JUITA: Jurnal Informatika e-ISSN: 2579-8901; Vol. 13, Issue 3, November 2025

Real-Time Detection of Outdoor Parking Space Availability Using YOLOv8

Rafi Ardinata Riskiansyah[1*], Yohanes Setiawan[2], Farah Zakiyah Rahmanti[3]

[1,2,3]*Department of Information Technology, Telkom University, Surabaya Campus, Surabaya, 60231, Indonesia*

`*corr-author: rafiardinata@student.telkomuniversity.ac.id`

Abstract - Finding an empty parking spot in open areas, particularly in busy locations such as shopping centers, remains a significant challenge. This study proposes a real-time system for detecting outdoor parking space availability using the YOLOv8 algorithm, selected for its speed and accuracy in object detection. The dataset consists of 131 annotated images, expanded through three augmentation techniques (rotation, shearing, and flipping) to increase variability. Model training was performed with multiple hyperparameter configurations and evaluated using precision, recall, F1-score, accuracy, and mAP@50. The best configuration, obtained with the Adam optimizer, achieved 96.74% precision, 99.06% recall, 99.17% mAP@50, and 77.91% accuracy. While the system performed effectively and responsively in real-time daytime scenarios, a key limitation is its reduced performance under nighttime conditions due to low visibility and image noise. This research contributes by demonstrating YOLOv8's potential to improve real-time detection of parking spaces, particularly through handling occlusions and lighting variations, which remain challenges in outdoor environments.

Keywords: computer vision, parking detection, real-time, YOLOv8

I. INTRODUCTION

Parking spaces are designated areas for vehicles to stop temporarily, serving as an essential component of modern transportation systems. When managed effectively, it facilitates efficient vehicle flow and accessibility [1]. However, in outdoor parking areas, low visibility and high vehicle density often cause drivers to struggle in finding available space. This challenge frequently results in vehicles circulating unnecessarily around parking areas, which increases congestion and fuel consumption [2].

Recent advancements in computer vision (CV) and deep learning (DL) have motivated researchers to explore automated monitoring of parking areas [3]. These technologies enable real-time observation of parking spaces, thereby improving resource utilization and user convenience. Among object detection algorithms, the *You Only Look Once* (YOLO) family has gained wide adoption due to its ability to localize and classify objects simultaneously in a single forward pass [4]. Compared to traditional multi-stage approaches, YOLO achieves faster inference while maintaining high accuracy, making it well-suited for real-time applications [5]. Furthermore, YOLO's strong performance in detecting vehicles has positioned it as a cost-effective option for developing intelligent parking systems [6].

Over the years, different YOLO versions have been developed, each introducing improvements in speed, accuracy, and adaptability. YOLOv5, for example, is recognized for its high precision and low false-positive rates but often exhibits relatively lower recall, limiting its robustness in safety-critical applications [7]. In contrast, the latest version, YOLOv8, has demonstrated superior recall performance and improved adaptability under challenging conditions such as low illumination or cluttered scenes [8]. These strengths highlight YOLOv8's potential for addressing challenges in real-time parking detection, particularly in outdoor environments with occlusions and lighting variability.

Integrating YOLOv8 into intelligent parking systems has shown promising results, achieving real-time detection speeds of up to 45 frames per second (FPS) [9]. Such processing efficiency enhances responsiveness in open parking lots while maintaining reliable differentiation between occupied and empty parking spaces, even under suboptimal conditions. Although combining YOLOv8 with OpenCV offers benefits such as optimized video stream tracking and real-time response [10], challenges remain in maintaining consistent performance under low-light scenarios.

The objective of this study is a real-time system for outdoor parking availability detection using YOLOv8. The choice of YOLOv8 is motivated by its architectural improvements, including the transition to an anchor-free detection approach and the use of CSPDarknet53 as its backbone, which together improve inference speed and robustness [11]. These features enhance the model's ability to detect small and densely packed objects, making it well-suited for edge-based parking applications.

