vehicles

MDPI

Article

Lightweight Parking-Line Segmentation for Automated Parking in Underground Garages via Shallow Detail Injection and Deep Semantic Slimming

Zhanzheng Guo [1,*], Xiaolong Shan [1], Dongmei Sun [1], Guangwei Cheng [2] and Yuxuan Wang [1]

[1] School of Vehicle and Traffic Engineering, Henan University of Science and Technology, Luoyang 471003, China; 240320030360@stu.haust.edu.cn (X.S.); sdmyanzi@163.com (D.S.); 240320030367@stu.haust.edu.cn (Y.W.)
[2] School of Intelligent Vehicles, Luoyang Institute of Science and Technology, Luoyang 471023, China; chenggwtom@163.com
* Correspondence: guozhanzheng@126.com; Tel.: +86-18736251063

Abstract

Parking-line segmentation for automated parking in underground garages is challenged by sparse targets, perspective-compressed thin structures, and visually similar disturbances such as cracks, joints, uneven illumination, and reflections. We propose a lightweight YOLO26n-sem framework with Semantic-Preserving Shallow Residual Detail Injection (SRDI) and Interface-Preserving Deep Semantic Slimming (IPDS). SRDI injects scale-aligned P2 details into the P3 pathway through a bounded learnable residual coefficient, while IPDS reduces hidden widths in deep P5 modules without changing the external P5 interface or top-down propagation topology. On a four-class bird's-eye-view dataset acquired by a vehicle-mounted surround-view system, the model achieves 85.966% mIoU, 83.081% Parking IoU, and 87.700% Thin Recall, improving YOLO26n-sem by 0.500, 0.973, and 1.556 percentage points, respectively. Parameters decrease by 24.9%, from 1.631 M to 1.225 M, and latency decreases from 8.8203 to 7.0321 ms (142.21 FPS). Zero-shot evaluation on an independent public underground-garage dataset achieves 77.837% mIoU and 72.625% Parking IoU, exceeding the baseline by 1.416 and 2.917 percentage points, respectively, without target-domain adaptation. Overall, the proposed model improves segmentation accuracy while reducing parameter count and inference latency, and its advantage over the baseline remains observable under the tested cross-garage domain shift.

Keywords: parking-line segmentation; semantic segmentation; automated parking; lightweight neural network; real-time perception

Check for updates

Received: 27 September 2026
Revised: 17 October 2026
Accepted: 22 October 2026
Published: 25 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Intelligent driving and automated parking systems require reliable visual information in confined, low-speed, and structurally complex parking environments. Perception of parking spaces and their markings directly affects space localization, drivable-area estimation, and parking-path planning. Recent deep learning methods for visual parking-space perception include object detection, keypoint regression, semantic segmentation, and graph neural networks [1–3]. Their reliability, however, remains sensitive to worn markings, occlusion, geometric variation, and complex illumination. Weak lighting, artificial light sources, local reflections, and alternating shadows make parking line recognition particularly difficult in underground garages.

Prior research has addressed parking-space perception through several complementary formulations. Suhr et al. [4] combined global and local information in an end-to-end one-stage parking-space detector. Bui et al. [5,6] improved detection through region-specific multiscale feature extraction and progressive assembly of parking-space components. Min et al. [7] represented parking-space marking points as a graph and learned their connectivity with an attentional graph neural network. Beyond keypoint and geometric detection, Lai et al. [8] used panoramic-image semantic segmentation for real-time parking-space detection. Other studies addressed uneven illumination [9], multi-cue recovery in complex parking environments [10], and attentional semantic segmentation with instance matching [11]. These approaches mainly recover complete parking-space corners, entrances, or geometric relations. Ning et al. [12] instead combined Coordinate Attention (CoordAtt) and atrous spatial pyramid pooling (ASPP) for worn parking-line segmentation, emphasizing directional representation and multiscale context. The present study addresses a different pair of architectural questions: how to supplement P3 with P2 detail without replacing the P3 semantic pathway, and how to reduce P5 internal capacity without changing its external cross-scale interface.

Unlike geometric parking-space detection, parking line semantic segmentation requires pixel-wise classification of elongated, sparse targets with low pixel occupancy. This challenge resembles road-marking and lane-line perception. Hou et al. [13] showed that lightweight networks may inadequately represent sparse road markings under weak illumination and occlusion. Wang et al. [14] used temporal feature fusion to compensate for local information loss caused by shadows, glare, and vehicle occlusion. CLRNet [15] demonstrated the value of high-level semantics for target discrimination and low-level spatial detail for accurate lane-line localization. RVLD [16] improved temporal stability through recursive propagation across adjacent frames. In underground garages, perspective further narrows distant parking lines. Cracks, construction joints, stains, and reflective edges can also generate similar high-frequency responses in shallow features. A suitable model must therefore preserve local line details without indiscriminately introducing shallow texture noise.

Real-time semantic segmentation research has long examined the balance between high-resolution spatial information and low-resolution category semantics. Fast-SCNN [17] combines learned downsampling with lightweight feature fusion. BiSeNet V2 [18] uses separate Detail and Semantic Branches to represent spatial detail and semantic information. STDC-Seg [19] further reduces the computational cost of an independent spatial branch. DDRNet [20] continuously exchanges features between two resolution streams. PIDNet [21] separately models detail, context, and boundary information to reduce interference between high-frequency detail and low-frequency semantics. FBSNet [22] similarly uses a lightweight bilateral architecture to combine local detail with contextual information. GCNet-S [23] combines a dual-branch architecture with reparameterizable blocks, merging their training-time multi-convolution and multi-path structures into a single 3×3 convolution per block for efficient inference. These studies show that multiscale or multiresolution features benefit fine-grained segmentation, although maintaining a high-resolution branch increases computation and memory usage.

Transformers and lightweight attention mechanisms have expanded the design space for real-time semantic segmentation. PP-LiteSeg [24] reduces decoding cost with a lightweight decoder and unified attention. RTFormer [25] combines dual-resolution processing with GPU-friendly Transformer attention. TopFormer [26] uses a token pyramid and semantic injection for mobile multiscale modeling. SeaFormer [27], AFFormer [28], and EfficientViT [29] reduce dense high-resolution prediction costs through compressed axial attention, linear Transformers, and hardware-efficient multiscale attention, respectively.

SegFormer [30] and SegNeXt [31] enhance multiscale representation using hierarchical Transformers and convolutional attention. These methods primarily target general scenes and do not explicitly address the coexistence of attenuated thin-line features and distracting linear background textures in underground garages.

Several techniques specifically address the preservation of elongated structures. Boundary IoU [32] complements region IoU by emphasizing boundary quality, while clDice [33] promotes centerline connectivity through a topological constraint. DySample [34] improves high-resolution feature recovery through learnable sampling, and Coordinate Attention [35] strengthens position-sensitive representation. Adding detail-processing operations, however, does not necessarily improve region accuracy or deployment efficiency. Studies of lightweight networks also show that parameter count and floating-point operations (FLOPs) do not fully characterize runtime behavior because memory access, operator type, and hardware parallelism materially affect latency [36–40]. FasterNet and MobileOne illustrate that lower FLOPs do not necessarily imply lower latency, while RepViT, EfficientFormer, and EfficientFormerV2 emphasize hardware-aware design. Parking-line segmentation in underground garages therefore requires separate control of shallow-detail recovery, deep-semantic capacity, model size, and measured inference latency.

Accordingly, Ultralytics YOLO26n-sem [41] is adopted as the baseline to address two complementary challenges: the attenuation of sparse parking-line detail during downsampling and the parameter cost of deep semantic processing. The main contributions are as follows:

(1) We propose Semantic-Preserving Shallow Residual Detail Injection (SRDI), which introduces scale-aligned P2 features into the Neck-P3 representation through a bounded learnable residual while preserving the original semantic pathway.
(2) We introduce Interface-Preserving Deep Semantic Slimming (IPDS), which reduces the internal capacity of deep P5 modules while retaining their external feature interfaces and the original P5-to-P4-to-P3 propagation topology.
(3) We conduct comprehensive comparative, ablation, line-width-stratified, difficult-sample, and cross-garage evaluations. The proposed model improves mIoU and Parking IoU by 0.500 and 0.973 percentage points over YOLO26n-sem while reducing the parameter count by 24.9% and inference latency by 20.3%. Zero-shot evaluation on an independent underground-garage dataset further yields gains of 1.416 and 2.917 percentage points in mIoU and Parking IoU, respectively, demonstrating a favorable accuracy–efficiency trade-off and partially retained performance under the tested cross-garage domain shift.

