

Article

A Topology-Aware Lane Instance Segmentation Network Based on Dynamic Snake Convolution and High-Resolution Feature Pyramid

Sen Wang ¹, Baohua Guo ^{1,*}, Weifan Gu ¹, Xiaoyu Zhang ¹, Anthony Sigama ² and David Bassir ³

¹ School of Energy Science and Engineering, Henan Polytechnic University, Jiaozuo 454003, China

² Faculty of Science, Engineering and Agriculture, University of Venda, Thohoyandou 0950, South Africa

³ Smart Structural Health Monitoring and Control Laboratory, DGUT-CNAM, Dongguan University of Technology, Dongguan 523808, China; david.bassir@utbm.fr

* Correspondence: guobaohua@139.com

Abstract

Accurate lane detection is essential for autonomous driving, but the elongated geometry of lane markings and the information loss caused by perspective projection still make continuous lane perception challenging. Existing Convolutional Neural Network (CNN) methods are limited by fixed receptive fields and repeated downsampling, which can weaken narrow curvilinear structures and distant lane details. To address these issues, this paper proposes an enhanced single-stage lane instance segmentation framework based on YOLOv11. Dynamic Snake Convolution (DSConv) is integrated into the feature extraction network as an adaptive sampling operator for finite-width lane-mask representations, enabling local sampling trajectories to better follow narrow curved lane structures. A high-resolution P2 branch (Stride = 4) is further constructed in the Feature Pyramid Network (FPN) to compensate for fine-grained spatial details lost in deep feature maps. A reproducible point-to-mask training and mask-to-point evaluation pipeline is also established for the sparse TuSimple annotations. Experiments on the TuSimple benchmark, which mainly represents structured daytime highway scenes, show that the proposed method improves mask Precision, Recall, and mAP50 by 8.5, 9.0, and 6.8 percentage points, respectively, compared with YOLOv11-seg, while reaching 103.09 FPS under the reported testing configuration. Qualitative results on a newly constructed CULane subset further indicate improved lane continuity under reflections, illumination changes, night scenes, and complex urban backgrounds, while failure cases reveal remaining sensitivity to lane-like linear structures and weak lane visibility. The results suggest a practical accuracy–efficiency trade-off within the YOLO-style segmentation framework, although robustness in broader real-world scenarios still requires further verification.



Academic Editor: Aryya Gangopadhyay

Received: 20 June 2026

Revised: 16 July 2026

Accepted: 18 July 2026

Published: 21 July 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: lane detection; instance segmentation; dynamic snake convolution; YOLOv11; autonomous driving

1. Introduction

As a core component of autonomous driving perception systems, lane detection technology provides crucial environmental semantic information for Lane Keeping Assistance, Adaptive Cruise Control, and trajectory planning [1–3]. Despite significant progress in visual perception technology in recent years, lane markings—as a specific type of road sign—still present severe challenges for high-precision detection due to their inherent

geometric attributes: thin curvilinear strip-like structures, extreme aspect ratios, and drastic scale changes under perspective projection. How to maintain the structural continuity of lane markings while ensuring real-time performance remains an urgent problem to be solved.

Current lane detection algorithms can be broadly categorized into two classes: traditional digital image processing-based methods and deep learning-based methods. In traditional lane detection research, algorithms typically rely on hand-crafted feature extraction techniques and geometric modeling methods. Common feature extraction means include edge detection, color feature analysis, and texture feature extraction [4]. Regarding geometric modeling, linear models, curve models, or hybrid models are often adopted to fit lane lines.

Yoo et al. [5] proposed a gradient-enhancing conversion method based on linear discriminant analysis, achieving robust detection under multi-illumination conditions by dynamically optimizing grayscale image weights. This method still has limited adaptability in complex scenes such as extreme multi-source lighting or large curvature curves. Jiang and Li [6] employed an improved Hough transform combined with Kalman filter tracking to enhance the real-time performance and noise robustness of straight line detection, yet its adaptability remains constrained on roads with complex large curvatures. Wang et al. [7] proposed a visual detection algorithm based on a straight-curve model, achieving smooth transitions between straight and curved roads through piecewise modeling and geometric constraints. While suitable for roads with small to medium curvature, it lacks generalization capability for large curvature curves and unstructured roads. Although hand-crafted features align with human visual intuition, they rely on expert experience and have limited generalization capabilities. Traditional visual methods often fail to stably and correctly detect lanes when facing complex environments such as occlusion, missing lane markings, or uneven lighting.

In recent years, benefiting from the rapid development of deep learning technology and the significant improvement in the computing power of onboard computing platforms, deep learning-based lane detection methods have made breakthrough progress. Compared to traditional algorithms relying on manual features, deep learning models can automatically learn high-level semantic features of images, significantly enhancing detection robustness in complex scenes [8–11].

In early research, Line-CNN [12], inspired by Faster R-CNN [13], achieved effective lane localization by designing an LPU module similar to a Region Proposal Network and a ray-based anchor mechanism. Its high computational complexity limited inference speed and became a bottleneck for real-time applications. Addressing real-time requirements, Qin et al. [14] proposed the UltraFast algorithm, which adopted a lightweight backbone and a structure-aware module to drastically improve detection speed. This model has limited modeling capability for geometric structures, resulting in a significant drop in accuracy when dealing with large curvature curves or drastic lane changes [15,16].

Among segmentation-based methods, Neven et al. [17] proposed LaneNet, which pioneered the instance segmentation detection paradigm. By assigning an embedding vector to each pixel to distinguish different lane instances, it broke free from the limitation of a fixed number of lanes. The detection accuracy of this method is heavily influenced by embedding quality, and the MeanShift clustering algorithm in post-processing is computationally expensive, making it difficult to meet the high real-time requirements of autonomous driving systems. To address the shortcomings of LaneNet, Guo et al. [18] introduced a dual attention mechanism to enhance feature representation from both channel and spatial dimensions, effectively improving the model's perception of lane structures and anti-interference capability under complex backgrounds. Spatial Convolutional Neural

Network (SCNN) [19] enabled the network to fully model the long-range spatial dependencies of lanes through a row-wise and column-wise message passing mechanism. Although this structure fits the morphological characteristics of lanes, the dense message passing operations lead to a substantial increase in computational volume, resulting in low network efficiency and high hardware requirements [20–23].

The application of attention mechanisms has further propelled the development of the field. Hou et al. [24] proposed a self-attention distillation method, which significantly enhanced the model's ability to extract lane features by guiding the shallow network to learn rich structural information from deep features. This work promoted the widespread application of attention mechanisms in lane detection, providing new ideas for lightweight and high-performance model design. Ai et al. [25] combined a small object category attention mechanism with a multi-scale feature fusion strategy, optimizing for the characteristics of small and sparsely distributed lane targets in unstructured road scenes, thereby improving segmentation accuracy and adaptability in complex environments such as rural roads and curves. Building on the Transformer architecture, Han et al. [26] designed a lane-specific self-attention mechanism that effectively captures the overall shape features and local semantic information of lanes by modeling long-range dependencies between pixels. This method performs outstandingly in curve fitting and occlusion recovery tasks but entails relatively large computational overhead [27,28].

Despite the significant progress made by deep learning-based methods, they generally face problems such as large model parameters, high computational complexity, or heavy reliance on cumbersome post-processing. Consequently, it is difficult to satisfy the dual standards of high accuracy and high real-time performance simultaneously on resource-constrained onboard devices.

Targeting the contradiction between computational efficiency and geometric perception capability mentioned above, this paper proposes an improved YOLOv11 structure-aware instance segmentation network. The novelty of this work lies in the task-oriented adaptation and systematic evaluation of DSConv and a high-resolution P2 branch within a lightweight YOLOv11-seg lane instance segmentation framework. Beyond architectural adaptation, this work also establishes a reproducible point-to-mask supervision and mask-to-point evaluation pipeline for applying YOLO-style instance segmentation to TuSimple sparse lane annotations. The main contributions of this paper are summarized as follows.

1. **Task-oriented DSConv adaptation for lane masks.** We incorporate the existing DSConv operator into the YOLOv11-seg backbone and neck to reduce the geometric mismatch between fixed square convolution kernels and elongated lane-mask structures. This adaptation is designed to improve the continuity of curved lane responses while retaining the real-time characteristics of a single-stage detector.
2. **High-resolution spatial compensation.** We construct a high-resolution P2 feature pyramid branch (Stride = 4) to alleviate the loss of fine spatial details caused by repeated downsampling. This branch is intended to improve the segmentation of distant, narrow, and worn lane markings, which are easily weakened in P3–P5-only prediction heads.
3. **Reproducible point-to-mask supervision and evaluation pipeline.** We transform the sparse TuSimple lane points into finite-width instance masks using a fixed morphological dilation setting shared by all YOLO-based baselines, and restore predicted masks to point sequences for official TuSimple evaluation. This protocol enables the YOLO-based segmentation framework to be trained and evaluated consistently under both mask-level and point-level metrics, making the reported improvements more traceable and reproducible.

2. Proposed Model

2.1. YOLOv11 Baseline Model

As the latest iteration of current one-stage detectors, YOLOv11 has undergone deep reconstruction in feature extraction and multi-scale fusion, inheriting the efficiency of YOLOv8 [29–35]. This model follows the classic Backbone–Neck–Head design paradigm, aiming to achieve the optimal balance between real-time performance and accuracy.

In the Backbone level, YOLOv11 upgrades the original C2f module to the C3k2 structure. By introducing adjustable kernel sizes and residual connections [36], C3k2 enhances the gradient flow efficiency of shallow features. Additionally, the model embeds a Cross Stage Partial Spatial Attention (C2PSA) module after the Spatial Pyramid Pooling—Fast (SPPF) layer, utilizing attention mechanisms to strengthen global perception of key spatial features [37]. In the Neck, the architecture continues the Path Aggregation Network (PANet) design [38], achieving efficient fusion of multi-scale semantic features through bidirectional top-down and bottom-up paths. The Head adopts a decoupled design (Decoupled Head), separating classification and regression tasks, which, combined with the Anchor-Free mechanism [39], significantly reduces computational load. The model structure is shown in Figure 1.

