Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 4, 2025 Serial no. 114
DOI: 10.22034/tjee.2025.66183.4982

Lane Detection in Presence of Occlusion using Deep Neural Network

A. Rahimi Bidmeshgi, A. Behrad*
Electrical Engineering Department, Faculty of Engineering, Shahed University, Tehran, Iran.
alireza.rahimi@shahed.ac.ir, behrad@shahed.ac.ir
*Corresponding author: A. Behrad

Received: 01/03/2025, Revised: 23/04/2025, Accepted: 08/06/2025.

Abstract
Lane detection is an important component in the development of autonomous vehicles, facilitating the real-time identification of driving paths and compliance with traffic regulations. Despite the promising performance of current models in controlled environments, they often encounter significant challenges in real-world scenarios, such as occluded lane visibility caused by snow, dust, traffic, or the absence of lane markings. This study presents a new approach to lane detection that leverages the spatiotemporal attributes of video frames by combining Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNNs) to enhance performance in the presence of occlusions. We employ a CNN to extract high-level spatial features from video frames, while an LSTM aggregates these features over time to model temporal dependencies and infer occluded segments when visual cues are absent. By representing lane marking detection as a sequential learning problem, the combined CNN-LSTM network effectively extracts spatiotemporal features. This dual architecture integrates both spatial and temporal information, thereby increasing robustness against occlusions and varying lighting conditions. The proposed model was evaluated under two conditions: low and high occlusion, using separate datasets, and was compared with the baseline architecture. The results confirm the effectiveness of the proposed approach. In low occlusion conditions, the model achieves an F1 score of about 96%, similar to the baseline method. In contrast, the baseline model suffers a performance drop in high occlusion scenarios, while the proposed model remains robust, also achieving an F1 score of about 96%.

Keywords
Lane detection, Deep models, LSTM, ResNet, High occlusion.

1. Introduction

Lane detection is crucial for Advanced Driver-Assistance Systems (ADAS), autonomous vehicles and surveillance system [1], enabling functions like lane keeping, lane departure warnings, and safe lane changes. However, real-time and accurate lane detection in real-world conditions is challenging, especially due to occlusions caused by other vehicles, pedestrians, adverse weather, or road obstructions. These occlusions can obscure lane markings, making detection difficult for both traditional algorithms and some advanced models.

Traditional lane detection techniques, such as edge detection [2] and Hough transform [3, 4] are effective in controlled environments but lack the robustness for dynamic and occluded scenarios. They often struggle to achieve real-time performance due to limited processing speed and reliance on simple geometric features, which are inadequate for the complex interpretations needed in occlusion situations.

On the other hand, modern lane segmentation models based on Convolutional Neural Networks (CNN) excel at capturing spatial features but frequently struggle to meet the real-time demands of lane detection [5]. The CNN models typically lack the temporal processing capabilities needed to accurately predict missing lane segments across frames and tend to be computationally intensive, further hindering their applicability in real-time contexts. Consequently, there is a pressing need for a lane detection model that can effectively manage both spatial and temporal complexities while satisfying real-time performance requirements.

To address these challenges, this paper proposes an integrated model that combines the spatial feature extraction of deep CNN architectures like ResNet18 and ResNet34 with the sequential processing of Long Short-Term Memory (LSTM) networks. This hybrid approach aims to enhance both the accuracy and real-time performance of lane detection in occluded conditions.

The primary objective of this research is to develop a robust and efficient lane detection model that maintains high performance in occluded conditions while achieving real-time processing speeds essential for ADAS and autonomous driving applications. By addressing spatial and temporal complexities, this model aims to deliver high-accuracy lane detection with rapid processing capabilities, thereby enhancing the reliability and safety of lane-based driving assistance systems.

The main objective of this research is to develop a robust and efficient lane detection model that performs well in occluded conditions while achieving the real-time processing speeds necessary for ADAS and autonomous driving applications. By addressing spatial and temporal complexities, this model seeks to provide high-accuracy

lane detection with fast processing capabilities, enhancing the reliability and safety of lane-based driving assistance systems.

