

DCRnet: An Effective 2D and 3D Lane Detection Method On Highway

Long Zhao^{1†}, Xiaoye Liu^{2†}, Linxiang Li^{3*}

¹College of Information Engineering, East University of Heilongjiang,
Harbin, 150066, China.

²College of Economics and Trade, East University of Heilongjiang,
Harbin, 150066, China.

³*Software Center, Suzhou Automotive Research Institute, Tsinghua
University, Suzhou, 215152, China.

*Corresponding author(s). E-mail(s): lilinxiang_ysu@163.com;

Contributing authors: zhaolong@nefu.edu.cn; lxy2022@hljeu.edu.cn;

†These authors contributed equally to this work.

Abstract

Lane detection is a crucial component in autonomous driving systems, yet challenges persist in achieving accurate detection and seamless deployment of 3D lane detection on highways. To address these issues, we propose a novel striped lane representation that more closely mirrors the real-world characteristics of lane lines. Additionally, we introduce a double-branch cross-layer refinement network, designed to enhance model robustness while accelerating training convergence. To further enhance detection performance, we develop the stripes IOU loss function, specifically tailored for evaluating striped lane representations. To simplify deployment, we implement a prior-based spatial projection correction mechanism, effectively mapping 2D detection results into 3D space. Our algorithm achieves a 97% accuracy on the TuSimple and an average F1-measure of 79.9 on the CULane, all while maintaining considerable runtime efficiency. These results strongly validate the effectiveness and practicality of our approach.

Keywords: Lane Detection, DCRnet, Striped Lane Representation, Stripes IOU loss, Deep Learning

1 Introduction

Lane detection serves as a fundamental perception module in autonomous driving systems, facilitating precise localization, trajectory planning, and motion control[10, 16, 8, 26]. By accurately identifying and tracking lanes, vehicles can infer their position within the lane, predict the trajectory of surrounding vehicles, and make informed decisions to ensure safe navigation. Now, the lane detection algorithm is widely used in Advanced Driver Assistance System(ADAS), such as Lane Keeping Assist(LKA), which help prevent accidents and enhance whole driving safety[25].

Traditional lane detection algorithms, such as Hough Transform and edge detection methods, often struggle with complex road conditions, including varying lighting, occlusions, and worn-out lane markings. These methods are typically limited by their reliance on hand-crafted features and assumptions about road geometry. Recent approaches leveraging deep learning have shown promise in overcoming these challenges by learning robust feature representations from large datasets. Convolutional Neural Networks (CNNs), for instance, have been widely adopted due to their ability to capture intricate patterns in the data[18]. However, these methods still face difficulties in accurately capturing the real-world morphology of lane lines and ensuring seamless deployment in 3D spaces[30].

To address these issues, we propose a novel striped lane representation that closely aligns with the actual characteristics of lane lines. This innovative representation improves the accuracy of lane detection by providing a more precise description of lane morphology. Additionally, we introduce a Double-branch Cross-layer Refinement Network (DCRnet) designed to enhance model robustness and accelerate training convergence. The proposed StripesIOU loss function serves as a superior evaluation metric, specifically tailored to the striped lane representation, ensuring more accurate assessment of detection performance.

One of the critical challenges in lane detection is the transition from 2D to 3D space. The complexity of 3D lane detection arises from the need to accurately map lane markings detected in 2D images onto a 3D model of the road. This process is often hampered by inaccuracies in the initial 2D detection and the lack of robust projection mechanisms. To simplify the deployment of 3D lane detection schemes, we employ a prior-based spatial projection correction mechanism. This mechanism effectively maps 2D detection results into 3D space, facilitating downstream task execution and reducing deployment complexity. By leveraging prior knowledge about the road geometry, our method enhances the accuracy and reliability of the 3D lane projections[19].

Moreover, the proposed lane projection correction mechanism, based on a priori information, is corroborated through comparisons with high-definition maps (HD maps). This comparison highlights the precision and reliability of our method in projecting 2D lane detections into 3D space, providing a robust foundation for subsequent autonomous driving tasks such as vehicle navigation and collision avoidance.

Recent literature supports the need for more accurate and deployable lane detection methods. For instance, UFLD[20] highlight the challenges in detecting lane markings under varying conditions and emphasize the potential of deep learning approaches to address these issues. Similarly, ONCE3D[30] discuss the importance of 3D lane

detection for autonomous driving and the inherent difficulties in deploying these systems. Our work builds on these foundational studies, addressing the identified gaps and presenting a novel solution to enhance lane detection accuracy and deployment feasibility.

Our algorithm demonstrates exceptional performance, achieving 97% accuracy on the TuSimple dataset and an average F1-measure of 79.9 on the CULane dataset, while maintaining substantial runtime efficiency. These results underscore the effectiveness of our approach in real-world scenarios. The robustness of our model is further validated through extensive testing on proprietary datasets, which encompass a diverse range of road conditions and environmental factors. The focus of our work can be summarized as follows:

- We propose an innovative striped lane line representation that more accurately reflects the true morphology of lane lines. This new representation enhances the precision of lane detection by providing a detailed and realistic description of lane markings.
- A dual-branch cross layer refinement network is introduced, designed to improve the robustness of the network while accelerating the training convergence speed. This architecture leverages the strengths of both branches to enhance overall performance and stability.
- We develop the StripesIOU loss function, specifically tailored for the striped lane line representation. This loss function, combined with a structural loss function, provides a comprehensive evaluation metric, significantly improving detection accuracy.
- To reduce the complexity of deploying the algorithm in 3D space, we propose a lane projection correction mechanism based on prior information. This strategy effectively maps 2D detection results into 3D space, facilitating easier implementation.
- The proposed algorithm is rigorously tested on public datasets, achieving a 97% accuracy rate on the TuSimple dataset and an average F1 score of 79.9 on the CULane dataset. These results are achieved while maintaining a high inference speed, demonstrating the effectiveness and efficiency of our approach.

