IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 July 2024, accepted 18 August 2024, date of publication 22 August 2024, date of current version 2 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3447738

RESEARCH ARTICLE

Enhancing Lane Recognition in Autonomous Vehicles Using Cross-Layer Refinement Network

PRANAV CHAUDHARI, RAGHAVENDRA ACHAR, (Member, IEEE), AND SANJAY SINGH, (Senior Member, IEEE)

Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

Corresponding author: Sanjay Singh (sanjay.singh@manipal.edu)

ABSTRACT Lanes are different sections of a roadway that are marked or assigned to traffic movement. They are used to organize and govern the passage of vehicles on roads. A lane is essential for the visual navigation system of an autonomous vehicle. The concept of a lane represents a traffic sign that has significant meaning. However, it also exhibits a unique local pattern that requires detailed low-level characteristics to identify its location accurately. Utilizing various feature levels is crucial for achieving effective lane recognition. Therefore, this study used a Cross-Layer Refinement Network (CLRNet) to enhance lane recognition by incorporating high and low-level lane-detecting characteristics. This approach involves identifying lanes based on high-level semantic properties, and refining them using low-level features. The proposed method aims to improve the localization accuracy by leveraging additional contextual information and local-specific lane characteristics. The network architecture combines the elements of LeNet-5 and AlexNet, utilizing the more profound architecture of AlexNet for complex feature learning and localized pattern recognition from LeNet-5. A global context is acquired to enhance the lane feature representation. The Line Intersection over Union (LIoU) loss function, which treats the lane line as a whole unit rather than as individual segments, is employed to enhance the localization accuracy. The experimental results demonstrate the superior performance of the proposed method compared to existing state-of-the-art lane detection algorithms.

INDEX TERMS AlexNet, IoU, lane detection, LeNet-5, ROI gather.

I. INTRODUCTION

Lanes are designated divisions on roads and highways that direct and divide traffic movements. Painted lines on the surface of the road often represent lanes and help organize, and regulate the vehicle flow. Lane identification is an essential task for computer vision and autonomous driving systems. It involves recognition and monitoring of lane borders on road using visual inputs, often photos or video frames acquired by cameras placed on a vehicle [1]. Lane detection is essential in various applications, including lane departure warning systems, autonomous navigation, and advanced driver-assistance systems (ADAS). The form and color of lane marks may vary based on local traffic laws and standards. Common lane markers are solid white lines, dashed white lines, double yellow lines, and arrows denoting unique lane usage or limits.

Many techniques have demonstrated promising performance using CNN's excellent feature representation [2]. However, there are still significant difficulties in determining the express lanes. Lanes have high-level semantics, basically the lane geometry (shape and curvature) and the context of the road, but a distinct local pattern that requires explicit low-level characteristics (i.e, the edges and color of the lanes), to effectively localize [1]. The question of how to successfully use multiple feature levels in a CNN still needs to be addressed. In Fig. 1(a), the landmark and lane lines possess distinct meanings but share common characteristics. Distinguishing between them becomes more accessible by utilizing high-level semantics and global context.

However, considering the road's long and thin structure with a basic local layout, locality also plays a crucial role. The detection outcome based on high-level characteristics is

The associate editor coordinating the review of this manuscript and approving it for publication was Ivan Wang-Hei Ho.

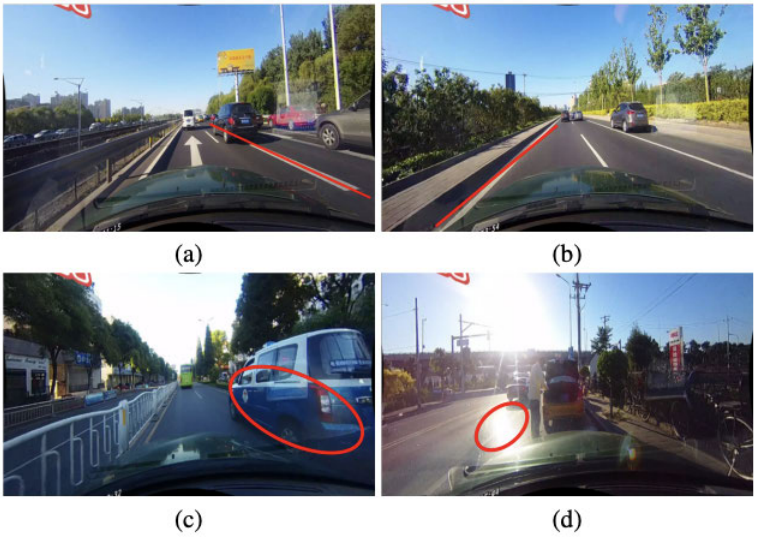

FIGURE 1. Illustrations of challenging scenarios for lane detecting. (a) The detection result of low-level characteristics. Due to a lack of global context, it misidentifies a landmark as a roadway. (b) The detection result of high-level characteristics. It forecasts incorrect lane localization. (c) The circumstance in which the lane is completely taken by an automobile. (d) When the lane is blurred due to poor illumination.

presented in Fig. 1(b), where the lane is identified but lacks a precise placement. Thus, successful lane recognition requires high and low-level information because it is complementary. Previous detectors [3], [4], [5], [6] demonstrated the importance of local and global characteristics for lane recognition. However, they did not exploit high and low-level features, leading to incorrect detection performance.

Our system introduces a sophisticated approach to autonomous vehicle lane detection. An imaging unit captures road images encompassing crowded, dazzling, shadowy, nighttime, curved, no-line, and arrow-laden conditions. The subsequent image processing units refine these images by resizing, normalizing, and augmentation to optimize the features crucial for lane detection. Its architecture involves a hybrid deep-learning approaches that incorporates LeNet-5 [7] and AlexNet [8]. This system can be seamlessly integrated into broader autonomous vehicle systems, contributing to cross-layer refinement in lane detection and offering a fresh perspective on training models.

Controlled by a central unit, the system utilizes a Feature Pyramid Network (FPN) [9] to generate feature maps and seamlessly integrate high and low-level lane detection attributes. Global context information is enhanced by extracting a Region of Interest (ROI), thereby improving the representation of lane features.

One aspect of the system is the use of Line Intersection over Union (LIoU) loss computation. It treats lane as a unified entity and assesses the overlap between the predicted and ground-truth lanes. The process dynamically selects and assigns projected lanes as positive samples for cost representations, ultimately refining lane identification. System exhibits flexibility by regressing the entire lane and optimizing the performance based on Intersection over Union (IoU) as a loss function. The line IoU loss is akin to the IoU metric for bounding boxes and measures the similarity between the predicted and ground-truth lanes. In addition, the system incorporates a cost representation, associating penalties with various aspects of lane detection performance, including the position accuracy and spatial smoothness. The lane detection method follows an architectural process encompassing image capture, preprocessing, representation, and refinement, showcasing an advanced and innovative approach for autonomous vehicle lane detection.

The effectiveness of the proposed approach was demonstrated using three lane detection benchmarks: CULane, TuSimple, and LLAMAS. The experimental results reveal that the proposed technique achieves state-of-the-art accuracy across all datasets. The main contributions of this study are summarized as follows:

- This study elucidates the synergistic relationship between high and low-level characteristics in lane recognition. It leverages detailed features and a broader contextual understanding, thereby enhancing lane detection accuracy.
- We develop a hybrid model, LeAlexNet, by integrating the architectures of LeNet-5 and AlexNet. This combined architecture serves as the backbone of our network, harnessing the strengths of both models to effectively capture and process lane characteristics from input data.
- We incorporated the ROIGather module into our network, which is a versatile component that can be integrated into other networks. It significantly improves the representation of lane characteristics by focusing on

the regions of interest and enhancing the model's ability to accurately discern and track lane markings.
- We employed the Lane Intersection over Union (LIoU) loss function, which is specifically optimized for lane recognition tasks that treat the lane as a holistic unit during regression. This leads to significant improvements in performance by ensuring more precise and consistent lane predictions.
- We introduced a novel mF1 metric designed to compare the localization accuracies of different detectors comprehensively.
- We extensively tested our architecture under various challenging conditions that typically complicate lane detection tasks. These conditions included night driving, shadows, glare (dazzle), lane arrows, crowded environments, curved roads, and intersections (cross scenarios).

The rest of this paper is organized as follows. In Section II related works on lane-detection systems are reviewed. Section III provides the proposed LeAlexNet model. Section IV provides the experimental and implementation details. Section V reports the results and discusses the experimental data. Finally, Section VI concludes the paper and discusses future work.

II. RELATED WORKS

Deep Learning has seen significant advancements over the past few decades, particularly in Car Lane Detection. Despite these developments, there remains a critical gap in understanding the robustness and generalization of lane detection models under diverse road environments. Addressing this gap is essential for enhancing the safety and reliability of autonomous driving systems, especially in real-world scenarios where environmental factors can significantly affect lane detection accuracy. In this section, we review the existing literature on Lane Detection in Deep Learning, focusing on the robustness of models, their ability to generalize across different environments, and the integration of multi-modal data for improved accuracy. This review will highlight the strengths and limitations of current approaches and set the stage for the contributions of our work.

