JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6236

COMPARATIVE STUDY OF YOLO VERSIONS FOR DETECTING VACANT CAR PARKING SPACES

Muhammad Fathurrahman[1]; Anan Nugroho[2*]; Ahmad Zein Al Wafi[3];

Electrical Engineering Department [1,2,3]
Universitas Negeri Semarang, Semarang, Indonesia[1,2,3]
https://unnes.ac.id/[1,2,3]
071203fathur@students.unnes.ac.id[1], anannugroho@mail.unnes.ac.id[2*], ahmadzeinalwafi@outlook.com[3]

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *The increasing vehicle density in urban areas has made parking space availability a significant challenge. With technological advancements, efficient smart parking systems based on object detection have become essential. This study evaluates the performance of YOLO versions 3 to 11 in detecting vacant parking spaces in urban environments, focusing on real-time processing, high accuracy with limited datasets, and adaptability to varying conditions. Using 4,215 annotated images and two test videos, YOLOv7 achieved the highest overall accuracy of 99.57% with an average FPS of 30.79, making it the most effective model for smart parking applications. YOLOv8 and YOLOv11 followed closely, with accuracies of 98.51% and 98.72%, respectively, and average FPS rates of 32.31 and 31.99, balancing precision and speed, which are ideal for real-time applications. Meanwhile, YOLOv5 stood out for its exceptional processing speed of 33.92 FPS. These results highlight YOLO's potential to revolutionize smart parking systems by significantly enhancing both detection precision and operational efficiency.*

Keywords*: YOLO, car detection, parking spaces, smart parking, urban mobility.*

Intisari— *Meningkatnya kepadatan kendaraan di daerah perkotaan telah menjadikan ketersediaan ruang parkir sebagai tantangan yang signifikan. Dengan kemajuan teknologi, sistem parkir cerdas yang efisien berbasis deteksi objek menjadi sangat penting. Penelitian ini mengevaluasi performa YOLO versi 3 hingga 11 dalam mendeteksi ruang parkir kosong di lingkungan perkotaan, dengan fokus pada pemrosesan waktu nyata, akurasi tinggi meskipun menggunakan dataset terbatas, serta kemampuan adaptasi terhadap berbagai kondisi. Menggunakan 4.215 gambar yang telah diberi anotasi dan dua video uji, YOLOv7 mencapai akurasi keseluruhan tertinggi sebesar 99,57% dengan FPS rata-rata 30,79, menjadikannya model yang paling efektif untuk aplikasi parkir cerdas. YOLOv8 dan YOLOv11 mengikuti dengan akurasi masing-masing sebesar 98,51% dan 98,72%, serta FPS rata-rata 32,31 dan 31,99, menyeimbangkan presisi dan kecepatan yang ideal untuk aplikasi real-time. Sementara itu, YOLOv5 menonjol dengan kecepatan pemrosesan yang luar biasa sebesar 33,92 FPS. Hasil ini menunjukkan potensi YOLO untuk merevolusi sistem parkir cerdas dengan secara signifikan meningkatkan presisi deteksi dan efisiensi operasional.*

Kata Kunci*: YOLO, deteksi mobil, tempat parkir, parkir cerdas, mobilitas perkotaan.*

INTRODUCTION

Parking has become a critical issue in urban areas due to the continuous rise in car ownership [1]. This trend poses new challenges for urban space management, particularly in terms of parking availability [2]. As the number of vehicles increases, there is an urgent need for more efficient and innovative parking management systems [3]. With the advancement of remote monitoring technology, including smart parking systems, it has become increasingly important to provide users with efficient and practical ways to locate available parking spaces[4]. Thus, implementing sensor-

based smart parking systems has emerged as a viable solution[5]. However, the use of sensors currently incurs high costs, energy consumption, and maintenance, especially when implemented on a large scale [6]. While sensor-based solutions, such as infrared or ultrasonic detectors, offer a partial solution, they are prone to inaccuracies, particularly when objects other than cars pass through the sensors [7]. Additionally, traditional image processing methods, while alternative solutions, still suffer from low accuracy, unreliability in scenarios involving complex object colors, and limited flexibility due to their dependence on manually set parameters and thresholds [8].

Given the limitations of traditional methods, deep learning-based computer vision approaches have emerged as a promising alternative. This technology offers fast and accurate detection capabilities, making it highly suitable for implementation using existing CCTV systems in parking areas, thereby simplifying the monitoring process [9]. Furthermore, deep learning enables automatic feature extraction, learns complex patterns, and adapts seamlessly to diverse datasets. Therefore, object detection for vehicle identification in parking areas has become an essential component of modern smart parking systems. Previous studies on vehicle parking detection have utilized various approaches. For instance, a study employing the Deep Extreme Learning Machine (DELM) algorithm trained on a dataset of 21,431 samples achieved a training accuracy of 94.37% [10]. Other studies reported 98% accuracy in vehicle classification using a ResNet-50-based deep learning model trained using CNN-based models with a dataset of 10,440 images across 13 vehicle classes [11]. Another study achieved an accuracy of 95% utilizing the *PKLot* dataset, consisting of more than 12,000 images with 3 cameras, to develop a multi-angle parking detection system using a modified MobileNetV3 Model [12]. Additionally, the application of You Only Look Once version 3 (YOLOv3) with a custom dataset of 4,900 images demonstrated an accuracy of 94.7% in detecting vacant parking spaces [13].

Several studies have identified YOLO as a strong candidate for smart parking systems due to its speed and accuracy. While YOLO has shown significant promise in object detection, there is limited research comparing the performance of different YOLO versions for parking space detection. However, with the continuous development of YOLO across multiple versions, there remains a lack of clarity on how technological changes influence detection performance in parking slot scenarios. This study aims to fill this gap by systematically evaluating the performance of YOLO versions 3 through 11 in detecting vacant parking spots from different perspectives. By analyzing accuracy and processing speed across various scenarios, the study provides a comprehensive assessment of YOLO's evolution in this specific application.

Among object detection approaches, YOLO achieves high accuracy even with smaller datasets, compared to other methods such as CNN, LSTM, AlexNet, and VGG-6 [14]. The YOLO architecture processes images in a single forward pass through the neural network, contributing to its speed and real-time capability [15]. YOLO can detect objects at speeds of up to 100 FPS and has a smaller file size[16]. With a limited training dataset, this method is effective, performing well in detection speed, making it a more efficient choice for real-time applications [17]. This efficiency is particularly valuable in scenarios where large datasets are unavailable or when rapid model development is required [18]. Specifically, YOLOv3 to YOLOv11 are equipped with Multi-Scale Object Detection capabilities, enabling the detection of objects of different sizes within a single inference process. Additionally, YOLO has efficient reasoning capabilities for various types of direct input, including single images, multiple images, videos, and external camera input[17].

The primary research gap addressed by this study lies in the limited comparative analyses of different YOLO versions for parking slot detection. While prior studies have demonstrated YOLO's effectiveness for general object detection [19], [20], [21], [22], few have systematically examined its version-specific performance in real-world parking environments. The unpredictability of how different YOLO architectures perform under varying conditions, such as long-range and close-range perspectives, necessitates a structured investigation. This study aims to explore the strengths and weaknesses of each YOLO version and provides empirical evidence on their suitability for parking slot detection, considering factors such as detection accuracy and real-time processing capabilities. The findings of this study contribute to the advancement of computer vision technologies by offering a structured evaluation of model selection strategies for smart parking applications. By systematically analyzing the impact of different YOLO architectures on parking slot detection, this research provides a foundation for optimizing model deployment based on specific implementation requirements. These insights are expected to benefit both the research community and industry practitioners by facilitating informed decision-making regarding the trade-offs between

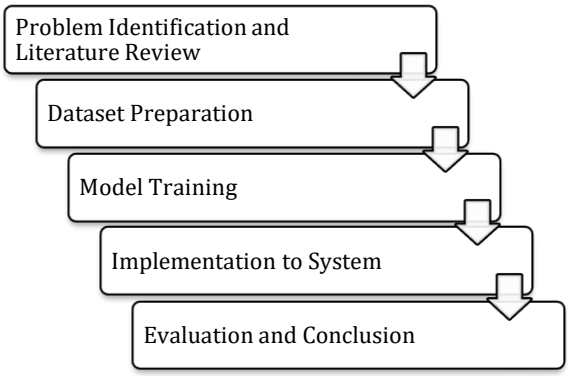
Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6236

detection accuracy, computational efficiency, and real-time applicability. Furthermore, the study serves as a reference for future advancements in intelligent parking systems, supporting the development of more efficient and scalable solutions in urban environments.

