

A Transformer-Based Smart Parking Monitoring System with Automatic Labeling and Real-Time Notification

Hyun Il Chong¹ and Kwan-Hee Yoo^{2,*}

¹ Department of Industrial AI, Chungbuk National University, Cheongju, South Korea; chohi@chungbuk.ac.kr

² School of Computer Science, Chungbuk National University, Cheongju, South Korea; khyoo@chungbuk.ac.kr

* Correspondence

<https://doi.org/10.5392/IJoC.2026.22.3.069>

Manuscript Received 8 May 2026; Received 19 June 2026; Accepted 30 July 2026

Abstract: This paper presents NOVA Parking, a transformer-based smart parking monitoring system. Conventional convolutional neural network (CNN) detectors perform poorly when deployed in unfamiliar parking lots with varying viewpoints, markings, and lighting conditions. To address this issue, NOVA Parking employs Grounding DINO, an open-vocabulary vision-language detector that aligns text prompts with image regions for robust detection across different environments. The system integrates closed-circuit television (CCTV) acquisition, intersection-over-union (IoU)-based tracking, region-of-interest (ROI)-based parking violation analysis, a Segment Anything Model (SAM)-assisted automatic labeling pipeline, and web/mobile services with real-time push notifications. On a dataset of 14,908 images from three locations, the proposed approach achieves a mean Average Precision (mAP) of 95.23% at a threshold of 0.5 for trained lots and 94.12% for unseen lots, retaining approximately 99% of the trained-lot mAP. In comparison, YOLOv12 and Faster R-CNN achieve mAPs of 81.67% and 78.34%, respectively. A five-year analysis further indicates that NOVA Parking has a total cost of ownership that is roughly 71% lower than that of magnetometer-based systems, confirming the practicality and economic feasibility of NOVA Parking for intelligent transportation systems (ITS).

Keywords: Smart Parking System; Object Detection; Intelligent Transportation Systems (ITS); Real-time Monitoring; Transformer; Grounding DINO

1. Introduction

Rapid urbanization and the steady growth in vehicle ownership have made parking scarcity a persistent social and economic problem in major cities worldwide. A substantial portion of urban traffic congestion is associated with drivers searching for available parking spaces, which increases travel time, fuel consumption, and emissions [1]. To address this issue, a wide range of smart parking systems (SPSs) have been proposed, spanning sensor-based, camera-based, and hybrid architectures [2, 3].

Traditional parking monitoring systems commonly rely on embedded sensors, such as magnetometers, ultrasonic sensors, or other wireless vehicle detectors installed at individual parking spaces [3-5]. Although these systems can provide reliable occupancy sensing, they typically require per-space hardware deployment, communication infrastructure, and periodic maintenance, which increase installation and operational costs, particularly in large outdoor parking facilities [3, 4]. In addition, magnetic sensing in real roadside environments may be affected by adjacent vehicles, vehicle geometry (e.g., high-chassis vehicles), and environmental conditions, which can degrade detection reliability unless additional sensing or fusion mechanisms are introduced [5].

Recent advances in computer vision and deep learning have made camera-based parking monitoring an attractive alternative. By using a single overhead or roadside camera to monitor multiple parking spaces, camera-based systems can reduce per-space infrastructure costs while providing rich visual information for occupancy monitoring, event analysis, and user-facing services [2], [6, 7]. In particular, convolutional neural network (CNN)-based object detectors such as Faster R-CNN [8], Mask R-CNN [9], and recent You Only Look Once (YOLO)-family models [10, 11] have demonstrated strong in-domain accuracy and real-time processing capability, thereby enabling practical deployment in parking-lot surveillance scenarios.

However, field deployment of CNN-based detectors reveals a critical limitation: models trained on one parking lot often experience performance degradation when applied to unseen lots with different viewpoints, pavement markings, layout geometry, shadows, or illumination conditions. Bridging this domain gap typically requires additional data collection, site-specific annotation, and repeated fine-tuning, which reduces scalability and increases deployment cost. This challenge is particularly important in outdoor parking environments, where visual conditions vary substantially across locations and seasons [6], [12, 13].

To address this limitation, this study investigates a transformer-based vision-language detection approach for smart parking monitoring. Specifically, we adopt **Grounding DINO** [14], an open-vocabulary multimodal detector built on the DETection TRansformer (DETR)-family paradigm [15], which aligns textual prompts with image regions through cross-modal feature fusion. Unlike conventional closed-vocabulary detectors, Grounding DINO can leverage text prompts (e.g., vehicle, occupied parking space, empty parking space) to support semantically guided detection and improve adaptability to previously unseen environments.

Building on this detector, we present **NOVA Parking**, a full-stack smart parking monitoring system with automatic labeling and real-time notification. The system integrates closed-circuit television (CCTV)-based video acquisition, transformer-based object detection, intersection-over-union (IoU)-based tracking-by-detection, ROI-based parking-violation analysis, and web/mobile notification services. In addition, we design an automatic labeling pipeline with optional Segment Anything Model (SAM)-based label refinement [16] to reduce manual annotation effort and support scalable dataset construction for parking occupancy monitoring. This paper substantially extends our preliminary study [17] with an automatic labeling pipeline (optional SAM-based label refinement), ROI-based violation analysis, a quantitative cross-site generalization study, and a detailed cost-benefit analysis.

The main contributions of this paper are as follows:

- We design and implement a full-stack smart parking monitoring system centered on Grounding DINO and validate its deployment on a physical outdoor parking-lot prototype.
- We propose an automatic labeling pipeline—combining open-vocabulary Grounding DINO detection, geometric parking-space estimation, and rule-based occupancy assignment, with SAM box-prompted segmentation as an optional label-refinement step—that constructs parking monitoring datasets with substantially reduced manual annotation effort.
- We develop NOVA Parking algorithms for IoU-based object ID tracking and region-of-interest (ROI)-based event analysis to support stable monitoring and event generation in CCTV streams.
- We conduct a quantitative comparison against CNN-based baselines (Faster R-CNN, Mask R-CNN, and YOLOv12) on both trained and unseen environments, and perform a cost-benefit analysis against conventional wireless sensor-based parking monitoring systems.

2. Related Work

Smart parking research can be broadly categorized into sensor-based systems, camera-based vision systems, and integrated service platforms [2-4]. Sensor-based systems typically use magnetometers, ultrasonic sensors, infrared sensors, or wireless sensor networks to detect vehicle presence at individual parking spaces [3-5]. These approaches are widely used in both indoor and roadside parking scenarios because they can provide direct occupancy signals and low-latency event updates. However, large-scale deployment requires per-space sensing hardware, network provisioning, and long-term maintenance, which may limit scalability in outdoor parking lots [3, 4]. Moreover, magnetometer-based sensing can be susceptible to interference from adjacent vehicles or vehicle-dependent magnetic signatures, motivating the use of signal fusion and hybrid sensing strategies [5]. In addition, transaction-level entry and exit data have been used to identify available parking spaces in managed facilities [18].

Camera-based parking monitoring addresses these limitations by using image analysis to infer parking occupancy and vehicle presence over multiple spaces from a small number of cameras [2], [6, 7]. Early methods relied on background subtraction, handcrafted features, or classical classifiers, which were often sensitive to viewpoint changes, illumination variation, and weather conditions. With the emergence of deep learning, CNN-based solutions significantly improved occupancy detection accuracy and robustness [6]. Representative datasets such as PKLot [12] and CNRPark-EXT (Extended) [13] accelerated research in this area by providing large-scale labeled examples collected under diverse camera views and environmental conditions.

CNN-based object detection architectures have become a dominant choice for parking-related visual recognition tasks. **Faster R-CNN** [8] introduced region proposal networks for efficient two-stage detection, while **Mask R-CNN** [9] extended this framework with instance segmentation capability. More recent one-stage detectors in the **YOLO family** offer higher inference speed and strong accuracy, making them suitable for real-

time smart parking applications [10, 11]. Nevertheless, these CNN-based detectors are typically trained under closed-vocabulary settings and may require retraining or fine-tuning when deployed to new parking sites with different scene characteristics.

Transformer-based detectors provide an alternative formulation for object detection. **DETR** [15] demonstrated end-to-end set prediction using transformer encoder-decoder architectures, reducing reliance on hand-crafted detection components such as anchor design and non-maximum suppression. Building on the DETR family, **Grounding DINO** [14] incorporates language guidance through multimodal fusion, including a feature enhancer, language-guided query selection, and a cross-modality decoder, thereby enabling open-vocabulary object detection from textual prompts. This property is highly relevant to smart parking monitoring, where target semantics (e.g., vehicles, occupied spaces, empty spaces) can be specified in natural language and reused across sites without strict class-specific redesign.