Abualsaud et al., [6] presented an innovative approach to tackle the challenge of multi-lane detection in road environments. They introduced the concept of affinity fields, which leverage pixel-wise relationships to enhance the robustness and accuracy of multi-lane detection. The inference time of the Laneaf model may be relatively high, which can limit its real-time performance in specific applications. Zheng et al., [10] leveraged the concept of cross-layer refinement to enhance lane detection accuracy and robustness. By integrating multi-scale features from different layers of a deep neural network, CLRNet captures high-level and low-level lane information, enabling more precise and reliable lane detection. Its generalization to unseen scenarios, such as different road types or unique traffic conditions, may vary. Further validation and testing across diverse real-world environments are necessary to assess their generalization capabilities [9]. Lane marking variations, such as faded or distorted lane lines, may affect this. Simultaneously, the model demonstrated robustness in handling such variations to some extent.

Behrendt and Soussan [11] introduced a novel approach for unsupervised lane marker detection by leveraging a map information. The model employs image processing techniques and deep learning algorithms to accurately refine and localize lane markers. However, their approach may encounter challenges when dealing with variability in map data, such as missing or incomplete lane marker information. Cai and Vasconcelos [12] presented an in-depth exploration of the Cascade R-CNN framework that aims to improve the quality and accuracy of object detection. Each cascade stage progressively improves the detection quality by carefully selecting highly confident region proposals and refining their localization [13]. Its ability to effectively handle challenging detection scenarios, such as objects with small sizes, heavy occlusions, and low-resolution images, presents challenges in detecting fine-grained objects with subtle visual differences or intricate structures.

Zheng et al. [5] presented a novel approach to lane detection using a Recurrent Feature-Shift Aggregator (RESA) model. It uses recurrent neural networks (RNNs) to capture spatial and temporal dependencies for lane detection. Their generalizations to new and unseen environments may vary. The performance of the model under different road types, weather conditions, or unique traffic scenarios should be further investigated. Yoo et al. [14] introduced an approach to lane marker detection using an end-to-end framework based on row-wise classification. Their method eliminates the need for explicit lane marker segmentation or localization by directly classifying each row of pixels in an image as either a lane marker or a background [15]. By operating on a row-by-row basis, this method can rapidly process images in real-time, making it suitable for time-sensitive applications. However, it is necessary to capture fine-grained details of lane markers, such as subtle changes in lane widths and complex lane geometries.

Loshchilov and Hutter [16] uniformly applied a single weight decay factor to all the model parameters during training. This approach can inadvertently penalize both the magnitude and direction of weights, leading to suboptimal performance. The regularization impact on the model performance was optimized [11] to achieve more flexibility in adjusting the regularization strength for different parameters. The performance and effectiveness of a Digital Wide Dynamic Range (DWDR) may vary depending on the model's architecture and the specific learning task. He et al. [17] proposed a residual learning framework that allows the network to focus on learning the difference between the input and the desired outputs. The network can skip over one or more layers and pass the input directly to the subsequent layers. The network can effectively learn residual mapping by propagating the original input through these blocks. The network can efficiently propagate

gradients and avoid the vanishing gradient problem, leading to better optimization and higher accuracy. Understanding the internal representations and decision-making processes of deep residual networks is challenging.

Tabelini et al. [15] leveraged the power of deep neural networks to estimate lane boundaries by using polynomial regression functions. The annotated images were then used to train the Polylanenet model. In the deep polynomial regression stage, the model learns to predict polynomial coefficients that represent lane boundaries. Errors or inaccuracies in annotations can affect the ability of the model to accurately estimate lane boundaries. Huang et al. [18] leveraged deep learning techniques to regress 3D anchors for lane boundaries directly from monocular images. The ability to estimate 3D lane information from a single monocular camera eliminates the need for additional sensors or stereo-vision setups. It may be sensitive to noisy or ambiguous input. Poor image quality, low-resolution cameras, and challenging weather conditions can affect model performance. Estimating accurate depth information using a single camera remains a challenging task.

Huang et al. [18] introduced a novel monocular 3D lane detection method that eliminated the need for bird-eye-view (BEV) transformations. Defining 3D lane anchors and extracting features directly from front-view (FV) representations incorporates structural and contextual information to improve the prediction accuracy. The method also employs a global optimization technique to reduce the lateral error in the predictions. However, this assumes Flat Ground, leading to inaccuracies in estimating lane positions, particularly in real-world scenarios where road surfaces may be partially flat. Lee and Liu [19] proposed a lightweight design using depth-wise separable convolutions (DSUNet) for end-to-end learning of lane detection and path prediction (PP) in autonomous driving scenarios. A novel PP algorithm was also integrated with convolutional neural networks (CNN) to create a simulation model (CNN-PP) that facilitates qualitative, quantitative, and dynamic assessments of CNN performance in real-time autonomous driving scenarios. Although the study claimed that DSUNet is superior to UNet in terms of model size, inference speed, and performance metrics, it lacks a comprehensive comparative analysis with other existing methods or architectures for lane detection and path prediction in autonomous driving.

III. METHODOLOGY

The notations used in the rest of the paper is listed in Table 1.

TABLE 1. List of notations used.

Symbol	Meaning
x, y, θ	Coordinates of the lane
H	Height of an image
X	Convolutional layer input
M	Convolution area
W	Element in the convolution kernel
m , n	Size of the convolution kernel
a	Offset
$f(\cdot)$	Activation function
y	Element within pooled region
$down(\cdot)$	Downsampling
w	Weight
b	Bias
Z	Fully Connected layer input
$\{L_0, L_1, L_2\}$	Layers of Feature Pyramid Network
R_0, R_1, R_2	Refinements
P_t	Lane's prior attributes
T	Total number of refinements
N_p	Number of points in lane prior
φ	Normalized softmax function
C_{cls}	Focal cost of predictions and labels
C_{sim}	Similarity cost
C_{dis}	Average pixel distance of all valid lane points
C_{xy}	Distance of the initial point coordinates
C_θ	Discrepancy in angle θ
L_{cls}	Focal loss computed between predictions and labels
L_{xytl}	Smooth ℓ_1 loss
L_{LIoU}	Line IoU Loss
e	Expanded radius
C_{clip}, S_{clip}	Counts of valid points and ground truth points

A. LANE REPRESENTATION

Lanes are narrow and lengthy, with solid form priors; therefore, a preset lane prior can assist the network in better localizing lanes. Rectangular boxes often represent objects in standard object detection [20]. Nonetheless, the box must be used adequately to describe a long line. We depict lanes with similarly spaced 2-D points, that is, $P = \{(x_1, y_1) \ldots (x_N, y_N)\}$ where x and y are the coordinates of the lane. A *lane* is defined as a set of attributes. Our study employs a Lane Prior representation [20] where the y-coordinates of points are uniformly sampled vertically across the image, that is, $y_i = \frac{H}{N-1} * i$. This representation establishes a linkage between the x-coordinate and the corresponding $y_i \in Y$. Our system anticipates each lane based on four components: 1) probabilities of foreground and background, 2) the length of the lane before, 3) the starting point of the lane line and the angle between the x-axis of the Lane Prior representation (x, y, θ), and 4) it utilizes N offsets that represent the horizontal distance between the predicted lane and its ground truth.

B. CROSS LAYER REFINEMENT

In neural networks, deep high-level features exhibit strong responses to complete objects and possess higher semantic interpretations. In contrast, shallow low-level features capture the local contextual information. Additional valuable contextual information, such as lane lines or landmarks, can be extracted by enabling lane objects to access high-level features. Fine-detail characteristics are crucial for achieving an excellent localization accuracy in lane detection. For object detection, a feature pyramid is constructed to leverage the pyramidal nature of the ConvNet feature hierarchy. This approach assigns different object sizes to different levels of a feature pyramid. However, assigning lanes to multiple classes presents challenges because high and low-level qualities are crucial for accurate lane detection.

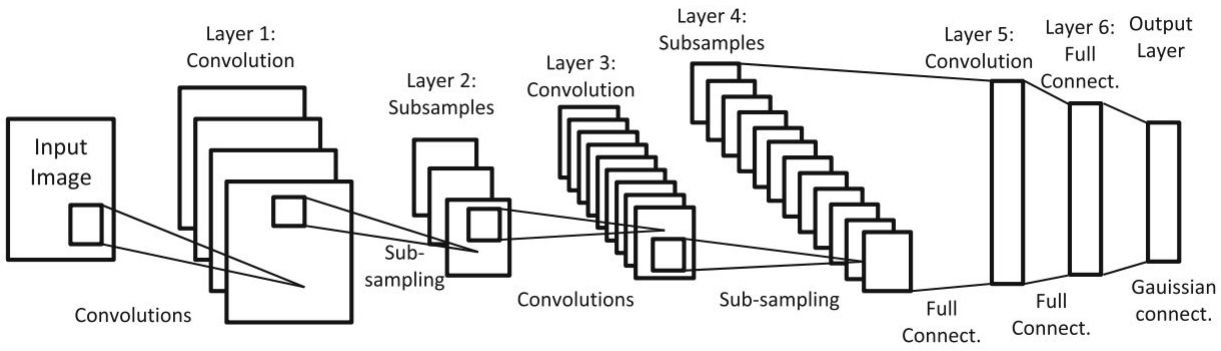

FIGURE 2. LeNet architecture [21].