2. Materials and Methods

2.1. Baseline YOLO26n-Sem Network

We use YOLO26n-sem as the baseline network for parking line semantic segmentation in underground garages. The network consists of a Backbone, a top-down Neck, and a Semantic Segment Head. The Backbone extracts multiscale features, while the Neck progressively propagates deep semantic information through a top-down pathway. The Semantic Segment Head uses Neck-P3/8 as the main prediction feature and Neck-P4/16 for auxiliary supervision during training.

For the four-class task, the number of semantic categories is K = 4, corresponding to background, parking line, guide mark, and auxiliary line. The input size is fixed at H × W = 1024 × 1024. YOLO26n uses the official n-scale configuration, with depth and

width multipliers of 0.50 and 0.25, respectively. After scaling, the Backbone generates four feature levels, P2/4, P3/8, P4/16, and P5/32. These features are denoted as follows:

$$F_2^B \in \mathrm{R}^{64\times\frac{H}{4}\times\frac{W}{4}} \tag{1}$$

$$F_3^B \in \mathrm{R}^{128\times\frac{H}{8}\times\frac{W}{8}} \tag{2}$$

$$F_4^B \in \mathrm{R}^{128\times\frac{H}{16}\times\frac{W}{16}} \tag{3}$$

$$F_5^B \in \mathrm{R}^{256\times\frac{H}{32}\times\frac{W}{32}} \tag{4}$$

The superscript B denotes Backbone features. Spatial resolution decreases with network depth, whereas semantic abstraction increases. P2 retains relatively rich local edges and geometric details, while P5 provides stronger high-level contextual representations.

The Neck aggregates features through a top-down pathway. It first upsamples P5 to the 1/16 scale and concatenates it with Backbone-P4 along the channel dimension. C3k2 fusion then produces the Neck-P4 feature:

$$F_4^N = C_4\left(\mathrm{Concat}\left[U\left(F_5^B\right), F_4^B\right]\right) \tag{5}$$

Here, the superscript N denotes a Neck feature, $U(.)$ denotes 2× nearest-neighbor upsampling, and $C_4(.)$ denotes the C3k2 fusion operation that produces Neck-P4.

Because P5 and P4 contain 256 and 128 channels, respectively, their concatenation has 384 input channels. The fused output is F_4^N

$$F_4^N \in \mathrm{R}^{128\times\frac{H}{16}\times\frac{W}{16}} \tag{6}$$

Next, F_4^N is upsampled to the 1/8 scale and fused with Backbone-P3 as

$$F_3^N = C_3\left(\mathrm{Concat}\left[U\left(F_4^N\right), F_3^B\right]\right) \tag{7}$$

This operation yields

$$F_3^N \in \mathrm{R}^{64\times\frac{H}{8}\times\frac{W}{8}} \tag{8}$$

These operations form the top-down P5-to-P4-to-P3 semantic propagation pathway. Although P5 is not directly input to the final Semantic Segment Head, its high-level semantics propagate through P4 to P3. P5 therefore contributes to the semantic discrimination of the main prediction feature.

In the Semantic Segment Head, F_3^N is the main prediction feature and is mapped by convolution to class logits:

$$Z_{\mathrm{main}} = H_3\left(F_3^N\right) \in \mathrm{R}^{K\times H/8\times W/8} \tag{9}$$

For the four-class task,

$$Z_{\mathrm{main}} \in \mathrm{R}^{4\times 128\times 128} \tag{10}$$

During training, F_4^N is used to construct an auxiliary branch Z_{aux} for stronger intermediate semantic supervision. This auxiliary branch is not used for inference

$$Z_{\mathrm{aux}} = H_4\left(F_4^N\right) \in \mathrm{R}^{4\times 64\times 64} \tag{11}$$

The auxiliary branch strengthens intermediate semantic supervision only during training and does not contribute to the final inference output. The main-branch logits

are bilinearly upsampled to the original input size. Pixel-wise argmax then produces the four-class semantic segmentation map.

2.2. Semantic-Preserving Shallow Residual Detail Injection

Parking lines in underground garages are elongated, sparse, and scale-variable, and their pixels occupy only a small proportion of a bird's-eye-view image. Perspective projection further reduces the apparent width of distant lines, so repeated downsampling can attenuate local responses or fragment line structures. At the same time, cracks, construction joints, stains, shadows, and reflective regions can generate similar high-frequency responses. High-resolution shallow features therefore contain both useful geometric evidence and distracting non-target textures.

Shallow features retain spatial detail, whereas deeper features provide stronger semantic discrimination. For elongated targets, indiscriminate shallow-feature fusion may amplify background textures together with useful line evidence. In baseline YOLO26n-sem, Backbone-P2/4 preserves local edges and narrow structures, while Neck-P3/8 already integrates high-level semantics propagated from P4 and P5. We therefore propose Semantic-Preserving Shallow Residual Detail Injection (SRDI). Unlike a conventional concatenation-based fusion block, an added P2 prediction level, or a spatially gated attention module, SRDI leaves the original Neck-P3 pathway and segmentation head unchanged and adds a scale-aligned P2 projection only as a globally scaled residual (Figure 1).

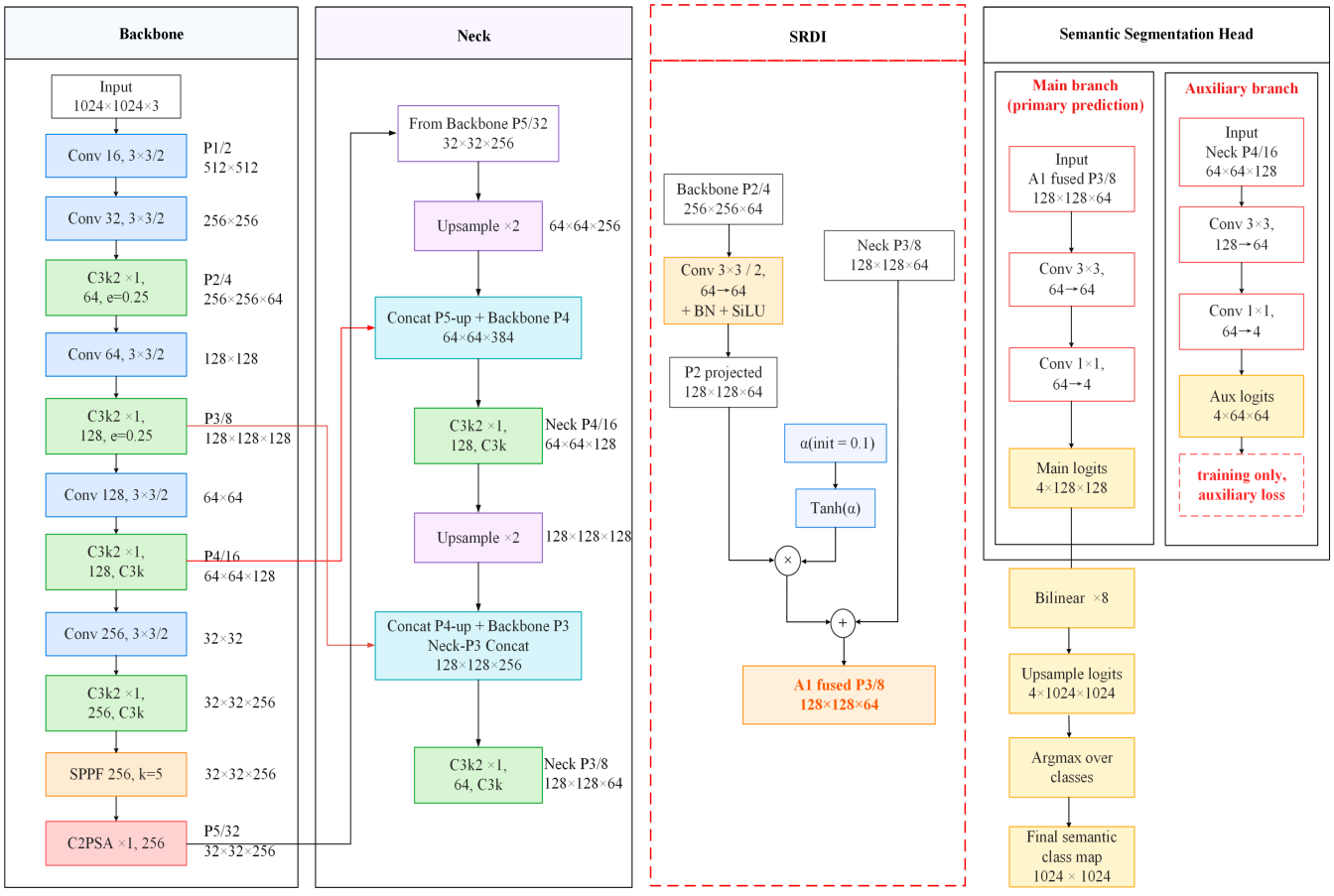

Figure 1. Overall architecture of the network with the SRDI module. Arrows indicate feature propagation, while different colors distinguish network components and operations.

