GSENet: Global Semantic Enhancement Network for Lane Detection

Junhao Su[1*], Zhenghan Chen[4*], Chenghao He[2*], Dongzhi Guan[1†], Changpeng Cai[1], Tongxi Zhou[5], Jiasheng Wei[6], Wenhua Tian[1], Zhihuai Xie[3†]

[1]Southeast University
[2]East China University of Science and Technology
[3]Tsinghua University
[4]Peking University
[5]Institute of Automation Chinese Academy of Sciences
[6]Fudan University

Abstract

Lane detection is the cornerstone of autonomous driving. Although existing methods have achieved promising results, there are still limitations in addressing challenging scenarios such as abnormal weather, occlusion, and curves. These scenarios with low visibility usually require to rely on the broad information of the entire scene provided by global semantics and local texture information to predict the precise position and shape of the lane lines. In this paper, we propose a Global Semantic Enhancement Network for lane detection, which involves a complete set of systems for feature extraction and global features transmission. Traditional methods for global feature extraction usually require deep convolution layer stacks. However, this approach of obtaining global features solely through a larger receptive field not only fails to capture precise global features but also leads to an overly deep model, which results in slow inference speed. To address these challenges, we propose a novel operation called the Global feature Extraction Module (GEM). Additionally, we introduce the Top Layer Auxiliary Module (TLAM) as a channel for feature distillation, which facilitates a bottom-up transmission of global features. Furthermore, we introduce two novel loss functions: the Angle Loss, which account for the angle between predicted and ground truth lanes, and the Generalized Line IoU Loss function that considers the scenarios where significant deviations occur between the prediction of lanes and ground truth in some harsh conditions. The experimental results reveal that the proposed method exhibits remarkable superiority over the current state-of-the-art techniques for lane detection. Our codes are available at: https://github.com/crystal250/GSENet.

Introduction

The rise of deep neural network (Glorot, Bordes, and Bengio 2011) has led to increased applications in autonomous driving and advanced driver-assistance systems. Among these, lane detection is a fundamental task that plays a crucial role in controlling vehicle maneuvers and identifying lane boundaries. However, lane detection remains challenging, especially in complex scenarios.

(a) (b) (c) (d)

Figure 1: Some challenging scenarios for lane detection. (a) Demonstrates curved lanes, and our Angle Loss significantly improves the detection results in such scenarios. (b) Illustrates lanes under dazzle conditions, where lane detection heavily relies on global semantics. (c) Represents a no line scenario. (d) Depicts lanes are occupied by other vehicles, making lane detection exceptionally challenging.

Most traditional lane detection techniques (Canny 1986; Hough 1962; Sobel, Feldman et al. 1968) require manual parameter tuning to adapt to different road conditions and lighting environments, which can introduce variability and potential errors, leading to decreased stability and performance of the system. In these methods, operators often engage in the extraction of edge features (Canny 1986) or convert the color space of lane from RGB to HSV or HLS. Subsequent steps include binary thresholding, denoising, and other color-based processing methods are applied to acquire information on the lanes. Ultimately, the Hough Transform (Hough 1962) and lane fitting techniques, such as the Least Squares method and Random Sampling Consensus (RANSAC) (Fischler and Bolles 1981), are utilized to determine the lanes. Therefore, the manual parameter adjustments and feature extraction instability in traditional lane detection methods have led to inconsistencies and caused many difficulties in practical applications.

*Equal contribution.

†Corresponding author: Zhihuai Xie (xzhuai.me@gmail.com) and Dongzhi Guan (guandongzhi@seu.edu.cn).

Early neural network models heavily rely on instance segmentation and anchor-based object detection techniques. Despite their successes, these methods still face lane detection challenges, as depicted in Figure 1, such as poor visibility under adverse conditions and intricate lane configurations. Recent studies (Pan et al. 2018; Zheng et al. 2021, 2022) have aimed to tackle these issues. For instance, UFLD (Qin, Wang, and Li 2020) effectively utilizes lane coherence and shape loss to enhance detection speed and identify irregular lanes. However, its performance remains inadequate in various scenarios. Similarly (Zheng et al. 2022) introduces a cross-to-fine mechanism for improved lane detection models, yet it falls short in fully integrating global semantics and local features. Furthermore, it lacks comprehensive investigation into real-world challenging scenarios. We believe that precise lane prediction in complex scenarios necessitates the fusion of accurate global semantics and local features, along with a more refined loss function. Accurate prediction relies on comprehensive scene information from global semantics, encompassing visible lanes, road markings, vehicle and pedestrian positions and directions, to determine lane features in unseen parts. It also requires combining rich texture information from local features with targeted loss function penalties in diverse complex scenarios to precisely determine lane positions and shapes.

In this paper, we introduce the novel GSENet framework, recognizing that lanes in complex scenarios heavily rely on global semantics. To address this, we propose a new global feature extraction system comprising GEM and TLAM. Initially, feature maps from the top of the backbone are processed by GEM to obtain accurate and comprehensive global features. These features are then utilized in the upper structure and directly distilled to classification and regression heads via the TLAM pipeline in an auxiliary head form. Moreover, we introduce the Angle Loss to align predicted and GT lanes' shapes by considering their angles. Additionally, our GLIoU Loss extends predicted points into rectangles, enhancing performance compared to LIoU Loss (Zheng et al. 2022) and leading to smoother lane predictions. Our primary contributions are as follows:

- We have substantially improved lane detection capabilities by introducing GSENet.
- We have developed an innovative global semantic enhancement module, composed of GEM and TLAM, to fully leverage the global semantics within the network.
- We address the highly challenging task of lane detection in demanding scenarios by introducing the Angle Loss and GLIoU Loss.
- The performance of our proposed GSENet has been validated across multiple benchmark datasets, achieving state-of-the-art results.