2 Related Work

2D lane detection. LaneNet[27] is a multi-task model that divides the instance segmentation task into “semantic segmentation” and “vector representation of pixels.” It then clusters the results from these two branches to obtain instance segmentation results. To differentiate between lanes, SCNN[17] introduces a sliced CNN that allows information to be passed across rows and columns, treating different lanes as distinct classes and transforming lane detection into a multi-class segmentation task. HybridNets[23] proposes efficient segmentation headers and box/class prediction networks based on weighted bi-directional feature networks, which can be weighted with automatically customizable anchors at each level within the bi-directional feature network. CurveLane-NAS[29] integrates adaptive point blending and architecture search to improve the precision of lane marking extraction in complex environments. This approach involves a tailored neural architecture search specifically designed for lane

detection tasks, which identifies optimal network structures for different road types and conditions. Moreover, it combines this with adaptive point blending to better handle curves and complex lane geometries, enhancing overall detection performance.

Similarly, CLRNet[32] employs cross-layer refinement networks to enhance feature representation and boundary detection, demonstrating superior performance in challenging scenarios. This model utilizes a multi-scale feature extraction strategy where information from different layers of the network is refined and combined, leading to more precise lane boundary predictions. CLRNet’s approach to cross-layer interaction allows the model to better capture fine details at various scales, crucial for accurate lane detection in varied environments.

UFAST[20] uses row anchors to divide lanes into patches, speeding up computation. UFAST-v2[21] employs a series of hybrid anchors that use sparse coordinates to represent lanes. This anchor-driven representation treats the lane detection task as an ordered classification problem to determine lane coordinates. SGNet[22] introduces a top-down vanishing-point-guided anchoring mechanism that generates powerful anchors to efficiently capture a wide variety of lanes. O2SFormer[34] proposes a one-to-many label assignment method, combining one-to-one and one-to-many label assignments to improve training efficiency while maintaining end-to-end detection. To address optimization challenges in one-to-one allocation, hierarchical soft labeling is introduced to enhance results by adjusting the positive weights of lane anchors across different decoder layers.

3D lane detection. 3D-LaneNet[5] introduces two innovative concepts: intra-network inverse perspective mapping and anchor-based lane representation. Intra-network IPM projection facilitates the flow of dual representation information between the regular image view and the top view. The anchor-per-column output representation enables an end-to-end approach, replacing common heuristics for lane estimation with an object detection framework. 3D-LaneNet+[4] proposes a camera-based DNN method for 3D lane detection with uncertainty estimation. This method is based on a semi-local Bird’s Eye View (BEV) grid representation, decomposing lanes into simple lane segments. PersFormer[2] is an end-to-end monocular 3D lane detector with a Transformer-based spatial feature transformation module. This model utilizes camera parameters as a reference and generates BEV features by focusing on relevant local regions of the view. It employs a unified 2D/3D anchor design and includes an auxiliary task to detect 2D/3D lanes simultaneously, enhancing feature consistency. LaneCPP[19] emphasize continuous 3D lane representation and the integration of physical priors to improve detection reliability in dynamic environments (Jin et al., 2023). This approach leverages 3D geometric information to provide a more accurate and robust representation of lanes, which is particularly beneficial for autonomous driving applications.

ONCE-3DLanes[30] proposes an extrinsic-free, anchor-free method that returns the 3D coordinates of lanes in the image view without converting the feature map to BEV. The algorithm projects the image to the top view and regresses the 3D coordinates using a set of predefined anchors without the supervision of manually produced anchors and extrinsic parameters. BEV-LaneDet[24] introduces the concept of a “Virtual Camera,” which unifies the intrinsic/extrinsic parameters of cameras

mounted on different vehicles to ensure consistency in spatial relationships between cameras. Additionally, it proposes a simple yet effective 3D lane representation called keypoint representation, which is more suitable for representing complex and diverse 3D lane structures. PETRv2[15] explores the utilization of information from historical frames for temporal modeling, significantly improving the performance of 3D object detection. It extends the 3D position embedding (3D PE) proposed in PETR[14] for temporal modeling and demonstrates that 3D PE can achieve alignment of spatial positions of objects across different frames.

3 Method

3.1 The Lane Description

Lanes exhibit various forms, but they share certain common characteristics, such as being long and thin. Leveraging this prior knowledge allows the network to learn lane features more effectively. However, traditional object detection methods typically use rectangular bounding boxes to represent objects, which are ill-suited for the elongated and narrow nature of lanes[31]. While some researchers have represented lanes using discrete strings of points or straight lines, these methods are generally effective but fall short when dealing with curved or discontinuous lanes. To address these limitations, we propose the use of line stripes to represent lanes, offering a more accurate fit to their real-world shapes, as shown in Fig.1.

Specifically, lane is expressed as a sequence of stripes, *i.e.*, $\text{Lane}_i = \{S_1, S_2, \dots, S_n\}$, $S_i = (x_{ci}, y_{ci}, \theta_i, l_i)$. In which, x_{ci} and y_{ci} denote the midpoints of the i -th line stripes, θ_i denotes the angle with the x -axis, and l_i denotes half the length of the line stripes.

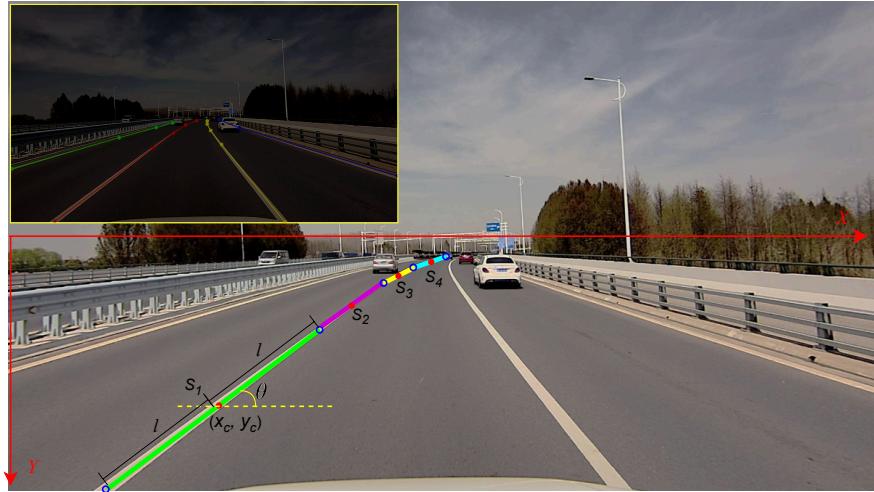


Fig. 1 Illustration of the lane description. As you can see from the ground truth in the upper left, the line stripes are the best choice to fit the real lanes. The different colored lines in the figure represent several stripes of a lane, where each stripe is represented by the coordinates of its center point (x_c, y_c) , the angle to the x -axis θ , and half of its length l .