2.2.1. Shallow-Feature Scale Alignment and Detail Projection

Let the Backbone P2/4 feature be F_2^B and the Neck P3/8 feature be F_3^N. The two features have the same channel count but differ by a factor of two in spatial scale, preventing direct element-wise fusion. We therefore use a learnable downsampling convolution F_2^B with a 3×3 kernel and stride 2. Batch normalization and a SiLU activation then perform scale alignment and local feature re-encoding. The shallow-detail projection is expressed as

$$D_2 = \phi\left(F_2^B\right) = \mathrm{SiLU}\left(\mathrm{BN}\left(W_{2\to3}^{3\times3,s=2} * F_2^B\right)\right) \tag{12}$$

where $W_{2\to3}^{3\times3,s=2}$ denotes a 3×3 convolution with stride 2, and * denotes convolution. The convolution retains 64 output channels, yielding

$$D_2 \in \mathrm{R}^{64\times\frac{H}{8}\times\frac{W}{8}} \tag{13}$$

The projected D2 feature matches Neck-P3 in both spatial size and channel dimension, allowing element-wise residual fusion. Unlike fixed pooling or interpolation, the stride-2 convolution performs scale conversion while learnably re-encoding local neighborhood responses. SRDI does not concatenate channels. It therefore leaves the input dimension of the Semantic Segment Head and the structure of the main segmentation head unchanged. This design exploits the complementary roles of P2 and P3: P2 retains fine spatial details relevant to thin parking lines, whereas P3 incorporates higher-level semantic context. Therefore, SRDI uses P2 as a supplementary detail source while retaining the original P3 prediction pathway, avoiding the additional complexity of a separate high-resolution prediction head.

2.2.2. Bounded Shallow Residual Injection and Semantic Preservation

After spatial and channel alignment, assigning equal weights to the projected P2 feature and Neck-P3 would assume equal importance of shallow detail and deep semantics. Such fusion could amplify non-target responses from cracks, joints, and reflections. We instead introduce a learnable global scalar α and map it through the hyperbolic tangent function to obtain the bounded coefficient γ.

$$\gamma = \tanh(\alpha) \tag{14}$$

The learnable scalar α is initialized to 0.1, corresponding to an initial residual coefficient of approximately 0.0997. This initialization introduces the projected P2 feature with a small initial scaling factor, while allowing its global contribution to be adjusted during training. The scalar α was jointly optimized with the network parameters using AdamW, reaching a final learned value of 0.135986 (γ = 0.135154). Because $-1 < \gamma < 1$, the coefficient controls the global scaling of the projected shallow feature and prevents unconstrained amplification of this coefficient. The bound does not constrain the absolute magnitude of the projected feature itself. A positive γ produces additive global modulation, whereas a negative γ produces subtractive global modulation; γ is not a spatially or channel-wise selective gate.

The SRDI output feature is defined as

$$\tilde{F_3^N} = F_3^N + \gamma D_2 = F_3^N + \tanh(\alpha) D_2 \tag{15}$$

where

$$\tilde{F_3^N} \in \mathrm{R}^{64\times\frac{H}{8}\times\frac{W}{8}} \tag{16}$$

Its spatial dimensions and channel count are identical to those of the original Neck-P3 feature. After introducing SRDI, the original main classification head therefore requires no redesign. It can directly use $\widetilde{F}_3^N$ instead of F_3^N for the main semantic prediction.

This formulation preserves the identity pathway of Neck-P3 and introduces the projected P2 feature only as a complementary residual. It therefore avoids directly replacing the main semantic representation with shallow features. The additional computation arises mainly from the P2-to-P3 projection convolution; the subsequent Neck and Semantic Segment Head remain unchanged. The learned scalar γ controls the global residual scale and should not be interpreted as the exact contribution ratio of P2 or as a content-adaptive attention map. The projected P2 features and residual scaling coefficient are jointly optimized under the segmentation loss, allowing the network to learn task-relevant shallow representations rather than indiscriminately transferring all local textures. Nevertheless, the global residual coefficient does not explicitly suppress spatially localized noise, and its effectiveness in balancing detail recovery and background interference is evaluated empirically in Section 3.4.

SRDI uses a stride convolution to align Backbone-P2 with Neck-P3 and injects the projected feature through a bounded global residual coefficient. The main Neck-P3 semantic pathway and the Semantic Segment Head are preserved. The design is intended to recover local evidence for thin parking lines while limiting the structural disruption caused by shallow-feature fusion.

2.3. Interface-Preserving Deep Semantic Slimming

After SRDI is introduced, high-resolution P2/4 features supplement the shallow detail available at Neck-P3. Resource-constrained deployment nevertheless requires control of model size and measured runtime. P5/32 is the deepest and lowest-resolution level in the YOLO26n-sem Backbone. It extracts high-level semantics and supplies deep context to the Neck. For a 1024×1024 input, P5 has a feature size of $256 \times 32 \times 32$. C3k2, SPPF, and C2PSA successively transform this feature before it is upsampled and fused with Backbone-P4. Although P5 does not directly perform pixel classification, it remains an important input to the top-down P5-to-P4-to-P3 semantic propagation pathway.

A common lightweight design directly reduces the number of P5 output channels. P5, however, is not an isolated terminal feature level. Its output is upsampled, concatenated with Backbone-P4, and propagated to the main P3 prediction feature. Reducing the external P5 width therefore decreases P5 capacity while also changing the Neck-P4 concatenation dimension and subsequent fusion interfaces. Direct channel reduction couples internal capacity changes with cross-level interface changes, complicating targeted slimming while preserving the original Neck.

We therefore introduce Interface-Preserving Deep Semantic Slimming (IPDS), which decouples the internal capacity of the deep P5 modules from the external semantic-interface width received by the Neck. IPDS retains the P5 input and output channels, the P5-to-P4-to-P3 fusion pathway, SRDI, and the Semantic Segment Head. Only the principal hidden widths within the P5 C3k2, SPPF, and C2PSA modules are reduced. This change lowers internal representational capacity and parameterization without altering the cross-scale interfaces; it does not constitute channel pruning. Figure 2 shows the internal structures of C3k2, SPPF, and C2PSA.

Let the original principal hidden width of the m-th deep feature-transformation module in P5 be $C_{h,m}$. IPDS compresses this width to

$$C'_{h,m} = \rho C_{h,m} \tag{17}$$

where ρ is the internal-capacity compression ratio.

We apply a uniform hidden-capacity ratio to avoid introducing module-specific channel-search hyperparameters. Based on the sensitivity analysis presented in Section 3.4.3, the compression ratio is set to $\rho = 0.50$ in the final model, reducing the principal internal width from 128 to 64 while retaining the 256-channel external semantic interface. This setting provides a favorable balance between segmentation accuracy and model efficiency among the evaluated compression ratios and is not intended to represent a theoretically optimal value.

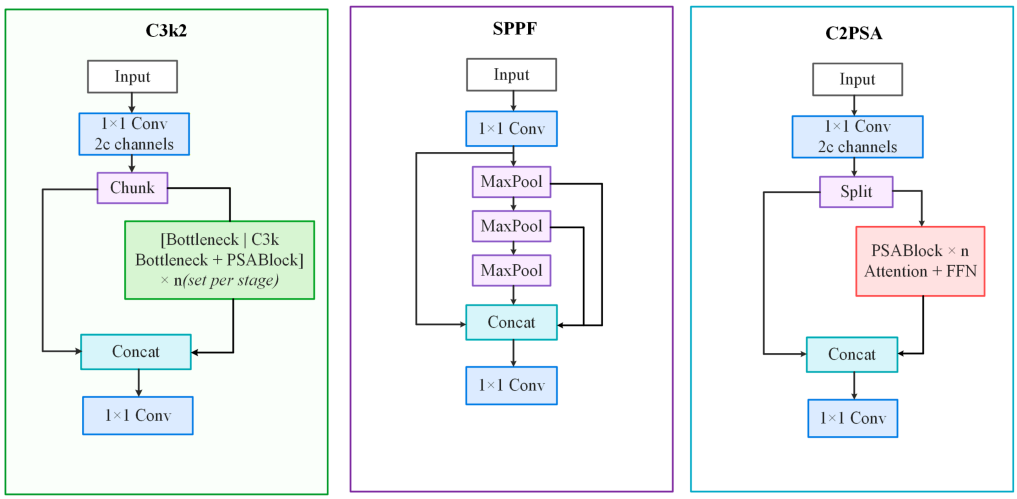

Figure 2. Internal structures of the C3k2, SPPF, and C2PSA modules. Arrows indicate feature flow, and different colors distinguish the modules and their internal operations.

