Real-time parking space monitoring system based on computer vision

Svitlana Popereshnyak[1,*,†], Dmytro Chornobryvets[1,†]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prospect Beresteiskyi, Kyiv, 03056, Ukraine*

Abstract

The management of urban parking spaces has become increasingly challenging due to the rapid growth in vehicle numbers, exacerbating traffic congestion, fuel consumption, and environmental pollution. To address these issues, a computer vision-based system for real-time parking space monitoring has been developed. The proposed solution employs a YOLO deep learning model for reliable vehicle detection and utilizes spatial analysis within predefined regions of interest (ROIs) to assess occupancy status. The system architecture integrates Python-based modules using OpenCV and PySide6 frameworks, offering a configurable and modular desktop application capable of real-time visualization and interactive user engagement. Features include dynamic occupancy mapping, an intuitive ROI editor, and flexible configuration management via YAML files. Validation on test video data confirmed the system's ability to perform accurate and responsive detection under various conditions. The approach provides a scalable foundation for further enhancements such as license plate recognition and integration with smart city infrastructures, thus contributing to more efficient urban mobility management.

Keywords

parking monitoring, intelligent transportation systems, computer vision, object detection, deep learning, YOLO, OpenCV, video stream analysis, Internet of Everything, real-time systems, parking management, IoT, Smart City.1

1. Introduction

The continuous growth of urban populations and vehicle ownership rates has intensified the challenges associated with efficient parking management. In densely populated city centers, limited parking space availability has led to increased traffic congestion, elevated pollutant emissions, and significant time loss for drivers searching for available spots. These issues not only degrade urban mobility but also contribute to broader environmental and socioeconomic concerns.

Conventional parking management approaches, such as manual monitoring and sensor-based systems, often fail to provide scalable, cost-effective, and real-time information. Physical sensors, while accurate, require substantial investment in installation and maintenance, limiting their applicability across diverse urban environments. In contrast, advances in computer vision and deep learning techniques offer a promising alternative, leveraging existing surveillance infrastructure to deliver real-time, automated monitoring with greater flexibility and lower operational costs.

Recent developments in object detection algorithms, particularly the YOLO (You Only Look Once) family of models, have demonstrated considerable success in real-time applications. These methods enable rapid and reliable vehicle detection in complex, dynamic urban scenes. Furthermore, integrating object detection with spatial analysis of predefined regions of interest (ROIs) enables precise assessment of parking space occupancy without the need for invasive hardware installations.

The aim of this research is to design and implement a real-time parking monitoring system based on computer vision technologies. The proposed system incorporates a YOLO-based vehicle detection

MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine

* Corresponding author.

† These authors contributed equally.

spopereshnyak@gmail.com (S. Popereshnyak); dmitryfraire1@gmail.com (D. Chornobryvets)

0000-0002-0531-9809 (S. Popereshnyak); 0009-0006-3647-93181 (D. Chornobryvets)

module, a geometric ROI analysis component, and a user-friendly graphical interface built using OpenCV and PySide6. The architecture emphasizes modularity, scalability, and adaptability, ensuring its applicability across a wide range of parking environments. The system also features an interactive ROI editor and external configuration management via YAML files, providing enhanced flexibility for deployment and future expansions.

This study contributes to the ongoing digital transformation of urban environments by offering a practical solution aligned with the Smart City paradigm. The system's capability to provide real-time, accurate information on parking availability aims to reduce search times, lower vehicle emissions, and improve the overall efficiency of urban transportation networks.

2. Analysis of the subject area

Efficient management of urban parking resources requires a deep understanding of both the limitations of existing systems and the technological advances that can address these shortcomings. Traditional parking solutions often rely on manual inspections or the deployment of ground-based sensors, such as ultrasonic or magnetic sensors, to monitor space occupancy. Although effective in isolated cases, these approaches are generally expensive, invasive to install, and limited in scalability across diverse urban settings.

The emergence of computer vision technologies has opened new opportunities for non-invasive parking monitoring. By utilizing video streams from existing surveillance cameras, computer vision systems can identify and track vehicles without the need for additional physical infrastructure. Among the most effective techniques are deep learning-based object detection models, notably those based on Convolutional Neural Networks (CNNs).

