Can I Park Here? Understanding Complicated Parking Signs on the Street

Junlai Shi, Yuankai He, Yuxin Wang, Weisong Shi
Department of Computer and Information Science, University of Delaware, Newark, USA
{junlai, willhe, yuxw, weisong}@udel.edu

***Abstract*—Parking is an important component of urban traffic management, and accurate recognition of parking sign rules is essential for intelligent parking. Although computer vision has advanced in traffic sign recognition, parking signs differ as they often include free-form text, symbols, and stacked formats, making understanding more complex. Current research on parking signs primarily focuses on signs detection, while studies on rule understanding remain insufficient. To address this problem, we propose a parking rule checking (PaRC) service designed to understand complicated parking signs and support parking decision making. PaRC is an on-device hybrid framework with (1) a YOLOv11n-based Signs Detection (SD) module to extract parking sign regions, (2) a fine-tuned Donut–driven Rule Parsing (RP) module to parse structured rules, and (3) an interpretable program-based Decision Making (DM) module to produce the final parking decision. We construct datasets by consolidating multiple U.S. street-view parking sign collections. PaRC achieves an F1-score of 96.15% with an inference time of 0.90 s, surpassing on-device models in speed while maintaining accuracy comparable to that of LVLMs. Most remaining errors arise from severely blurred signs that even humans struggle to recognize, and PaRC also performs well on low-end devices.**

***Index Terms*—autonomous driving, intelligent parking, hybrid framework, parking sign understanding, on-device inference, interpretable decision making**

I. Introduction

Parking is an essential part of urban transportation, directly influencing both traffic safety and efficiency. According to an INRIX research report [1], drivers in the United States spend an average of 17 hours per year searching for parking, with nationwide parking-related issues causing an estimated economic loss of 73 billion USD annually. These losses arise not only from wasted time but also from fines, enforcement costs, and potential safety risks associated with illegal parking. To mitigate the burden that parking problems impose on the economy and quality of life, the development of intelligent parking solutions is imperative.

Accurate recognition of parking signs is central to intelligent parking systems, as it determines whether a vehicle can be parked legally and safely. In many urban environments, parking sign regulations can be highly confusing. Their colors, free-form text, and time-based restrictions often combine to define parking rules, and even human drivers frequently struggle to interpret them. In addition, multiple parking signs are sometimes stacked together [2], making it even more difficult to interpret the rules, as shown in Fig. 1. For example, in the United Kingdom, an ITV News survey [3] reported that 81% of drivers found parking signs confusing, and approximately

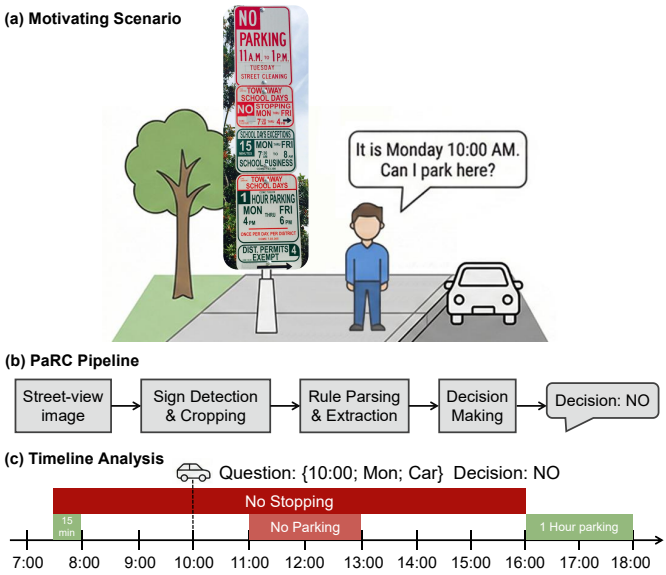

Fig. 1. Illustration of the parking sign interpretation challenge and PaRC. (a) Motivating Scenario: A driver encounters a complex parking pole with stacked signs and queries the feasibility of parking at that time (Monday 10:00 AM). (b) PaRC Pipeline: PaRC processes the street-view image through sign detection, rule parsing, and decision-making modules to formulate an answer. (c) Timeline Analysis: A schematic timeline illustrates the complex parking rules for this scene. The "No Stopping" rule (red bar) covers the queried time, leading to the "No" decision.

30% of parking ticket appeals cited unclear signage as the primary reason. This phenomenon is not unique to the UK. In the United States, drivers have similarly criticized multilayered parking signs as "towering, unintelligible behemoths [4]" while major cities such as Los Angeles [2], [5] have explicitly acknowledged the confusing nature of these signs in official documents and initiated projects aimed at "simplifying parking signs [4], [5]." This implies that if even human drivers find it difficult to accurately interpret these rules, the challenge for autonomous driving systems is even greater.

In recent years, computer vision techniques have achieved remarkable progress in traffic sign recognition [6]–[8]. Conventional traffic signs such as speed limits or prohibitions typically possess highly standardized shapes and symbols, with clear information and unambiguous semantics, allowing machines to reliably detect and interpret their meanings. Parking signs, however, present unique challenges. Unlike conventional

pictorial signs, parking signs often convey rules through a combination of text and symbols, such as "Parking only from 7 a.m. to 7 p.m. on weekdays," sometimes accompanied by directional arrows. This requires autonomous systems not only to perform high-accuracy optical character recognition (OCR) and symbol detection, but also to incorporate natural language processing (NLP) techniques to parse the underlying semantics. Furthermore, in real-world scenarios, multiple signs are often colocated, jointly constraining parking conditions. For instance, one sign may restrict parking to limited hours on weekdays, while another imposes an additional condition that only resident vehicles are permitted. Autonomous systems must reason over the logical relationships between such signs, and integrate contextual information (e.g., time, date, vehicle type) to determine parking feasibility. This ability is essential for ensuring safe and rules-compliant autonomous driving.

Although computer vision and deep learning have made substantial progress in traffic sign recognition, existing research has largely concentrated on conventional signs (e.g., speed limits, prohibitions), which are standardized and relatively simple. In contrast, research on parking signs remains relatively limited. Existing studies primarily focus on detection and classification tasks, such as leveraging YOLO-based networks for localization and categorization of parking signs in street-view images [9], [10], or incorporating text recognition to improve the readability of sign content [11]. While these methods achieve progress in visual recognition and model efficiency, they largely remain at the level of appearance or literal information, lacking systematic semantic understanding of the parking rules. Furthermore, when multiple parking signs are stacked or when rules involve dynamic constraints such as time or vehicle type, these approaches are unable to perform logical reasoning and integrated decision-making, limiting their applicability for providing comprehensive parking decisions in autonomous driving systems.

To address these challenges, this paper proposes PaRC, a parking rule checking service for parking sign understanding and informed parking decision making. Starting from raw video input, the on-device framework performs parking sign detection, semantic parsing of the rules into structured representations, and final decision making regarding parking feasibility. Unlike previous studies that focus solely on sign detection, this work instead emphasizes the complete pipeline from perception to understanding and decision making, thereby aiming to achieve a truly deployable solution for intelligent parking in real-world settings.