MATERIALS AND METHODS

This study was conducted in several stages, as illustrated in Figure 1.



Source : (Research Results, 2024)
Figure 1. Research stages

The process illustrated in Figure 1 began with problem identification and a literature review to understand parking issues, existing detection methods, and YOLO-based approaches. A comprehensive dataset of vehicle images was then prepared to ensure diverse and representative training data, simulating real-world parking conditions for model training and evaluation. YOLO versions 3 to 11 were trained using identical hyperparameters to ensure a fair performance comparison. After training, all YOLO versions were trained with identical hyperparameters to ensure a consistent performance comparison. The detection results were analyzed based on accuracy and frames per second (FPS) to assess each model's efficiency in identifying vacant parking spaces. Finally, the performance of different YOLO versions was compared to determine the most effective model for real-time parking space detection.

YOLO Architecture Advancements

YOLO, or You Only Look Once, is a groundbreaking object detection framework introduced by Redmon *et al*. in 2016, which revolutionized the field with its single-stage architecture, enabling real-time performance while maintaining high accuracy. Unlike traditional multi-stage detectors that separate region proposal and classification, YOLO reframes object detection as a single regression problem, predicting bounding boxes and class probabilities directly from full images in one way. Over time, YOLO's mechanism has evolved significantly, driven by the need to balance computational efficiency, accuracy, and adaptability to diverse datasets.

YOLO's evolution from YOLOv3 to YOLOv11 reflects significant advancements in object detection mechanisms, with each version introducing architectural and methodological innovations to improve performance and efficiency. YOLOv3, released in 2018, introduced key concepts such as the Spatial Pyramid Pooling (SPP) block and the Darknet-53 backbone, which allowed the model to extract richer features for better performance in detecting objects at multiple scales [23]. YOLOv4, released in 2020, further improved the model by adding the Mish activation function and using the CSPDarknet-53 backbone, which provided better feature reuse [24], [25]. That same year, YOLOv5 was introduced, which adopted anchor-free detection and the SWISH activation function, along with PANet, which optimized the feature aggregation process for more accurate detections [26], [27].

In 2022, the development of YOLO models took a major leap forward with YOLOv6 and YOLOv7. YOLOv6 introduced self-attention mechanisms and continued the trend of anchor-free object detection, allowing for more flexible and efficient object localization [18][28]. YOLOv7, on the other hand, incorporated transformers and the Extended-Efficient Layer Aggregation Network (E-ELAN) reparameterization technique, further boosting the model's ability to handle complex detection tasks by improving the flow of information across the network [29], [30]. In 2023, YOLOv8 continued this trend by introducing Generative Adversarial Networks (GANs), which improved the quality of detections while maintaining its anchor-free detection approach [31], [32].

The advancements continued in 2024 with YOLOv9, YOLOv10, and YOLOv11. YOLOv9 brought the Proportional Gradient Inference (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) techniques, which improved model efficiency and robustness during inference [33], [34]. YOLOv10 introduced a novel approach to Non-Maximum Suppression (NMS)-free training by using consistent dual assignments, which help in making the model faster [35], [36]. Finally, YOLOv11 introduces new components such as the Spatial Pyramid Pooling Fast (SPFF) and the Cross Stage Partial with Spatial Attention (C2PSA) block, which further enhances the model's ability to capture

complex patterns and improve its computational efficiency[15], [37]. These ongoing advancements reflect the continuous evolution of YOLO as one of the most powerful frameworks in the field of object detection, as summarized in the iterations of YOLO presented in Table 1.

Table 1. YOLO: Evolution of models

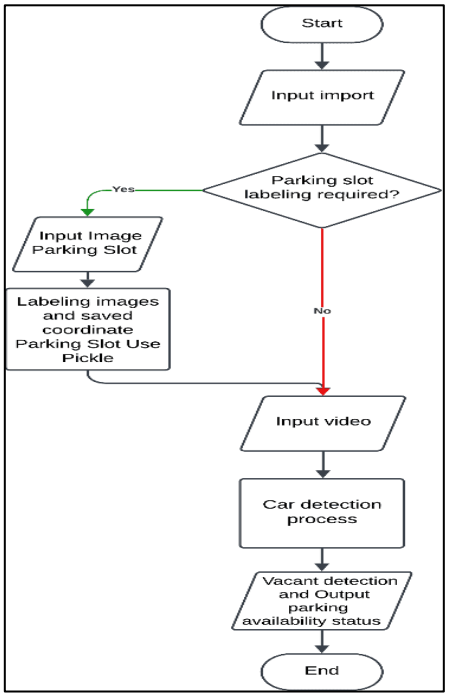
Release	Year	Contribution	Framework
YOLOv3	2018	SPP Block, Darknet-53 Backbone	Darknet
YOLOv4	2020	Mish activation, CSPDarknet-53 backbone	Darknet
YOLOv5	2020	Anchor-free detection, SWISH activation, PANet	Pytorch
YOLOv6	2022	Self-attention, anchor-free OD	Pytorch
YOLOv7	2022	Transformers, E-ELAN reparameterisation	Pytorch
YOLOv8	2023	GANs, anchor-free detection	
YOLOv9	2024	PGI and GELAN	Pytorch
YOLOv10	2024	Consistent dual assignments for NMS-free training	Pytorch
YOLOv11	2024	Spatial Pyramid Pooling Fast (SPFF), C2PSA Block (Cross Stage Partial with Spatial Attention):	Pytorch

Source : (Research Results, 2024)

Table 1 highlights the contributions, frameworks, release years, and features of each YOLO version. This comparison underscores the architectural modifications introduced in each iteration, which result in variations in accuracy and efficiency. These differences are particularly significant in the context of this study, which focuses on detecting vacant parking spaces. Furthermore, the selection of YOLOv3 to YOLOv11 in this study is based on the availability of these versions as open-source models on the official documentation site https://docs.ultralytics.com/ [38]. Additionally, the exclusion of YOLOv1 and YOLOv2 is due to the absence of Multi-Scale Object Detection in YOLOv1 and the limited capabilities of Multi-Scale Object Detection in YOLOv2 [18]. These features are critical for this study, which requires real-time detection that is effective in both close and distant conditions.

System Design

The workflow of the parking availability detection system design used in this study is presented in Figure 2.



Source : (Research Results, 2024)

Figure 2. Design system workflow

Figure 2 illustrates the various steps involved in detecting parking space availability using a camera-based system.

1. Library Import

At this stage, several libraries are imported to facilitate the detection process. The Open-Source Computer Vision (OpenCV) library is utilized to process images and videos [39], while the Pickle library enables the storage of predefined parking slot data [40]. Additionally, the Torch library is employed to access the model via CUDA for GPU-based computations [41].

2. Input Image Parking Slot

At this stage, screenshots from the parking lot video footage, representing the parking area conditions, will be included. The images are captured with the same resolution as the video and saved in PNG file format, with a total of two images.

3. Labeling images and saved coordinates Parking Slots

In this stage, we manually annotated each parking slot using OpenCV, and the coordinates were stored in a Pickle file for future use in the detection process, as illustrated in Figure 3.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6236

Source : (Research Results, 2024)
Figure 3. Illustration of parking coordinate storage

Figure 3 illustrates the manual labeling process, where parking slots are delineated with dimensions of 15 x 30 pixels. The coordinates of these labeled parking slots are stored for subsequent use in the detection process. Steps 2 and 3 are performed only when necessary; otherwise, the process proceeds directly to Step 4.

4. Input Video

In this stage, the video data used to test the YOLO versions is introduced. The input consists of two video clips, one lasting 45 seconds and the other 60 seconds. Each parking scenario is captured in two separate videos, representing close-range and long-range perspectives, both saved in .mp4 format.