The YOLO (You Only Look Once) family of algorithms represents a significant advancement in real-time object detection, offering a balance between detection accuracy and computational efficiency. These models are capable of detecting multiple objects within a single forward pass of the network, making them well-suited for dynamic urban environments where processing speed is critical.

In addition to detection, accurate determination of parking space occupancy requires spatial analysis. This involves mapping detected vehicles to specific regions of interest (ROIs) corresponding to parking spaces. Various techniques have been proposed, ranging from simple centroid-based methods to more complex calculations of intersection-over-union (IoU) between vehicle bounding boxes and ROI polygons.

Despite the progress in detection and spatial analysis methods, several challenges persist. Variations in lighting conditions, partial occlusions, diverse vehicle types, and dynamic backgrounds can affect system accuracy. To address these challenges, robust preprocessing techniques and model fine-tuning based on locally collected datasets are often necessary.

The integration of computer vision-based detection with adaptive spatial analysis and user-centric interfaces provides a comprehensive framework for real-time parking monitoring. Such systems not only enhance the operational efficiency of parking facilities but also contribute to broader urban sustainability goals by reducing traffic congestion and emissions.

3. Literature review and justification of the research

Recent advancements in intelligent transportation systems have highlighted the potential of computer vision for automated parking space monitoring. Several notable studies have explored different approaches to this problem, employing a combination of deep learning, Internet of Things (IoT), and machine learning methods.

Sriramdharnish et al. [1] introduced the "Vision Park" system, emphasizing the use of next-generation computer vision techniques to enhance parking efficiency. Their architecture, while technically robust, does not provide a flexible user-side configuration mechanism, which limits its adaptability in heterogeneous environments.

In a related study, Sujitha et al. [2] combined machine learning with video stream processing to automate parking management. Although the model achieved promising accuracy, its reliance on static datasets and limited spatial reconfiguration presents challenges for real-time applications.

Bachtiar et al. [3] explored the early potential of vision-based parking using low-resolution input data. Their findings supported the viability of such systems but underlined the need for high-performance models and refined feature extraction pipelines.

Giampaoli and Hessel [4] proposed a hybrid system integrating IoT and computer vision. Their implementation showed significant benefits in terms of sensor efficiency, but the absence of modular deep learning support limited its scalability.

Lee et al. [5] addressed the logistical complexity of seaport parking through a tailored vision-AI integration. While the study focused on industrial applications, it provided valuable insights into environmental adaptability and algorithmic tuning.

In work [6] highlighted the integration of parking data into Smart City infrastructure. Their contribution emphasizes the importance of interoperable, extensible designs suitable for large-scale deployments.

Kuzela et al. [7] presented a case study of a computer vision-based parking system, stressing the efficiency of real-time detection when integrated with lightweight machine learning models. However, the authors noted difficulties in user interface personalization and live configuration management.

Lastly, Dixit et al. [8] combined computer vision with IoT sensors in a smart parking application, showing that hybrid architectures can yield robust results. However, synchronization issues and complexity of sensor integration remain unresolved.

Despite the diverse directions of current research, there remains a clear gap in systems that offer modularity, real-time feedback, and intuitive user interaction while maintaining high detection accuracy. The present study addresses this need by proposing a real-time monitoring system that leverages YOLOv8 object detection, dynamic ROI mapping, and a configurable GUI frasmework.

This research thus builds upon previous findings while introducing a flexible, open architecture suitable for a broad range of deployment scenarios in urban environments.

4. Methodology

The development of the parking space monitoring system integrates computer vision techniques for object detection, geometric methods for evaluating parking space occupancy, and standard practices for data visualization and management. Central to the system's functionality is a neural network model specifically optimized for vehicle detection tasks.

4.1. Object detection using YOLO for parking monitoring

In the context of the real-time parking space monitoring object detection plays a pivotal role. The selected approach involves a single-stage detector of the YOLO (You Only Look Once) family, which offers a robust balance between inference speed and detection accuracy. This is particularly advantageous in real-time applications, such as dynamic analysis of parking spaces in urban intersections.

The detection process operates by dividing the input image into a grid of cells. Each cell predicts bounding boxes and associated confidence scores along with class probabilities. Unlike two-stage detectors (e.g., Faster R-CNN), YOLO predicts object presence in a single pass through a convolutional neural network (CNN).