Related Work

According to the strategy of lane status description, apart from the traditional lane detection methods mentioned earlier, the current mainstream lane detection methods are predominantly based on CNNs (LeCun et al. 1989). CNN-based Generally can be divided four categories:

Segementation-Based Methods

Segmentation methods are the earliest and most commonly used approaches in lane detection based on CNN methods. They involve pixel-level classification, resulting in high accuracy but slower processing due to per-pixel calculations. Early methods (Pan et al. 2018) treat lane detection as a multi-category instance segmentation problem and propose a spatial CNN to learn the prior knowledge of shape. To reduce the computational burden, RESA(Zheng et al. 2021) has been introduced after each OPS Stride. Despite these advancements, other segmentation methods (Wang, Ren, and Qiu 2018; Xu et al. 2020) still suffer from slow speed and perform poorly under occlusion or extreme conditions.

Row-Wise-Based Methods

Row-wise lane detection methods prioritize speed enhancement and lane shape prediction. UFLD (Qin, Wang, and Li 2020) transforms the problem by meshing, converting it into a classification challenge. CondLaneNet (Liu et al. 2021a) introduces conditional convolution for refined lane detection, and a recurrent instance module tackles lane bifurcation. UFLDv2 (Qin, Zhang, and Li 2022) recently proposes a hybrid anchor system to reduce positioning errors, building on UFLD. Despite introducing a novel classification loss, lateral lanes are excluded due to grid settings, necessitating post-processing.

Anchor-Based Methods

Anchor-based lane detection resembles object detection algorithms like YOLO (Redmon et al. 2016; Redmon and Farhadi 2018; Bochkovskiy, Wang, and Liao 2020; Wang, Bochkovskiy, and Liao 2023). It employs pre-set multi-ray anchors, pinpointing lanes through Non-Maximum Suppression (NMS) (Felzenszwalb et al. 2009) based on Intersection over Union (IoU). This method offers an end-to-end model with high accuracy.

In the realm of two-stage anchor-based lane detection, Line-CNN (Li et al. 2019) employs Faster R-CNN (Girshick 2015) as the lane detector, while LaneATT (Tabelini et al. 2021a) uses a versatile one-stage detection algorithm. CLRNet (Zheng et al. 2022) is a comprehensive model that partitions anchors into 72 points for balanced performance and speed. It introduces a coarse-to-fine mechanism and ROIGather for global information capture.

However, the anchor-based approach's reliance on predefined anchors limits adaptability, posing challenges in diverse extreme scenarios.

Polynomial-Regression-Based Methods

Unlike the approaches mentioned earlier, polynomial regression methods directly generate polynomial equations to represent lanes. These methods involve regressing coefficients and other parameters (e.g., eta for confidence scores). An influential approach in this category is PolyLaneNet (Tabelini et al. 2021b), which made significant contributions.

Another noteworthy method is LSTR (Liu et al. 2021b) which employs a transformative approach to predict DETR-based polynomials and achieves a remarkable processing

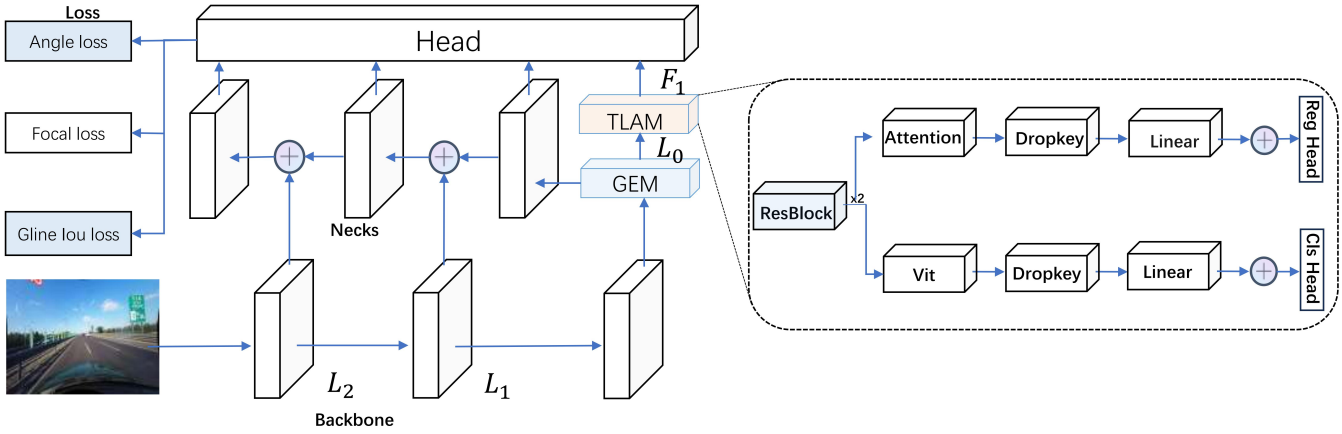


Figure 2: Overview of the proposed GSENet, the image is processed through the backbone, producing feature maps using FPN (Lin et al. 2017) and the Global Feature Extraction Module. Lane classification and regression are performed using features from FPN and refined global semantics from the Top Layer Auxiliary Module. In TLAM, we employ DropKey (Li et al. 2023) as an alternative to conventional dropout. The introduced Angle Loss and GLIoU Loss further improve the model's performance.

speed of 420 FPS. However, its accuracy performance is unsatisfactory.

