FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

An Improved UFLD-V2 Lane Line Recognition Method

LiKang Bo[1,2]*, Fei Lu Siaw[1], Tzer Hwai Gilbert Thio[1]

[1]*Centre for Sustainability in Advanced Electrical and Electronic Systems (CSAEES), Faculty of Engineering, Built Environment and Information Technology, SEGi University, 47810 Petaling Jaya, Selangor, Malaysia;*
[2]*Hebei Vocational University of Technology and Engineering, No.473, Quannan West Street, Xindu District, Xingtai, Hebei, China, 054000*

***Correspondence:** *LiKang Bo; bolk88@163.com*

ABSTRACT- Lane line recognition remains a crucial component of autonomous driving, particularly under complex scenarios involving illumination changes and occlusions. This paper presents a structurally efficient and robust improvement of the UFLD-V2 architecture, designed for real-time and reliable lane detection. The proposed method integrates three lightweight yet complementary components: *(1)* Res2Net, replacing the original ResNet backbone, enhances multi-scale feature extraction and inference efficiency through reparameterization; *(2)* an Efficient Multi-scale Attention (EMA) module captures fine-grained contextual details across varying scene complexities; and *(3)* the Simple Attention Module (SimAM) is applied in the segmentation head to suppress background noise and improve localization accuracy. Unlike prior work that uses these modules in isolation, we propose a tailored integration strategy that achieves a favorable trade-off between accuracy and computational cost. Extensive experiments on the TuSimple dataset show the effectiveness of our method, achieving 0.95947 accuracy, 0.0262 false positive rate, 0.02328 false negative rate, and an F1 score of 0.96517. Our approach surpasses several state-of-the-art models, including UFLD, PolyLaneNet, EL-GAN, SAD, CurveFormer++, BEVLaneDet, and PersFormer, particularly under challenging conditions. These findings highlight the potential of our approach for practical deployment in intelligent driving systems.

Keywords: Lane Line Recognition, UFLD-V2, Res2Net, EMA Attention Module, SimAM Module.

ARTICLE INFORMATION
Author(s): LiKang Bo, Fei Lu Siaw, Tzer Hwai Gilbert Thio;
Received: 21/01/2025; **Accepted**: 25/04/2025; **Published**: 18/06/2025;
E- ISSN: 2347-470X;
Paper Id: IJEER 2101-05;
Citation: 10.37391/ijeer.130211
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-13/ijeer-130211.html

crossref member
CROSSREF.ORG THE CITATION LINKING BACKBONE

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

With the rapid development of autonomous driving technology, lane line detection has become a fundamental task for ensuring vehicle safety and enabling accurate path planning [1]. In real-world environments, however, lane detection remains highly susceptible to performance degradation under challenging conditions such as illumination variation, occlusion, and road surface clutter [2]. Traditional rule-based or handcrafted-feature methods typically lack the adaptability and robustness required for such scenarios.

Recent advances in deep learning, especially those involving convolutional neural networks (CNNs), have led to significant improvements in lane detection performance. Among these, UFLD-V2 has emerged as a highly regarded framework due to its balance between structural efficiency and detection accuracy. Unlike earlier approaches such as UFLD [3], UFLD-V2[4] introduces a dedicated detection head and adaptive anchor mechanisms that enable fast, structured lane localization. However, despite its solid baseline performance, UFLD-V2 still exhibits several limitations:

- Insufficient multi-scale feature representation, which hinders its adaptability to varying lane shapes and scales;
- Limited attention modeling, which weakens its ability to focus on important spatial and semantic information;
- Inadequate robustness in the presence of low visibility, background noise, or occluded lanes.

To address these challenges, we propose an enhanced UFLD-V2-based framework that integrates three lightweight yet complementary modules:

1. A Res2Net backbone to improve multi-scale feature extraction and hierarchical representation;
2. An Efficient Multi-Scale Attention (EMA) module to refine context modeling across spatial and channel dimensions;
3. A Simple Attention Module (SimAM) in the segmentation head to suppress background interference and enhance localization accuracy.

Unlike previous works that employ such modules in isolation, our design implements a task-specific, latency-aware integration strategy. We also apply reparameterization techniques to the backbone network during inference to compress multi-branch paths into a single branch, thereby preserving real-time performance.

This design achieves a favorable balance between detection accuracy, model robustness, and computational efficiency, making it well-suited for practical deployment in intelligent driving systems.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

The rest of the paper is organized as follows: *Section 2* reviews related work, *section 3* introduces the proposed method, *section 4* presents experimental results, and *Section 5* concludes the paper.

2. RELATED WORKS

Lane detection, as a crucial component of autonomous driving systems, plays a vital role in ensuring vehicle safety, supporting decision-making, and maintaining driving stability[5]. Consequently, it has become a research hotspot in the fields of intelligent transportation and computer vision in recent years. Traditional lane detection methods primarily rely on handcrafted features and classical image processing techniques, such as color thresholding, edge detection, and the Hough transform. These approaches can deliver acceptable performance under controlled conditions; however, they often lack robustness in complex environments involving shadow interference, illumination changes, lane occlusions, and road texture variability. This limits their adaptability to diverse and unstructured road scenarios. With the rapid development of deep learning, convolutional neural network (CNN)-based methods have become dominant[6]. These methods enable end-to-end learning, automatically extracting multi-scale features from input images—ranging from low-level textures to high-level structural semantics—leading to more accurate and stable lane detection under challenging conditions[7]. Furthermore, the integration of attention mechanisms and self-supervised learning techniques in recent work has further improved generalization and robustness, enabling more reliable performance in real-world driving scenes[8].