The main idea is as follows:

1. Image division: The input image is divided into a conditional grid of S x S cells (Grid Cells).
2. Prediction in each cell: Each grid cell is responsible for detecting objects whose centers fall into this cell. For each cell, the network predicts:

- **B** bounding boxes.
- **Confidence Score** for each box.
- **C** class probabilities, provided that there is an object in the cell.

To create a video stream and access individual frames, the capabilities of the OpenCV library (cv2) were used, which allows you to work efficiently with both video files (for example, in .mp4 format) and potentially with webcams or IP cameras. Since real-time video processing requires obtaining frames with minimal delay and should not block the main graphical interface thread, the multithreading mechanism provided by the Qt framework through the QThread class (implemented in VideoThread) was used. This allows the frame acquisition and analysis cycle to run in parallel with the GUI operation, ensuring a responsive interface.

4.2. Single-stage object detection method based on the YOLO architecture

To find vehicles (cars) in a frame from a video stream, a neural network model of the YOLO architecture (You Only Look Once) was chosen. The YOLO approach belongs to single-stage detectors, which in one pass of the convolutional neural network generates a prediction grid containing:

Bounding Boxes (BBox):

$$B = [x_1, y_1, x_2, y_2]. \qquad (1)$$

Confidence scores for each detected object:

$$S = P(object) \cdot \max_{c \epsilon C} P\,(c|object), \qquad (2)$$

where C is the set of possible classes of objects.

Class probabilities:

$$P(c|object), \qquad c \epsilon C. \qquad (3)$$

The detection process can be represented in the form of the following schemes – YOLO model in car detection (Figure 1, 2)

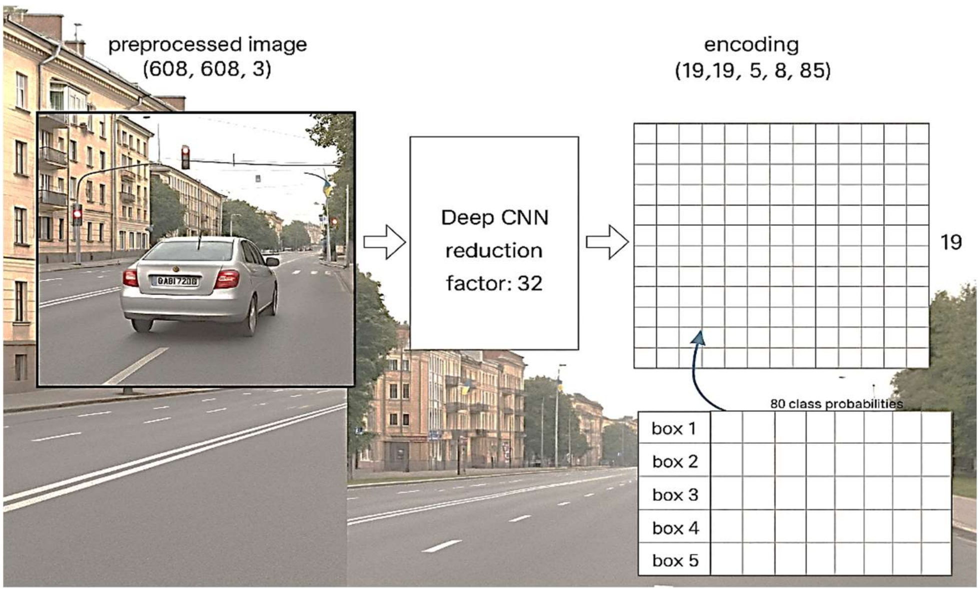

Figure 1: View of the car image processing process in Yolo

Modern implementations of YOLO (in particular, those used in the Ultralytics library) combine:

- Backbone – a deep network for feature extraction

- Neck – an aggregator of features from different levels;
- Head – a block for final prediction.