Several prior studies have attempted to tackle parking management using CV and DL approaches. In [12], YOLOv4 was implemented for vehicle detection and achieved 72.8% accuracy; however, the system struggled with object occlusions in crowded scenarios. In [13], a lightweight variant of YOLOv5, termed T-YOLO, was introduced by integrating multi-scale convolution and attention mechanisms. The proposed model achieved 96.3% precision in tiny vehicle detection from a cenital camera perspective, surpassing the baseline YOLOv5 while preserving real-time inference performance at approximately 30 fps. In [14], AlexNet was combined with YOLO for vehicle detection in CCTV footage and reported 93.48% accuracy, but the evaluation was limited to controlled lighting environments. In [15], the HOG technique was used to identify parking spaces, which required precise camera positioning and was difficult to generalize across parking layouts.

From this review, it is evident that prior approaches either struggled with occlusion, lacked robustness in low-light conditions, or required rigid setups that limited generalization. Moreover, limited datasets have constrained model generalization and real-world applicability. This research addresses these gaps by: (1) Applying YOLOv8 for real-time parking space detection in outdoor environments; (2) Enhancing model performance through hyperparameter tuning and data augmentation; and (3) Evaluating the system under varying conditions, including nighttime scenarios with reduced visibility. By doing so, this research demonstrates YOLOv8's advantages in handling occlusion, lighting variability, and small-object detection, while contributing toward the development of efficient and intelligent parking management systems.

This research is organized into four key phases: (1) Preparation, in which relevant data is gathered and arranged; (2) Labeling, in which the parking spaces and the vehicles are labelled using bounding boxes; (3) Training, in which YOLOv8 models are trained, optimized, and tuned through hyperparameters; and (4) Comparison, in which both models are tested against important metrics to assess their application to real-life parking detection situations.

II. METHOD

This study employed an experimental approach to develop and evaluate a real-time parking detection system based on YOLOv8. The model was selected due to its anchor-free architecture, high detection accuracy, and rapid inference speed, which provide better generalization than traditional machine learning methods and earlier YOLO versions. Data were collected through drone-based image capture in outdoor parking areas under varying lighting conditions, followed by labeling and preprocessing. Although the dataset consisted of only 131 annotated images, augmentation techniques such as rotation, inversion, and brightness adjustment were applied to partially mitigate the limitation of dataset size. Model performance was assessed across four experimental configurations using precision, recall, F1-score, mAP@50, and accuracy. The overall workflow, from data acquisition and annotation to training and mobile deployment, is illustrated in Fig. 1.

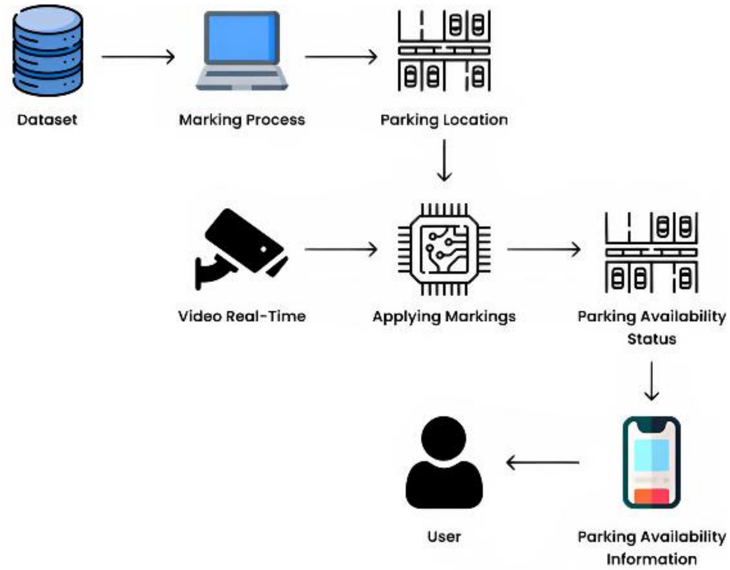

Fig. 1 System workflow of parking availability detection

A. Data Preparation

The research dataset is composed of photographs and videos captured by a drone in external parking lots during the day. Altogether, 131 frames were extracted from the video recordings, as well as combined with aerial shots, all annotated with marker boxes for cars and parking spots. The data set was split into three parts: 70% for training, 20% for validation and 10% for testing of the models. In order to increase the diversity of the dataset and the generalization ability of the model, augmentation methods such as rotation, inversion, and brightness modification were performed [16]. Nevertheless, the dataset size of only 131 images remains a limitation for deep learning applications, as it may restrict the model's ability to generalize. While augmentation helps partially, it cannot fully substitute for the diversity provided by larger real-world datasets.