FIGURE 3. AlexNet architecture [22].

1) LeNet-5 ARCHITECTURE

LeNet-5, a specific neural network architecture, addresses the challenge of extracting high-level and low-level features by assigning lane objects to all levels of the feature pyramid. It enables the sequential identification of lanes by taking advantage of high and low-level features. As shown in Fig. 2, in the initial phase, the convolutional layer was succeeded by the average pooling layer in the second phase. This pattern recurs, with a convolutional layer in the third stage and an average pooling layer in the fourth stage. The subsequent phase encompasses a flattening convolutional layer, followed by two fully-connected layers, culminating in a softmax classifier. In the first phase, the primary convolutional layer accepts the input as a grayscale image measuring 32 × 32, containing six feature maps with dimensions 5 × 5. The stride employed was one, yielding a transformation of the image dimensions from 32 × 32 × 1 to 28 × 28 × 6. The average pooling layer uses a filter with dimensions 2 × 2 and a stride of 2. This resulted in the image being resized to 14 × 14 × 6. It then proceeds to the subsequent convolutional layer, featuring 16 distinct maps, each sized 5 × 5, with a stride of 1. However, of these 16 maps, only ten were connected to the six maps of the preceding layer. This strategy was implemented to effectively manage the number of connections and introduces network asymmetry. Consequently, the training parameters were reduced to 1,516 from 2,400, and the connections were decreased to 151,600 from 240,000.

The fourth layer also comprises an average pooling operation with a stride of two and a 2 × 2 filter. Although mirrors the second layer, they accommodate 16 feature maps, resulting in a 5 × 5 × 16 output. Subsequently, the fifth layer was established with 120 feature maps, each of size 1 × 1, constituting a fully connected convolutional layer. These 120 features are linked to 400 nodes in the fourth layer. The sixth layer is comprised of 84 units and operates as a fully connected layer. Ultimately, the final output layer, which is also a fully connected layer, produces ten values spanning from 0 to 9.

2) AlexNet ARCHITECTURE

As shown in Fig. 3, AlexNet [22] comprises eight weight layers, encompassing five convolutional layers and three fully-connected layers. The architecture incorporates three max-pooling layers following the first, second, and fifth convolutional layers. The initial convolutional layer employs 96, 11 × 11 filters with a stride of 4 pixels and 2-pixel padding. The subsequent convolutional layers maintain a stride and padding of one pixel. The second convolutional layer contained 256 filters, each sized 5 × 5. The third, fourth, and fifth convolutional layers comprised 384, 384, and 256 filters, respectively, with dimensions 3 × 3.

3) COMBINED ARCHITECTURE OF LeNet-5 AND AlexNet (LeAlexNet)

In Fig. 4, the initial convolutional layer convolves the 224 × 224 × 3 input image using 96 kernels of size 11 × 11 × 3, with a stride of four pixels (which represents the spacing between neighboring receptive field centers). The second convolutional layer processes the (response-normalized and pooled) output from the first convolutional layer, utilizing 256 kernels of size 5 × 5 × 48. The third, fourth, and fifth convolutional layers were sequentially connected without intervening pooling or normalization layers. The third convolutional layer involves 384 kernels of size 3 × 3 × 256, linked to the (normalized, pooled) outputs of the second

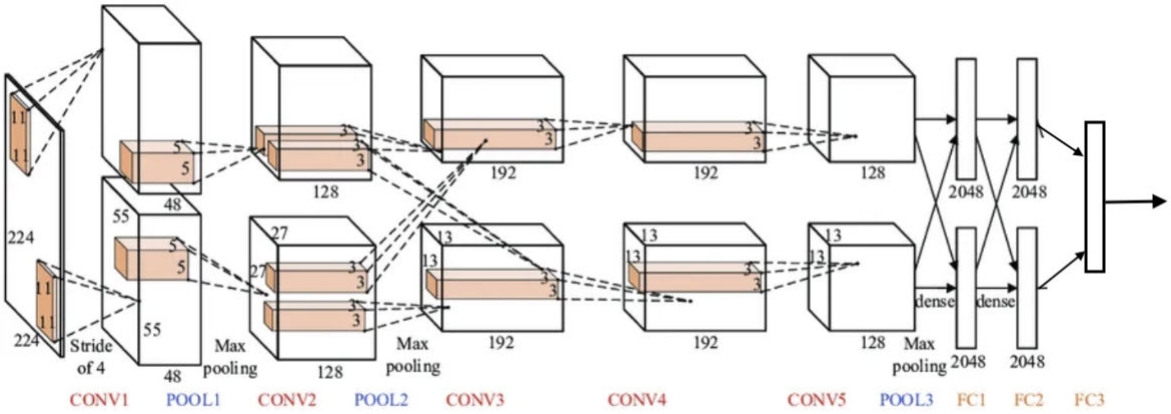

FIGURE 4. An overview of the proposed hybrid network (LeAlexNet). The main idea revolves around using AlexNet for more of a deeper architecture, and LeNet5 is used to extract features from the images specifically.

convolutional layer. Subsequently, the fourth convolutional layer employs 384 kernels of size $3 \times 3 \times 192$, whereas the fifth convolutional layer used 256 kernels of size $3 \times 3 \times 192$. Each fully connected layer contained 4096 neurons.

The proposed hybrid network (LeAlexNet) maximizes the multinomial logistic regression objective, which can be understood as maximizing the mean log probability of the correct label within the prediction distribution across all instances in the training dataset. One approach is to utilize high-level characteristics for initial lane detection, which helps obtain coarse localization of lanes using smaller neural networks. Once lanes are identified, they can be enhanced by incorporating more detailed features. To delve deeper into neural networks and leverage their capabilities, we can employ AlexNet architecture. This architecture is known for its deep design and is well-suited for complex feature extraction and pattern recognition tasks.

For convolutional layer input X, the convolution process is defined as:

$$con = f\left(\sum_{i,j\in M} X_{ij} * W_{m-i,n-j} + a\right) \tag{1}$$

where x represents the element in the convolution area M of the input X, W represents the element in the convolution kernel, m and n represent the size of the convolution kernel, a represents the offset, and $f(\cdot)$ represents the activation function.

For the pooling layer input, the pooling process is defined as follows:

$$pool = down(max(y_{i,j})) \quad i, j \in p \tag{2}$$

In this context, y denotes the element within the pooled region p in input Y of the pooling layer. The process denoted by '$down(\cdot)$' indicates downsampling and preserves the highest values within the pooled region.

In the case of the input Z for the fully connected layer, z represents an element within this input. w is the weight, b is the bias term, and $f(\cdot)$ is the activation function applied to element z.

$$full = f(w \times z + b) \tag{3}$$

4) REFINEMENT STRUCTURE

We aimed to utilize the hierarchical arrangement of features within a ConvNet, which encompasses meanings spanning from the basic to advanced levels, and construct a feature pyramid encompassing profound semantic understanding across all levels. We used the ResNet architecture as the backbone of our architecture, where we used $\{L_0, L_1, L_2\}$ in our Feature Pyramid Network. As shown in Fig.5, our approach to cross-layer refinement commences at the topmost level, L_0, and progressively approaches L_2. We have used R_0, R_1, R_2 as the corresponding refinements.

$$P_t = P_{t-1} \circ R_{t-1}(L_{t-1}, P_{t-1}) \quad t = 1, \ldots T \tag{4}$$

where $\circ$ denotes the combination of two layers and T denotes the total number of refinements. Our approach initiates detection from the uppermost layer, thereby providing significant semantic information. Parameter P_t represents the lane's prior attributes, encompassing starting point coordinates (x, y, θ). For the initial layer, L_0, the parameter P_0 are distributed uniformly across the image plane. The enhancement process denoted as R_t, employs P_t to obtain region-of-interest (ROI) lane features, followed by two fully connected (FC) layers to generate the refined parameter P_t. Step-by-step improvement of the lane prior and feature extraction play a pivotal role in ensuring the effectiveness of the cross-layer refinement technique.