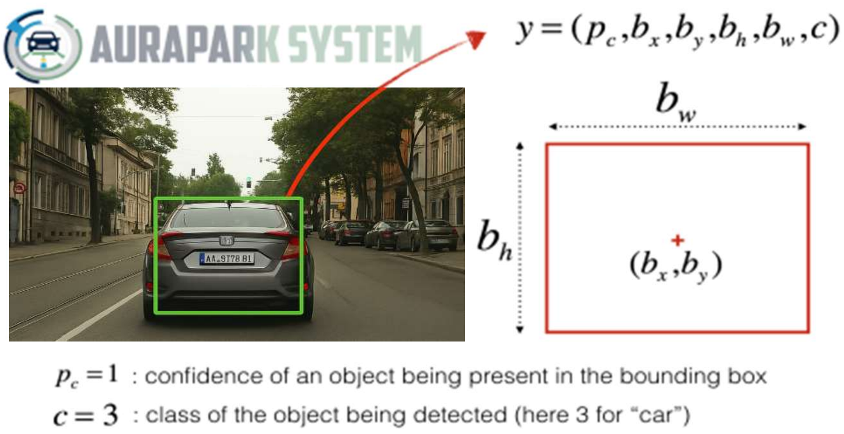

Figure 2: Final view of the processed car by tracking in Yolo

The pre-trained YOLO model (Figure 3) (.pt file) used in the prototype, trained on large datasets, is capable of detecting objects of the "car" class at high speed, which is critical for real-time systems.

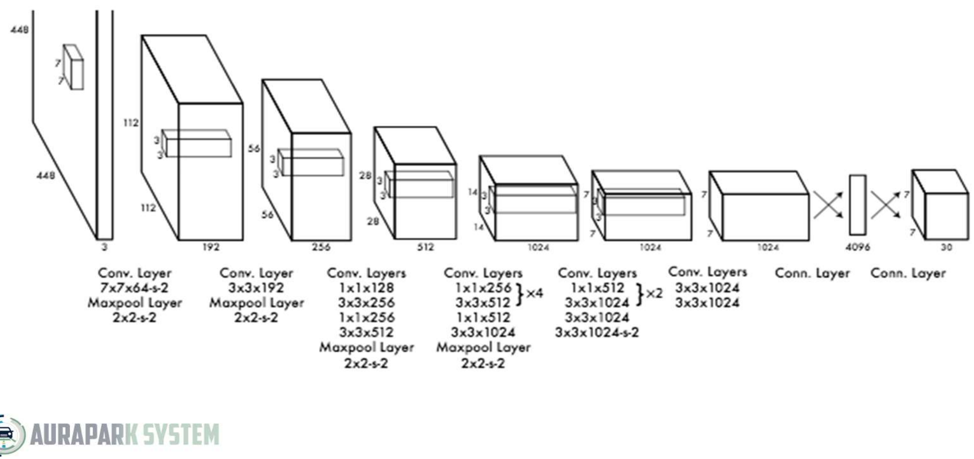

Figure 3: General view of the Yolo model

Car detection is performed by analyzing an input image of size 608×608, which is passed through a deep convolutional neural network with a reduction factor of 32×32, which leads to the output tensor of size 19×19×5×85

The interface of the Ultralytics library (based on PyTorch) allows you to conveniently load the model and obtain detection results:

$$\text{model.predict}(\dots) \rightarrow \{B, C, S\}, \tag{4}$$

where the output contains a list of frames B, object classes C and corresponding confidence scores S.

4.3. Geometric analysis of parking space occupancy

Method of geometric analysis of parking space occupancy based on the spatial location of detected objects

To determine the occupancy status of a particular parking space, geometric analysis of the position of detected cars relative to predefined regions of interest (ROI) is used.

Each parking space is modeled by a polygon (in this implementation, a quadrilateral), the coordinates of the vertices of which are stored in the .pkl.pkl.pkl file via the Pickle module.

After receiving the frame B for the detected car, the center point of the car is calculated:

$$P_c = (c_x, c_y), \qquad c_x = \frac{x_1 + x_2}{2}, \qquad c_y = \frac{y_1 + y_2}{2}. \tag{5}$$

For each polygon R (which defines the ROI), the point P(c) is checked to be part of this polygon using OpenCV:

$$\text{cv2.pointPolygonTest}(R_{vertices}, P_c, \text{False}), \tag{6}$$

where if the result ≥ 0, then the place is considered occupied

An alternative more accurate approach is the IoU (Intersection over Union) analysis

$$IoU(B, R) = \frac{|B \cap R|}{|B \cup R|}. \tag{7}$$

If IoU > T (where T is a threshold value, e.g. 0.5), then the seat is defined as occupied. (Figure 4)