The principal internal hidden width of P5 is therefore reduced from 128 to 64. In the equation, $C_{h,m}$ denotes the principal hidden representation width rather than a uniform channel count assigned mechanically to every internal convolution. C3k2, SPPF, and C2PSA retain their original feature-transformation mechanisms. Their principal hidden representations are reduced to 50% according to their existing channel-generation rules. IPDS therefore changes neither their basic topologies nor the set of branches and operators.

While reducing internal capacity, IPDS explicitly preserves the external input and output widths of P5:

$$C_{\mathrm{P5,in}} = C_{\mathrm{P5,out}} = 256 \tag{18}$$

The slimmed P5 output therefore still satisfies

$$F_5' \in \mathrm{R}^{256 \times \frac{H}{32} \times \frac{W}{32}} \tag{19}$$

After 2× upsampling, the 256-channel P5 feature is still concatenated with the 128-channel Backbone-P4 feature. The Neck-P4 input after IPDS therefore remains

$$C_{\mathrm{Neck\text{-}P4}}^{\mathrm{IPDS}} = 256 + 128 = 384 \tag{20}$$

Consequently, IPDS requires no changes to the P5 output interface, Neck-P4, subsequent Neck-P3, SRDI, or the Semantic Segment Head. The original P5-to-P4-to-P3 semantic propagation pathway is preserved. Deep convolutions, feature concatenation, and attention-related operations are performed in a narrower internal representation space, reducing deep-stage parameterization and theoretical computation. This module-level effect should not be conflated with the GFLOPs of the complete SRDI + IPDS network, which also include the added SRDI projection and are reported separately in Section 3.2.

2.4. Loss Function

Our modifications affect feature extraction and propagation. To maintain identical training conditions, neither SRDI nor IPDS introduces an additional supervision term. We retain the YOLO26n-sem semantic segmentation objective. During training, P3/8 provides the main prediction branch and P4/16 provides an auxiliary branch. The main branch is supervised jointly by cross-entropy (CE) and Dice losses, while the auxiliary branch uses CE for deep supervision. The YOLO26 semantic segmentation implementation uses the same loss combination and assigns a weight of 0.4 to the auxiliary CE term.

The CE loss supervises pixel-wise class prediction. The Dice loss optimizes overlap between the predicted and ground-truth regions, reducing the influence of the dominant background on sparse classes such as parking lines. The main-branch loss is expressed in simplified form as

$$\mathcal{L}_{main} = \mathcal{L}_{CE} + \mathcal{L}_{Dice} \tag{21}$$

For the multiclass Dice loss, the Dice coefficient is computed for each class over the valid pixels in the current mini-batch. The four class-wise coefficients are then averaged with equal weights, using a smoothing term of 1.0. Label value 255 denotes ignored regions and is excluded from both CE and Dice calculations.

The P4/16 auxiliary prediction is further used to compute an auxiliary CE loss during training. The total loss is therefore

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{Dice} + 0.4\mathcal{L}_{CE}^{aux} \tag{22}$$

Before loss calculation, logits from both branches are restored to the label size by bilinear interpolation. Pixels with label value 255 are ignored in the CE and Dice losses. The auxiliary branch is removed during inference, and only the main P3 branch generates the segmentation map. Auxiliary supervision therefore adds no deployment-time inference cost. The P4 auxiliary loss follows the original YOLO26n-sem training configuration and provides additional supervision for intermediate semantic representations, facilitating feature optimization along the P4-to-P3 pathway. This supervision is consistently retained across all YOLO26-based variants to ensure comparable training conditions, while its independent contribution to segmentation accuracy is not separately quantified in this study.

3. Results

3.1. Dataset and Experimental Settings

3.1.1. Dataset

We evaluate the proposed method on a manually annotated dataset collected in one underground garage. The dataset was constructed from four distinct driving sequences acquired within the same underground garage. Nonadjacent frames were sampled at fixed temporal intervals to reduce redundancy, and the dataset was partitioned at the sequence level. The four sequences were kept mutually exclusive across the training, validation, and test subsets, ensuring that all images from the same sequence were assigned to only one subset. The resulting dataset contains 960 RGB bird's-eye-view images resized to 1024×1024 pixels, with 640, 160, and 160 images used for training, validation, and testing, respectively. The training subset was used for parameter optimization, the fixed validation subset for model selection and ablation studies, and the test subset for final evaluation after the experimental protocol had been fixed. Although this sequence-disjoint partition reduces temporal overlap among the subsets, all sequences were collected in the same underground garage; therefore, the internal test results represent within-garage rather than cross-garage evaluation.

Pixel-level semantic labels were annotated manually. Based on the road structure relevant to automated parking, valid pixels were assigned to four classes: background, parking line, guide mark, and auxiliary line. Labelme (version 7.0.4) was used to delineate polygons along the visible target boundaries. Regions with uncertain classes, severe occlusion, or ambiguous boundaries were assigned the ignore label 255. These regions were excluded from training and evaluation.

3.1.2. Experimental Environment and Training Parameters

All YOLO26 variants and ablation experiments were run in the same software and hardware environment. The environment comprised Python 3.11.15, PyTorch 2.13.0+cu126, PyTorch CUDA runtime 12.6, and Ultralytics 8.4.114 on an NVIDIA GeForce RTX 4060 Laptop GPU(NVIDIA Corporation, Santa Clara, CA, USA). The main loss combined pixel-wise CE, four-class macro-averaged Dice, and auxiliary P4 CE during training. Table 1 lists the complete training configuration.

Table 1. Summary of the main training settings.

Parameter	Setting
Input size	1024×1024
Training/validation/test split	640/160/160
Number of classes	4
Ignore index	255
Batch size	4
Optimizer	AdamW
Initial learning rate	1×10^{-3}
Final learning-rate factor	0.01
Weight decay	5×10^{-4}
Maximum epochs	150
Early-stopping patience	30
Random seed	42
Deterministic mode	True
Training precision	AMP
Main loss	CE + macro Dice
Auxiliary supervision	Original P4 auxiliary CE; each comparison model retains its native training-only auxiliary design
GPU	RTX 4060 Laptop GPU

3.1.3. Evaluation Metrics

We evaluate overall segmentation accuracy, parking-line recognition, boundary quality, thin-line recovery, difficult-sample behavior, and model efficiency. Mean Intersection over Union (mIoU) and Parking IoU are treated as the primary region-level criteria. Precision, Recall, F1 score, boundary F1 with a 3-pixel tolerance (BF1@3 px), width-stratified Recall, P10, and Worst10 serve as complementary diagnostic metrics. Efficiency is characterized using model parameters, GFLOPs, inference latency, and FPS. Selected metrics are defined below.

$$IoU_k = \frac{TP_k}{TP_k + FP_k + FN_k} \tag{23}$$

$$mIoU = \frac{1}{K}\sum_{k=1}^{K} IoU_k \tag{24}$$

$$P = \frac{TP}{TP + FP} \tag{25}$$

$$R = \frac{TP}{TP + FN} \tag{26}$$

$$F_1 = \frac{2PR}{P + R} \tag{27}$$

Here, TP_k, FP_k, and FN_k denote the true-positive, false-positive, and false-negative pixel counts for class k, respectively. K is the number of classes and is set to 4. Precision reflects false parking line predictions, whereas Recall reflects missed ground-truth parking line pixels.

Parking lines are elongated targets, so region IoU alone may underrepresent local boundary deviations. We therefore report BF1@3 px for parking-line boundaries using a 3-pixel matching tolerance. BF1 is defined as

$$BF_1 = \frac{2P_B R_B}{P_B + R_B} \tag{28}$$

where P_B and R_B denote boundary Precision and Recall within the 3-pixel tolerance, respectively. Boundary-sensitive evaluation complements mask IoU, which is less sensitive to local edge errors.