B. Research Tools and Materials

In developing a parking spaces availability detection system, selection of the appropriate hardware and software is crucial to support the research and testing of the model. Hardware is required for data capturing and model processing while software is used in data processing, labeling, model training, and results analysis. The Table I below shows the hardware and software used in this study and their corresponding specifications.

C. Dataset Labelling

The collected images were labeled with the use of bounding boxes to designate vehicles and parking lots. The labeling step was performed using Roboflow, which is an open-source graphical image annotation tool. Each vehicle is marked with a bounding box and the parking lot is divided into vehicles and empty (like the car in Fig. 2, done in Roboflow). The labeled information is in YOLO format, consisting of the class id with the normalized coordinates of the bounding box.

The systematic approach applied aimed to improve the accuracy and precision achieved with the labeled data. To avoid subjectivity, the labeling of "empty" space was standardized using clear criteria. A parking space was annotated as *empty* only if the entire marked area was completely free of vehicles, while any space with more than one-third of its area covered by a vehicle was labeled as occupied. Ambiguous cases, such as partial obstruction or shadow interference, were consistently classified as occupied to maintain annotation reliability across the dataset. Labeling of all captured images was done meticulously through the Roboflow platform. Vehicles present in the parking lot were labeled with many different bounding boxes, further categorized into Cars and empty parking lots. To improve the reliability of high-quality annotations, there was an additional process where each labeled image was reviewed for accuracy and bounding boxes were changed where any discrepancies were detected. The final annotation dataset is exported in YOLO format for easy use with the object detection deep learning model. Fig. 3 illustrates the distribution of labels in the dataset used to train the object detection model. This bar graph shows the number of instances of the two categories, using blue bars to designate "cars" and light blue to designate "empty". A representative example of the dataset samples used in this study is presented in Fig. 4, which illustrates the diversity of input data after preprocessing.

TABLE I
HARDWARE & SOFTWARE

Hardware	Specification
Laptop	AMD Ryzen 7 4800H / 16 GB / 1TB / Windows 11 Home
Camera CCTV	Resolution 1920x1080 / Frame Rate Max 30 FPS
Smartphone	iOS 18.5 / 4 GB / 128 GB
Drone	Resolution 4K 3840x2160 / Frame Rate Max 30 FPS
Visual Studio Code	1.89.1 Version

Fig. 2 Labeled image (red: cars, purple: empty)

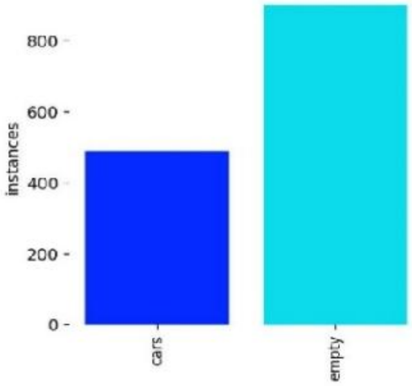

Fig. 3 Data distribution for training

Fig. 4 Dataset sample

In the training process of the object detection model, data preprocessing constitutes a critical step to ensure that the dataset conforms to the requirements of the YOLOv8 algorithm. To improve generalization and robustness in real-world scenarios, the dataset underwent several formatting and augmentation procedures.

1) *Auto Orientation Adjustment*: All images were first automatically aligned based on their EXIF metadata to ensure consistent orientation across the dataset. This step minimizes detection errors caused by image rotation or skewed perspectives.

2) *Image Resizing to 1280x1280*: Each image was resized to a fixed resolution of 1280×1280 pixels using a stretch-resize method. This normalization of input dimensions is necessary to meet the input requirements of YOLOv8 and to optimize computational efficiency during training.

3) *Data Augmentation*: To expand the diversity of the dataset and prevent overfitting, augmentation techniques were applied. For every original image, three additional augmented versions were generated using the following methods: Flipping, Rotation, and Shearing.

