real-world environments [8].

Blockchain Deep Learning for Parking

Prior to the advent of CNN technology, parking space detection predominantly used classification-based techniques and these techniques were typically performed with high accuracy with the help of specific slots placed over the incoming video feeds. More recently, CNNs have provided significant improvements in parking zone detection capabilities by allowing the automatic building of visual features that maximize the usefulness of parking space detections. An application of YOLOv3 combined with a CNN-based methodology allowed for real-time vehicle detection in structured parking garages and achieved a 94% accuracy rate through bounding box regression. Through instance segmentation and determining occupancy via pixel-wise masks based on Intersection over Union (IoU), Mask R-CNN achieved a mean Average Precision (mAP) of 0.901 for the detection and detection of vehicles in parking environments. While embedded systems utilized capabilities for edge deployment to achieve 94% accuracy via a combination of YOLOv3 and CNN within real-time surveillance applications emphasizing high confidence detection on occupancy tasks, there is a lack of systematic ablation studies that are more focused on evaluating overall sensor robustness throughout changing environmental conditions than peak performance in laboratory-type setups [9].

Research Gap

The authors of YOLOv8 point out the following significant shortcomings that justified this work being done. First, no one characterizes how robust the output of IP camera systems is when facing poor conditions of illumination, as well as blurring, shadowy areas, and glare resulting from street lights. Second, there are also no available ablation studies that have tested all of the augmentation strategies for varying scale models and hyperparameters that were available to camera-based detection pipelines that have been used to test LeNet/AlexNet model architectures [10].

Systems that detect street parking show limited ability to apply YOLO models across different areas, where roads appear in different views and cars are oriented differently, thus requiring improvements in KD-tree Spatial Adaptation, optimization for the most suitable search process, and the implementation of fine-tuning processes unique to each location [11].

Currently, the automatic detection of parking spaces does not provide clear results regarding occupancy classification due to numerous variables including, but not limited to, camera placement (geometries), distortion of perspective, and environmental conditions. Therefore, manual assistance via clustering algorithms or heuristic modification is needed to process more complicated, irregularly shaped lots [12]. With so many different implementations, Data Set Splits/Random Seed/Training Configurations are hindering the ability to replicate a study's results when it comes to parking detection compared to using Open CNRPark as a baseline since Open CNRPark has been established as a baseline due to the fact that it provides a standardized set of annotated images taken from several different cameras at multiple different times of day and in numerous weather conditions making the evaluation of models consistent [13].

Proposed work

The proposed work delivers an AI-powered smart parking platform that replaces expensive, hardware-intensive sensor installations with a camera-centric, vision-based framework for parking slot occupancy detection and monitoring. Unlike existing YOLO-based smart parking systems that primarily focus on occupancy detection using a single dataset or model configuration, the proposed framework introduces a comprehensive multi-site evaluation by combining the PKLot benchmark dataset with real-world parking lot data collected from a college campus. In addition, it provides a systematic comparison of multiple YOLOv8 variants, investigates the impact of different data augmentation strategies on robustness under challenging environmental conditions (e.g., low light, blur, and noise), and evaluates cross-dataset generalizability. The framework is further designed for edge-friendly, real-time deployment in large multi-camera installations and serves as a scalable foundation for higher-level intelligent parking services, including license plate recognition, violation tracking, and decision-support analytics.

Parking Dataset, Annotations, and Splits

The dataset consists of a combination of two sources: PKLot (6,000 images) combined with a new collection of 850 images found at a local campus. This total combined dataset has 6,850 images containing 84,327 annotation files on parking slots [14]. There are many variations present in the PKLot dataset; variations of weather conditions (sunny, cloudy, rainy), times (morning, afternoon, evening), and occupancy rates (20%, 40%, 60%, 80%, 100%). However, the local campus portion provided additional information regarding layout and viewpoint information for de-coupling overfitments to any one location. Each slot contains one of two semantic classes, either "occupied" when a vehicle was previously parked in that spot, or "empty" when occupied by a visible parking line marker with no vehicles currently parked within the slot. Tight bounding boxes exist around all vehicles or lines surrounding slots. Annotation is performed using the Roboflow platform, where two independent annotators label every image and resolve disagreements through discussion to maintain consistent criteria for borderline cases such as partial occlusions or ambiguous markings.