3.2 The Network Architecture

To achieve fast and accurate lane detection while enhancing the network's robustness, this paper introduces a novel two-branch neural network architecture. The proposed network comprises four main components: the backbone (encoder), feature pyramid network[12], decoder, and multilayer perceptron.

We employ ResNet[7], a widely acclaimed model, as the backbone for feature extraction. To integrate information across various receptive fields, a feature pyramid is constructed using feature maps from three layers of the encoder. The multilayer perceptron, equipped with a cascade refinement mechanism, processes this pyramid to produce precise lane detection results.

To further accelerate network convergence during training and enhance robustness, the decoder is designed with multiple up-sampling layers as a segmentation branch. This design ensures efficient learning and reliable performance under varying conditions. The complete network architecture is illustrated in Fig.2.

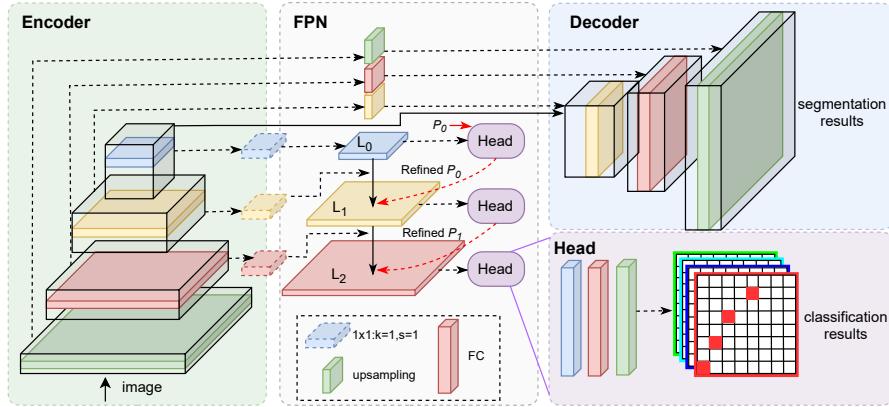


Fig. 2 Illustration of the network architecture proposed in this paper. The entire network architecture is divided into two branches and four components. Both the auxiliary segmentation branch and the classification branch share a backbone for feature extraction. The classification branch comprises a feature pyramid and a multilayer perceptron. The lane parameters undergo cascaded refinement through the three layers of the feature pyramid, ultimately yielding optimized detection results.

3.3 The Cascade Refinement

Deep neural networks are composed of a cascade of feature extraction layers, progressively extending features from low-dimensional to high-dimensional spaces. Low-dimensional features retain rich detail information but fail to adequately express the multifaceted characteristics of an object. Conversely, high-dimensional features capture the complexity and diversity of features but often neglect local contextual information.

The Feature Pyramid Network (FPN) is a deep learning architecture widely used for object detection and semantic segmentation tasks[12]. FPN enhances detection

accuracy and robustness by constructing a feature pyramid capable of detecting targets at different scales. It achieves this through a top-down pathway and lateral connections, which fuse features from various levels. This feature fusion combines the detailed information from low-level features with the high-level semantic information, thereby improving the effectiveness of target detection and segmentation.

Given these capabilities, FPN is particularly effective for lane detection. By fully utilizing features from different dimensions, we can perform overall lane detection using high-dimensional feature maps and subsequently refine the results with low-dimensional feature maps. This approach leverages both the comprehensive understanding of the scene provided by high-dimensional features and the fine details captured by low-dimensional features, ensuring accurate and robust lane detection.

We take ResNet as the backbone to extract the features, and use $\{L_0, L_1, L_2\}$ to denote the feature maps generated by FPN in three different layers, as shown in Fig2. In this paper, the cascade refinement mechanism is gradually extended from the high-dimensional L_0 layer to the low-dimensional L_2 , and $R(\cdot)$ represents the refinement[31]:

$$S_i = S_{i-1} \otimes R_i(L_{i-1}, S_{i-1}). \quad (1)$$

where S_i is the learnable parameter of lane ($S_i = (x_{ci}, y_{ci}, \theta_i, l_i)$), S_0 is an initialization parameter which is uniformly sampled in the image, and \otimes denotes the calibration operation.

3.4 Stripes IoU Loss

In contemporary research, various methods are employed to evaluate lane detection results. However, the global regularity and local diversity inherent in lane lines pose significant challenges for accurate evaluation. In anchor-based schemes, results are typically evaluated based on the relationship between predicted points and preset anchor points. In segmentation-based schemes, the evaluation relies on the recall and accuracy of pixel points.

In this paper, we draw upon the advantages of both approaches and integrate them with the specific characteristics of lane lines. We propose a novel stripes IoU loss method that effectively captures the unique features of lane lines, ensuring a more accurate and robust evaluation.