This paper introduces a novel lane detection methodology that integrates CNN-based spatial feature extraction with LSTM networks to capture spatiotemporal attributes essential for handling occlusions. The CNN layers effectively extract detailed spatial information from video frames, while the LSTM network ensures continuity in inference across frames, enhancing resilience against partial lane visibility. Additionally, an anchor-driven ordinal classification method is leveraged [6] further increasing the model's detection speed without compromising accuracy.

2. Related work

According to the algorithm pipeline, existing approaches for lane detection can be divided into two categories: bottom-up and top-down modelling. In the top-down strategy, lane proposals are generated using the global information of the image and high-level features. Then, the representation of each lane may be refined through post-processing. The bottom-up methods first detect lane pixels or features. Then, these features or lane pixels are grouped or aggregated into lane candidates.

Traditional lane detection approaches primarily rely on low-level image processing techniques and a bottom-up methodology [6]. These methods typically utilize low-level visual cues from various color spaces, such as the HSI color model and edge extraction algorithms. One of the earliest efforts was by Bertozzi and Broggi, who used edge extraction in a stereo vision system to identify lanes and obstacles [7]. In addition to edge detection, some researchers have employed various color spaces, hough transform, projective geometry, and inverse perspective mapping to utilize prior knowledge that lanes are typically parallel in real-world scenarios [8-10]. However, despite exploring various traditional features, these methods generally struggle in complex environments due to their dependence on low-level semantic information [11, 12].

To enhance robustness, tracking has emerged as a popular post-processing solution [12]. Techniques like Markov models and Conditional Random Fields (CRFs) have also been used to improve lane detection accuracy [13-15]. Additionally, several methods have incorporated learning mechanisms such as template matching, decision trees, and support vector machines [16, 17].

With the development of deep learning, many methods based on deep neural networks have shown improved performance in lane detection tasks. These approaches typically model lane detection as a segmentation task, utilizing heat maps to indicate the presence and locations of lanes.

For example, the Vanishing Point Guided Network (VPGNet) introduces a multi-task segmentation network guided by vanishing points to detect lanes and road markings [18]. Sequential convolutional neural network (SCNN) improves pixel-wise segmentation through specialized convolution operations that aggregate information across dimensions [19]. Chng et al. enhanced SCNN by separately identifying and constructing straight and curved active lanes [20]. Similarly, Zheng et al. used a recurrent feature-shift aggregator to expand the receptive field for lane detection [21].

Deep segmentation approaches are typically computationally expensive and unsuitable for real-time applications. Consequently, some studies have explored mechanisms to reduce the computational complexity of deep models. Self-Attention Distillation (SAD) uses an attention distillation mechanism where higher layers serve as teachers for lower layers, which act as students [22]. Hou et al. leveraged inter-region affinity distillation to enhance the performance of the student network, allowing shallower architectures to achieve results comparable to deeper networks through attention mechanisms [23]. CurveLane-NAS introduces a deep neural architecture specifically tailored for segmentation in lane detection [24]. Lane-AF proposes a voting mechanism in affinity fields based on segmentation outputs for lane detection [25]. Qu et al. modeled local patterns to make global predictions using a global geometry decoder in a bottom-up approach [26].

In addition to the bottom-up segmentation approaches, several studies have explored alternative formulations for lane detection. In contrast to bottom-up approaches, top-down perspectives emphasize global information, making them especially beneficial in situations where visual cues are absent. For instance, Li et al. employed LSTM networks to effectively handle the elongated structure of lanes [27]. Philion proposed the Fast-Draw method, a convolutional model for lane detection that decodes lane structures without delegating the task to the post-processing stage [28]. Another group of methods treats lane detection as an instance segmentation problem by clustering binary segments. Hsu et al. proposed an end-to-end lane detection network utilizing differentiable least-squares fitting to directly predict the polynomial coefficients of lane curves [29]. Similarly, PolyLaneNet [30] and methods of [31, 32] propose predicting lane polynomial coefficients through deep polynomial regression and transformer models, respectively. LaneATT adopts an approach to lane detection by using an anchor-based deep model that, similar to other generic deep object detectors, utilizes anchors in the feature pooling step [33]. Based on the concepts of vanishing points and object detection frameworks, some methods have used line anchors guided by vanishing points to enhance lane detection accuracy [34, 35].

Some lane detection algorithms focus on the 3D localization of lane positions relative to the host vehicle, which is crucial for autonomous driving [36-40].