D. YOLOv8 Architecture

YOLO which stands for You Only Look Once-is a real-time object detection framework that analyzes the entire image in a single forward process through a convolutional neural network. The latest iteration, Fig. 5 illustrates YOLOv8 architecture [17], is equipped with three detection heads P3, P4, and P5 that enable the algorithm to identify small, medium, and large objects with higher precision.

To evaluate model performance, researchers typically rely on Precision, Recall, and mean Average Precision at an Intersection over Union threshold of 0.5, abbreviated as mAP@50. Central to these measures is the IoU metric, which measures how closely a predicted bounding box aligns with the ground truth box; a higher IoU signifies more accurate detection. Collectively, these indicators serve as key benchmarks for assessing the effectiveness of object detection systems. Model evaluation is performed using common metrics in object detection such as Precision, Recall, F1-Score, and mAP@50, which measure the accuracy and completeness of model predictions, respectively. Details of metric calculations can refer to [18-20]. YOLOv8 features an architecture equipped with three detection heads [21].

E. Confusion Matrix

The evaluation process requires a set of quantitative metrics to assess the performance of the YOLOv8 model in detecting parking availability. In this study, the metrics employed include accuracy, precision, recall, and F1-score [20]. These four measures collectively provide a comprehensive evaluation of the model's classification performance, capturing its ability to correctly identify both occupied and empty parking spaces while maintaining a balance between accuracy and sensitivity.

F. Model Training

The present experiment investigates the capacity of YOLOv8 to identify parking spaces directly from unprocessed camera footage. Training occurred in real-world outdoor settings, allowing the model to learn the subtle visual cues that separate empty bays from those already in use.

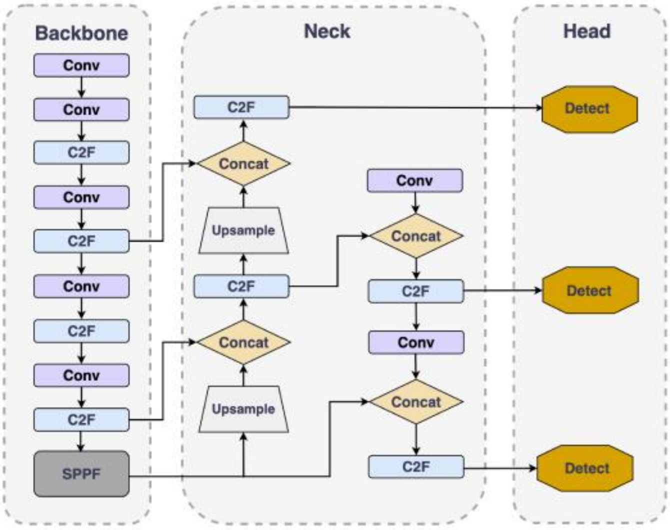

Fig. 5 YOLOv8 architecture applied in this study for real-time parking space detection

This task is challenging due to busy backgrounds and small targets—passing cars that frequently blend with shadows or graphic clutter. To ensure reliable performance, several configurations of hyperparameters were tested throughout the training process. The details of these configurations, including epoch count, image size, batch volume, and optimizer type, are presented in Table II.

Table II summarizes all hyperparameter configurations applied to the four YOLOv8 training sessions. Each session attempted to assess the extent to which parameter tuning and optimizer selection could improve the model's accuracy in recognizing vehicles in high-resolution images.

G. *Comparison of Training Results*

To assess the performance of the YOLOv8 architecture across distinct setups, four separate training experiments were carried out, each introducing different calibrations to hyperparameters and optimizer choice. Experiment 1 used the baseline configuration recommended by the framework: 100 epochs, 1280-pixel input size, batch size of 8, and optimizer selection set to automatic. No manual adjustment of learning rate, momentum, or related parameters was performed. Experiment 2 repeated this setup but swapped the automatic optimizer for the Adam algorithm, which is frequently praised for adapting learning rates on a per-parameter basis and often speeds up convergence while smoothing training dynamics. Experiment 3 kept the prior settings, yet added a tuning phase that lasted 50 epochs and ran for five iterations; the optimizer still reverted to automatic selection during both the tuning and the subsequent full training. Experiment 4 merged the two improvements, starting with the tuning phase under Adam and then continuing for 100 epochs with the same image-size and batch-size parameters defined earlier, creating a compound strategy of learning-rate adaptation and parameter exploration. By systematically comparing the four experimental configurations, the study sought to isolate the effects of Hyperparameter tuning and optimizer selection on model performance as measured by precision, recall, and the Mean Average Precision at IoU equal to 0.5. Results show that the jointly tuned settings paired with the Adam optimizer, labeled Experiment 4, yielded superior scores on every metric.