5. Car detection process

In this stage, YOLO detects cars captured in the input video. The process begins by feeding the video into the model, which then processes each frame video. YOLO utilizes a backbone network to extract fundamental features from the image, such as shapes and textures, which are crucial for object detection. These features are subsequently processed in the neck through a feature pyramid network to detect objects across multiple scales. In the final stage, YOLO's head generates predictions in the form of bounding boxes and confidence scores for each detected car, as shown in Figure 4.

Source : (Research Results, 2024)
Figure 4. Illustration of car detection results

By dividing the image into a grid, as depicted in Figure 4, YOLO can directly predict the location and confidence level of each car object in every frame. This method accurately produces bounding boxes to mark the position of each detected car.

6. Vacant detection and Output parking availability status

In this stage, vehicle detection results from YOLO are integrated with pre-labeled parking slot data, which were labeled using OpenCV and stored with Pickle. The integration results are visualized in Figure 5.

Source : (Research Results, 2024)
Figure 5. Illustration of Detecting an Empty Car Parking Space

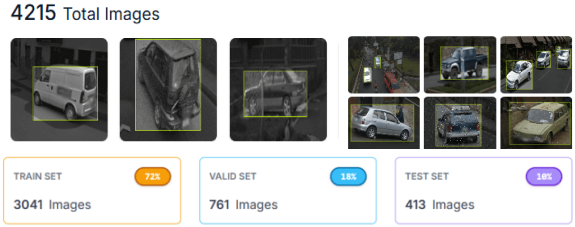
As shown in Figure 5, the parking space detection system identifies the status of parking slots. A parking slot is marked as occupied if a vehicle is detected within the pre-labeled area by YOLO, and it is highlighted in red. Conversely, if no vehicle is detected by YOLO within the labeled area, the space is considered vacant and displayed in green. This output effectively distinguishes between occupied and available parking spaces. Additionally, this stage includes calculating the video processing speed during YOLO model execution, contributing to real-time responsiveness.

Dataset

This study utilizes a dataset comprising 4,215 images of private cars captured under various conditions, including differences in color, angle, type, and shape. The dataset is designed to replicate real-world parking scenarios, ensuring the model's robustness across diverse environments. The variations in the dataset include different car colors and types to encompass all possible vehicle variations entering the parking area, varying distances and angles to ensure accurate detection from both close and far perspectives, enabling the model to function effectively with different CCTV angles, and different lighting conditions (sunny and cloudy) to enhance the model's adaptability to changing weather conditions. These conditions were selected because they are ideal for parking situations, particularly during working hours from morning to afternoon. The dataset is systematically partitioned into three subsets: a training set comprising 3,041 images, a validation set with 761 images, and a test set containing 413 images. Data collection and annotation were facilitated by

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6236
JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

Roboflow, a platform designed to streamline the labeling and management of datasets [42]. Each car object in the images was precisely annotated with bounding boxes, serving as critical reference points for the model to identify regions containing car objects [42]. For model training, the dataset is configured with a batch size of 25, enabling the simultaneous processing of multiple images to expedite the learning process. The training is conducted over 130 epochs. These settings are tailored to produce a robust *.pt* model for YOLOv5-YOLOv11 (Pytorch) and *.weight* for YOLOv3-YOLOv4 (Darknet), optimized for vehicle detection within the system. Some examples of the dataset are illustrated in Figure 6.



Source : (Research Results, 2024)
Figure 6. Collection of research datasets

Experimental Setup

The experimental setup for this study integrates both software and hardware components to evaluate the performance of various YOLO versions in detecting the availability of empty parking spaces. The configuration includes the following:

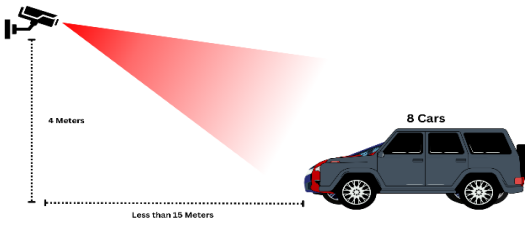
1. ASUS TUF GAMING A15 Laptop with the following specifications:
   a. Processor: AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz
   b. Graphics: NVIDIA® GeForce RTX™ 3060 Laptop GPU, 1752MHz* at 140W (1702MHz Boost Clock+50MHz OC,115W+25W Dynamic Boost)
   c. RAM: 16 GB
   d. Storage: 512 GB
   e. OS: Windows 10 Home
2. Camera CCTV Hikvision 1080P DS-2CD1021-I 2MP
3. 15-meter long USB 2.0 Male to Female cable
4. Software used:
   a. Visual Studio Code
   b. Google Colab Colaboratory
   c. Python 3.11.9
   d. OpenCV 4.10.0
   e. Torch 2.1.0
   f. Pickle Library
   g. CUDA 11.8

The study involved testing each YOLO model on the ASUS TUF GAMING A15, utilizing its GPU, to analyze performance using two parking lot surveillance videos. These videos represented different camera angles, specifically near and far perspectives, to assess the ability of each YOLO version to detect vehicles at varying distances:

1. Video 1: A 60-second simulation video providing an overview of a parking area with 8 parking slots. The footage was recorded using a distant camera positioned at a height of 0.5 meters and less than 15 meters away from the parking site, offering a broad perspective of the area.
2. Video 2: A 45-second video presenting a close-up view of the parking area, covering 5 parking slots. This footage was captured from a distance of less than 4 meters and a height of 0.5 meters, providing a detailed perspective of individual parking spaces.

The video data utilized in this study was captured using a Hikvision 1080P CCTV camera designed for outdoor use with a color video composition. To optimize computational efficiency, the videos were resized to a resolution of 480p. Two different parking lot conditions were recorded for this study. Video 1 was captured from a camera positioned approximately 15 meters away from the parking area, with a duration of 60 seconds. Video 2 was recorded at a closer range, from a camera placed less than 4 meters away, with a duration of 45 seconds. Both videos were saved in the (.mp4) format for consistency in processing. Illustrations of the video capture setup and camera distance are presented in Figures 7, 8, 9, and 10.

Source : (Research Results, 2024)
Figure 7. Illustration of video 1 capture

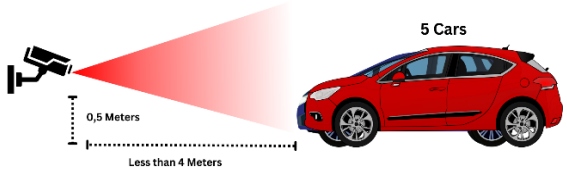


Source : (Research Results, 2024)
Figure 8. Illustration of video 1 camera distance

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6236

Source : (Research Results, 2024)
Figure 9. Illustration of video 2 capture



Source : (Research Results, 2024)
Figure 10. Illustration of video 2 camera distance

In this study, the videos presented in Figures 7, 8, 9, and 10 were utilized to evaluate parking availability detection. The analysis was conducted on two distinct video recordings, each representing unique camera perspectives and parking slot configurations. Both videos capture diverse attributes of parked vehicles, including variations in color, brand, and condition, providing a comprehensive dataset for detection analysis.

Performance Metric

To evaluate the performance of each YOLO version in this study, two primary metrics were utilized: accuracy and efficiency, with the latter assessed through the detection speed measured in Frames Per Second (FPS). Accuracy is the key performance metric, quantifying the model's ability to detect parking slot availability. This metric is calculated based on the proportions of *True Positive* (TP*), True Negative* (TN), *False Positive* (FP), and *False Negative* (FN), as detailed in Table 2.

Table 2. Measurement Scenario

Measurement	Quantity
True Positive (TP):	A parking slot is correctly detected as available.
True Negative (TN):	A parking slot is correctly detected as occupied.
False Positive (FP):	A parking slot is incorrectly detected as available, but it is actually occupied.
False Negative (FN):	A parking slot is incorrectly detected as occupied, but it is actually available.