A fixed-seed, random stratified split was applied using the random seed of 42 as the basis for creating a random stratified split of the dataset. In this split, a total of 4,795 images, or 70%, went to the training set; 1,028 images, or 15%, to the validation set; and 1,027 images, or 15%, to the testing set. The number of occupied and empty slots for each data subset was as follows: training, 36,289 occupied and 22,739 empty; validation, 7,777 occupied and 4,873 empty; testing, 7,776 occupied and 4,873 empty. The overall number of slots in the dataset shows a share of 61.5% with respect to the number of slots occupied and a share of 38.5% empty, which represents a realistic nature of how parking space is used.

A YAML file (parking_data.yaml) is used to document the dataset, including many file paths to train, val, and test image folders and identifying the two classes used in the integration with the YOLO training pipeline.

Dataset split and statistics: Random stratified split maintaining class distribution with fixed seed (42):

Training: 4,795 images (70%) - 36,289 occupied and 22,739 empty (59,028 total objects)

Validation: 1,028 images (15%) - 7,777 occupied and 4,873 empty (12,650 total objects)

Test: 1,027 images (15%) - 7,776 occupied and 4,873 empty (12,649 total objects)

Class distribution: Occupied slots represent 61.5% of the time, and empty slots represent 38.5% of the time; therefore, these percentages correspond to what you would expect from actual vehicle parking occupancy percentages. 87.6% of the images were taken from the PKLot dataset, and the remainder (12.4%) were taken from the local campus dataset sources.

The dataset configuration is specified in YAML format.

Preprocessing, Augmentation, and Robustness Design

To provide equitable input from disparate suppliers, all images are resized using bilinear interpolation at 640 × 640 pixels and letterboxing maintains the aspect ratio. The color scheme used is RGB utilizing a range of normalizations (0,1). This represents our baseline pipeline and enables us to establish robustness under realistic degradations using a structured augmented strategy. Within the augmentation strategy, we include mosaic composition for context diversity as well as a 50% probability of horizontal flipping, and hue, saturation, and value jittering (perturbations), random scaling which ranges from 0.5× to 1.5×, as well as random translations which can be as high as ±10% in each direction.

Additional augmentation sets build on this baseline to isolate the effect of specific perturbations: Set A adds small random rotations of ±10°, Set B injects Gaussian blur with a 3×3 kernel at probability 0.1, Set C applies global brightness adjustments of ±20%, and Set D combines all of these transformations to create a strongly augmented regime. These configurations are used both during training and in synthetic robustness tests, enabling quantitative analysis of how each augmentation recipe impacts performance under dark, blurred, and noisy conditions - a critical requirement for outdoor parking lots subject to varying illumination, weather, and camera quality.

YOLOv8-Based Detection Backbone and System Architecture

The detection core of the proposed platform is based on the YOLOv8 family of single-stage object detectors. In this study, four YOLOv8 variants (YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l) were systematically evaluated to analyze the trade-offs between detection accuracy, computational complexity, inference latency, and deployment efficiency for real-time smart parking applications. YOLOv8 employs a C2f-based backbone with an SPPF module for multi-scale feature extraction, together with a feature pyramid and path aggregation network that enhances feature fusion across multiple scales. Its anchor-free detection head separates classification and bounding-box regression tasks, providing stable training and efficient inference for parking occupancy detection.

All four versions, YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l, were benchmarked by the authors, measuring from 3.2M to 43.7M parameters and a range of FLOPS from 8.7G to 165.2G, respectively. These numbers correlate to model file sizes of 6.3MB-83.7MB per version. The range of these variants allows users to understand the accuracy/throughput trade-offs of deploying YOLOv8 from small-footprint embedded devices to large-capacity GPUs. Figure *1* shows how YOLOv8 is integrated into a full smart parking infrastructure.

Raw video captured from Real-Time Streaming Protocol-enabled CCTV cameras goes into a preprocessing step - the frames are normalized (made uniform) to prepare for the computer vision engine to recognize vehicles and vacant parking slots using YOLOv8 (and optionally) using DeepSORT for a consistency over time, and finally using OCR to automatically read license plates - synchronizing the results to the parking slot status, bookings, and license plate logs in real time to show on dashboards and analytic services. This study focuses on the YOLOv8 family to investigate its suitability for smart parking applications. Evaluation of newer YOLO versions and other recent object detection models is left for future work.