Parking-line pixels are stratified according to the local width of the ground-truth annotations, defined as the pixel thickness of the annotated line region measured across the local line direction at the 1024 × 1024 evaluation resolution. Three fixed width groups are used: thin lines with a local width of no more than 16 pixels, medium lines with a width greater than 16 pixels and no more than 18 pixels, and thick lines with a width greater than 18 pixels. Recall is computed separately for each width group, and Thin Recall refers to the recall of parking-line pixels belonging to the thin-line group. The same width thresholds and evaluation procedure are applied to all evaluated models. P10 is the 10th percentile of image-wise Parking IoU values, and Worst10 is the mean Parking IoU of the 10 lowest-performing images. Efficiency is evaluated using parameter count, GFLOPs at an input resolution of 1024 × 1024, batch-1 inference latency, and FPS. Latency is measured on the same NVIDIA GeForce RTX 4060 Laptop GPU with all models operating in evaluation mode and under identical numerical precision. Before timing, 100 warm-up iterations are performed to stabilize GPU execution. The input tensor is preloaded onto the GPU, after which latency is averaged over 1000 forward passes using CUDA synchronization to account for asynchronous GPU execution. Data loading, image decoding, and host-to-device transfer are excluded from the timing procedure so that the reported latency reflects model inference rather than the surrounding data pipeline. FPS is calculated from the measured mean single-frame latency. The same input resolution, batch size, precision, and timing protocol are applied to all evaluated models.

3.1.4. External Dataset and Cross-Garage Evaluation Protocol

To evaluate whether the observed improvements extend beyond the garage used for model development, an independent public underground-parking dataset introduced by Liu et al. [42] was used for cross-garage evaluation. The dataset was collected in a real underground parking lot using four vehicle-mounted fisheye cameras to generate bird's-eye-view images. The released images used in this study have an original spatial size of approximately 1090 × 860 pixels.

From the 700 images available in the independent external dataset, 200 labeled image–mask pairs were selected through purposive rather than random sampling. The selection aimed to cover diverse challenging conditions, including low illumination, reflections, worn parking markings, partial occlusions, and distracting floor textures. All samples were selected before model inference, independently of model predictions, and consistently used

for evaluating all methods. This sampling strategy was intended to assess cross-garage performance under challenging visual conditions rather than to provide a statistically representative random sample of the entire dataset.

The annotation taxonomy of the external dataset differs slightly from that of our dataset. The released masks contain four semantic categories—background, groundline, parkingline, and traffic-arrow—whereas our dataset uses background, parking line, guide mark, and auxiliary line. To enable consistent four-class evaluation, the external annotations were harmonized with the target taxonomy according to the fixed mapping shown in Table 2.

Table 2. Mapping from the released external-mask categories to the target taxonomy.

External Category	Target Category	Target ID
background	background	0
parkingline	parking line	1
traffic-arrow	guide mark	2
groundline	auxiliary line	3

The semantic mapping was defined before inference and applied identically to all evaluated models. No model-specific or prediction-dependent modification of the external annotations was performed.

The external dataset was used exclusively for testing. No external image was used for training, validation, model selection, hyperparameter tuning, or checkpoint selection. All models were evaluated using the frozen checkpoints obtained from the original dataset, without fine-tuning, test-time adaptation, threshold adjustment, or additional post-processing. External images were processed at the same 1024 × 1024 input resolution used in the original experiments, and annotation masks were resized using nearest-neighbor interpolation to preserve discrete class labels. This protocol therefore constitutes a zero-shot cross-garage evaluation under a change in acquisition environment.

3.2. Comparative Experiments

We compare the proposed model with five representative real-time semantic segmentation networks: Fast-SCNN [17], BiSeNet V2 [18], DDRNet-23-Slim [20], PIDNet-S [21], and GCNet-S [23]. Evaluation uses the same test images, 1024 × 1024 inputs, batch size of 1, ignore index of 255, and a single main prediction pathway for all models. Testing was performed after model selection, and the reported test results were not used for subsequent architecture or hyperparameter selection. Tables 3 and 4 report the comparisons.

Table 3. Comparison of segmentation performance (internal test set, n = 160).

Model	mIoU	Parking IoU	Parking Recall	Parking F1	BF1@3 px	Thin Recall	P10	Worst10
Fast-SCNN	84.354	81.122	87.262	90.244	92.172	86.718	72.141	66.657
BiSeNetV2	85.391	82.857	87.980	90.625	92.004	87.462	74.068	63.089
DDRNet-23-Slim	84.871	81.163	86.075	89.602	90.833	85.466	68.963	59.270
PIDNet-S	84.928	82.125	87.113	90.185	91.598	86.530	72.476	64.302
GCNet-S	84.998	81.621	87.739	90.484	91.472	85.254	73.456	65.414
YOLO26n-sem	85.466	82.108	86.718	90.295	92.329	86.144	74.354	67.808
Ours	85.966	83.081	88.229	90.759	92.366	87.700	76.549	68.317

Table 4. Comparison of model complexity and inference efficiency.

Model	Parameters (M)	GFLOPs	Latency (ms)	FPS
Fast-SCNN	1.435	3.7105	7.8737	127.01
BiSeNet V2	3.492	49.1754	15.6503	63.90
DDRNet-23-Slim	5.732	18.2398	6.6982	149.29
PIDNet-S	7.717	23.7271	15.4533	64.71
GCNet-S	9.169	22.1020	7.9680	125.50
YOLO26n-sem	1.631	11.7359	8.8203	113.37
Ours	1.225	11.8186	7.0321	142.21

As shown in Table 3, the proposed model achieves the best overall segmentation performance, with an mIoU of 85.966% and a Parking IoU of 83.081%. Compared with YOLO26n-sem, these values increase by 0.500 and 0.973 percentage points, respectively, while Thin Recall improves by 1.556 percentage points. The proposed model also obtains the highest Parking Recall, Parking F1, P10, and Worst10 among the evaluated methods. GCNet-S provides competitive performance, reaching 84.998% mIoU and 81.621% Parking IoU, but remains below the proposed model in the primary region-level and difficult-sample metrics. In contrast, the improvement in BF1@3 px is marginal, indicating that the main gains arise from parking-line region accuracy, thin-line recovery, and robustness on difficult samples rather than boundary refinement.

Table 4 shows that the proposed model contains 1.225 M parameters, the smallest among the compared methods and approximately 24.9% fewer than YOLO26n-sem. Its computational cost remains close to that of the baseline (11.8186 versus 11.7359 GFLOPs), while latency decreases from 8.8203 to 7.0321 ms, corresponding to 142.21 FPS. Although DDRNet-23-Slim achieves lower latency and Fast-SCNN requires fewer GFLOPs, both yield lower segmentation accuracy. GCNet-S also provides competitive inference speed but requires substantially more parameters and computation. The proposed model does not minimize every individual complexity metric, but it combines higher segmentation accuracy with substantially fewer parameters and lower latency than YOLO26n-sem.

3.3. Cross-Garage External Evaluation

To examine whether the performance gains observed on the original test set persist beyond the model-development environment, all frozen models were further evaluated on the independent external underground-garage dataset described in Section 3.1.4. Table 5 reports the zero-shot cross-garage results using the harmonized four-class taxonomy.

Table 5. Zero-shot cross-garage performance on the external test set (n = 200).

Model	mIoU	Parking IoU	Parking Recall	P10	Worst10
Fast-SCNN	73.612	65.427	73.905	48.214	40.736
BiSeNet V2	74.235	68.816	75.122	52.607	37.241
DDRNet-23-Slim	72.748	64.593	72.431	44.726	32.618
PIDNet-S	75.306	67.154	75.217	50.633	38.275
GCNet-S	74.173	67.829	74.362	51.742	40.319
YOLO26n-sem	76.421	69.708	74.584	54.116	45.204
Ours	77.837	72.625	78.342	59.268	48.413

As shown in Table 5, the proposed model achieves the highest values across all reported external-domain metrics, with an mIoU of 77.837%, a Parking IoU of 72.625%, and a Parking Recall of 78.342%. Compared with YOLO26n-sem, these metrics improve by 1.416, 2.917, and 3.758 percentage points, respectively. The proposed model also increases

P10 from 54.116% to 59.268% and Worst10 from 45.204% to 48.413%, corresponding to gains of 5.152 and 3.209 percentage points. The comparatively larger improvements in Parking Recall and the lower-tail metrics indicate that the benefit is particularly evident in recovering parking-line regions and maintaining performance on difficult external samples.

A clear performance decrease is nevertheless observed when transferring from the original test environment to the external garage, indicating a substantial domain shift between the two evaluation settings. For YOLO26n-sem, mIoU decreases from 85.466% to 76.421% and Parking IoU from 82.108% to 69.708%, corresponding to reductions of 9.045 and 12.400 percentage points, respectively. For the proposed model, the corresponding decreases are 8.129 percentage points in mIoU and 10.456 percentage points in Parking IoU. Thus, in addition to retaining higher absolute accuracy on the external dataset, the proposed model exhibits smaller degradation in both primary region-level metrics than the baseline.