In our approach, the predicted stripe ${}^P S_i = ({}^P x_{ci}, {}^P y_{ci}, \theta_i, l_i)$, where $i \in (0, H)$ and H is the number of predicted stripes, is sampled at equal distances and treated as predicted points $({}^P x_{ci_j}, {}^P y_{ci_j})$, where $j \in (0, N)$ and N is the number of samples. These predicted points $({}^P x_{ci_j}, {}^P y_{ci_j})$ serve as the centers of circles used to identify ground truth points $({}^g x_k, {}^g y_k)$ within a threshold range r for matching calculations, as illustrated in Fig.3. It is worth noting that during preprocessing, the equidistant sampling length of the ground truth points is set to half of the equidistant sampling length of the predicted stripes. This approach increases the density of the ground truth points and reduces evaluation error. The evaluation of results is ultimately based on the total number of correctly matched points:

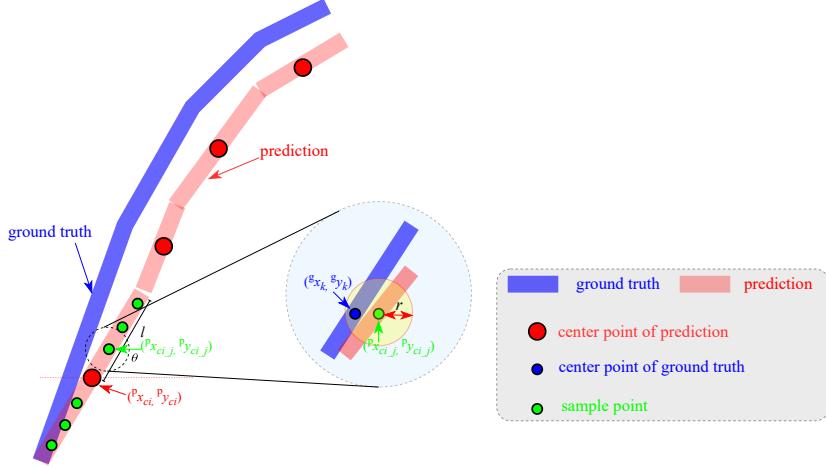


Fig. 3 Illustration of the stripe IoU loss.

$$StripesIoU = \sum_{i=0}^H \frac{N_{PG}}{N_P + N_G - N_{PG}}, \quad (2)$$

$$N_{PG} = \sum_{j=0}^N Match({}^P P_{ci_j}, r). \quad (3)$$

where N_P and N_G represent the number of prediction points and the number of ground truth points, respectively, and N_{PG} represents the number of valid prediction points, i.e., the number of points with successful matching. $Match(\cdot)$ is denoted as matching with the ground truth point in a circle centered at ${}^P P_{ci_j}$ (${}^P P_{ci_j} = ({}^P x_{i_j}, {}^P y_{i_j})$) with radius r .

Then, the Stripes IoU loss l_{siou} is defined as:

$$l_{siou} = 1 - StripesIoU. \quad (4)$$

For lanes with pronounced structural features, we enhance the structural loss function by incorporating the Structural Similarity Index Measure (SSIM) loss[9]. The SSIM loss function is designed to assess the structural information within an image, focusing on local regions to quantify the loss in structural similarity. This approach ensures that the evaluation captures detailed structural variations, providing a more accurate measure of similarity between the predicted and ground truth lane structures. The SSIM loss function is formally defined as follows:

$$l_{ssim} = 1 - \frac{(2\mu_x\mu_y + C_1)(\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \quad (5)$$

where μ_x and μ_y represent the mean of the predicted result and ground truth, respectively, and σ_{xy} is their covariance. C_1 and C_2 are hyperparameters, $C_1 = 0.01^2$, $C_2 = 0.03^2$.

3.5 2D to 3D Lane Projection

By inputting the image captured by the camera into the proposed model, we can obtain the pixel coordinates of the lanes in the image. However, due to the perspective mapping during the imaging process, the obtained lane detection results lack depth information, making it difficult to infer their specific position in the real world. Consequently, these results cannot be directly applied to downstream path planning tasks.

On highways, there are two priori informations: road is flat and fixed lane width. Utilizing the assumption of a flat road, we can roughly infer the 3D spatial location of the lane using pinhole imaging principles. Additionally, by considering the fixed lane width, we can correct the estimated scale factor to further optimize the 2D to 3D projection results and enhance accuracy.

Assuming we have the intrinsic matrix M of the camera, the extrinsic matrix T relative to the vehicle, and the camera height h . Given the model's output, the pixel coordinate v of the lane (where the lane height is 0) can be determined in the image coordinate system. Based on the principles shown in Fig 4, and assuming the road is flat and the camera is not rotated, the scale factor s_0 can be calculated using the principles of pinhole imaging and the normalized imaging plane.

$$s_0 = \frac{h}{v} = \frac{z}{1}. \quad (6)$$

in which z is the distance(depth) of the lane point.

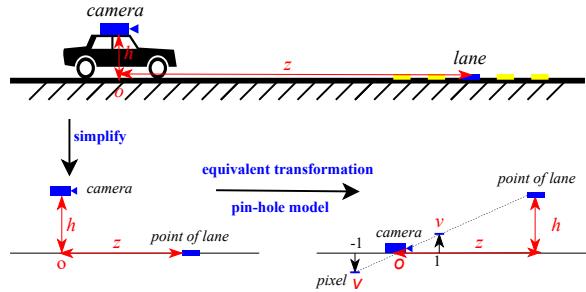


Fig. 4 Illustration of the projection of a 2D lane into 3D space. Based on the assumptions that the ground is flat and the lane is attached to the ground (height is 0), and according to the principle of pinhole imaging, we can derive the scaling factor of perspective mapping by calculating the ratio of the mounting height h of the camera to the imaging height v of the pixel point of the lane. This allows us to obtain the 3D spatial distance (depth) of the lane.

The points in the vehicle coordinate system is transformed to pixels in the image through the following process[1]:

$$P_{\text{img}} = s \cdot M \cdot T \cdot P_{\text{car}}. \quad (7)$$

in which,

$$P_{\text{img}} = [u, v, 1]^T, P_{\text{car}} = [X, Y, Z, 1]^T, T = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ \mathbf{0} & 1 \end{bmatrix}, M = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

where s is scale factor, P_{img} represents points in the image coordinate system, P_{car} represents points in the vehicle coordinate system, T represents the extrinsic matrix of the camera with respect to the vehicle, and M represents the intrinsic matrix of the camera.