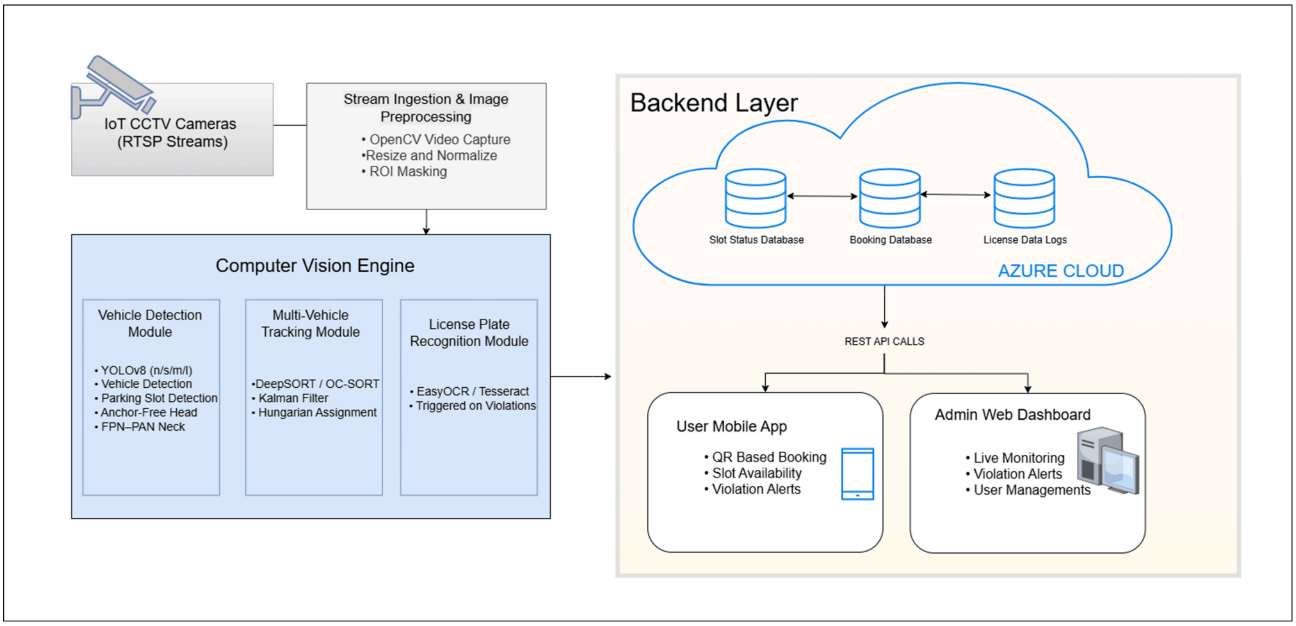

FIGURE 1: System Architecture of Smart Parking with License Plate Recognition Using Computer Vision (YOLOv8, DeepSORT, and OCR)

Training Strategy, Hyperparameters, and Environment

The experiments for both training and inference were performed on Ubuntu 22.04 using a computer that had an NVIDIA RTX 3060 (12 GB VRAM) as well as an Intel Core i7-12700K and 32 GB DDR4 RAM. All experiments were performed using Python 3.10.12, PyTorch 2.0.1, CUDA 11.8, cuDNN 8.7, Ultralytics YOLOv8 8.0.147, NumPy 1.24.3, OpenCV 4.8.0, Pillow 10.0.0, ONNX v1.14.0 and onnxruntime-gpu 1.15.1, with eight data-loader workers and a global random seed of 42 for reproducibility. The full YOLOv8m training command (which included the dataset path, hyperparameters, augmentation flags, and project naming) is documented so that practitioners can either repeat the training protocol exactly as described in the document or make use of the information to adapt the command to run on their own computer. The training and validation losses as well as detection metrics during the 100 epochs of training are summarized in Fig. 2, with precision, recall, and mAP@0.5 all exceeding 99% by approximately epoch 60, demonstrating stable optimization.