In addition to detection accuracy, practical smart parking systems require scalable data preparation and service integration. Manual annotation remains a bottleneck in site-specific deployment, especially when parking-space templates and occupancy labels must be created for new camera views. Promptable segmentation models such as SAM [16] provide useful building blocks for semi-automatic or automatic labeling workflows, enabling faster dataset construction and label refinement. Furthermore, recent smart parking studies increasingly emphasize end-to-end deployment aspects-including real-time streaming, event notification, edge/cloud processing, and mobile/web interfaces-rather than standalone model evaluation [2], [11].

Despite these advances, two gaps remain prominent in the literature. First, many camera-based systems report high performance on trained datasets but provide limited analysis of **cross-site generalization** under unseen parking environments. Second, comparatively fewer studies present a **full-stack deployment framework** that integrates multimodal detection, automatic labeling, temporal event analysis, and real-time user notification in a unified system. The proposed NOVA Parking system is designed to address these gaps by combining Grounding DINO-based transformer detection with an automatic labeling pipeline (optional SAM-based label refinement), NOVA Parking algorithms, and an operational web/mobile monitoring service.

3. Proposed System Architecture and Transformer-Based Detection Model

3.1 Overall System Architecture of the Proposed Smart Parking Monitoring System

Figure 1 presents the overall architecture of the proposed intelligent parking monitoring system, which is designed as an integrated end-to-end platform for real-time parking-space surveillance, occupancy analysis, and user notification. The system consists of multiple functional modules, including Collection, AI Detection, Notification, Dashboard, Mobile Web, and FCM-based alert delivery, which are interconnected through database and streaming interfaces. This modular design enables scalable deployment in outdoor parking environments while supporting both administrator-level monitoring and end-user parking information services.

The **Collection** module acquires video streams from CCTV cameras through a network interface and manages acquisition-related configurations, including video input settings, collection status management, and operational parameter control. These video streams are transmitted to the AI processing pipeline, where the **AI Detection** module loads the trained detection model, performs parking-space occupancy detection, and applies post-processing to generate structured prediction results. The output of this module includes occupancy-related detection results and metadata that can be used for subsequent aggregation, visualization, and event-driven notification.

The **Notification** module aggregates detection outputs in real time and determines parking-space status updates for downstream services. It manages Real-Time Streaming Protocol (RTSP)-based video relay for live monitoring and transmits event information to the Firebase Cloud Messaging (FCM) sender for push notification delivery. This design allows the system to provide real-time parking availability updates to users while maintaining compatibility with web and mobile service channels.

The **Dashboard** and **Mobile Web** modules provide user-facing interfaces for system operation and information access. The dashboard is intended primarily for administrators and business users, supporting status services, CCTV monitoring, RTSP streaming, and system management functions. In contrast, the mobile web interface focuses on lightweight user interaction, including parking status inquiry, push alarm reception, and configuration management. The integration of web and mobile clients improves service accessibility and supports real-time dissemination of parking information across heterogeneous user groups.

At the platform level, the architecture also includes infrastructure components such as an **RTSP Router**, **FCM Pusher**, **database management system (DBMS) management**, and **Map application programming interface (API) integration** (e.g., Mapbox/Naver Map), which support data routing, alert transmission,

persistent storage, and map-based visualization. In addition, the lower pipeline in Figure 1 summarizes the model development workflow, including data collection, data processing, label data generation, model training, model evaluation, and model saving. This workflow reflects the practical lifecycle of the proposed system, linking offline model development and online deployment within a unified smart parking service framework.

Overall, the architecture in Figure 1 demonstrates that the proposed system is not limited to object detection alone, but rather constitutes a deployable smart parking monitoring platform that integrates data acquisition, AI inference, service orchestration, and real-time user notification.

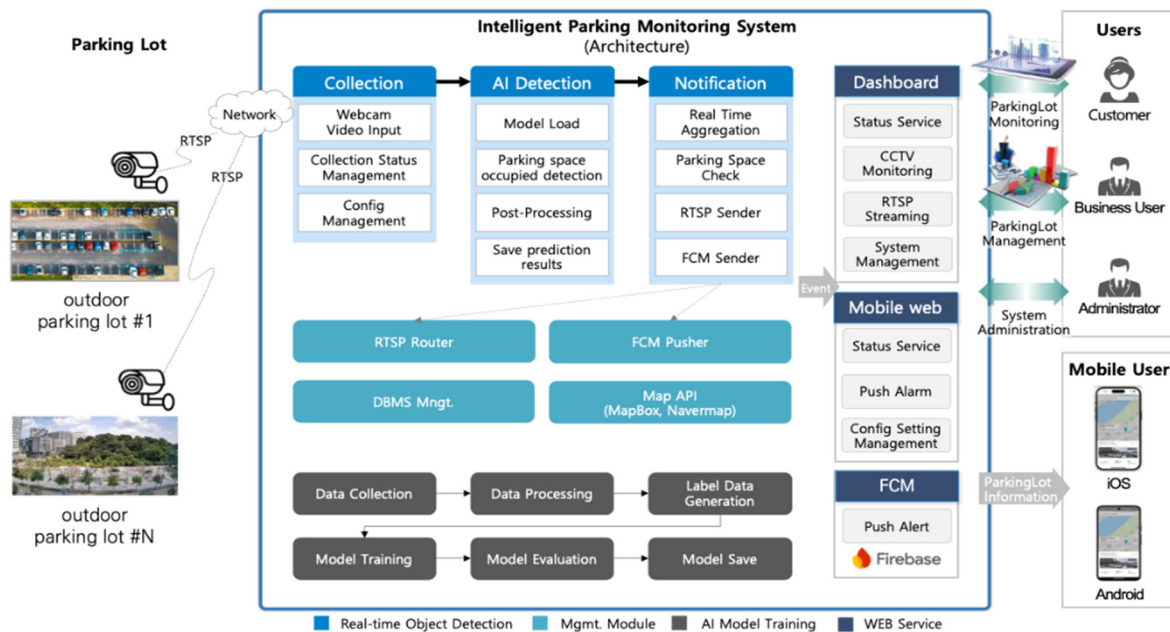


Figure 1. Overall architecture of the proposed intelligent parking monitoring system, including data collection, AI detection, notification, dashboard/mobile services, database management, and FCM-based alert delivery.

3.2 Transformer-Based Multimodal Detection Module

Figure 2 illustrates the transformer-based multimodal object detection module adopted as the core recognition engine in the proposed smart parking monitoring system. The module is designed to jointly process **parking-lot image frames** and **text prompts** describing target semantics (e.g., *vehicle*, *occupied parking space*, and *empty parking space*), thereby enabling semantically guided detection in diverse outdoor parking environments. Unlike conventional CNN-based detectors with fixed class definitions, the proposed framework leverages text-image alignment to improve robustness and adaptability when deployed in parking lots with previously unseen scene characteristics.

As shown in Figure 2, the detection pipeline begins with modality-specific representation learning using separate **Text Backbone** and **Image Backbone** networks. The text backbone transforms the input prompt into contextualized language embeddings, while the image backbone extracts hierarchical visual features from the input frame. These outputs form the initial text and image feature representations, preserving modality-specific semantic and spatial information prior to multimodal interaction. Such a design is beneficial for parking monitoring tasks, where fine-grained visual cues (e.g., parking-line boundaries, partial vehicle occlusion, and shadows) must be interpreted in conjunction with semantic categories defined by text prompts.

The extracted features are then processed by a **Feature Enhancer Layer**, which performs bidirectional cross-modal interaction through self-attention and cross-attention operations. This layer refines text and image features by allowing each modality to attend to relevant information from the other, thereby producing semantically aligned representations. In the image branch, deformable attention further improves the model's ability to handle scale variation and geometric distortion caused by camera viewpoint differences, installation height, and perspective effects. This feature enhancement mechanism is particularly important in real-world parking lots, where environmental conditions vary substantially across sites.

To support grounded object detection, the module employs **Language-Guided Query Selection**, which initializes decoder queries using the refined multimodal features. By conditioning object queries on both textual

semantics and visual evidence, the model can focus more effectively on parking-relevant regions and reduce ambiguity caused by background structures, lane markings, or transient scene artifacts. This query initialization strategy contributes to stable occupancy-related detection performance across trained and unseen environments.