We construct the dataset for PaRC's sign detection model by merging multiple publicly available U.S. street-view parking sign datasets. Furthermore, we systematically parse the textual rules contained in parking signs and convert them into structured JSON representations, which serve as reliable ground truth for training the rule understanding model. Although PaRC focuses on U.S. parking signs, the proposed framework is readily generalizable to other regions, requiring mainly region-specific data for effective adaptation. The main contributions of PaRC are listed below:

- **On-device Service:** We present an on-device service for parking sign understanding and decision-making. To the best of our knowledge, this is the first complete work specifically targeting semantic understanding of parking rules and automated decision making in the context of autonomous driving.
- **Interpretable Decision-Making:** We design a three-module pipeline: Signs Detection (SD), Rule Parsing (RP), and Decision Making (DM). SD extracts parking signs from raw video frames, RP parses each sign into structured semantic rules, and DM makes decisions using these rules and environmental context. Together, they form a transparent workflow that translates raw visual inputs into parking decisions, as shown in Fig. 1.
- **Superior Performance:** We conduct experiments to evaluate the effectiveness of PaRC. It demonstrates superior performance to existing LVLMs and open-source edge-deployable models in terms of both accuracy score and inference time on parking sign understanding and decision-making tasks. Notably, most errors occur in low-resolution or severely blurred images that are also indistinguishable to human drivers.

The remainder of this paper is organized as follows: Section II reviews the related work, Section III analyzes the current problems and challenges, Section IV discusses the design of PaRC, Section V presents the implementation and evaluation of PaRC, and Section VI provides the conclusion and outlook on future work.

II. Related Work

This section first reviews existing studies on parking sign recognition, and then focuses on its core components: sign detection and rule understanding. Given the limited research specifically dedicated to parking signs, we draw upon the more established body of work in traffic sign recognition when discussing sign detection, as the principles of detection are largely consistent across both domains.

A. Parking Sign Recognition

Recent studies on parking sign recognition can be broadly categorized into two main aspects: dataset construction and different approaches.

1) Dataset: Dataset construction has long been a major challenge for model training. Conventional approaches, such as manual collection or extraction from street-view images, are often inefficient and limited in scale [9], [10]. Irshad et al. [12] proposed a framework that combines active learning, transfer learning, and crowdsourcing tools to efficiently expand training datasets and mitigate data imbalance issues caused by the variability of parking signs in shape, color, scale, and background in street-level imagery.

2) Approaches: Existing studies on parking sign recognition primarily employ deep learning techniques for detection and classification. Faraji et al. [9] used YOLOv7X on a custom dataset to address challenges such as small sign size and visual diversity, but the work remains limited to appearance-level

recognition, lacking rule understanding. Yin Jin et al. [10] proposed a complete pipeline from raw images to detection and rule understanding, implementing real-time detection and model lightweighting with YOLOv5 while benchmarking multiple variants. However, the emphasis was still on detection models, with no in-depth exploration or implementation of rule understanding. Zhong et al. [11] extended detection with text recognition to enhance the readability of sign content, yet this approach remained at the lexical level without achieving logical reasoning over parking rules. Overall, these studies predominantly concentrate on detection and classification, while systematic understanding and reasoning over the complex rules encoded in parking signs remain largely underexplored.

B. Signs Detection

Traffic sign detection has long been a fundamental task in autonomous driving and intelligent transportation systems. Traditional approaches relied on handcrafted features and classical machine learning methods, but they often suffered from limited robustness to variations in illumination, viewpoint, and occlusion. With the rapid advancement of deep learning, convolutional neural networks (CNNs) and object detection frameworks have become the dominant paradigm. Methods based on YOLO [6], [13] and R-CNN series [8], [14], have demonstrated high accuracy and real-time performance in detecting and classifying standard traffic signs, such as speed limits, stop signs, and no-entry signs [15]. These approaches leverage large annotated datasets, feature hierarchies, and end-to-end learning to achieve stable recognition under diverse traffic conditions. However, while effective for standardized traffic signs, these methods do not directly address the additional challenges posed by parking signs, which often contain textual information, multi-part symbols, and context-dependent rules that require deeper semantic understanding.

C. Rule Understanding

Transforming visual information into structured semantic rules is a critical step in understanding parking signs. Existing approaches can be broadly divided into three categories:

1) OCR + Sequence-to-Sequence: Early approaches typically rely on a combination of optical character recognition (OCR) and sequence-to-sequence models. In this paradigm, textual sequences are first extracted from images using traditional OCR tools (e.g., Tesseract [16]) or deep learning-based OCR models (e.g., PaddleOCR [17] and Microsoft Azure Vision [18]).

After OCR extraction, the text remains at the character level and cannot adequately capture the underlying semantic rules. To achieve precise understanding, it must be further transformed into a structured representation that can be processed by computers. This transformation can be accomplished using sequence-to-sequence models (e.g., BART [19]) or grammar-based rule parsers, which map linear character sequences into semantically explicit structured forms.

This two-stage pipeline offers modularity and benefits from mature text recognition techniques. However, it suffers from several drawbacks: recognition errors in OCR can significantly propagate to downstream semantic parsing, thereby increasing the overall error rate; moreover, parking signs often contain both text and symbols, which require additional symbol recognition models. The separation of visual and linguistic modeling may also fail to capture contextual dependencies among symbols, text, and spatial layout.

TABLE I
COMPARISON OF RELATED WORK BY TASK-LEVEL CAPABILITIES IN PARKING SIGN RECOGNITION.

Method	DC	SR	TR	RP	SU	MSR
A. Parking Sign Recognition						
Irshad et al. [12]	△	✓	×	×	×	×
Faraji et al. [9]	△	✓	×	×	×	×
Jin et al. [10]	✓	✓	△	△	×	×
Zhong et al. [11]	×	×	✓	×	×	×
B. Signs Detection						
YOLO-based [6], [13]	×	✓	×	×	×	×
R-CNN-based [8], [14]	×	✓	×	×	×	×
C. Rule Understanding						
OCR Systems [16], [17]	×	×	✓	×	×	×
Seq2Seq Models [19]	×	×	✓	✓	△	×
E2E Parsers [20]	×	✓	✓	✓	✓	△
LVLMs (cloud/edge)	×	✓	✓	△	✓	✓
PaRC	✓	✓	✓	✓	✓	✓

Abbreviations: **DC**: Dataset Construction, **SR**: Symbol Recognition, **TR**: Text Recognition, **RP**: Rule Parsing, **SU**: Semantic Understanding, **MSR**: Multi-Sign Rule Reasoning.
Symbols: ✓: Fully Supported, △: Partially Supported, ×: Not Supported.

2) End-to-End Parsers: To mitigate pipeline errors, a distinct paradigm of End-to-End (E2E) Parsers has evolved. Epitomized by Vision Transformer-based architectures like Donut [20], these models unify visual feature extraction and sequence modeling into a single differentiable network. By exploiting latent correlations among text, symbols, and spatial layouts, they effectively perform Rule Parsing (RP) and output structured formats (e.g., JSON) directly from images. However, these models are typically constrained to processing signs individually. Consequently, when multiple signs coexist in a complex scene, they struggle to capture the inter-sign spatial dependencies, thereby limiting their effectiveness in holistic Multi-Sign Rule Reasoning (MSR).