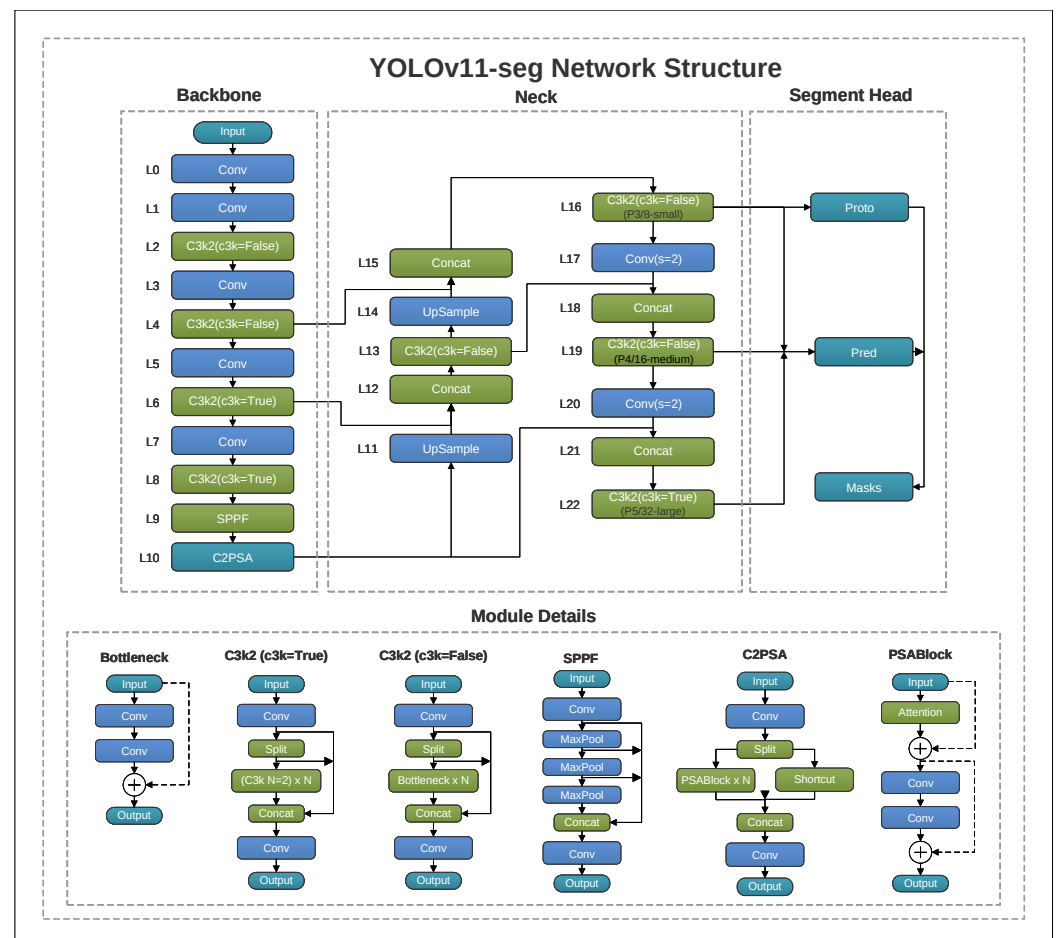


Figure 1. YOLOv11-seg baseline architecture.

2.2. Analysis of Limitations in YOLOv11

Although YOLOv11 demonstrates excellent real-time performance and generalization in general object detection, its Convolutional Neural Network (CNN) Architecture designed for blocky objects (e.g., vehicles, pedestrians) exhibits significant architecture–feature misalignment when processing lanes with slender curvilinear structural features [15]. We

concretize this misalignment into the following three core challenges, which serve as the direct motivation for our improvement strategy.

1. **Geometric Misalignment of Standard Convolution.** YOLOv11's feature extraction relies on fixed 3×3 standard square kernels [40]. This fixed grid sampling mechanism assumes targets have compact spatial distributions. Lanes are typical non-rigid long strip manifolds, with curvatures changing drastically in image space. When a square receptive field is applied to large curvature curves, the sampling grid inevitably covers a large amount of irrelevant background area, leading to valid lane texture features being diluted in feature maps. This geometric non-adaptability makes it difficult for the network to precisely decouple high-frequency edge details of lanes from background noise under complex backgrounds. To this end, this paper introduces DSConv [40] to address this geometric misalignment through adaptive deformation sampling.
2. **Spatial Information Dissipation in Deep Layers.** Accurate lane segmentation relies heavily on pixel-level spatial localization precision. Native YOLOv11 employs strided convolution to construct a deep network with a maximum downsampling rate of $32\times$, causing feature map resolution to drop drastically. For distant vanishing points or lanes with extremely narrow visual widths [41], their critical geometric features are easily smoothed or completely lost (Vanishing) during multiple pooling and strided convolution processes [20]. This high-semantics, low-resolution feature level distribution directly leads to missed detections and discontinuities for subtle targets, making it difficult to meet the continuity and localization requirements of lane-level navigation. Targeting this bottleneck, this paper constructs a high-resolution P2 branch (Stride = 4) to compensate for spatial information dissipation by explicitly introducing shallow details. Earlier lane-detection studies likewise emphasized vanishing-point guidance and lane-sensitive architecture design for preserving global lane geometry under perspective distortion [42,43].
3. **Conflict between Local Texture Perception and Long-range Structural Continuity.** In complex scenarios such as TuSimple, lane lines often suffer from interference due to illumination changes, wear, blur, and vehicle occlusion. Semantic segmentation essentially belongs to pixel-level classification; standard convolution mainly focuses on local texture features and lacks the ability to explicitly model long-range context dependency [21]. Due to the insufficient perception of the overall lane direction, the model often struggles to infer disconnected segments when facing occluded or blurred areas, resulting in discontinuous lane-mask responses. This paper uses the iterative sampling mechanism of DSConv to extend local feature aggregation along curvilinear structures, thereby improving the structural continuity of lane-mask responses.

In summary, to break through the performance bottleneck in slender structure segmentation while maintaining the high inference speed advantage of the YOLO series, this paper proposes a real-time lane segmentation solution that balances computational efficiency with geometric perception capability. By introducing adaptive sampling and a high-resolution feature pyramid, the proposed framework aims to improve the extraction of narrow curvilinear lane-mask features and the preservation of distant fine-grained spatial details.

2.3. Model Overall Architecture

As shown in Figure 2, based on the YOLOv11n-seg baseline, this paper constructs an instance segmentation framework that balances lightweight design with structure-aware perception capability. Targeting the geometric characteristics of lanes, we modify the feature extraction and multi-scale fusion stages.

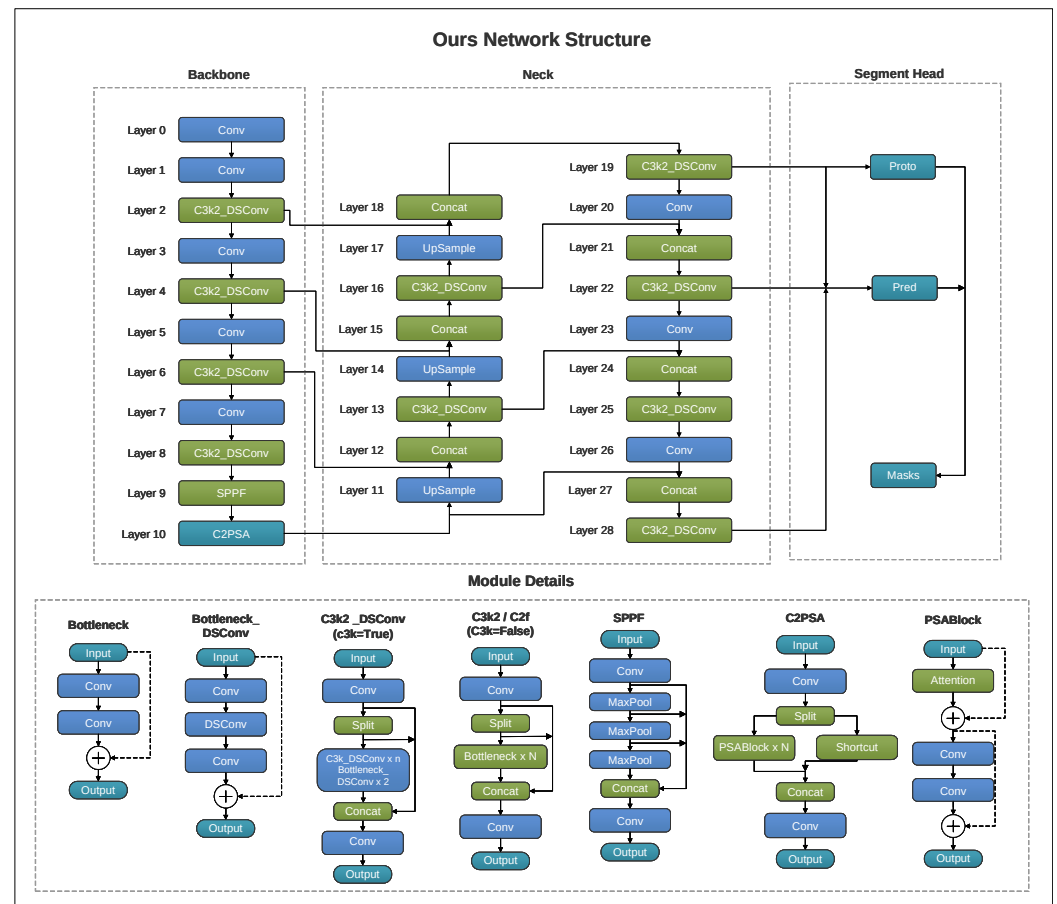


Figure 2. Overall architecture of the proposed lane instance segmentation network.

In the Backbone and Neck, we replace the standard C3k2 modules with the proposed C3k2_DSConv modules. This design integrates DSConv, aiming to introduce adaptive sampling for slender curved features while maintaining deep semantic reuse.