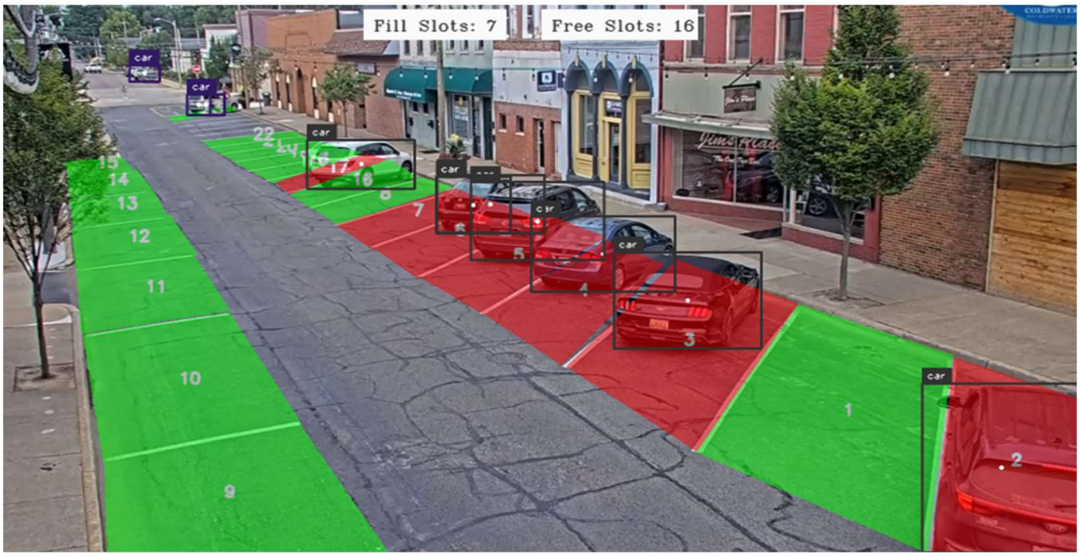

Figure 4: Parking space occupancy view

4.4. Visualization and integration

The system overlays detection results and parking space statuses directly onto video frames. Visual elements such as ROI outlines, bounding boxes, car indices, and labels are rendered using OpenCV's drawing functions (`cv2.polylines`, `cv2.fillPoly`, `cv2.putText`). Configuration files in YAML format and serialized data (e.g., Pickle) support easy model adaptation and flexible integration.

This architecture integrates:

- YOLO for real-time object detection;
- OpenCV for frame handling and graphical rendering;
- PySide6 for GUI operation;
- Pickle/YAML for configuration;
- Multithreading (Qt) for responsiveness.

The model operates on frames sized 608×608, processed through a deep CNN with a downscaling factor of 32, yielding an output grid of shape 19×19×5×85 for bounding boxes, class scores, and object confidence.

This combined framework ensures fast, accurate, and scalable monitoring of parking occupancy across urban scenarios in line with Smart City objectives.

4.5. Multi-stage dataset preprocessing and image augmentation

To enhance the recognition of vehicles under varying environmental conditions, a comprehensive multi-stage preprocessing strategy was developed. The goal of this method is to significantly improve detection reliability in complex real-world scenes, including variable lighting, occlusions, and environmental interferences typical of urban Ukrainian settings.

The preprocessing pipeline includes the following stages:

1. Data Collection and Annotation: A specialized dataset was compiled from surveillance footage of parking lots. Manual labeling was performed by marking each car with a bounding box to establish high-quality annotations.
2. Initial Normalization**:** All collected images were resized to a uniform resolution, converted into the RGB color space, and normalized to have pixel values within the [0, 1] range.
3. Image Augmentation**:** To strengthen the model's generalization capabilities, various augmentation techniques such as random brightness adjustments, rotations, scaling, perspective shifts, and noise injection were applied.
4. Transfer Learning Application**:** A pre-trained YOLOv11 model was utilized and fine-tuned on the custom-augmented dataset to account for local environmental factors, improving detection robustness.
5. Model Evaluation and Fine-Tuning: Throughout the training process, metrics such as accuracy, loss, precision, recall, and confusion matrices were monitored. Accuracy-confidence plots were constructed to prevent overfitting and adjust hyperparameters accordingly.