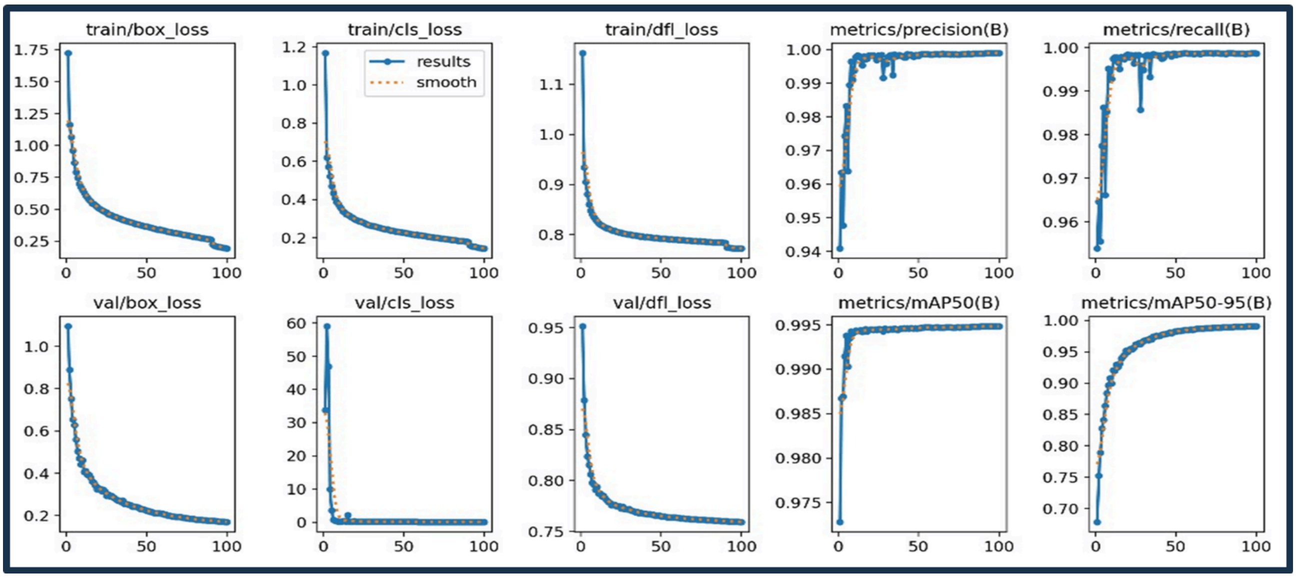

FIGURE 2: Training Metrics for YOLOv8m over 100 Epochs Showing Convergence of Precision, Recall, and mAP@0.5 to Above 99% After Epoch 60

Evaluation Protocol, Metrics, and Extensibility

The platform is evaluated using standard object detection metrics: precision, recall, average precision (area under the precision-recall curve), mAP@0.5, and mAP@0.5:0.95 averaged over IoU thresholds from 0.5 to 0.95 in steps of 0.05, along with F1-score to capture the precision-recall balance. In addition, inference latency per image and throughput in frames per second are measured for each YOLOv8 variant to quantify deployment feasibility under real-time constraints on the target RTX 3060 GPU and in ONNX-optimized edge settings. Models' robustness is examined through evaluation of the performance of models trained and validated with the defined augmentation sets on a set of images that have been degraded synthetically (e.g., via Solar/Situational Darkness, Blur, and Noise), thereby gaining a better understanding of their resilience to these types of conditions as they would typically occur in a "real-world" parking condition.

The platform is evaluated using standard object detection metrics commonly adopted in prior work on parking occupancy detection and real-time detectors. Precision and recall are defined as

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

where

TP = true positive

FP = false positive

FN = false negative.

The F1-score captures the balance between precision and recall as

$$F1 = 2.\ Precision.\ \frac{Recall}{Precision + Recall} \tag{3}$$

IoU between a predicted bounding box and ground truth box is given by

$$IoU = |Bp \cup Bgt||Bp \cap Bgt| \tag{4}$$

where

Bp = predicted bounding box

Bgt = ground truth box

$\cap$ = intersection area

$\cup$ = union area.