The scale factor optimization process is illustrated in Fig 5. Assuming that the width of the lane in the image coordinate system is X , our country stipulates that the lane width of highway is $w_0 = 3.75$ meters. The lane width deduced from the scale factor s_0 obtained is w_1 . Therefore, the optimized scale factor s_1 is:

$$\frac{X}{s_0} = w_1, \frac{X}{s_1} = w_0. \Rightarrow s_1 = s_0 \cdot \frac{w_1}{w_0}. \quad (8)$$

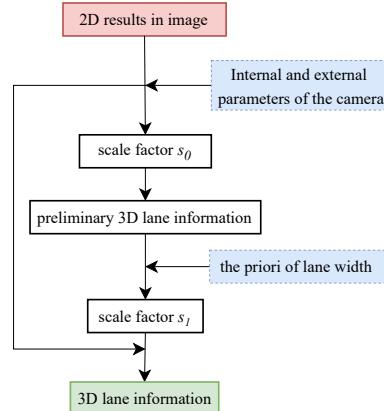


Fig. 5 Illustration of the scale factor optimization process.

4 Experiments

In this section, we illustrate the effectiveness of the proposed method through extensive experiments, which are primarily divided into the following aspects:

- Experimental Setting: We provide details on the experimental setup, including dataset details, evaluation metrics, and computer configuration.
- Performance Comparison with Other Algorithms: We compare the performance of our method with that of existing algorithms, demonstrating its advantages and capabilities.
- Presentation of 2D and 3D Results: We present the results of both 2D and 3D lane detection, demonstrating the accuracy and robustness of our approach across various scenarios.

4.1 Experimental setting

Datasets and evaluation metrics. In the field of lane detection in autonomous driving, CULane and TuSimple are two widely used datasets. CULane is collected in multiple complex scenes, including narrow, fork, rainy, etc., with larger volume. While TuSimple is an ideal dataset, which mainly collects highways image data with stable light. The detailed information about the datasets can be seen in Table 1.

Table 1 Datasets details

Dataset	Size	Train	Validation	Test	Scenarios	Metrics
CULane	1640 × 590	88,880	9,675	34,680	highway	F1-measure
TuSimple	1280 × 720	3,268	358	2,782	urban&highway	accuracy

The difference in the way TuSimple and CULane label their data leads to differences to evaluate their results. For the TuSimple dataset, the main evaluation metric is accuracy, which is calculated as follows:

$$accuracy = \frac{\sum_{clip} P_{clip}}{\sum_{clip} G_{clip}}. \quad (9)$$

in which P_{clip} represents the count of correctly predicted lane points, while G_{clip} represents the total number of ground truth lane points in each clip.

As for the evaluation metric of CULane, Each lane is labeled as a banded area with a width of 30 pixels, and for this area-type object, the intersection-over-union (IOU) between the ground truth and the prediction is generally used as evaluation metric, in which the IOU larger than 0.5 is considered to be the correct result, and the evaluation metrics are calculated by the following formula:

$$\left\{ \begin{array}{l} Precision = \frac{TP}{TP + FP} \\ Recall = \frac{TP}{TP + FN} \\ F1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \end{array} \right. . \quad (10)$$

where TP is the true positive (classifier predicts positive, and actually positive), FP is the false positive (classifier predicts positive, but actually negative), and FN is the false negative (classifier predicts negative, but actually positive).

The software and hardware used for the experiments in this paper as well as the parameter configurations for training are shown in Tab2.

Table 2 experimental configuration.

Hardware & Software		Training parameters	
Categories	Parameters	Categories	Parameters
CPU	Intel Core i7 5930K	Optimizer	Adam
RAM	64G	Initial learning rate	4e-4
GPU	Nvidia GTX 1080Ti	Learning rate strategy	Cosine decay
Graphics memory	12G	Batch size	32
Operating System	Ubuntu20.04	Epochs of TuSimple	100
Framework	Pytorch1.9.0	Epochs of CULane	50
CUDA	11.1		

4.2 Results of public datasets

In this section, we compare the performance of the algorithms proposed in this paper on the TuSimple and CULane datasets with the existing algorithms. For the TuSimple dataset, we obtain relevant results from the benchmark for comparison. The main algorithms include Res18-Seg[3], Res34-Seg[3], LaneNet[27], EL-GAN[6], SCNN[17], Res34-UFLD-v1[20], Res18-UFLD-v1[20], SAD[11], Res18-CondLaneNet[13], Res34-CondLaneNet[13], Res18-CLRNet[31] and Lane2Seq[33]. We utilize accuracy and inference runtime per image as evaluation metrics, and the results are presented in Table 3.

Table 3 Comparison with other methods on TuSimple test set.

Method	Accuracy	Runtime(ms)	Multiple
Res18-Seg	92.69	25.3	5.3x
Res34-Seg	92.84	50.5	2.6x
LaneNet	96.38	19.0	7.0x
EL-GAN	96.39	>100	<1.3x
SCNN	96.53	133.5	1.0x
SAD	96.64	13.4	10.0x
Res34-UFLD-v1	96.06	5.9	22.6x
Res18-UFLD-v1	95.87	3.2	41.7x
Res18-ConLanNet	94.43	4.6	29.7x
Res34-ConLanNet	95.37	5.1	26.2x
Res18-CLRNet	96.84	7.8	17.1x
Res34-CLRNet	96.87	7.9	16.9x
Lane2Seq	96.72	14.2	9.4x
Res18-Ours	96.9	6.4	20.9x
Res34-Ours	97.0	6.5	20.5x

As can be seen in Tab3, our algorithm presents a comparative analysis of various lane detection methods evaluated on the TuSimple. Our proposed method achieves the highest accuracy at 97.0%, surpassing all other methods. Notably, it maintains a competitive runtime of 6.5 ms and achieves a $20.5x$ multiple, indicating a balanced trade-off between accuracy and efficiency.