The initialized queries are subsequently refined in a **Cross-Modality Decoder**, where self-attention, image cross-attention, and text cross-attention are applied iteratively. Through this decoding process, each query integrates spatial localization cues from image features and semantic constraints from text features, resulting in object-level embeddings that are both spatially precise and semantically grounded. The decoder outputs are then used to predict bounding boxes and associated detection confidences for parking-related objects and occupancy states.

Model optimization is performed using a combination of **localization loss** and **contrastive loss**, as indicated in Figure 2. The localization loss enforces accurate spatial prediction of target regions, while the contrastive loss improves correspondence between text concepts and visual regions. Joint optimization of these objectives enhances both detection accuracy and cross-modal consistency, which is essential for robust parking-space monitoring under heterogeneous weather, lighting, and scene conditions.

In the proposed system, the outputs of this detection module are forwarded to downstream components for tracking, temporal status analysis, parking-space state aggregation, and real-time notification. Therefore, the architecture in Figure 2 provides the foundational perception capability that enables the practical deployment of the proposed smart parking monitoring platform.

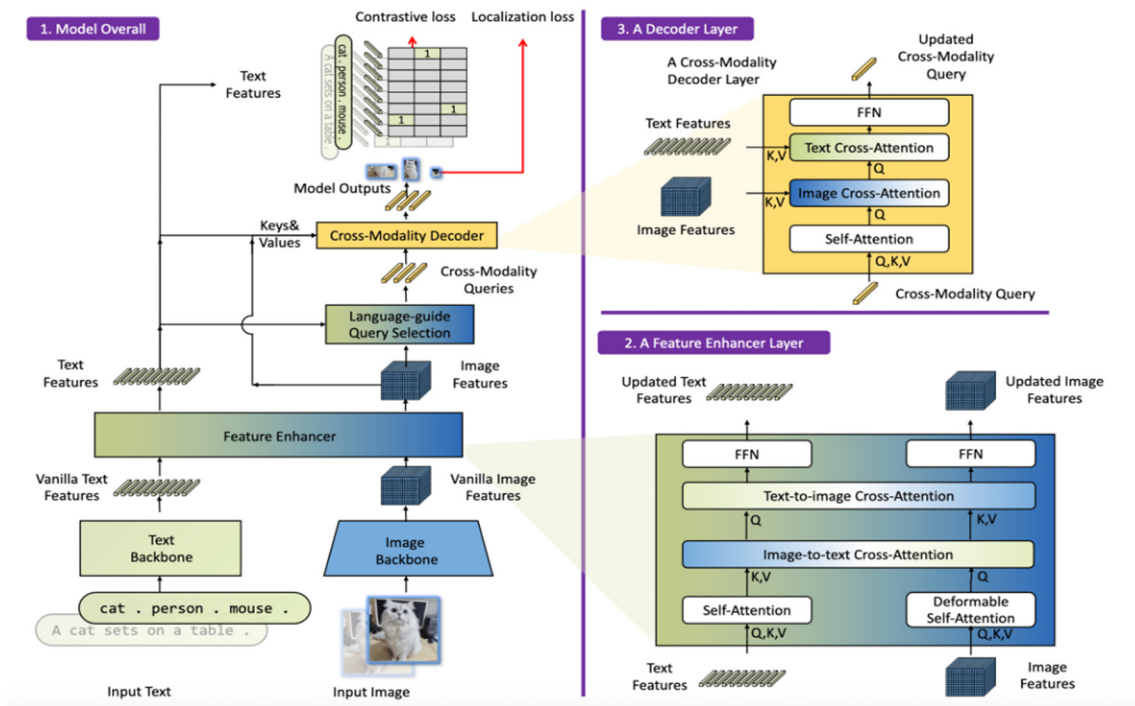


Figure 2. Overall architecture of the proposed transformer-based smart parking monitoring model. The framework encodes text prompts and parking-lot images using modality-specific backbones, enhances cross-modal representations through a feature enhancer, and performs grounded object detection via language-guided query selection and a cross-modality decoder.

3.3 Automatic Labeling Pipeline with Optional SAM-Based Refinement

To address the inefficiency and high labor cost of manual annotation, an automatic labeling process was designed to generate parking-related labels with minimal human intervention (Figure 3). The proposed process begins with the construction of a small seed dataset composed of manually annotated parking-lot images. These seed annotations provide the initial supervision required to define the target semantics (e.g., vehicles and parking-space occupancy states) and to initialize the labeling pipeline. By restricting manual labeling to a limited subset of images, the annotation burden can be substantially reduced while preserving sufficient domain-specific information for automatic dataset expansion.

A pretrained Grounding DINO model is then used to infer parking-related objects from newly collected parking-lot images. Owing to its text-guided detection capability and cross-environment adaptability, the

pretrained model can generate preliminary detections under varying viewpoints, scene layouts, and illumination conditions. In this study, the detector outputs are used as the primary source of information for constructing occupancy labels and estimating parking-space geometry.

To refine these coarse detections into pixel-accurate object boundaries, the proposed pipeline employs the Segment Anything Model (SAM) in a box-prompted configuration. Specifically, each vehicle bounding box produced by the pretrained Grounding DINO detector is passed to SAM as a box prompt (rather than a point prompt), and SAM returns the corresponding instance segmentation mask. The tight axis-aligned bounding box enclosing this mask—i.e., the minimum and maximum mask coordinates along the horizontal and vertical axes—is then taken as the refined object bounding box. This box-prompt strategy requires no manual point selection; tightening detector boxes via segmentation is a general property of SAM and is not separately quantified as a detection-accuracy gain in this study. In the reported pipeline, the parking-space geometry (Eq. (1)) and the Empty/Occupied label of each space (Eq. (2)) are derived directly from the Grounding DINO vehicle boxes. SAM box-prompted refinement is provided as an optional, auxiliary step rather than a mandatory stage of label generation: it tightens the detector boxes, and the agreement between the reported boxes and their SAM-refined counterparts is quantified in Section 4.5 (Table 5). The final Empty/Occupied state of each parking space is therefore decided by the IoU-based occupancy rule of Eq. (2), not by the segmentation mask itself.

After occupancy-related detections are obtained, representative parking-space bounding boxes are estimated by aggregating multiple detection results observed in the same parking-lot scene. Let $g_m^{(t)} = [x_m^{(t)}, y_m^{(t)}, w_m^{(t)}, h_m^{(t)}]$ denote the bounding box parameters of the m -th parking space observed at frame t , and let T_m be the number of valid observations for that parking space. The average parking-space bounding box is computed as

$$\hat{G}_m = \frac{1}{T_m} \sum_{t=1}^{T_m} g_m^{(t)} = \left[\frac{1}{T_m} \sum_{t=1}^{T_m} x_m^{(t)}, \frac{1}{T_m} \sum_{t=1}^{T_m} y_m^{(t)}, \frac{1}{T_m} \sum_{t=1}^{T_m} w_m^{(t)}, \frac{1}{T_m} \sum_{t=1}^{T_m} h_m^{(t)} \right]. \quad (1)$$

This averaging operation reduces the effect of frame-level noise, temporary occlusion, and unstable detections, thereby producing a stable parking-space template set for subsequent label generation. In our implementation, vehicle detections accumulated over multiple frames are grouped into parking-space clusters by box-center distance (50 pixels) before averaging.

Based on the estimated parking-space set, labels for empty parking spaces are generated by excluding parking spaces matched to detected vehicle regions. Let $G = \{G_1, G_2, \dots, G_M\}$ denote the set of estimated parking-space bounding boxes and $D = \{D_1, D_2, \dots, D_N\}$ denote the set of detected vehicle bounding boxes. Using an overlap criterion based on intersection-over-union (IoU) with threshold τ , the set of empty parking spaces is defined as

$$E = \left\{ G_i \in G \mid \max_{1 \leq j \leq N} \text{IoU}(G_i, D_j) < \tau \right\}, \text{IoU}(G_i, D_j) = \frac{|G_i \cap D_j|}{|G_i \cup D_j|}. \quad (2)$$

Accordingly, parking spaces not included in E are labeled as occupied. This formulation enables automatic derivation of both **Empty** and **Occupied** labels from pretrained model detections, significantly reducing manual annotation effort while maintaining labeling consistency across heterogeneous parking-lot environments. The occupancy-assignment IoU threshold of Eq. (2) is set to 0.5 in our implementation.

Overall, the proposed automatic labeling process combines pretrained vision-language detection, geometry-based parking-space estimation, and rule-based occupancy assignment—with SAM box-prompted segmentation available as an optional box-refinement step—to support efficient and scalable dataset construction for smart parking monitoring applications.