Addressing the scale differences caused by perspective effects, we construct a four-scale feature pyramid (P2–P5). Specifically, we introduce a high-resolution P2 branch (Stride = 4) into the bidirectional fusion path (Feature Pyramid Network (FPN) and PANet). This enables the prediction head to operate on a finer spatial grid, enhancing detection capabilities for distant texture-less targets. The implementation keeps the fusion path focused on the P2–P5 multi-scale prediction structure.

In the segmentation head design, this paper adopts an asymmetric input strategy: the prototype branch utilizes P3 features to balance memory consumption, while the detection branch retains complete P2 input to ensure pixel-level regression accuracy for lane boundaries.

2.4. Structure-Aware Recognition Based on Dynamic Snake Convolution

Addressing the geometric misalignment problem caused by standard convolutions being limited to fixed rectangular sampling grids as described in Section 2.2, this paper integrates DSConv [40] into the backbone and neck networks. Unlike traditional strategies that attempt to enlarge the receptive field by stacking convolution layers, DSConv allows the convolution kernel to adaptively adjust sampling positions based on the gradient changes of input features, thereby improving the coverage of finite-width curvilinear lane-mask structures without increasing computational depth.

2.4.1. Construction of the C3k2_DSConv Module

To incorporate morphology-adaptive feature extraction into the network, this paper designed the C3k2_DSConv module, as shown in Figure 3. This module follows the Cross Stage Partial (CSP) design. Input features are first channel-compressed via a 1×1 convolution and then split into a main branch and a residual branch. The main branch serially stacks N Bottleneck_DSConv blocks for non-linear curvilinear feature extraction, while the residual branch is used to retain original information and promote stable gradient propagation.

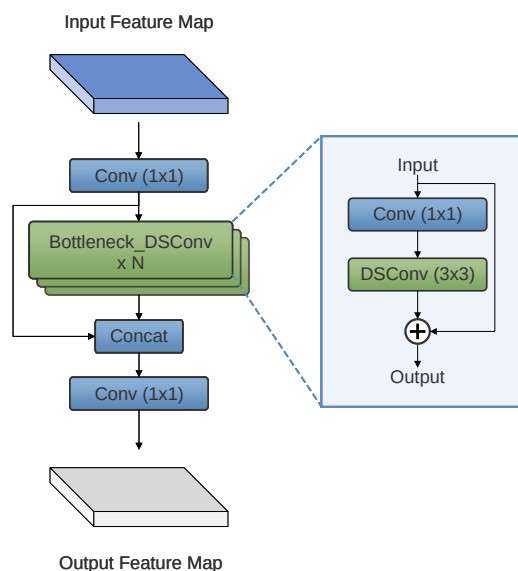


Figure 3. Structure of the C3k2-DSConv module.

Inside the Bottleneck_DSConv, as shown in the magnified view in Figure 3, this paper replaces the standard 3×3 convolution in the residual bottleneck with DSConv. This allows the sampling grid of the convolution kernel to adaptively adjust according to the slender and curved morphology of the lanes. The outputs of the two branches are concatenated and then fused and remapped via a 1×1 convolution to obtain the module output feature map.

2.4.2. Iterative Adaptive Sampling Mechanism

The core innovation of DSConv lies in breaking the free diffusion mode of sampling points in standard Deformable Convolution [44]. Instead, it proposes a coordinate generation strategy based on Iterative Accumulation to maintain the continuity of elongated structures.

As shown in Figure 4a, the rigid square grid of standard convolution struggles to fit curved lane structures, leading to sparse effective features; in contrast, DSConv can dynamically adjust sampling points according to the target morphology, improving the coverage of narrow strip-like lane masks. This is intuitively verified in feature responses: as shown in Figure 5, the response of standard convolution in high-curvature regions often appears discontinuous, manifesting as local fractures in continuous structures, whereas the adaptive sampling of DSConv can more stably aggregate high response values along the lane direction, thereby reducing local mask fragmentation in the illustrated example.

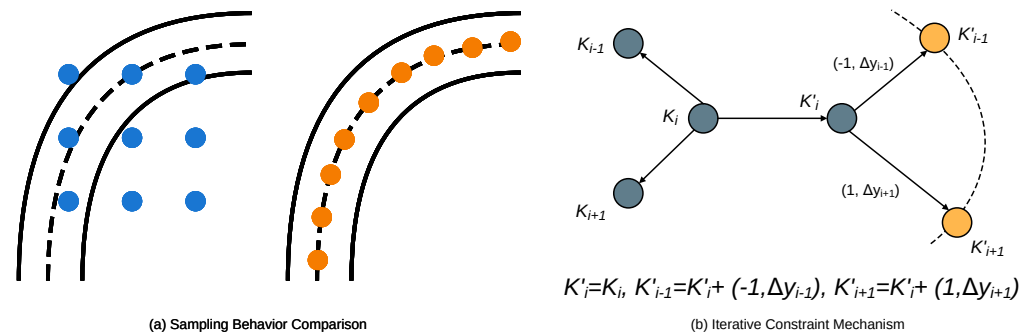


Figure 4. Comparison of geometric adaptability and iterative coordinate generation in DSConv.

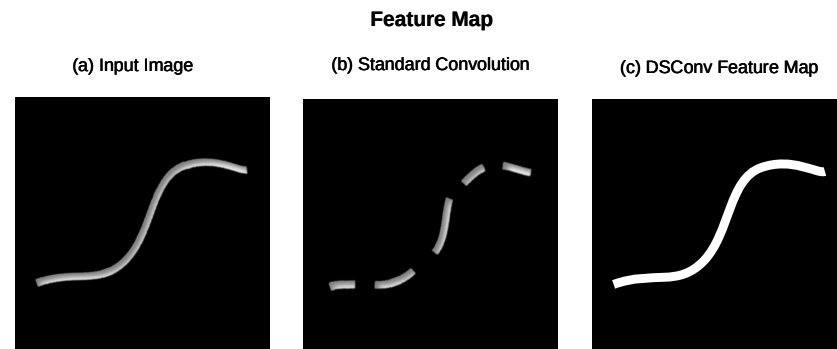


Figure 5. Visual comparison of feature responses generated by standard convolution and DSConv on a curved lane-like structure.

The specific coordinate generation mechanism is shown in Figure 4b. DSConv decouples the deformation of the convolution kernel into two independent processes along the orthogonal X-axis and Y-axis. Taking the X-axis direction as an example, for a convolution kernel of size K , the selection of its grid positions is a cumulative process. Assuming the center coordinate is K_i , the coordinate of the grid point at distance c from the center is not fixed but is based on the previous position plus a learnable offset.

The coordinate calculation formula is shown in Equation (1).

$$K_{i \pm c} = \begin{cases} (x_{i+c}, y_{i+c}) = (x_i + c, y_i + \sum_{i=i-c}^{i+c} \Delta y), \\ (x_{i-c}, y_{i-c}) = (x_i - c, y_i + \sum_{i=i-c}^i \Delta y), \end{cases} \quad (1)$$

Similarly, deformation in the Y-axis direction follows a similar iterative logic. This Chain-like Dependency mechanism ensures the continuity of the convolution kernel's receptive field, enabling it to adaptively fit the curved geometric features of lanes and effectively suppressing uncontrolled drift of the receptive field towards background regions.

2.4.3. Dual-Path Parallel Implementation and Feature Resampling

In specific implementation, the DSConv module adopts a dual-path parallel modeling architecture (as shown in Figure 6). Input feature maps are fed into the DSConv_x and DSConv_y branches respectively to independently capture morphological features in orthogonal directions.

Within each branch, an Offset Conv layer first regresses the incremental offset field for the corresponding axis. Subsequently, a Grid Generator calculates the deformed sampling coordinates based on the iterative strategy of Equation (1). Given that the calculated coordinates are usually floating-point numbers, to ensure differentiability of operations

and enable end-to-end training, this paper adopts Bilinear Interpolation to map feature values from continuous space to the discrete grid.

$$K' = \sum_q G(q, p) \cdot X(q), \quad (2)$$

where $X(q)$ is the feature value at integer position q in the feature map, $G(\cdot)$ is the interpolation kernel, and p is the deformed coordinate. The output features of the two orthogonal branches are fused through element-wise summation and processed via Batch Normalization (BN) and activation functions. This design supports the model's adaptation to the variable morphology of lanes and helps preserve the continuity of slender structures.

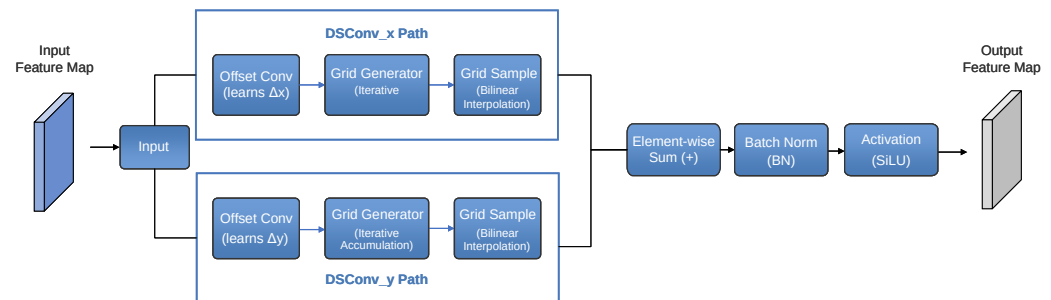


Figure 6. Internal implementation flowchart of the DSConv module.

2.4.4. Applicability of DSConv to Narrow Lane Structures

Lane markings in camera images are closer to one-dimensional centerline structures with finite annotation width than to wide tubular anatomical objects. Therefore, this study does not assume that lanes and vessels are morphologically identical. After the sparse TuSimple lane points are converted into dilated binary masks, each lane instance becomes a narrow strip with a finite width, which provides a practical supervision form for DSConv-based adaptive sampling.