Feature-based Lane detection methods primarily rely on handcrafted features and classical image processing pipelines. For example, some approaches utilize color thresholding in HSV or CIELab color spaces, edge detection, and the Hough transform to extract lane markings[9-11]. These techniques can achieve good performance under stable lighting and clear road conditions. However, their strong dependence on color and edge contrast limits their adaptability in real-world environments with variable illumination, shadowing, or blurred markings.

Model-based methods incorporate geometric constraints to represent lane trajectories using straight lines, parabolas, or splines. Lane boundaries are typically extracted using algorithms such as RANSAC, least squares fitting, or vanishing point estimation[12-14]. These methods provide interpretable results and can perform well in structured road scenarios. Nevertheless, they often assume ideal road geometries or consistent lane visibility, which makes them less effective in unstructured or highly dynamic environments.

Deep learning-based lane detection methods have become dominant due to their ability to learn hierarchical visual features and adapt to complex driving environments. Compared with traditional techniques, CNN-based approaches offer greater robustness against illumination changes, occlusions, and road texture variations. End-to-end frameworks such as instance segmentation models [15, 16], generative adversarial methods like EL-GAN[16], and regression-based networks like PolyLaneNet[17] have achieved strong performance on benchmarks like TuSimple and CULane.

More recently, a number of UFLD-based extensions have been proposed to further improve accuracy and robustness while maintaining real-time efficiency. MCA-UFLD[18], for example, utilizes a lightweight MobileNetV2 backbone with coordinate attention and a vanishing point branch to enhance semantic perception without sacrificing speed. Other methods incorporate Split-Attention and ASFF modules to enhance multi-scale feature fusion, resulting in notable gains in F1-score on CULane[8]. Additionally, techniques such as random masking and smooth curve loss have been employed to improve performance under occlusions and enhance the continuity of predicted lane lines. To further increase generalization under poor visibility, contrastive learning strategies such as CLLD have also been introduced[19].

These studies highlight the effectiveness of integrating lightweight backbones, attention mechanisms, and auxiliary learning strategies for lane detection. Inspired by these advances, we propose a structurally enhanced UFLD-V2 framework that incorporates a Res2Net backbone for multi-scale feature extraction, an Efficient Multi-scale Attention (EMA) module for refined spatial-channel modeling, and a Simple Attention Module (SimAM) to suppress background interference. Unlike prior works that use such modules independently, our method adopts a task-driven, latency-aware integration strategy designed to maintain real-time applicability while significantly improving robustness and accuracy. Detailed descriptions are provided in *section 3*.

3. PROPOSED METHODOLOGY

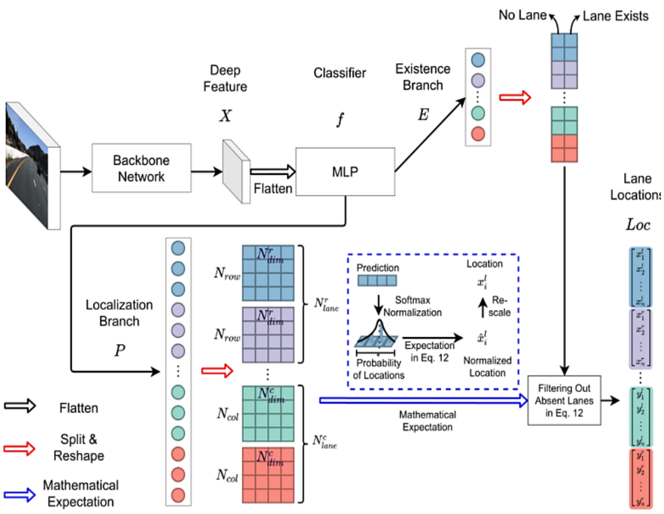


Figure 1. UFLDv2 network architecture

Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

Figure 1 is the basic network studied in this paper. It consists of two main branches: existence branch and localization branch. The input image is first extracted through the backbone network to extract deep features, and then a multi-layer perceptron (MLP) is used to determine whether there is a lane in the image. If there is a lane, the localization branch is responsible for predicting the specific location of the lane. Finally, the final lane location is output.

To enhance the robustness and precision of intelligent vehicle lane line detection in complex scenarios, this paper optimizes the UFLD-V2 model, proposing an innovative method that combines multi-scale feature fusion and attention mechanisms. The method fully utilizes feature extraction network capabilities and efficient attention mechanisms, optimizing and reconstructing the original model's backbone network, feature extraction modules, and segmentation head. Specific improvements are as follows:

3.1. Optimizing Backbone Feature Extraction Network

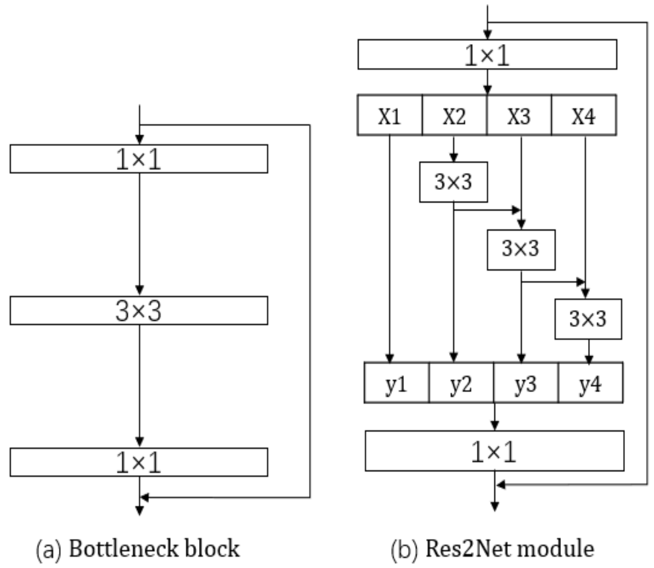


Figure 2. Bottleneck block and Res2Net module

In the UFLD-V2 model, the backbone network has limitations in multi-scale feature extraction capabilities. To address this limitation, this paper adopts Res2Net[20] to replace ResNet. Res2Net introduces hierarchical representation of multi-scale features, efficiently capturing local and global information in complex scenarios. Compared to traditional ResNet, Res2Net enhances feature expression capabilities through multi-scale group convolutions, better adapting to complex road environments. Additionally, this paper uses reparameterization technology to optimize Res2Net. During the training phase, multi-branch structures enhance learning capabilities, while in the inference phase, multiple branches are merged into a single branch, significantly improving inference efficiency and reducing computational overhead.

3.2. Introducing EMA Attention Module

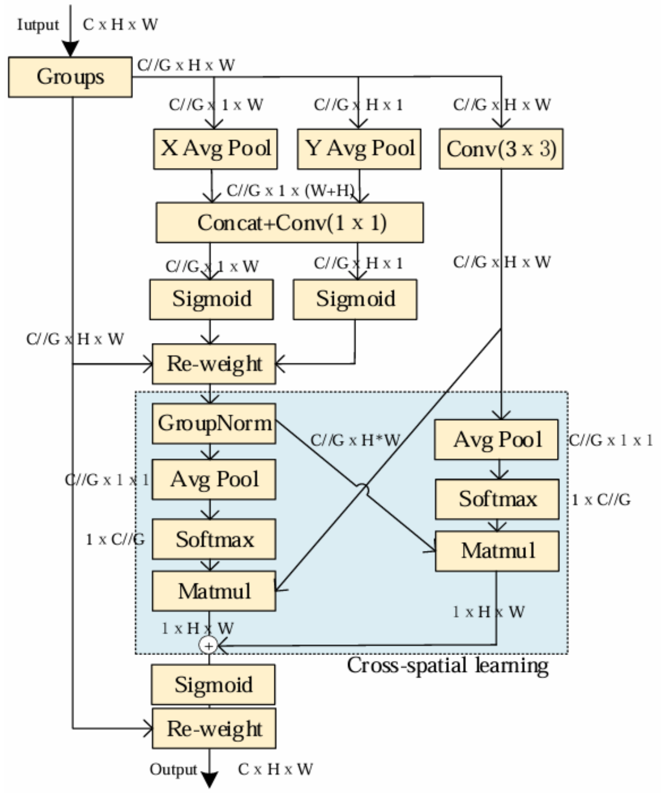


Figure 3. EMA module

The Efficient Multi-Scale Attention (EMA) module enhances lane detection under complex scenarios by fusing multi-scale spatial context and reweighting feature channels dynamically. It consists of multi-scale convolutional paths and channel-wise attention computation.

First, the input feature $F \in \mathcal{R}^{C\times H\times W}$ is passed through multiple convolution branches:

$$F_1 = Conv_{1\times1}(F),\ \ F_3 = Conv_{3\times3}(F),\ \ F_5 = Conv_{5\times5}(F) \quad (1)$$

These are concatenated and fused:

$$F_{fused} = Concat(F_1, F_3, F_5) \longrightarrow BN \longrightarrow ReLU \quad (2)$$

Then, a global average pooling generates channel statistics zzz, and a two-layer MLP computes attention weights:

$$z = \frac{1}{H\cdot W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_{fused}\,(i,j)\ , a = \sigma(W_2 \cdot ReLU(W_1 \cdot z)) \quad (3)$$

Finally, the output is obtained by applying the learned weights:

$$F_{out} = F_{fused} \odot a \quad (4)$$

EMA is embedded into the output of the deepest stage of Res2Net in our model, enabling enhanced detail recognition and robustness in scenarios with occlusion, lighting variation, or cluttered backgrounds.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

3.3. Optimizing Segmentation Head with SimAM Module

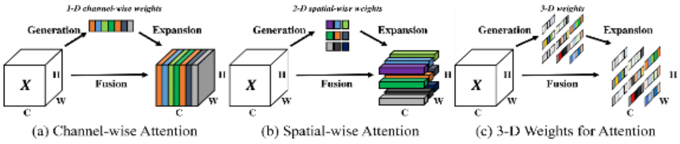


Figure 4. SimAM module

The Similarity Attention Mechanism (SimAM) is a lightweight, parameter-free attention module inspired by neuroscience, which evaluates the importance of each neuron in a feature map. Unlike traditional attention methods (*e.g.*, SE-Net), SimAM does not rely on convolution or MLP operations, thus ensuring low computational cost and real-time performance.