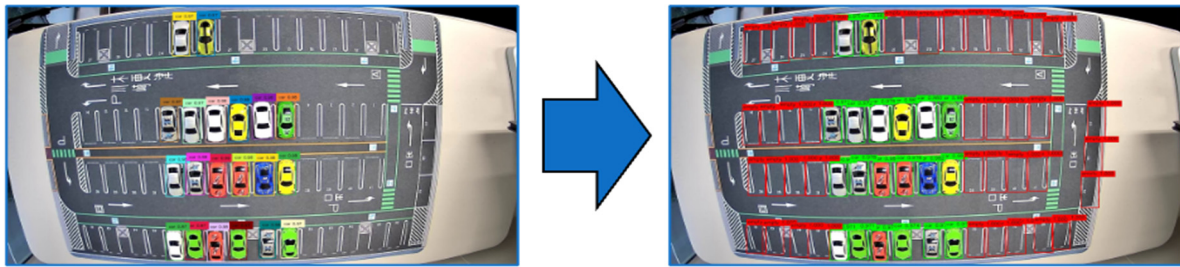


Figure 3. Automatic labeling process for parking monitoring data generation. A small manually annotated seed dataset and a pretrained Grounding DINO model are used to infer occupancy-related detections; the average parking-space geometry and the empty/occupied labels are generated automatically from these detections (Eqs. (1)–(2)), while SAM box-prompted segmentation provides an optional refinement of the vehicle boxes whose localization effect is quantified in Table 5.

3.4 NOVA Parking Algorithms

This section describes the core online decision logic of the proposed NOVA Parking system, referred to as the NOVA Parking algorithms, which transform frame-level object detections into temporally consistent monitoring results and event-level notifications. While the transformer-based detector in Section 3.2 provides object-level predictions for each frame, practical parking monitoring requires additional temporal reasoning to (i) maintain stable object identities across frames and (ii) determine whether a detected vehicle constitutes a parking violation within a designated region. To this end, the proposed system employs two complementary algorithms: Algorithm 1 (IoU-based Tracking-by-Detection) and Algorithm 2 (ROI-based Parking-Violation Detection).

Algorithm 1 maintains persistent object identities in CCTV video streams by associating detections over time using intersection-over-union (IoU)-based matching. The resulting tracked objects are then used by Algorithm 2, which evaluates whether a tracked vehicle remains within a restricted region of interest (ROI) for longer than a predefined dwell-time threshold. This two-stage design is intended to improve the reliability of event generation by reducing false alarms caused by momentary stops, missed detections, or short-term occlusions.

The purpose of Algorithm 1 is to assign temporally consistent IDs to detected objects in consecutive frames of a CCTV stream. Given frame-level detections from the object detection module, the algorithm compares the current detections with previously maintained tracks and preserves object IDs when sufficient spatial overlap is observed. In the proposed system, this tracking-by-detection strategy is used to stabilize vehicle trajectories and to provide persistent object references for subsequent parking-violation analysis.

At each frame, the age of all active tracks is updated, and the IoU matrix is computed between the set of existing track bounding boxes and the current detection bounding boxes. If the maximum IoU for a detection exceeds a predefined threshold T_{track} , the detection is assigned to the matched track and the original object ID is preserved. Otherwise, a new track is initialized with a new ID. The output of the algorithm is an updated track set containing object IDs, bounding boxes, confidence scores, and track states. An example output is shown as an ID-confidence pair (e.g., ID52780, 0.95), indicating object ID 52780 with detection confidence 95%.

Algorithm 1: IoU-based Tracking-by-Detection

Input: `dets_xyxy`, `dets_score`, `dets_cls`

Output: `updated_tracks` (with IDs, bboxes, scores, and states)

```

1: for each track t in tracks do
2:   t.age ← t.age + 1
3: end for

4: A ← [t.bbox for t in tracks]
5: I ← IoU_matrix(A, dets_xyxy)
6: for each detection d in dets_xyxy do
7:   (i*, iou_max) ← best_match(I, d)
8:   if iou_max ≥ T_track then
9:     assign d to tracks[i*]      // preserve ID
10:    update tracks[i*] state, bbox, score, class
11:   else
12:     create new track with new ID

```

```

13:   end if

14: end for

15: remove or deactivate stale tracks (optional, by max_age)

16: return updated_tracks

```

Figure 4 shows an example of the tracking output in which persistent IDs are overlaid on detected objects across video frames. This ID-preserving mechanism is essential for downstream temporal analysis because parking violation events must be determined based on the dwell time of the same vehicle, rather than frame-wise detections treated independently.



Figure 4. Example output of IoU-based tracking-by-detection with persistent object IDs and confidence scores overlaid on CCTV video frames.

Algorithm 2 determines whether a tracked vehicle should be classified as a parking violation (or stopping event) within a predefined restricted area. The algorithm operates on the outputs of Algorithm 1 and evaluates geometric overlap between tracked vehicle regions and user-defined ROI polygons (e.g., NO_PARK or NO_STOP zones). In addition to spatial overlap, a temporal dwell-time condition is imposed so that short-term stopping or transient occlusions do not immediately trigger violation alarms.

For each tracked vehicle, the bounding box is converted into a polygon representation and compared with the predefined ROI(s) using an IoU-based overlap criterion. If the overlap exceeds a threshold T_{roi} , the vehicle is considered a candidate match for a restricted region. The algorithm then computes the elapsed dwell time of the matched tracked object within the ROI. An event is emitted only when the dwell time exceeds a predefined threshold τ_{dwell} . This temporal filtering is designed to reduce false positives caused by vehicles temporarily slowing down, maneuvering, or passing through the ROI boundary. In the present deployment this overlap criterion is realized as a vehicle-center membership test against the parking-space ROIs (equivalent to membership in a NO_PARK/NO_STOP region), i.e., the special case with dwell-time threshold $\tau_{dwell} = 0$; Section 4.6 evaluates only this spatial test, whereas the dwell-time branch is the intended general formulation and is not yet quantitatively evaluated.

Algorithm 2: ROI-based Parking-Violation Detection

Input: updated_tracks, roi_set, current_time

Output: parking_violation_events

```

1: events ← ∅

2: for each track t in updated_tracks do

3:   veh_poly ← bbox_to_polygon(t.bbox)

4:   match ← None

5:   for each ROI_j in roi_set do

6:     if IoU(veh_poly, ROI_j.polygon) ≥ T_roi then

```

```

7:         match ← (ROI_j.label, j)           // e.g., "NO_PARK", "NO_STOP"
8:         break
9:     end if
10: end for
11: if match is not None then
12:     if t.enter_time is undefined then
13:         t.enter_time ← current_time
14:     end if
15:     dwell_time ← current_time - t.enter_time
16:     if dwell_time ≥ τ_dwell then
17:         emit_event(t.id, match, dwell_time, t.bbox, t.score)
18:         events ← events U {event}
19:     end if
20: else
21:     reset t.enter_time (or clear ROI association)
22: end if
23: end for
24: return events

```

Figure 5 illustrates an example of ROI-based parking-violation/stopping-area detection. By combining persistent object IDs from Algorithm 1 with ROI overlap and dwell-time constraints in Algorithm 2, the proposed approach is designed to achieve more stable event detection than frame-based rule matching alone. This design is particularly important in real-world outdoor parking lots, where detection confidence may fluctuate due to illumination changes, shadows, or partial occlusions.



Figure 5. Example of ROI-based parking-violation and stopping-area detection in the proposed NOVA Parking system.

3.5 Real-Time Monitoring and Notification Workflow

The proposed real-time parking monitoring system operates as an online pipeline that converts live CCTV streams into occupancy-aware visual outputs and user notifications. As shown in Figure 6, the workflow consists of four core processes: RTSP-based video input, parking-space detection and status analysis, empty-space counting, and real-time notification service.

In the Webcam Video Input stage, the system receives and decodes real-time RTSP streams from CCTV cameras installed in the parking lot. The decoded frames are then forwarded to the AI inference pipeline for frame-by-frame processing.

In the Parking Space Detection stage, the system detects vehicles and parking-space regions, extracts bounding boxes and class labels, and analyzes parking occupancy status based on detection and tracking results. The inferred status of each parking space is visually overlaid on the video stream to provide an annotated monitoring output.

In the Empty Parking Space Detection stage, the system aggregates per-space occupancy states to compute the current number of available parking spaces in real time. This step converts object-level detection outputs into service-level parking availability information.

Finally, in the Notification Service stage, parking availability updates are delivered to users through Firebase Cloud Messaging (FCM). When a parking-space status change is detected, the system generates an event and transmits a real-time notification to connected web/mobile clients.

Overall, this workflow integrates video acquisition, AI-based occupancy analysis, and event-driven notification into a unified smart parking monitoring service.