III. RESULT AND DISCUSSION

A. *Model Training Results using YOLOv8*

In the YOLOv8 model training process, hyperparameter adjustment plays an important role in optimizing the model's performance in detecting objects. The following Table III presents the best values of the hyperparameters used during the YOLOv8 model training process. These values were obtained through a process of experimentation aimed at achieving a balance between convergence speed and optimal detection accuracy.

With this combination of parameter settings, the model achieved good detection performance in accurately detecting the availability of parking spaces in outdoor environments. 38 The YOLOv8 model was trained with several configurations to evaluate the model's performance in detecting the availability of outdoor parking spaces. The configurations used included variations in the number of epochs, the type of optimizer, and training methods with and without hyperparameter tuning. The performance comparison results of the YOLOv8 model configurations can be seen in Table IV, which presents the results from the four experimental scenarios conducted. Each experiment yielded different metric values, reflecting the influence of training configurations on the performance of the YOLOv8 model in detecting parking spaces.

Based on the results shown in Table IV, it can be seen that each model configuration produces different evaluation metric values, namely precision, recall, and mAP@50. Among the four experimental scenarios, Experiment 2 achieved the best overall performance, with the highest recall value of 99.06%, precision of 96.74%, and mAP@50 of 99.17%. These values indicate that the model in Experiment 2 is capable of accurately detecting nearly all correct objects. Thus, it can be concluded that a simple yet effective configuration, such as that in Experiment 2, proved to produce the most accurate and reliable object detection model in this study. Conversely, improper or excessive tuning, such as in Experiment 4, can actually reduce the overall performance of the model.

B. *YOLOv8 Model Performance in Parking Detection*

Fig. 6 presents the car parking spaces detection outcomes produced by the YOLOv8 architecture applied in Experiment 2, the configuration that yielded the highest evaluation scores: Precision 96.74 percent, Recall 99.06 percent, and mAP@50 99.17 percent. Within the graphical output, the entire parking zone is delineated by colour-coded bounding boxes red for occupied space, green for empty ones, and a blue polygon marking the overall region of interest (ROI). Above the image region, a summary label automatically updates, currently reading Used Parking 18/42 and Parking Available 24/42, thus providing an instant snapshot of space occupancy. On the basis of this evidence, the model operates effectively under full daylight, accurately separating filled from empty spots and counting available spaces with both high precision and speed.

Fig. 7 shows how well the YOLOv8 model detects parking spaces at night. This variant, which comes from Experiment 2, records a precision of 96.74 percent, recall of 99.06 percent, and an mAP@50 of 99.17 percent, making it the top setup tested so far. During the nighttime test, the system identified 26 occupied space marked by red boxes and 16 free space marked by green boxes, from an overall total of 42 bays. However, a detailed examination indicates a mismatch between the visual boxes and the numeric label that says "Used Parking 27/42" and "Parking Available 15/42." Such errors probably stem from dark corners that the model missed and from several duplicate or misclassified bounding boxes, both of which skewed the final counts. Even with these setbacks, the models core daytime capability shines through, underscoring the value of more nighttime images and smarter data augmentation to sharpen detection when lighting is poor.