For the CULane dataset, the methods include SCNN[17], Res34-SAD[11], SAD[11], Res18-UFLD-v1[20], Res34-UFLD-v1[20], O2SFormer[35], Res18-CondLaneNet[13], Res34-CondLaneNet[13], ADNet[28], Res18-CLRNet[31] and Lane2Seq[33]. We use the F1-measure to evaluate the performance. The CULane dataset includes 9 common scenarios, such as sunny, night, and curve, etc., and we compute the average for all scenarios for convenience, and the results are shown in Tab4.

Table 4 Comparison with other methods on CULane test set.

Method	Average F1-measure(%)	Runtime(ms)	Multiple
SCNN	71.6	133.5	1.0x
Res34-SAD	70.7	50.5	2.6x
SAD	70.8	13.4	10.0x
Res18-UFLD-v1	68.4	3.1	43.0x
Res34-UFLD-v1	72.3	5.7	23.4x
O2SFormer	77.8	4.7	28.4x
Res18-CondLaneNet	78.7	4.6	29.0x
Res34-CondLaneNet	79.3	5.1	26.2x
ADNet	78.9	5.3	25.2x
Res18-CLRNet	79.6	7.8	17.1x
Res34-CLRNet	79.7	7.9	16.9x
Lane2Seq	79.2	14.2	9.4x
Res18-Ours	79.7	6.4	20.9x
Res34-Ours	79.9	6.5	20.5x

As shown in Table 4, on the CULane dataset, Our proposed methods, Res34-Ours achieve the highest average F1-measure at 79.9%, respectively, while maintaining competitive runtimes of 6.5 ms , resulting in multiples of $20.5x$, demonstrating their superior effectiveness and efficiency in lane detection tasks on the CULane dataset. This conclusion is consistent with that on TuSimple.

4.3 Results of our dataset

In this section, we introduce our dataset and showcase the detection results obtained by the algorithm proposed in this paper on images, along with the 3D results obtained after mapping 2D to 3D projection.

Our dataset. We curated a dataset comprising approximately 20,000 images of flat road sections, encompassing highways, urban expressways, and partially flat urban roads, where all motorized lanes maintain a consistent width of 3.75 meters. These images were captured under diverse weather conditions, including sunny and cloudy environments, to ensure robustness in various scenarios. The dataset adheres to the TuSimple format, with each image annotated for up to four lanes: the current lane

and adjacent left and right lanes, depending on their presence. Lane markings were annotated up to the vanishing point or until they became indistinguishable.

2D results. During testing, it was observed that most lane detection algorithms exhibited satisfactory performance during normal lane-keeping phases. However, when the vehicle underwent a lane change, the algorithm failed to detect the lanes accurately. This issue stemmed from the limited amount of training data related to vehicle lane changes, leading to a “long tail” phenomenon. Consequently, the model developed a bias towards specific local features during the training process, focusing solely on a particular region. Take UFLD as an example, although UFLD incorporates an auxiliary segmentation branch, it adopts a 5-class segmentation approach that does not treat lanes as a unified feature. As a result, the segmentation fails to accurately capture lane changes. The shortcomings of UFLD in handling lane changes are depicted in Fig 6.

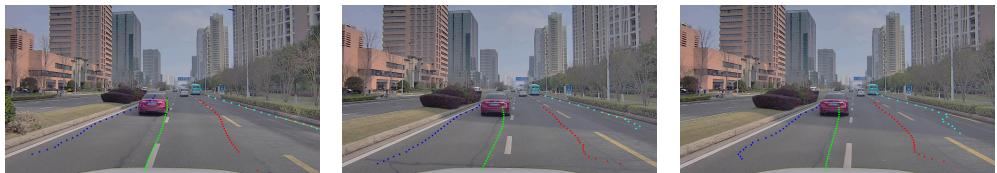


Fig. 6 Illustration of the UFLD fails in the case of the lane change. In the image, you can see that the vehicle is shifting gradually to the left. The detected lanes are gradually deviated until finally it is completely confused.

In this paper, we treat all lane lines as an uniform category, without differentiating between their types or considering their relative positional relationships. Thereby achieving satisfactory lane detection results. The outcomes are illustrated in Fig7. From left to right, the sequence includes the original image, ground truth, classification result, and segmentation result. Different colors are employed in the classification result to denote different lanes, whereas black uniformly represents lanes in the segmentation result, with white indicating the background.

In addition to the standard parallel lanes on the road, there are intersections where the lanes diverge to form a fork (one lane extends to become two), or where two lanes merge to become one, as illustrated in Fig8. Common scenarios also include lanes being obscured by vehicles, as depicted in Fig9, or blurred due to surface dirt, as shown in Fig10.

3D results. To validate the effectiveness of the proposed 2D to 3D spatial lane projection scheme, we conducted validation tests on a flat road using HD map where the lane width is 3.75 meters. As depicted in Fig11, the upper image illustrates the camera observation, while the lower image showcases the lane boundary line (white straight lines) and the center line (green straight lines) of the HD map. The four slightly thicker colored lines (red, yellow, blue, and indigo) represent the recognized lanes, respectively.

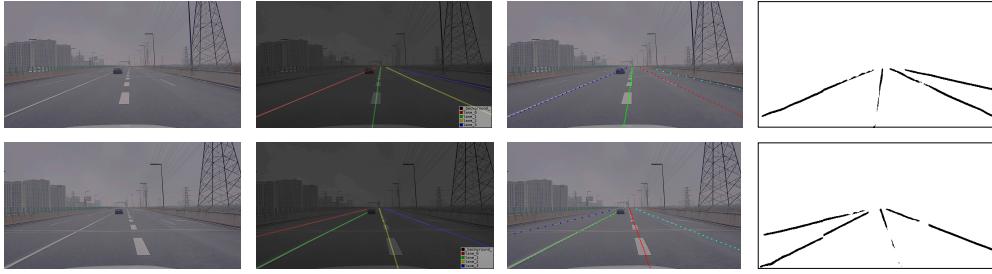


Fig. 7 The lane detection algorithm successfully recognizes lanes even when vehicles change lanes. In the image, the vehicle is gradually shifting to the left, yet the detection results remain accurate.