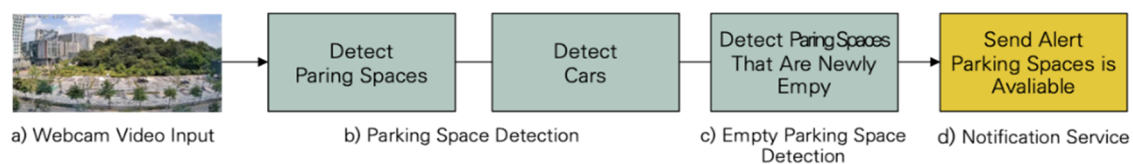


Figure 6. Real-time parking monitoring workflow of the proposed system, including RTSP-based CCTV video input, parking-space detection and status analysis, empty-space counting, and FCM-based real-time notification service.

4. Experimental Results

4.1 Experimental Setup

To evaluate the proposed system under diverse parking-lot conditions, two miniature outdoor parking-lot testbeds were constructed and instrumented with CCTV cameras for experimental validation (Figure 7).

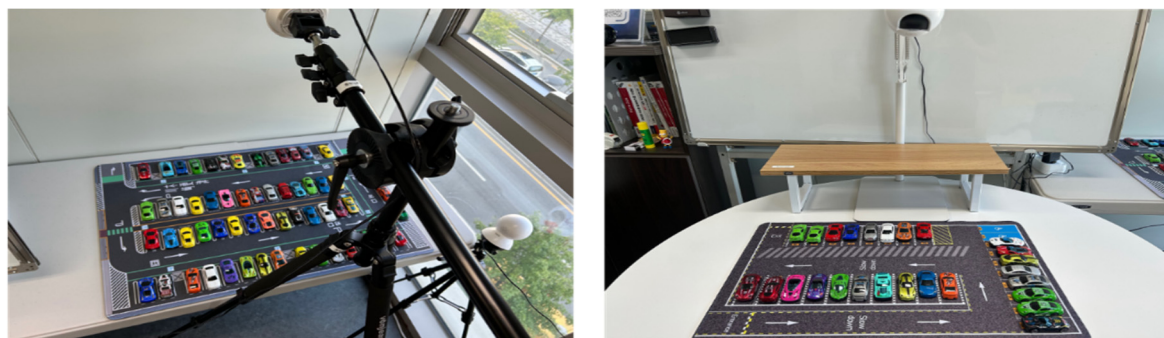


Figure 7. Miniature parking-lot testbeds used for experimental validation.

All models were trained and evaluated on a workstation equipped with Ubuntu 25.04, an NVIDIA RTX 5070 Ti graphics processing unit (GPU) (16 GB video random-access memory, VRAM), and 32 GB system memory (Table 1). The software environment included PyTorch, TensorFlow, and Docker, while system implementation and service modules were developed using PyCharm, Xcode, and Android Studio. The web/mobile user interface was implemented using React and Flutter, and real-time alert delivery was implemented via Firebase Cloud Messaging (FCM). The transformer-based detection engine used in the proposed system was Grounding DINO with a Swin-T backbone.

Table 1. Hardware and software environment used for model training, evaluation, and system implementation.

Category	Specification
OS	Ubuntu 25.04
GPU	NVIDIA RTX 5070 Ti (16 GB)
Memory	32 GB
Software	PyTorch, TensorFlow, Docker
Develop IDE	PyCharm, xCode, Android Studio
UI/UX	react/Flutter
Alert Message	Firebase FCM
Model	Grounding DINO (Swin-T backbone)

The closed-vocabulary CNN baselines (Faster R-CNN, Mask R-CNN, and YOLOv12) were fine-tuned on the full 11,177-image training split, whereas the open-vocabulary Grounding DINO was adapted from the released `groundingdino_swint_ogc` checkpoint using only 36 labeled images, exploiting its grounded pretraining for data-efficient adaptation. All detectors were then evaluated on the same held-out test images, including parking lots not seen during training, to compare cross-site generalization under a common protocol. The pretrained weights, fine-tuning, input resolution, data augmentation, confidence/IoU thresholds, and (for Grounding DINO) the text prompt of each model are summarized in Table 2.

Table 2. Training and inference configuration of the compared detectors.

Configuration item	Faster R-CNN / Mask R-CNN / YOLOv12	Grounding DINO (Swin-T)
Pretrained weights	Faster R-CNN: COCO-pretrained ResNet-50-FPN Mask R-CNN: ImageNet-pretrained R-50 YOLOv12: COCO-pretrained YOLOv12n	Swin-T with grounded pre-training
Adaptation to parking data	Fully fine-tuned on the 11,177-image training split	Box-supervised fine-tuning of the released checkpoint (Section 3.3)
Input resolution	YOLOv12: 640×640 Faster R-CNN / Mask R-CNN: 800×1,333	800×1,333
Data augmentation	YOLOv12: mosaic, horizontal flip, HSV color jitter, random erasing, scale/translate jitter Faster R-CNN: none Mask R-CNN: random flip and scale jitter	None (resize and normalization only)
Confidence threshold	YOLOv12: 0.25 Faster R-CNN: 0.5 Mask R-CNN: 0.05	box 0.90 / text 0.90 (auto-labeling) box 0.91 / text 0.90 (real-time deployment)
NMS IoU threshold	YOLOv12: 0.7 (training) / 0.5 (evaluation) Faster R-CNN / Mask R-CNN: 0.5 (framework defaults)	Class-wise NMS, IoU 0.3 vehicle-to-slot occupancy assignment at IoU ≥ 0.5 (auto-labeling, Eq. (2))
Text prompt	Not applicable	“empty . car .” for occupancy detection “car .” for initial vehicle labeling
Training schedule	YOLOv12: SGD, lr 0.01, momentum 0.9, batch 16, 100 epochs; Faster R-CNN: SGD, lr 0.005, momentum 0.9, batch 2, 10 epochs; Mask R-CNN: SGD, lr 2.5e-4, batch 2, 300 iterations	Adam, lr 1e-5, 2,000 epochs

Because the four detectors are built on different frameworks and were adapted with very different amounts of data, each was trained with its framework's standard, recommended recipe rather than a single shared schedule: the CNN baselines on the full 11,177-image training split (YOLOv12: SGD, lr 0.01, 100 epochs; Faster R-CNN: SGD, lr 0.005, 10 epochs; Mask R-CNN: base lr $2.5e-4$, 300 iterations), and Grounding DINO from the released `groundingdino_swint_ogc` checkpoint on only 36 labeled images (Adam, lr $1e-5$, 2,000 epochs). We note as a limitation that these epoch and iteration counts are not directly comparable across frameworks, that the hyperparameters were not exhaustively re-tuned per model, and that the training-set sizes differ greatly—an asymmetry that, if anything, favours the CNN baselines; the markedly smaller unseen-lot degradation of Grounding DINO (Section 4.4) therefore reflects the data efficiency of open-vocabulary grounding rather than equal-condition superiority.

4.2 Dataset

The dataset used in this study consists of 14,908 parking-lot images collected under diverse environmental conditions, including sunny, overcast, and rainy weather. The images were obtained from multiple parking-lot sites (UFPR04, UFPR05, and PUCPR), which exhibit different camera viewpoints, layout characteristics, and on-site visual conditions.

All images were annotated in YOLO format with vehicle bounding boxes and parking-space occupancy labels (Empty and Occupied). The dataset was split into 11,177 training images (75.0%), 2,486 validation images (16.7%), and 1,245 test images (8.3%) for model development and evaluation.

In addition, an automatic labeling dataset was constructed using 36 images captured from Mini Parking Lot Set #1 and Set #2. Through the proposed automatic labeling pipeline, each image was annotated with vehicle bounding boxes and parking-space occupancy labels. All 36 images were used for training in the automatic labeling experiment, and 7 separately captured images were used as a held-out test set to evaluate generalization performance.

The two datasets play complementary roles. The 14,908-image benchmark (UFPR04, UFPR05, and PUCPR) provides the 11,177-image training split that fine-tunes the CNN baselines and the held-out images that evaluate all detectors (Tables 3 and 4). The 36-image testbed set, auto-labeled by the pipeline of Section 3.3, serves a dual role—adapting Grounding DINO to the deployment domain with only a few labels (Section 4.1) and validating the automatic labeling pipeline (Section 4.5)—with seven further images held out for on-site evaluation. These 36 images are not part of the benchmark but constitute the small, data-efficient adaptation set used by the open-vocabulary detector.