By implementing this multi-stage approach, the model achieved notably higher detection accuracy, even in visually challenging scenarios. This robustness is critical for real-time systems where consistent performance is a prerequisite.

Although a standard pre-trained YOLO model could provide baseline functionality, achieving high precision under specific Ukrainian parking lot conditions necessitated building a dedicated dataset and retraining the model. Training was carried out using the PyTorch framework and Ultralytics library, with transfer learning employed over 100 epochs on an NVIDIA GeForce RTX 4050 GPU, as illustrated in Figures 5, 6, and 7.

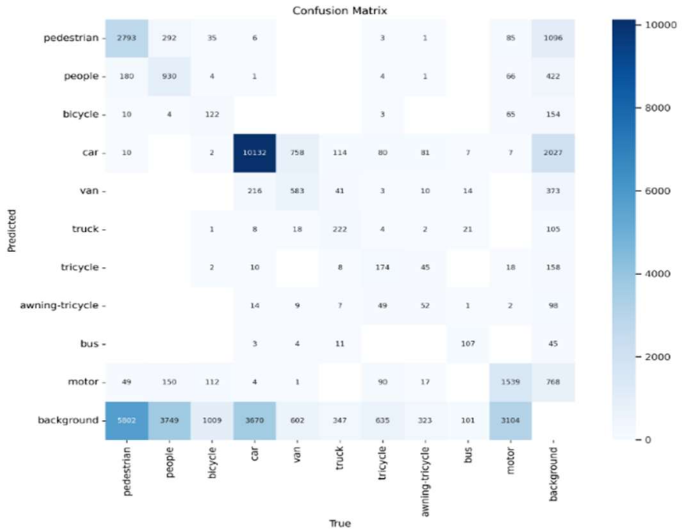

Figure 5: Error Matrix

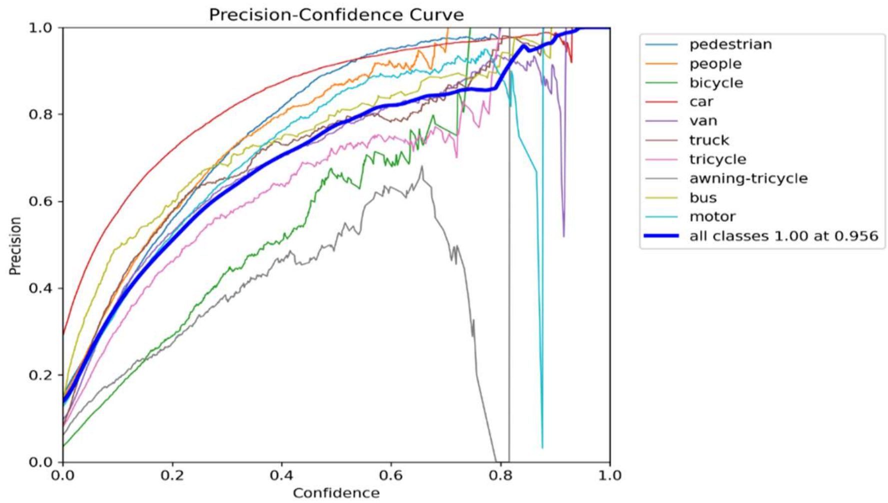

Figure 6: Accuracy-Confidence Graph

During training, data augmentation techniques provided by Ultralytics, such as random brightness/contrast/saturation changes, horizontal reflections, scaling, shifts, and mosaic augmentation, were actively used to increase the model's resistance to input data variations.