Method

Our lane detection method builds upon the State-of-the-Art CLRNet (Zheng et al. 2022) as the baseline, integrating multiple enhancements and innovations.

Global Feature Extraction Module (GEM)

Motivation. In some CNN-based lane detection methods (Zheng et al. 2021, 2022; Qin, Wang, and Li 2020; Liu et al. 2021a; Qin, Zhang, and Li 2022) the integration of global semantic information poses notable challenges. However, in some complex scenarios, predicting the invisible lanes caused by circumstances such as occlusions or low-light conditions necessitates reliance on the integrated information provided by global semantics. These difficulties significantly impact the detection accuracy in such complex scenarios. Therefore, a neural network with an enhanced ability to acquire global semantics is necessary. Traditional methods for global feature extraction usually require deep convolution layer stacks. Thus, in pursuit of a more efficient and systematic global feature extraction module, we introduce a novel structure named GEM. GEM is designed to provide superior global features to the upper-level structures.

GEM Structure. The GEM is composed of two branches that complement each other to achieve more accurate global features, as can be seen in Figure 3. In lower branch, the top layer feature map from the backbone is fed into the MLP-mixer (Rumelhart, Hinton, and Williams 1986) network after undergoing a simple channel scaling and normalization process. The MLP-mixer network establishes preliminary global feature relationships and spatial relationships through the interaction of spatial feature information and channel feature information. However, this network has limitations in capturing fine-grained global features, and it can only extract coarse global features. To address this issue, we propose another branch. In upper branch, we first pass the feature map through a dilated convolution, enabling it to analyze more context information through a larger receptive field. Subsequently, the feature map is partitioned into P sub-blocks and distributed among h heads for processing. In each head, we calculate the similarity between each pixel in the P regions and the pixels in other regions, and take a weighted sum operation. By dividing it into multiple sub-blocks and using a multi-head mechanism for similarity computation, we not only obtain more accurate global features through the high-granularity calculation method of sub-blocks, but the multi-head mechanism can also simultaneously focus on multiple spatial positions, better capturing long-distance dependencies. Additionally, we introduce a SimAm (Yang et al. 2021) block to compensate for the absence of 3D spatial attention in the network's final stages. It is a parameter-free 3D attention module, aiming at improving the representation ability of the global semantic information. Finally, the distinct granular-level global semantic features derived from both branches are fused. This fusion compensates for the individual shortcomings of each branch and leverages their strengths, culminating in a comprehensive and precise global feature representation.

Top Layer Auxiliary Module (TLAM)

Motivation. Our motivation is to leverage the rich global semantic information present in the top layer feature map to further assist the neural network in classification and regression heads. Based on the (Vaswani et al. 2017) and its applications in the field of computer vision (Carion et al. 2020; Dosovitskiy et al. 2020), it is believed that the network's ability to detect challenging samples can be significantly improved by using transformer to further enhance the semantic representation. Therefore, we propose TLAM, a novel auxiliary head for distilling top layer global features from GEM to the final classification and regression stage.

TLAM Structure. In TLAM, the top layer feature maps

Method	Backbone	mF1	F1@50	F1@75	Normal	Crowded	Dazzle	Shadow	No line	Arrow	Cross	Night
SCNN	VGG16	38.84	71.60	39.84	90.60	69.70	58.50	66.90	43.40	84.10	1990	66.10
RESA	ResNet50	47.86	75.30	53.39	92.10	73.10	69.20	72.80	47.70	88.30	1503	69.90
E2E	ERFNet	-	74.00	-	91.00	73.10	64.50	74.10	46.60	85.80	2022	67.90
UFLD	ResNet18	38.94	68.40	40.01	87.70	66.00	58.40	62.80	40.20	81.00	1743	62.10
UFLD	ResNet34	-	72.30	-	90.70	70.20	59.50	69.30	44.40	85.70	2037	66.70
PINet	Hourglass	46.81	74.40	51.33	90.30	72.30	66.30	68.40	49.80	83.70	1427	67.70
LaneATT	ResNet34	49.57	76.68	54.34	92.14	75.03	66.47	78.15	49.39	88.38	1330	70.72
LaneATT	ResNet122	51.48	77.02	57.50	91.74	76.16	69.47	76.31	50.46	86.29	1264	70.81
UFLDv2	ResNet34	-	76.0	-	92.50	74.80	65.50	75.50	49.20	88.80	1910	70.80
LaneAF	DLA34	50.42	77.41	56.79	91.80	75.61	71.78	79.12	51.38	86.88	1360	73.03
SGNet	ResNet34	-	77.67	-	92.07	75.41	67.75	74.31	50.90	87.97	1373	72.69
FOLOLane	ERFNet	-	78.80	-	92.70	77.80	75.20	79.30	52.10	89.00	1569	74.50
CondLane	ResNet34	53.11	78.74	59.39	93.38	77.14	71.17	79.93	51.85	89.89	1387	73.92
CondLane	ResNet101	54.83	79.48	61.23	93.47	77.44	70.93	80.91	54.13	90.16	1201	74.80
CANet	ResNet34	-	79.16	-	93.58	77.88	73.11	75.06	51.68	90.09	1176	73.92
CANet	ResNet101	-	79.86	-	93.60	78.74	70.07	79.35	52.88	90.18	1196	74.91
CLRNet	ResNet18	55.23	79.58	62.21	93.30	78.33	73.31	79.66	53.14	90.25	1321	75.11
CLRNet	ResNet34	55.14	79.73	62.11	93.49	78.06	74.57	79.92	54.01	90.59	1216	75.02
CLRNet	ResNet101	55.55	80.13	62.96	93.85	78.78	72.49	82.33	54.50	89.79	1262	75.51
CLRNet	DLA34	55.64	80.47	62.78	93.73	79.59	75.30	82.51	54.58	90.62	1155	75.37
GSENet(ours)	ResNet18	55.93	80.42	63.50	93.66	79.14	74.80	81.91	54.30	89.99	1045	75.80
GSENet(ours)	ResNet34	56.04	80.58	63.37	93.80	79.42	75.34	82.27	54.83	90.67	1072	76.07
GSENet(ours)	ResNet101	**56.53**	80.84	**64.23**	**94.05**	79.90	74.94	82.21	55.63	**90.78**	1164	76.08
GSENet(ours)	DLA34	56.45	**81.13**	64.08	93.91	**80.30**	**76.36**	**83.41**	**56.25**	90.36	**1036**	**76.26**