Source : (Research Results, 2024)

The accuracy metric provides a comprehensive understanding of YOLO's performance in detecting parking slot availability by considering both correct and incorrect detections. Accuracy is calculated using the formula shown in Equation (1):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

In addition to accuracy, the detection speed of each YOLO version is assessed using the FPS metric. FPS is calculated using Equation (2):

$$FPS = \frac{N}{T} \quad (2)$$

where:
N is the total number of frames processed in a detection session.
T is the total time required to process N frames, measured in seconds

The selection of Accuracy and FPS as the primary metrics reflects their relevance in evaluating the detection system's performance in real-world, real-time environments. Accuracy assesses the model's precision in classifying parking slots as available or occupied, which is crucial for the system's reliability. A higher accuracy indicates the model's capability to minimize false positives and false negatives, thereby enhancing user trust in the parking availability information provided [43]. On the other hand, FPS measures the model's processing speed, a critical factor for real-time applications [44]. High FPS ensures the system can promptly respond to changes in parking slot occupancy, offering users up-to-date information and facilitating efficient decision-making in dynamic parking management scenarios. The combination of these metrics provides a holistic evaluation of the model's performance, making it well-suited for deployment in practical environments requiring both accuracy and responsiveness.

RESULTS AND DISCUSSION

Despite this adjustment, the study achieved satisfactory detection results. The study videos were recorded during the daytime in an outdoor setting to represent real-world parking conditions during busy working hours, with shaded and sunny lighting conditions.

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6236

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

Video Detection Results

The detection results are illustrated in Figures 11, 12, 13, and 14, highlighting the system's performance under varying conditions.

Source : (Research Results, 2024)

Figure 11. Input Video 1

Source : (Research Results, 2024)

Figure 12. Output Detection Video 1

Source : (Research Results, 2024)

Figure 13. Input Video 2

Source : (Research Results, 2024)

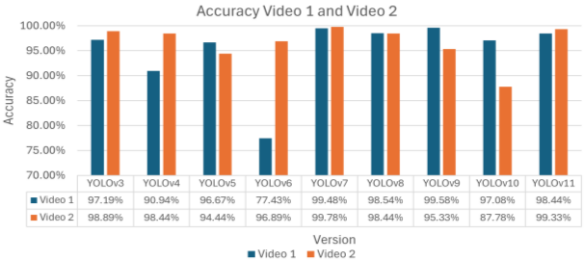
Figure 14. Output Detection Video 2

Figures 11, 12, 13, and 14 illustrate the visual input and output of Video 1 and Video 2, showcasing the scenes before and after detection using the YOLO method. The detection process successfully identifies vehicles entering the parking area, marked by bounding boxes that indicate objects classified as cars. Each bounding box is accompanied by the class label "0," representing cars, a confidence score, and the parking slot's availability status. The parking slot's status, represented by a colored indicator (pickle), changes dynamically: it turns red when a bounding box corresponding to a detected car overlaps with a parking slot, indicating occupancy. Conversely, it remains green when no bounding box is detected within the parking slot, signifying availability.

In Video 1, which captures the parking area from a long-range perspective, the system successfully detects eight parking slots. Conversely, in Video 2, recorded from a closer range, the system identifies five parking slots. While certain versions of YOLO are capable of detecting vehicles positioned behind the target area, this study specifically focuses on the five front parking slots that are clearly visible to the camera and unobstructed by significant obstacles. Additionally, the system calculates and displays the FPS on the top-left corner of the video in real-time, providing an ongoing measure of detection speed throughout the analysis. This FPS value is instrumental in evaluating the performance of each YOLO version during the detection process.

Accuracy Result of Video 1 and Video 2

The detection results from Videos 1 and 2, processed using different YOLO versions, were stored and subjected to slicing at a rate of two frames per second. This process yielded a total of 120 frames for Video 1 and 90 frames for Video 2. We calculated the accuracy of each YOLO version using the frames processed during detection, which were calculated based on these frames. The accuracy results for Video 1 across all YOLO versions are presented in Figure 15.



Source : (Research Results, 2024)

Figure 15. YOLO accuracy results on Video 1 and Video 2

Figure 15 illustrates the detection accuracy of various YOLO versions on Video 1, representing a distant camera's scenario detecting vacant parking spaces. Several versions of YOLO, including YOLOv3, YOLOv5, YOLOv7, YOLOv8, YOLOv9, YOLOv10, and YOLOv11, demonstrate high accuracy, exceeding 95%, with YOLOv9 achieving the highest accuracy of 99.58%. Notably, YOLOv7 also performs exceptionally well, reaching 99.48%, followed closely by YOLOv8 at 98.54% and YOLOv11 at

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

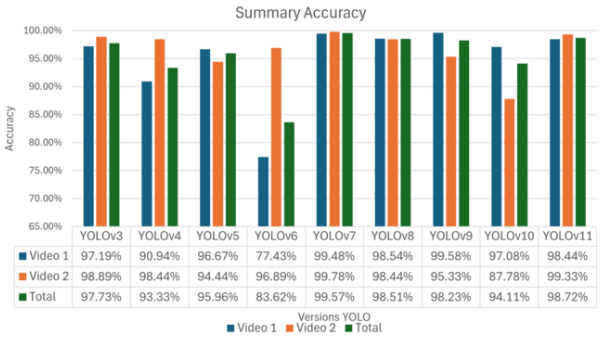
VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6236

98.44%. Although YOLOv4 achieves a lower accuracy of 90.94% compared to other versions, it still maintains reliable performance. On the other hand, YOLOv6 shows significantly lower accuracy at 77.43%, making it the least effective among the evaluated versions in this context. The results indicate that while most YOLO versions are highly effective in detecting vacant parking spaces from a distance, certain versions like YOLOv6 may require further optimization or adjustments to enhance their performance in this specific use case.

Also in Figure 15 illustrates the accuracy results of various YOLO versions on Video 2, captured at a distance of less than 4 meters, showcasing performance variations across different versions. YOLOv7 achieved the highest accuracy at 99.78%, followed by YOLOv11 with an accuracy of 99.33%. Other versions, such as YOLOv3, YOLOv4, and YOLOv8, also demonstrated strong performance with accuracies of 98.89%, 98.44%, and 98.44%, respectively. Meanwhile, YOLOv6 delivered a solid accuracy of 96.89%, followed by YOLOv9 with 95.33% and YOLOv5 with 94.44%. However, YOLOv10 exhibited the lowest accuracy on Video 2, at 87.78%, indicating that this version may be less optimized for close-range detection scenarios. These findings highlight that most YOLO versions perform exceptionally well in detecting parking availability at close range, with YOLOv7 and YOLOv11 leading in performance.

Accuracy Comparison Results

To evaluate the performance of each YOLO version in detecting parking availability, a summary comparison of accuracy results is presented. The comparison includes detection accuracy for Video 1 and Video 2, as shown in Figure 16.



	YOLOv3	YOLOv4	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
Video 1	97.19%	90.94%	96.67%	77.43%	99.48%	98.54%	99.58%	97.08%	98.44%
Video 2	98.89%	98.44%	94.44%	96.89%	99.78%	98.44%	95.33%	87.78%	99.33%
Total	97.73%	93.33%	95.96%	83.62%	99.57%	98.51%	98.23%	94.11%	98.72%

Source : (Research Results, 2024)

Figure 16. Summary YOLO accuracy results

Figure 16 shows the comparison of YOLO variants in Video 1, representing long-distance detection, and Video 2, representing short-distance detection, reveals distinct performance patterns for each model. YOLOv3 demonstrates balanced accuracy between both scenarios, achieving 97.19% for Video 1 and 98.89% for Video 2, highlighting its robustness in handling varying object distances. Similarly, YOLOv4 shows moderate performance, with 90.94% accuracy for Video 1 and a slightly improved 98.44% for Video 2, indicating its stronger capability in short-distance detection. YOLOv5 performs consistently well, maintaining high accuracy for both scenarios, though it shows a slight drop in Video 2 (94.44%) compared to Video 1 (96.67%), suggesting minor sensitivity to object proximity. Interestingly, YOLOv6 exhibits a significant improvement in short-distance detection, achieving 96.89% accuracy for Video 2 compared to 77.43% for Video 1, indicating that it is better optimized for closer object detection. On the other hand, YOLOv7 outperforms all other variants, achieving the highest accuracy in both scenarios, with 99.48% for Video 1 and 99.78% for Video 2, showcasing its exceptional adaptability and efficiency in detecting objects across varying distances. Similarly, YOLOv8 delivers strong and consistent results, achieving 98.54% accuracy for Video 1 and 98.44% for Video 2, highlighting its versatility in diverse detection contexts. YOLOv9 performs exceptionally well for Video 1, achieving the highest accuracy in the long-distance scenario at 99.58%. However, its performance slightly declines in Video 2, with an accuracy of 95.33%, indicating potential challenges in short-distance detection. In contrast, YOLOv10 shows a notable drop in accuracy for Video 2 (87.78%) compared to Video 1 (97.08%), suggesting its limitations in handling short-distance scenarios. Finally, YOLOv11 demonstrates excellent performance in both scenarios, achieving 98.44% accuracy for Video 1 and 99.33% for Video 2, reflecting its reliability and balanced detection capabilities across different distances.