The precision-recall curve of a particular IoU score represents the value of Average Precision (AP) and the mAP at 0.5 IoU score for all detected objects, giving an overall evaluation of the performance. The mAP@0.5:0.95 metric represents the average AP of the detected objects at different IoU scores ranging from 0.5 to 0.95 with a step of 0.05 and provides a more stringent evaluation of the quality of object localization. The current implementation uses YOLOv8 as the main function; however, the architecture is designed modularly so that it can support other parts, such as Deep SORT for multi-object tracking, OCR programs such as EasyOCR that allow for automated license plate recognition, and rule-based engines for violation analysis and booking integration. In the future, researchers will be able to utilize transformer-based vision models, multi-task learning techniques, and techniques for reducing model size (i.e., pruning and quantization) to further improve accuracy, interpretability, and efficiency to develop a complete city-scale Smart Parking Infrastructure.

Results And Discussion

The experimental evaluation of the proposed smart parking system focuses on three major aspects: (i) accuracy-latency trade-offs across YOLOv8 variants, (ii) comparative performance against established baseline methods, and (iii) robustness under synthetic degradations. All models were trained and evaluated on the composite dataset of 6,850 images containing 84,327 annotated parking slots.

Performance of YOLOv8 Variants

Among the evaluated variants (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l), the YOLOv8-Medium model provided the most effective balance between detection accuracy and computational efficiency. On the held-out test set, YOLOv8-Medium achieved: 99.4% mAP@0.5, Precision ≈ 99%, Recall ≈ 99%, ~67 FPS on an NVIDIA RTX 3060 GPU. The YOLOv8n model achieved higher throughput (~140 FPS) but with slightly reduced detection accuracy. Conversely, YOLOv8l showed only a marginal improvement in mAP (~0.006 gain over YOLOv8m) while requiring significantly higher computational resources (165.2 GFLOPs), making it less practical for multi-camera continuous deployment.

Comparative Analysis with Baseline Methods

To validate the effectiveness of the proposed framework, a comparison was conducted against widely reported baseline methods in parking occupancy detection literature.

Earlier YOLOv3-based systems report approximately 94% detection accuracy in structured parking environments. Mask R-CNN-based approaches achieve around 0.901 mAP under controlled datasets. Lightweight MobileNetV3-based models provide improved efficiency but experience 10-20% performance degradation under real-world illumination and occlusion variations.

In contrast, the proposed YOLOv8-Medium model achieves 99.4% mAP@0.5, representing an improvement of approximately 5-6% over YOLOv3-based systems and nearly 9-10% over Mask R-CNN approaches, while sustaining real-time throughput of ~67 FPS. A summary of the comparative performance is presented in Table 1.

Method	Performance	Speed	Notes
YOLOv3 [9]	~94%	Not real-time multi-stream	Structured garages
Mask R-CNN [9]	0.901 mAP	Low FPS	Two-stage detector
MobileNetV3 [3]	High efficiency	Moderate	Sensitive to lighting
YOLOv8-M (Proposed)	**99.4% mAP@0.5**	**~67 FPS**	Real-time multi-camera

TABLE 1: Comparison of Parking Occupancy Detection Systems

FPS, Frames Per Second; mAP, Mean Average Precision; R-CNN, Region-Based Convolutional Neural Network

Conclusions

This paper applied and compared a YOLOv8-based smart parking occupancy detection system that was tested on a dataset of 6,850 images comprising 84,327 labeled parking slots. Of all the YOLOv8 models that were tested, the YOLOv8m had the best trade-off between accuracy and performance, with correct results of mAP@0.5 equal to 99.4% paired with precision and recall around 99%, while being able to run at an encouraging rate of approximately 67 FPS on an NVIDIA RTX 3060 GPU. An analysis of the experimental results confirmed the performance of the proposed framework in real-time installation of the smart parking occupancy detection system in diverse conditions. Notwithstanding the fact that the performance of the proposed technology is considerably better than the baseline detection technologies discussed in the literature, statistical analysis and comparative revisiting of the latest achievements in object detection technologies are still among the most significant areas for further research.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Jeevitha K, Mujahid Ghouse Mohamed, Gokulavasan S, Karthikeyan P, Amudaishwariyan V