Table 1: State-of-the-art results on CULane. As we can see, we have attained the highest performance in challenging scenarios encompassing Crowded, Dazzle, Shadow, No line, Arrow, Cross, and Night.

from GEM will perform two different types of self-attention (Vaswani et al. 2017) computations to enhance the representation of global semantic features, so that global features can more effectively act on the classification and regression heads separately. Before performing self-attention computations, a simple residual network is employed to further extract global semantic information from the top layer feature map. The top layer feature map generated by the backbone is denoted as L_0.

$$F_{top} = \phi(\phi(L_0)), \tag{1}$$

$$S_1, S_2 = Auxihead_1(F_{top}), Auxihead_2(F_{top}), \tag{2}$$

where ϕ is a simple residual network, $Auxihead_1$ first divides the ${F_{top}}^{B\times C\times H\times W}$ into patches (Dosovitskiy et al. 2020), then the obtained ${F_{top}}^{B\times N\times H\times W}$ is flattened and reshaped into ${F_{top}}^{B\times (H\times W)\times N}$, self-attention computations are applied to ${F_{top}}^{B\times (H\times W)\times N}$, consequently S_1 is obtained. After undergoing Dropkey (Li et al. 2023) processing, S_1 is concatenated to the classification heads F_{cls}. The $Auxihead_2$ simply flattens and reshapes ${F_{top}}^{B\times C\times H\times W}$ into ${F_{top}}^{B\times (H\times W)\times N}$, then we compute the self-attention of the ${F_{top}}^{B\times (H\times W)\times N}$ to obtain S_2, After undergoing Dropkey processing, then S_2 is concatenated to the regression heads F_{reg}.

Angle Loss

Motivation. In the lines formed by connecting the predicted points with the ground truth points, noticeable angles exist between adjacent short line segments. The presence of these angles often results in disparities between the predicted and GT lane shapes. If we can minimize these angles, the shape of the predicted lane can closely approximate the GT lane's shape. We have derived a straightforward and efficient algorithm to compute the Angle Loss.

Formula. Firstly, the coordinates of a predicted lane point are defined as (x_i^P, y_i^P), and the corresponding GT point is (x_i^G, y_i^G), and the angle θ_i between them is expressed as follows:

$$\theta_i = arctan\left|\frac{(\frac{y_i^P - y_{i-1}^P}{x_i^P - x_{i-1}^P}) - (\frac{y_i^G - y_{i-1}^G}{x_i^G - x_{i-1}^G})}{1 - (\frac{y_i^P - y_{i-1}^P}{x_i^P - x_{i-1}^P}) \times (\frac{y_i^G - y_{i-1}^G}{x_i^G - x_{i-1}^G})}\right|, \tag{3}$$

where the overall angle θ between the entire predicted lane and the GT lane is defined as the average of the angles between the individual short lines. It is calculated as follows:

$$\theta = \frac{\sum_{i=2}^{N} \theta_i}{N-1}, \tag{4}$$

the final angle loss is quantified using the cosine function as follows:

$$\mathcal{L}_{angle} = 1 - cos\theta. \tag{5}$$

Our Angle Loss offers the following advantages: (1) It enhances the model's sensitivity to variations in lane directions, resulting in predicted lane shapes that closely resemble the GT lanes. (2) The Angle Loss involves convenient