Under this mask-based setting, DSConv is used to guide the sampling trajectory along the local lane direction on intermediate feature maps. Its iterative accumulation mechanism constrains neighboring sampling locations to evolve in a chain-like manner, reducing unrestricted offset drift toward background regions. This design is expected to improve local continuity for curved lane masks, and its actual effect is evaluated through ablation experiments and qualitative comparisons.

2.5. Feature Reconstruction with High-Resolution P2 Branch

In the YOLOv11 baseline architecture, the detection and segmentation heads mainly rely on P3, P4, and P5 feature levels, with corresponding downsampling strides of 8, 16, and 32, respectively. Although this multi-scale design effectively balances semantic information and computational efficiency in general object detection tasks, it reveals limitations when processing lane segmentation tasks. As typical slender curvilinear strip-like structures, under the influence of image perspective effects, distant lanes often appear as extremely narrow lines or even discrete pixels, possessing extremely low spatial occupancy. As the downsampling rate increases, these critical subtle geometric features are easily smoothed or lost during the feature abstraction process of deep networks, leading to fractures, blurring, or even missed detections in distant lane segmentation results, seriously affecting the model's reconstruction of continuous lane structures.

To break this resolution bottleneck, this paper constructs a dedicated high-resolution P2 branch (downsampling stride is only 4) in the FPN [45]. Shallow feature maps rich in texture and edge information are introduced into the multi-scale fusion path, forming a

P2–P3–P4–P5 four-scale prediction system. The introduction of the P2 branch explicitly compensates for the lack of spatial geometric information in the deep network, allowing the model to perform regression on subtle targets directly on a higher-resolution feature map. This not only enhances the model's response intensity to distant extremely narrow lanes but also improves the boundary localization accuracy for severely worn or discontinuous lanes, realizing refined reconstruction of lane structural details [46–48].

The P2 branch provides high-resolution spatial details, while the P3–P5 branches provide progressively stronger semantic information; after concatenation, the multi-scale features are further refined by subsequent C3k2_DSConv blocks so that high-resolution spatial compensation and local morphology-adaptive feature extraction can be jointly learned during training. DSConv mainly addresses the geometric mismatch of fixed convolutional sampling and the local continuity of curved lane masks, whereas the P2 branch compensates for the loss of fine-grained spatial details caused by repeated downsampling. Their combination is designed as a practical trade-off between local morphology adaptation and high-resolution spatial compensation. The high-resolution P2 branch increases the number of feature locations involved in the subsequent fusion path, and the computational cost may increase if offset-related operations are not constrained. To reduce this risk, the P2 branch is used as an additional prediction and fusion level rather than replacing all deeper feature levels. DSConv is embedded within the C3k2 structure instead of being applied as an independent dense offset module to all raw high-resolution feature maps.

3. Experimental Setup and Evaluation Metrics

3.1. Experimental Environment Configuration

The experimental environment configuration is as follows: Windows 11 operating system, CPU: 12th Gen Intel® Core™ i5-12600KF (3.70 GHz), GPU: NVIDIA GeForce RTX 4060 Ti (16 GB), 32 GB RAM. The programming language is Python 3.12.7, the deep learning framework is PyTorch 2.5.1, and CUDA version is 12.7.33.

3.2. Training Configuration and Reproducibility Settings

To improve reproducibility, all YOLO-series models were trained under the same input resolution and training schedule. The input images were resized to 640×640 . The batch size was set to 32, and four data-loading workers were used. All models were trained for 100 epochs using the YOLO segmentation loss, which consists of bounding-box regression loss, classification loss, distribution focal loss, and mask loss. The optimizer was initialized according to the YOLO training configuration, with an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005, followed by cosine learning-rate decay and warm-up in the early training stage. The same data augmentation strategy, including random scaling, flipping, and color perturbation, was applied to all YOLO-based models.

Model selection was performed using the best checkpoint on the validation set according to mask mAP₅₀, and the selected checkpoint was then evaluated on the test set. During inference, the confidence threshold, non-maximum suppression setting, input resolution, and mask-to-lane restoration procedure were kept identical for YOLOv11-seg and the proposed model to ensure that the observed improvement mainly originated from the architectural modifications.

3.3. Dataset

To ensure the scientific validity and comparability of the research results, this paper selects the authoritative and widely cited TuSimple dataset in the computer vision field for experimental verification. Originating from the 2017 TuSimple Lane Detection Challenge, this dataset serves as one of the authoritative benchmarks in academia for evaluating the

performance of lane segmentation algorithms. The experiment strictly follows the official partition protocol, dividing the 6408 sampled images into a training set of 3268 images, a validation set of 358 images, and a test set of 2782 images, with detailed information presented in Table 1. This standardized data partitioning is widely adopted by mainstream algorithms such as SCNN, LaneATT, and Ultra-Fast Lane Detection (UFLD), thereby providing a fair comparison benchmark for the improved YOLOv11 model proposed in this paper.

Table 1. Detailed information of the TuSimple dataset.

Dataset	Training Set	Validation Set	Test Set	Lanes	Environment	Resolution
TuSimple	3268	358	2782	≤ 5	Highway	1280 × 720

The original annotations of the TuSimple dataset are sparse discrete points, while the YOLOv11 segmentation head requires polygon masks as supervision signals. To resolve this modality mismatch, this paper adopts a label generation strategy based on morphological dilation. Specifically, we first reconstruct the lane centerline based on original coordinate points [49,50] and use it as an axis to perform morphological dilation with a fixed width (set to 6 pixels in this paper) to generate binary segmentation masks with physical meaning. Because the generated mask width directly affects mask-level IoU and mAP₅₀, the same mask-generation width is used for all compared YOLO models to avoid preprocessing-induced bias. On this basis, we map the masks to polygon vertex sequences through a contour extraction algorithm and introduce the Douglas–Peucker algorithm for adaptive downsampling of vertices, eliminating redundant noise while retaining geometric precision. The finalized normalized labels not only fit the supervision requirements of instance segmentation but also alleviate the foreground–background class imbalance problem in slender object detection. During the inference evaluation stage, the masks output by the model are skeletonized and restored to point sequences, strictly following the official TuSimple evaluation standards to ensure fairness of comparison.

3.4. Evaluation Metrics

To ensure the objectivity and comparability of our experimental results, we strictly adhere to the widely recognized evaluation protocol of the authoritative TuSimple benchmark. Addressing the characteristics of discrete point sequence annotations, the core evaluation relies on point-level matching: a predicted lane is considered a True Positive (TP) only if over 85% of its valid sampled points exhibit a horizontal deviation of less than 20 pixels relative to the ground truth; otherwise, it is classified as a False Positive (FP) or False Negative (FN). Building upon this standard, we adopt a dual-track evaluation metric system to assess the model’s detection accuracy, segmentation quality, and computational efficiency. The first track employs the official TuSimple benchmark metrics to facilitate comparison with domain-specific lane detection algorithms, while the second track utilizes standard instance segmentation metrics to analyze the architectural enhancements within the YOLO series. F1-score is additionally reported as a unified mask-level summary metric that balances Precision and Recall.

1. **Accuracy.** As the core metric for the TuSimple benchmark, Accuracy reflects the point-level prediction correctness. It is calculated as the ratio of correctly predicted points to the total number of ground-truth points. The calculation formula is shown in Equation (3).

$$\text{Accuracy} = \frac{\sum C_i}{\sum S_i}, \quad (3)$$

where C_i denotes the number of correctly predicted lane points in the i -th test image, and S_i represents the total number of ground-truth points in that image.

2. **False Positive (FP) Rate.** In the TuSimple evaluation, the FP rate measures the proportion of wrongly predicted lanes (e.g., predicted lanes where less than 85% of the sampled points match the ground truth) among all predicted lanes. The calculation formula is shown in Equation (4).

$$FP = \frac{F_{pred}}{N_{pred}}, \quad (4)$$

where F_{pred} represents the number of false positive lanes, and N_{pred} is the total number of lanes predicted by the model.

3. **False Negative (FN) Rate.** The FN rate evaluates the proportion of missed ground-truth lanes. The calculation formula is shown in Equation (5).

$$FN = \frac{M_{pred}}{N_{gt}}, \quad (5)$$

where M_{pred} denotes the number of missed ground-truth lanes, and N_{gt} is the total number of actual ground-truth lanes.

4. **Precision.** To evaluate the general mask-level segmentation capability, Precision measures the reliability of the detection results, representing the proportion of true positive predictions among all positive predictions. The calculation formula is shown in Equation (6).

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (6)$$

where TP represents True Positives, and FP represents False Positives at the instance mask level.

5. **Recall.** Recall evaluates the model's coverage capacity for real objects, representing the proportion of successfully detected instances among all ground-truth instances. The calculation formula is shown in Equation (7).

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (7)$$

where FN denotes False Negatives at the instance mask level.

6. **Mean Average Precision (mAP_{50}).** As a comprehensive metric for evaluating instance segmentation models, mAP_{50} computes the average precision across all categories at an Intersection over Union (IoU) threshold of 0.5. The calculation formula is shown in Equation (8).

$$mAP_{50} = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} \int_0^1 P(R) dR, \quad (8)$$

where N_{cls} is the number of classes (for lane detection, $N_{cls} = 1$), and $P(R)$ represents the Precision–Recall curve function.

7. **F1-score.** To provide a single mask-level indicator that balances missed and false detections, F1-score is added to the YOLO-series comparison. It is calculated from Precision and Recall as shown in Equation (9).

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (9)$$

Given that lane detection algorithms often need to be deployed on resource-constrained autonomous driving onboard embedded platforms, the model's resource occupation and computational efficiency are key dimensions for measuring its engineering practicality. In addition to conventional detection accuracy, this paper adopts Frames Per Second (FPS) to evaluate inference efficiency in a unified hardware environment. FPS is reported in the comparison tables as a reference for inference efficiency under the stated hardware and testing conditions. The deployment-oriented speed is calculated from the average single-image end-to-end latency, including preprocessing, network inference, and post-processing. To further quantify the lightweight degree of the model, this paper introduces Parameters and Floating-Point Operations (FLOPs) as evaluation metrics. Parameters represent the sum of the model's learnable weights, directly reflecting the model's spatial complexity and storage requirements. FLOPs quantify the computational scale of a single forward inference, objectively reflecting the model's computational complexity and load requirements on hardware computing power, thereby measuring the feasibility of the algorithm in engineering implementation.