Acquisition, analysis, or interpretation of data: Jeevitha K, Mujahid Ghouse Mohamed, Gokulavasan S, Karthikeyan P, Amudaishwariyan V

Drafting of the manuscript: Jeevitha K, Mujahid Ghouse Mohamed, Gokulavasan S, Karthikeyan P, Amudaishwariyan V

Critical review of the manuscript for important intellectual content: Jeevitha K, Mujahid Ghouse Mohamed, Gokulavasan S, Karthikeyan P, Amudaishwariyan V

Supervision: Jeevitha K, Mujahid Ghouse Mohamed, Gokulavasan S, Karthikeyan P, Amudaishwariyan V

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The datasets used in this study are publicly available from existing repositories (PKLot dataset). The processed data and code supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

This paper is accepted and presented in International Conference on Sustainable Smart Computing and Communications (ICSSCC 2026) organized in-person and online (Hybrid Mode) by School of Computer Science and Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune, India.

References

1. Amato G, Carrara F, Falchi F, et.al.: Car parking occupancy detection using smart camera networks and Deep learning. 2016 IEEE Symposium on Computers and Communication (ISCC). 2016, 1212-1217. 10.1109/ISCC.2016.7543901
2. Paidi V, Fleyeh H, Nyberg RG: Deep learning-based vehicle occupancy detection in an open parking lot using thermal camera. IET Intelligent Transport Systems. 2020, 14:1295-1302. 10.1049/iet-its.2019.0468
3. Yuldashev Y, Mukhiddinov M, Abdusalomov AB, Nasimov R, Cho J: Parking lot occupancy detection with improved MobileNetV3. Sensors. 2023, 23:7642 .
4. Cho KN, Oh HW, Lee SE: Vision-based parking occupation detecting with embedded AI processor. 2021 IEEE International Conference on Consumer Electronics (ICCE). 2021, 1-2. 10.1109/ICCE50685.2021.9427661
5. Encío L, Díaz C, Del-Blanco CR, Jaureguizar F, García N: Visual parking occupancy detection using extended contextual image information via a multi branch output ConvNeXt network. Sensors. 2023, 23:3329.
6. Skariah R: Effective image-based parking occupancy detection using masked region based convolutional neural network. National College of Ireland, Dublin, Ireland; 2022.
7. Begum G, Shirisha K, Rakesh E, Supraja TS: Car parking detection system using deep learning. International Journal of Advanced Engineering and Management (IJAEM). 2023, 5:850-854.
8. Farley A, Hanry H, Hendra: Real time IP camera parking occupancy detection using deep learning. Procedia Computer Science. 2021, 179:606-614. 10.1016/j.procs.2021.01.046
9. Garta IY, Manongga WE, Huang SW, Chen RC: On-street parking space detection using YOLO models and recommendations based on KD-tree suitability search. Computers, Materials & Continua. 2025, 85:4457-4471. 10.32604/cmc.2025.067149
10. Grbić R, Koch B: Automatic vision-based parking slot detection and occupancy classification. Expert Systems with Applications. 2023, 225:120147. 10.1016/j.eswa.2023.120147
11. CNRPark: A dataset for visual occupancy detection of parking lots. (2026). Accessed: Jan 01: http://cnrpark.it/.
12. Martynova A, Kuznetsov M, Porvatov V, et al.: Revising deep learning methods in parking lot occupancy detection. arXiv. 2023, 10.48550/arXiv.2306.04288

13. Zhao N, Wang K, Yang J, Luan F, Yuan L, Zhang H: CMCA-YOLO: A study on a real-time object detection model for parking lot surveillance imagery. 2024, 13:1557.

14. de Almeida PRL, Oliveira LS, Britto AS, Silva EJ, Koerich AL: PKLot - A robust dataset for parking lot classification. Expert Systems with Applications. 2015, 42:4937-4949. 10.1016/j.eswa.2015.02.009

How to cite this article:
K J, Mohamed M, S G, et al. (August 23, 2026) Vision-Based Smart Parking Occupancy Detection Using Deep Learning Object Detection Models. Cureus J Comput Sci 3 : es44389-026-00238-0. DOI https://doi.org/10.7759/s44389-026-00238-0