Fig. 8 The lane detection algorithm effectively handles fork scenarios. In the first row of images, the lane diverges at the intersection, forming two lanes, while in the bottom image, the two lanes merge into one lane on the right.

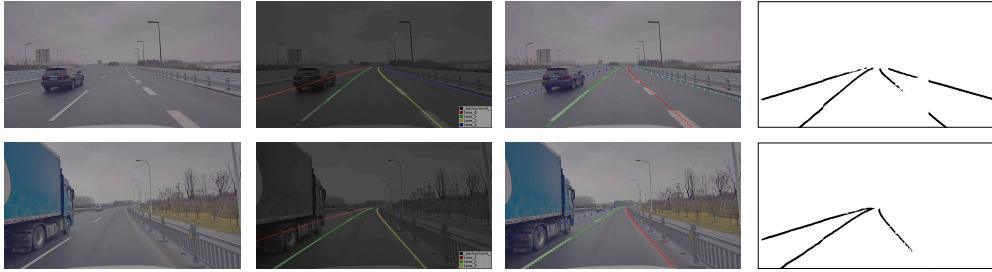


Fig. 9 Lane detection in the presence of vehicle occlusion. In the first image, a car partially obscures part of the lanes, while in the second image, a large truck blocks almost the entire lane.

5 Conclusion

In this study, we present a novel striped lane representation designed to accurately reflect the real morphology of lanes, enhancing their characterization in detection tasks. We introduce a DCRnet, which significantly improves training convergence speed and network robustness. This novel representation employs a more precise



Fig. 10 Lane detection in the case of blurred lanes. Both sides of the road often accumulate a large amount of sludge due to stagnant water, resulting in poorly defined lanes. The lane on the right is particularly unclear due to the heavy dirt covering it.

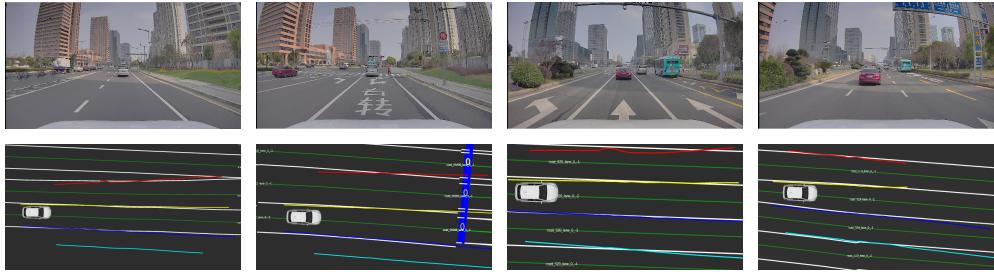


Fig. 11 The 2D lane detection results are projected into 3D space and compared with a HD map. In the visualization (Fig11), the white and green straight lines correspond to the lane boundary lines and center lines of the HD map, respectively. The red, yellow, blue, and indigo represent the four recognized lanes, respectively. It's worth noting that since there are non-motorized lanes in the two images on the left, they do not show in the HD map. However, our algorithm does not differentiate between the types of lanes and outputs all the detected results.

stripes IOU loss function, combined with structural loss, to rigorously evaluate detection performance. Our algorithm demonstrates exceptional performance, achieving 97% accuracy on the Tusimple dataset and an average F1 score of 79.9 on the CULane dataset, all while maintaining competitive inference speeds. The robustness of our approach is further validated through tests on proprietary datasets. Additionally, to mitigate the complexity associated with deploying 3D lane detection systems, we propose a lane projection correction mechanism based on a priori information, facilitating the projection of 2D detection results into 3D space. This advancement supports seamless execution of downstream tasks. The efficacy of the mechanism is substantiated through comparisons with HD maps.

Acknowledgement(s)

We would like to express our gratitude to Dr. Long Zhao for his invaluable guidance and support throughout this study. We also thank East University of Heilongjiang and Suzhou Automotive Research Institute for providing the necessary resources and

funding. Lastly, we appreciate the feedback and encouragement from our colleagues, which greatly contributed to the completion of this research.

Disclosure statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Funding

This work is supported by Key Research and Development Plan Project of Heilongjiang Province under Grant GZ20210163 and Natural Science Foundation of Heilongjiang Province under Grant LH2020F040.

References

- [1] MD Anderson et al. “Consumer response to integrated pest management and certification”. In: *Agriculture, ecosystems environment* 60.2-3 (1996), pp. 97–106.
- [2] Li Chen et al. “Performer: 3d lane detection via perspective transformer and the openlane benchmark”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 550–567.
- [3] Liang-Chieh Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *CoRR* abs/1606.00915 (2016). arXiv: [1606.00915](https://arxiv.org/abs/1606.00915). URL: <http://arxiv.org/abs/1606.00915>.
- [4] Netalee Efrat et al. “3D-LaneNet+: Anchor Free Lane Detection using a Semi-Local Representation”. In: *CoRR* abs/2011.01535 (2020). arXiv: [2011.01535](https://arxiv.org/abs/2011.01535). URL: <https://arxiv.org/abs/2011.01535>.
- [5] Noa Garnett et al. “3D-LaneNet: end-to-end 3D multiple lane detection”. In: *CoRR* abs/1811.10203 (2018). arXiv: [1811.10203](https://arxiv.org/abs/1811.10203). URL: <http://arxiv.org/abs/1811.10203>.
- [6] Mohsen Ghafoorian et al. “EL-GAN: Embedding Loss Driven Generative Adversarial Networks for Lane Detection”. In: *CoRR* abs/1806.05525 (2018). arXiv: [1806.05525](https://arxiv.org/abs/1806.05525). URL: <http://arxiv.org/abs/1806.05525>.
- [7] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: [1512.03385](https://arxiv.org/abs/1512.03385). URL: <http://arxiv.org/abs/1512.03385>.
- [8] Aharon Bar Hillel et al. “Recent progress in road and lane detection: a survey”. In: *Machine Vision & Applications* 25.3 (2014), pp. 727–745.
- [9] Alain Hore and Djemel Ziou. “Image quality metrics: PSNR vs. SSIM”. In: *2010 20th international conference on pattern recognition*. IEEE. 2010, pp. 2366–2369.
- [10] Yuenan Hou et al. “Learning Lightweight Lane Detection CNNs by Self Attention Distillation”. In: *CoRR* abs/1908.00821 (2019). arXiv: [1908.00821](https://arxiv.org/abs/1908.00821). URL: <http://arxiv.org/abs/1908.00821>.