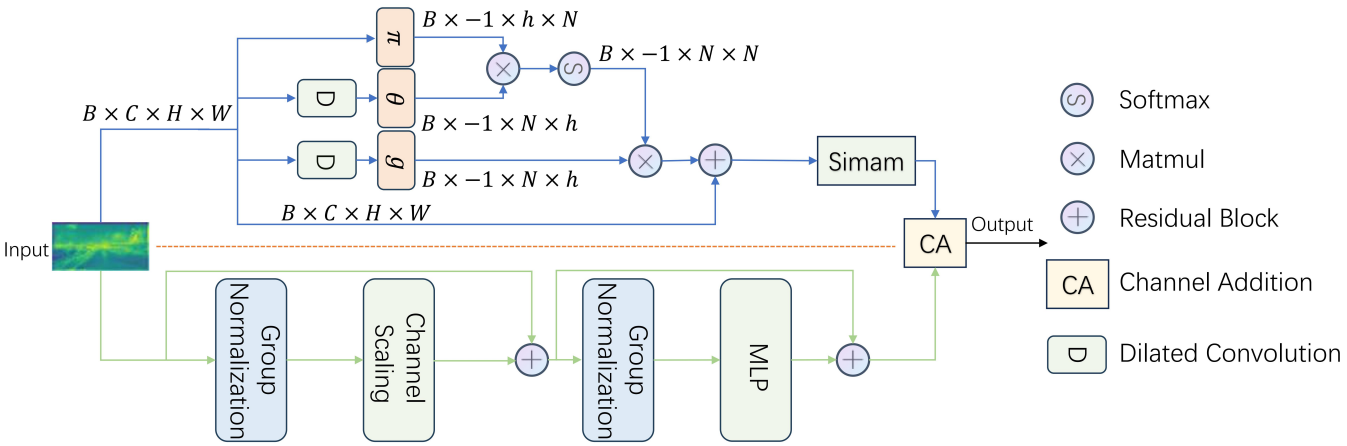


Figure 3: The feature maps are derived from the top layer of the backbone and then propagated into two branches.

Method	Backbone	F1(%)	Acc(%)	FP(%)
SCNN	VGG16	95.97	96.53	6.17
RESA	ResNet34	96.93	96.82	3.63
PolyLaneNet	EfficientNetB0	90.62	93.36	9.42
UFLD	ResNet34	88.02	95.86	18.91
LaneATT	ResNet34	96.77	95.63	3.53
LaneATT	ResNet122	96.06	96.10	5.64
UFLDv2	ResNet34	96.22	95.56	3.18
CondLaneNet	ResNet34	96.98	95.37	2.20
CondLaneNet	ResNet101	97.24	96.54	2.01
FOLOLane	ERFNet	96.59	**96.92**	4.47
CANet	ResNet34	97.44	96.66	2.32
CANet	ResNet101	97.77	96.76	1.92
CLRNet	ResNet18	97.89	96.84	2.28
CLRNet	ResNet34	97.82	96.87	2.27
CLRNet	ResNet101	97.62	96.83	2.37
GSENet	ResNet18	**97.98**	96.82	**1.79**
GSENet	ResNet34	97.94	96.88	2.04
GSENet	ResNet101	97.90	96.81	2.15

Table 2: State-of-the-art results on TuSimple. Additionally, F1 was computed using the official source code.

computation and straightforward gradient calculations, leading to minimal complexity overhead while significantly improving the overall model performance.

Generalized Line IoU Loss (GLIoU Loss)

Motivation. Drawing inspiration from the Line IoU Loss (Zheng et al. 2022), it's evident that computing the loss via discrete points as independent variables yields suboptimal results. Through a clever approach, we extend each predicted point and its corresponding GT point into a rectangle. By transforming points into geometric shapes, we exploit the inherent geometric relationships between consecutive points, which greatly facilitates achieving smoother predicted lane shapes. Moreover, the model enforces an additional penalty to facilitate enhanced regression accuracy. We have developed a computationally efficient algorithm that is amenable to parallel computation for calculating the Generalized Line IoU Loss (GLIoU Loss).

Formula. We consider a predicted point (x_i^P, y_i^P) and its corresponding GT point (x_i^G, y_i^G). Two adjacent points below them are (x_{i-1}^P, y_{i-1}^P) and (x_{i-1}^G, y_{i-1}^G). We connect these two points to form a line, treating it as the rectangle's length. Extending this point perpendicularly along the line to both sides for a length of e allows us to create rectangles, the value of e is set to 15. Specifically, the area of the two rectangles can be calculated as follows: $S_{rec}^P = 2 \times e \times \sqrt{(x_i^P - x_{i-1}^P)^2 + (y_i^P - y_{i-1}^P)^2}$ and $S_{rec}^G = 2 \times e \times \sqrt{(x_i^G - x_{i-1}^G)^2 + (y_i^G - y_{i-1}^G)^2}$. We can determine angles θ_i^P and θ_i^G between the predicted line and the GT line concerning the vertical direction. Then, the area of the bounding rectangle can be computed using these coordinates:

$$y_{top} = y_i + max(e \times sin\theta_i^P, e \times sin\theta_i^G), \tag{6}$$

$$y_{bottom} = y_{i-1} - max(e \times sin\theta_i^P, e \times sin\theta_i^G), \tag{7}$$