C. ROI GATHER

1) ROI ALIGN

After assigning lane priors to each feature map, we used ROIAlign [23] to retrieve the lane-prior features. However, the contextual information provided by these traits needs to be more comprehensive. The lane instance may be occupied

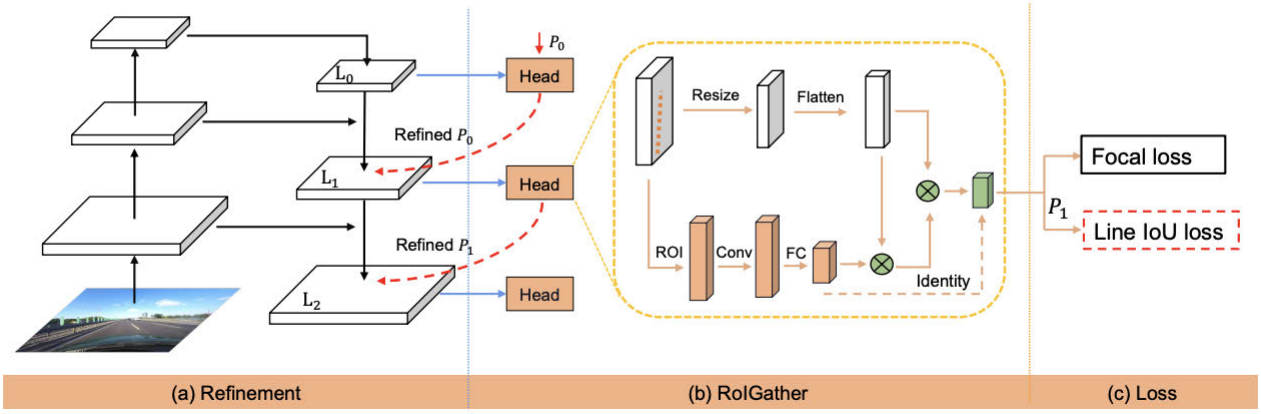

FIGURE 5. Overview of CLRNet architecture [10]: (a) The network begins by generating feature maps using the Feature Pyramid Network (FPN) structure [1]. These feature maps serve as the foundation for lane refinement, starting from high-level features and gradually transitioning to low-level features. (b) Each lane instance is refined using multiple heads, incorporating additional contextual information to enhance lane prediction. (c) The network performs both classification and regression of lane priors. (d)The suggested Line Intersection over Union (Line IoU) loss optimizes the regression performance. This loss function aids in further improving the accuracy of lane regression.

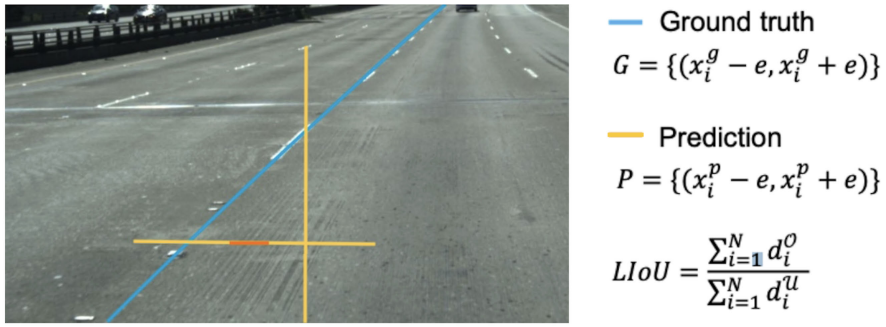

FIGURE 6. Line IoU illustration. Integrating the IoU of the extended segment in terms of sampling x_i location yields LIoU. Here, the horizontal yellow line depicts the union(d_i^u), and within this, the orange line depicts the Intersection(d_i^o). The Blue line represents the Ground Truth lane points, and the vertical yellow line represents the predicted lane points.

or indistinct in rare circumstances due to poor illumination. As a result, there were no visible indications of lane presence. Neighboring characteristics must examined to determine whether a pixel belongs to a lane. We can collect more relevant contextual information to better understand lane characteristics. To that goal, previous convolutions were added along the lanes. In the proposed method, each pixel within the preceding lane can gather information from the surrounding pixels. It allows lane representation to be reinforced in occupied regions based on the knowledge acquired from the surrounding context [24]. Additionally, relationships were established between the features of the preceding lane and the entire feature map. By doing so, the model can leverage more contextual data and develop more robust and comprehensive feature representations for lane detection.

2) ROI GATHER STRUCTURE

The ROIGather component is designed to be lightweight and simple to integrate. It receives a feature map and lane priors as input, where each lane prior comprises N points. For every individual lane prior, we employ the ROIAlign technique to extract the corresponding ROI feature (represented as $X_p \in \mathbb{R}^{C \times N_p}$). In contrast to the ROIAlign method employed for bounding boxes, we adopted a distinct approach by uniformly selecting N_p points from the lane prior. These points were used with bilinear interpolation to accurately determine the input feature values at these positions. For the ROI features originating from layers L_1 and L_2, we combined the ROI features from prior layers to amplify the feature representations. Convolutions were subsequently conducted on these extracted ROI features, allowing us to gather neighboring features for each lane pixel. We leverage a fully connected operation to extract the lane further prior to the feature $X_p \in \mathbb{R}^C$ to manage the memory resources. The feature map was resized to $X_f \in \mathbb{R}^{C \times H \times W}$ and flattened to $X_f \in \mathbb{R}^{C \times HW}$.

To gather the global context for the features of lane priors, we first compute the attention matrix W between the ROI lane prior features (X_p) and the global feature map (X_f) [25],

which is written as:

$$W = \varphi\left(\frac{X_p^T X_f}{\sqrt{C}}\right) \tag{5}$$

where φ is the normalized softmax function. The aggregated features are expressed as follows:

$$G = WX_f^T \tag{6}$$

Result G represents the additional contribution of X_f to X_p, chosen from all locations within X_f. Eventually, we combine this output with the original input X_p.

D. LINE INTERSECTION OVER UNION LOSS

As stated previously, the lane prior comprises distinct points that must be regressed against the ground truth. These points can be regressed using a frequently used distance loss, such as smooth-ℓ_1. Traditional distance-based loss functions treat individual points as independent variables, which oversimplifies the lane regression process and can result in less accurate predictions. Instead, the IoU metric [26] offers a more comprehensive approach. The IoU considers the lane as a whole unit and evaluates the overlap between the predicted and ground truth lanes. Using IoU as a loss function, the model can regress the entire lane and optimize its performance based on assessment metrics, leading to improved accuracy in lane regression tasks. For every point within the projected lane, as depicted in Fig. 6, we start by elongating it (x_i^p) by an extent of e to form a line segment. The Line Intersection over Union (LIoU) loss function is a metric used in lane detection tasks, particularly in autonomous driving systems. It measures the accuracy of the predicted lane lines by comparing them with the ground truth lane lines. Subsequently, IoUs can be computed between the elongated line segment and its corresponding ground truth, and this calculation is expressed as follows:

$$IoU = \frac{d_i^o}{d_i^u} = \frac{min(x_i^p + e, x_i^g + e) - max(x_i^p - e, x_i^g - e)}{max(x_i^p + e, x_i^g + e) - min(x_i^p - e, x_i^g - e)} \tag{7}$$

where, $x_i^p + e$ and $x_i^p - e$ represent the extended boundaries of x_i^p, and $x_i^g + e$ and $x_i^p - e$ signify the related ground truth points. It is important to note that d_i^o can take on negative values, enabling optimization to occur even when dealing with non-overlapping line segments. Subsequently, the LIoU can be perceived as the amalgamation of an infinite array of points along a line. To streamline the equation and facilitate its calculation, it is converted into a discrete formulation.

$$LIoU = \frac{\sum_{i=1}^{N} d_i^o}{\sum_{i=1}^{N} d_i^u} \tag{8}$$

The LIoU loss is defined as:

$$L_{LIoU} = 1 - LIoU. \tag{9}$$

This loss function is minimized during training to improve the alignment between the predicted and ground truth lines. Here, LIoU converges to -1 when the two lines are far apart. Our research considers a straightforward and efficient method for calculating LIoU loss. The LIoU loss offers two key advantages:

1) *Simplicity and Differentiability*: The LIoU loss is designed to be simple and differentiable, allowing for efficient parallel calculations during training. This ensures that the loss can be easily incorporated into existing optimization frameworks.
2) *Treating Lanes as Whole Units*: By considering the lane as a complete unit, the LIoU loss enables the model to anticipate and optimize the overall performance of lane detection.

This holistic perspective improves the accuracy and effectiveness of the lane detection algorithm. Overall, the LIoU loss function offers a straightforward and practical approach for optimizing lane detection models, combining simplicity and the ability to treat lanes as unified entities.