The core idea of SimAM is to define an energy function E_i for each neuron x_i, which consists of its deviation from the mean and suppression from other neurons:

$$E_i = (x_i - \mu)^2 + \lambda \sum_{j \neq i} x_j^2 \qquad (5)$$

where μ is the average response of all neurons in the same channel and λ is a suppression coefficient. The importance score is computed as:

$$\alpha_i = \frac{1}{1+e^{-E_i}} \qquad (6)$$

This scalar α_i is used to reweight the neuron's activation, thereby enhancing foreground features and suppressing irrelevant background. In our model, SimAM is inserted after the pooling layer in the segmentation head, significantly improving the detection precision for complex scenarios.

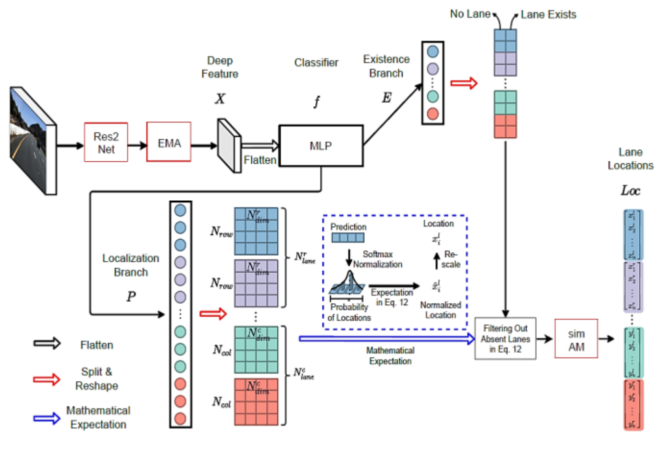


Figure 5. Improved Network Architecture

By optimizing the backbone network, introducing multi-scale attention mechanisms, and combining lightweight attention modules, the lane line detection method proposed in this paper demonstrates stronger robustness and higher detection precision when facing complex scenarios. Experimental results show that the improved model significantly outperforms the original UFLD-V2 in multiple complex scenarios, verifying the effectiveness and superiority of the proposed method.

3.4. Integration Strategy and Efficiency Considerations

To comprehensively improve robustness, accuracy, and efficiency in complex scenarios, this paper integrates three modules—Res2Net, EMA, and SimAM—into the UFLD-V2 framework. Each module plays a distinct yet complementary role in addressing the limitations of the original model.

Res2Net, replacing the original ResNet backbone, enhances multi-scale feature extraction by introducing hierarchical residual-like connections. During the training phase, its multi-branch structure improves the learning capability. During inference, reparameterization is applied to merge branches into a single-path structure, significantly accelerating prediction and reducing computational cost.

The EMA (Efficient Multi-scale Attention) module is added to both low-level and high-level feature maps within the Res2Net backbone. By dynamically fusing multi-scale spatial context and computing adaptive channel attention, EMA effectively enhances fine-grained feature representation in scenarios involving occlusion, lighting variations, and background clutter.

SimAM, a parameter-free lightweight attention mechanism, is inserted after the pooling layer in the segmentation head. By computing importance scores for each neuron based on an energy function, SimAM suppresses irrelevant background and enhances the localization of lane boundaries without additional computational burden.

This integration strategy results in a structurally efficient and inference-friendly model. The proposed method was trained on the TuSimple dataset, with the input resolution downscaled to 320×800 to improve speed. The training was conducted using 200 epochs, batch size of 16, Adam optimizer with a learning rate of 0.05 (adjusted using cosine annealing), and a weight decay of 0.0005. All experiments were performed under Ubuntu 20.04, using the PyTorch framework and NVIDIA H800 GPU.

Notably, the improved model achieves an inference speed of 235 FPS on RTX 2080Ti-equivalent hardware, which is only slightly lower than UFLD-V2 (260 FPS), while significantly outperforming other attention or transformer-based models such as CurveFormer++ (110 FPS). These results confirm that the improved architecture maintains real-time processing capability while achieving better robustness and accuracy, making it well-suited for practical deployment in autonomous driving systems.

4. RESULTS

This section presents the experimental setup, data preprocessing steps, evaluation metrics, ablation study, performance benchmarking, inference efficiency, and visual analysis of the proposed method. All experiments follow the configuration

Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

detailed in *section 3.4*, ensuring consistency between architectural design and training procedures.

4.1. Training Settings

The model was trained on the TuSimple dataset, with the original image resolution (1280×720) downscaled to 320×800 to improve computational efficiency. Training was conducted for 200 epochs using a batch size of 16. The Adam optimizer was employed with an initial learning rate of 0.05, which was decayed using a cosine annealing schedule. A weight decay of 0.0005 was applied to mitigate overfitting. All experiments were conducted on Ubuntu 20.04 using the PyTorch framework and an NVIDIA H800 GPU.

To improve model generalization and reduce overfitting, a variety of data augmentation techniques were applied online during training *via* PyTorch's data pipeline. These include:

- Random horizontal flipping, which simulates mirrored driving scenarios and enhances robustness to diverse road geometries;
- Random brightness and contrast adjustment, which introduces lighting variation to help the model learn under changing illumination, such as shadows, overexposure, and sunlight reflections;
- Gaussian noise injection and slight affine transformations, which simulate real-world camera disturbances and environmental imperfections.