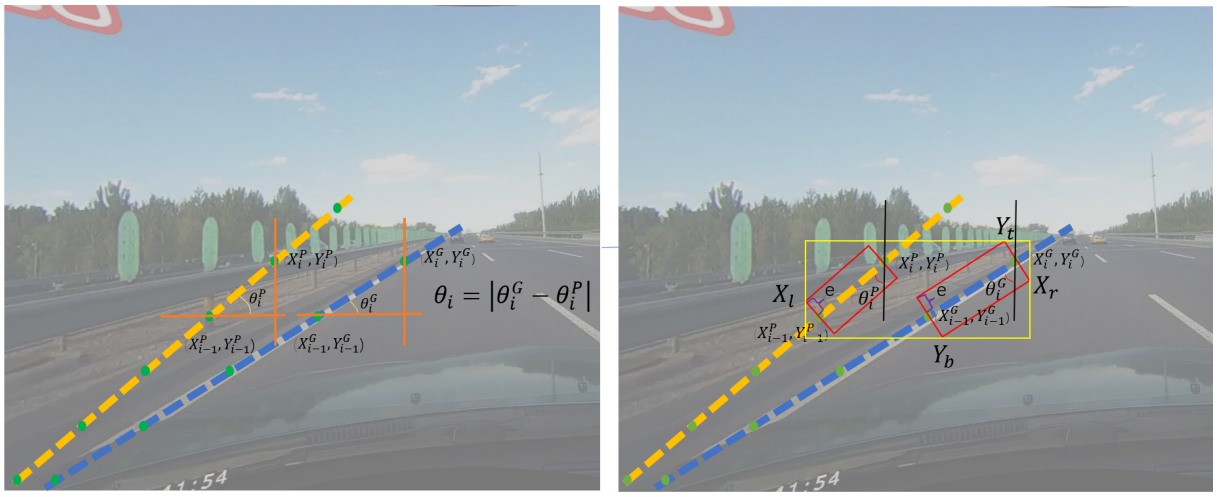
$$\begin{aligned} x_{left} = min(&x_i^P - e \times cos\theta_i^P, x_{i-1}^P - e \times cos\theta_i^P, \\ &x_i^G - e \times cos\theta_i^G, x_{i-1}^G - e \times cos\theta_i^G), \end{aligned} \tag{8}$$

$$\begin{aligned} x_{right} = max(&x_i^P + e \times cos\theta_i^P, x_{i-1}^P + e \times cos\theta_i^P, \\ &x_i^G + e \times cos\theta_i^G, x_{i-1}^G + e \times cos\theta_i^G), \end{aligned} \tag{9}$$

then we can calculate $S_{bound} = (y_{top} - y_{bottom}) \times (x_{right} - x_{left})$, based on CLRNet (Zheng et al. 2022), we define the IoU for points as follows:

$$IoU = \frac{min(x_i^P + e, x_i^G + e) - max(x_i^P - e, x_i^G - e)}{max(x_i^P + e, x_i^G + e) - min(x_i^P - e, x_i^G - e)}, \tag{10}$$

furthermore, the GIoU for points can be expressed as:



(a) (b)

Figure 4: Illustration of Angle Loss and GLIoU Loss, (a) Angle Loss computes the average of the angles between predicted points and their corresponding GT points on each lane. (b) If a predicted point is distant from its GT point, their extended rectangles won't intersect, resulting in an additional penalty through GLIoU.

$$G = \frac{S_{bound} - min(S_{bound}, S_{rec}^P + S_{rec}^G)}{S_{bound}}, \tag{11}$$

$$GIoU = \frac{d_i^{GI}}{d_i^{GU}} = \begin{cases} IoU - G, (2 \le i \le N) \\ IoU, i = 1 \end{cases}, \tag{12}$$

exactly, GLIoU can be viewed as a combination of GIoU for a finite set of points, then the GLIoU loss can be calculated as:

$$GLIoU = \frac{\sum_{i=1}^{N} d_i^{GI}}{\sum_{i=1}^{N} d_i^{GU}}, \tag{13}$$

then, the GLIoU loss can be calculated as:

$$\mathcal{L}_{GLIoU} = 1 - GLIoU. \tag{14}$$

GLIoU offers the following advantages: (1) Treating isolated points of the predicted lane as a cohesive unit for regression enhances the overall model performance. (2) Utilizing the geometric relationships between adjacent points facilitates smoother predictions for the lane, optimizing its continuity.

Experiment

Datasets

Our experiments were conducted on two widely recognized and extensively used lane detection datasets in the industry: CULane (Pan et al. 2018) and Tusimple (TuSimple 2020).

CULane is a large-scale lane detection data set consists of 88.9k training data, 9.7k verification data, and 34.7k test set data. The pixel value of all its pictures is 1640×590, covering various autonomous driving scenarios and a large number of challenging scenes such as urban roads, country roads, crowded, abnormal weather environments, lighting conditions, etc.

TuSimple is another large-scale lane detection dataset developed by the self-driving company Tucson. The data set consists of 3.3k training set, 0.4k validation set and 2.8k validation set, the pixel values of all pictures are 1280×720. The most distinctive feature of TuSimple is that it has a more detailed lane change model, such as the width and shape of the lane.

Evaluation Metric

In the CULane (Pan et al. 2018) dataset, we use F1-measure. Calculate the errors of predictions and ground truth through IoU. The calculation formula is as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{15}$$

If the IoU of predicted lanes with the ground truth lanes is greater than the specified threshold, the prediction is considered True Positive(TP). Otherwise, it is classified as False Positive(FP). In addition, this paper also uses mF1 (Zheng et al. 2022) proposed by CLRNet as one of the metrics. The mF1 is defined as:

$$mF1 = \frac{\sum_{i=10}^{19} F1@(i \times 5)}{10}. \tag{16}$$

Among them, F1@50 and F1@75 are the F1 scores under IoU 0.5 and 0.75 respectively, and in TuSimple (TuSimple 2020), we use the evaluation formula of:

$$Accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}, \tag{17}$$

where C_{clip} and S_{clip} are the number of correct points and the number of ground truth respectively. Whether it is judged

Angle Loss	GLIoU Loss	TLAM	GEM	mF1	F1@50	F1@75
				55.23	79.58	62.21
✓				55.61	79.81	62.76
✓	✓			55.61	80.03	63.09
✓	✓	✓		55.87	80.31	**63.55**
✓	✓	✓	✓	**55.93**	**80.42**	63.51

Table 3: Abalation study of each method. Results were obtained using ResNet18 backbone on the CULane dataset.

as a correct point is based on whether more than 85% of the pixels in the ground truth are correctly predicted. In addition to the traditional evaluation metrics, the TuSimple dataset also includes an additional metrics: False Positives(FP), where $FP = \frac{F_{pred}}{N_{pred}}$.