3) Large Vision-Language Models (LVLMs): Recent advancements in LVLMs have introduced new paradigms for parking sign understanding. These models, ranging from cloud-based foundations (e.g., GPT-series) to edge-optimized variants (e.g., Qwen-VL-series), demonstrate impressive capabilities in direct, black-box 'image-to-decision' inference. However, when decision interpretability is mandated, it becomes essential to enforce an explicit Rule Parsing (RP) task. In this specific task, LVLMs exhibit distinct limitations. As general-purpose models, they often struggle to strictly adhere to rigid domain-specific schemas without extensive and fragile prompt engineering. Consequently, their performance in generating standardized rules is often inferior to that of specialized End-to-End Parsers.

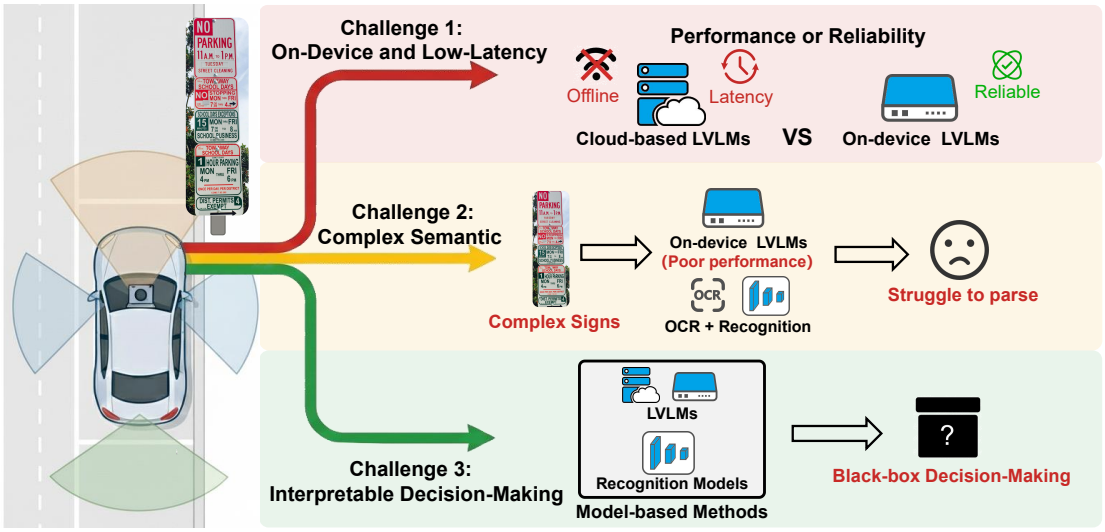

Fig. 2. Illustration of three critical challenges hindering reliable autonomous parking: **(1) On-Device and Low-Latency Deployment:** Safety-critical applications necessitate offline-capable, low-latency inference, rendering cloud-dependent LVLMs impractical. **(2) Complex Semantic Parsing:** Parking regulations are often defined by complex, visually diverse, and stacked signs, which existing traffic sign recognition methods struggle to comprehend holistically. **(3) Interpretable Decision-Making:** Safety-critical driving decisions require transparent reasoning, whereas end-to-end black-box models lack the necessary auditability and reliability.

As summarized in Table I, specialized research on parking sign recognition remains scarce. Most methods (Parts A & B) are confined to surface-level detection, neglecting multimodal semantics. In Rule Understanding (Part C), OCR pipelines suffer from error propagation, while E2E Parsers lack intrinsic decision logic and multi-sign reasoning. Although LVLMs enable direct inference, their capability to generate the interpretable structured JSON required for safety auditing often proves less reliable than specialized E2E parsers.

III. Problem Statements and Challenges

Parking is a uniquely challenging component of autonomous driving, requiring not only precise motion control in narrow areas but also reliable understanding of complex parking regulations. Unlike standardized traffic rules governing regular driving, parking constraints often vary dynamically with time, date, location, and vehicle type. These regulations are primarily conveyed through parking signs, many of which are stacked, visually diverse, or presented in mixed formats. Ensuring safe and lawful parking decisions therefore depends on accurately detecting these signs, understanding their semantics, and making interpretable and reliable decisions. To meet these requirements, three key challenges must be addressed:

Challenge 1: On-Device and Low-Latency Deployment

Autonomous vehicles frequently operate in network-limited or offline conditions, making cloud-dependent inference unsuitable for safety-critical tasks. Large foundation models (including LVLMs), while powerful, typically require remote API access or substantial on-board computational resources, leading to unpredictable latency and high deployment costs. Parking rule understanding, as a well-defined and localized task, demands lightweight models that can operate efficiently on edge hardware with stable, low, and fully controllable inference time. This necessity calls for a compact solution that remains effective without relying on external computation.

Challenge 2: Semantic Parsing of Complex Parking Signs

Parking signs differ fundamentally from conventional traffic signs, which rely on standardized pictorial symbols. Parking signs often embed combined text, symbols, temporal constraints, directional cues, and vehicle-type exceptions, and multiple signs may jointly define a single regulation. Existing traffic sign recognition models focus primarily on detecting or classifying simple visual symbols and lack the capability to integrate heterogeneous information into a coherent rule representation. Effective autonomous parking therefore requires a model capable of recognizing and interpreting these complex semantic elements.

Challenge 3: Interpretable Decision-Making

Parking decisions directly influence road safety and legal compliance; therefore, black-box model predictions are insufficient in regulated autonomous driving scenarios. Relying solely on an end-to-end model to produce the final decision is inherently unreliable, as such models obscure the reasoning process and offer no guarantees of consistency or auditability. Instead, decisions must be deterministic, transparent, and fully interpretable, enabling developers and regulators to examine the underlying logic and ensuring consistent behavior under identical conditions.

In summary, an effective parking sign recognition system must simultaneously address the tripartite challenges depicted in Fig. 2: parsing complex semantic rules, operating within strict latency constraints on edge devices, and

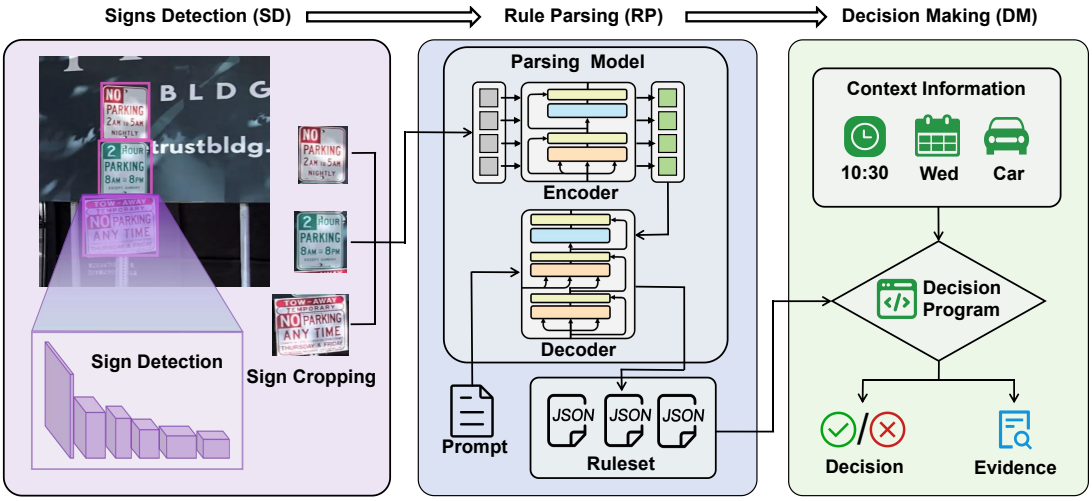

Fig. 3. Architecture of the PaRC service. We strategically decouple perception from logic. **Signs Detection (SD)** first isolates sign regions to ensure downstream efficiency. **Rule Parsing (RP)** then bridges the complex semantic gap by mapping visual inputs to a standardized JSON schema using an end-to-end model. Finally, **Decision Making (DM)** executes explicit logic on the parsed rules and context information (e.g., whether a car is allowed to park at 10:30 on Wednesday), ensuring the decision process is transparent and grounded.

providing interpretable decision-making logic. These requirements—*semantic complexity*, *deployment feasibility*, and *decision interpretability*—form the core design philosophy of the PaRC service, which is elaborated in Section IV.