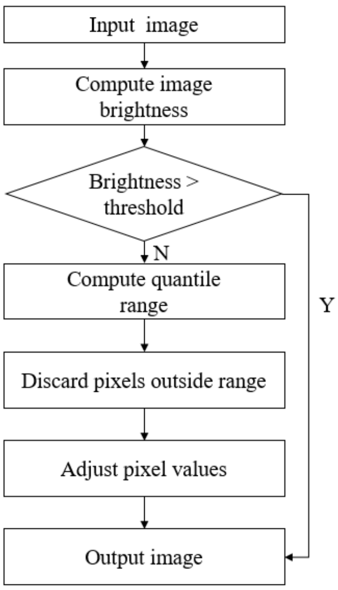


Figure 6. Flowchart of the Adaptive Brightness Compensation Enhancement (ABCE) algorithm

Additionally, we introduce a targeted low-light enhancement method called adaptive brightness compensation enhancement (ABCE) to further improve robustness under poor illumination. As illustrated in *figure 6*, ABCE first computes the grayscale brightness of the input image. If the brightness is above a defined threshold, the image remains unchanged. Otherwise, the algorithm:

- Computes the intensity range between the 5th and 95th percentiles of pixel values;
- Discards pixels outside this range to remove outliers;
- Applies linear contrast stretching or gamma correction to enhance important visual details;
- Outputs the adjusted image for model training.

This preprocessing strategy selectively enhances dark or shadowed images during training, ensuring lane markings remain visible while maintaining a lightweight and efficient training pipeline.

The ABCE module enhances underexposed training images by removing intensity outliers and normalizing contrast, improving lane visibility without increasing model complexity.

(a)

(b)

Figure 7. Visual comparison before and after applying ABCE algorithm

The left image shows the original low-light input from the TuSimple dataset, where lane markings are difficult to distinguish due to poor illumination. The right image displays the result after applying the proposed adaptive brightness compensation enhancement (ABCE) algorithm, which improves local contrast and lane visibility. This preprocessing enhances the input quality before lane detection, especially under shadowed or underexposed conditions.

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

4.2. Dataset Preprocessing and Label Generation

To facilitate structured training, the original TuSimple dataset annotations in JSON format were converted into a standardized format using a custom preprocessing script (convert_tusimple.py). The process involves the following key steps:

- ***Lane classification:*** Lane lines are extracted from the annotation files and classified based on their slope. Lanes are sorted from left to right, and lines with a length shorter than 90 pixels are filtered out to exclude incomplete or irrelevant annotations.
- ***Segmentation label generation:*** For each input image, a corresponding segmentation mask is generated where pixel values {1, 2, 3, 4} represent the four lane lines from left to right. These masks serve as the ground truth for segmentation-based training.
- ***Structured label files:*** Two text files are generated—train_gt.txt for training and test.txt for testing. The training file records the path to each image and the binary existence indicators for lane positions, while the test file contains paths to test images for evaluation without labels.
- ***Annotation caching:*** Parsed Lane information is saved into a cache file (tusimple_anno_cache.json) to reduce preprocessing time during repeated training runs.

This preprocessing step ensures that the lane annotations are consistent, structured, and suitable for deep learning-based lane detection frameworks, while remaining fully compatible with the official TuSimple evaluation protocol.

4.3. Evaluation indicators

This paper uses the following evaluation indicators commonly used in the field of object detection to evaluate the lane line detection performance: Accuracy (Acc), False Negative (FN), False Positive (FP) and F1 score. The definitions of each indicator are as follows:

Accuracy: used to measure the ability of the model to make correct predictions.

$$A_{cc} = \frac{\sum_{clip} C_{clip}}{\sum_{slip} Sslip} \qquad (7)$$

C_{clip} represents the number of lane line points correctly predicted by the model, and S_{slip} represents the number of lane line points in the corresponding group-truth.

Missed detection rate: This value is the proportion of lane line points identified as non-lane line points in the model's prediction.

$$FN = \frac{M_{pred}}{N_{gt}} \qquad (8)$$

Among them, M_{pred} represents the number of lane line points missed by the model in the prediction, and N_{gt}represents the number of all lane line points in the label.

False positive rate: This value is the ratio of non-lane points predicted by the model as lane line points.

$$FP = \frac{F_{pred}}{N_{pred}} \qquad (9)$$

Among them, F_{pred} represents the number of lane line points that are misdetected during prediction, and N_{pred} represents the total number of lane line points in the prediction results.