Considering the combined accuracy of Video 1 and Video 2, YOLOv7 stands out as the most effective model, achieving the highest overall accuracy of 99.57%. Following closely, YOLOv11 achieves 98.72%, and YOLOv8 delivers 98.51%, both showing exceptional performance and adaptability across varying distances. YOLOv3 also performs admirably, with a combined accuracy of 97.73%, demonstrating its robustness despite being one of the older variants. YOLOv5 achieves a respectable overall accuracy of 95.96%, maintaining consistent performance across both scenarios. On the other hand, YOLOv4 achieves 93.33%, indicating reliable but not exceptional performance. YOLOv9, despite achieving the highest accuracy for Video 1, records a combined accuracy of 98.23%, slightly lower than YOLOv7, YOLOv8,

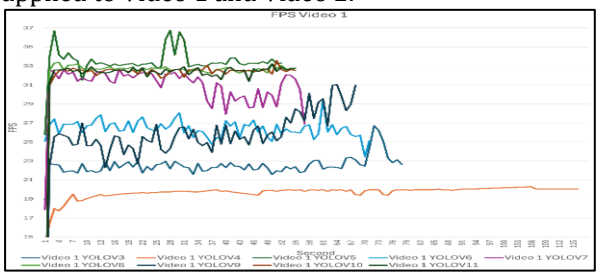
VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6236

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

and YOLOv11 due to its decline in accuracy for Video 2. YOLOv10 shows a combined accuracy of 94.11%, further indicating its challenges in short-distance detection. Lastly, YOLOv6 records the lowest overall accuracy of 83.62%, reflecting its significant performance gap compared to the other variants, particularly in long-distance detection. This phenomenon can be attributed to several factors, including the presence of other objects mistakenly detected as vehicles, cars obstructed by other objects, or limitations in the specific YOLO version's ability to accurately detect objects from a distance. These results highlight the strengths and limitations of each YOLO variant, emphasizing that while advanced versions generally excel, the choice of model should align with the specific requirements of the detection context.

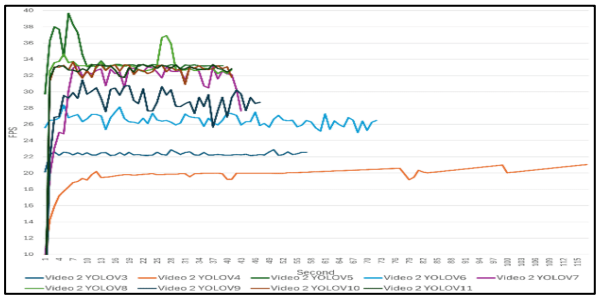
FPS Stability Result Video Detection

Alongside detection accuracy, the FPS achieved by each YOLO version varies significantly. Figures 17 and 18 illustrate the FPS detection performance trends for each YOLO version when applied to Video 1 and Video 2.



Source : (Research Results, 2024)

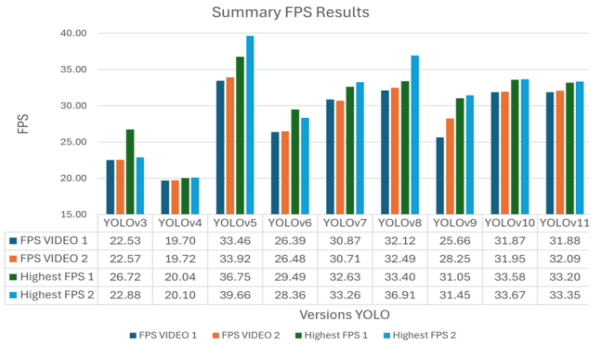
Figure 17. YOLO FPS results on Video 1



Source : (Research Results, 2024)

Figure 18. YOLO FPS results on Video 2

Based on Figures 17 and 18 overall, YOLOv4 demonstrates the most stable FPS performance, maintaining consistent detection rates despite its relatively lower average FPS compared to other versions. This suggests that while YOLOv4 may not be the fastest model, its stability in frame processing could make it suitable for applications where consistent performance is prioritized over raw speed. Other versions, such as YOLOv3, YOLOv10, and YOLOv11, also show stable FPS with relatively high averages, indicating their ability to balance detection speed with consistency. These versions are well-suited for scenarios requiring a combination of speed and reliability. In contrast, YOLOv6, YOLOv7, and YOLOv9 exhibit more fluctuating FPS trends, as reflected in their detection graphs. Although these fluctuations are not excessively significant, they highlight occasional inconsistencies in frame processing, which could impact applications demanding highly stable real-time performance. YOLOv5, which achieves the highest overall FPS among all versions, stands out for its exceptional speed. However, its detection graph reveals occasional instability during certain seconds of video playback, indicating variability in its processing rates. The summarized FPS results are illustrated in Figure 19, providing a comparative overview of the stability and speed of each YOLO version.



	YOLOv3	YOLOv4	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
FPS VIDEO 1	22.53	19.70	33.46	26.39	30.87	32.12	25.66	31.87	31.88
FPS VIDEO 2	22.57	19.72	33.92	26.48	30.71	32.49	28.25	31.95	32.09
Highest FPS 1	26.72	20.04	36.75	29.49	32.63	33.40	31.05	33.58	33.20
Highest FPS 2	22.88	20.10	39.66	28.36	33.26	36.91	31.45	33.67	33.35

Source : (Research Results, 2024)

Figure 19. Summary YOLO FPS results

Based on Figure 19, YOLOv5 consistently achieves the highest average FPS, with 33.46 for Video 1 and 33.92 for Video 2, highlighting its superior speed and efficiency in both scenarios. Furthermore, YOLOv5 records the highest FPS of 39.66 in Video 2, underscoring its capability to handle rapid processing under optimal conditions. YOLOv8 follows closely with an average FPS of 32.12 for Video 1 and 32.49 for Video 2, reflecting its efficient performance in real-time applications. It achieves a peak FPS of 36.91 in Video 2, further solidifying its status as a high-performing variant. Similarly, YOLOv7 and YOLOv10 exhibit strong and comparable performance, with YOLOv7 recording average FPS values of 30.87 for Video 1 and 30.71 for Video 2, and YOLOv10 achieving slightly higher values of 31.87 and 31.95 for the respective videos. Both models show competitive peak FPS values, with YOLOv7 reaching 33.26 and YOLOv10 achieving 33.67 in Video 2. YOLOv11 also demonstrates commendable speed, with average

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6236

FPS values of 31.88 for Video 1 and 32.09 for Video 2, and a peak FPS of 33.35 in Video 2, making it a reliable choice for applications requiring consistent frame rates. In contrast, YOLOv9 shows moderate performance with average FPS values of 25.66 for Video 1 and 28.25 for Video 2, and a peak FPS of 31.45 in Video 2, indicating room for improvement in optimizing its processing speed.

YOLOv6 exhibits consistent but slightly lower performance, with average FPS values of 26.39 for Video 1 and 26.48 for Video 2, and a peak FPS of 29.49 in Video 1, suggesting its suitability for applications with moderate speed requirements. YOLOv3, while achieving competitive accuracy, has relatively modest FPS values of 22.53 for Video 1 and 22.57 for Video 2, and a peak FPS of 26.72 in Video 1, reflecting its older architecture and processing limitations. Finally, YOLOv4 records the lowest FPS among all variants, with averages of 19.70 for Video 1 and 19.72 for Video 2, and a peak FPS of 20.10 in Video 2. This highlights its limited speed capabilities, which may restrict its applicability in real-time or high-throughput detection tasks.