4. Experimental Results and Analysis

4.1. Comparative Experiments

To evaluate the effectiveness of the improved algorithm proposed in this paper, multi-dimensional comparative experiments were designed. For comparison with task-specific lane detection algorithms (e.g., SCNN, LaneATT), this paper follows the official standard evaluation protocol of the TuSimple dataset in the lane detection domain, adopting Accuracy, FP rate, and FN rate as quantitative metrics. When validating the architectural evolution within the YOLO series and the effectiveness of the improved modules proposed herein, Precision, Recall, F1-score, and mAP₅₀ are utilized for evaluation.

4.1.1. Comparison with Mainstream Lane Detection Algorithms

To verify the effectiveness of the proposed algorithm in lane segmentation tasks, representative lane detection methods with different prediction paradigms are selected for comparison, including the segmentation-based SCNN, the anchor-based LaneATT, the cross-layer refinement method CLRNet, and the conditional-convolution-based CondLaneNet. The original YOLOv11-seg model is also included to clarify the gains introduced by the proposed modules within the same YOLO-style segmentation pipeline. The evaluation metrics include Accuracy, FP, FN, FPS, parameter count, and FLOPs to describe point-level detection accuracy and computational characteristics.

Different methods use different input resolutions, prediction forms, and post-processing strategies. Therefore, the table reports the main testing conditions explicitly, including input size, post-processing requirement, FPS, parameter count, and FLOPs.

Table 2 reports the comparison between the proposed method and representative lane detection approaches under the official TuSimple evaluation protocol. Since different methods adopt different input resolutions, prediction representations, and post-processing strategies, this table is used to contextualize the proposed method among existing lane detection approaches rather than to claim a strictly controlled comparison under identical training settings.

Within the YOLO-style instance segmentation framework, the proposed method achieves clear improvements over the YOLOv11-seg baseline. The Accuracy increases from 91.36% to 94.58%, corresponding to a gain of 3.22 percentage points. The FP rate decreases from 18.40% to 11.80%, and the FN rate decreases from 22.29% to 8.46%. These results indicate that the combination of DSConv and the P2 branch improves the prediction quality of YOLOv11-seg under the point-level TuSimple evaluation protocol, with reductions in

both falsely predicted lanes and missed lane instances. Meanwhile, the proposed model contains 3.65 M parameters, which is substantially smaller than SCNN, LaneATT, CLRNNet, and CondLaneNet, indicating that the model scale remains relatively compact.

Table 2. Performance comparison of representative lane detection methods under explicitly stated testing conditions.

Method	Backbone	Input Size	Acc (%)	FP (%)	FN (%)	Post-Processing	FPS	Params (M)	FLOPs (G)
SCNN	VGG16	512×288	94.16	7.35	8.25	Clustering/spline fitting	99.31	20.53	69.97
LaneATT	ResNet34	640×360	95.63	3.53	2.92	Anchor filtering/NMS	34.79	22.13	18.01
YOLOv11-seg	YOLO11n	640×640	91.36	18.4	22.29	NMS + mask-to-point restoration	471.73	2.83	5.18
CLRNNet	ResNet18	800×320	96.84	2.28	1.92	NMS	195.84	11.77	12.03
CondLaneNet	ResNet18	800×320	95.48	2.18	3.80	Seed NMS + lane point restoration	226.54	11.93	11.93
Ours	DSCov+P2	640×640	94.58	11.80	8.46	NMS + mask-to-point restoration	103.09	3.65	19.5

Compared with task-specific lane detection models, the performance boundary of the proposed method is also clear. CLRNNet achieves the highest Accuracy and the lowest FN among the compared methods, reaching 96.84% and 1.92%, respectively. CondLaneNet obtains the lowest FP rate of 2.18%, and LaneATT also outperforms the proposed method in terms of FP and FN. These results show that lane-specific architectures based on point-sequence prediction or specialized lane representations still have advantages under the official TuSimple point-level metrics. The main value of the proposed method lies in improving the adaptability of a general YOLO-style instance segmentation framework for lane-mask prediction, rather than replacing all task-specific lane detection architectures. Compared with SCNN, the proposed method achieves a slightly higher Accuracy with substantially fewer parameters and lower FLOPs. Compared with the original YOLOv11-seg baseline, the proposed method obtains better official point-level detection performance at the cost of additional computation. Under the testing configuration reported in the table, the proposed method reaches 103.09 FPS, indicating a practical trade-off between accuracy improvement and computational complexity.

4.1.2. Comparison with YOLO Series Baseline Models

To further validate the improvement effects on the YOLOv11 architecture, under the identical experimental environment, the proposed method was vertically compared with the segmentation versions of YOLOv5, YOLOv6, YOLOv8, YOLOv9, YOLOv10, and YOLOv11. Table 3 presents the detailed data of each model regarding Precision, Recall, F1-score, mAP₅₀, and resource consumption.

Table 3. Performance comparison between the proposed method and YOLO baseline models.

Method	Precision (%)	Recall (%)	F1 (%)	mAP ₅₀ (%)	FPS	Params (M)	FLOPs (G)
YOLOv5-seg	51.8	50.9	51.3	33.6	488.47	2.76	11.0
YOLOv6-seg	55.6	54.1	54.8	40.9	530.26	4.41	15.2
YOLOv8-seg	53.3	50.8	52.0	32.7	478.18	3.26	12.0
YOLOv9t-seg	55.7	56.1	55.9	40.9	222.27	3.13	54.1
YOLOv10n-seg	55.1	54.6	54.8	39.1	467.49	2.85	11.7
YOLOv11-seg	53.7	53.7	53.7	39.5	471.73	2.83	10.2
Ours	62.2	62.7	62.4	46.3	103.09	3.65	19.5

Table 3 presents the comparison between the proposed method and YOLO-series segmentation models under the same mask-level evaluation setting. All models are evaluated with the same input resolution, label-generation process, and evaluation pipeline. Therefore, this table provides a more direct comparison of different YOLO-style segmentation architectures on the lane-mask segmentation task.

As shown in Table 3, the proposed method achieves the highest Precision, Recall, F1-score, and mAP₅₀ among the compared models. Compared with the YOLOv11-seg baseline, the Precision increases from 53.7% to 62.2%, the Recall increases from 53.7% to 62.7%, the F1-score increases from 53.7% to 62.4%, and the mAP₅₀ increases from 39.5% to 46.3%. The corresponding gains are 8.5, 9.0, 8.7, and 6.8 percentage points, respectively. The simultaneous improvement in Precision and Recall indicates that the proposed structure does not improve recall simply by increasing the number of predictions, but improves the reliability of positive mask predictions while reducing missed lane instances. The improvement in mAP₅₀ further suggests that, under the same 6-pixel mask-label setting, the predicted masks have better overlap quality with the generated lane-instance annotations.

Compared with other YOLO segmentation baselines, the proposed method also maintains a consistent advantage. YOLOv6-seg and YOLOv9t-seg obtain the highest baseline mAP₅₀ of 40.9%, whereas the proposed method reaches 46.3%, corresponding to a gain of 5.4 percentage points over the strongest baseline result. YOLOv9t-seg achieves the highest baseline F1-score of 55.9%, while the proposed method reaches 62.4%, with an improvement of 6.5 percentage points. These results indicate that the combination of DSConv and the P2 branch provides a consistent performance gain within the YOLO-style segmentation model family, rather than being effective only for the YOLOv11-seg baseline.

In terms of model complexity, the proposed method introduces additional computational cost compared with YOLOv11-seg. The parameter count increases from 2.83 M to 3.65 M, and the FLOPs increase from 10.2 G to 19.5 G. This increase mainly comes from the additional feature-fusion and prediction path introduced by the high-resolution P2 branch. Compared with the computationally heavier YOLOv9t-seg, the proposed method achieves higher Precision, Recall, F1-score, and mAP₅₀ while reducing FLOPs from 54.1 G to 19.5 G. Under the testing configuration reported in the table, the proposed method reaches 103.09 FPS, which is lower than the lightweight YOLO baselines but still maintains a relatively high inference efficiency. Overall, the proposed method obtains better lane-mask segmentation accuracy with an interpretable increase in computational complexity, making it a strengthened baseline for YOLO-style lane instance segmentation.

4.2. Ablation Studies

To quantitatively evaluate the effects of Dynamic Snake Convolution and the high-resolution P2 branch on model performance, ablation experiments with single-module and combined-module settings are conducted based on YOLOv11-seg, as shown in Table 4. Group A denotes the original YOLOv11-seg baseline, Group B introduces only the P2 branch, Group C introduces only C3k2_DSConv, and Group D represents the complete model with both the P2 branch and C3k2_DSConv.

Table 4. Ablation study of the P2 branch and DSConv.

Exp. Group	Baseline (YOLOv11)	P2 Branch	C3k2_DSConv	Mask P (%)	Mask R (%)	Mask mAP ₅₀ (%)	FLOPs (G)
A	✓			53.7	53.7	39.5	10.2
B	✓	✓		55.6	53.8	41.4	24.0
C	✓		✓	53.0	50.7	38.8	11.8
D (Ours)	✓	✓	✓	62.2	62.7	46.3	19.5

Note: ✓ indicates that the corresponding component is included in the experimental configuration; a blank cell indicates that the component is not included.

As shown in Table 4, introducing the P2 branch alone improves Mask Precision from 53.7% to 55.6%, Mask Recall from 53.7% to 53.8%, and Mask mAP₅₀ from 39.5% to 41.4%. This result indicates that high-resolution shallow features provide finer spatial information for lane-mask segmentation and improve the reliability of positive mask predictions and

the overall overlap quality to a certain extent. The performance gain brought by the P2 branch is still moderate, while the FLOPs increase from 10.2 G to 24.0 G, indicating that the high-resolution feature path introduces a notable computational burden.