TABLE II
HYPERPARAMETER CONFIGURATION

Experiment to	Hyperparameter Tuning			Hyperparameter Training			
	Epoch	Iteration	Optimizer	Epoch	Imgsz	Batch	Optimizer
1	-	-	-	100	1280	8	Auto
2	-	-	-	100	1280	8	Adam
3	50	5	Auto	100	1280	8	Auto
4	50	5	Adam	100	1280	8	Adam

TABLE III
BEST VALUE PARAMETER RESULTS USED

Parameter	Value	Parameter	Value	Parameter	Value
lr0	0.01	cls	0.5	shear	0.0
lrf	0.01	hsv_h	0.015	perspective	0.0
momentum	0.937	hsv_s	0.7	flipud	0.0
weight_decay	0.0005	hsv_v	0.4	fliplr	0.5
warmup_epoch	3.0	degrees	0.0	mosaic	1.0
warmup_momentum	0.8	translate	0.1	mixup	0.0
box	7.5	scale	0.5	copy_paste	0.0

TABLE IV
PERFORMANCE COMPARISON RESULTS OF YOLOV8 CONFIGURATION MODELS

Experiment to	Epoch	Precision	Recall	mAP@50
1	80	97.24%	98.21%	98.70%
2	80	96.74%	99.06%	99.17%
3	97	97.33%	97.68%	98.87%
4	73	97.34%	97.15%	99.06%

Fig. 6 YOLOv8 detection performance in daytime

Fig. 7 YOLOv8 detection performance at night

C. Confusion Matrix Analysis

The evaluation results of the YOLOv8 model from four training experiments are summarized in Table V, while the confusion matrix of the best-performing model (Experiment 2) is illustrated in Fig. 8.

Table V shows the accuracy obtained from four YOLOv8 training experiments. Experiment 2 achieved the highest accuracy of 77.91%, followed closely by Experiment 3 with 77.74%. However, accuracy alone can be misleading due to class imbalance in the dataset. For instance, the number of samples differs between categories, with the “Cars” class having fewer support values than the “Empty” class in most experiments (Table VI). In such cases, high accuracy may disproportionately reflect the dominant class rather than balanced performance across categories. Therefore, class-specific metrics were analyzed to ensure a more reliable evaluation.

D. Class Specific Model Performance Analysis

Table VI presents per-class performance results, including precision, recall, F1-score, and support values. The results show that the “Empty” class consistently achieved higher precision and F1-scores compared to the “Cars” class across all experiments. For example, in Experiment 3 the “Empty” class reached 95.00% precision and 91.57% F1-score with 129 support values, while the “Cars” class recorded 86.79% precision and 81.78% F1-score with only 119 support values. This difference reflects the relative ease of detecting empty space, which typically have clearer visual boundaries and less occlusion than occupied space.

The fluctuation in the “Cars” class, especially the steep decline in Experiment 4 (60.38% precision, 66.32% F1-score with only 87 support samples), highlights the impact of optimizer misalignment and insufficient class balance. By contrast, Experiments 2 and 3 yielded more stable results across both categories, indicating that moderate hyperparameter tuning improves generalization rather than favoring a single class. Although statistical confidence intervals were not computed in this study, the consistent margin between Experiments 2/3 and the weaker performance of Experiments 1/4 suggests that the observed improvements are meaningful. Future work should incorporate bootstrapped confidence intervals and further dataset balancing to strengthen the reliability of performance comparisons.

Hyperparameter tuning and class balance strongly influenced YOLOv8’s performance. Experiment 3 yielded the most stable results, with Cars reaching 86.79% precision and 81.78% F1-score (119 support samples), and Empty achieving 95.00% precision and 91.57% F1-score (129 support samples). These outcomes indicate that moderate tuning improves generalization across classes. By contrast, Experiment 4 recorded only 66.32% F1-score for Cars with just 87 support samples, underscoring the model’s weakness in handling class imbalance and optimizer misalignment.

The consistently higher scores of the Empty class across experiments can be explained by clearer space boundaries and more support samples compared to the Cars class, which suffered from occlusion and visual clutter. Lighting conditions further amplified these challenges: during nighttime evaluations, the model frequently misclassified or skipped space due to weak edge and texture cues, reaffirming YOLOv8’s vulnerability in low-light environments.