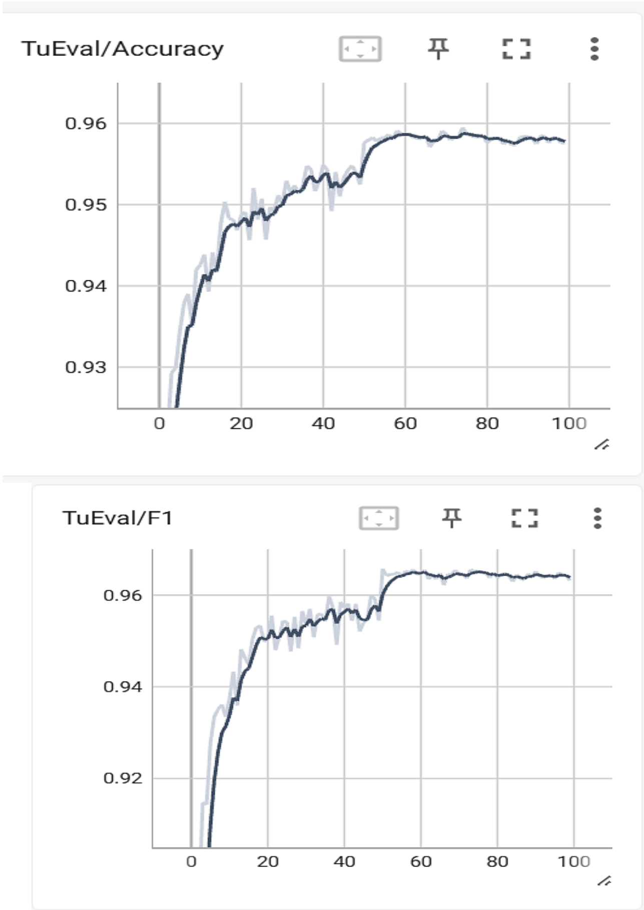
F1 score: An indicator that comprehensively evaluates the performance of a classification model. It is often used to measure the balance between the accuracy and recall of binary and multi-classification models. It is the harmonic average of the two.

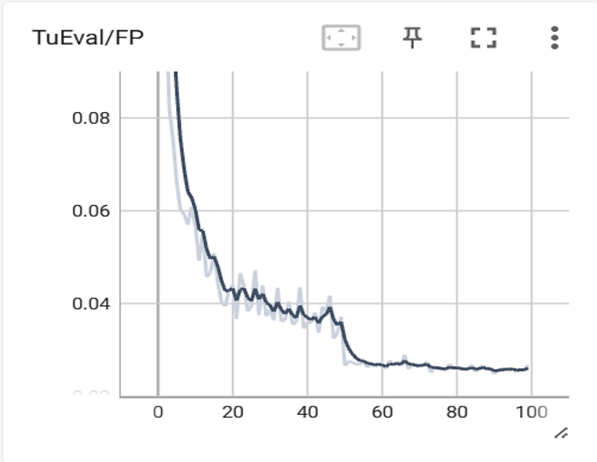
$$F1 = \frac{2 \times P \times R}{P + R} \qquad (10)$$

Among them, $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$ TP represents the number of samples correctly recognized by the model, and TN represents the number of negative samples recognized by the model as not lane lines.

4.4. Performance Curves

Figure 8 shows the training progression of accuracy, false positives, false negatives, and F1 score over 200 epochs. The model demonstrates stable convergence and consistent performance gains across all metrics.





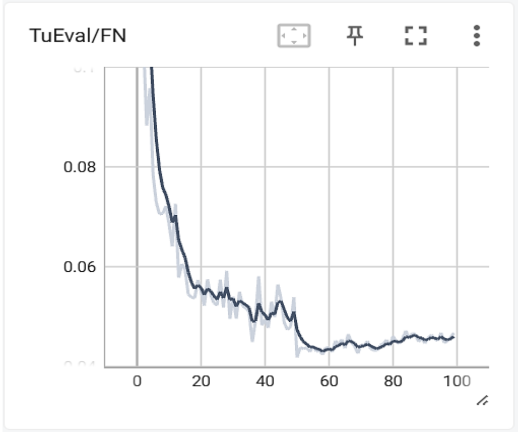


Figure 8. Training curves of Accuracy, False Positive Rate, False Negative Rate, and F1 Score over the first 100 epochs. The complete training spanned 200 epochs, with continued performance refinement observed beyond the 100-epoch mark. Only the initial portion is visualized here for clarity

4.5. Ablation experiment

Table 1 presents an ablation study evaluating the individual and combined effects of Res2Net, EMA, and SimAM. Performance is recorded at each stage, showing incremental improvements

Table 1. Comparison of datasets at different stages

Res2Net	EMA	SimAM	Accuracy (%)	FP (%)	FN (%)	F1 Score (%)
–	–	–	95.79	2.94	4.85	96.09
√	–	–	95.66	2.69	4.81	96.23
–	√	–	95.83	2.80	4.58	96.20
–	–	√	95.81	2.75	4.62	96.18
√	√	–	95.85	2.93	4.52	96.26
√	–	√	95.89	2.65	4.06	96.38
–	√	√	95.87	2.59	3.85	96.41
√	√	√	**95.95↑**	**2.62↓**	**2.33↓**	**96.52↑**

(Note: All results are averaged over three repeated runs under fixed random seeds)

Table 1 presents the results of ablation experiments conducted to evaluate the individual and combined contributions of the Res2Net backbone, the Efficient Multi-scale Attention (EMA) module, and the Simple Attention Module (SimAM). The baseline model (first row) corresponds to the original UFLD-V2 framework without any modifications.

From the results, it is evident that each module contributes positively to the overall performance. When introduced individually, all three components lead to improvements in either accuracy or robustness. Among the single-module variants, SimAM provides the most significant gains, reducing both the false positive rate (FP) and false negative rate (FN) from 2.94% and 4.85% to 2.75% and 4.62%, respectively.

The integration of Res2Net and EMA shows clear complementary benefits—enhancing both multi-scale feature extraction and contextual attention modeling.

The best performance is achieved when all three modules are integrated (last row), resulting in an accuracy of 95.95%, FP rate of 2.62%, FN rate of 2.33%, and an F1 score of 96.52%.