When C3k2_DSConv is introduced alone, the model obtains 53.0% Mask Precision, 50.7% Mask Recall, and 38.8% Mask mAP₅₀, which are lower than those of the baseline model. This phenomenon suggests that adaptive sampling alone cannot stably improve the segmentation of extremely narrow lane masks when high-resolution spatial details are insufficient. For distant and narrow lane markings in TuSimple, spatial information in deep feature maps has already been weakened by repeated downsampling. Under this condition, the morphology-adaptive sampling of DSConv may not be fully utilized and may even affect mask prediction stability due to insufficient foreground responses.

The complete model achieves the best results after introducing both the P2 branch and C3k2_DSConv, with Mask Precision, Mask Recall, and Mask mAP₅₀ reaching 62.2%, 62.7%, and 46.3%, respectively. Compared with the YOLOv11-seg baseline, these three metrics are improved by 8.5, 9.0, and 6.8 percentage points, respectively. Compared with the P2-only setting, the complete model further improves Precision, Recall, and mAP₅₀. Compared with the C3k2_DSConv-only setting, the complete model recovers the performance and clearly surpasses the baseline level. These results indicate that the high-resolution spatial details provided by the P2 branch offer a more effective feature basis for the morphology-adaptive sampling of DSConv, while DSConv further enhances the local continuity representation of elongated curved structures in multi-scale features.

In terms of computational complexity, the complete model has 19.5 G FLOPs, which is higher than the 10.2 G of the baseline model but lower than the 24.0 G of the P2-only setting. This result shows that the proposed structure does not simply introduce a linear computational increase by stacking two modules, but forms a more reasonable structural combination between P2-based high-resolution spatial compensation and DSConv-based local morphology modeling. Considering both accuracy and computational cost, Group D achieves the best mask-level segmentation performance among the four settings, indicating that the P2 branch and C3k2_DSConv are complementary and jointly contribute to the performance improvement of the proposed model.

4.3. Visual Analysis

4.3.1. TuSimple Dataset

To analyze the segmentation behavior on the TuSimple dataset, representative samples involving lane splits, curved lanes, distant thin lane markings, and vehicle occlusion are selected for qualitative comparison among YOLOv8-seg, YOLOv9-seg, YOLOv10-seg, YOLOv11-seg, and the proposed model. Figure 7 presents the predictions by scene and model, focusing on lane continuity, curvature adaptability, and far-end detail preservation.

Figure 7a shows a highway exit scene with lane splitting and a large-curvature lane. The right lane is close to the image boundary and exhibits an evident curved structure, which makes its spatial morphology relatively complex. The baseline models can identify part of the main lanes, but they show missed responses or short fragmented predictions on the right curved lane. By contrast, the proposed model generates a more continuous mask in the curved region and maintains relatively stable responses for the middle straight lane and the left distant lane, indicating that elongated-structure feature modeling helps improve the geometric adaptability to large-curvature lanes.

Figure 7b corresponds to a multi-lane highway scene where preceding vehicles and perspective compression jointly increase the difficulty of lane recognition. As lane markings extend toward the far end, their pixel width gradually decreases, and local occlusion appears near vehicles. The YOLO-series baselines can segment the main lane regions, but their

responses around distant thin lanes and occluded areas tend to be unstable, locally broken, or prematurely truncated. The proposed model maintains clearer responses on the left, middle, and right lanes, with a more complete far-end extension. This result is consistent with the intended role of the high-resolution P2 branch, which preserves shallow fine-grained spatial information and alleviates feature weakening caused by repeated downsampling.

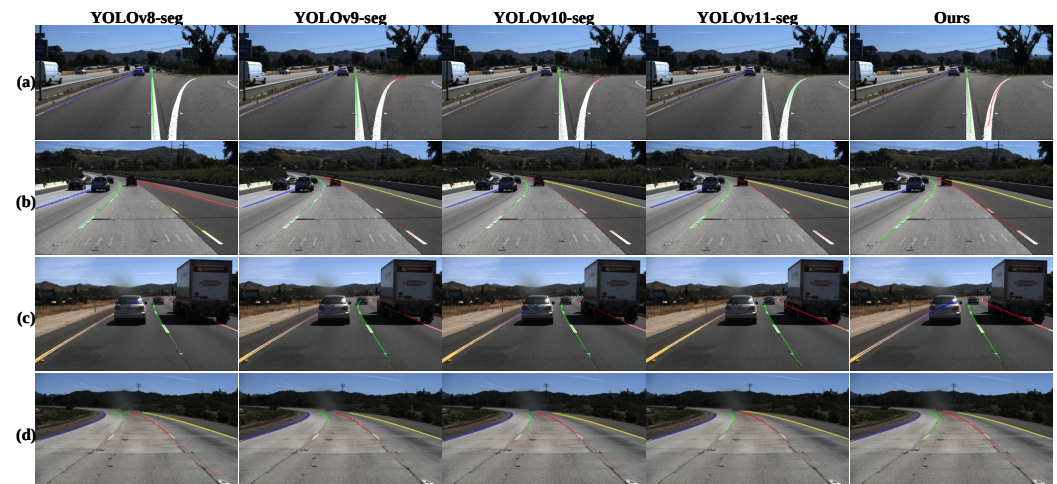


Figure 7. Qualitative comparison on representative TuSimple samples. (a) A highway-exit scene with lane bifurcation and a sharply curved lane; (b) a multi-lane highway scene with distant thin lanes and vehicle interference; (c) a scene with near-range vehicle occlusion and large-vehicle interference; (d) a continuous curved-road scene affected by perspective compression. Different line colors are used only to distinguish individual lane instances and do not indicate different semantic categories.

Figure 7c presents a scene with near-range vehicle occlusion and large-vehicle interference. The vehicle on the left and the truck on the right occlude part of the lane region, while illumination variation and shadow reduce the contrast between lane markings and the road background. The baseline models can recognize the main lanes, but their boundaries near vehicle edges and shadow regions are unstable or shortened. The proposed model preserves more continuous predictions for the middle lane and the right lane boundary, suggesting that the improved multi-scale fusion provides more effective semantic compensation when local visible information is insufficient.

Figure 7d shows a continuous curved-road scene. The lane markings follow an arc-like layout, the far region is strongly affected by perspective compression, and near-range road cracks and texture changes may also interfere with segmentation. Some baselines capture the main curved lanes, but local interruptions, jagged boundaries, or unsmooth curve fitting can still be observed near the near and far ends. The proposed model produces more continuous predictions on the left, middle, and right curved lanes, and the curve trajectories remain more consistent with the road geometry. This phenomenon indicates that DSConv-based directional modeling and dynamic sampling can enhance the representation of curved slender structures, and can improve mask continuity when combined with multi-scale feature fusion.

Overall, the qualitative results in Figure 7 show that the proposed model achieves more stable lane-structure preservation in representative TuSimple scenes involving large-curvature lanes, distant thin lane markings, and vehicle occlusion. Compared with YOLO-series segmentation baselines, the proposed method reduces far-end truncation, local discontinuity, and fragmented responses to some extent. These visual observations are consistent with the quantitative results and support the positive effect of high-resolution feature preservation, elongated-structure modeling, and dynamic feature recovery on lane segmentation quality.

4.3.2. New CULane Dataset

To further examine generalization beyond TuSimple, a CULane subset with a scale close to TuSimple is constructed from the official CULane dataset, and representative samples under different road structures, lighting conditions, and traffic environments are selected for visualization. The subset includes more challenging cases, such as curves, shadows, vehicle occlusion, strong glare, and scenes without clear lane markings; Figure 8 shows representative results on daytime urban roads, elevated roads, tunnel or underpass entrances, nighttime roads, and low-illumination urban roads. Overall, the proposed model still maintains relatively stable segmentation performance compared with YOLOv11-seg, producing more continuous lane responses under glare, distant thin lane markings, curved lanes, and weak illumination, while its adaptability to highly complex scenarios still requires further verification.

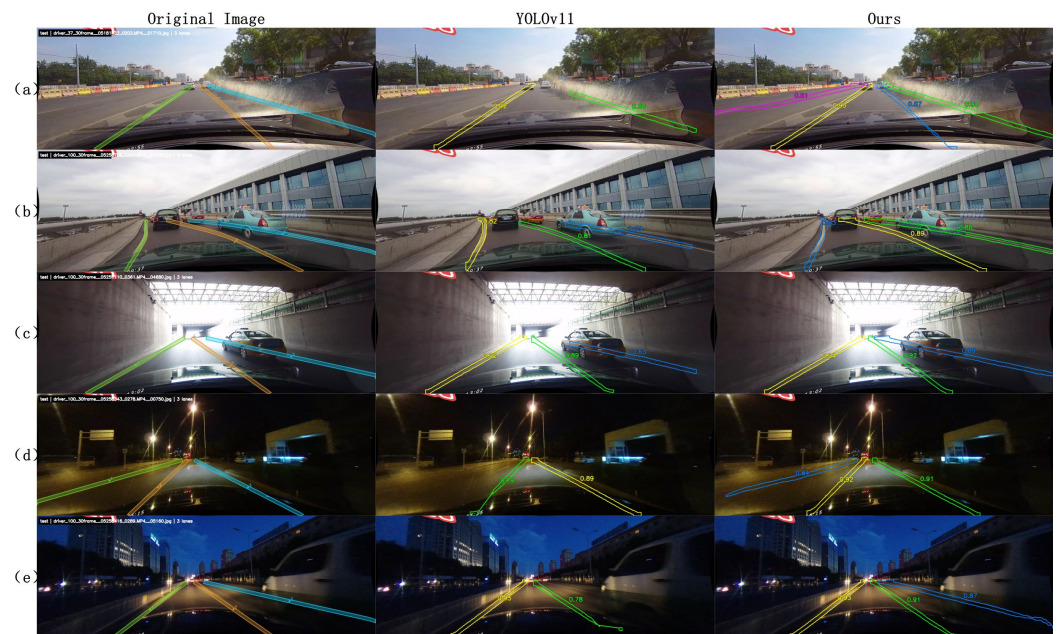


Figure 8. Qualitative comparison on representative samples from the newly constructed CULane subset. (a) A daytime urban-road scene with strong road-surface reflection; (b) an elevated-road or expressway scene with vehicles, guardrails, and building-edge interference; (c) a tunnel or underpass entrance scene with a strong illumination transition; (d) a nighttime road scene affected by low illumination and road-surface reflections; (e) a low-illumination urban-road scene with vehicle lights, building shadows, and road-edge interference. Different line colors are used only to distinguish individual lane instances and do not represent different semantic categories or performance levels.