E. TRAINING AND INFERENCE

In the training phase, each ground truth lane dynamically selects and assigns one or more projected lanes as positive samples. This selection is determined by assigning cost, which quantifies the cost or suitability of assigning a projected lane to a ground truth lane, which is defined as:

$$C_{assign} = w_{sim}C_{sim} + w_{cls}C_{cls} \tag{10}$$

$$C_{sim} = (C_{dis}C_{xy}C_{\theta})^2 \tag{11}$$

where C_{cls} is the focal cost of the predictions and labels, and C_{sim} represents the similarity cost between the projected lanes and the actual ground truth. C_{sim} comprises three segments: C_{dis} denotes the average pixel distance of all valid lane points, C_{xy} denotes the distance of the initial point coordinates, and C_{θ} indicates the discrepancy in angle θ. These factors were normalized within the range of [0, 1]. The coefficients w_{cls} and w_{sim} determine the relative importance of each defined component. Each ground truth lane is associated with a dynamic number (determined by the top-k) of the projected lanes based on the evaluation of C_{assign}. The details of the training loss and inference are as follows:

1) TRAINING LOSS

The training loss consists of classification and regression losses. Regression loss was calculated exclusively for the assigned samples. The total loss function is given by:

$$L_{total} = w_{cls}L_{cls} + w_{xytl}L_{xytl} + w_{LIoU}L_{LIoU} \tag{12}$$

where L_{cls} denotes the focal loss computed between the predictions and labels. Meanwhile, L_{xytl} stands for the smooth ℓ_1 loss applied to the starting point coordinate(x, y,θ), and lane length regression l. L_{LIoU} corresponds to the LIoU loss involving the predicted lane and actual ground truth. Additionally, there is an option to introduce auxiliary segmentation loss [13]. However, this auxiliary loss was

solely incorporated during training and did not affect the inference results.

2) INFERENCE

We employed non-maximum suppression (NMS) [20] to eliminate high-overlap lanes after setting a threshold with a classification score to filter the background lanes (low-scoring lane priors). Our technique can also be NMS-free if we utilize a one-to-one assignment, that is, set top-$k = 1$.

IV. EXPERIMENTS

A. DATASETS

We conducted experiments on three benchmark datasets that are commonly used for lane detection evaluation: CULane, TuSimple, and LLAMAS.

The CULane dataset [3] is a challenging large-scale dataset that is specifically designed for lane detection. It comprises three complex units: crowded scenes, nighttime images, and intersections categorized as normal, crowded, dazzle, shadow, no line, arrow, curvy, and night. The dataset contains 100 000 images, divided into training, validation, and test sets. Each image in the CULane dataset has a resolution of 1640×590 pixels.

The TuSimple dataset [27] is a widely used benchmark for lane detection, which focuses primarily on highway scenes. It comprises 3,268 training, 358 validation, and 2,782 test images. All images in the TuSimple dataset have a 1280×720 pixels resolution.

The LLAMAS dataset [4] is another large-scale dataset that is used for lane detection. It comprises over 100 000 images, and the lane markers in the dataset are accurately annotated using precise maps. However, the labels of the test set are not publicly available; therefore, we submitted our detection results to the LLAMAS benchmark testing website for evaluation.

These benchmark datasets provide diverse and representative samples for evaluating the performance of lane detection algorithms.

B. IMPLEMENTATION

Our experiments used pre-trained LeNet-5 and AlexNet architectures for our lane detection models. All input images were resized to dimensions of 320×800 pixels to ensure consistent input sizes. We applied a random affine transformations and horizontal flips to diversify the training data for data augmentation. The optimization process employed the AdamW optimizer [28] with an initial learning rate of $1e^{-}3$. We also utilized the cosine decay learning rate method [29] with a power of 0.9. The training process varied across the different benchmark datasets. We trained the models on the CULane dataset for 21 epochs, the TuSimple dataset for 56 epochs, and the LLAMAS dataset for 26 epochs. All experiments were conducted using a single GPU, utilizing the PyTorch framework. In our network architecture, we set the number of lane priors (N) to 72 and the sampled number (N_p) to 36. For the ROIGather module, the resized height (H) and width (W) were set to 10 and 25, respectively, with a channel size (C) of 64. The expanded radius (e) used in the Line IoU (LIoU) calculation was set to 15. The coefficients of assigning cost were set as $w_{cls} = 1$ and $w_{sim} = 3$, balancing the importance of classification and similarity in the loss function.

C. EVALUATION METRICS

F1 is an assessment metric for TuSimple, CULane, and LLAMAS. The IoU between forecasts and ground truth was determined. True positives (TP) are predicted lanes with an IoU greater than a certain threshold (0.5). F1 is defined as follows:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{13}$$

Building upon the COCO [30] detection metric, we have used the mF1 metric to better compare the localization performance of the different methods. The mF1 metric is defined as follows:

$$mF1 = \frac{(F1@50 + F1@55 + \cdots + F1@95)}{10} \tag{14}$$

where $F1@50, F1@55, \ldots, F1@95$ are $F1$ metrics when IoU thresholds are $0.5, 0.55, \ldots, 0.95$, respectively. It is a break from traditional methods, which reward detectors with better localization results. For the CULane dataset, to evaluate different conditions, the evaluation formula is:

$$Accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \tag{15}$$

where C_{clip} and S_{clip} represent the counts of valid points and ground truth points, respectively, within a given image [16]. In lane detection, a projected lane is considered accurate or proper if more than 85% of the predicted lane points fall within a 20-pixel distance from the corresponding ground truth points. The TuSimple dataset also provides false positive (FP) and false negative (FN) rates, further contributing to the evaluation of lane detection performance.

V. RESULTS AND DISCUSSIONS

A. PERFORMANCE ON CULane

We present the results of our proposed technique on the CULane lane detection benchmark dataset in Fig.7 and compare them with other commonly used lane detection algorithms. Our proposed approach achieved a new state-of-the-art performance on the CULane dataset, achieving an mF1 measure of 54.59. When considering the LeNet5 and AlexNet versions of our method, we obtained an F1@50 score of 78.23, which is slightly lower than that of CondLaneNet (ResNet101), but slightly higher than that of CondLaneNet (ResNet18). Notably, our method outperformed CondLaneNet (ResNet18) by 2.8% in terms of mF1, indicating that our approach outperformed lane positions with high localization accuracy. Our results were slightly

TABLE 2. Comparison of proposed algorithm with other state-of-the art lane detection methods for different scenarios on CULane dataset.

Method	Backbone	mF1	F1@50	Normal	Crowded	Dazzle	Shadow	No Line	Arrow	Curve	Night
SCNN [3]	VGG16	38.84	71.60	90.60	69.70	58.50	66.90	43.40	84.10	64.40	66.10
RESA [5]	ResNet50	47.86	75.30	92.10	73.10	69.20	72.80	47.70	88.30	70.30	69.90
FastDraw [4]	ResNet50	-	-	85.90	63.60	57.00	69.90	40.60	79.40	65.20	57.80
E2E [18]	ERFNet	-	74.00	91.00	73.10	64.50	74.10	46.60	85.80	71.90	67.90
UFLD [16]	ResNet18	38.94	68.40	88.70	66.00	58.40	62.80	40.20	81.00	57.90	62.10
PlNet [27]	HourGlass	46.81	74.40	90.30	72.30	66.30	68.40	49.80	83.70	65.20	67.70
LaneATT [24]	ResNet122	51.48	77.02	91.74	76.16	69.47	76.31	50.46	86.29	64.05	70.81
LaneAF [6]	DLA34	50.42	77.41	91.80	75.61	71.78	79.12	51.38	86.88	72.70	73.03
FOLOLane [9]	ERFNet	-	78.80	92.70	77.80	75.20	79.30	52.10	89.00	69.40	74.50
CondLane [2]	ResNet18	51.84	78.14	92.87	75.79	70.72	80.01	52.39	89.37	72.40	73.23
CondLane [2]	ResNet101	54.83	79.48	93.47	77.44	70.93	80.91	54.13	90.16	75.21	74.80
CLRNet [11]	ResNet18	55.23	79.58	93.30	78.33	73.71	79.66	53.14	90.25	71.56	75.11
CLRNet [11]	ResNet34	55.14	79.73	93.49	78.06	74.57	79.92	54.01	90.59	72.77	75.02
CLRNet [11]	ResNet101	**55.55**	80.13	**93.85**	78.78	72.50	82.33	54.50	89.79	75.57	75.51
CLRNet [11]	DLA34	55.64	**80.47**	93.73	79.59	75.30	82.51	**54.58**	90.62	74.13	75.37
CLRNet(ours)	LeNet5-AlexNet	54.59	79.77	92.75	**81.46**	**77.84**	**83.07**	53.76	**92.34**	**74.75**	**78.39**