YOLOv7 outperforms all other versions, including YOLOv8 through YOLOv11, in terms of accuracy. This improvement may be attributed to the implementation of the E-ELAN (Extended Efficient Layer Aggregation Network) architecture in YOLOv7, which improves feature integration and allows for more precise detection under varying conditions, , particularly under varying conditions such as distance, scale, and lighting (e.g., sunny and cloudy), as observed in this study [45]. Consequently, YOLOv7 achieves superior accuracy over earlier versions like YOLOv3 to YOLOv6. In contrast, post-YOLOv7 versions (YOLOv8 to YOLOv11) prioritize a balance between accuracy and speed (FPS) and no longer implement the E-ELAN architecture [46]. These versions employ more efficient techniques to maintain rapid performance without significantly compromising accuracy. However, this trade-off results in slightly lower accuracy compared to YOLOv7, as the focus shifts toward computational efficiency and stable FPS [45]. The study's findings confirm this, showing that while YOLOv8 and YOLOv11 maintain stable accuracy and FPS, they do not exceed YOLOv7's accuracy levels.

Although YOLOv5 achieves the highest FPS (33.92), its lower accuracy compared to YOLOv7 highlights the importance of balancing FPS with detection precision, particularly for real-time applications. In this context, YOLOv7 emerges as the most effective model due to its ability to deliver an optimal balance between speed and accuracy, making it highly suitable for implementation in smart parking systems. It achieves the highest overall accuracy of 99.57%, excelling in both long-range (99.48%) and close-range (99.78%) detection scenarios. Although its average FPS of 30.79 is not the fastest among the evaluated models, it comfortably exceeds the 30 FPS threshold for real-time applications [47], making it ideal for systems prioritizing precision. However, its FPS stability is slightly less consistent, which may be a consideration for applications requiring highly stable frame rates. While FPS is an important metric, especially for real-time applications, it is not the sole determinant of usability; in this case, accuracy plays a critical role. For systems where accuracy is the top priority, ensuring precise detection and minimizing errors such as false positives or false negatives, YOLOv7 is the clear choice due to its unparalleled precision.

Following YOLOv7, YOLOv11 is a strong contender, achieving an overall accuracy of 98.72% and an average FPS of 31.99, striking a near-perfect balance between accuracy and speed. Similarly, YOLOv8 offers a balanced approach, achieving 98.51% accuracy and an average FPS of 32.31, with consistent FPS stability, making it a versatile choice for various parking detection scenarios. These versions demonstrate robust performance across different conditions, making them suitable for real-time applications where both accuracy and speed are critical. YOLOv10, while achieving competitive accuracy (97.08% for long-range and 87.78% for close-range), exhibits lower performance in close-range detection, suggesting limitations in handling short-distance scenarios. Despite its FPS stability, with an average of 31.87 for Video 1 and 31.95 for Video 2, its lower accuracy in close-range scenarios may limit its applicability in environments requiring high precision across varying distances.

On the other hand, YOLOv9, while achieving the highest accuracy for long-range detection (99.58%), shows a slight decline in close-range scenarios (95.33%), indicating potential challenges in maintaining consistent performance across varying distances. Its FPS, averaging 25.66 for Video 1 and 28.25 for Video 2, raises concerns for real-time implementation due to its relatively lower speed. In terms of FPS stability, YOLOv4 demonstrates the most consistent performance, maintaining stable frame rates despite its relatively lower average FPS (19.70 for Video 1 and 19.72 for Video 2). However, its lower accuracy (90.94% for Video 1 and 98.44% for Video 2) may limit its applicability in scenarios requiring high precision.

YOLOv6, on the other hand, shows improved accuracy in short-distance detection, achieving 96.89% accuracy for Video 2 compared to 77.43% for Video 1, indicating that it is better optimized for closer object detection. However, it demonstrates the lowest accuracy, particularly in detecting small objects at long distances. This limitation may be due to its fully anchor-free object detection (OD) architecture, which predicts bounding boxes directly from object center points without traditional anchor boxes [48]. While this simplifies the detection process, it can struggle with accurately identifying small objects, especially in occluded or complex backgrounds. Notably, YOLOv5 introduced anchor-free detection as an optional feature, retaining anchor boxes by default [49]. In contrast, YOLOv7 to YOLOv11 fully adopt anchor-free methods, incorporating additional enhancements like transformers, GANs, PGI, and GELAN to improve detection performance [46]. This study highlights YOLOv6's limitations in detecting small objects at long ranges, resulting in lower accuracy and suboptimal FPS compared to other versions, making it less suitable for long-range parking scenarios.

The results of this study demonstrate significant achievements, with satisfactory levels of accuracy and frame per second (FPS). A performance comparison of various YOLO versions, particularly the latest ones, reveals substantial improvements over previous studies. Research comparing multiple object detection methods for parking slot detection provides valuable insights, highlighting YOLO as one of the optimal methods [14]. However, a prior study by utilized only YOLOv3 and did not include the latest YOLO versions. Another study evaluating the latest YOLO versions compared YOLOv8 and YOLOv10 for vehicle detection, achieving high accuracy [50]. Unfortunately, this study did not incorporate YOLOv11, lacked FPS-based speed detection analysis, and omitted specific discussions on parking slot availability detection. Additionally, a study comparing object detection algorithms under varying weather conditions evaluated YOLOv3 to YOLOv7, reporting impressive results for YOLOv7 in terms of accuracy and inference time (FPS) [51]. Nevertheless, this study did not include YOLOv8 through YOLOv11 and was not focused on parking slot detection.

This study evaluates the performance of YOLO (You Only Look Once) in detecting vehicles of various sizes, ranging from small to large, in both near and far parking scenarios. Additionally, the study conducts tests under different weather conditions, specifically sunny and cloudy environments, as well as in the presence of obstructions such as trees. The results demonstrate that YOLO performs effectively in detecting vehicles under sufficient sunlight and shaded conditions. However, this study is limited to sunny and cloudy conditions, with no testing conducted in darker lighting environments. This limitation opens avenues for further research into YOLO's performance under low-light conditions.

The findings of this study are influenced by several factors, particularly the density of detected parking spaces, which significantly impact the FPS (Frames Per Second) of YOLO-based detection systems. This is evident in the results, where Video 1, which detected a higher number of cars, exhibited a lower FPS compared to Video 2, which detected only five parking spaces. Additionally, the study focused on testing under sunny and cloudy weather conditions, as supported by the dataset used. However, the results may yield lower accuracy when implemented in other real-world conditions, such as rainy or nighttime environments. Furthermore, the models used in this study may experience reduced accuracy when applied to parking areas with higher density or less structured layouts, such as those with unclear or frequently changing parking spaces, as well as improper camera placement that is not perpendicular to the parking area. Irregular parking arrangements may obstruct vehicle visibility, leading to decreased detection accuracy.

This study has certain limitations, including the relatively small number of video samples (two videos with specific camera perspectives) and the use of standard parking lot layouts. A larger and more diverse dataset could provide a broader range of scenarios to evaluate the performance of different YOLO versions under various environmental factors, such as weather conditions and lighting. Additionally, the reliance on a fixed CCTV setup in this study limits its generalizability to real-world dynamic environments, where parking spaces may be occupied by different vehicle types and orientations, potentially affecting detection accuracy.

Despite these limitations, the findings from this study contribute significantly to the ongoing development of efficient smart parking systems. By identifying which YOLO versions perform best under varying conditions, this research can guide future implementations of real-time vehicle detection systems, particularly in urban parking infrastructures. Furthermore, the study highlights the potential of YOLO-based systems to integrate seamlessly with existing surveillance infrastructure, offering a cost-effective solution for

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6236

parking management. This work can serve as a foundational reference for future studies aiming to optimize YOLO models for different detection contexts.