4.3 Baseline Performance on Trained Environments

Table 3 summarizes the detection performance of Faster R-CNN, Mask R-CNN, YOLOv12, and Grounding DINO on parking lots included in the training data. The training and inference conditions for all compared detectors are summarized in Table 2. As discussed in Section 4.1, the two model families were adapted with very different amounts of data (36 images for Grounding DINO versus 11,177 for the CNN baselines), so the comparison contrasts each detector under its standard configuration rather than under identical training. Because Grounding DINO was not trained on any benchmark lot, both the “trained” and “unseen” environments are effectively unseen to it, and its in-domain accuracy reflects open-vocabulary transfer.

As shown in Table 3, YOLOv12 achieved the highest overall performance on trained environments in terms of precision, recall, F1-score, mean Average Precision (mAP), and inference speed (118.23 frames per second, FPS). Grounding DINO achieved competitive performance, with Precision = 0.9567, Recall = 0.9834, F1 = 0.9698, mAP@0.5 = 95.23%, and mAP@0.5:0.95 = 89.45%, while maintaining practical inference speed (42.15 FPS) for real-time monitoring applications.

These results indicate that, although CNN-based detectors (particularly YOLOv12) can achieve very high accuracy in trained environments, the transformer-based Grounding DINO model also provides strong baseline detection performance under in-distribution conditions.

Table 3. Detection performance on trained parking lots

Model	Prec.	Recall	F1	mAP@.5	mAP@.5:.95	FPS
Faster R-CNN	0.9886	1.0	0.9942	99.01	98.71	29.68
Mask R-CNN	0.9442	0.9754	0.9595	84.76	42.99	23.70
YOLOv12	0.9989	0.9986	0.9987	99.46	98.74	118.23
Grounding DINO	0.9567	0.9834	0.9698	95.23	89.45	42.15

4.4 Generalization to Unseen Environments

To evaluate cross-environment robustness, all models were further tested on parking lots withheld from training. Table 4 compares detection accuracy (mAP@0.5) across trained and unseen deployment environments.

Grounding DINO maintained 94.12% mAP@0.5 on unseen parking lots, compared with 95.23% on trained lots, preserving approximately 99% of its trained-lot performance. In contrast, YOLOv12 decreased from 99.46% to 81.67% (a 17.79 percentage-point drop), and Faster R-CNN decreased from 99.01% to 78.34% (a 20.67 percentage-point drop). These results demonstrate that the proposed transformer-based approach provides substantially stronger generalization to unseen environments than CNN-based baselines.

The superior robustness of Grounding DINO is attributed to its vision-language grounding mechanism, which encourages the detector to rely on semantically meaningful features rather than site-specific texture patterns (e.g., pavement appearance, lane markings, and illumination artifacts). This property is particularly beneficial in smart parking deployments, where camera placement and environmental conditions vary significantly across sites. As discussed in Section 4.1, the two model families were adapted with very different amounts of data, so these cross-site gains reflect the data efficiency of open-vocabulary grounding under each detector’s standard configuration rather than superiority under identical training conditions.

Table 4. Detection accuracy across deployment environments (mAP@0.5). The GD vs. YOLOv12 (rel.) column reports the relative change of Grounding DINO mAP@0.5 with respect to the strongest CNN baseline (YOLOv12), computed as $(\text{mAP_GD} - \text{mAP_YOLOv12}) / \text{mAP_YOLOv12} \times 100\%$; positive values indicate that Grounding DINO is more accurate.

Environment	Grounding DINO	Faster R-CNN	YOLOv12	GD vs. YOLOv12 (rel.)
Trained lot	95.23%	99.01%	99.46%	-4.2%
Unseen lot	94.12%	78.34%	81.67%	+15.2%

4.5 Automatic Labeling Quality Evaluation

To assess the reliability of the automatic labeling pipeline (Section 3.3), the labels it generates for the mini-testbed images were verified against human-checked ground truth. The ground truth was established by a single expert annotator (the first author): each automatically generated box was rendered as a colour-coded overlay on its source frame (vehicles in green, empty spaces in red) and inspected individually, and any box whose object class or occupancy state did not match the visible scene was corrected before being accepted as ground truth. Because the verification was performed by one annotator, no inter-annotator disagreement arose and no consensus procedure was required; the use of a single annotator rather than multiple independent annotators with a formal agreement protocol is acknowledged as a limitation of the present evaluation. Three metrics were used: (i) the mean intersection-over-union (IoU) between each automatically generated (Grounding DINO) vehicle box and the corresponding SAM-refined tight bounding box obtained by box-prompted segmentation (Section 3.3), which quantifies how closely the reported boxes already agree with SAM-refined boxes—and hence the magnitude of localization change that SAM refinement would introduce—rather than a detection-accuracy gain; (ii) label accuracy, the proportion of automatically generated boxes whose object class matches the ground truth; and (iii) Empty/Occupied classification accuracy, the proportion of parking spaces whose occupancy state is correctly assigned by the IoU-based rule of Eq. (2) with threshold 0.5. Table 5 reports these metrics on the 36 training images and the 7 held-out test images; the test images include frames captured from a changed camera viewpoint, for which the parking-space geometry was re-estimated by the aggregation rule of Eq. (1) before occupancy assignment.

Table 5. Quality of the automatically generated labels against human-verified ground truth, established by a single expert annotator (see Section 4.5).

Metric	Train set (36 images)	Test set (7 images)
Mean bounding-box IoU (vehicle)	0.937	0.934
Label accuracy (%)	100.0	100.0
Empty/Occupied accuracy (%)	100.0	100.0
No. of parking spaces evaluated	2,124	413

As shown in Table 5, the automatically generated labels achieve a mean bounding-box IoU of 0.934 and an Empty/Occupied classification accuracy of 100.0% on the held-out test images, indicating that the pipeline produces labels of sufficient quality to bootstrap site-specific dataset construction with substantially reduced manual effort. These results should be read in light of the limited scale of this evaluation: the held-out set is only seven miniature-testbed images verified by a single annotator (thirteen additional frames—ten sparse-occupancy captures with too few vehicle observations for the Eq. (1) geometry estimation and three night-time frames outside the daytime parking-space template—were excluded), so the perfect accuracy reflects controlled conditions, and a larger, more diverse evaluation with multiple independent annotators is left for future work. Moreover, because the reported boxes already lie close to their SAM-refined counterparts and occupancy is decided by a coarse IoU rule (Eq. (2)), SAM serves as auxiliary label refinement rather than a performance driver: we did not run a controlled Grounding DINO-only versus Grounding DINO + SAM comparison and claim no separate detection gain from SAM.

4.6 ROI-Based Violation Detection Evaluation

Beyond per-frame occupancy accuracy, the practical value of the NOVA Parking algorithms (Section 3.4) depends on the reliability of parking-violation detection. The spatial component of Algorithm 2 classifies each detected vehicle as legally parked—its center lies inside a parking-space region of interest (ROI), where the ROIs are obtained automatically from the parking-space geometry of Section 3.3—or as a violation, when the vehicle is stopped outside all parking ROIs (for example in a driving lane, crosswalk, or no-parking zone). This spatial rule was evaluated at the frame level on a 37-frame image sequence of the testbed spanning normal occupancy and staged violation situations. Every detected vehicle was visually checked against the predicted state to obtain ground truth, and precision, recall, and F1-score of violation detection were computed. The results are summarized in Table 6, and a representative example is shown in Figure 8.

Table 6. Frame-level performance of the spatial ROI-membership parking-violation rule (the dwell-time component of Algorithm 2 is not included).

Metric	Value
No. of frames evaluated	37
No. of vehicle instances	554
No. of ground-truth violations	12
Precision	1.00
Recall	1.00
F1-score	1.00

As shown in Table 6, the ROI-based rule correctly flagged all 12 out-of-slot vehicles among 554 vehicle instances with no false alarms ($F1 = 1.00$). The perfect separation reflects the clear spatial distinction between marked parking spaces and driving lanes in the testbed; performance in dense real-world lots with ambiguous boundaries is expected to be lower. We emphasize that this evaluation assesses only the instantaneous spatial ROI-membership rule. The dwell-time-based violation detection of Algorithm 2 (Section 3.4)—which distinguishes momentary stops from sustained violations and requires timestamped video—was not exercised in these experiments and has therefore not been quantitatively verified; its evaluation is left as future work (Section 5).