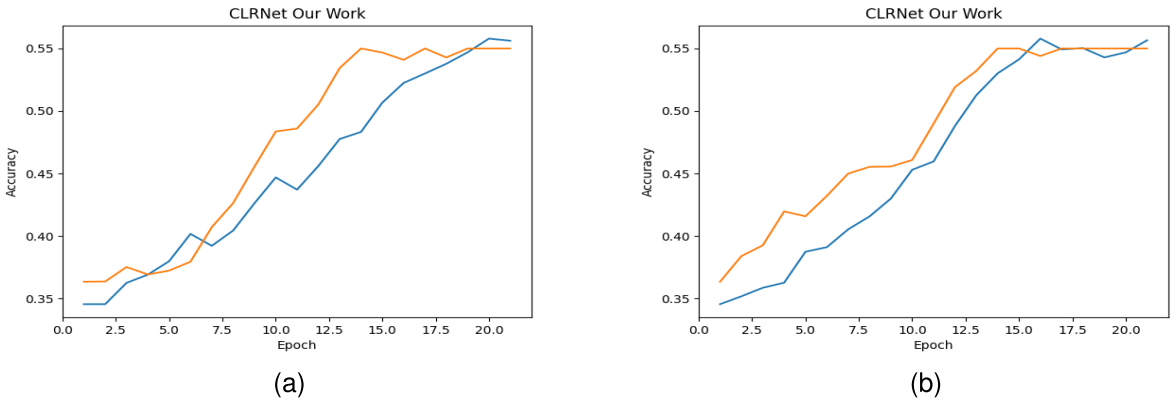

FIGURE 7. Lane detection accuracy performance on the CULane dataset without and with affine transformation applied.

lower than those of CLRNet (ResNet18) and CLRNet(ResNet 101) by approximately 0.8% mF1. Compared with the line anchor-based approach LaneATT [31], our LeNet5 and AlexNet versions achieved 7.18% mF1 and 3.45% F1@50 better, respectively. Meanwhile, CLRNet can attain 206 FPS with an NVIDIA 1080Ti GPU and TensorRT, making it suitable for real-time lane detection. The graph results for different conditions on the CULane dataset are shown in Fig.8.

Segmentation-based approaches, such as RESA, do not forecast lanes as a whole unit, thus limiting the smoothness of the lanes. Because the proposal only denotes one lane starting point, it is simple to overlook several lane instances. In these complicated settings, our technique can predict continuous and smooth lanes, demonstrating that it can acquire a global context and has great capacity to recognize correct lanes. Our approach is comparable to CLRNet, although it performs somewhat worse on this dataset.

Our method showed better results than CLRNet(the best of all architectures considered) under different conditions, as listed in Table 2. Additionally, we assessed these detectors and provided evaluations in terms of the mF1 and F1 scores at a threshold of 50 (F1@50). In the "Cross" category, only instances of false positives were presented. The metrics reported for these categories were established based on the F1 score at threshold of 50 (F1@50). Visualisation of lane detection results for different scenarios in the CULane dataset is shown in Fig.9.

B. PERFORMANCE ON LLAMAS

Our technique outperforms PolyLaneNet, LaneATT, and CLRNet on the test set by 7 F1@50, 2-3 F1@50, and 2.6 F1@50, respectively, as shown in Table 3. Although LaneAF [6] gets 96.90 F1@50 in the valid dataset, its inference performance could be better (approximately 20FPS), making deployment difficult. Furthermore, our technique outperforms LaneAF by approximately 2 mF1, demonstrating that our method is more accurate in terms of localization, as shown in Fig.10.

C. PERFORMANCE ON TuSimple

The performance gaps between the different approaches on this dataset were small, suggesting that the results have reached a high saturation level. However, our proposed

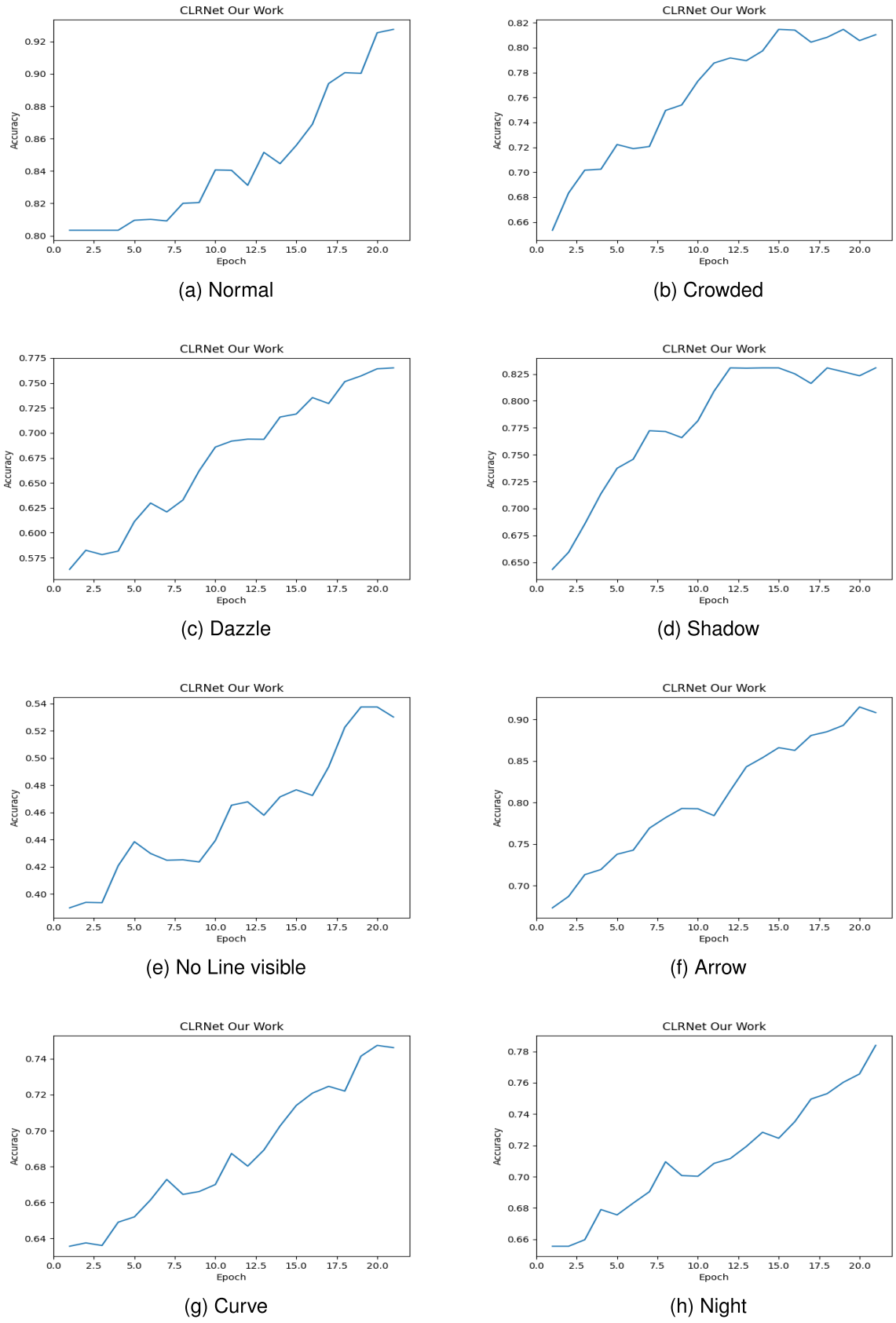

FIGURE 8. Accuracy performance of CLRNet(ours) using the combination of LeNet5 and ALexNet architecture on CULane dataset under different possible conditions that obstruct intelligent driving systems.