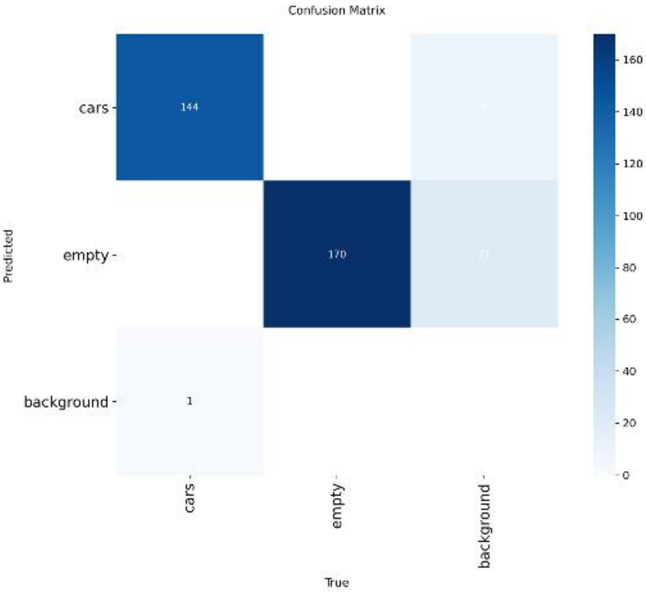

Fig. 8 Confusion matrix of the best YOLOv8 model experiment

TABLE V
COMPARISON OF YOLOV8 MODEL ACCURACY RESULTS

Experiment to	Accuracy (%)
1	75.38%
2	77.91%
3	77.74%
4	66.16%

TABLE VI
SPECIFIC COMPARISON OF YOLOV8 MODEL PERFORMANCE

Experiment to	Class	Precision	Recall	F1-Score	Support Values
1	Cars	79.25%	75.00%	77.06%	112
	Empty	93.33%	94.12%	93.72%	119
2	Cars	77.36%	87.23%	82.00%	94
	Empty	93.33%	91.06%	92.18%	123
3	Cars	86.79%	77.31%	81.78%	119
	Empty	95.00%	88.37%	91.57%	129
4	Cars	60.38%	73.56%	66.32%	87
	Empty	91.67%	87.30%	89.43%	126

Overall, the results show that while YOLOv8 is effective for real-time detection, its robustness depends on a more balanced dataset and richer illumination diversity. Future work should expand data collection, particularly for underrepresented classes such as Cars in cluttered or low-light conditions, and apply augmentation or multimodal inputs to sustain accuracy across diverse scenarios.

IV. CONCLUSION

This study developed a real-time outdoor parking space detection system using YOLOv8, achieving strong results with 96.74% precision, 99.06% recall, 99.17% mAP@50, and 77.91% accuracy. While the model performed well in detecting both empty and occupied spaces, its reliability declined under low-light conditions, and the limited dataset of 131 images constrained generalization, with augmentation providing only partial compensation. Future research should expand data collection, especially for nighttime scenes, and leverage larger benchmark datasets such as PKLot or CNRPark-EXT to enhance robustness and comparability. Despite these limitations, the proposed system demonstrates strong potential for practical deployment in intelligent parking management and urban transportation systems.

ACKNOWLEDGEMENT

The authors would like to thank the Information Technology Study Program, Telkom University Surabaya, for the support and facilities provided during the research process. The author would also like to thank the supervisors for their valuable guidance, input, and encouragement during the process of completing this research. Finally, the author would like to thank the author's family and colleagues for their continuous motivation and moral support so that this research can be carried out.

REFERENCES

[1] A. Kurek and E. Macioszek, “Drivers’ Subjective Assessment of the Ease of Finding a Vacant Parking Space in an Area Equipped with Vehicle Detection Devices,” *Sensors*, vol. 22, no. 18, Sep. 2022, doi: 10.3390/s22186734.

[2] K. Kumar, V. Singh, L. Raja, and S. N. Bhagirath, “A Review of Parking Slot Types and their Detection Techniques for Smart Cities,” *Smart Cities*, vol. 6, no. 5, pp. 2639–2660, Oct. 2023, doi: 10.3390/smartcities6050119.

[3] I. K. Gunawan, I. Putu, A. Bayupati, K. S. Wibawa, I. M. Sukarsa, and L. A. Kurniawan, “Indonesian Plate Number Identification Using YOLACT and Mobilenetv2 in the Parking Management System,” *JUITA: Jurnal Informatika*, vol. 9, no. 1, pp. 69–76, May 2021, doi: 10.30595/juita.

[4] A. Ahad and F. Kidwai, “Detection of vacant parking space in adverse weather Based on Improved YOLO Network Model,” Apr. 2024. doi: 10.21203/rs.3.rs-4175314/v1.