These results confirm the effectiveness of the proposed integration strategy: the combination of a multi-scale backbone design, attention-based enhancement, and spatial noise suppression significantly improves both precision and robustness in lane detection.

To further evaluate the computational feasibility of our method, we report its model size, FLOPs, and inference speed in *table 2.* Despite a significantly higher parameter count due to the deeper backbone and attention modules, our model maintains real-time performance (235 FPS), demonstrating its suitability for deployment in autonomous driving systems.

Table 2. Model Complexity and Inference Efficiency

Model	Backbone	Params (M)	FLOPs (G)	FPS	Notes
UFLD-V2 (Baseline)	ResNet-18	10.8	13.6	260	As reported in [3]
Ours (Full Model)	Res2Net-50	~200.0	~35–50	235	Estimated from p[th] and testing

(Note: Parameter counts vary with backbone size. UFLD-V2 uses ResNet-18, while the proposed model adopts Res2Net-50 along with EMA and SimAM modules. FLOPs are approximate estimates based on architectural analysis (input size 320×800). FPS is measured on an NVIDIA RTX 2080Ti using batch size = 1).

4.6. Benchmark Comparison

Table 3 compares the improved model against baseline and state-of-the-art methods. Our model consistently outperforms others in all four-evaluation metrics.

Table 3. Comparative performance with existing methods on TuSimple test set

Model	Accuracy (%)	FP Rate (%)	FN Rate (%)	F1 Score (%)
UFLD	95.82	19.05	3.92	87.87
PolyLaneNet	93.36	9.42	9.33	90.63
EL-GAN	94.90	4.14	3.36	96.26
SAD	95.64	60.20	20.50	95.92
CurveFormer++	95.81	2.75	3.12	96.30
BEV-LaneDet	95.38	3.56	3.78	95.47
PersFormer	95.69	3.22	2.99	96.08
Improved UFLD-V2	**95.95↑**	**2.62↓**	**2.33↓**	**96.52↑**

(Note: Metrics for Improved UFLD-V2 are averaged over three runs)

As shown in *table 2*, the proposed Improved UFLD-V2 achieves the best overall performance across all evaluation metrics. It obtains the highest accuracy of 95.95%, outperforming the weakest baseline, PolyLaneNet (93.36%), by 2.59 percentage points.

In terms of false positive rate, our model achieves the lowest value of 2.62%, which is significantly lower than that of EL-GAN (4.14%) and dramatically lower than SAD (60.20%), indicating enhanced capability in suppressing erroneous detections.

For false negative rate, the proposed model also performs best with only 2.33%, improving upon the next-best EL-GAN (3.36%) by 1.03 percentage points and outperforming SAD (20.50%) by a wide margin.

The overall detection quality is reflected in the F1 score, where the Improved UFLD-V2 achieves 96.52%, slightly surpassing EL-GAN (96.26%) and CurveFormer++ (96.30%).

Compared with recent transformer-based approaches such as CurveFormer++, BEV-LaneDet, and PersFormer, our method consistently achieves superior performance across all four metrics. Specifically, it improves accuracy by up to 0.32 percentage points, reduces FP and FN rates by up to 0.97% and 0.54%, respectively, and enhances F1 score by up to 1.45 percentage points.

These results demonstrate that the proposed method not only achieves state-of-the-art detection accuracy, but also offers greater robustness and reliability across diverse scenarios, making it well-suited for practical deployment in autonomous driving systems.

4.7. Inference Speed Evaluation

To evaluate the real-time performance of our method, we compare the inference speed (FPS) of the proposed Enhanced UFLD-V2 with several representative lane detection models. For fairness, all FPS values are measured or estimated under a unified hardware setting equivalent to RTX 2080Ti.

As shown in *table 4*, our model achieves an inference speed of 235 FPS, which is slightly lower than the original UFLD-V2 (260 FPS) but significantly higher than other transformer-based methods such as CurveFormer++ (110 FPS). This demonstrates that the proposed improvements—including Res2Net, EMA, and SimAM—offer substantial accuracy gains without sacrificing real-time inference capability. The complete comparison is shown below;

Table 4. Inference speed (FPS) comparison of different lane detection models (unified on RTX 2080Ti)

Model	FPS
UFLD	326
UFLD-V2	260
PolyLaneNet	115
CurveFormer++	110
Enhanced UFLD-V2	235

(Note: FPS values are either measured directly or scaled from published data based on performance-equivalent conversions to RTX 2080Ti)

4.8. Qualitative Visualization

To assess the robustness and generalization of the proposed method beyond the TuSimple dataset, we performed qualitative testing on additional challenging scenes from the CULane and BDD100K datasets.

Figure 9 presents the model's visual outputs in five typical driving conditions:
(a) Normal: standard daylight from TuSimple
(b) Occlusion: vehicle-blocked lanes from TuSimple
(c) Hlight and
(d) Shadow: strong illumination and contrast from CULane
(e) Night: low-light driving scenes from BDD100K

As observed, the proposed model consistently achieves robust and accurate lane detection, maintaining continuity and reducing false positives even under adverse visual disturbances. These results provide strong visual evidence of the model's generalization ability across diverse urban and illumination conditions. *Figure 9* Visualization of detection results under different conditions: (a) Normal, (b) Occlusion, (c) Road color variation.