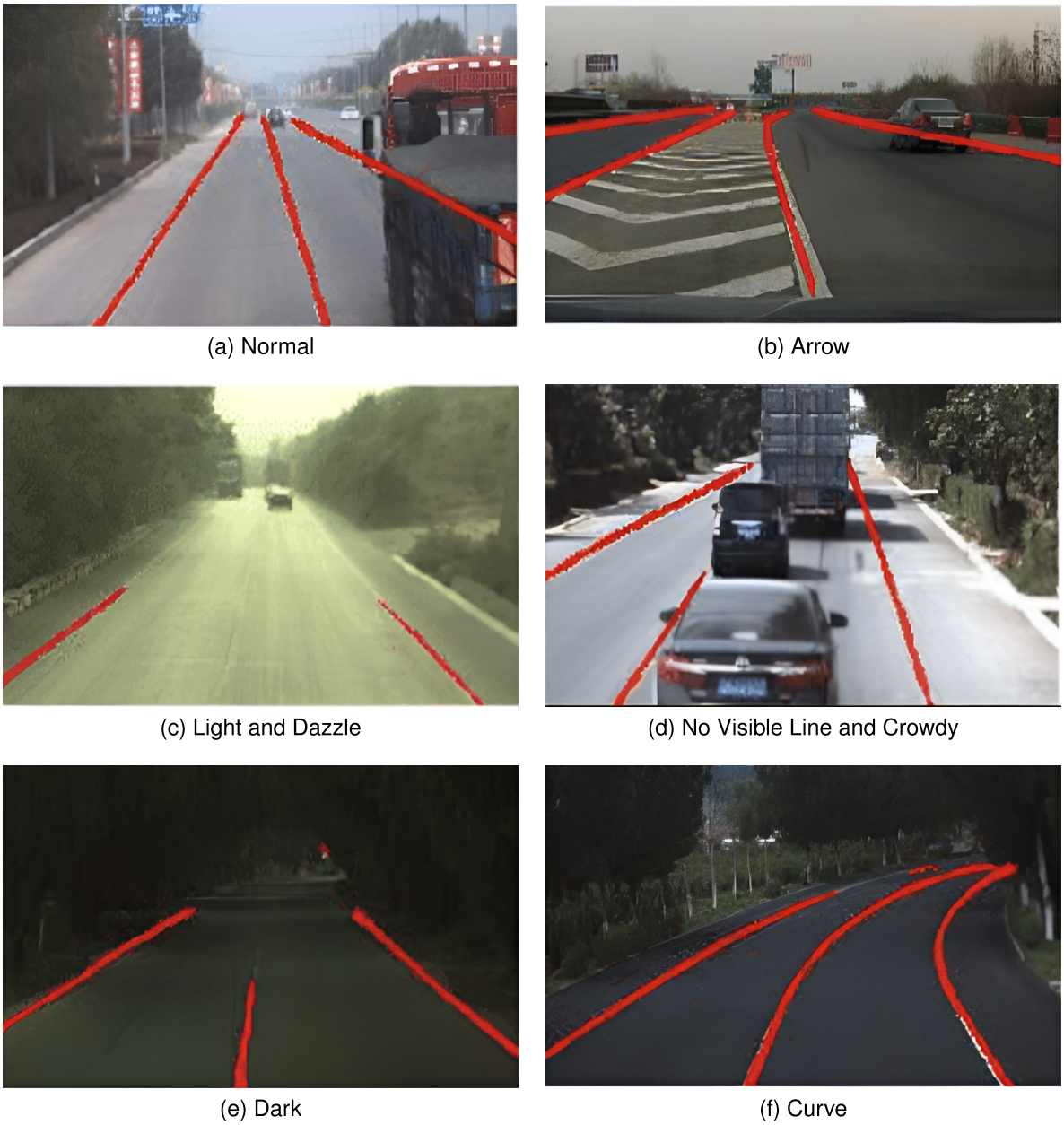

FIGURE 9. Visualisation of lane detection results for different scenarios in CULane dataset.

TABLE 3. Comparison of proposed algorithm with other state-of-the art lane detection methods on LLAMAS dataset.

Method	Backbone	mF1	F1@50	F1@75
PolyLaneNet [27]	EfficientNetB0	48.82	90.20	45.40
LaneATT [24]	ResNet34	69.24	94.62	82.38
LaneATT [24]	ResNet122	70.83	95.15	84.04
LaneAF [6]	DLA34	69.31	96.90	84.71
CLRNet [11]	ResNet18	71.61	96.96	85.59
CLRNet [11]	DLA34	71.21	**97.16**	85.33
CLRNet(ours)	LeNet5 and AlexNet	**73.48**	97.02	**91.74**

strategy still achieves a new state-of-the-art F1 score (see Fig.11, surpassing the previous performance of CLRNet (ResNet18) by 0.8% as listed in Table 4.

D. ABLATION STUDY

We ran numerous tests using the CULane dataset to demonstrate the usefulness of the various components of the proposed technique.

1) OVERALL ABLATION STUDY

By incorporating the LIoU loss, Cross-Layer Refinement, and ROIGather into the LeNet-5 and AlexNet baselines, we observed gradual improvements in performance. Removing the LIoU loss leads to a modest increase in mF1 from 51.90 to 52.80, indicating improved localization accuracy. Additionally, the introduction of the Cross-Layer Refinement further enhanced mF1 to 54.09. These improvements were

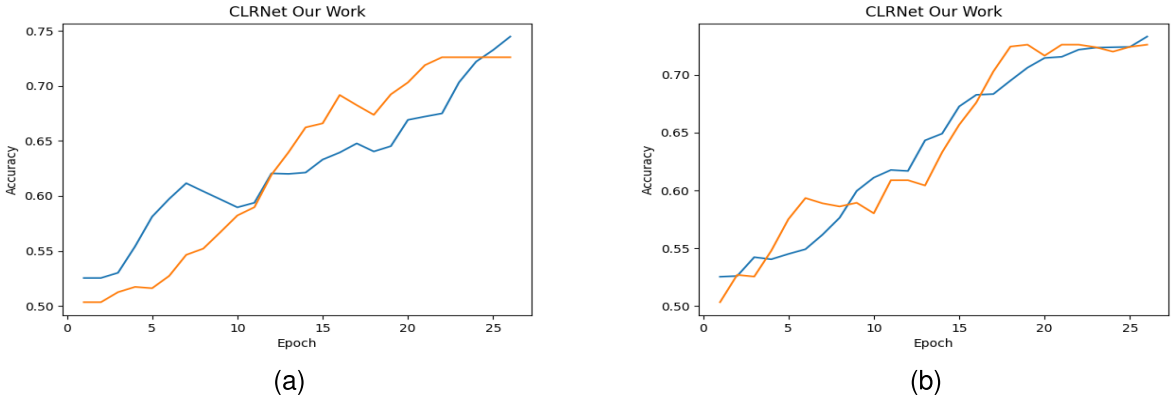

FIGURE 10. Lane detection accuracy performance on the LLAMAS dataset without and with affine transformation applied.

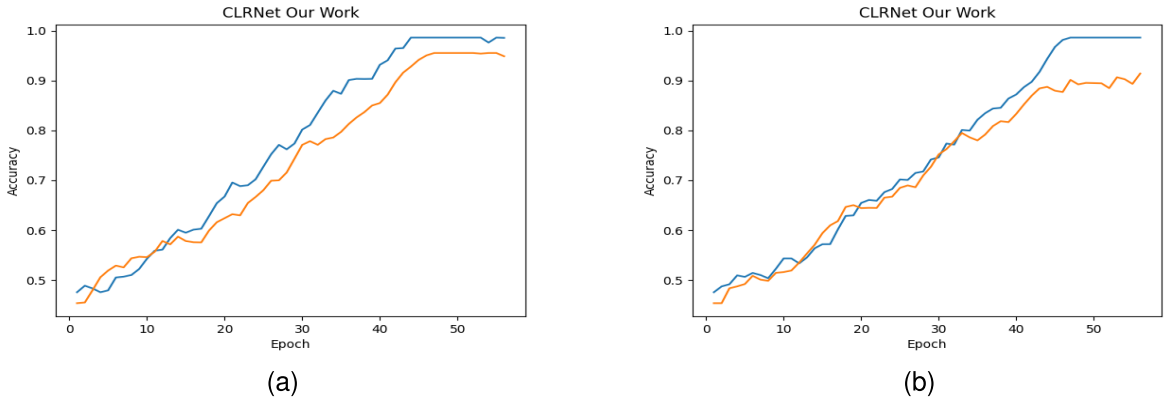

FIGURE 11. Lane detection accuracy performance on the TuSimple dataset without and with affine transformation applied.

TABLE 4. Comparison of proposed algorithm with other state-of-the art lane detection methods on TuSimple dataset.

Method	Backbone	F1	Acc	FP	FN
SCNN [3]	VGG16	95.98	96.52	6.19	**1.78**
RESA [5]	ResNet34	96.93	96.82	3.63	2.48
PolyLaneNet [27]	EfficientnetB0	90.64	93.33	9.44	9.31
E2E [18]	ERFNet	96.23	96.02	3.21	4.28
UFLD [16]	ResNet18	87.89	95.82	19.05	3.90
LaneATT [24]	ResNet18	96.71	95.54	3.56	3.01
LaneATT [24]	ResNet122	96.05	96.08	5.64	2.14
FOLOLane [9]	ERFNet	96.59	96.92	4.47	2.28
CondLane [2]	ResNet18	97.05	95.24	2.16	3.82
CondLane [2]	ResNet101	97.22	96.53	**2.03**	3.52
CLRNet [11]	ResNet18	97.89	96.84	2.28	1.92
CLRNet [11]	ResNet34	97.84	96.83	2.27	2.08
CLRNet [11]	ResNet101	97.61	96.83	2.33	2.38
CLRNet(ours)	LeNet5-AlexNet	**98.65**	**97.17**	2.30	1.89

consistent across various evaluation metrics, such as mF1, F1@50, and F1@75, demonstrating the effectiveness of combining high and low-level semantic characteristics for accurate lane recognition. The inclusion of ROIGather also contributes to a 0.5% increase in mF1, highlighting the positive impact of incorporating a rich global context in improving lane feature representation.

2) ANALYSIS OF ROI GATHER

In the feature map, the brightness of the color indicates the magnitude of the weight, with brighter colors indicating higher weight. The proposed ROIGather mechanism demonstrates two key capabilities. First, it effectively gathers global context by incorporating rich semantic information from the entire feature map. Second, it can capture distinctive features of foreground lanes even in a scenario where occlusion is present. This visualization demonstrates the effectiveness of ROIGather in enhancing the model's understanding of the lane context and capturing important lane characteristics.