Overall, the external evaluation shows that the performance gains of the proposed model are partially retained under the tested cross-garage domain shift. The higher Parking IoU, Parking Recall, P10, and Worst10 relative to YOLO26n-sem indicate improved parking-line recovery and lower-tail performance on the external dataset. Nevertheless, the absolute performance decrease relative to the original test set demonstrates that variations in illumination, floor appearance, marking configuration, scene geometry, and imaging characteristics remain challenging. These results should therefore be interpreted as evidence of improved generalization under the tested zero-shot cross-garage setting rather than universal robustness across arbitrary parking facilities or imaging systems.

3.4. Ablation Studies

3.4.1. Overall Effects of SRDI and IPDS

To isolate the effects of SRDI and IPDS on segmentation accuracy, model complexity, and inference latency, we construct three variants of the YOLO26n-sem baseline. These variants use IPDS only, SRDI only, or both modules. Segmentation accuracy is evaluated on the same fixed validation set, while inference latency is measured using the unified protocol described in Section 3.1.3. The results are summarized in Table 6.

Table 6. Ablation results for SRDI and IPDS (validation set, n = 160).

Variant	SRDI	IPDS	mIoU	Parking IoU	BF1@3 px	Thin Recall	P10	Worst10	Parameters (M)	Latency (ms)
YOLO26n-sem			89.465	83.282	91.236	88.542	73.208	68.908	1.631	8.8203
A0		√	89.191	82.967	91.124	87.362	72.668	67.819	1.189	6.9352
A1	√		90.218	84.254	92.306	89.942	76.199	71.668	1.668	8.9423
Ours	√	√	90.065	83.981	92.146	89.250	75.649	70.317	1.225	7.0321

Note: √ indicates that the corresponding module is included in the model configuration.

The baseline achieves 89.465% mIoU and 83.282% Parking IoU with 1.631 M parameters. Adding SRDI alone increases mIoU and Parking IoU to 90.218% and 84.254%, corresponding to gains of 0.753 and 0.972 percentage points, respectively. BF1@3 px and Thin Recall increase by 1.070 and 1.400 percentage points, while P10 and Worst10 increase by 2.991 and 2.760 percentage points, respectively. The parameter count increases from 1.631 M to 1.668 M, corresponding to an increase of approximately 2.27%. Inference latency increases slightly from 8.8203 ms to 8.9423 ms, indicating that the accuracy improvements are achieved with limited additional runtime overhead. These validation results are consistent with the intended role of SRDI in supplementing shallow line detail, although the ablation alone does not isolate a causal feature-level mechanism.

With IPDS alone, the parameter count decreases by approximately 27.1%, from 1.631 M to 1.189 M. This reduction is accompanied by decreases of 0.274 and 0.315 percentage points in mIoU and Parking IoU, respectively. Meanwhile, inference latency decreases from 8.8203 ms to 6.9352 ms, corresponding to a reduction of 21.37%. This result confirms that IPDS improves inference efficiency independently of SRDI. IPDS therefore reduces internal model capacity and parameter count at the cost of a limited validation-accuracy degradation. It should not be characterized as lossless compression or as an accuracy-improving module.

Combining SRDI and IPDS yields 1.225 M parameters, approximately 24.9% fewer than the original YOLO26n-sem. Relative to the IPDS-only variant with 1.189 M parameters, adding SRDI increases the reported parameter count by approximately 0.036 M, corresponding to an increase of approximately 3.0%. The combined model reaches 90.065% mIoU, 83.981% Parking IoU, 92.146% BF1@3 px, and 89.250% Thin Recall. These values exceed the baseline by 0.600, 0.699, 0.910, and 0.708 percentage points, respectively, while P10 and Worst10 increase by 2.441 and 1.409 percentage points. Adding SRDI to the IPDS-only variant improves validation accuracy, allowing the combined model to exceed the baseline in the reported validation metrics while retaining a substantially lower parameter count. Compared with IPDS-only, the combined model improves mIoU by 0.874 percentage points, with an additional inference latency of only 0.0969 ms.

The combined SRDI + IPDS model remains less accurate than the SRDI-only variant under this validation ablation. Its value lies in retaining improvements over the baseline while reducing the parameter count to 1.225 M. Compared with the SRDI-only variant with 1.668 M parameters, the combined model reduces the parameter count by approximately 26.6%. The evidence therefore supports distinct roles: SRDI mainly contributes accuracy and thin-line recovery, whereas IPDS mainly controls deep internal capacity and model size. Overall, the combined model improves mIoU by 0.600 percentage points while reducing the parameter count by approximately 24.9% and inference latency by 20.27% relative to the baseline, demonstrating a favorable balance between segmentation accuracy, model size, and inference efficiency.

3.4.2. Structural Comparisons Around SRDI

To examine the structural choices around SRDI, we use the SRDI-only model (A1) as the reference. A2 retains A1's P2 projection and bounded global residual coefficient, but applies a spatial- and channel-dependent gate to the projected feature before residual addition. A3 retains A1's residual fusion and replaces only the stride-2, 3×3 P2 projection with PixelUnshuffle(2) followed by a 1×1 convolution, batch normalization, and SiLU. A4 retains the complete SRDI module and replaces only the Neck's P4-to-P3 nearest-neighbor upsampling with DySample; the P5-to-P4 upsampling remains unchanged. Thus, A2 examines local gating, A3 examines P2 scale alignment, and A4 serves as a separate control for upstream semantic upsampling. All configurations are evaluated on the same fixed validation set under the reported training protocol, and the results are summarized in Table 7.

As shown in Table 7, the original SRDI configuration (A1) achieves the highest mIoU (90.218%), Parking IoU (84.254%), P10 (76.199%), and Worst10 (71.668%) among the evaluated configurations. Compared with A1, the spatially and channel-wise gated A2 achieves higher Parking Recall (91.148% vs. 90.578%) and Thin Recall (90.580% vs. 89.942%), but lower Precision (88.906% vs. 89.951%) and Parking IoU (83.840% vs. 84.254%). This indicates that the additional gating improves line-pixel sensitivity but does not provide an overall advantage in region-level accuracy. For scale alignment, A3 replaces the original stride-2 convolution with PixelUnshuffle-based projection. Although A3 achieves higher

Precision (90.773%), its lower Recall (89.718%) and Parking IoU (83.221%) indicate a trade-off between prediction precision and parking-line recovery, supporting the selection of the original convolutional projection under the evaluated conditions.

Table 7. Performance comparison of SRDI and related structural variants on the fixed validation set (n = 160).

Method	mIoU	Parking IoU	Precision	Recall	BF1@3 px	Thin Recall	P10	Worst10
SRDI (A1)	90.218	84.254	89.951	90.578	92.306	89.942	76.199	71.668
A2	89.624	83.840	88.906	91.148	91.524	90.580	72.503	68.892
A3	89.858	83.221	90.773	89.718	92.085	89.062	76.133	71.524
A4	89.692	83.146	89.124	90.634	92.563	89.998	75.205	70.897

Unlike A2 and A3, A4 retains the complete SRDI module and modifies only the Neck's P4-to-P3 upsampling operation. A4 achieves a higher BF1@3 px (92.563% vs. 92.306%), but lower mIoU (89.692% vs. 90.218%) and Parking IoU (83.146% vs. 84.254%). This suggests that learnable upstream semantic upsampling can improve boundary matching without necessarily improving overall region overlap. Overall, these results support the original SRDI configuration as a favorable choice for balancing region-level accuracy and difficult-sample performance. The comparisons further indicate that additional local gating, alternative P2 projection, and modified upstream upsampling do not consistently improve the primary segmentation metrics under the tested protocol.

3.4.3. Sensitivity to the IPDS Compression Ratio

To examine the sensitivity of IPDS to the internal-capacity compression ratio, we vary ρ while keeping SRDI, the external P5 interface, network topology, training schedule, loss functions, and evaluation protocol unchanged. All variants are evaluated on the same fixed validation set. A ratio of ρ = 1.0 corresponds to the uncompressed SRDI-only configuration, whereas smaller values progressively reduce the principal hidden widths of the P5 modules. Table 8 summarizes the results.

Table 8. Sensitivity analysis of the IPDS compression ratio (validation set, n = 160).