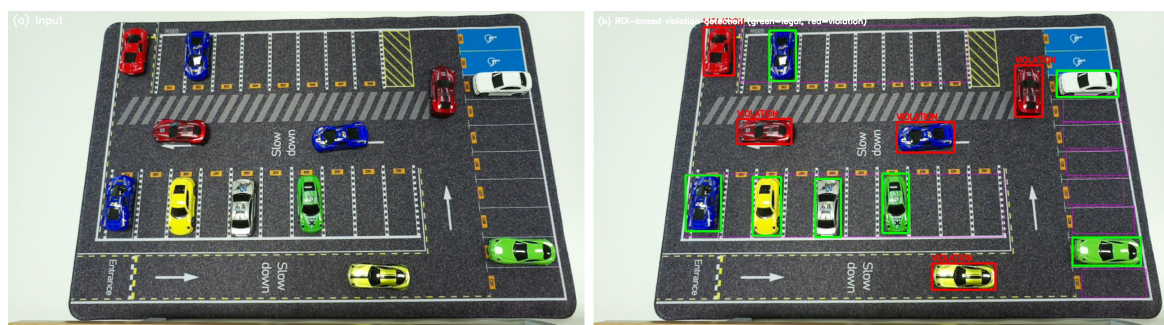


Figure 8. ROI-based parking-violation detection on the testbed: (a) input frame; (b) detection result.

4.7 Cost-Benefit Analysis

To assess practical deployment feasibility, we compared the total cost of ownership (TCO) of the proposed **camera-based AI system** with that of a **wireless magnetometer-based parking monitoring system** for a representative **100-space outdoor parking lot** over a five-year period (Table 7).

As summarized in Table 7, the magnetometer-based system incurs high initial hardware and installation costs because a sensor must be installed for each parking space, typically requiring in-ground construction work. In addition, annual maintenance costs remain high due to battery replacement and sensor maintenance. In contrast, the proposed vision-based system requires only **4-8 cameras** to monitor the same lot, leading to lower initial hardware costs, simplified installation, and reduced maintenance costs, which are primarily software-oriented.

Based on the estimated values in Table 7, the proposed system reduces the **five-year TCO** from approximately **USD 120,000** to **USD 35,000**, corresponding to an estimated **71% cost reduction**. This result highlights the economic advantage of camera-based AI monitoring, particularly in large-scale outdoor parking facilities where scalability is a major concern.

Table 7. Five-year cost comparison: camera-based AI vs. wireless magnetometer (100 spaces)

Cost Category	Magnetometer System	Proposed AI System
Hardware (initial)	High (sensor × 100)	Low (4-8 cameras)
Installation labor	High (in-ground work)	Low (surface mount)
Annual maintenance	High (battery, sensor)	Low (software update)
Scalability	Linear (cost per space)	Sub-linear (camera FOV)
Estimated 5-yr TCO	~USD 120,000	~USD 35,000 (-71%)

The vision-based approach achieves an estimated 71% reduction in five-year TCO, primarily because camera hardware covers multiple spaces, and maintenance is limited to software rather than embedded sensor hardware.

To make the basis of this estimate explicit, Table 8 itemizes the five-year total cost of ownership for the two systems on a representative 100-space outdoor lot. The figures are representative market-based estimates in 2025 USD and can be adjusted to local prices; electricity, which is comparable for both systems, is excluded.

Table 8. Itemized five-year total cost of ownership (TCO) for a 100-space outdoor lot (2025 USD).

Cost item	Unit basis / assumption	Magnetometer	Proposed AI
Sensing hardware	Mag.: 100 in-ground sensors @ \$100; AI: 6 outdoor PoE cameras @ \$400	10,000	2,400
Installation labor	Mag.: in-ground coring @ \$300/space; AI: surface mount @ \$300/camera	30,000	1,800
Network infrastructure	Gateways/PoE switch and cabling	8,000	2,000
Edge/GPU server	One edge-GPU inference server (AI only)	–	8,000
Connectivity (5 yr)	Mag.: \$2,000/yr; AI: \$600/yr	10,000	3,000
Maintenance (5 yr)	Mag.: battery+sensor \$30/space/yr + 10%/yr replacement; AI: software/service \$1,500/yr	20,000	7,500
Central server & software (5 yr)	Platform, integration, and operation	27,000	10,300
Commissioning / contingency	Site survey, testing, spares (mag. only)	15,000	–
Five-year TCO	Sum of the above	~120,000	~35,000

Under these assumptions the proposed system reduces the five-year TCO from approximately USD 120,000 to USD 35,000, a reduction of about 71% $((120,000 - 35,000)/120,000)$. The advantage stems mainly from the sub-linear scaling of camera coverage—a few cameras observe the entire lot, whereas the magnetometer system requires per-space sensing hardware, in-ground installation, and recurring battery/sensor maintenance.

4.8 Implementation of the NOVA Parking Web and Mobile Application

To improve usability and service accessibility, the proposed NOVA Parking system was implemented as an integrated web and mobile application platform. The platform enables both administrators and general users to access real-time parking information and supports efficient parking monitoring and operation management (Figures 9 and 10).

The **web dashboard** provides functions for real-time parking-lot monitoring, status visualization, and system management, allowing administrators to inspect occupancy conditions and monitor live CCTV streams. The **mobile interface** provides user-oriented access to parking availability information and real-time notification services, thereby improving convenience in practical parking scenarios.

These implementation results demonstrate that the proposed system is not limited to offline detection experiments, but can be deployed as a practical smart parking monitoring service with real-time visualization and user interaction capabilities.

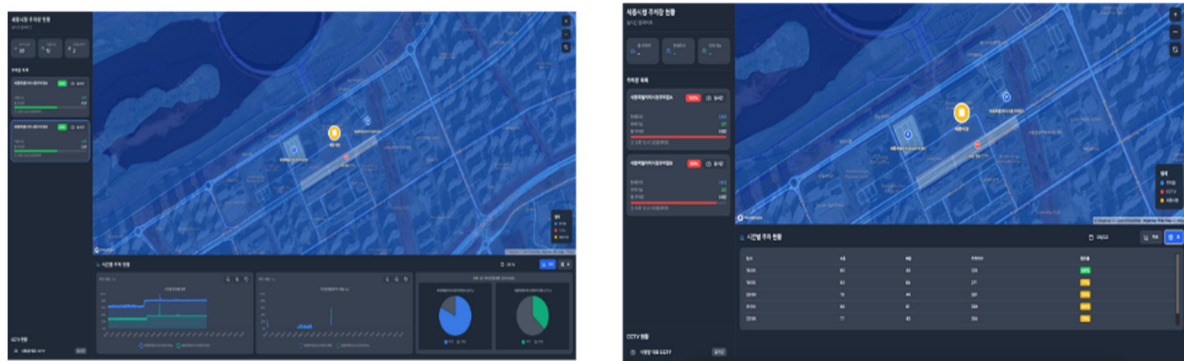


Figure 9. Parking-lot dashboard interface of the proposed NOVA Parking system.

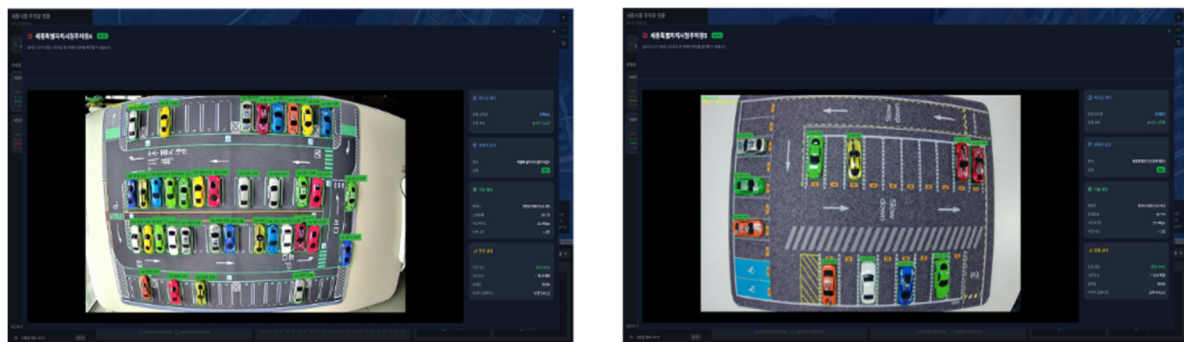


Figure 10. Real-time parking-lot monitoring interface with occupancy-aware visualization.

5. Conclusion and Future Work

This paper presented NOVA Parking, a transformer-based smart parking monitoring system designed to improve cross-environment robustness in camera-based parking surveillance. The proposed system addresses a key limitation of conventional CNN-based object detectors—namely, performance degradation in previously unseen parking environments—by adopting Grounding DINO, an open-vocabulary vision-language transformer detector that aligns textual prompts with visual features. This design enables semantically guided object detection and contributes to more stable parking occupancy monitoring under variations in camera viewpoint, pavement markings, and illumination conditions.