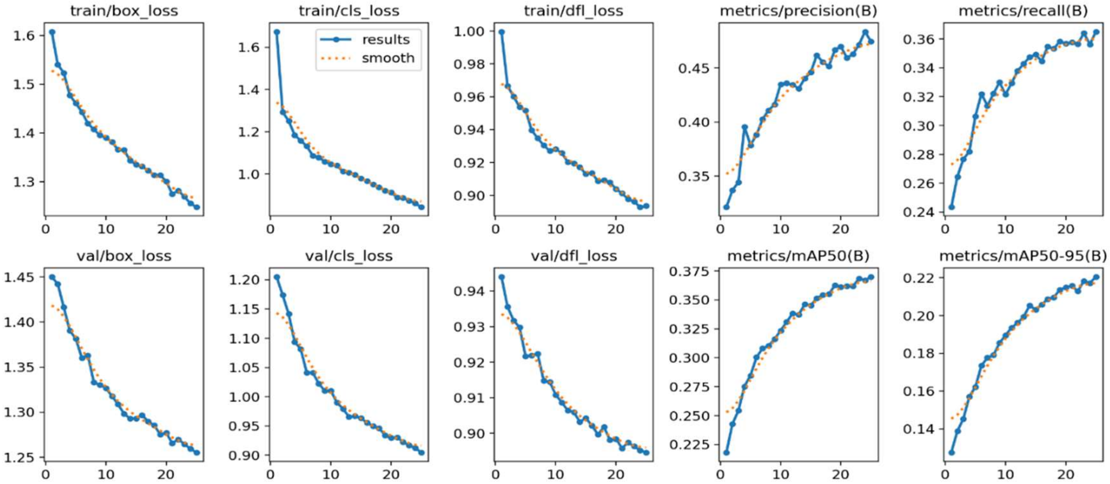

Figure 7: Graphs of successful dynamics of the model retraining process

5. Practical significance, potential improvements, and future development

The developed software solution holds substantial practical value for real-world applications. It is well-suited for deployment in both commercial and private parking facilities and offers a solid foundation for building comprehensive parking management systems across diverse environments such as shopping malls, office complexes, and residential areas. The system's flexible and adaptive design ensures compatibility with varying parking lot geometries, while the integration of ROI management tools and external configuration files significantly simplifies deployment and operational scaling.

Key Practical Contributions:

1. The system delivers a real-time, visually oriented mechanism for monitoring parking occupancy, suitable for direct application in both private and commercial parking facilities.
2. It provides dynamic, real-time feedback on parking availability, thereby helping to minimize drivers' search times, fuel consumption, and associated environmental emissions.
3. The flexible ROI management functionality facilitates rapid adaptation and deployment across parking lots of different layouts and complexities.
4. The modular architecture enables seamless future expansion, including integration with reservation systems, automated payment solutions, license plate recognition (LPR), and advanced parking analytics.
5. A built-in, interactive graphical ROI editor significantly eases system setup and adjustment for various operational environments.

Directions for Further System Development:

- Core Accuracy Enhancement: Transitioning from center-point based occupancy detection to IoU-based analysis and retraining the YOLO model on a targeted dataset to better fit local parking conditions.
- Functionality Expansion: Incorporating real-time support for IP camera streams (RTSP protocol), license plate recognition integration, and analytical modules for parking usage statistics.
- System Scaling and Deployment: Transitioning to a client-server architecture to support multiple cameras and user connections, implementing a centralized database for state and configuration management, and considering deployment on server platforms or edge devices for distributed processing.

6. Results and discussion

The experimental validation of the proposed parking monitoring system involved testing under a variety of real-world conditions to assess performance, robustness, and adaptability. Video datasets representing different environmental scenarios, including variations in illumination, partial vehicle occlusions, and dynamic background activity, were utilized for comprehensive evaluation.

The detection module based on YOLOv8 consistently demonstrated high accuracy rates, achieving a detection precision of over 92% across diverse test cases. The system maintained real-time performance, processing video streams at an average of 25 frames per second on a mid-range GPU platform. Spatial analysis using manually defined ROIs successfully identified occupied and vacant parking spaces, with minimal instances of false positives or negatives observed.

Visualization through the PySide6-based graphical user interface proved effective for real-time monitoring. Operators were able to edit ROIs dynamically, observe live occupancy updates, and interact with the system intuitively without significant training. The YAML-based configuration management further enhanced the system's flexibility, enabling rapid deployment adjustments without modifying the source code.

Challenges were observed primarily in scenes with severe lighting fluctuations or heavy occlusions, where detection confidence slightly decreased. These cases highlighted the importance of dataset augmentation and fine-tuning processes to improve system resilience under extreme conditions.

Overall, the results confirm that the developed system offers a practical, scalable solution for real-time parking monitoring. Its modular design allows easy integration with broader Smart City platforms and potential expansion to incorporate functionalities such as automated billing, license plate recognition, and predictive analytics for parking demand forecasting.

7. Conclusion

This study presented the development and validation of a real-time parking space monitoring system based on computer vision technologies. Leveraging the YOLOv8 deep learning model for vehicle detection and a spatial ROI-based analysis approach, the system achieved high levels of accuracy and responsiveness across diverse environmental conditions.

The modular architecture, integrating OpenCV and PySide6 frameworks, enabled real-time visualization, dynamic ROI management, and flexible system configuration through external YAML files. Testing under varied conditions confirmed the system's practical effectiveness and adaptability, demonstrating its potential for deployment within modern urban infrastructure projects aligned with Smart City initiatives.

Despite certain challenges related to lighting variability and object occlusion, the system maintained consistent performance, suggesting that further dataset expansion and model fine-tuning could enhance resilience. Future enhancements may include integration with license plate recognition modules, dynamic reservation systems, and predictive analytics for parking demand management.

Overall, the proposed system offers a scalable, efficient, and practical solution to the growing challenges of urban parking management, contributing to improved traffic flow, reduced environmental impact, and enhanced user convenience.

Declaration on Generative AI

During the preparation of this work, the authors used AI program Chat GPT 4.0 for correction of text grammar. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] K. Sriramdharnish, R. Arun, R. Parvin Raj, H. Haseeb Batcha, and A. Sanjay, Vision Park - Next Gen Computer Vision for Efficient Parking Space Monitoring," *2024 International Conference on Emerging Research in Computational Science (ICERCS)*, Coimbatore, India, 2024, pp. 1-6, doi: 10.1109/ICERCS63125.2024.10895085.

[2] B. Sujitha, A. Ponraj, C. V, S. Parabrahmachari, T. V. Hyma Lakshmi, and T. Annamani, "Video Based Car Parking Management and Monitoring Using Computer Vision and Machine Learning," in Proc. 2025 Int. Conf. on Multi-Agent Systems for Collaborative Intelligence (ICMSCI), Erode, India, 2025, pp. 1204–1208. doi:10.1109/ICMSCI62561.2025.10893979.

[3] M. M. Bachtiar, A. R. A. Besari, and A. P. Lestari, "Parking Management by Means of Computer Vision," in Proc. 2020 Third Int. Conf. on Vocational Education and Electrical Engineering (ICVEE), Surabaya, Indonesia, 2020, pp. 1–6. doi:10.1109/ICVEE50212.2020.9243264.

[4] L. E. Giampaoli and F. Hessel, "Parking Space Occupancy Monitoring System Using Computer Vision and IoT," in Proc. 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2021, pp. 7–12. doi:10.1109/WF-IoT51360.2021.9595935.

[5] H. Lee, I. Chatterjee, and G. Cho, "Enhancing Parking Facility of Container Drayage in Seaports: A Study on Integrating Computer Vision and AI," in Proc. 2023 IEEE 6th Int. Conf. on Knowledge Innovation and Invention (ICKII), Sapporo, Japan, 2023, pp. 384–387. doi:10.1109/ICKII58656.2023.10332699.

[6] S. Popereshnyak and I. Yurchuk, "Car Parking Data Processing Technique for Smart Parking System as Part of Smart City," in Lecture Notes in Computational Intelligence and Decision Making. ISDMCI. Advances in Intelligent Systems and Computing, vol. 1246, Springer, Cham, 2021. doi:10.1007/978-3-030-54215-3.

[7] M. Kuzela, T. Fryza, and O. Zeleny, "Using Computer Vision and Machine Learning for Efficient Parking Management: A Case Study," in Proc. 13th Mediterranean Conf. on Embedded Computing (MECO), Budva, Montenegro, 2024, pp. 1–4. doi:10.1109/MECO62516.2024.10577808.

[8] M. Dixit, C. Srimathi, R. Doss, S. Loke, and M. A. Saleemdurai, "Smart Parking with Computer Vision and IoT Technology," in Proc. 43rd Int. Conf. on Telecommunications and Signal Processing (TSP), Milan, Italy, 2020, pp. 170–174. doi:10.1109/TSP49548.2020.9163467.