IV. PaRC Service Design

To operationalize the design philosophy, we propose the PaRC service that strategically decouples visual perception from logical reasoning. As illustrated in Fig. 3, the PaRC service avoids the opacity and latency of monolithic end-to-end models by assigning distinct roles to three specialized modules. First, to ensure *deployment feasibility*, both the **Signs Detection** and **Rule Parsing** modules utilize lightweight architectures tailored for on-device inference: the former enables rapid localization and precise cropping of sign regions to maximize downstream parsing efficiency, while the latter addresses *semantic complexity* by interpreting multimodal visual cues into structured text. Finally, the **Decision Making** module guarantees *decision interpretability* by applying deterministic logic to these rules to yield consistent parking permissions. To enable integration with autonomous driving systems, we further design the PaRC Service Interface, which formalizes the input and output specifications of PaRC.

A. Signs Detection (SD)

The effective area occupied by parking signs in raw video frames is typically low, with frames containing substantial background information and irrelevant objects. Directly feeding raw frames into the semantic comprehension model would lead to significant computational redundancy and poor inference efficiency. Therefore, the SD module functions as a preliminary object detector, responsible for accurately locating each parking sign within the raw video frame, cropping and extracting it into independent image blocks, and batching these blocks for delivery to the Rule Parsing module.

The SD module comprises two components: Sign Detection and Sign Cropping. For Sign Detection, common methodologies include YOLO, R-CNN variants, and Transformer-based approaches. Given the strict requirements for real-time performance inherent to this task, and the core necessity being solely the acquisition of bounding boxes rather than deep semantic comprehension, the YOLO framework was ultimately selected. Upon determining the parking sign's bounding box, the Sign Cropping component extracts the sub-image based on the coordinates and passes it to the subsequent process.

B. Rule Parsing (RP)

The rules displayed on parking signs are commonly presented as a combination of free-form text and symbols, which are difficult for computer decision-making programs to process directly. The RP module is designed to bridge this semantic gap. It employs an end-to-end rule comprehension model to transform the cropped parking sign images into a structured semantic rule representation (e.g., JSON format), thereby achieving the critical conversion from visual perception to rule parameterization, which provides a standardized input for subsequent decision-making.

The RP module converts sign images into structured semantic rules. Conventional methods include End-to-End Parsers and the OCR plus Sequence-to-Sequence (OCR-Seq-Seq) approach. OCR-Seq-Seq requires two sequential models, increasing system complexity and resource use. Moreover, its inherent error accumulation and propagation limit overall system robustness. Consequently, we adopt an End-to-End Parsing Model for direct, precise mapping from visual features to structured rules.

Specifically, the RP module adopts an encoder-decoder architecture, eliminating the reliance on traditional OCR modules. The **encoder** serves as a visual feature extractor, transforming the raw input document image into a sequence

TABLE II
EXAMPLE OF PARKING RULES JSON PARSING

Field	Content
Text	2 hour parking 8 a.m. to 6 p.m. district no. 06 permits exempt ←
Category	`restricted`
Parameters	`duration_minutes`: 120
Conditions	`time`: 08:00 – 18:00
Exceptions	`permit_type`: "district no. 06"
Location	`applies_to_direction`: LEFT
Notes	NONE

of high-dimensional visual feature vectors. These features encapsulate the semantic information of both textual elements and graphical symbols. Subsequently, the **decoder**—taking both the encoded features and a prompt (utilized to distinguish downstream tasks and initiate the generation process) as inputs—effectively translates the visual content into a structured sequence (e.g., key-value pairs) that constitutes the JSON representation. By employing this end-to-end parsing approach, the module extracts a specific rule for each cropped sign, which are subsequently aggregated to form the final street-view ruleset.

To enable the selected End-to-End Model to directly transform visual features into a usable output, we propose a standardized structured representation for parking rules (in JSON format). This detailed structure is designed to cover the vast majority of real-world parking regulations. Specifically, the representation consists of seven fields: `Text`, `Category`, `Parameters`, `Conditions`, `Exceptions`, `Location`, and `Notes`. An example that conforms to the parking rules phrasing is shown in Table II.

- **Text**: The original text content displayed on the parking sign, kept without structural transformation.
- **Category**: The core type of the rule, such as `PERMITTED`, `NO_PARKING` or `NO_STOPPING`.
- **Parameters**: Quantitative or descriptive attributes supplementing the category, such as `duration_minutes`, `tow_away` or `max_vehicle_weight`.
- **Conditions**: Preconditions required for the rule to take effect, such as `time`, `day` or `vehicle_type`.
- **Exceptions**: Cases in which the rule does not apply, such as certain `vehicle_type` or `permit_holder`.
- **Location**: Spatial applicability details of the rule, such as directional indicators (`applies_to_direction`) or distance constraints (`applies_to_distance`).
- **Notes**: Supplementary notes that cannot be fully captured by the other fields.

C. Decision Making (DM)

The DM module serves as the decision-making core of the entire framework. Its primary task is to first integrate the structured rules derived from multiple parking sign images, logically inferring the complete set of constraints for the parking area. Subsequently, the DM module receives Contextual Information inputs, including but not limited to the current time, date, and vehicle type. Finally, by integrating the parsed parking rules and real-time contextual information, the DM module renders the final parking decision, including the decision result and supporting evidence.

The implementation of the Decision Making module can be either program-based or model-based. Program-based approaches are characterized by their speed, interpretability, and minimal overhead. Consequently, within our PaRC framework, we have chosen to implement Decision Making by using the program.

TABLE III
ABSTRACT INTERFACE FIELDS FOR PARC SERVICE

Category	Field Name	Description
Input (I, C)	`image_frame`	Captured visual scene
	`timestamp`	Query time context
	`vehicle_attr`	Vehicle-specific profiles
Output (R)	`is_permissible`	Binary parking decision
	`duration_limit`	Maximum stay duration
	`evidence_trace`	Decision-making logic
	`visual_anchor`	Spatial bounding box

D. PaRC Service Interface Definition

To facilitate the integration of PaRC into autonomous driving systems, we define a standardized abstract interface. This interface treats the PaRC service as a functional mapping:

$$R_{res} = PaRC(I_{img}, C_{ctx}) \tag{1}$$

where I_{img}, C_{ctx}, and R_{res} denote the input image, context, and response spaces, respectively. The structured fields are summarized in Table III.