CONCLUSION

This study evaluates YOLO versions 3 through 11 for detecting vacant parking spaces and identifies YOLOv7 as the most effective model, achieving the highest accuracy (99.57%) with a suitable FPS (30.79) for real-time applications. This makes it ideal for systems that prioritize precision. YOLOv8 and YOLOv11 also demonstrated strong performance, achieving overall accuracies (98.51% and 98.72%) along with stable and high FPS rates (32.31 and 31.99). These versions are well-suited for applications requiring a balance between accuracy and speed, making them equally effective for smart parking solutions. Additionally, YOLOv5 stood out for its notable computational efficiency, making it particularly suitable for real-time implementations. However, the study identified certain limitations. The dataset used in this research consisted of a relatively small number of videos, with fixed perspectives and standard parking layouts, which may not fully capture the diversity of real-world scenarios. Environmental factors, such as variable weather conditions and dynamic lighting, were not incorporated into the testing, potentially limiting the generalizability of the findings. Additionally, some YOLO versions exhibited challenges in maintaining consistent performance under specific conditions, signaling the need for further refinement.

A key contribution of this study lies in its comprehensive and systematic approach to evaluating YOLO versions for parking space detection, addressing both theoretical and practical challenges. By providing empirical evidence and actionable insights, this research advances smart parking system development and lays the groundwork for future innovations in urban mobility and parking management. Furthermore, this study fills a significant research gap by offering a structured evaluation of YOLO versions specifically for parking slot detection, an area that has been largely unexplored in previous literature. While prior research has primarily focused on general object detection or individual YOLO versions, this study presents a comparative analysis that quantifies the strengths and weaknesses of each model within the parking management domain. Additionally, this study incorporates YOLOv11, the latest version released in September 2024, providing valuable insights into advancements in YOLO architectures and their applicability to real-world challenges such as smart parking systems.

As a recommendation, future research should focus on expanding datasets to include more varied parking configurations, environmental conditions, and vehicle orientations to improve model robustness. Incorporating adaptive pre-processing techniques for changing weather and lighting conditions, as well as testing on dynamic surveillance systems, could enhance the practical applicability of YOLO-based detection methods. Furthermore, exploring hybrid architectures, such as combining YOLO with advanced techniques like transformers or Graph Neural Networks (GNNs), may further optimize detection performance and efficiency. Additionally, Future studies should explore the performance of YOLO models in nighttime conditions and integrate IoT or cloud-based systems to enhance scalability and real-time responsiveness. Low-light environments pose significant challenges for object detection, often requiring additional preprocessing techniques or infrared imaging to maintain accuracy. By addressing these limitations, future implementations can ensure consistent performance regardless of lighting conditions. Additionally, the integration of IoT or cloud platforms could allow for centralized data processing, remote monitoring, and automated updates, thereby increasing operational efficiency and reducing maintenance overhead in smart parking systems.

REFERENCE

[1] E. Khakimova and E. Tokhirov, “CAR PARKING PROBLEMS IN CITIES, CAUSES AND SOLUTIONS,” *Universum:Technical sciences*, vol. 110, no. 5, May 2023, doi: 10.32743/UniTech.2023.110.5.15513.

[2] C. Ma, X. Huang, and J. Li, “A review of research on urban parking prediction,” *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 11, no. 4, pp. 700–720, Aug. 2024, doi: 10.1016/j.jtte.2023.11.004.

[3] I. B. Asianuba and A. Aliyu, “An Improved IOT Smart Parking System,” *International Journal of Recent Engineering Science*, vol. 10, no. 2, pp. 7–12, Apr. 2023, doi: 10.14445/23497157/IJRES-V10I2P102.

[4] A. Alhardi and E. Dincelli, “Smart Parking Systems and their Impact on the E7ciency of Parking O7cers,” *Americas Conference on Information Systems (AMCIS) 2023*, 2023.

[5] A. I. Pulungan, S. Sumarno, I. Gunawan, H. S. Tambunan, and A. R. Damanik, "Rancang Bangun Sistem Parkir dan Ketersediaan Slot Parkir Otomatis Menggunakan Arduino," *Jurnal Ilmu Komputer dan Informatika*, vol. 2, no. 2, pp. 127–136, Jun. 2022, doi: 10.54082/jiki.33.

[6] A. Raj and S. D. Shetty, "Smart parking systems technologies, tools, and challenges for implementing in a smart city environment: a survey based on IoT & ML perspective," *International Journal of Machine Learning and Cybernetics*, vol. 15, no. 7, pp. 2673–2694, Jul. 2024, doi: 10.1007/s13042-023-02056-5.

[7] Md. R. Alam, S. Saha, Md. B. Bostami, Md. S. Islam, Md. S. Aadeeb, and A. K. M. M. Islam, "A Survey on IoT Driven Smart Parking Management System: Approaches, Limitations and Future Research Agenda," *IEEE Access*, vol. 11, pp. 119523–119543, 2023, doi: 10.1109/ACCESS.2023.3327306.

[8] D. Rajawat, B. P. Lohani, A. Rana, A. Srivastava, P. Yadav, and S. Gupta, "Object Detection in Images and Videos Using OpenCV: A Comparative Study of Deep Learning and Traditional Computer Vision Techniques," in *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, IEEE, Dec. 2023, pp. 141–146. doi: 10.1109/UPCON59197.2023.10434536.

[9] N. Zhao, K. Wang, J. Yang, F. Luan, L. Yuan, and H. Zhang, "CMCA-YOLO: A Study on a Real-Time Object Detection Model for Parking Lot Surveillance Imagery," *Electronics (Basel)*, vol. 13, no. 8, p. 1557, Apr. 2024, doi: 10.3390/electronics13081557.

[10] S. Yamin Siddiqui, M. Adnan Khan, S. Abbas, and F. Khan, "Smart occupancy detection for road traffic parking using deep extreme learning machine," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 727–733, Mar. 2022, doi: 10.1016/j.jksuci.2020.01.016.

[11] M. M. Hasan, Z. Wang, M. A. I. Hussain, and K. Fatima, "Bangladeshi Native Vehicle Classification Based on Transfer Learning with Deep Convolutional Neural Network," *Sensors*, vol. 21, no. 22, p. 7545, Nov. 2021, doi: 10.3390/s21227545.

[12] Y. Yuldashev, M. Mukhiddinov, A. B. Abdusalomov, R. Nasimov, and J. Cho, "Parking Lot Occupancy Detection with Improved MobileNetV3," *Sensors*, vol. 23, no. 17, Sep. 2023, doi: 10.3390/s23177642.

[13] W. Li, L. Cao, L. Yan, J. Liao, and Z. Wang, "Vacant parking slot detection and tracking during driving and parking with a standalone around view monitor," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 6, pp. 1539–1551, May 2021, doi: 10.1177/0954407020980869.

[14] 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.

[15] L. He, Y. Zhou, L. Liu, and J. Ma, "Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites," *Buildings*, vol. 14, no. 12, p. 3777, Nov. 2024, doi: 10.3390/buildings14123777.

[16] P. Azevedo and V. Santos, "Comparative analysis of multiple YOLO-based target detectors and trackers for ADAS in edge devices," *Rob Auton Syst*, vol. 171, p. 104558, Jan. 2024, doi: 10.1016/j.robot.2023.104558.

[17] 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, Nov. 2023, doi: 10.3390/make5040083.

[18] A. Vijayakumar and S. Vairavasundaram, "YOLO-based Object Detection Models: A Review and its Applications," *Multimed Tools Appl*, Mar. 2024, doi: 10.1007/s11042-024-18872-y.

[19] H. N. Syifa' and A. Nugroho, "PERFORMANCE OF THE YOLOV5 ALGORITHM TO DETECT HUMANS IN THE REAR EXCAVATOR AREA," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 10, no. 1, pp. 197–207, Aug. 2024, doi: 10.33480/jitk.v10i1.5576.

[20] Y. Wang, Y. Jiang, H. Xu, C. Xiao, and K. Zhao, "Detection Method of Key Ship Parts Based on YOLOv11," *Processes*, vol. 13, no. 1, p. 201, Jan. 2025, doi: 10.3390/pr13010201.

[21] Y. Li, W. Leong, and H. Zhang, "YOLOv10-Based Real-Time Pedestrian Detection for Autonomous Vehicles," in *2024 IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA)*,

IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/ICSIPA62061.2024.10686546.