[5] F. Xiao, H. Wang, Y. Li, Y. Cao, X. Lv, and G. Xu, “Object Detection and Recognition Techniques Based on Digital Image Processing and Traditional Machine Learning for Fruit and Vegetable Harvesting Robots: An Overview and Review,” Mar. 01, 2023, *MDPI*. doi: 10.3390/agronomy13030639.

[6] O. G. Ajayi, J. Ashi, and B. Guda, “Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images,” *Smart Agricultural Technology*, vol. 5, Oct. 2023, doi: 10.1016/j.atech.2023.100231.

[7] I. Apeināns, M. Sondors, L. Litavniece, S. Kodors, I. Zarembo, and D. Feldmane, “Cherry Fruitlet Detection using YOLOv5 or YOLOv8,” *Vide. Tehnologija. Resursi - Environment, Technology, Resources*, vol. 2, pp. 29–33, 2024, doi: 10.17770/etr2024vol2.8013.

[8] G. Jocher, S. Waxmann, and K. Verbitski, “YOLOv8: State-of-the-Art Object Detection Model,” Ultralytics Documentation. [Online]. Available:

https://docs.ultralytics.com/guides/hyperparameter-tuning/

[9] Y. Gao, W. Liu, H. C. Chui, and X. Chen, "Large Span Sizes and Irregular Shapes Target Detection Methods Using Variable Convolution-Improved YOLOv8," *Sensors*, vol. 24, no. 8, Apr. 2024, doi: 10.3390/s24082560.

[10] 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," Jul. 2022, doi: 10.48550/arXiv.2207.02696.

[11] E. U. Armin, A. Purnama Edra, F. I. Alifin, I. Sadidan, I. P. Sary, and U. Latifa, "Performa Model YOLOv8 untuk Deteksi Kondisi Mengantuk pada pengendara mobil," *BRAHMANA: Jurnal Penerapan Kecerdasan Buatan*, vol. 5, no. 1, pp. 67–76, Dec. 2023, doi: 10.30645.

[12] G. Novandra Rizkatama, A. Nugroho, and dan Alfa Faridh Suni, "Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir berbasis Python dan YOLO v4," *Edu Komputika*, vol. 8, no. 2, 2021, doi: 10.15294/edukomputika.

[13] D. P. Carrasco, H. A. Rashwan, M. A. Garcia, and D. Puig, "T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks," *IEEE Access*, vol. 11, pp. 22430–22440, Mar. 2023, doi: 10.1109/ACCESS.2021.3137638.

[14] E. Tanuwijaya and C. Fatichah, "Penandaan Otomatis Tempat Parkir Menggunakan YOLO untuk Mendeteksi Ketersediaan Tempat Parkir Mobil pada Video CCTV," *BRILIANT: Jurnal Riset dan Konseptual*, vol. 5, no. 1, 2020, doi: 10.28926/briliant.

[15] T. R. Calista, N. W. A. Majid, and R. Andrian, "Implementasi Image Processing dan Histogram of Oriented Gradient untuk Mendeteksi Slot Parkir Suatu Supermarket," *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 11, no. 3, p. 453, Jul. 2023, doi: 10.26418/justin.v11i3.55412.

[16] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text Data Augmentation for Deep Learning," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00492-0.

[17] G. Yao, S. Zhu, L. Zhang, and M. Qi, "HP-YOLOv8: High-Precision Small Object Detection Algorithm for Remote Sensing Images," *Sensors*, vol. 24, no. 15, Aug. 2024, doi: 10.3390/s24154858.

[18] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Comput Appl*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, doi: 10.1007/s00521-023-08809-1.

[19] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification," *BMC Med Inform Decis Mak*, vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12911-021-01691-8.

[20] A. I. Tanggraeni and M. N. N. Sitokdana, "Analisis Sentimen Aplikasi E-Government Pada Google Play Menggunakan Algoritma Naïve Bayes," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 2, pp. 785–795, Jun. 2022, doi: 10.35957/jatisi.v12i3.

[21] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach Learn Knowl Extr*, vol. 5, no. 4, pp. 1680–1716, Dec. 2023, doi: 10.3390/make5040083.