Implement Detail

We apply ResNet18, ResNet34, ResNet101 (He et al. 2016), DLA34 (Yu et al. 2018) as backbones. In terms of data processing, for all data sets, we cut the input data to 800×320. The same data augmentation: random affine transformation such as translation, rotation and scaling, random horizontal flips. In terms of optimization, the AdamW optimizer (Kingma and Ba 2014) and cosine decay learning rate strategy are adopted. Similar to (Zheng et al. 2022). In the CULane and TuiSimple datasets, we set epoch=15, lr=6e-4, batchsize=24, epoch=70, lr=1.0e-3, batchsize=40, h=8, P=10, respectively. The angle loss weight in all datasets is set to 15, and the balance between GLIoU Loss and Angle Loss is controlled by a hyperparameter α, adjusting their proportions. The combined loss incorporating GLIoU Loss and Angle Loss is formulated as follows:

$$\mathcal{L}_{comb} = \alpha \times \mathcal{L}_{GLIoU} + (1 - \alpha) \times \mathcal{L}_{angle}. \tag{18}$$

Based on our experiments, we define α as 0.98. In addition, our network is implemented based on pytorch framework and trained on a single GeForce RTX 4090 GPU.

Comparison With the SOTA Results

CULane Dataset. Our method's performance on the CULane dataset is presented here, along with comparisons to other state-of-the-art techniques. When utilizing DLA34 (Yu et al. 2018) as the backbone, we achieve an F1 score of 81.13 at F1@50 on the CULane dataset, reaching a state-of-the-art level. As indicated in Table 1, noteworthy results emerge when employing ResNet18 (He et al. 2016) as the backbone. We obtain a score of 80.42 at F1@50, surpassing CLRNet (Zheng et al. 2022) (ResNet18) by 0.84 points. This even outperforms CLRNet (ResNet101), underscoring the substantial enhancement our global semantic approach brings to lane localization and regression accuracy. Similarly, in Table 1, using ResNet101 as the backbone leads to mF1 (Zheng et al. 2022) and F1@75 scores that surpass CLRNet (ResNet101) by 0.98 and 1.27 points, respectively.

Figure 5 illustrates the outcomes of lane detection, highlighting significant differences. Competing methods encounter hurdles in occlusions, curved lanes, and extreme scenarios, resulting in subpar performance. In contrast, our method excels, thriving in challenging scenarios. Its robustness shines, effectively addressing difficulties and yielding dependable, satisfactory lane detection results.

TuSimple Dataset. The performance of our method on the TuSimple benchmark dataset is presented in Table 2. Notably, performance distinctions among various methods are minimal, suggesting the bottleneck in advancements on this dataset. Despite its challenging nature, we achieve a noteworthy F1@50 score of 97.98, outperforming the current state-of-the-art by 0.09 points. Additionally, we attain state-of-the-art results in the False Positives (FP) metric, demonstrating a substantial 6.8% enhancement compared to prior approaches. These achievements underscore our method's effectiveness in addressing lane detection challenges and its superior performance on the TuSimple dataset.

Ablation Study

To ascertain the effectiveness of each component in our method and ensure that each contributes to the improvement in detection performance, we conduct multiple ablation studies on the CULane (Pan et al. 2018) dataset.

Overall Ablation Study. In Table 3, we present the results of the comprehensive ablation study. Firstly, we validate the effectiveness of each individual component, when we add the Angle Loss to the baseline (Zheng et al. 2022), the score of F1@50 increases from 79.58 to 79.81. Subsequently, through experimentations we discover that using a single hyperparameter α to control the balance between GLIoU Loss and Angle Loss is more beneficial for the model. Applying this approach achieves an improvement of F1@50 from 79.58 to 80.03, and we will elaborate on the choice of hyperparameter α in the supplementary materials. Moreover, we further add the TLAM, which leads to a significant improvement of F1@50 score from 80.03 to 80.31, this result suggests that calculating the self-attention (Vaswani et al. 2017) of the top layer feature map can greatly enhance the model's ability to gather global information effectively. Finally, by adding the GEM on top of the previous improvements, we achieve an F1@50 score of 80.42, which further demonstrates the effectiveness of the GEM component in enhancing the overall performance of the model.

Ablation Study on TLAM's Number of Residual Blocks. We conduct an ablation study on the number of residual blocks (He et al. 2016) in TLAM, as shown in Table 4. TLAM utilizes residual blocks before self-attention calculations (Vaswani et al. 2017) to boost global semantics in the feature map. The optimal number of blocks is pivotal for

Figure 5: Visualization results of LaneATT, CLRNet and our method on CULane testing set.