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

Figure 9. Qualitative lane detection results across five representative scenarios from three datasets: *(a)* Normal, *(b)* Occlusion (TuSimple), *(c)* High Light (CULane), *(d)* Shadow (CULane), and *(e)* Night (BDD100K). Each row displays three sample outputs produced by the proposed method under the given condition. The results demonstrate the model's ability to generalize across varying road structures, lighting conditions, and occlusion patterns.

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 277-286| e-ISSN: 2347-470X

5. CONCLUSION AND FUTURE WORKS

This paper presents an enhanced lane detection framework based on the UFLD-V2 architecture, incorporating Res2Net for multi-scale feature extraction, the EMA attention module for robust feature representation in complex scenarios, and the SimAM module for refined lane boundary segmentation. Experimental results on the TuSimple dataset validate the effectiveness of the proposed approach, achieving an accuracy of 0.95947, a false positive rate of 0.0262, a false negative rate of 0.02328, and an F1 score of 0.96517, thereby outperforming the original UFLD-V2 and several state-of-the-art baselines.

The proposed method achieves a favorable balance between detection robustness and real-time inference efficiency. Its lightweight design and high FPS performance demonstrate that the model is well-suited for deployment in practical autonomous driving systems, where both accuracy and speed are essential for reliable operation in diverse driving conditions.

Future research will focus on further enhancing the model's efficiency, adaptability, and generalization capabilities in real-world applications. This includes optimizing inference through network compression and pruning, incorporating domain adaptation to improve cross-scenario robustness, integrating multimodal sensor inputs (e.g., LiDAR, GPS) for more reliable environmental perception, exploring advanced attention mechanisms for spatiotemporal modeling, and leveraging self-supervised learning to reduce reliance on large-scale labeled datasets.

REFERENCES

[1] Ni, J., et al., *A survey on theories and applications for self-driving cars based on deep learning methods.* Applied Sciences (Switzerland), 2020. 10(8).

[2] Rui, s. and c. hui, *Lane detection algorithm based on geometric matrix sampling.* Chinese Science: Information Science, 2017(47(04)): p. 455-467.

[3] Qin, Z., H. Wang, and X. Li. *Ultra Fast Structure-Aware Deep Lane Detection*. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2020.

[4] Qin, Z., P. Zhang, and X. Li, *Ultra Fast Deep Lane Detection With Hybrid Anchor Driven Ordinal Classification.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024. 46(5): p. 2555-2568.

[5] Angayarkanni, S.A., R. Sivakumar, and Y.V. Ramana Rao, *A review on traffic congestion detection methodologies and tools.* International Journal of Advanced Science and Technology, 2019. 28(16): p. 1400-1414.

[6] Sohail, A., et al., *Data-driven approaches for road safety: A comprehensive systematic literature review.* Safety Science, 2023. 158.

[7] Mamun, A.A., et al., *A Comprehensive Review on Lane Marking Detection Using Deep Neural Networks.* Sensors, 2022. 22(19).

[8] Yang, D., W. Bao, and K. Zheng, *Lane Detection of Smart Car based on Deep Learning.* Journal of Physics: Conference Series, 2021. 1873(1): p. 012068.

[9] Narote, S.P., et al., *A review of recent advances in lane detection and departure warning system.* Pattern Recognition, 2018. 73: p. 216-234.

[10] Ma, C.X., et al., *A Real-time Semantic Segmentation Model for Lane Detection.* Journal of Network Intelligence, 2024. 9(4): p. 2234-2257.

[11] Yoo, J.H., et al., *A robust lane detection method based on vanishing point estimation using the relevance of line segments.* IEEE Transactions on Intelligent Transportation Systems, 2017. 18(12): p. 3254-3266.

[12] Wang, J., et al. *Lane boundary detection based on parabola model*. in *2010 IEEE International Conference on Information and Automation, ICIA 2010*. 2010.

[13] Wang, J. and X. An. *A multi-step curved lane detection algorithm, based on hyperbola-pair model*. in *2010 IEEE International Conference on Automation and Logistics, ICAL 2010*. 2010.

[14] Huang, A.S., et al., *Finding multiple lanes in urban road networks with vision and lidar.* Autonomous Robots, 2009. 26(2-3): p. 103-122.

[15] Neven, D., et al. *Towards End-to-End Lane Detection: An Instance Segmentation Approach*. in *IEEE Intelligent Vehicles Symposium, Proceedings*. 2018.

[16] Ghafoorian, M., et al. *EL-GAN: Embedding loss driven generative adversarial networks for lane detection*. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2019.

[17] Tabelini, L., et al. *PolyLaneNet: Lane estimation via deep polynomial regression*. in *Proceedings - International Conference on Pattern Recognition*. 2020.

[18] Han, L., et al. *Lane Detection Method Based on MCA-UFLD*. in *2023 IEEE 8th International Conference on Intelligent Transportation Engineering (ICITE)*. 2023.

[19] Chen, S., & Zhang, Y. (2024, August). Lane detection algorithm based on improved UFLD. In Fourth International Conference on Image Processing and Intelligent Control (IPIC 2024) (Vol. 13250, pp. 748-755). SPIE.

[20] Gao, S.H., et al., *Res2Net: A New Multi-Scale Backbone Architecture.* IEEE transactions on pattern analysis and machine intelligence, 2021. 43(2): p. 652-662.