The main contributions of this work can be summarized as follows. First, we developed an automatic labeling pipeline—combining open-vocabulary Grounding DINO detection, geometric parking-space estimation, and rule-based occupancy assignment, with SAM box-prompted segmentation as an optional label-refinement step—that significantly reduces manual annotation effort and supports efficient dataset construction for parking-space occupancy monitoring. Second, we proposed an IoU-based tracking-by-detection algorithm to maintain temporally consistent vehicle identities in CCTV video streams, which is essential for robust state analysis and event generation. Third, we formulated a parking-violation analysis algorithm (Algorithm 2) that combines ROI-based spatial matching with a dwell-time constraint; in this work we quantitatively evaluated its

spatial ROI-membership component (Section 4.6, Table 6), whereas the dwell-time component is presented as the general formulation and is left for future quantitative validation. Finally, we integrated these components into a practical web/mobile smart parking service platform with real-time visualization and FCM-based notification, demonstrating end-to-end deployability beyond offline detection experiments.

Experimental results showed that the proposed transformer-based approach provides strong generalization performance in unseen parking lots. In particular, Grounding DINO maintained high detection accuracy on withheld deployment environments and outperformed CNN-based baselines in cross-site robustness. Compared with YOLOv12 on unseen lots, the proposed approach achieved an improvement of approximately 12.45 percentage points in mAP@0.5 (equivalently, approximately 15.2% relative improvement, computed as $(94.12 - 81.67)/81.67 \times 100\%$). In addition, the cost-benefit analysis indicated that the camera-based AI monitoring system can substantially reduce the five-year total cost of ownership (TCO) compared with conventional wireless magnetometer-based deployments, with an estimated reduction of approximately 71% for a representative 100-space outdoor parking lot.

Despite these promising results, several limitations remain, and the future directions below are organized to address them directly. First, the current system relies on a single-camera viewpoint per lot, so it is susceptible to blind spots and inter-vehicle occlusion in dense or multi-lane layouts. Second, the detector reports occupancy and violation states but not vehicle identity or permit status, which limits enforcement-grade services. Third, real-time throughput and end-to-end latency depend on the available hardware and on edge-to-cloud network conditions, which may vary across deployment sites. Fourth, the region-of-interest definitions used for violation analysis are specified in fixed image coordinates, so they must be re-specified when the camera viewpoint changes; automatic adaptation of the ROIs to viewpoint changes is not yet integrated into the online pipeline. Fifth, the labeling and violation-detection accuracies (Sections 4.5 and 4.6) were established on small, single-annotator testbed sets, so the perfect scores are evidence on small, internally verified sets rather than a guarantee of large-scale field performance.

Future work will therefore focus on five directions, each tied to a limitation above and to the supporting evidence in the experiments and implementation that exposed it (e.g., the cross-site evaluation of Section 4.4, the dense-lot caveat of Section 4.6, and the viewpoint-dependent geometry re-estimation noted in Section 4.5). First, to mitigate the blind spots and occlusion inherent to the single-camera configuration, multi-camera sensor fusion will be investigated to extend coverage and reconstruct occluded regions in large-scale facilities. Second, to overcome the lack of vehicle identity, license-plate recognition (LPR) and permit-aware enforcement will be incorporated to support advanced parking-management and violation-handling services. Third, to address the hardware-dependent throughput and edge-to-cloud latency, edge deployment with model optimization (quantization, pruning, and distillation) will be explored to reduce inference latency, minimize cloud dependency, and improve operational scalability in real-world intelligent transportation system (ITS) deployments. Fourth, to remove the dependence on fixed ROI coordinates, the parking-space geometry estimation of Section 3.3 (Eq. (1)) will be extended to regenerate the violation ROIs automatically from detections whenever the camera viewpoint changes, enabling viewpoint-robust violation analysis without manual re-specification. Fifth, a larger, more diverse evaluation—spanning illumination, occlusion, vehicle sizes, lane markings, and viewpoints—together with a multi-annotator agreement study and a quantitative evaluation of the dwell-time-based violation detection of Algorithm 2 on timestamped video will be conducted to establish external validity.

Acknowledgments: This work was supported by the Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201462) and was conducted during the research year of Chungbuk National University in 2026.

Conflicts of Interest: The authors declare no conflict of interest.

References

- [1] INRIX, Searching for parking costs the UK £23.3 billion a year (2017). Accessed: May. 18, 2018. [Online] Available: <https://inrix.com/press-releases/parking-pain-uk/>
- [2] A. Fahim, M. Hasan, and M. A. Chowdhury, “Smart parking systems: comprehensive review based on various aspects,” *Heliyon*, vol. 7, no. 5, p. e07050, 2021, doi: <https://doi.org/10.1016/j.heliyon.2021.e07050>.

- [3] V. Paidi, H. Fleyeh, J. Håkansson, and R. G. Nyberg, "Smart parking sensors, technologies and applications for open parking lots: a review," *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 735-741, 2018, doi: <https://doi.org/10.1049/iet-its.2017.0406>.
- [4] T. Perković, P. Šolić, H. Zargiasl, D. Čoko, and J. J. P. C. Rodrigues, "Smart parking sensors: state of the art and performance evaluation," *Journal of Cleaner Production*, vol. 262, p. 121181, 2020, doi: <https://doi.org/10.1016/j.jclepro.2020.121181>.
- [5] L. Lou, J. Zhang, Y. Xiong, and Y. Jin, "An improved roadside parking space occupancy detection method based on magnetic sensors and wireless signal strength," *Sensors*, vol. 19, no. 10, p. 2348, 2019, doi: <https://doi.org/10.3390/s19102348>.
- [6] D. Acharya, W. Yan, and K. Khoshelham, "Real-time image-based parking occupancy detection using deep learning," in *Proceedings of the 5th Annual Conference of Research@Locate, CEUR Workshop Proceedings*, vol. 2087, pp. 33-40, 2018. [Online] Available: <https://ceur-ws.org/Vol-2087/paper5.pdf>
- [7] C. Polprasert, C. Sruiyiam, P. Pisawongprakan, and S. Teravetchakarn, "A camera-based smart parking system employing low-complexity deep learning for outdoor environments," in *Proc. 2019 17th International Conference on ICT and Knowledge Engineering (ICT&KE)*, 2019, doi: <https://doi.org/10.1109/ICTKE47035.2019.8966901>.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017, doi: <https://doi.org/10.1109/TPAMI.2016.2577031>.
- [9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980-2988, doi: <https://doi.org/10.1109/ICCV.2017.322>.
- [10] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025, doi: <https://doi.org/10.48550/arXiv.2502.12524>.
- [11] G. P. C. P. da Luz, G. M. Sato, L. F. G. Gonzalez, and J. F. Borin, "Smart parking with pixel-wise ROI selection for vehicle detection using YOLOv8, YOLOv9, YOLOv10, and YOLOv11," *Internet of Things*, vol. 36, p. 101858, 2026, doi: <https://doi.org/10.1016/j.iot.2025.101858>.
- [12] P. R. L. de Almeida, L. S. Oliveira, A. S. Britto Jr., E. J. Silva Jr., and A. L. Koerich, "PKLot - a robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937-4949, 2015, doi: <https://doi.org/10.1016/j.eswa.2015.02.009>.
- [13] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Systems with Applications*, vol. 72, pp. 327-334, 2017, doi: <https://doi.org/10.1016/j.eswa.2016.10.055>.
- [14] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: marrying DINO with grounded pre-training for open-set object detection," in *Proc. European Conference on Computer Vision (ECCV)*, LNCS, Springer, 2024, doi: https://doi.org/10.1007/978-3-031-72970-6_3.
- [15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. European Conference on Computer Vision (ECCV)*, LNCS, vol. 12346, Springer, pp. 213-229, 2020, doi: https://doi.org/10.1007/978-3-030-58452-8_13.
- [16] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 3992-4003, doi: <https://doi.org/10.1109/ICCV51070.2023.00371>.
- [17] H. I. Chong and K. H. Yoo, "An intelligent parking monitoring system using transformer-based object detection," in *Proceedings of AIR-RES 2026, Las Vegas, USA*, 2026.
- [18] J. So, N. Kim, and J. H. Jeong, "A parking space identification system based on entry and exit data for an efficient parking environment," *The Journal of The Institute of Internet, Broadcasting and Communication*, vol. 24, no. 4, pp. 195-200, 2024, doi: <https://doi.org/10.7236/JIIBC.2024.24.4.195>.