Res blocks	mF1	F1@50	F1@75	Shadow	No line
No TLAM	55.23	79.58	62.21	79.66	53.14
0 × blocks	55.20	79.57	62.22	81.21	53.40
1 × blocks	**55.32**	79.60	62.30	81.10	53.89
2 × blocks	55.28	**79.72**	**62.56**	81.51	**54.39**
3 × blocks	55.19	79.61	62.52	**81.82**	53.22
4 × blocks	55.14	79.64	62.26	80.54	53.94

Table 4: Abalation studies of the number of residual blocks of TLAM.

Weight	mF1	F1@50	F1@75	Curve	Cross
0	55.23	79.58	62.21	71.56	1321
10	55.35	79.75	62.62	72.90	1205
15	**55.61**	**79.81**	**62.76**	**73.84**	**995**
20	55.12	79.65	62.42	73.01	1126
25	55.15	79.54	61.95	73.33	1165

Table 5: Abalation studies of the weight of Angle Loss.

model performance. Results indicate that using 0 or 1 blocks can yield worse performance than the baseline due to insufficient global semantics enrichment. On the other hand, using 3 or 4 blocks can lead to a decline in performance compared to using 2 blocks. This suggests that too many blocks may sacrifice fine details, causing performance degradation. Importantly, TLAM excels in challenging scenarios like Shadows and No line, enhancing global semantics to enable the model's success in tough conditions, thereby achieving substantial performance gains.

Ablation Study on Angle Loss Weight. Table 5 depicts ablation studies on the weight of Angle Loss. The results highlight the advantageous impact of Angle Loss when its weight is small. However, exceeding a weight of 20 shifts the model's focus excessively on optimizing this loss, causing a sharp performance decline even below the baseline. The optimal Angle Loss weight is determined to be 15, elevating the F1@50 score from 79.58 to 79.81. Angle Loss significantly improves model performance in challenging scenarios such as Curve and Cross. In the Curve scenario, the F1@50 score gains 2.28 points, while in the Cross scenario, an impressive 24.7% improvement is observed. These outcomes vividly underscore the efficacy of our Angle Loss.

Conclusion

In this work, we propose GSENet, a novel lane detection network that detects the lane by enhancing the global semantic and introducing two new loss function, aiming to solve the lane detection problem in various difficult scenarios. We propose that GEM and TLAM modules extract rich global features, moreover, we design Angle Loss and GLIoU Loss for difficult scenarios. Our method has been thoroughly validated on both the CULane (Pan et al. 2018) and TuSimple (TuSimple 2020) benchmark datasets. The results demonstrate that our approach has achieved state-of-the-art performance on these datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.52278154) and the Natural Science Foundation of Jiangsu Province (BK20231429).

References

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6): 679–698.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2009. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9): 1627–1645.

Fischler, M. A.; and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381–395.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323. JMLR Workshop and Conference Proceedings.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hough, P. V. 1962. Method and means for recognizing complex patterns. US Patent 3,069,654.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551.

Li, B.; Hu, Y.; Nie, X.; Han, C.; Jiang, X.; Guo, T.; and Liu, L. 2023. DropKey for Vision Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22700–22709.

Li, X.; Li, J.; Hu, X.; and Yang, J. 2019. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1): 248–258.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Liu, L.; Chen, X.; Zhu, S.; and Tan, P. 2021a. Condlanenet: a top-to-down lane detection framework based on conditional convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3773–3782.

Liu, R.; Yuan, Z.; Liu, T.; and Xiong, Z. 2021b. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 3694–3702.

Pan, X.; Shi, J.; Luo, P.; Wang, X.; and Tang, X. 2018. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Qin, Z.; Wang, H.; and Li, X. 2020. Ultra fast structure-aware deep lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, 276–291. Springer.

Qin, Z.; Zhang, P.; and Li, X. 2022. Ultra fast deep lane detection with hybrid anchor driven ordinal classification. *IEEE transactions on pattern analysis and machine intelligence*.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature*, 323(6088): 533–536.

Sobel, I.; Feldman, G.; et al. 1968. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, 271–272.

Tabelini, L.; Berriel, R.; Paixao, T. M.; Badue, C.; De Souza, A. F.; and Oliveira-Santos, T. 2021a. 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*, 294–302.

Tabelini, L.; Berriel, R.; Paixao, T. M.; Badue, C.; De Souza, A. F.; and Oliveira-Santos, T. 2021b. Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 6150–6156. IEEE.

TuSimple. 2020. TuSimple Benchmark. https://github.com/TuSimple/tusimple-benchmark/. Accessed September,2020.2,5.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2023. 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*, 7464–7475.

Wang, Z.; Ren, W.; and Qiu, Q. 2018. Lanenet: Real-time lane detection networks for autonomous driving. *arXiv preprint arXiv:1807.01726*.

Xu, H.; Wang, S.; Cai, X.; Zhang, W.; Liang, X.; and Li, Z. 2020. 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*, 689–704. Springer.

Yang, L.; Zhang, R.-Y.; Li, L.; and Xie, X. 2021. Simam: A simple, parameter-free attention module for convolutional neural networks. In *International conference on machine learning*, 11863–11874. PMLR.

Yu, F.; Wang, D.; Shelhamer, E.; and Darrell, T. 2018. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2403–2412.

Zheng, T.; Fang, H.; Zhang, Y.; Tang, W.; Yang, Z.; Liu, H.; and Cai, D. 2021. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3547–3554.

Zheng, T.; Huang, Y.; Liu, Y.; Tang, W.; Yang, Z.; Cai, D.; and He, X. 2022. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 898–907.