The semantic definitions are detailed below:

Input Specification: `image_frame` provides the visual source I_{img}. `timestamp` and `vehicle_attr` constitute the query context C_{ctx}.

Output Specification: Beyond the decision results regarding binary decision and maximum duration, PaRC yields an `evidence_trace` (parsed rules) and `visual_anchor` (sign location) to ensure decision interpretability.

V. PARC IMPLEMENTATION AND EVALUATION

We conducted the model training and evaluation on a machine equipped with an Intel(R) Core(TM) i7-11700 @ 2.50GHz CPU, 64GB of RAM, and a GeForce RTX 3060 graphics card with 12GB of VRAM. This section describes the construction of the dataset and presents the model implementation and evaluation, which cover three aspects: Signs Detection, Rule Parsing, and Decision Making. Finally, we evaluate the performance of PaRC on low-end hardware.

A. Dataset Construction

Our dataset comprises three parts — the Detection Dataset, the Cropped Dataset, and the Test Dataset — which are used for training the Signs Detection model, training the Rule Parsing model, and testing models' performance, respectively.

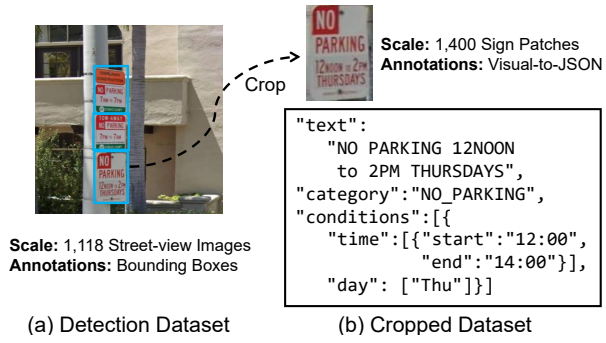

Fig. 4. Visualization of the Detection and Cropped subsets. (a) The Detection Dataset consists of street-view scenes annotated with bounding boxes for sign localization. (b) The Cropped Dataset comprises individual sign patches extracted from (a), paired with hierarchical JSON annotations.

Fig. 5. Examples of seed samples and their augmented views. Visualizations of the data augmentation strategy (e.g., perspective transformation, brightness adjustment) used to evaluate the model's robustness against real-world visual corruptions.

1) Detection Dataset: Given the scarcity of public image datasets focusing on U.S. parking signs, we consolidated three open-source datasets available on the Roboflow platform [21]–[23] to establish our Detection Dataset. This dataset comprises 1,118 raw street-view images intended for training the object detection model. The dataset was partitioned into a training set (70%), a validation set (20%), and a test set (10%). Of these images, 992 have a resolution of 416×416 pixels, while the remaining images have a resolution of 640×640 pixels.

2) Cropped Dataset: The Cropped Dataset was constructed based on the Detection Dataset by cropping each individual parking sign from the raw street-view images, as shown in Fig. 4. After filtering and discarding signs that were excessively blurred, the resulting dataset contains 1,400 independent sign images. The average resolution of these images is approximately 98×94 pixels. The Ground Truth for this dataset consists of annotated 'Visual-to-JSON' rule mappings for each parking sign, with labels created and verified by human annotators. This dataset was partitioned into a training set (80%), a validation set (10%), and a test set (10%).

3) Test Dataset: We constructed a specialized Test Dataset to rigorously evaluate the robustness and generalization capability of the proposed model. Initially, 40 high-quality images were selected from the Detection Dataset's test split to serve as seed samples. To simulate diverse real-world acquisition conditions, we applied a comprehensive data augmentation strategy to these seed images (see Fig. 5). Techniques included brightness/contrast adjustment, perspective transformation, and geometric operations such as translation and rotation.

This generation process yielded four augmented variants for each seed image, resulting in a total of 200 evaluation samples (40 base $\times 5$ variants). This dataset is exclusively reserved for the performance assessment of both Rule Parsing and Decision Making tasks, ensuring that the model is tested against varying degrees of visual corruption and geometric perturbation.

B. Signs Detection

For Signs Detection (SD), the rapid advancements in the field of object detection in recent years, particularly the mature research ecosystem targeting Traffic Signs Detection, provide a robust foundation. Parking Signs Detection shares a high degree of similarity and reusability in its technical approach with standard traffic sign detection, with the main distinction being the training dataset. Consequently, the focus of this study is not on innovating SD methodologies, but rather on directly integrating and leveraging the most advanced and highly efficient existing detection technologies.

TABLE IV
VALIDATION PERFORMANCE OF YOLOV11N FOR PARKING SIGNS DETECTION: COMPARISON BETWEEN THE BEST AND LAST CHECKPOINTS.

Checkpoint	Epoch	mAP@50	mAP@50–95
Best (Optimal)	60	0.972	0.686
Last (Final)	100	0.951	0.631

We adopted the widely recognized and high-performing YOLO framework. Specifically, we employed the YOLOv11 [24] model (the latest public release at the time of our experiments). Using the detection dataset described earlier, we trained the YOLOv11n model for 100 epochs, and its performance metrics are reported in Table IV. Notably, YOLOv11n is the smallest detection model in the YOLOv11 family; however, its performance is already sufficient for our scenario. Considering both inference speed and model compactness, YOLOv11n is selected as PaRC's detection model, and the best weights at epoch 60 are used.

C. Rule Parsing

The Rule Parsing (RP) module aims to perform a rule-based mapping of the natural language information and symbolic constraints present on parking signs, thereby making them understandable and processable by computer decision programs. In this section, we designed experiments to evaluate the Rule Parsing capability of PaRC's model and other methods.

1) PaRC's Method: We selected the widely-adopted Donut [20] model as PaRC's base model. Donut is an OCR-free, end-to-end model specialized in visual document understanding and information extraction tasks. In this work, we adopt a *low-resource* fine-tuning paradigm, where the pretrained

Parking Sign Image	PaddleOCR 3.0	Azure Vision	End-to-End Parsers (PaRC's Fine-tuned Donut)
 Sign 1	12NO0N=3PM SEET SWLEPIN THURSDAY	R 12NOON = 3PM THURSDAY	"category":"NO_PARKING", "conditions":{ "day":"Thu", "time":{"end":"15:00","start":"12:00"}, "traffic_event":"street sweeping"}
 Sign 2	2 HOUR PARKING 8AM 4PM 6PM 8PM	HOUR 2 PARKING 8 AM 4rw 6PM* 8PM	"category": "RESTRICTED", "conditions":[{"day":["Mon","Tue","Wed","Thu","Fri"], "time":[{"end":"16:00","start":"08:00"}, {"end":"20:00","start":"18:00"}]}, {"day":"Sat", "time":{"end":"20:00","start":"08:00"}], "parameters":{"duration_minutes":"120"}

Fig. 6. Qualitative comparison of OCR baselines (PaddleOCR 3.0, Azure Vision) versus End-to-End Parsers. The examples illustrate the limitations of OCR methods in handling blurry text, symbolic icons, and non-standard layouts (e.g., vertical text).

TABLE V
DETAILED CONFIGURATION OF THE MODEL ARCHITECTURE AND FINE-TUNING HYPERPARAMETERS.