In Figure 8a, the daytime urban road contains a strong reflection region on the right side, where the contrast between lane markings and the road surface is reduced. YOLOv11-seg shows partial discontinuity in this area, whereas the proposed model maintains a clearer lane response and better consistency with the road geometry. The proposed model also detects one additional lane in the middle region compared with YOLOv11-seg and predicts another lane on the left that is not covered by the annotation. These additional predictions are visually plausible in the original image, indicating that the model can capture richer visible road-linear structures. Since they are not fully matched with the dataset labels, they may still be counted as redundant predictions under the standard evaluation protocol and are interpreted only as supplementary qualitative observations rather than direct evidence of quantitative improvement.

Figure 8b corresponds to an elevated or expressway scene with preceding vehicles, traffic cones, guardrails, and building edges. YOLOv11-seg can recognize the main lane

markings, but local discontinuity appears around vehicles and in the far region. The proposed model provides more complete predictions for the middle and right lanes, and the lane extension remains more consistent with the road structure, suggesting that multi-scale feature fusion helps improve lane-structure representation in complex backgrounds.

Figure 8c shows a tunnel or underpass entrance scene with obvious illumination transition. Strong light near the exit and local shadows jointly increase the difficulty of lane recognition. YOLOv11-seg produces unstable responses near the high-brightness region, with some lane markings appearing as short fragments. The proposed model maintains better continuity on the middle lane and the right lane boundary, indicating a certain degree of adaptability to local illumination changes.

Figure 8d is a nighttime road scene, where low illumination, headlight reflection, and road-surface highlights reduce the contrast between lanes and background. YOLOv11-seg detects part of the main lanes, but local omissions remain. The proposed model maintains more stable responses at both near and far ranges, with clearer lane extension, suggesting that elongated-structure modeling helps improve lane continuity in low-contrast scenes.

Figure 8e presents a low-illumination urban road with building shadows, vehicle lights, and road-edge interference. YOLOv11-seg shows some truncation in the distant lane region, while the proposed model preserves the lane trajectory more completely and exhibits better continuity at the far end. This observation indicates that high-resolution feature preservation and dynamic feature recovery can provide more sufficient spatial information for thin lane markings in weak-texture scenes.

Taken together, the visualization results on the new CULane subset show that the proposed model maintains relatively good lane-structure preservation across multiple scene types. Compared with YOLOv11-seg, the proposed method reduces local discontinuity and far-end truncation in glare, illumination transition, nighttime, and distant thin-lane cases. Additional predictions that are visible in the image but not covered by annotations are interpreted cautiously as supplementary evidence of road-structure perception.

4.3.3. Failure Scene Analysis

To further analyze the limitations of the model in complex road environments, representative failure cases from the new CULane subset are examined. These cases include elevated-road occlusion, urban road-edge interference, and complex traffic backgrounds. The visualization results indicate that, although the proposed model generally preserves lane continuity, false responses and unstable confidence may still occur when non-lane linear structures are frequent or local scene information is insufficient.

Figure 9a,b show that the proposed model may identify non-lane structures as lane markings. These scenes contain guardrail edges, curbs, bridge structures, shadow boundaries, and road borders with elongated linear textures whose geometric appearance is similar to that of real lane markings. The strong response of the proposed model to slender structures helps improve lane continuity, but may also increase the risk of false responses to similar linear background patterns. These cases indicate that single-frame visual features remain insufficient to fully distinguish lane markings from non-lane linear structures in complex urban and elevated-road environments.

Figure 9c shows that the confidence of some individual lane predictions is lower than that of YOLOv11-seg. Vehicle occlusion, small distant targets, complex road textures, and illumination variation reduce the visibility of several lane markings in this scene. Although the proposed model detects the main lane structures, the confidence of some local predictions remains unstable, indicating that its local discriminative ability under low-visibility and complex-background conditions still requires improvement.

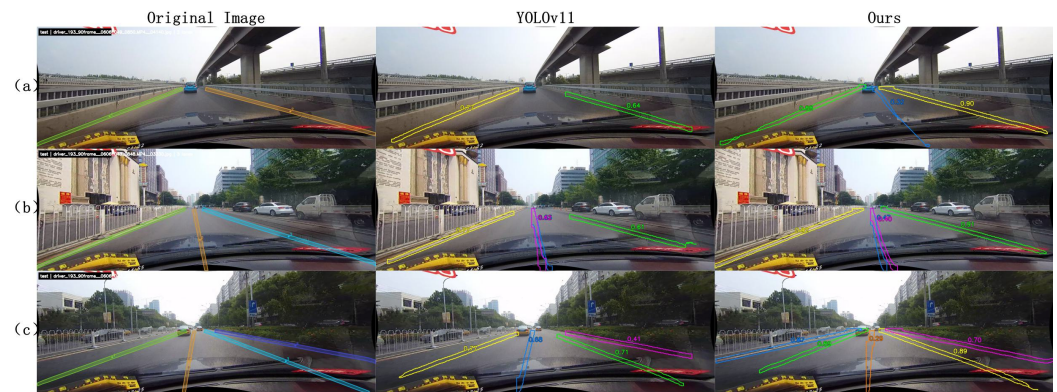


Figure 9. Representative failure cases on the newly constructed CULane subset. (a) An elevated-road scene in which shadow boundaries cause false lane responses; (b) an urban-road scene in which road boundaries and similar elongated linear textures result in redundant predictions; (c) a challenging road scene in which complex pavement textures and illumination variations lead to lower confidence for some lane predictions. Different line colors are used only to distinguish individual lane instances, while the numerical annotations indicate prediction confidence scores.

Overall, the observed failure cases are mainly associated with two issues: redundant predictions caused by non-lane linear structures that resemble real lane markings, and insufficient confidence caused by occlusion, distant targets, and complex illumination. These observations indicate that the proposed model still has limitations in complex road environments. Future work may incorporate lane-geometry constraints, temporal consistency modeling, and hard-sample enhancement to reduce false responses to similar linear backgrounds and improve prediction stability in challenging scenarios.

5. Conclusions

This paper addresses two key challenges in lane instance segmentation: the difficulty of preserving slender curvilinear lane structures and the loss of distant small lane targets after repeated downsampling. An improved YOLOv11-seg framework is proposed by integrating DSCConv into feature extraction and constructing a high-resolution P2 feature pyramid branch. A reproducible mask-generation and mask-to-point restoration pipeline is also established to adapt sparse TuSimple lane annotations to YOLO-style instance segmentation training and official point-level evaluation. Experiments on the TuSimple dataset show that, compared with the YOLOv11-seg baseline, the proposed method improves mask Precision by 8.5 percentage points, mask Recall by 9.0 percentage points, and mask mAP50 by 6.8 percentage points. The model contains 3.65 M parameters and 19.5 GFLOPs, and reaches 103.09 FPS under the reported testing configuration.

The ablation study indicates that DSCConv and the P2 branch are more effective when used together than when introduced separately, suggesting a complementary relationship between morphology-adaptive sampling and high-resolution spatial compensation. Qualitative analyses on TuSimple and the new CULane subset further show that the proposed model can improve visual lane continuity in curved, reflective, low-light, and partially occluded scenes. The comparison with representative lane detectors and the failure cases also clarify the performance boundary of the proposed method: specialized lane detection architectures remain competitive under official task-specific metrics, and lane-like linear backgrounds or weak lane visibility may still lead to redundant responses or unstable confidence. Future work will extend the evaluation to broader datasets such as CULane and BDD100K, quantify structural continuity and failure modes more systematically, and explore pruning, quantization, and deployment on edge computing devices.

Author Contributions: Conceptualization, S.W. and B.G.; Methodology, S.W. and W.G.; Software, S.W. and X.Z.; Validation, B.G., W.G. and A.S.; Formal analysis, S.W. and D.B.; Investigation, W.G., X.Z. and A.S.; Resources, B.G.; Data curation, X.Z. and A.S.; Writing—original draft, S.W.; Writing—review & editing, B.G. and D.B.; Visualization, W.G.; Supervision, X.Z.; Project administration, B.G. and D.B.; Funding acquisition, B.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Plan Project of Henan Provincial Department of Transportation (Grant No. 2023-2-1), the Henan Provincial Science and Technology Decision-Making Consultation Project (Grant No. SKXJCZX-2026-29B) supported by Henan Association for Science and Technology, and the Henan Provincial Science and Technology Research Project supported by Henan Provincial Department of Science and Technology (Grant No. 262102521014)