Qin et al. proposed the Ultra Fast Lane Detection (UFLD) approach, framing lane detection as an anchor-driven ordinal classification problem that utilizes global features [41]. They employed row and hybrid anchor representations to reduce learning complexity and accelerate detection speed. While they account for spatial consistency to address occlusion and lighting changes, their method does not consider temporal consistency.

In recent years, several algorithms have been proposed to improve lane detection accuracy [42, 43]; however, their robustness needs enhancement in challenging real-world conditions.

3. Proposed method

Real-world lane detection algorithms must be robust against challenging conditions such as occlusion, dust, rain, snow, unpaved roads, and varying traffic environments. To address these challenges, we propose a modified version of UFLD algorithm [38] that leverages both spatial and temporal features to achieve high accuracy, even in occlusion scenarios. Our method integrates CNN for spatial feature extraction and LSTM network for temporal modelling. This combination enables the system to retain historical context from previous frames and infer missing or obstructed lane features, making it more suitable for dynamic and complex driving environments with lane occlusion.

While UFLD performs well in normal driving scenarios, it struggles with occlusions, limiting its effectiveness for real-world autonomous driving. In contrast, our proposed method incorporates a temporal modeling component—an LSTM layer—into UFLD's hybrid anchor-driven ordinal classification framework. This enhancement enables the network to capture sequential dependencies between frames and maintain the temporal continuity of lane structures, which is crucial when visual cues are partially obscured due to occlusion, poor lighting, or abrupt environmental changes.

UFLD uses hybrid anchor-driven ordinal classification to reduce computational complexity and localization errors, whereas our approach enhances this framework by adding an LSTM layer to capture temporal dependencies between consecutive frames. This integration allows the model to retain historical context, improving performance in situations with limited visual cues, such as occlusions or low-light conditions.

The proposed method integrates an LSTM layer before the row-based selection process to enhance the temporal model's memory, enabling it to retain and utilize information from previous frames. This allows the model to better handle situations with limited visual cues, such as low-light conditions or lane obstructions. The LSTM layer captures sequential dependencies between frames, facilitating more accurate predictions of lane positions based on historical context. Consequently, this approach aims to improve lane detection accuracy and strengthen the model's resilience in challenging environments, leading to more reliable lane detection systems for real-time applications in autonomous driving.

Similar to UFLD, we use ResNet as the backbone for extracting high-level frame descriptors for lane detection. ResNet utilizes residual learning, enabling models to learn residual mappings instead of direct mappings. We specifically utilized ResNet18 and ResNet34 based on their performance in deep learning applications. ResNet18 is known for its simplicity and speed, making it ideal for rapid processing, while ResNet34 strikes a balance between depth and accuracy, achieving strong results across various benchmarks. Both models employ skip connections to improve gradient flow during backpropagation, aiding the learning of complex features while maintaining performance as network depth increases.

We integrate an LSTM layer after the ResNet neural network to enhance the model's ability to capture long-term contextual dependencies and maintain lane continuity. The recurrent nature of LSTM allows it to gather information from spatial locations across continuous frames, improving the model's ability to handle occlusions, loss of visual cues, and complex lane configurations. This enhanced contextual awareness can lead to more accurate lane predictions, especially in challenging scenarios.

By placing an LSTM on top of the ResNet feature extractor, the model effectively aggregates and refines the extracted visual features. The LSTM's gating mechanisms and memory cells allow it to selectively emphasize relevant features while suppressing irrelevant ones, resulting in a clearer representation of lane characteristics. This refined feature aggregation enhances the model's ability to distinguish lanes from the background, improving accuracy, recall, and F1 scores. The combination of ResNet and LSTM is particularly beneficial for handling lanes with varying orientations, such as side lanes, which are often more horizontal than primary lanes. While the row and column anchor system helps reduce localization error magnification, the LSTM layer further enhances the model's adaptability to different lane orientations. Its recurrent connections enable more effective modelling of spatial relationships between anchors, leading to more precise predictions for lanes with diverse orientations. This synergistic integration of ResNet and LSTM increases robustness against challenging conditions such as occlusions, poor lighting, and complex backgrounds.