Category	Hyperparameter Setting
Visual Encoder (Donut-Swin)	
Input Resolution	320×320
Patch Size	4×4
Window Size	10
Embedding Dimension	128
Stage Depths	[2, 2, 14, 2]
Attention Heads	[4, 8, 16, 32]
Text Decoder (mBART)	
Decoder Layers	4
Model Dimension (d_{model})	1024
FFN Dimension	4096
Attention Heads	16
Vocabulary Size	57,582
Task-specific Tokens	60
Training Strategy	
Optimizer	AdamW
Learning Rate	2×10^{-5}
Effective Batch Size	8
Gradient Accumulation	2
Training Epochs	15
Warmup Steps	30
Max Target Length	370
Precision	AMP (FP16)

`Donut-Base` model is updated end-to-end on our **Cropped Dataset** using only a limited number of labeled examples.

More specifically, the visual encoder follows the Donut-Swin architecture, and the text decoder is an mBART-style Transformer adapted with 60 task-specific special tokens (e.g., category, duration_minutes and vehicle_type). Key architectural and training hyperparameters are summarized in Table V.

2) Baseline Models: As mentioned in Section II, existing approaches to rule understanding include **OCR+Sequence-to-Sequence**, **End-to-End Parsers**, and **LVLMs**.

The OCR + Seq2Seq category is divided into traditional methods and deep learning-based OCR models. For traditional methods, we tested Tesseract [16] and found it almost unusable for low-resolution parking signs, failing to yield meaningful recognition results. Among deep learning-based models, we evaluated PaddleOCR 3.0 [17] and Microsoft Azure Vision (Image Analysis 4.0) [18], comparing their performance against End-to-End Parsers (specifically, PaRC's fine-tuned Donut model). The results (Fig. 6) demonstrate that OCR methods lack accuracy in recognizing blurry text (e.g., "street sweeping" on Sign 1) and fail to identify symbolic content (e.g., the No Parking symbol on Sign 1). Furthermore, they struggle with misaligned layout structures, exemplified by the inability to recognize the vertical "8am to 8pm" text and process the position of numeral "2" on Sign 2.

Evidently, OCR methods are better suited for formatted documents and are inadequate for complex parking sign scenarios involving mixed symbols and misaligned layouts. Since OCR errors and information loss would render the subsequent Seq2Seq model ineffective, we discontinue testing the OCR + Seq2Seq approach in our subsequent evaluation, focusing only on End-to-End Parsers (used in PaRC) and LVLMs.

Regarding LVLMs, we evaluated both existing cloud-based LVLMs and edge-deployable lightweight models. We assessed the performance of cloud-based LVLMs via API calls, while utilizing tools such as Ollama to deploy open-source lightweight models locally for evaluation. The specific performance of these models will be discussed later in this section.

3) Evaluation Metrics: To assess rule parsing quality, we evaluate the model on the structured fields defined in Section IV. Since the `Text` field contains only raw OCR output and does not reflect semantic parsing ability, it is excluded from scoring. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$ denote the set of applicable structured fields for a given parking sign, where $k \leq 6$ depending on which fields are present. To eliminate the effect of weight assignment on the score, each field is given equal weight, computed as

$$S = \frac{100}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \mathbf{1}[model(f) = GT(f)], \tag{2}$$

where $\mathbf{1}[\cdot]$ is an indicator function that returns 1 if the model's prediction for field f matches the ground truth and 0 otherwise. Thus, if only $|\mathcal{F}| = 4$ fields are present for a particular sign, each field contributes 25 points to the overall score.

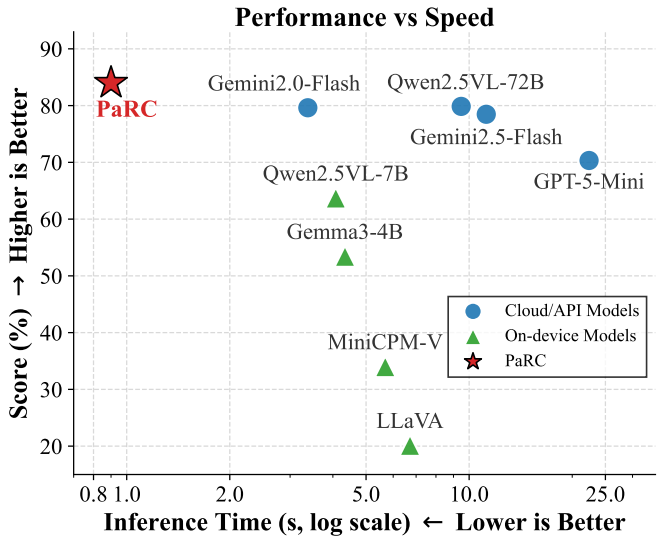

Fig. 7. Comparison of Rule Parsing Performance Across Different Models. All models are evaluated using the latest available API interfaces and open-source releases as of November 10, 2025. Time denotes end-to-end latency from issuing an inference request to receiving the full model output; for online LVLMs, this includes network overhead from API communication.

4) Prompt Design: For the parsing task, the model is instructed to act as an expert annotator that converts the information on each parking sign into a structured JSON record. The prompt emphasizes exhaustive extraction of visible rule content and requires strictly well-formed JSON output without additional natural-language commentary. All LVLMs baselines share the same parsing prompt for fairness.

5) Performance Comparison: We evaluated the Rule Parsing performance of the selected models, and the average score per image and the inference time are shown in Fig. 7.

Experimental results show that PaRC achieves the best performance in both Rule Parsing Score and Inference Time. In contrast, several online LVLMs perform similarly overall, while on-device lightweight models show weaker ability. Notably, Qwen2.5VL-7B reaches a relatively usable level.

A closer analysis shows that no model scored above 90. This is mainly because images in the Cropped Dataset have a low average resolution of only about 98.43×93.99 pixels. At this resolution, text on some signs is severely pixelated, making the information ambiguous and difficult even for human observers to decipher accurately, as shown in Fig. 5 and Fig. 6. Consequently, an absolutely correct answer may not exist during evaluation. This phenomenon effectively tests model robustness under extreme visual degradation.

D. Decision Making

The cloud-based LVLMs and lightweight open-source models were not fine-tuned against our dataset and defined standardized JSON rules, and we were limited to guiding their output through Prompt Engineering. This leads to poor Rule Parsing performance for these models. If Rule Parsing is used as the sole evaluation criterion, this presents a degree of bias or unfairness towards them, as these models inherently possess the ability to directly infer and provide the final parking decision without relying on the JSON format generated by Rule Parsing. Based on this consideration, we further designed experiments to evaluate the models' ability to directly make the final parking decision (Yes/No).

Algorithm 1: Basic Binary Parking Decision Logic

Input : Rule Set $\mathcal{R}$, Query Context $\mathcal{C}$
Output: Decision $d \in \{\text{YES}, \text{NO}\}$
`// Allow if no rules`
if $\mathcal{R} = \emptyset$ **then**
  return YES;
`// Filter and sort prohibitive rules`
$\mathcal{R}_p \leftarrow \{r \in \mathcal{R} \mid r.cat \in \{\text{NOPARK}, \text{NOSTOP}\}\}$;
Sort $\mathcal{R}_p$ by priority (descending);
`// Check against prohibitions`
foreach $r \in \mathcal{R}_p$ **do**
  if `Matches`$(r.cond, \mathcal{C})$ **then**
    `// Check for exceptions`
    if not `CheckException`$(r.except, \mathcal{C})$ **then**
      return NO;
return YES;

1) Evaluation Metrics: We designed 10 Scenario Test Cases for each sign image. To simplify testing, each case is defined by a unique set of contextual information: {Time; Date; Vehicle Type}. These 10 scenarios aim to cover a wide range of possible conditions, including different times of day, rules without time constraints, boundary-time conditions, various vehicle types, and whether exemptions apply. The models must provide a binary decision (Yes/No) on whether parking is permitted in each scenario. We evaluated the model performance using the standard binary classification metrics, F1-Score and Accuracy.

2) Methodology and Baselines: Within PaRC's overall framework, since the Rule Parsing module successfully converts the parking rules into a program-understandable JSON format, we utilize a deterministic rule program for final decision-making, rather than relying on a language model. This program-based decision mechanism offers significant advantages: it ensures extremely high running efficiency and complete interpretability. For the other baseline models used in comparative analysis, we followed the same testing methodology as the Rule Parsing phase.

The core logic for establishing instantaneous parking feasibility is formalized in Algorithm 1. This process treats the decision as a binary (Yes/No) classification problem, mapping a rule set $\mathcal{R}$ and a query context $\mathcal{C}$ (comprising time t, vehicle profile v, and date d) to a decision space $d \in \{\text{YES}, \text{NO}\}$.

The algorithm adopts a default-permissive strategy, implicitly allowing parking in the absence of constraints ($\mathcal{R} = \emptyset$). Conversely, when rules are present, it proceeds by isolating prohibitive constraints ($\mathcal{R}_p$), prioritizing safety-critical restric-

TABLE VI
COMPARISON OF DECISION MAKING PERFORMANCE ACROSS DIFFERENT MODELS.

Model	PaRC	Gemini2.0-Flash	Gemini2.5-Flash	GPT-5-Mini	Qwen2.5VL-72B	Gemma3-4B	Qwen2.5VL-7B	MiniCPM-V	LLaVA
F1 (%)	**96.15**	74.24	**96.43**	**95.73**	76.89	57.07	49.55	28.85	53.68
Acc. (%)	**94.90**	66.55	**95.35**	**94.40**	68.80	50.50	49.70	23.80	48.05
Time (s)	**0.90**	2.97	13.40	17.44	4.41	3.55	2.53	2.22	3.17

[1] The evaluated models are consistent with those in Fig. 7, all representing the latest versions available as of November 10, 2025. The left side of the table presents the performance of online LVLMs, ranging from Gemini 2.0-Flash to Qwen2.5VL-72B; the right side displays the evaluation results for open-source locally deployed Vision-Language Models, spanning from Gemma3-4B to LLaVA.

Algorithm 2: Parking Decision & Duration Estimation

Input : Rule Set $\mathcal{R}$, Query Context $\mathcal{C} = \{t_{start}, v_{type}\}$, Horizon T_{max}
Output: Decision $d \in \{\text{YES}, \text{NO}\}$, Duration δ^*

```
// Define rule subsets by category
```
$\mathcal{R}_p \leftarrow \{r \in \mathcal{R} \mid r.cat \in \{\text{NOPARK}, \text{NOSTOP}\}\}$;
$\mathcal{R}_r \leftarrow \{r \in \mathcal{R} \mid r.cat \in \{\text{RESTRICTED}\}\}$;
```
// Check Immediate Feasibility
```
foreach $r \in \mathcal{R}_p$ **do**
  if `IsActive(`r, t_{start}`)` **then**
    return $\{\text{NO}, 0\}$; `// Immediate prohibition`
```
// Calculate Restricted Duration
```
$\delta_{limit} \leftarrow T_{max}$;
foreach $r \in \mathcal{R}_r$ ***where*** `IsActive(`r, t_{start}`)` **do**
  $\tau_{remain} \leftarrow r.end - t_{start}$; `// Time window remaining`
  $\delta_{limit} \leftarrow \min(\delta_{limit},\ r.duration,\ \tau_{remain})$;
```
// Look-ahead for Future Conflicts
```
$\delta_{conflict} \leftarrow T_{max}$;
Find minimum $\Delta t \in (0, \delta_{limit}]$ such that:
  $\exists r \in \mathcal{R}_p$, `IsActive(`$r, t_{start} + \Delta t$`)` is TRUE
if *such* Δt *exists* **then**
  $\delta_{conflict} \leftarrow \Delta t$;
```
// Constraint Aggregation
```
$\delta^* \leftarrow \min(\delta_{limit}, \delta_{conflict})$;
if $\delta^* \geq T_{max}$ **then**
  return $\{\text{YES}, \text{UNLIMITED}\}$;
return $\{\text{YES}, \delta^*\}$;

tions (e.g., *No Stopping* $\succ$ *No Parking*) to ensure regulatory compliance. The function $\mathcal{F}_{\text{Matches}}$ evaluates whether a specific rule applies to the current context $\mathcal{C}$. To guarantee robustness, the algorithm incorporates an exception handling mechanism via $\mathcal{F}_{\text{CheckException}}$, which filters out prohibitions negated by specific exemptions (e.g., resident permits or loading zones). A decision of NO is returned if and only if a prohibitive rule matches the context without a valid exception; otherwise, the state defaults to YES.

While Algorithm 1 establishes a basic binary (Yes/No) decision model, practical applications such as autonomous vehicle and human driving require not only immediate feasibility checks but also predictive estimates of the maximum permissible duration to avoid future infractions. Algorithm 2 meets this need by extending the decision boundary into the temporal domain to estimate the maximum permissible parking duration, δ^*, within a finite time horizon T_{max}.

The procedure first categorizes the rule set into prohibitive rules ($\mathcal{R}_p$) and restricted rules ($\mathcal{R}_r$) (e.g., time-limited parking). Initial feasibility is validated by checking $\mathcal{R}_p$ against the start time t_{start}; any active prohibition triggers an immediate rejection. Subsequently, the algorithm computes two temporal constraints:

- Restriction Limit (δ_{limit}): Computed by aggregating active restrictions in $\mathcal{R}r$. The duration is limited by the minimum of the global horizon $Tmax$, the rule-specific allowance ($r.duration$), and the remaining time in the current regulatory window (τ_{remain}).
- Conflict Look-ahead ($\delta_{conflict}$): A predictive step that scans the interval $(t_{start}, t_{start} + \delta_{limit}]$ for impending prohibitions. It identifies the earliest future time point Δt at which a latent prohibition from $\mathcal{R}_p$ becomes active, thereby truncating the allowed duration.

The final estimated duration δ^* is derived from the aggregation of these constraints: $\delta^* = \min(\delta_{limit}, \delta_{conflict})$. This ensures the vehicle not only parks legally at t_{start} but also vacates the spot before any conflicting regulation takes effect.

3) Prompt Design: For the decision task, the model is positioned as a U.S. parking rule expert. Given a street-view image and ten query scenarios Time; Date; Vehicle Type, the prompt asks it to determine whether parking is allowed and answer strictly with a fixed-format JSON list, where each output contains only an `"answer"` field with value `"YES"` or `"NO"`. The same decision prompt is used for all LVLMs.

4) Performance Comparison: The performance results for PaRC and the baseline models are shown in the Table VI. Since we use a program instead of a model for Decision Making and its processing time per image is negligible, PaRC's decision-making time is almost equal to its Rule Parsing time. The results show that PaRC, Gemini 2.5-Flash, and GPT-5-Mini achieve the best overall performance. However, PaRC has a clear advantage in inference speed, requiring only 0.90 s per image, whereas the other two LVLMs require more than ten seconds of waiting time. For drivers, such latency feels frustrating. Overall, lightweight models offer faster inference, while larger models generally improve performance but substantially increase processing time.

TABLE VII
COMPARISON ACROSS DIFFERENT HARDWARE PLATFORMS.

Name	Personal Desktop	Personal Laptop	OrangePi 4 Pro
CPU	i7-11700	i5-1335U	Allwinner A733
GPU	RTX 3060	-	-
Memory	64GB	16GB	4GB
Score	84.00	80.21	78.12
F1 (%)	96.15	95.41	94.36
Acc. (%)	94.90	93.85	92.55
Time (s)	0.90	1.86	5.46

[1] The table compares the hardware platform used in the previous experiments, namely Personal Desktop, with low-end devices. Since the GPUs on these devices are limited, we run PaRC on the CPU and therefore do not report their GPU specifications in the table. For deployment on low-end hardware, we apply post-training dynamic quantization to the model, which reduces inference time at the cost of a slight drop in output quality.

E. Low-End Deployment

To verify the performance of PaRC on low-end hardware, we deploy it on a personal laptop without a discrete GPU and on an OrangePi 4 Pro 4G development board, as shown in Table VII. The Score corresponds to the rule parsing task, while F1 and Acc. correspond to the decision making task, which are aligned with Fig. 7 and Table VI, respectively.

Compared with the RTX 3060 platform used in previous experiments, PaRC shows some performance decline on low-end devices, but it still achieves performance comparable to that of online LVLMs and runs faster than the other models on the RTX 3060 platform in Fig. 7 and Table VI. In particular, on the OrangePi 4 Pro, which costs only US$35, PaRC still achieves acceptable performance and speed, which demonstrates its adaptability to low-end hardware.

VI. CONCLUSION AND FUTURE WORK

Intelligent parking decision-making requires methods that are offline, multimodal, and interpretable. This paper proposes PaRC for parking sign understanding and decision-making. By combining SD, RP, and DM modules, PaRC enables efficient, interpretable parking decisions. The results show that PaRC achieves performance comparable to online LVLMs while running faster than on-device lightweight models. It also maintains acceptable performance and speed on low-end devices, demonstrating practical utility.

During model evaluation, we found that the extremely low legibility of some dataset images, which even human observers struggle to discern accurately, limited the final accuracy scores. Consequently, future work will likely focus on collecting and creating high-quality parking sign datasets to further improve performance in this domain.

REFERENCES

[1] INRIX, "Searching for parking costs Americans $73 billion a year." [Online]. Available: https://inrix.com/press-releases/parking-pain-us/, 2017. Accessed: Jan. 3, 2026.

[2] Stopsigns and More, "Stack of no parking signs cause laughter, ridicule in LA." [Online]. Available: https://www.stopsignsandmore.com/p-2824-stack-of-no-parking-signs-cause-laughter-ridicule-in-la.aspx, 2014. Accessed: Jan. 3, 2026.

[3] C. Choi, "Signs of confusion: Moves to end perplexing parking signs." [Online]. Available: https://www.itv.com/news/2024-12-04/signs-of-confusion-moves-to-end-perplexing-parking-signs, 2024. ITV News, Accessed: Jan. 3, 2026.

[4] B. M., "Los Angeles moves to simplify parking signs." [Online]. Available: https://californiacitynews.org/2014/10/los-angeles-moves-simplify-parking-signs.html, 2014. Accessed: Jan. 3, 2026.

[5] A. Nazarian, "A new look for the city's parking signs." [Online]. Available: https://cd2.lacity.gov/news/new-look-citys-parking-signs, 2015. Accessed: Jan. 3, 2026.

[6] X. Meng, X. Zhang, K. Yan, and H. Zhang, "Real-time detection and recognition of live panoramic traffic signs based on deep learning," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 584–588, IEEE, 2018.

[7] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2584–2601, 2017.

[8] D. Tabernik and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 4, pp. 1427–1440, 2019.

[9] P. Haji Faraji, *Efficient street parking sign detection and recognition using artificial intelligence*. PhD thesis, University of British Columbia, 2023.

[10] Y. Jin, *Real-time parking sign detection for smart street parking*. University of Washington, 2022.

[11] D. Zhong, *Dictionary-Guided Text Recognition for Smart Street Parking*. University of Washington, 2023.

[12] H. Irshad, Q. Mirsharif, and J. Prendki, "Crowd sourcing based active learning approach for parking sign recognition," *arXiv preprint arXiv:1812.01081*, 2018.

[13] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Deep convolutional neural network for enhancing traffic sign recognition developed on yolo v4," *Multimedia Tools and Applications*, vol. 81, no. 26, pp. 37821–37845, 2022.

[14] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient cnns in the wild," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 975–984, 2018.

[15] Z. Wang, J. Wang, Y. Li, and S. Wang, "Traffic sign recognition with lightweight two-stage model in complex scenes," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 2, pp. 1121–1131, 2020.

[16] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2, pp. 629–633, IEEE, 2007.

[17] C. Cui, T. Sun, M. Lin, T. Gao, Y. Zhang, J. Liu, X. Wang, Z. Zhang, C. Zhou, H. Liu, Y. Zhang, W. Lv, K. Huang, Y. Zhang, J. Zhang, J. Zhang, Y. Liu, D. Yu, and Y. Ma, "Paddleocr 3.0 technical report," 2025.

[18] Microsoft, "What is image analysis? — foundry tools | microsoft learn." [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/overview-image-analysis?tabs=4-0, 2025. Accessed: Jan. 3, 2026.

[19] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[20] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, and S. Park, "Ocr-free document understanding transformer," in *European Conference on Computer Vision (ECCV)*, 2022.

[21] ucbw251, "w251-final-project-parking dataset." [Online]. Available: https://universe.roboflow.com/ucbw251/w251_final_project_parking, 2022. Roboflow Universe. Accessed: Jan. 3, 2026.

[22] PS, "Psigns dataset." [Online]. Available: https://universe.roboflow.com/ps-0dxq5/psigns-jgebu, 2023. Roboflow Universe. Accessed: Jan. 3, 2026.

[23] A. Ismatillaev, "Parking signs dataset." [Online]. Available: https://universe.roboflow.com/avazbek-ismatillaev-zjamd/parking-signs-pfbv3, 2023. Roboflow Universe. Accessed: Jan. 3, 2026.

[24] G. Jocher and J. Qiu, "Ultralytics yolo11 (version 11.0.0)." [Online]. Available: https://github.com/ultralytics/ultralytics, 2024. License: AGPL-3.0. Accessed: Jan. 3, 2026.