Compression Ratio	Hidden Width	mIoU	Parking IoU	Parameters (M)	Latency (ms)
1.00	128	90.218	84.254	1.668	8.9423
0.75	96	90.163	84.107	1.425	7.9682
0.50	64	90.065	83.981	1.225	7.0321
0.25	32	88.986	82.362	1.047	6.3742

The results show that moderate IPDS compression substantially reduces model size and inference latency with only minor accuracy degradation. When ρ is reduced from 1.00 to 0.50, mIoU and Parking IoU decrease by only 0.153 and 0.273 percentage points, respectively. Meanwhile, the parameter count decreases from 1.668 M to 1.225 M, corresponding to a reduction of approximately 26.6%, and latency decreases from 8.9423 ms to 7.0321 ms, corresponding to a reduction of approximately 21.4%. In contrast, reducing ρ to 0.25 leads to more pronounced declines of 1.232 percentage points in mIoU and 1.892 percentage points in Parking IoU relative to the uncompressed configuration with ρ = 1.00, indicating that excessive compression begins to constrain deep semantic representation. Therefore, ρ = 0.50 is adopted as the final IPDS setting because it provides a favorable balance between segmentation accuracy, model size, and measured inference latency.

3.5. Failure-Case Analysis

Although the proposed model achieves improved quantitative performance on both the original and external test sets, several challenging cases remain unresolved. To examine the remaining failure modes of the proposed model, Figure 3 presents four representative examples from the original test set. Each case illustrates challenging conditions involving weak or ambiguous parking-line evidence and visually similar background structures.

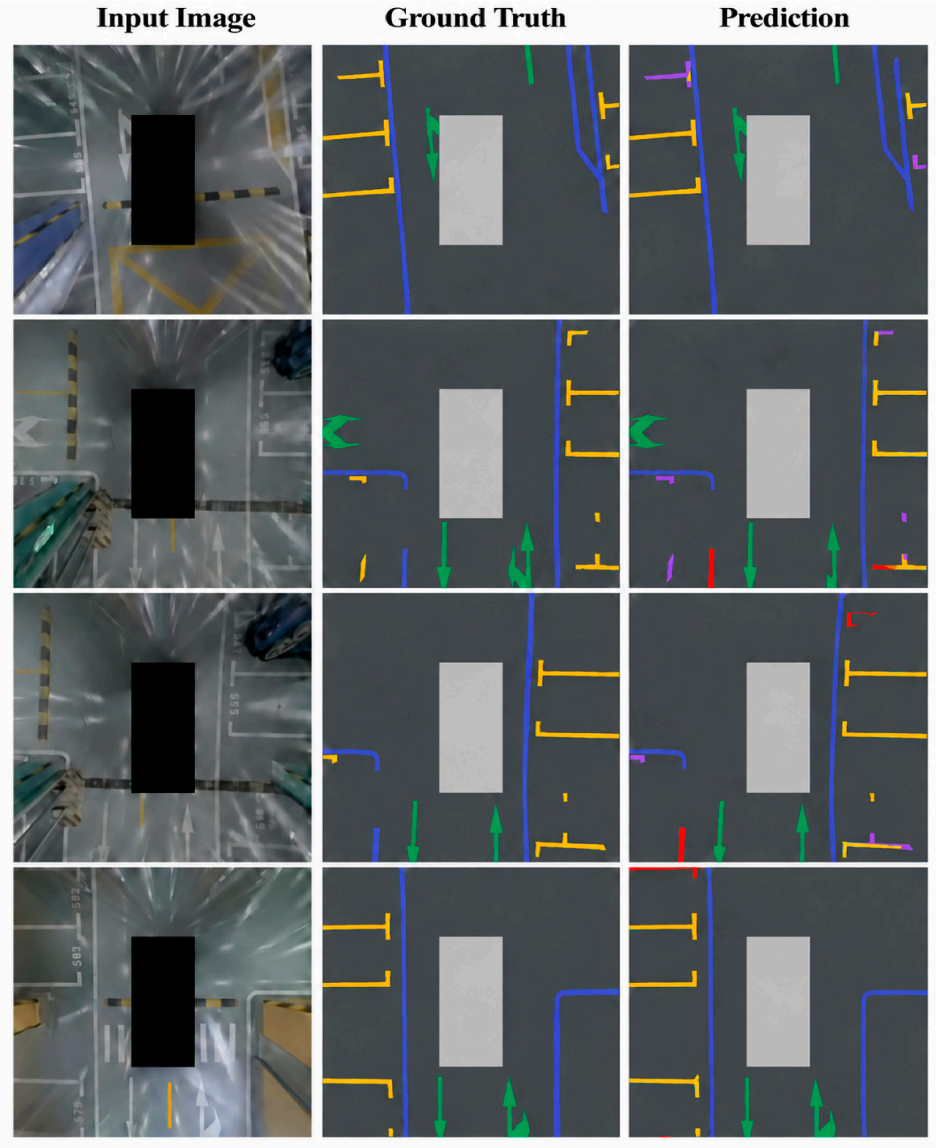

Figure 3. Representative failure cases of the proposed model on the original test set. Each row shows one challenging example, with the input image, ground-truth annotation, and model prediction shown from left to right. Yellow, blue, and green denote parking-line, auxiliary-line, and guide-mark regions, respectively, while the dark background represents the background class. Red and purple in the prediction maps indicate false-positive (FP) and false-negative (FN) parking-line responses, respectively. The central gray region denotes the ignored ego-vehicle area and is excluded from evaluation.

Figure 3 shows that perspective compression, worn markings, and local occlusion can weaken the visual evidence of true parking lines, leading to missed or fragmented predictions. In contrast, uneven illumination, floor reflections, and competing linear textures may generate spurious responses. To quantify these failure patterns, an exploratory image-level analysis was conducted on all 160 test images by examining the original images,

ground-truth annotations, and model predictions. Failure categories were assigned based on the correspondence between observed segmentation errors and visual conditions. Local errors involving perspective-compressed or peripheral thin parking lines were observed in 66 images (41.25%), followed by worn or occluded markings in 28 images (17.50%). Illumination-related errors and false positives associated with linear background structures were each observed in 10 images (6.25%). Additionally, seven images exhibited linear false positives of uncertain origin. These categories may overlap, and the reported proportions indicate image-level occurrences of local errors rather than whole-image failure rates. Future work will focus on improving cross-scale preservation of weak line structures and suppressing responses to visually similar background patterns.

4. Conclusions and Future Work

This work develops a lightweight YOLO26n-sem-based parking-line segmentation model by combining Semantic-Preserving Shallow Residual Detail Injection (SRDI) with Interface-Preserving Deep Semantic Slimming (IPDS). SRDI supplements the Neck-P3 representation with scale-aligned shallow details while preserving the original semantic pathway, whereas IPDS reduces the internal capacity of the deep P5 modules without altering the external P5 interface or the P5-to-P4-to-P3 propagation topology. Ablation results confirm their complementary roles: SRDI primarily improves segmentation accuracy and thin-line recovery, while IPDS mainly reduces model size. The compression-ratio sensitivity analysis further shows that $\rho = 0.50$ provides a favorable accuracy–efficiency operating point among the evaluated settings. Their combination therefore achieves a practical balance between segmentation performance and model efficiency.

On the internal test set, the proposed model achieves 85.966% mIoU, 83.081% Parking IoU, and 87.700% Thin Recall, exceeding YOLO26n-sem by 0.500, 0.973, and 1.556 percentage points, respectively. The parameter count is reduced by approximately 24.9%, from 1.631 M to 1.225 M, while single-frame latency decreases from 8.8203 ms to 7.0321 ms, corresponding to 142.21 FPS. Although GFLOPs increase slightly from 11.7359 to 11.8186, the overall results demonstrate a favorable accuracy–size–latency trade-off. Zero-shot evaluation on an independent public underground-garage dataset further achieves 77.837% mIoU and 72.625% Parking IoU, outperforming YOLO26n-sem by 1.416 and 2.917 percentage points, respectively. These results indicate that the performance advantage of the proposed framework is partially retained under the tested cross-garage domain shift.

The current evaluation remains limited to one model-development garage and one additional public environment; therefore, broader cross-garage and cross-device generalization has yet to be established. In addition, runtime measurements were obtained using PyTorch on an RTX 4060 Laptop GPU and do not represent embedded automotive deployment. Difficult cases involving severe perspective compression, worn markings, occlusion, reflections, and competing linear structures also remain challenging. Future work will therefore extend evaluation to additional garages and imaging systems, improve the recovery of weak parking-line structures and suppression of spurious linear responses, and investigate hardware-aware deployment using optimized inference backends such as TensorRT.

Author Contributions: X.S. and Y.W. collected and made training samples, D.S. designed the experimental scheme and algorithm, X.S. and G.C. built the training model, Z.G. compiled and organized the experimental data, X.S. wrote the paper with the help of Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Scientific and Technological Research Project of Henan Province (No. 232102240022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The self-constructed dataset is not publicly available due to privacy considerations associated with potentially identifiable information, including license plates, but may be obtained from the corresponding author upon reasonable request for academic use. The external dataset used for cross-garage evaluation is publicly available from Liu et al. [42].