- [11] Yuenan Hou et al. “Learning Lightweight Lane Detection CNNs by Self Attention Distillation”. In: *CoRR* abs/1908.00821 (2019). arXiv: [1908.00821](https://arxiv.org/abs/1908.00821). URL: <http://arxiv.org/abs/1908.00821>.
- [12] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.
- [13] Lizhe Liu et al. “CondLaneNet: a Top-to-down Lane Detection Framework Based on Conditional Convolution”. In: *CoRR* abs/2105.05003 (2021). arXiv: [2105.05003](https://arxiv.org/abs/2105.05003). URL: <https://arxiv.org/abs/2105.05003>.
- [14] Yingfei Liu et al. “PETR: Position Embedding Transformation for Multi-View 3D Object Detection”. In: *ArXiv* abs/2203.05625 (2022). URL: <https://api.semanticscholar.org/CorpusID:247411100>.
- [15] Yingfei Liu et al. “PETRv2: A Unified Framework for 3D Perception from Multi-Camera Images”. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), pp. 3239–3249. URL: <https://api.semanticscholar.org/CorpusID:249375226>.
- [16] Sandipann P. Narote et al. “A review of recent advances in lane detection and departure warning system”. In: *Pattern Recognition* 73 (2018), pp. 216–234. ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2017.08.014>. URL: <https://www.sciencedirect.com/science/ARTICLE/pii/S0031320317303266>.
- [17] Angshuman Parashar et al. “SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks”. In: *CoRR* abs/1708.04485 (2017). arXiv: [1708.04485](https://arxiv.org/abs/1708.04485). URL: <http://arxiv.org/abs/1708.04485>.
- [18] Monirul Islam Pavel, Siok Yee Tan, and Azizi Abdullah. “Vision-based autonomous vehicle systems based on deep learning: A systematic literature review”. In: *Applied Sciences* 12.14 (2022), p. 6831.
- [19] Maximilian Pittner, Joel Janai, and Alexandru P Condurache. “LaneCPP: Continuous 3D Lane Detection using Physical Priors”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 10639–10648.
- [20] Zequn Qin, Huanyu Wang, and Xi Li. “Ultra Fast Structure-aware Deep Lane Detection”. In: *CoRR* abs/2004.11757 (2020). arXiv: [2004.11757](https://arxiv.org/abs/2004.11757). URL: <https://arxiv.org/abs/2004.11757>.
- [21] Zequn Qin, Pengyi Zhang, and Xi Li. “Ultra fast deep lane detection with hybrid anchor driven ordinal classification”. In: *IEEE transactions on pattern analysis and machine intelligence* (2022).
- [22] Jinming Su et al. “Structure guided lane detection”. In: *arXiv preprint arXiv:2105.05403* (2021).
- [23] Dat Vu, Bao Ngo, and Hung Phan. “Hybridnets: End-to-end perception network”. In: *arXiv preprint arXiv:2203.09035* (2022).
- [24] Ruihao Wang et al. “BEV-LaneDet: a Simple and Effective 3D Lane Detection Baseline”. In: *arXiv preprint arXiv:2210.06006* (2022).
- [25] Yue Wang, Dinggang Shen, and Eam Khwang Teoh. “Lane detection using spline model”. In: *Pattern Recognit. Lett.* 21 (2000), pp. 677–689. URL: <https://api.semanticscholar.org/CorpusID:14178297>.

- [26] Yue Wang, Eam Khwang Teoh, and Dinggang Shen. “Lane detection and tracking using B-Snake”. In: *Elsevier Limited* 4 (2004).
- [27] Ze Wang, Weiqiang Ren, and Qiang Qiu. “LaneNet: Real-Time Lane Detection Networks for Autonomous Driving”. In: *CoRR* abs/1807.01726 (2018). arXiv: 1807.01726. URL: <http://arxiv.org/abs/1807.01726>.
- [28] Lingyu Xiao et al. “ADNet: Lane Shape Prediction via Anchor Decomposition”. In: (2023). arXiv: 2308.10481 [cs.CV].
- [29] Hang Xu et al. “Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*. Springer. 2020, pp. 689–704.
- [30] Fan Yan et al. “ONCE-3DLanes: Building Monocular 3D Lane Detection”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 17122–17131. URL: <https://api.semanticscholar.org/CorpusID:248496788>.
- [31] Tu Zheng et al. “CLRNet: Cross Layer Refinement Network for Lane Detection”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 888–897. URL: <https://api.semanticscholar.org/CorpusID:247595006>.
- [32] Tu Zheng et al. “Clrnet: Cross layer refinement network for lane detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 898–907.
- [33] Kunyang Zhou. “Lane2Seq: Towards Unified Lane Detection via Sequence Generation”. In: *ArXiv* abs/2402.17172 (2024). URL: <https://api.semanticscholar.org/CorpusID:268033131>.
- [34] Kunyang Zhou and Rui Zhou. “End to end lane detection with one-to-several transformer”. In: *arXiv preprint arXiv:2305.00675* (2023).
- [35] Kunyang Zhou and Rui Zhou. “End to end lane detection with one-to-several transformer”. In: *arXiv preprint arXiv:2305.00675* (2023).