Data Availability Statement: The data supporting the results of this study are publicly available. This research is based on a publicly sourced dataset: the TuSimple dataset, accessible at <https://github.com/TuSimple/tusimple-benchmark>, accessed on 18 March 2026.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Li, K.; Han, Z.W.; Chen, J.J.; Li, B.; Wang, H.B. Fuzzy output feedback cooperative control for lane-keeping assistance systems. *Transducer Microsyst. Technol.* **2023**, *42*, 95–102. [https://doi.org/10.13873/J.1000-9787\(2023\)12-0095-04](https://doi.org/10.13873/J.1000-9787(2023)12-0095-04).
- Lü, C.; Cui, G.; Meng, X.; Lu, J.; Xu, Y.; Gong, J. Graph representation method for pedestrian intention recognition of intelligent vehicle. *Trans. Beijing Inst. Technol.* **2022**, *42*, 688–695. <https://doi.org/10.15918/j.tbit1001-0645.2021.330>.
- Liu, F.; Lu, Z.; Lin, X. Vision-Based Environmental Perception for Autonomous Driving. *arXiv* **2022**, arXiv:2212.11453. <https://doi.org/10.48550/arXiv.2212.11453>.
- Tu, C.; van Wyk, B.J.; Hamam, Y.; Djouani, K.; Du, S. Vehicle Position Monitoring Using Hough Transform. *IERI Procedia* **2013**, *4*, 316–322. <https://doi.org/10.1016/j.ieri.2013.11.045>.
- Yoo, J.H.; Lee, S.W.; Park, S.K.; Kim, D.H. A Robust Lane Detection Method Based on Vanishing Point Estimation Using the Relevance of Line Segments. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3254–3266. <https://doi.org/10.1109/TITS.2017.2679222>.
- Jiang, L.B.; Li, J.X. Optimization Algorithm for Lane Detection Based on Improved Hough Transform and Double-Point Removal R-Least Squares. *Sci. Technol. Eng.* **2020**, *20*, 2070–2076. <https://doi.org/10.3969/j.issn.1671-1815.2020.05.054>.
- Wang, H.F.; Wang, Y.F.; Zhao, X.M.; Wang, G.P.; Huang, H.; Zhang, J. Lane Detection of Curving Road for Structural Highway With Straight-Curve Model on Vision. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5321–5330. <https://doi.org/10.1109/TVT.2019.2913187>.
- Tang, J.; Li, S.; Liu, P. A Review of Lane Detection Methods Based on Deep Learning. *Pattern Recognit.* **2021**, *111*, 107623. <https://doi.org/10.1016/j.patcog.2020.107623>.
- Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2017; pp. 2980–2988. <https://doi.org/10.1109/ICCV.2017.324>.
- Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 263–272. <https://doi.org/10.1109/TITS.2017.2750080>.
- Zou, Q.; Jiang, H.; Dai, Q.; Yue, Y.; Chen, L.; Wang, Q. Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks. *IEEE Trans. Veh. Technol.* **2019**, *69*, 41–54. <https://doi.org/10.1109/TVT.2019.2949603>.
- Zhou, Y.C.; Qi, H.Z.; Ma, Y. End-to-End Wireframe Parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2019; pp. 962–971. <https://doi.org/10.1109/ICCV.2019.00105>.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>.
- Qin, Z.; Wang, H.; Li, X. Ultra Fast Structure-Aware Deep Lane Detection. In *Proceedings of Computer Vision—ECCV 2020: 16th European Conference*; Springer International Publishing: Cham, Switzerland, 2020; pp. 276–291. https://doi.org/10.1007/978-3-030-58586-0_17.
- Tabelini, L.; Berriel, R.; Paixao, T.M.; Badue, C.; De Souza, A.F.; Oliveira-Santos, T. Keep Your Eyes on the Lane: Real-Time Attention-Guided Lane Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2021; pp. 294–302. <https://doi.org/10.1109/CVPR46437.2021.00036>.
- Tabelini, L.; Berriel, R.; Paixao, T.M.; Badue, C.; De Souza, A.F.; Oliveira-Santos, T. PolyLaneNet: Lane Estimation via Deep Polynomial Regression. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*; IEEE: Piscataway, NJ, USA, 2021; pp. 6150–6156. <https://doi.org/10.1109/ICPR48806.2021.9412265>.

17. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards End-to-End Lane Detection: An Instance Segmentation Approach. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*; IEEE: Piscataway, NJ, USA, 2018; pp. 286–291. <https://doi.org/10.1109/IVS.2018.8500547>.
18. Guo, Z.; Huang, Y.; Wei, H.; Zhang, C.; Zhao, B.; Shao, Z. DALaneNet: A Dual Attention Instance Segmentation Network for Real-Time Lane Detection. *IEEE Sens. J.* **2021**, *21*, 21730–21739. <https://doi.org/10.1109/JSEN.2021.3100489>.
19. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial as Deep: Spatial CNN for Traffic Scene Understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2018; Volume 32. <https://doi.org/10.1609/aaai.v32i1.12314>.
20. Zheng, T.; Huang, Y.; Liu, Y.; Tang, W.; Yang, Z.; Cai, D.; He, X. CLRNNet: Cross Layer Refinement Network for Lane Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2022; pp. 898–907. <https://doi.org/10.1109/CVPR52688.2022.00097>.
21. Zheng, T.; Fang, H.; Zhang, Y.; Tang, W.; Yang, Z.; Liu, H.; Cai, D. RESA: Recurrent Feature-Shift Aggregator for Lane Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2021; Volume 35, pp. 3547–3554. <https://doi.org/10.1609/aaai.v35i4.16469>.
22. Liu, L.; Chen, X.; Zhu, S.; Tan, P. CondLaneNet: A Top-to-Down Lane Detection Framework Based on Conditional Convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 3773–3782. <https://doi.org/10.1109/ICCV48922.2021.00375>.
23. Ko, Y.; Lee, Y.; Azam, S.; Munir, F.; Jeon, M.; Pedrycz, W. Key Points Estimation and Point Instance Segmentation Approach for Lane Detection. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 8949–8958. <https://doi.org/10.1109/TITS.2021.3088488>.
24. Hou, Y.; Ma, Z.; Liu, C.; Loy, C.C. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2019; pp. 1013–1021. <https://doi.org/10.1109/ICCV.2019.00110>.
25. Ai, Q.L.; Zhang, J.R.; Wu, F.Q. AF-ICNet: Unstructured Scene Semantic Segmentation Method Based on Small Target Class Attention Mechanism and Feature Fusion. *Acta Photonica Sin.* **2023**, *52*, 0110001. <https://doi.org/10.3788/gzxb20235201.0110001>.
26. Han, J.; Deng, X.; Cai, X.; Yang, Z.; Xu, H.; Xu, C.; Liang, X. LaneFormer: Object-Aware Row-Column Transformers for Lane Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2022; Volume 36, pp. 799–807. <https://doi.org/10.1609/aaai.v36i1.19961>.
27. Liu, R.; Yuan, Z.; Liu, T.; Xiong, Z. End-to-End Lane Shape Prediction with Transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 3694–3702. <https://doi.org/10.1109/WACV48630.2021.00374>.
28. Wang, J.; Ma, Y.; Huang, S.; Hui, T.; Wang, F.; Qian, C.; Zhang, T. A Keypoint-Based Global Association Network for Lane Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1392–1401. <https://doi.org/10.1109/CVPR52688.2022.00145>.
29. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2016; pp. 779–788. <https://doi.org/10.1109/CVPR.2016.91>.
30. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725. <https://doi.org/10.48550/arXiv.2410.17725>.
31. Alif, M.A.R.; Hussain, M. YOLOv1 to YOLOv10: A Comprehensive Review of YOLO Variants and Their Application in the Agricultural Domain. *arXiv* **2024**, arXiv:2406.10139. <https://doi.org/10.48550/arXiv.2406.10139>.
32. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. <https://doi.org/10.48550/arXiv.2004.10934>.
33. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2023; pp. 7464–7475. <https://doi.org/10.1109/CVPR52729.2023.00721>.
34. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616. <https://doi.org/10.48550/arXiv.2402.13616>.
35. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458. <https://doi.org/10.48550/arXiv.2405.14458>.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
37. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–19. https://doi.org/10.1007/978-3-030-01234-2_1.

38. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2018; pp. 8759–8768. <https://doi.org/10.1109/CVPR.2018.00913>.
39. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430. <https://doi.org/10.48550/arXiv.2107.08430>.
40. Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; Yang, G. Dynamic Snake Convolution Based on Topological Geometric Constraints for Tubular Structure Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2023; pp. 6070–6079. <https://doi.org/10.1109/ICCV51070.2023.00560>.
41. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2017; pp. 50–58. <https://doi.org/10.1109/ICCV2017.89>.
42. Lee, S.; Kim, J.; Yoon, J.S.; Shin, S.; Bailo, O.; Kim, N.; Lee, T.H.; Hong, H.S.; Han, S.H.; Kweon, I.S. VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2017; pp. 1947–1955. <https://doi.org/10.1109/ICCV2017.215>.
43. Xu, H.; Wang, S.; Cai, X.; Zhang, W.; Liang, X.; Li, Z. CurveLane-NAS: Unifying Lane-Sensitive Architecture Search and Adaptive Point Blending. In *Proceedings of Computer Vision—ECCV 2020: 16th European Conference*; Springer International Publishing: Cham, Switzerland, 2020; pp. 689–704. https://doi.org/10.1007/978-3-030-58555-6_41.
44. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-Style ConvNets Great Again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2021; pp. 13733–13742. <https://doi.org/10.1109/CVPR46437.2021.01352>.
45. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2017; pp. 2117–2125. <https://doi.org/10.1109/CVPR.2017.106>.
46. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2017; pp. 2961–2969. <https://doi.org/10.1109/ICCV.2017.322>.
47. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2019; pp. 9157–9166. <https://doi.org/10.1109/ICCV.2019.00925>.
48. Tian, Z.; Shen, C.; Chen, H. Conditional Convolutions for Instance Segmentation. In *Proceedings of Computer Vision—ECCV 2020: 16th European Conference*; Springer International Publishing: Cham, Switzerland, 2020; pp. 282–298. https://doi.org/10.1007/978-3-030-58452-8_17.
49. Yin, C.; Cecotti, M.; Auger, D.J.; Fotouhi, A.; Jiang, H. Lane Centerline Extraction Based on Surveyed Boundaries: An Efficient Approach Using Maximal Disks. *Sensors* **2025**, *25*, 2571. <https://doi.org/10.3390/s25082571>.
50. Wang, Y.; Wang, W.; Liu, J.; Chen, T.; Wang, S.; Yu, B.; Qin, X. Framework for Geometric Information Extraction and Digital Modeling from LiDAR Data of Road Scenarios. *Remote Sens.* **2023**, *15*, 576. <https://doi.org/10.3390/rs15030576>.

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.