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

References

1. Wong, G.S.; Goh, K.O.M.; Tee, C.; Md. Sabri, A.Q. Review of vision-based deep learning parking slot detection on surround view images. *Sensors* **2023**, *23*, 6869. [CrossRef]
2. Ma, Y.; Liu, Y.; Zhang, L.; Cao, Y.; Guo, S.; Li, H. Research review on parking space detection method. *Symmetry* **2021**, *13*, 128. [CrossRef]
3. Ma, Y.; Liu, Y.; Shao, S.; Zhao, J.; Tang, J. Review of research on vision-based parking space detection method. *Int. J. Web Serv. Res.* **2022**, *19*, 1–25. [CrossRef]
4. Suhr, J.K.; Jung, H.G. End-to-end trainable one-stage parking slot detection integrating global and local information. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4570–4582. [CrossRef]
5. Bui, Q.H.; Suhr, J.K. CNN-based two-stage parking slot detection using region-specific multi-scale feature extraction. *IEEE Access* **2023**, *11*, 58491–58505. [CrossRef]
6. Bui, Q.H.; Suhr, J.K. One-stage parking slot detection using component linkage and progressive assembly. *IEEE Intell. Transp. Syst. Mag.* **2023**, *15*, 33–48. [CrossRef]
7. Min, C.; Xu, J.; Xiao, L.; Zhao, D.; Nie, Y.; Dai, B. Attentional graph neural network for parking-slot detection. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3445–3450. [CrossRef]
8. Lai, C.; Yang, Q.; Guo, Y.; Bai, F.; Sun, H. Semantic segmentation of panoramic images for real-time parking slot detection. *Remote Sens.* **2022**, *14*, 3874. [CrossRef]
9. Huang, C.; Yang, S.; Luo, Y.; Wang, Y.; Liu, Z. Visual detection and image processing of parking space based on deep learning. *Sensors* **2022**, *22*, 6672. [CrossRef]
10. Chen, Z.; Qiu, J.; Sheng, B.; Li, P.; Wu, E. GPSD: Generative parking spot detection using multi-clue recovery model. *Vis. Comput.* **2021**, *37*, 2657–2669. [CrossRef]
11. Zhou, S.; Yin, D.; Lu, Y. PASSIM: Parking slot recognition using attentional semantic segmentation and instance matching. In Proceedings of the 2022 IEEE 5th International Conference on Big Data and Artificial Intelligence (BDAI), Fuzhou, China, 8–10 July 2022; pp. 169–175. [CrossRef]
12. Ning, Y.; Liu, S.; Yan, D. Lightweight gated parallel fusion of CoordAtt and ASPP for degradation-robust monocular parking-line segmentation. *Appl. Sci.* **2026**, *16*, 8203. [CrossRef]
13. Hou, Y.; Ma, Z.; Liu, C.; Hui, T.-W.; Loy, C.C. Inter-region affinity distillation for road marking segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12486–12495. [CrossRef]
14. Wang, S.; Nguyen, C.; Liu, J.; Zhang, K.; Luo, W.; Zhang, Y.; Muthu, S.; Maken, F.A.; Li, H. Homography guided temporal fusion for road line and marking segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 1075–1085. [CrossRef]
15. Zheng, T.; Huang, Y.; Liu, Y.; Tang, W.; Yang, Z.; Cai, D.; He, X. CLRNet: Cross layer refinement network for lane detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 898–907. [CrossRef]
16. Jin, D.; Kim, D.; Kim, C.-S. Recursive video lane detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 8473–8482. [CrossRef]
17. Poudel, R.P.K.; Liwicki, S.; Cipolla, R. Fast-SCNN: Fast semantic segmentation network. In Proceedings of the 30th British Machine Vision Conference (BMVC 2019), Cardiff, UK, 9–12 September 2019; p. 289.
18. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. BiSeNet V2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 3051–3068. [CrossRef]
19. Fan, M.; Lai, S.; Huang, J.; Wei, X.; Chai, Z.; Luo, J.; Wei, X. Rethinking BiSeNet for real-time semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 9716–9725. [CrossRef]

20. Pan, H.; Hong, Y.; Sun, W.; Jia, Y. Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3448–3460. [CrossRef]
21. Xu, J.; Xiong, Z.; Bhattacharyya, S.P. PIDNet: A real-time semantic segmentation network inspired by PID controllers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 19529–19539. [CrossRef]
22. Gao, G.; Xu, G.; Li, J.; Yu, Y.; Lu, H.; Yang, J. FBSNet: A fast bilateral symmetrical network for real-time semantic segmentation. *IEEE Trans. Multimed.* **2023**, *25*, 3273–3283. [CrossRef]
23. Yang, G.; Wang, Y.; Shi, D.; Wang, Y. Golden cudgel network for real-time semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025; pp. 25367–25376. [CrossRef]
24. Peng, J.; Liu, Y.; Tang, S.; Hao, Y.; Chu, L.; Chen, G.; Wu, Z.; Chen, Z.; Yu, Z.; Du, Y.; et al. PP-LiteSeg: A superior real-time semantic segmentation model. *arXiv* **2022**, arXiv:2204.02681. [CrossRef]
25. Wang, J.; Gou, C.; Wu, Q.; Feng, H.; Han, J.; Ding, E.; Wang, J. RTFormer: Efficient design for real-time semantic segmentation with Transformer. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, and Online, 28 November–9 December 2022; Volume 35, pp. 7423–7436. [CrossRef]
26. Zhang, W.; Huang, Z.; Luo, G.; Chen, T.; Wang, X.; Liu, W.; Yu, G.; Shen, C. TopFormer: Token pyramid Transformer for mobile semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12083–12093. [CrossRef]
27. Wan, Q.; Huang, Z.; Lu, J.; Yu, G.; Zhang, L. SeaFormer: Squeeze-enhanced axial Transformer for mobile semantic segmentation. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, 1–5 May 2023.
28. Dong, B.; Wang, P.; Wang, F. Head-free lightweight semantic segmentation with linear Transformer. In Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI-23), Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 516–524. [CrossRef]
29. Cai, H.; Li, J.; Hu, M.; Gan, C.; Han, S. EfficientViT: Lightweight multi-scale attention for high-resolution dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 17302–17313. [CrossRef]
30. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Álvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with Transformers. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Virtual Conference, 6–14 December 2021; Volume 34, pp. 12077–12090.
31. Guo, M.-H.; Lu, C.-Z.; Hou, Q.; Liu, Z.; Cheng, M.-M.; Hu, S.-M. SegNeXt: Rethinking convolutional attention design for semantic segmentation. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, and Online, 28 November–9 December 2022; Volume 35, pp. 1140–1156. [CrossRef]
32. Cheng, B.; Girshick, R.; Dollár, P.; Berg, A.C.; Kirillov, A. Boundary IoU: Improving object-centric image segmentation evaluation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 15334–15342. [CrossRef]
33. Shit, S.; Paetzold, J.C.; Sekuboyina, A.; Ezhov, I.; Unger, A.; Zhylka, A.; Pluim, J.P.W.; Bauer, U.; Menze, B.H. clDice—A novel topology-preserving loss function for tubular structure segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 16560–16569. [CrossRef]
34. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 6027–6037. [CrossRef]
35. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722. [CrossRef]
36. Chen, J.; Kao, S.-H.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 12021–12031. [CrossRef]
37. Vasu, P.K.A.; Gabriel, J.; Zhu, J.; Tuzel, O.; Ranjan, A. MobileOne: An improved one millisecond mobile backbone. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7907–7917. [CrossRef]
38. Wang, A.; Chen, H.; Lin, Z.; Han, J.; Ding, G. RepViT: Revisiting mobile CNN from ViT perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 15909–15920. [CrossRef]
39. Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; Ren, J. EfficientFormer: Vision Transformers at MobileNet speed. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 12934–12949. [CrossRef]

40. Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Tulyakov, S.; Ren, J. Rethinking vision Transformers for MobileNet size and speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 16889–16900. [CrossRef]
41. Jocher, G.; Qiu, J.; Liu, M.; Lyu, S.; Akyon, F.C.; Kalfaoglu, M.E. Ultralytics YOLO26: Unified real-time end-to-end vision models. *arXiv* **2026**, arXiv:2606.03748. [CrossRef]
42. Liu, C.; Yao, J. The robust semantic SLAM system for texture-less underground parking lot. *J. Adv. Transp.* **2022**, *2022*, 9681455. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.