[22] D.-L. Nguyen, X.-T. Vo, A. Priadana, and K.-H. Jo, "Car Detector Based on YOLOv5 for Parking Management," 2023, pp. 102–113. doi: 10.1007/978-3-031-36886-8_9.

[23] M. Hou, W. Hao, Y. Dong, and Y. Ji, "A detection method for the ridge beast based on improved YOLOv3 algorithm," *Herit Sci*, vol. 11, no. 1, p. 167, Aug. 2023, doi: 10.1186/s40494-023-00995-4.

[24] Q. Yu, H. Liu, and Q. Wu, "An Improved YOLO for Road and Vehicle Target Detection Model," *Journal of ICT Standardization*, vol. 11, no. 2, pp. 197–216, 2023, doi: 10.13052/jicts2245-800X.1125.

[25] F. Li, T. Sun, P. Dong, Q. Wang, Y. Li, and C. Sun, "MSF-CSPNet: A Specially Designed Backbone Network for Faster R-CNN," *IEEE Access*, vol. 12, pp. 52390–52399, 2024, doi: 10.1109/ACCESS.2024.3386788.

[26] J. Doherty, B. Gardiner, E. Kerr, and N. Siddique, "BiFPN-YOLO: One-stage object detection integrating Bi-Directional Feature Pyramid Networks," *Pattern Recognit*, vol. 160, p. 111209, Apr. 2025, doi: 10.1016/j.patcog.2024.111209.

[27] J. Liu, Q. Cai, F. Zou, Y. Zhu, L. Liao, and F. Guo, "BiGA-YOLO: A Lightweight Object Detection Network Based on YOLOv5 for Autonomous Driving," *Electronics (Switzerland)*, vol. 12, no. 12, Jun. 2023, doi: 10.3390/electronics12122745.

[28] J. Wei, Y. Qu, M. Gong, Y. Ma, and X. Zhang, "VE-YOLOv6: A Lightweight Small Target Detection Algorithm," in *2024 4th International Conference on Neural Networks, Information and Communication (NNICE)*, IEEE, Jan. 2024, pp. 873–876. doi: 10.1109/NNICE61279.2024.10498732.

[29] W. Wang, Z. Dai, J. Liu, and P. Wu, "Underwater Target Detection Technology Based on YOLO v7," in *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering, ICSECE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 265–270. doi: 10.1109/ICSECE58870.2023.10263316.

[30] M. Yang, X. Tong, and H. Chen, "Detection of Small Lesions on Grape Leaves Based on Improved YOLOv7," *Electronics (Switzerland)*, vol. 13, no. 2, Jan. 2024, doi: 10.3390/electronics13020464.

[31] M. Shoman, T. Ghoul, G. Lanzaro, T. Alsharif, S. Gargoum, and T. Sayed, "Enforcing Traffic Safety: A Deep Learning Approach for Detecting Motorcyclists' Helmet Violations Using YOLOv8 and Deep Convolutional Generative Adversarial Network-Generated Images," *Algorithms*, vol. 17, no. 5, May 2024, doi: 10.3390/a17050202.

[32] Y. Xiang *et al.*, "Real-Time Detection Algorithm for Kiwifruit Canker Based on a Lightweight and Efficient Generative Adversarial Network," *Plants*, vol. 12, no. 17, Sep. 2023, doi: 10.3390/plants12173053.

[33] X. Wang, C. Zhang, Z. Qiang, C. Liu, X. Wei, and F. Cheng, "A Coffee Plant Counting Method Based on Dual-Channel NMS and YOLOv9 Leveraging UAV Multispectral Imaging," *Remote Sens (Basel)*, vol. 16, no. 20, Oct. 2024, doi: 10.3390/rs16203810.

[34] Y. Wang, Q. Rong, and C. Hu, "Ripe Tomato Detection Algorithm Based on Improved YOLOv9," *Plants*, vol. 13, no. 22, p. 3253, Nov. 2024, doi: 10.3390/plants13223253.

[35] A. Sundaresan Geetha, M. A. R. Alif, M. Hussain, and P. Allen, "Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy," *Vehicles*, vol. 6, no. 3, pp. 1364–1382, Aug. 2024, doi: 10.3390/vehicles6030065.

[36] J. Mei and W. Zhu, "BGF-YOLOv10: Small Object Detection Algorithm from Unmanned Aerial Vehicle Perspective Based on Improved YOLOv10," *Sensors*, vol. 24, no. 21, Nov. 2024, doi: 10.3390/s24216911.

[37] Y. Wan, H. Wang, L. Lu, X. Lan, F. Xu, and S. Li, "An Improved Real-Time Detection Transformer Model for the Intelligent Survey of Traffic Safety Facilities," *Sustainability*, vol. 16, no. 23, p. 10172, Nov. 2024, doi: 10.3390/su162310172.

[38] G. Jocher, "Ultralytics YOLO Docs," https://docs.ultralytics.com/models/.

[39] C. Duan and S. Luo, "Design of Pedestrian Detection System based on OpenCV," in *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, IEEE, Oct. 2022, pp. 256–259. doi: 10.1109/AIAM57466.2022.00055.

[40] N.-J. Huang, C.-J. Huang, and S.-K. Huang, "Pain Pickle: Bypassing Python Restricted Unpickler for Automatic Exploit Generation," in *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, IEEE, Dec. 2022, pp. 1079–1090. doi: 10.1109/QRS57517.2022.00111.

[41] H. Dai, X. Peng, X. Shi, L. He, Q. Xiong, and H. Jin, "Reveal training performance mystery between TensorFlow and PyTorch in the single GPU environment," *Science China Information Sciences*, vol. 65, no. 1, p. 112103, Jan. 2022, doi: 10.1007/s11432-020-3182-1.

[42] M. Pavithra, P. Surya Karthikesh, B. Jahnavi, M. Navyalokesh, Raja. C, and K. Lokesh Krishna, "Implementation of Enhanced Security System using Roboflow," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE, Mar. 2024, pp. 1–5. doi: 10.1109/ICRITO61523.2024.10522313.

[43] J. Mijalkovic and A. Spognardi, "Reducing the False Negative Rate in Deep Learning Based Network Intrusion Detection Systems," *Algorithms*, vol. 15, no. 8, p. 258, Jul. 2022, doi: 10.3390/a15080258.

[44] C. Mwitta, G. C. Rains, and E. Prostko, "Evaluation of Inference Performance of Deep Learning Models for Real-Time Weed Detection in an Embedded Computer," *Sensors*, vol. 24, no. 2, p. 514, Jan. 2024, doi: 10.3390/s24020514.

[45] 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," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2023, pp. 7464–7475. doi: 10.1109/CVPR52729.2023.00721.

[46] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, p. 336, Dec. 2024, doi: 10.3390/computers13120336.

[47] Y. Ye, X. Ma, X. Zhou, G. Bao, W. Wan, and S. Cai, "Dynamic and Real-Time Object Detection Based on Deep Learning for Home Service Robots," *Sensors*, vol. 23, no. 23, p. 9482, Nov. 2023, doi: 10.3390/s23239482.

[48] D. Yu *et al.*, "Anchor-Free Arbitrary-Oriented Object Detector Using Box Boundary-Aware Vectors," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 15, pp. 2535–2545, 2022, doi: 10.1109/JSTARS.2022.3158905.

[49] J. Cao, J. Zhang, B. Li, L. Gao, and J. Zhang, "RetinaMOT: rethinking anchor-free YOLOv5 for online multiple object tracking," *Complex & Intelligent Systems*, vol. 9, no. 5, pp. 5115–5133, Oct. 2023, doi: 10.1007/s40747-023-01009-3.

[50] A. Sundaresan Geetha, M. A. R. Alif, M. Hussain, and P. Allen, "Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy," *Vehicles*, vol. 6, no. 3, pp. 1364–1382, Aug. 2024, doi: 10.3390/vehicles6030065.

[51] Z. Haimer, K. Mateur, Y. Farhan, and A. A. Madi, "YOLO Algorithms Performance Comparison for Object Detection in Adverse Weather Conditions," in *2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, IEEE, May 2023, pp. 1–7. doi: 10.1109/IRASET57153.2023.10152924.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri