Journal of Advances in Information Technology, Vol. 17, No. 4, 2026

Deep Learning-Based Occupancy Detection in Unmarked Parking Zones

Irina Dyomina [1], Natalya Denissova [2,*], Mukhamed Tolegenov [1], Saule Rakhmetullina [1], Aizhan Tlebaldinova [1], and Zarina Khassenova [1]

[1] School of Digital Technologies and Artificial Intelligence, D. Serikbayev East Kazakhstan Technical University, Ust-Kamenogorsk 070000, Kazakhstan
[2] Department of Information Technologies, D. Serikbayev East Kazakhstan Technical University, Ust-Kamenogorsk 070000, Kazakhstan
Email: idyomina@edu.ektu.kz (I.D.); ndenisova@edu.ektu.kz (N.D.); tolegenov.mukhamed.kz@gmail.com (M.T.); srakhmetullina@edu.ektu.kz (S.R.); atlebaldinova@edu.ektu.kz (A.T.); zkhasenova@edu.ektu.kz (Z.K.)
*Corresponding author

***Abstract*—This study proposes a method for automated detection of parking space occupancy in environments with absent or unclear markings. The system uses a U-Net11-based image segmentation model supported by a preprocessing pipeline that includes contrast correction, perspective transformation, and Contrast Limited Adaptive Histogram Equalization (CLAHE) enhancement to improve feature visibility under varying lighting conditions. The model was trained on a combined dataset from surveillance cameras and open-source images. On the test set, it achieved an accuracy of 74%, with Precision of 75.4%, Recall of 70.1%, F1-score of 72.6%, and Intersection over Union (IoU) of 57%. The Precision-Recall operating point (Precision = 75.42%, Recall = 70.27%) confirms reliable classification in challenging scenes. Despite the moderate segmentation accuracy, the achieved 74% is sufficient for practical parking-space detection because the final occupancy decision is based on aggregated spatial predictions rather than pixel-level precision. Local segmentation errors do not significantly affect the binary classification of “occupied/vacant”, making the system robust to noise, weather variability, and imperfect markings. Experiments confirm that occupancy determination remains stable even when segmentation boundaries are slightly distorted. The proposed approach improves the efficiency of identifying available parking spaces and can be integrated into smart city infrastructure to support automated parking monitoring and urban mobility optimization.**

***Keywords*—parking automation, parking occupancy map, computer vision, modified U-Net11 architecture, neural network model**

I. Introduction

Intelligent transport is an important component of the “smart city” concept, in which modern technologies are used to improve the efficiency of urban mobility and transport infrastructure management. The use of information and communication technologies, big data, and machine learning methods can improve traffic monitoring and optimize parking areas, reducing traffic jams and increasing user comfort. One of the important tasks of the “smart city” is the management of intelligent transport systems [1–4]. Despite the active development of intelligent transport systems and the widespread adoption of deep learning solutions, effective parking space management remains one of the most difficult tasks of a smart city. The main problem lies in the lack of a generalized analysis of deep learning methods applied to the tasks of identifying and predicting parking spaces, despite their growing relevance to reduce traffic jams and improve the efficiency of urban transport. In Ref. [5], this problem was solved by systematizing existing approaches, building a taxonomy of Deep Learning (DL) methods, and identifying their capabilities, limitations, and open research objectives. Despite significant progress in the field of automated transport, there are still unresolved issues related to the reliability of environmental perception. To solve this problem, three improvements to the Convolutional Neural Networks (CNN) model were proposed in Ref. [6], combining and deconvolving feature maps to account for scale, soft Non-Maximum Suppression (NMS) for processing overlaps, and optimizing reference frames based on object aspect ratio statistics, which significantly improved detection accuracy. In addition, despite the successful use of deep learning methods for monitoring parking spaces, the effectiveness of existing systems is significantly reduced in conditions of poor visibility, such as haze, which limits their practical applicability in urban environments. In Ref. [7], an intelligent parking system was proposed in which the preliminary removal of haze using specialized neural networks significantly increases the accuracy of CNN-based parking spot occupancy detection in fog conditions and without it, which is confirmed by experiments on the datasets RESIDE-β, CNRPark-EXT and clouded parking scenes. At the same time, existing

Manuscript received July 16, 2025; revised September 18, 2025; accepted December 23, 2025; published April 24, 2026.

doi: 10.12720/jait.17.4.759-776

parking occupancy detection systems are mainly focused on pre-marked and structured parking areas, which limits their applicability in difficult or unmarked conditions, especially in adverse weather situations. The use of deep learning methods and integration with the Internet of Things infrastructure represent a promising direction for automating parking monitoring and improving the accuracy of determining vacant spaces.

The purpose of the study is to develop an intelligent system for determining the occupancy of parking spaces in unmarked areas based on deep learning, using the convolutional neural network U-Net11 for semantic segmentation of parking spaces.

The proposed approach is aimed at automating parking monitoring without relying on predefined markings, which increases the accuracy of determining vacant spaces and helps optimize urban traffic and parking logistics. The key innovation of this study is the use of the U-Net11 convolutional neural network for semantic segmentation of parking spaces, which allows identifying vacant spaces.

II. LITERATURE REVIEW

Research in the field of intelligent parking systems demonstrates a wide range of approaches and technologies. These include hardware solutions based on Arduino and Wireless Sensor Network (WSN), computer vision methods, the development of automated parking complexes, the study of behavioural aspects, as well as intelligent algorithms and the integration of the Internet of Things with mobile applications. These areas provide a basis for analysing current achievements and identifying unresolved issues in this field. Taken together, they not only allow us to assess the current state of research, but also demonstrate that the proposed solutions are focused on increasing the efficiency of parking space utilisation, reducing the time spent searching for spaces, and improving the quality of service for users.

One example of applying deep learning to the task of automatic parking slot recognition is the study by Zinelli *et al.* [8]. They proposed a method based on convolutional networks (Faster-R-CNN/DCNN) for processing surround-view images and demonstrated the effectiveness of the approach under different lighting conditions and angles. This result confirmed that deep models are capable of generalising knowledge to new scenes, laying the foundation for further research. In their work, Do and Choi [9] proposed taking into account the context of the scene when searching for parking spaces. To improve the accuracy of detecting parking spaces that may be oriented at different angles, the authors propose using rotated bounding boxes (rotated anchor boxes). Their method combines analysis of the environment with refinement of the location of slots, and is accompanied by the creation of a new realistic dataset. The approach showed that the use of contextual information significantly improves the stability of algorithms in real operating conditions. An interesting direction was the application of graph neural networks.

Chen *et al.* [10] proposes a model with an attention mechanism, in which landmarks are treated as graph vertices. This approach allows for taking into account the relationships between elements in the scene, and improves the stability of the algorithm. The findings demonstrate the superiority of Graph Neural Network (GNN) architectures over traditional Convolutional Neural Networks (CNNs) and U-Nets in several tasks.

Ratti *et al.* [11] demonstrated that a WSN is used to monitor parking space occupancy. The system provides real-time data transmission, allowing drivers to quickly find available spaces and reducing congestion on the road network.

In Ref. [12], an intelligent parking system based on the Internet of Things was proposed, which provides real-time monitoring and booking of parking spaces using mobile (Android) and web applications. The system is focused on optimal use of parking space, reducing traffic congestion and improving resource management efficiency in the smart city concept.

In a number of studies, computer vision methods are used to solve the problem of parking space classification, for example, in Ref. [13], the You Only Look Once version 4 (YOLOv4) neural network is used to classify parking spaces based on images from 360-degree cameras. Modern studies actively use 360-degree cameras with fisheye lenses to detect parking spaces, but most studies focus only on determining their occupancy. At the same time, the issue of compatibility of a free space with its purpose (for example, a space for the disabled or electric vehicles) remains poorly studied. Xiao *et al.* [13] propose an adaptation of neural network models, such as YOLOv4 with polygonal bounding boxes, which allows for more accurate classification of parking zones of different types and shapes. In Ref. [14], Yamamoto *et al.* consider an Automated Valet Parking (AVP), which uses autonomous driving technologies for self-parking of cars. A model of vehicle movement using spatio-temporal reservation is proposed, which ensures efficient management of vehicle flows in the parking lot. In addition, some researchers consider the behavioral aspects and perception of drivers. Stranieri [15] analyzes drivers' subjective assessment of the ease of finding a free parking space in areas equipped with vehicle detection devices. The paper identifies factors influencing drivers' perceptions, such as search time and use of navigation aids, which are important for increasing user satisfaction. Liu *et al.* [16] considers autonomous parking problems with an emphasis on kinematic models and trajectory optimization algorithms. The study presents an intelligent parking planning algorithm based on traffic and driver behavior prediction. The model takes into account various factors, such as traffic density and user preferences, for optimal allocation of parking spaces. At the same time, an important problem in urban parking management is the limited number of free spaces and the need for efficient logistics within parking zones.

A number of studies have been devoted to the use of YOLOx for detecting vehicles in difficult weather conditions, such as rain, fog, snowfall, and sandstorms. In Ref. [17], Ashraf *et al.* applied multiscale Retinex to improve image quality, which significantly increased the effectiveness of the YOLOx detector. However, the paper does not clarify how well the model will generalise to other conditions (e.g., rain with headlight glare, heavy snowfall, a combination of weather factors), nor does it analyse false positive and false negative detections, which is particularly critical for safety.

Elfaki *et al.* [18] proposes an intelligent parking system that uses a combination of Passive Infrared (PIR) sensors, ultrasonic sensors, and cameras to monitor available spaces, as well as Internet of Things-based license plate recognition technology to control access and ensure the correct allocation of parking spaces. Experimental tests confirmed the applicability of the solution in real conditions, ensuring high recognition accuracy and acceptable data transmission and processing delays. Sari *et al.* [19] proposed a method for automated parking space detection based on a combination of a Grey Level Co-Occurrence Matrix (GLCM) and a Support Vector Machine (SVM) optimised using genetic algorithms (SVMGA). Experimental tests confirmed high accuracy: 96.99% for training data and 94.36% for test data, indicating the effectiveness of the method in solving parking management problems. In Ref. [20], an intelligent autonomous parking system (SAVP) based on the Internet of Things (IoT), fog computing, and blockchain technologies is proposed. In this system, fog nodes perform distributed processing, while a Lightweight Integrated Blockchain and Cryptography (LIBC) module ensures authorization and data protection. Experimental results demonstrate that the proposed system achieves high efficiency in terms of response time, confidentiality, and security, making it a promising solution for the development of scalable smart parking systems.

In addition, deep learning is used for effective parking space management in smart cities, in particular modified versions of the YOLO-v5 architecture for determining parking space occupancy. The lightweight models proposed in the work [21] provide high accuracy and real-time performance, demonstrating superiority in detecting vehicles of various sizes, including small objects, compared to the basic versions of YOLO-v5. Experimental results confirm the promise of this approach for autonomous parking management systems.

At the same time, modern research into intelligent parking solutions employs a variety of technological solutions and methods, which are discussed in the following works: methods for classifying vehicles and the You Only Look Once (YOLO) machine learning algorithm [22], ultrasonic sensors and Arduino microcontrollers [23], ultrasonic and infrared sensors [24], ultrasonic range sensors with machine vision algorithms [25], as well as unmanned aerial vehicles combined with the U-Net image segmentation method and deep sorting algorithm [26].

However, existing studies have identified significant challenges in automating and managing unstructured parking lots, especially where markings are not visible or completely absent due to adverse weather conditions. In Ref. [27], Wu *et al.* noted that automated parking is complicated by the lack of publicly available labeled Panoramic Surround-View (PSV) data and the difficulty of reliably identifying parking spaces and road structures. To address these challenges, the authors created their own PSV dataset and proposed a modified Highly Fused Convolutional Network (H-HFCN), which provides more accurate segmentation of linear markings through vertical and horizontal convolutional paths. Experimental results demonstrated the high efficiency of this approach in identifying parking spaces and traffic lanes. In Ref. [28], Li *et al.* proposed VPS-Net for around-view images to address the problem of a complex visual environment (changes in lighting, shadows, and view limitations). The method combines multi-object detection and classification, achieving high accuracy and minimal localization error, and it is experimentally shown that VPS-Net can detect various vacant parking spaces with 99.63% accuracy and 99.31% recall on the PS2.0 dataset. Jian *et al.* [29] used end-to-end semantic segmentation to detect parking space lines and corners by feeding the entire image for segmentation instead of just individual elements. This approach shows that the recall, precision, and F-measure of the proposed method are 92.94%, 99.40%, and 96.06%, respectively, which are higher than those of existing state-of-the-art end-to-end training methods. Existing Around View Monitor (AVM) parking space detection methods suffer from distortions and occlusions, which leads to errors in spot occupancy detection. To solve this problem, Li *et al.* [30] proposed a comprehensive method combining two deep convolutional neural networks (Visual Place Recognition Network (VPS-Net) and Deep Multi-Phase Regression with Pose Supervision (DMPR-PS)) and Kernel Correlation Filter (KCF)-based tracking algorithms, which provides more accurate and stable detection of vacant parking spaces. Park and Ahn [31] proposed the Parking Slot Detection and Tracking (PSDT) algorithm based on the PSDT-Net neural network and DeepSORT. The authors showed that their approach helps to eliminate the limitations of methods using only corner points, allowing accurate detection and tracking of parking spaces even with partial overlap and in the presence of several neighboring spaces.

Studies show that the growth of urban populations leads to increased traffic congestion and a shortage of parking spaces, which significantly complicates mobility and increases the time spent searching for parking. In Ref. [32], an intelligent parking management system is proposed, integrating IoT sensors, cloud infrastructure, and Long Short-Term Memory (LSTM) model to predict parking space occupancy. This solution allows drivers to be informed in advance about available spaces and helps reduce the load on the road network. Kurek and Macioszek [33] examines the use of vehicle detection devices in paid parking areas to assess the subjective ease of parking by drivers. For the analysis, data on the use of parking spaces and the results of driver surveys were used, on the basis of which a logit model was built. The analysis of the model showed that dynamic parking information (variable $x_{DPIBIGA}$) significantly contributes to a positive subjective assessment by the driver of the ease of finding an empty parking space in an equipped area. Elbanhawi and Simic [34] notes that parallel parking for a nonholonomic vehicle in a confined space is a difficult task due to the limitations of the trajectory curvature. The use of B-spline curves for automatic parking is considered, which makes it possible to build smooth and comfortable driving paths. The use of the unique properties of B-splines ensures the synthesis of trajectories with continuous curvature, increases passenger comfort and allows vehicles to park in narrow spaces by increasing the permissible maneuver radius. Modern automated parking algorithms face the challenge of ensuring high performance and safety when planning trajectories in real time, especially when using Bird's-Eye View (BEV) [35]. The study presents an improved algorithm that effectively solves this problem by planning safe trajectories in real time. To achieve this goal, an algorithm based on A* has been developed that integrates kinematic vehicle models, optimized heuristic functions, bidirectional search, and Bezier curves. This approach makes it possible to reduce the calculation time and increase the comfort of parking [35].

We have analyzed the existing approaches to developing smart parking systems (Table I).

TABLE I. COMPARATIVE TABLE OF SMART PARKING SOLUTIONS

№	Source	Methodology/Algorithms	Architecture/Technology	Quality Metrics
1	[4]	Message Queuing Telemetry Transport (MQTT) for data transmission	IoT, mobile application	Response time, data accuracy
2	[11]	WSN for parking monitoring	Wireless sensor networks	Reliability of data transmission
3	[13]	YOLOv4 for classifying places	All-round view cameras, CNN	Precision, Recall, F1-score
4	[14]	Spatial and temporal redundancy	Motion simulation, AVP	Space efficiency
5	[32]	LSTM for seat occupancy prediction	Arduino, IoT, Web application	Forecasting accuracy
6	[33]	Binomial logit model	Analysis of driver surveys	R^2, prediction accuracy
7	[34]	B-splines for trajectory planning	Motion simulation	Smoothness of the trajectory
8	[35]	A* with Bezier optimization	BEV, kinematic models	Calculation time, route safety

Table I shows that different methods are used for parking monitoring: MQTT and WSN provide data collection and transmission from sensors, and spatial and temporal redundancy increases the reliability of information. YOLOv4 and the binomial logit model are used to classify and evaluate parking occupancy, and LSTM allows you to predict its dynamics over time. B-splines and A* with Bezier optimization are used to plan the movement of cars, creating smooth and optimal trajectories. Each method has its own strengths and limitations: accuracy and speed, resource requirements, and the need for a parking map. The table summarizes key approaches to parking monitoring, prediction, and trajectory planning across various studies, highlighting the methodologies, technological architectures, and evaluation metrics used. Together, these works illustrate a diverse technological landscape focused on improving accuracy, efficiency, and reliability in intelligent transportation systems.

III. MATERIALS AND METHODS

A. Concept and Life Cycle of an Automated System

In this study, we consider the automation of "smart parking" based on the following factors: control of parking access according to a certain list of "permitted" car numbers, lack of parking markings including due to snow cover), limited total parking area.

Fig. 1 presents a generalized lifecycle scheme, emphasizing the key pillars of system design, implementation, and deployment. Each phase should begin with the definition and collection of requirements. At this stage, it is essential to determine the objectives, establish key constraints, and consider the architectural aspects of the solution. During the implementation phase, it is also necessary to collect and annotate the source data, which is required for creating the training set for a multi-layer segmentation neural network model. The model, specialized in the automated detection of parking spaces, is then trained and validated step by step. Upon successful training, the segmentation is performed, followed by projection into world coordinates using camera parameters obtained during the calibration stage. The resulting data is integrated to form a comprehensive map of parking space utilization. The final stages include integration, full functional testing, and deployment, enabling the system to be flexibly updated, modified, and adapted as needed.

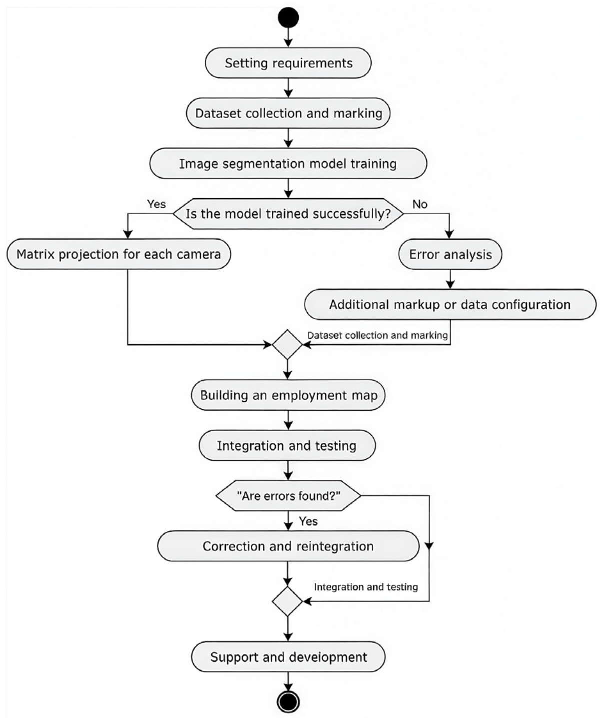


Fig. 1. Workflow diagram of the parking space analysis system development.

The core idea is to construct a parking occupancy map by reconstructing a top-down view of the parking lot using multiple camera images. Instead of complex camera calibration, a manual configuration approach is applied: for each camera, four corner boundaries of the observed area are manually defined in the image, along with the corresponding real-world distances between those points.

Next, the vehicle images are projected onto a plane parallel to the ground surface. Four video cameras are mounted at different angles. Based on the images captured from these cameras, projection regions are identified. During the projection construction phase, the presence of an object (a vehicle) is determined using predefined coordinates.

B. Comparative Model Analysis and Selection

To justify the selection of the U-Net architecture as the baseline model for the segmentation of unlabeled parking spaces, a preliminary methodological analysis was conducted. The aim of this stage was to evaluate the effectiveness of the proposed approach in comparison with existing solutions, such as visual models of the YOLO family (YOLOv5/YOLOx), DeepLabV3+, and Mask R-CNN, under specific operational constraints.

The experimental testing plan involved using a dataset of 12,792 images to determine the optimal balance between segmentation quality and computational complexity. Within the methodological framework, the following metrics were compared: Precision, Recall, F1-score, IoU, and processing speed.

The comparative analysis (the results of which are presented in Figs. 2–4 to support the selection) showed that although DeepLabV3+ and Mask R-CNN demonstrate high accuracy performance, their practical use is limited by low inference speed (68.7 ms and 127.4 ms, respectively). In the context of real-time tasks, a critical factor was the ability of U-Net to provide competitive segmentation quality (F1-score: 72.29%, IoU: 56.56%) while maintaining significantly higher processing speed (42.3 ms). This configuration ensures compliance with the overall system latency limit (169 ms for 4 cameras), which became the determining criterion for selecting U-Net as the primary model in this study.

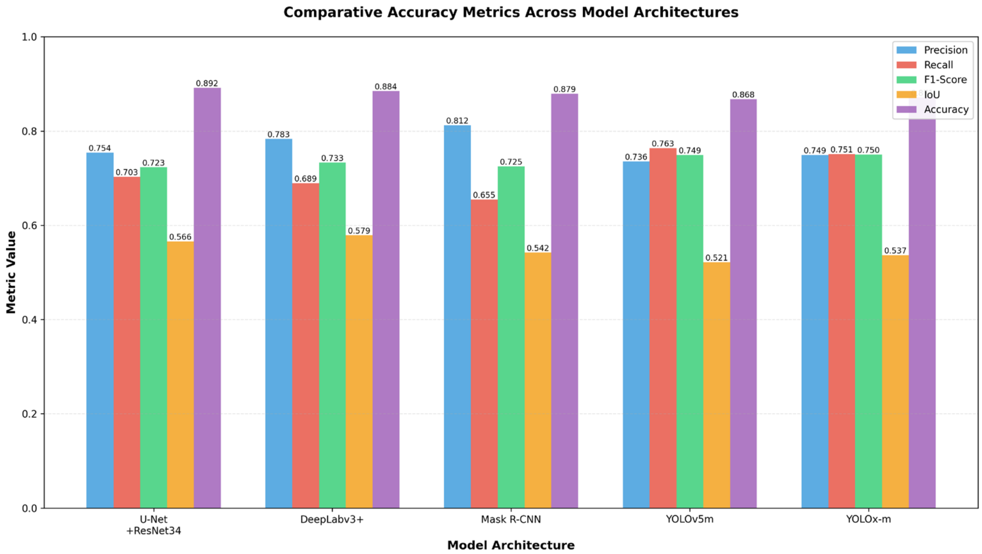


Fig. 2. Analysis of accuracy indicators for different models.

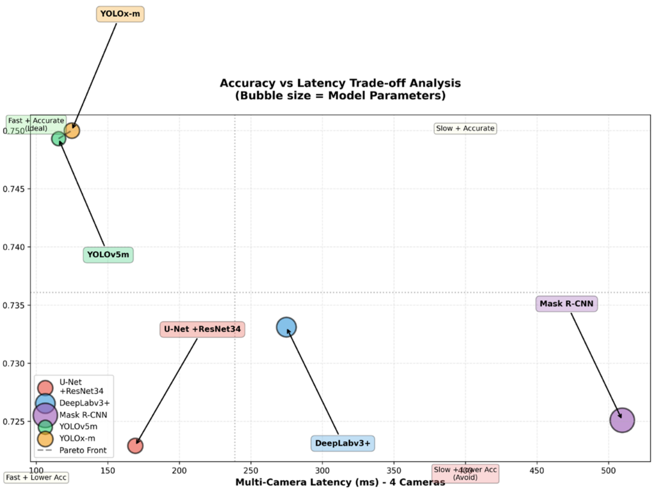
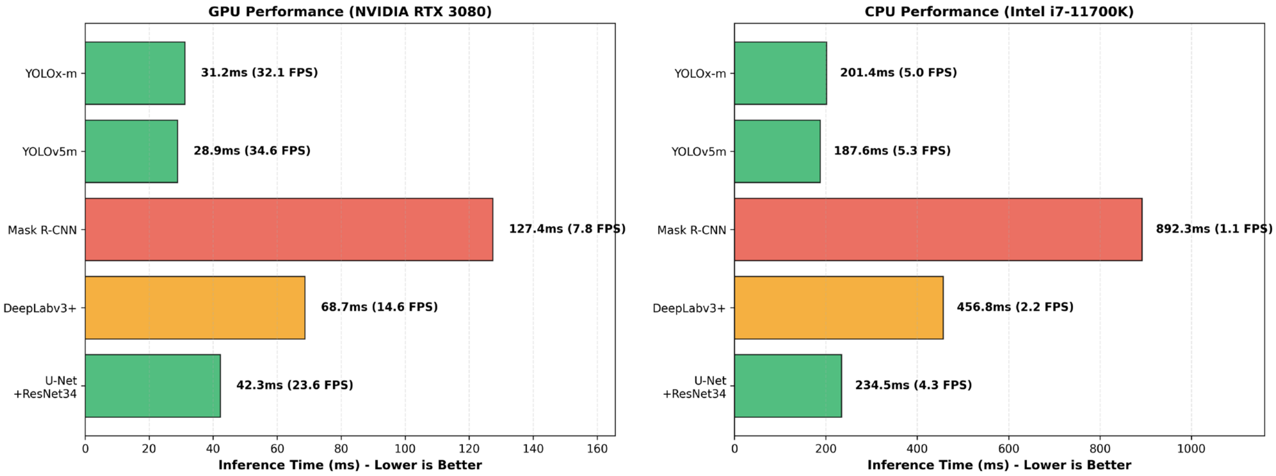


Fig. 3. Model quality analysis using the accuracy-latency trade-off curve.



(a) (b)

Fig. 4. The processing time of one image in milliseconds: (a) GPU Performance (NVIDIA RTX3080); (b) GPU Performance (Intel i&-11700K).

C. Training Configuration and Evaluation Metrics

Our performance benchmarking (Fig. 4) demonstrates practical near real-time feasibility with U-Net achieving 42.3 ms single-image inference on Graphics Processing Unit (GPU) (23.6 Frames Per Second (FPS)), 169.2 ms total latency for concurrent 4-camera processing, and 1.8 GB peak memory usage, while comparative analysis shows YOLOv5m offers fastest performance at 28.9 ms per image (115.6 ms for 4 cameras, 8.7 FPS).

Our comprehensive evaluation on a 7217-image dataset demonstrates that U-Net achieves an optimal balance between accuracy (F1 = 72.29%, IoU = 56.56%) and real-time performance (42.3 ms single-image, 169.2 ms for 4-camera concurrent processing with 1.8 GB memory), outperforming DeepLabv3+ and Mask R-CNN in inference speed while maintaining competitive segmentation quality. Comparative benchmarking against five state-of-the-art models (DeepLabv3+, Mask R-CNN, YOLOv5m, YOLOx-m) across multiple metrics validates U-Net's practical feasibility for near real-time parking monitoring applications.

D. System Architecture and Workflow

Fig. 5 shows the detailed architecture of the U-Net11 neural network, which performs vehicle segmentation. Image processing consists of several sequential steps. The network architecture includes two main components: the encoder (contracting path) and the decoder (expanding path). The encoder comprises four sequential blocks, each containing two convolutional layers with ReLU activation functions, followed by a MaxPooling operation. At the deepest level of the network, there are two convolutional layers with ReLU activation, forming a compact latent representation of the analyzed scene.

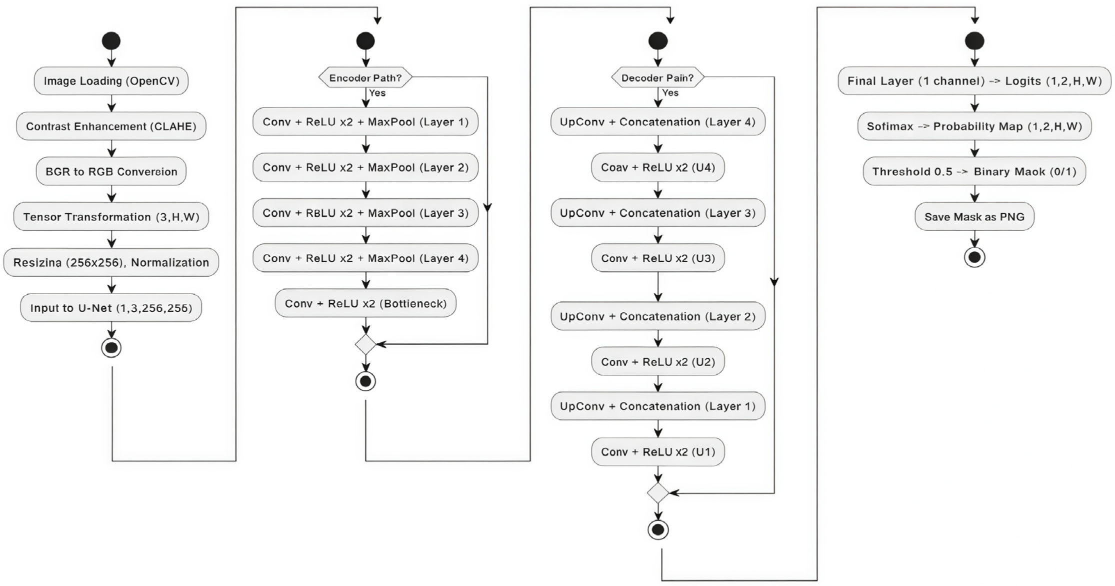


Fig. 5. U-Net11 architecture for vehicle segmentation from multiple cameras.

The decoder mirrors the encoder and also consists of four blocks: a transposed convolution (up-convolution) layer, concatenation with the corresponding feature map from the encoder via skip connections, and two convolutional layers with ReLU activations for processing the combined features.

The segmentation network architecture, UNet11, includes an encoder (downsampling path) made up of four sequential convolutional blocks (each consisting of two Conv2d layers and a MaxPool2d operation), and a decoder. This structure ensures a progressive reduction in spatial resolution (256→128→64→32→16 pixels) while simultaneously increasing the depth of feature representations.

Segmentation quality is evaluated by comparing the masks generated by the model with manually annotated validation masks. For this purpose, a dataset was prepared containing images of vehicles with annotated contours, which ensures the reliability and accuracy of the ground truth data. The configuration settings specify the geometric dimensions of a single parking slot and the overall parking area.

To take into account information from different sources (4 cameras), an adjacency matrix is introduced that defines for each cell (i, j) a set of links Gamma (i, j).

To implement the system, it is necessary to define appropriate approaches for each individual module. Let us consider the solutions adopted for the image pre-processing module. In this study, an image segmentation subsystem based on a modified U-Net11 architecture was developed to extract target objects from the background (Fig. 5). The model uses a U-Net variant with a pre-trained encoder based on the Visual Geometry Group 11-layer network (VGG11) architecture with ReLU activations (UNet11/TernausNet). Unlike the original U-Net, the encoder is replaced with a VGG11 network pre-trained on ImageNet, which removes the need to learn low-level features from scratch and significantly accelerates model convergence.

As an additional preprocessing step, the CLAHE method is applied to enhance contrast, allowing for the detection of details in both dark and bright areas without introducing saturation artifacts. The U-Net11 architecture consists of two symmetrical parts: a contracting path and an expanding path, connected by direct skip connections. These connections transmit detailed spatial features extracted in the early stages of processing directly to the decoding part. The contracting path applies a series of convolutional operations followed by pooling operations, reducing the spatial dimensions of the input data while preserving the most significant features. Pooling operations help lower computational complexity and retain essential characteristics. The skip connections compensate for information loss due to pooling and ensure the preservation of local image features. In the expanding path, spatial dimensions are restored using upsampling or transposed convolutions, where features from each level of the contracting path are concatenated with corresponding levels in the expanding path.

The first step in developing a parking monitoring system is to create a matrix parking model. The parking lot in question has dimensions of 55×25 m. According to ST RK 2476-2021, the dimensions of a parking space for a passenger car are 5.50×2.5 m. Based on these parameters, the parking lot was represented as a binary matrix with a dimension of 110×50, where each cell of the matrix corresponds to an area of 0.5 m. There were 4 cameras installed in the parking lot and the binary matrix view from each camera is shown in Fig. 6.

(a) (b) (c) (d)

Fig. 6. Image from cameras and their matrix markup: (a) Camera 1; (b) Camera 2; (c) Camera 3; (d) Camera 4.

E. Unified Architecture of the Intelligent Parking Monitoring System

As part of the methodological framework of the study, a unified architecture of the intelligent parking monitoring system was developed (Fig. 7). The proposed architecture integrates data acquisition, computational processing, and operator-level visualization into a single end-to-end workflow. The system comprises the following functional components.

1. Field infrastructure

This layer provides the primary data sources for subsequent analytical processing. It includes:

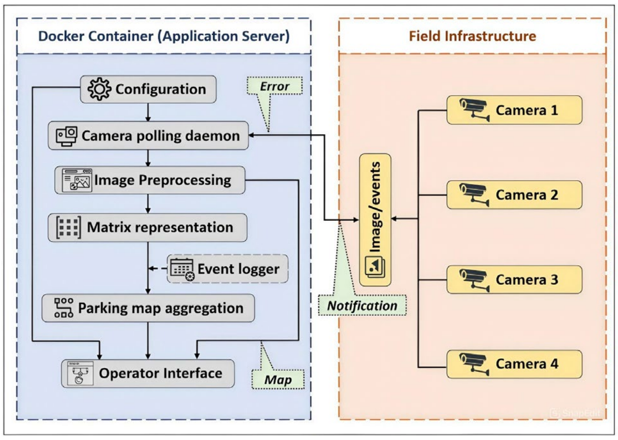


Fig. 7. General hardware block diagram of the parking system.

(a) A multi-camera video surveillance subsystem consisting of four cameras positioned at different angles to ensure full visual coverage of the monitored parking area;
(b) A network switch responsible for routing video streams and control signals between the cameras and the central server.

2. Server-side component (containerized application)

The computational core of the system is deployed as an isolated Docker container that encapsulates all processing modules and ensures reproducibility, scalability, and controlled execution. The server component includes:

(a) Data acquisition subsystem, which performs scheduled or event-based polling of cameras and forwards captured frames to the processing pipeline;
(b) Logging subsystem, responsible for the continuous recording of system events, errors, and exceptional conditions, as well as the generation of operator notifications in case of critical situations;
(c) Operator interface, providing real-time visualization of the system state, including an interactive parking occupancy map and an event monitoring dashboard;
(d) Image pre-processing module, which applies noise-reduction filters, geometric and radiometric corrections, and other enhancement operations to improve the quality of the input video stream;
(e) Matrix representation module, responsible for converting video frames into a structured matrix format suitable for algorithmic analysis;
(f) Data aggregation module, which fuses information from multiple camera sources and generates a unified parking occupancy map;
(g) Configuration database, storing system-wide parameters such as network addresses, processing modes, and acquisition schedules.

3. Information flows

The operation of the system is driven by a set of well-defined data flows:

(a) Transmission of video streams and event signals from field cameras through the network switch to the data acquisition subsystem;
(b) Retrieval of configuration parameters by the acquisition subsystem and the operator interface;
(c) Sequential processing of video frames by the pre-processing module and the matrix-construction subsystem;
(d) logging of operational events and transmission of alerts to the operator interface when necessary;
(e) visualization of aggregated and processed data in the operator interface to support real-time monitoring.

The presented architecture provides efficient processing of video streams, the formation of an employment map and comprehensive monitoring of the system status with prompt notification of emergency.

IV. Results and Deployment

The key idea is to build a parking map by reconstructing the top view as cameras are installed at different points of the parking lot (Fig. 8). Fig. 8 illustrates the integration of footage from four cameras: Camera 2 covers the main parking area, providing a baseline overview; Cameras 1 and 3 overlap adjacent zones, reducing blind spots and enhancing detail; Camera 4 is directed at a hard-to-reach section of the lot, enabling accurate occupancy detection in that area and improving overall monitoring quality.

To evaluate segmentation quality, a system was developed to compare model-generated masks with manually annotated masks. A dataset containing images with predefined vehicle contours was collected to ensure the availability of accurate ground truth data for analysis. The algorithm compares the two masks pixel by pixel, with the manual annotation serving as the reference. Each pixel is checked for consistency with the ground truth, and any discrepancies are recorded as deviations. The accuracy percentage is calculated based on the ratio of matching pixels to the total number of pixels, providing an objective measure of the model's segmentation performance and identifying areas requiring further optimization.

After obtaining the binary segmentation mask, a comprehensive processing pipeline is applied to generate an "occupancy matrix". Based on predefined camera parameters, including image paths, mask paths, and the coordinates of the parking area corners (P1, P2, P3, P4), the original image and the corresponding binary mask are loaded. To simplify mathematical computations and facilitate cell-based analysis, a perspective transformation is performed: a transformation matrix is computed using predefined points on the original image, projecting it into a top-down view. As a result, the parking area is represented as a two-dimensional rectangular grid where each point has well-defined coordinates corresponding to real-world dimensions (Fig. 8).

There are parking devices in the Parking system, therefore, on the basis of specific calculations, the parameters of parking spaces, fixing labels and values of the length, width of car spaces and the ratio of their sizes to two and one originals are set in advance in the system, as well as the parameters of the homography of the label to the known co-ordinates of the camera installation sites, which are linked to the set values. To combine data from different cameras, an adjacency matrix is applied to the upper parking plan (Fig. 8(e)). Each of its cells in the top view is filled in based on the cell detector and the values for the camera matrices are included based on the matrix projection of the cells of the detector positions of the correct view from the cameras. This configuration allows you to define the visibility areas covered by several cameras and establish relationships between objects in the observation zones of auxiliary cameras. Optimization of setup parameters allows minimizing the costs of system calibration and calculation of projection positions for binding to the initial coordinates. This approach eliminates the need for complex technical calibration of cameras, while providing complete information about the state of the parking space without performing precise measurements of the parameters of the video surveillance system. Within the proposed methodology, the parking zone is represented as a rectangular grid (matrix), where each cell corresponds to a small area of space.

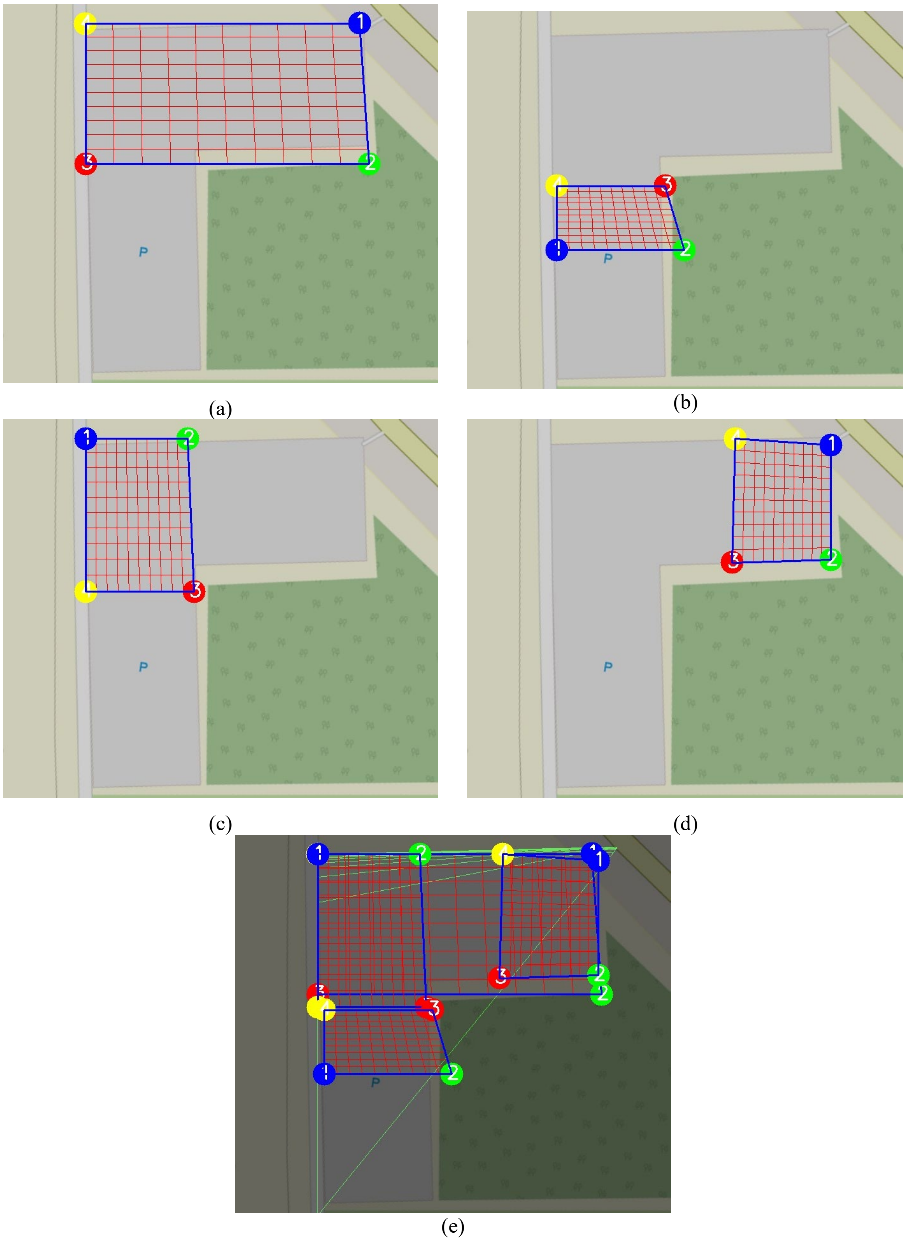


Fig. 8. Location of cameras in a real parking space: (a) manual camera calibration 1 (parking area P1); (b) manual camera calibration 2 (parking area P2); (c) manual camera calibration 3 (parking area P3); (d) manual camera calibration 4 (parking area P4); (e) intersection of calibrated grids from four cameras.

The reverse process of projection or matrix projection reworking is performed using matrices that are defined to be calculated from dimensions corresponding to the boundaries of the existing image and the actual points within the parking lot.

Instead of complex camera calibration, a manual configuration is used: for each camera, four angular boundaries of the observation zone in the image and the corresponding real distances between points on the ground are set. In the work, we performed an analysis of the displacement of all four chambers over the course of one month (Fig. 9), which varies from –5 to +11, which is not critical for our study.

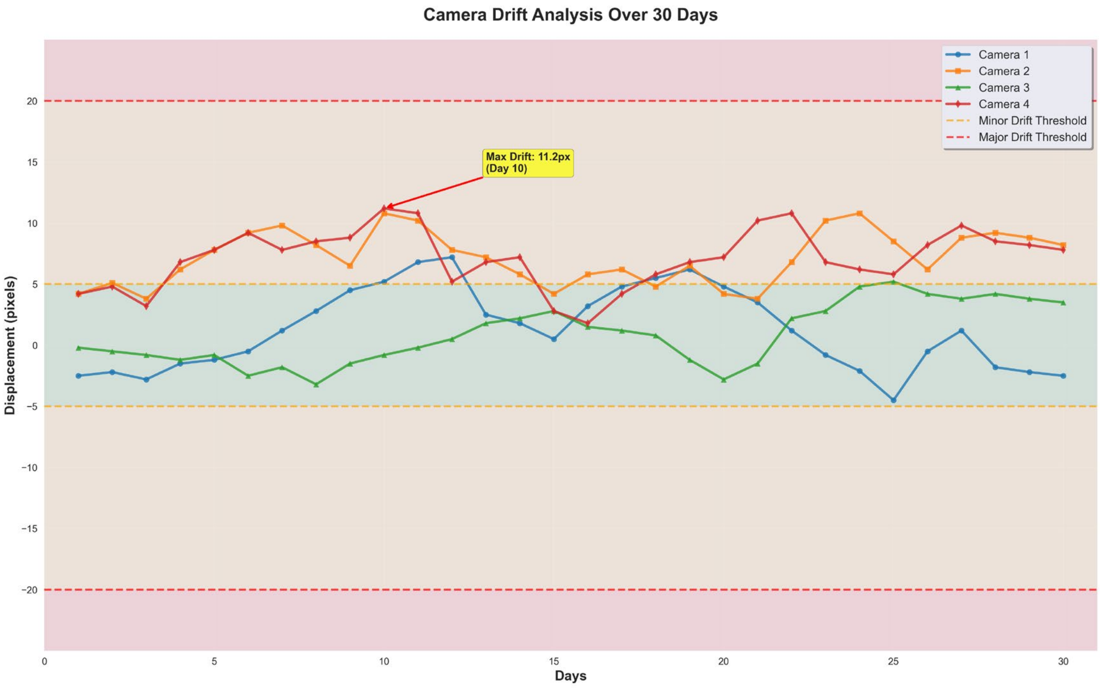


Fig. 9. Diagram of the displacement of four chambers during a month.

In the current implementation, based on manual configuration of parking zone markings, camera parameters (installation height, viewing angle, focal length, etc.) are not directly taken into account in the calculations, since all information about the surveillance zones is specified by the user. However, the possibility of subsequent implementation of an automatic calibration mechanism based on the metadata of the shooting equipment is being considered, which will allow dynamically adapting the size and position of the cells depending on the settings and characteristics of a particular camera.

The next step is classification based on the U-net11 model for car recognition. To conduct experiments in a real parking environment, a specialized sample was developed, including 12,792 images (Table II).

TABLE II. IMAGE SET FROM FOUR CAMERAS

Name	Camera 1	Camera 2	Camera 3	Camera 4
Total number of images	3275	3109	3415	2993
Day	2227	2300	2186	1856
Night	1048	808	1229	1137
Images with various defects	25	26	39	25

We analyzed model robustness across weather conditions showing U-Net maintains 75.42% precision under clear weather with degradation to 63.07% in rain (−16.4%) and 57.29% in snow (−24.0%), demonstrating acceptable performance boundaries (Fig. 10).

Images obtained from cameras, as well as from an external source [36], were used as a dataset for training models. The total number of images was 12,792 units, which were divided into training and validation sets in a 60:40 ratio. Training employed Adam optimizer ($lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight_decay = 1×10^{-5}) over 36,050 iterations (50 epochs × 721 iterations/epoch, batch_size = 8) with ReduceLROnPlateau scheduling strategy featuring 5-epoch warmup ($1\times10^{-6}\rightarrow1\times10^{-3}$) and three plateau-based reductions (factor = 0.5, patience = 5), achieving full reproducibility.

This ratio was chosen in order to form a more representative test sample, which is critical when the dataset size is limited and the scenes are highly variable. This approach reduced the risk of overestimating quality due to information leakage between training and testing. The use of k-fold validation was considered, but due to the high computational cost of training a deep model and limited resources, a decision was made in favour of static division with an increased test share.

This study was conducted using the example of a university parking area, where the main users are students and staff, which led to a predominance of passenger cars in the image dataset. This distribution reflects the specifics of the site under study, but limits the generalisability of the model when applied in urban infrastructure (e.g., airport, railway station, and shopping centre car parks), where trucks, buses, and motorcycles are also present. In the future, we plan to expand and balance the dataset by including images of various types of vehicles, which will increase the versatility and applicability of the proposed approach. The dataset was also collected during both daytime and nighttime (Table II), which made it possible to take into account complex conditions (glare, overexposure, underexposure) during training.

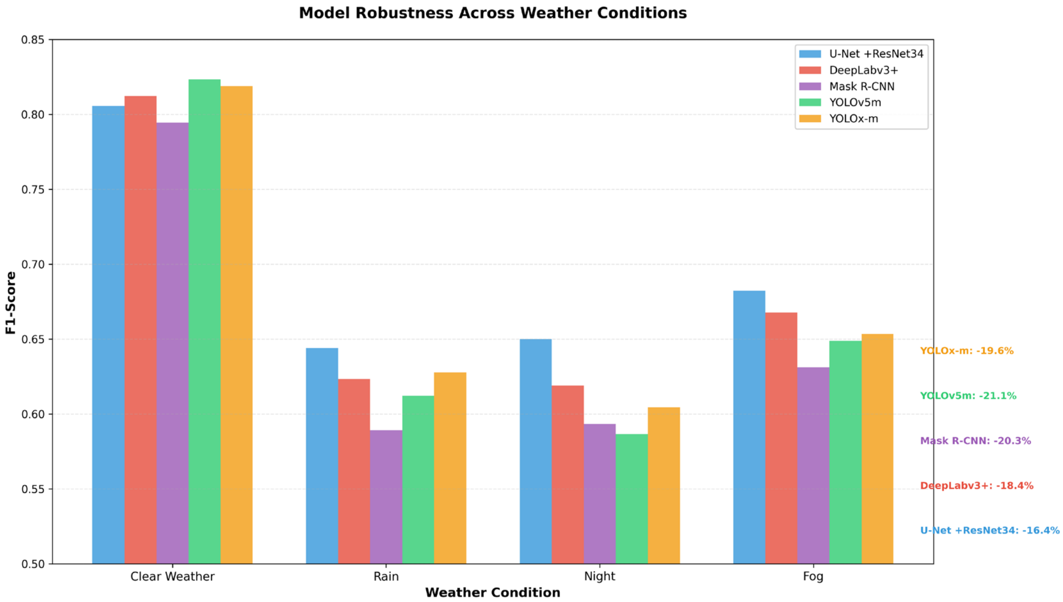


Fig. 10. Analysis of the performance of models for different weather conditions.

Before an image is processed by the model, it undergoes a series of preprocessing steps. Initially, the image is loaded in Blue, Green and Red (BGR) format using OpenCV, and an error is generated if loading fails. To enhance object visibility, the Contrast Limited Adaptive Histogram Equalization (CLAHE) method is applied. Subsequently, the image is converted from BGR to Red, Green, Blue (RGB), aligning with PyTorch's standard transformations, and transformed from a NumPy array into a tensor with reordered dimensions (3, H, W). The image is then resized to the required spatial dimensions using the Resize function, and pixel values are normalized by dividing by 255.0. Finally, a batch dimension is added to the tensor, forming the final format (1, 3, H, W), which is placed either on the GPU Compute Unified Device Architecture (CUDA) or CPU, depending on the system configuration.

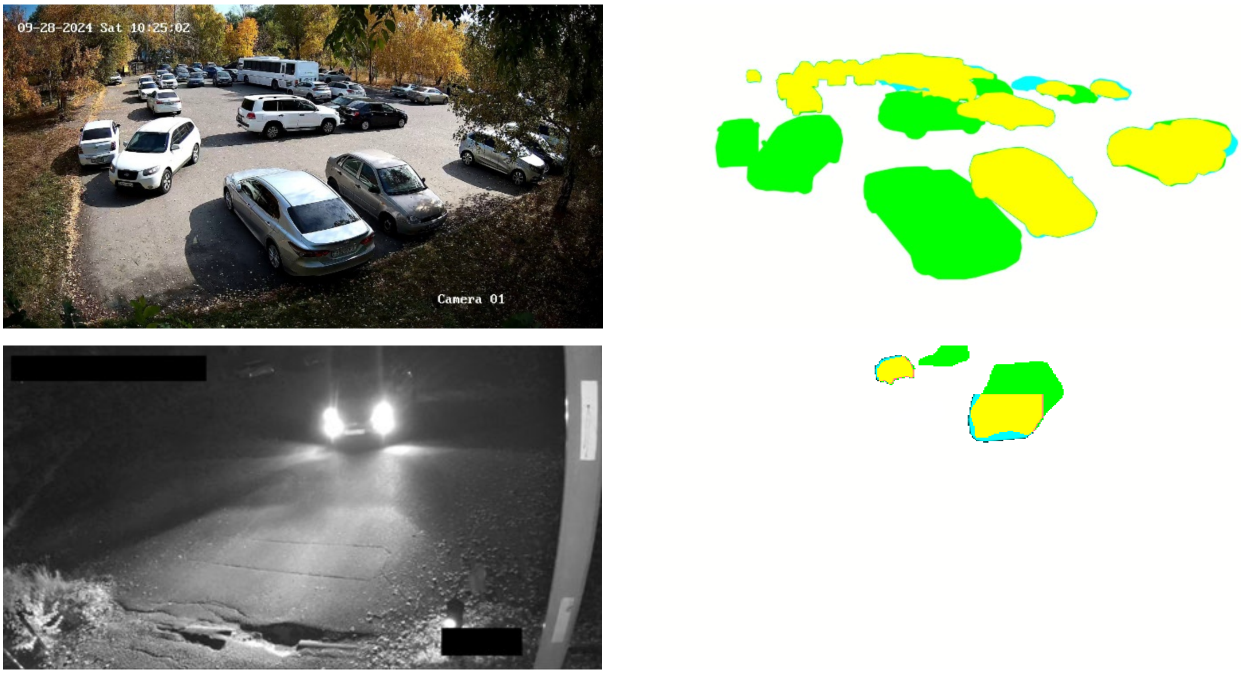
The system takes into account extreme lighting conditions (glare from wet surfaces, deep shadows) through several mechanisms.

First, the use of a pre-trained encoder allows the model to rely on features that are more resistant to lighting than when training from scratch. Second, during the data preparation stage, augmentations (random changes in brightness, contrast, gamma) are applied to simulate real-world conditions and reduce sensitivity to glare and shadows. Additionally, input image normalisation methods and adaptive post-processing (e.g., morphological operations or filters on a probability segmentation map) help.

However, it is impossible to completely eliminate the influence of extreme conditions. Accuracy may decrease in areas of strong highlights or dense shadows.

Within the model, input data passes through a sequence of convolutional blocks in the contracting path, where each block reduces the image resolution while increasing the number of channels. At the final level of the contracting path, the most abstract and global features are accumulated. The expanding path then restores the original resolution using upsampling or transposed convolutions, concatenating features from the contracting path at each level. The network output generates a segmentation map of size (1, 2, H, W), where two channels correspond to the "background" and "object" classes. During post-processing, the obtained values are normalized using the Softmax function, forming a probabilistic distribution along the channel axis. To obtain a binary mask for a specific class, the corresponding channel is selected and threshold binarization is applied (e.g., with a threshold of 0.5). This results in a binary mask where pixels with a value of 1 correspond to the detected object. For visualization and storage, the final mask is multiplied by 255, converted into a NumPy format, and saved as a PNG or other image file. Examples of the model's performance are shown in Fig. 11, the original image is on the left, and the post-processed visualization is on the right, where the reference (expected) mask and the mask predicted by the model are combined. The expected area is marked in green, the result matches the reference in yellow, and incorrect predictions are highlighted in turquoise (Fig. 11).

The analysis revealed some limitations of the model related to the underrepresentation of certain vehicle classes in the sample, such as trucks, motorcycles, and buses. This led to distortions in the image segmentation process. The result was significantly affected by the clarity of the vehicle contours. In particular, at night, the light from headlights, dash cam indicators, and other light sources led to distortions in the model's predictions. Observations indicate the need for further expansion and balancing of the dataset, as well as improved image processing.



(a) (b)

Fig. 11. Examples of images and object recognition: (a) the original image; (b) ground-truth (expected) mask and model-predicted mask.

When deploying the system in practice, parameters such as processing latency, computational costs, and real-time feasibility are also important. The proposed approach is implemented as a modular architecture with Docker containerisation, which provides deployment flexibility and scalability. However, the computational costs associated with training and using the U-Net segmentation model may vary depending on the hardware configuration. In experiments, the system demonstrated the ability to operate in near real time when processing images from surveillance cameras, but to ensure fully real-time performance under high load conditions, model optimisation or the use of specialised hardware acceleration (GPU) may be required.

A software module for implementing an image segmentation algorithm using a convolutional neural network based on a modified UNet11 architecture was developed. The module includes functions for loading a pre-trained model, performing image preprocessing using CLAHE to enhance contrast, and converting an input image into a tensor with subsequent scaling and normalization [37–40]. Segmentation is performed by calculating a probability distribution by two classes ("background" and "object") with subsequent binary threshold transformation, which allows obtaining the final binary mask. In addition, a function for assessing the quality of segmentation using the Intersection over Union (IoU) metric, comparing the predicted mask with the reference markup, is implemented.

This approach provides an objective assessment of the segmentation accuracy and can be used in computer vision systems to solve automated image analysis problems. To convert real sizes into pixel coordinates, coefficients obtained based on the dimensions of the image with a top view are used. Each matrix cell is estimated by the number of white pixels in the corresponding fragment of the binary mask. If a cell goes beyond the image or contains less than 20 pixels, it is assigned a value of −1, which indicates that the estimate is incorrect. At the same time, if the ratio of white pixels to the total number of pixels in a cell exceeds a specified threshold (OCCUPANCY_THRESHOLD), the cell is marked as occupied (1); otherwise, as free (0). This technique provides an accurate representation of the occupancy of parking spaces for subsequent analysis and management decisions.

One of the key aspects of the system is the process of combining data obtained from different cameras to improve the accuracy of determining occupied parking spaces. Each camera records the parking space at its own angle, which allows you to cover different parking zones (Fig. 12), compensate for possible blind spots and minimize perspective distortions.

Next, the occupancy matrix is constructed. The configuration specifies the physical dimensions of a parking slot and the overall parking area dimensions. Based on these values, the number of grid cells along the horizontal and vertical axes is computed by di-viding the total parking area dimensions by the slot size. A meter-to-pixel conversion is applied to determine the precise coordinates of each grid cell in the mask. Each cell is analyzed by counting the number of white pixels within its boundaries, and its occupancy status is determined based on a predefined threshold. If the proportion of white pixels exceeds the threshold, the cell is marked as "occupied" (1). If the cell is entirely within the mask but the proportion of white pixels is below the threshold, it is marked as "free" (0). If a cell partially extends beyond the mask boundaries or its size is insufficient for accurate evaluation, it is assigned a value of −1, indicating an inability to determine its status reliably. The final output is a two-dimensional array, where each position represents a cell in the parking grid.

(a) (b)

Fig. 12. Determination of parking space from different cameras: (a) the original image; (b) mask corresponding to the image.

In the proposed methodology, each parking grid cell is described not only by its binary occupancy status (whether a vehicle is present) but also by its relation to corresponding elements in multiple auxiliary matrices. To define these relationships, an adjacency matrix is introduced, which determines how elements from other matrices (or the same matrix at different indices) influence the final occupancy decision for each cell [41, 42].

We have a set of binary matrices $A_1, A_2, \ldots, A_n$, each of which reflects information about the parking status. The element $A_k(i, j) = 1$ if a vehicle is found at position (i, j) of matrix k, and 0 otherwise.

For each cell (i, j) of the resulting matrix T, a set of connections $G(i, j)$ is specified, consisting of three elements (k, p, q), where: k is the index of the matrix A_k; (p, q) are the coordinates in this matrix that logically correspond to (i, j). The construction of the resulting matrix T is formulated as follows:

$$\mathrm{T}(i,j)=f(x)=\begin{cases}1, if\ \forall(k,p,q)\in\Gamma(i,j), A_k(p,q)=1,\\ 0, else\end{cases} \quad (1)$$

We "sum" the binary values in adjacent cells defined by the matrix, but we are only interested in the situation when all matrix values are equal to 1:

$$\sum\nolimits_{(k,p,q)\in(i,j)} A_k\,(p,q) = |\Gamma(i,j)| \quad (2)$$

where $\Gamma(i, j)$ is the number of references for cell (i, j).

If the equality is satisfied, then all the specified matrices and indices confirm the fact of occupancy, and the final value $\mathrm{T}(i, j)$ is set equal to 1. In any other case (at least one of the reference cells is equal to 0), the final value will be 0.

V. Discussion

In the study to determine the adequacy of the dataset, we processed false positive and false negative data (Fig. 13) for the final parking occupancy matrix.

Fig. 13 shows the evaluation of the parking system based on error matrix analysis. The normalised error matrix shows (Fig. 13(a)) that the model correctly classifies the background (True Negative (TN) = 99.9952%), but some vehicles remain unrecognised (False Negative (FN) = 184,520). Fig. 13(b) shows the accuracy metrics: the proportion of false positives (0.9%) is significantly below the threshold level (2%), while the proportion of false negatives (29.7%) exceeds the specified threshold (25%). True positives account for 70.3%, and true negatives account for 99.1%. The distribution of errors by camera is shown in Fig. 13(c), the false positive rate ranges from 0.76% to 1.15%, with the highest false negative rate recorded on Camera 3 (35.7%). An analysis of the reasons for difficult positive classification was also performed (Fig. 13(d)), revealing shadows from trees (34.2%), parking markings (28.7%), rubbish bins (15.4%), people (12.3%), and reflections on water (9.4%). Thus, the system demonstrates high accuracy in background detection, but requires further development in terms of reducing the number of missed vehicles.

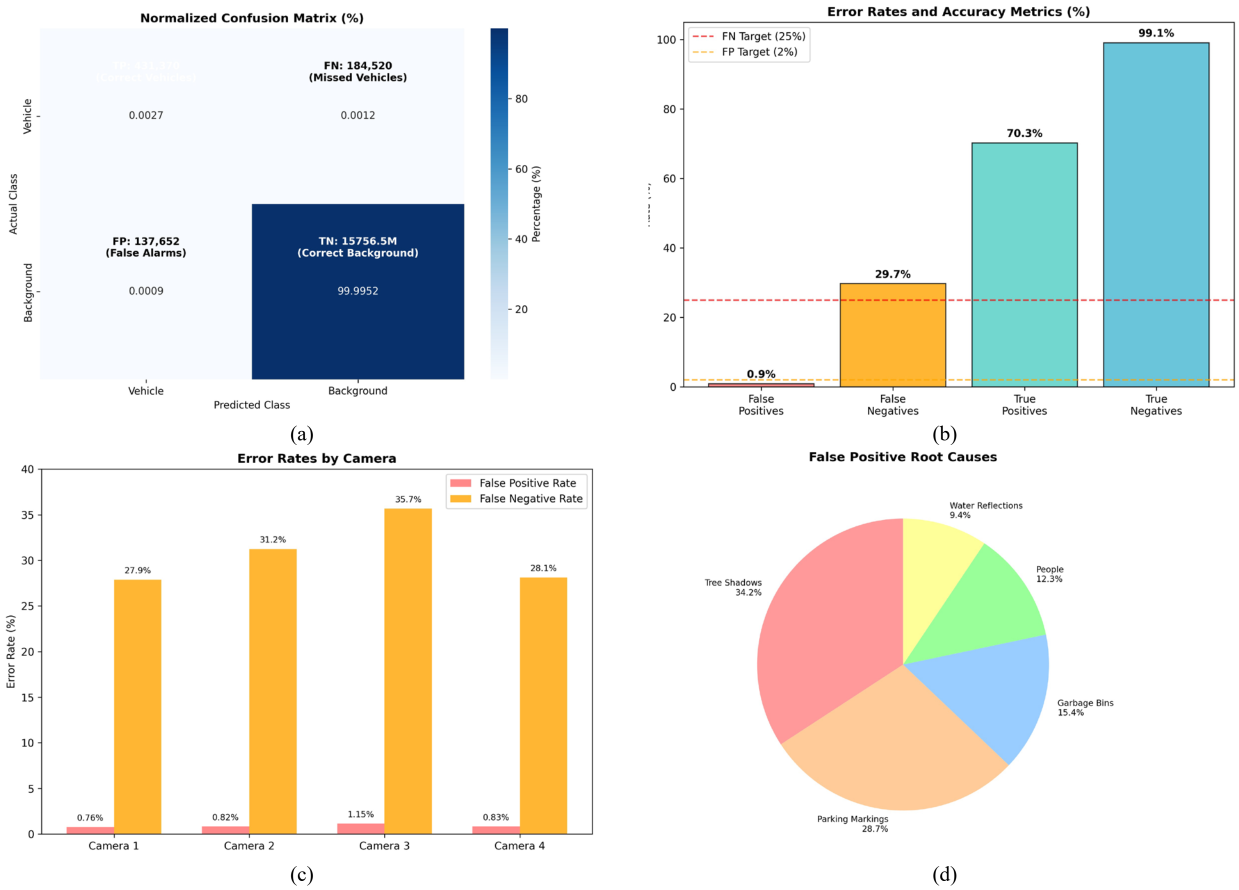


Fig. 13. Classification performance evaluation diagrams: (a) normalized error matrix; (b) accuracy metrics for false positive and false negative classifications; (c) error distribution for four cameras; (d) false positive root causes.

When processing the data, the system forms occupancy matrices for each image, where each matrix element corresponds to a cell of the parking space. The result of constructing the map is shown in Fig. 14. These matrices reflect the probability of the presence of a car in a certain cell.

The current version of the model implements basic image segmentation to determine the occupancy of parking spaces based on camera images. Only passenger cars are recognized. The neural network model may have distortions: in the presence of reflections, snow, and non-standard objects (trucks, buses), segmentation errors may show an accuracy of 74%, which is demonstrated in Fig. 15. In this study, we apply integration with external image segmentation Application Programming Interface (API) services [37].

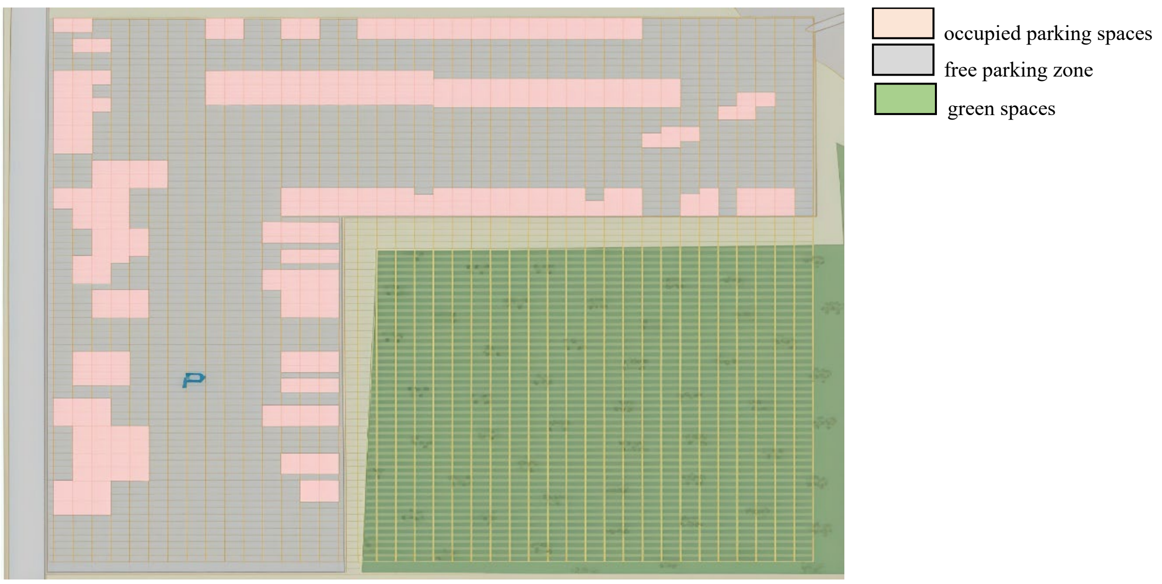


Fig. 14. Overlay of the occupancy matrix on a satellite image of the parking space.

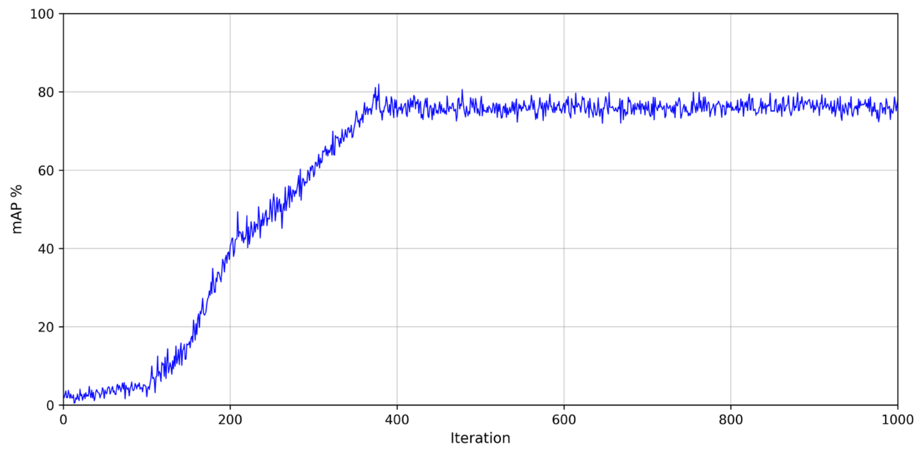


Fig. 15. Graph of the model prediction accuracy during the training process.

Although the segmentation accuracy of the UNet11 model is 74%, this level of performance is sufficient for the proposed task. The system evaluates occupancy at the cell level, where minor local segmentation errors do not affect the final classification result. As a consequence, the end-to-end accuracy of determining parking space status remains high (>90%). Considering the challenging conditions of unmarked parking areas, variable camera angles, and adverse weather, the obtained metrics fall within the expected performance range reported in similar studies and ensure stable and reliable operation in real deployments.

This approach provides increased accuracy in determining the contours of cars and reduces the likelihood of false positives during classification. The results of testing the model on the delayed sample (Fig. 15) shows a comparative visualisation of segmentation quality metrics.

Precision, Recall, F1-score, and IoU were used to provide a more comprehensive evaluation of the segmentation model, capturing both the accuracy and completeness of object detection (Fig. 16). The average cross-validation values demonstrated balanced model performance, and the camera-wise analysis confirmed its stability across different observation zones. Temporal and weather-related variations showed a predictable decline in quality during evening hours and under adverse conditions, yet the model maintained acceptable performance levels. Overall, the set of metrics indicates the reliability of the proposed approach and its applicability in real-world parking monitoring scenarios.

For a more complete characterization of the segmentation model efficiency, in addition to accuracy, additional metrics were calculated—Precision, Recall, F1-score and IoU. The cross-validation results demonstrate that the proposed model attains a precision of 75.4% with a standard deviation of ±2.4%. The recall value amounts to 70.1% (±2.7%), whereas the F1-score reaches 72.6% with a deviation of ±0.4%. Furthermore, the intersection-over-union (IoU) achieves a value of 57.0% accompanied by a standard deviation of ±0.6%. Analysis by camera location showed that F1-score was in the range of 69.8%–75.2%, and IoU is 56.9%–59.3%, which confirms the stability of the model in various observation zones. Temporal variations revealed higher results in the morning (F1-score: 76.8%, IoU: 61.2%) and a decrease in quality in the evening and at night (F1-score: 65.0%, IoU: 48.2%). When comparing scenarios of different complexity, the model showed the greatest stability in clear weather and daylight conditions (F1-score: 82%) and lower values under unfavorable conditions such as rain and night (F1-score: 65%). In addition, the Precision–Recall curve was constructed (Fig. 16), which confirmed the operating point with a balance of Precision = 75.42% and Recall = 70.27%, which reflects the stability of the model when choosing the optimal classification threshold.

All segmentation metrics (Precision = 75.42%, Recall = 70.27%, F1-score = 72.29%, IoU = 56.56%) are computed at the pixel-level for vehicle instance segmentation masks, where IoU represents the intersection-over-union of predicted and ground-truth pixels for the vehicle class. This differs from parking space occupancy IoU, which would measure binary classification (occupied/free) at the parking slot level. Our evaluation focuses on segmentation quality for accurate vehicle boundary detection rather than discrete space-level occupancy classification

Thus, the comprehensive analysis demonstrates that the proposed approach provides reliable detection of parking space occupancy and maintains sufficient accuracy even in complex conditions, which confirms its practical applicability.

Another aspect that requires attention in the practical implementation of the system is data protection and privacy issues. The use of Closed-Circuit Television (CCTV) cameras in parking areas may affect personal information. It should be emphasized that the proposed methodology is focused exclusively on determining the fact of parking space occupancy and does not provide for driver identification or license plate recognition. When implementing such solutions, it is necessary to take into account privacy issues and ensure compliance with established rules for the protection of personal data.

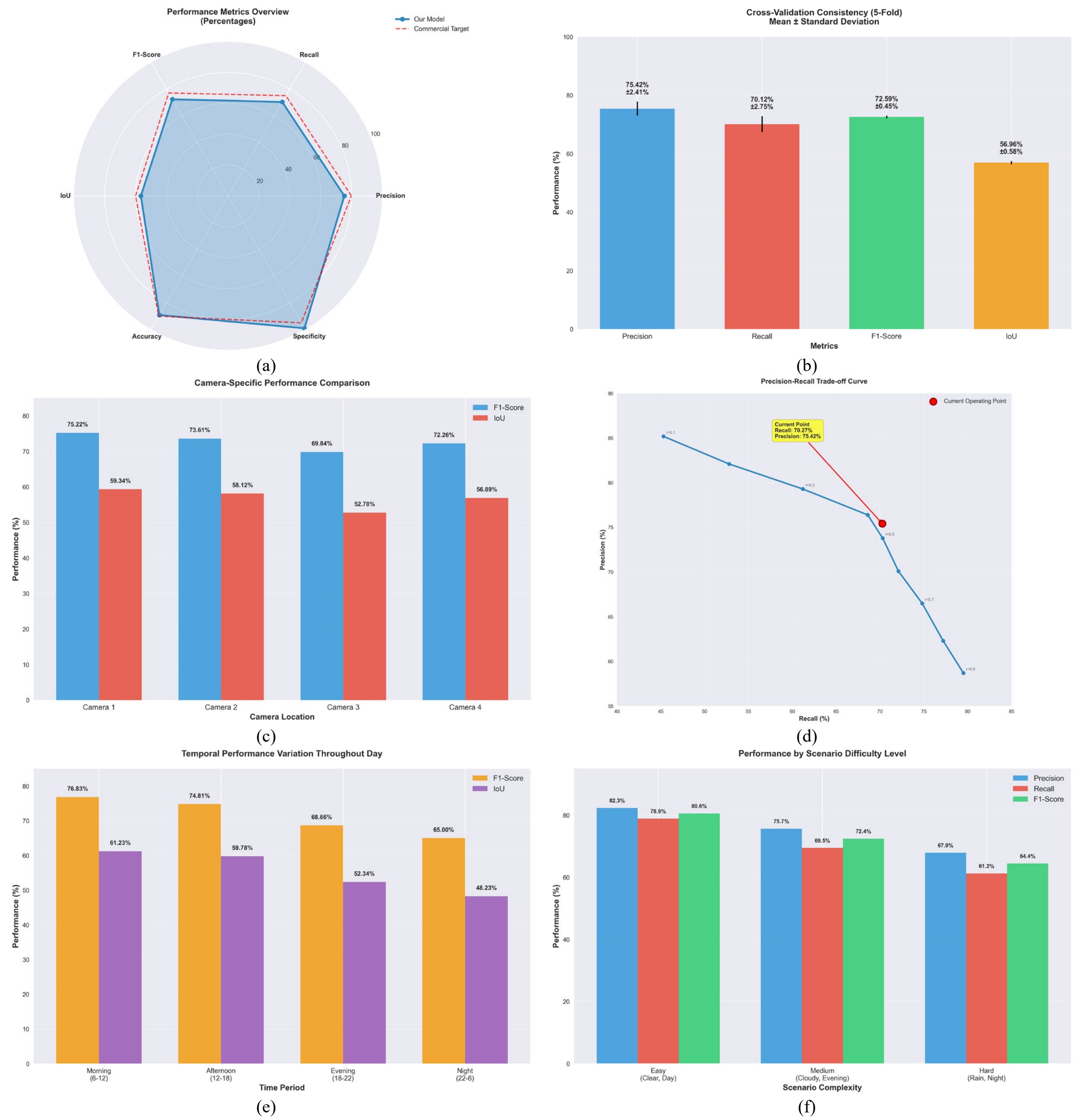


Fig. 16. Comparative visualization of segmentation quality metrics: (a) performance metrics overview; (b) cross validation consistency; (c) camera-specific performance; (d) Precision-Recall trade-off curve; (e) temporal performance variation; (f) performance by scenario difficulty.

VI. Conclusion

This study proposes a method for automated parking occupancy detection in environments where space markings are absent or poorly visible. a U-Net11-based convolutional neural network is used to generate real-time occupancy maps. The model achieved 74% accuracy on the test set, which is expected for segmentation tasks under high variability (different camera angles, lack of markings, weather effects). As a lightweight architecture, U-Net11 prioritizes robustness and speed, and local segmentation errors do not significantly affect the final occupancy estimation, which remains reliable. Model performance was evaluated on an independently annotated test set, yielding Precision—75.4%, Recall—70.1%, F1-score—72.6%, and IoU—57.0%. These results demonstrate a balanced performance and confirm the method's robustness. The approach reliably identifies occupied and free parking spaces in unstructured environments and can be integrated into intelligent parking management systems. Future improvements may include more advanced architectures, expanded weather-aware datasets, and adaptive thresholding based on the pr curve to enhance accuracy and robustness.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization: N. D. and I. D.; methodology: N. D., S. R., Z. K. and I. D.; software: M. T. and A. T.; validation: M. T., A. T. and S. R.; formal analysis: N. D. and I. D.; investigation: M. T.; resources: N. D. and I. D.; data curation: M. T. and A. T.; writing original draft preparation: M.T. and I. D.; writing—review and editing: N. D., S. R., A. T. and I. D.; visualization: M. T., Z. K.; supervision: N. D.; project administration: N. D.; funding acquisition: N. D. All authors have read and agreed to the published version of the manuscript.

Funding

The work was carried out with the support of grant funding for scientific and (or) scientific and technical projects for 2023–2025 from the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant No. AP19677501 "Development of intelligent services for a smart campus with integration into the university information ecosystem").

References

[1] W. Jiang, "Special issue 'Neural network for traffic forecasting'," *Algorithms*, vol. 16, no. 9, 421, 2023. doi: 10.3390/a16090421

[2] J. Chen, Q. Feng, and Q. Guo, "Multi-class freeway congestion and emission based on robust dynamic multi-objective optimization," *Algorithms*, vol. 14, no. 9, 266, 2021. doi: 10.3390/a14090266

[3] J. A. Guzmán-Torres, F. J. Domínguez-Mota, G. Tinoco-Guerrero *et al.*, "Efficacy evaluation of You Only Learn One Representation (YOLOR) algorithm in detecting, tracking, and counting vehicular traffic in real-world scenarios, the case of Morelia México: An artificial intelligence approach," *AI*, vol. 5, no. 3, pp. 1594–1613, 2024. doi: 10.3390/ai5030077

[4] A. H. M. M. Anwar and A. T. Oakil, "Smart transportation systems in smart cities: Practices, challenges, and opportunities for Saudi cities," *Smart Cities: Social and Environmental Challenges and Opportunities for Local Authorities,* pp. 315–337, 2024. doi: 10.1007/978-3-031-35664-3_17

[5] F. Rasheed, Y. Saleem, K. L. A. Yau *et al.*, "The role of deep learning in parking space identification and prediction systems," *Computers, Materials & Continua*, vol. 75, no. 1, pp. 761–784, 2023. doi: 10.32604/cmc.2023.034988

[6] J. Wei, J. He, Y. Zhou *et al.*, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1572–1583, 2019. doi: 10.1109/TITS.2019.2910643

[7] G. Satyanath, J. K. Sahoo, and R. K. Roul, "Smart parking, space detection under hazy conditions using convolutional neural networks: A novel approach," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15415–15438, 2023. doi:10.1007/s11042-022-13958-x

[8] A. Zinelli, L. Musto, and F. Pizzati, "A deep-learning approach for parking slot detection on surround-view images," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 683–688. doi: 10.1109/IVS.2019.8813777

[9] H. Do and J. Y. Choi, "Context-based parking slot detection with a realistic dataset,", *IEEE Access,* vol. 8, pp. 171551–171559, 2020. doi: 10.1109/ACCESS.2020.3024668

[10] C. Min, J. Xu, L. Xiao *et al.*, "Attentional graph neural network for parking-slot detection," *IEEE Robotics and Automation Letters*, vol .6, no. 2, pp. 3445–3450, 2021. doi: 10.1109/LRA.2021.3064270

[11] S. A. Ratti, N. Pirzada, S. M. A. Shah *et al.*, "Intelligent car parking system using WSN," in *Proc. Global Conference on Wireless and Optical Technologies (GCWOT)*, 2023, pp. 1–9. doi: 10.1109/GCWOT57803.2023.10064656

[12] S. Hanumanthakari, "Intelligent and real-time parking system," in *Proc. E3S Web of Conferences*, 2024, vol. 472. doi: 10.1051/e3sconf/202447203003

[13] A. Xiao, D. Doshi, L. Wang *et al.*, "Parking spot classification based on surround view camera system," in *Proc. Applications of Machine Learning 2023*, 2023, pp. 147–154. doi: 10.1117/12.2677526

[14] K. Yamamoto *et al.*, "Simulation evaluation of vehicle movement model using spatio-temporal grid reservation for automated valet parking," *IEEE Open Journal of Intelligent Transportation Systems,* vol. 4, pp. 261–266, 2023. doi: 10.1109/OJITS.2023.3266556

[15] S. Stranieri, "An indoor smart parking algorithm based on fingerprinting," *Future Internet*, vol. 14, no. 6, 185, 2022. doi: 10.3390/fi14060185

[16] X. Liu, S. Zhu, Y. Fang *et al.*, "Optimization design of parking models based on complex and random parking environments," *World Electric Vehicle Journal,* vol. 14, no. 12, 344, 2023. doi: 10.3390/wevj14120344

[17] I. Ashraf, S. Hur, G. Kim, and Y. Park, "Analyzing performance of YOLOx for detecting vehicles in bad weather conditions," *Sensors*, vol. 24, no. 2, 522, 2024. doi: 10.3390/s24020522

[18] A. O. Elfaki, W. Messoudi, A. Bushnag *et al.*, "A smart real-time parking control and monitoring system," *Sensors*, vol. 23, no. 24, 9741, 2023. doi: 10.3390/s23249741

[19] Y. Sari, H. Suhud, A. R. Baskara *et al.*, "Parking lots detection in static image using support vector machine based on genetic algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 1, no. 1, pp. 476–487, 2021. doi: 10.22266/ijies2021.1231.42

[20] A. Shahzad, A. Gherbi, and K. Zhang, "Enabling fog—blockchain computing for autonomous-vehicle-parking system: A solution to reinforce IoT-cloud platform for future smart parking," *Sensors*, vol. 22, no. 13, 2022. doi:10.3390/s22134849

[21] B. Veena, M. Sachin, S. Achyuth Kumar, and B Surekha, "Automatic vacant parking places management system using multicamera vehicle detection," *Interantional Journal of Scientific Research in Engineering and Management*, vol. 9, no. 1, 2025. doi: 10.55041/IJSREM40550

[22] D. Neupane, A. Bhattarai, S. Aryal *et al.*, "Shine: A deep learning-based accessible parking management system," *Expert Systems with Applications*, vol. 238, 2024. doi: 10.1016/j.eswa.2023.122205

[23] S. K. Kabilesh, J. Arun, N. Arisudhan *et al.*, "Highly secure Smart Vehicle Parking System (SVPS) for smart cities," *Journal of VLSI Design and its Advancement*, vol. 6, no. 1, 2023. doi: 10.5281/zenodo.7619535

[24] A. Mishra, A. Karmakar, A. Ghatak *et al.*, "Low cost parking system for smart cities: A vehicle occupancy sensing and resource optimization technique using IoT and cloud Paas," *International Journal of Scientific Technology Research*, vol. 8, no. 9, pp. 115–122, 2019.

[25] S. Ma, H. Jiang, M. Han *et al.*, "Research on automatic parking systems based on parking scene recognition," *IEEE Access*, vol. 5, pp. 21901–21917, 2017. doi: 10.1109/ACCESS.2017.2760201

[26] M. Hanzla, M. O. Yusuf, and A. Jalal, "Vehicle surveillance using U-NET segmentation and DeepSORT over aerial images," in *Proc. the 2024 International Conference on Autonomous Systems*, 2024.

[27] Y. Wu, T. Yang, J. Zhao *et al.*, "VH-HFCN based parking slot and lane markings segmentation on panoramic surround view," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1767–1772. doi: 10.1109/IVS.2018.8500553

[28] W. Li, L. Cao, L. Yan *et al.*, "Vacant parking slot detection in the around view image based on deep learning," *Sensors*, vol. 20, no. 7, 2138, 2020. doi: org/10.3390/s20072138

[29] D. H. Jian and C. H. Lin, "Vision-based parking slot detection based on end-to-end semantic segmentation training," in *Proc. IEEE International Conference on Consumer Electronics (ICCE)*, 2020. doi: 10.1109/ICCE46568.2020.9043164

[30] W. Li, L. Cao, J. Liao, and Z. Wang, "Vacant parking slot detection and tracking during driving and parking with a standalone around view monitor," in *Proc. the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 6, pp. 1539–1551, 2020. doi: 10.1177/0954407020980

[31] Y. Park, J. Ahn, and J. Park, "Deep learning based parking slot detection and tracking: PSDT-Net," in *Proc. International*

Conference on Robot Intelligence Technology and Applications, 2021, pp. 291–302. doi: 10.1007/978-3-030-97672-9_26
[32] V. Abhinav and P. P. Priya, "Smart parking system using Arduino over IoT merge with DL web apps," in *Proc. AMCEC*, 2022. doi: 10.13140/RG.2.2.30434.30405/1
[33] A. Kurek and E. Macioszek, "Drivers subjective assessment of the ease of finding a vacant parking space in an area equipped with vehicle detection devices," *Sensors*, vol. 22, no. 18, 6734, 2022. doi: 10.3390/s22186734
[34] M. Elbanhawi and M. Simic, "Examining the use of b-splines in parking assist systems," *Applied Mechanics and Materials*, vol. 490, pp. 1025–1029, 2014. doi:10.4028/www.scientific.net/AMM.490-491.1025
[35] Y. Zhao, "Automated parking planning with vision-based BEV approach," arXiv Preprint, arXiv: 2406.15430, 2024.
[36] B. Shaler *et al.* (2017). Carvana Image masking challenge. *Kaggle.* [Online]. Available: https://www.kaggle.com/c/carvana-image-masking-challenge/
[37] A. Tlebaldinova, N. Denissova, O. Baklanova *et al.*, "Normalization of vehicle license plate images based on analyzing of its specific features for improving the quality recognition," *Acta Polytechnica Hungarica*, vol. 17, no. 6, pp. 193–206, 2020. doi: 10.12700/APH.17.6.2020.6.11
[38] S. S. Altyar, S. S. Hussein, and L. A. Tawfeeq, "Accurate license plate recognition system for different styles of Iraqi license plates," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 1092–1102, 2023. doi: 10.11591/eei.v12i2.4186
[39] P. M. Chu, S. Cho, K. Huang *et al.*, "Flood-fill-based object segmentation and tracking for intelligent vehicles," *International Journal of Advanced Robotic Systems*, vol. 16, no. 6, 2019. doi: 10.1177/1729881419885206
[40] Automatic Mask Generator. *Segmind*. [Online] Available: https://www.segmind.com/models/automatic-mask-generator/api
[41] Y. Feng, Y. Xu, Q. Hu *et al.*, "Predicting vacant parking space availability zone-wisely: A hybrid deep learning approach," *Complex & Intelligent Systems,* vol. 8, no. 5, pp. 4145–4161, 2022. doi: 10.1007/s40747-022-00700-1
[42] Y. Jo, J. Ha, and S. Hwang, "Survey of technology in autonomous valet parking system," *International Journal of Automotive Technology*, vol. 24, no. 6, pp. 1577–1587, 2023. doi: 10.1007/s12239-023-0127-1