3) ANALYSIS OF CROSS-LAYER REFINEMENT

The detector was constructed using a single layer to initiate the refinement process. The results obtained from this initial setup, denoted as R_0, R_1, R_2, show comparable performance. R_2 achieved a relatively high F1@75 score but a low F1@50 score, indicating that low-level characteristics play a crucial role in accurately regressing lanes. However, relying solely on low-level information may result in erroneous detection owing to the loss of high-level semantic information. We start with the best-performing configuration, R_0, and progressively incorporate additional refinements. The improvement

observed from R_0 to R_1 is minimal, suggesting that alternative fusion feature strategies, such as combining all features, may be more effective in this case.

However, the refinement from R_0 to R_2 yields significant improvements, indicating that our cross-layer approach can better leverage the benefits of both high and low-level characteristics. In contrast, other fusion feature strategies, including the addition of all features together, show limited effectiveness. The superior performance of the refinement from R_0 to R_2 reaffirms the superiority of our cross-layer improvement approach, which effectively utilizes high-level and low-level characteristics to improve lane detection performance.

4) ANALYSIS OF LINE INTERSECTION OVER UNION LOSS

To determine the optimal weight for the smooth-ℓ_1 regression loss, we conducted experiments by adjusting the loss weight. Initially, when the regression weight was set to 1.5, the smooth-ℓ_1 regression loss was significantly higher than that of the classification loss. However, the results showed improved performance after lowering the weight to 0.5. On the other hand, the LIoU loss exhibited more consistent results and led to an overall performance improvement of approximately 1 point in mF1. This improvement was particularly notable for measures with a high overlap, such as F1@80 and F1@90. These experimental findings demonstrate that our LIoU loss can enhance the model performance and promote convergence.

E. SCOPE OF THE WORK

Our research introduces a cutting-edge system and methodology for autonomously detecting lanes in vehicles, leveraging a sophisticated hybrid deep learning architecture using LeNet5 and AlexNet. The featured CLRNet exhibited a remarkable lane detection accuracy. This method initiates the lane detection in semantically rich regions, establishing initial lane localizations. Notably, it predicts the entire lane as a cohesive entity, thereby enhancing the overall smoothness of the detected streets. Moreover, the methodology excels at capturing multiple instances of roads, even in the complex scenarios. The proposed architecture consistently outperformed traditional algorithms across benchmark datasets and real-world scenarios, underscoring its robustness and efficacy in autonomous vehicle lane detection.

VI. CONCLUSION

Our research highlighted the crucial balance between high-level semantics and low-level characteristics when recognizing lanes, especially express lanes, in autonomous vehicle navigation systems. Although CNNs offer powerful feature representation, effectively using multiple feature levels remains challenging. We found that express lanes pose unique challenges, requiring a nuanced approach that blends high-level semantics with detailed low-level characteristics for accurate localization. By understanding the interconnected nature of landmarks and lane lines, we stress the importance of considering global context and high-level semantics to distinguish between them. At the same time, we acknowledge the essential role of local characteristics, especially considering the elongated and slender structure of roads. Our findings highlight the need to integrate high and low-level information within CNN architectures for precise lane recognition. Bridging the gap between theory and practice in express lane detection is crucial. By adopting a holistic approach that simultaneously considers local and global characteristics, researchers can pave the way for advancements in autonomous vehicle navigation. The integration of diverse feature levels is vital for creating safer and more efficient autonomous transportation systems. Our method, which combines high-level semantic properties with low-level lane-detecting characteristics through a CLRNet, has significantly improved localization accuracy. Our approach outperforms existing lane detection algorithms by leveraging the strengths of the LeNet-5 and AlexNet architectures (LeAlexNet), incorporating additional contextual information, and employing the LIoU loss function. This study contributes to the ongoing development of safer and more reliable autonomous transportation technologies, pushing forward the integration of autonomous vehicles into everyday transportation infrastructure.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the anonymous reviewers for their invaluable feedback and constructive comments. Their thorough and insightful reviews have significantly enhanced the quality of this manuscript. They also sincerely appreciate their time and effort in providing detailed critiques and suggestions, which have been instrumental in refining their work. Their expertise and dedication to the peer review process are greatly acknowledged.

REFERENCES

[1] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[2] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: A top-to-down lane detection framework based on conditional convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3753–3762.

[3] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.

[4] J. Philion, "FastDraw: Addressing the long tail of lane detection by adapting a sequential prediction network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11574–11583.

[5] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, "RESA: Recurrent feature-shift aggregator for lane detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3547–3554.

[6] H. Abualsaud, S. Liu, D. B. Lu, K. Situ, A. Rangesh, and M. M. Trivedi, "LaneAF: Robust multi-lane detection with affinity fields," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7477–7484, Oct. 2021.

[7] H. Li, W. Peng, S. Adumene, and M. Yazdi, "An improved lenet-5 convolutional neural network supporting condition-based maintenance and fault diagnosis of bearings," in *Intelligent Reliability and Maintainability of Energy Infrastructure Assets*. Cham, Switzerland: Springer, 2023, pp. 53–71.

[8] S. Lu, Z. Lu, and Y.-D. Zhang, "Pathological brain detection based on AlexNet and transfer learning," *J. Comput. Sci.*, vol. 30, pp. 41–47, Jan. 2019.
[9] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, "Focus on local: Detecting lane marker from bottom up via key point," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14117–14125.
[10] T. Zheng, Y. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai, and X. He, "CLRNet: Cross layer refinement network for lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 888–897.
[11] K. Behrendt and R. Soussan, "Unsupervised labeled lane markers using maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 832–839.
[12] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[13] Z. Qin, W. Huanyu, and X. Li, "Ultra fast structure-aware deep lane detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 276–291.
[14] S. Yoo, H. Seok Lee, H. Myeong, S. Yun, H. Park, J. Cho, and D. H. Kim, "End-to-end lane marker detection via row-wise classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 4335–4343.
[15] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "PolyLaneNet: Lane estimation via deep polynomial regression," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6150–6156.
[16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[18] S. Huang, Z. Shen, Z. Huang, Z.-H. Ding, J. Dai, J. Han, N. Wang, and S. Liu, "Anchor3DLane: Learning to regress 3D anchors for monocular 3D lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 17451–17460.
[19] D.-H. Lee and J.-L. Liu, "End-to-end deep learning of lane detection and path prediction for real-time autonomous driving," *Signal, Image Video Process.*, vol. 17, no. 1, pp. 199–205, Feb. 2023.
[20] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Keep your eyes on the lane: Real-time attention-guided lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 294–302.
[21] N. Kanagaraj, D. Hicks, A. Goyal, S. Tiwari, and G. Singh, "Deep learning using computer vision in self driving cars for lane and traffic sign detection," *Int. J. Syst. Assurance Eng. Manage.*, vol. 12, no. 6, pp. 1011–1025, Dec. 2021.
[22] W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao, and Y. Rui, "Visualizing and comparing AlexNet and VGG using deconvolutional layers," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1–7.
[23] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key points estimation and point instance segmentation approach for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8949–8958, Jul. 2022.
[24] X. Li, J. Li, X. Hu, and J. Yang, "Line-CNN: End-to-end traffic line detection with line proposal unit," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 248–258, Jan. 2020.
[25] V. Likhosherstov, K. Choromanski, and A. Weller, "On the expressive power of self-attention matrices," 2021, *arXiv:2106.03764*.
[26] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.
[27] Y. Yang, H. Peng, C. Li, W. Zhang, and K. Yang, "LaneFormer: Real-time lane exaction and detection via transformer," *Appl. Sci.*, vol. 12, no. 19, p. 9722, Sep. 2022.
[28] Z. Zhang, "Improved Adam optimizer for deep neural networks," in *Proc. IEEE/ACM 26th Int. Symp. Quality Service (IWQoS)*, Jun. 2018, pp. 1–2.
[29] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.
[30] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jul. 2020, pp. 237–242.
[31] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "CurveLane-NAS: Unifying lane-sensitive architecture search and adaptive point blending," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 689–704.

PRANAV CHAUDHARI graduated from Manipal Institute of Technology, MAHE, Manipal. He has worked as an AI Research Intern at the Neuromorphic Sensor Signal Processing Laboratory at the University of Strathclyde, Glasgow, Scotland. His research interests are deep learning, neural networks, and computer vision.

RAGHAVENDRA ACHAR (Member, IEEE) received the master's and Ph.D. degrees in computer science and engineering from the National Institute of Technology Karnataka, Surathkal, India. He is currently an Associate Professor with the Department of Information and Communication Technology, Manipal Institute of Technology. His research interests include cloud computing and web technologies.

SANJAY SINGH (Senior Member, IEEE) received the degree from the Institution of Electronics and Telecommunication Engineers, New Delhi, India, in 2001, and the M.Tech. and Ph.D. degrees from Manipal Institute of Technology, Manipal, India, in 2003 and 2010, respectively. In 2004, he joined the Department of Information and Communication Technology, Manipal Institute of Technology, MAHE, where he is currently a Professor. His research interests include machine learning, neural networks, deep learning, game theory, and natural language processing. He is a Senior Member of ACM.