By combining spatial features from ResNet and temporal features from LSTM, our method achieves high accuracy not only in normal driving conditions but also in occlusion scenarios. This is critical for real-world applications, as the proposed method must perform equally well in both occluded and non-occluded environments.

Fig. 1 shows the architecture of the proposed algorithm for lane detection. The proposed architecture processes a sequence of input images from a video stream by first feeding them into the spatial feature extraction layer, where a ResNet network extracts frame features as follows:

$$\mathbf{x}_t = \mathrm{Re}\,sNet(I_t) \tag{1}$$

Here, $\mathbf{x}_t$ represents the spatial features extracted from the t-th frame, I_t. The spatial feature map $\mathbf{x}_t$ is then reshaped into a suitable sequence for the LSTM network. The LSTM network processes the sequence of $\mathbf{x}_t$ to extract temporal dependencies and contextual information. The core structure of an LSTM includes a memory cell and three regulatory gates that manage information flow by controlling input, output, and retention within the memory cell. For an input vector $\mathbf{x}_t$ at time step t, the output or hidden state of the LSTM, $\mathbf{h}_t$, is calculated using the following equations [44]:

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i\right) \tag{2}$$

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f\right) \tag{3}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o\right) \tag{4}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \phi\left(\mathbf{W}_{xg}\mathbf{x}_t + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_g\right) \tag{5}$$

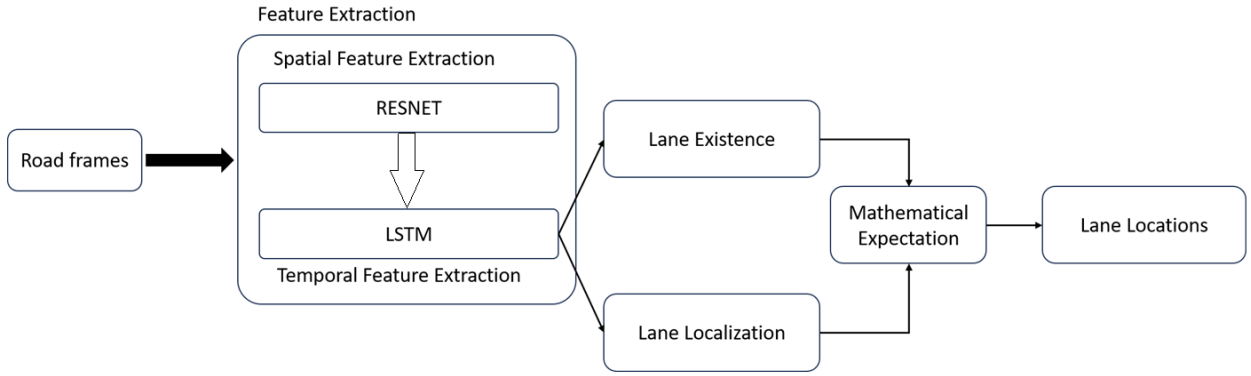

Fig. 1. Network Architecture. The architecture processes road frames through ResNet-based spatial feature extraction and LSTM-based temporal feature extraction. Features are passed to lane detection branches (existence and localization) and refined using an expectation-based approach for precise lane coordinate prediction and masking.

$$\mathbf{h}_t = \mathbf{o}_t \odot \phi(\mathbf{c}_t) \tag{6}$$

Here, $\mathbf{i}_t, \mathbf{f}_t$ and $\mathbf{o}_t$ represent the activation vectors for the input gate, forget gate, and output gate, respectively, while $\mathbf{c}_t$ denotes the cell state vector. Furthermore, $\mathbf{W}$ and $\mathbf{b}$ denote weight matrices and bias vectors, respectively, that are trained during the training phase of the network. Also, σ and ϕ define sigmoid and hyperbolic tangent functions, respectively, and $\odot$ specifies Hadamard or entry-wise product.

The output of LSTM layer is fed into the lane localization and existence branches to calculate the predictions of lane existences and their locations as follows [41]:

$$E^r, E^c = f^E(\mathbf{h}_t) \tag{7}$$

$$P^r, P^c = f^L(\mathbf{h}_t) \tag{8}$$

Here, f^E and f^L are classifier functions for the lane existence and localization branches implemented with multilayer perceptron and $\mathbf{h}_t$ is hidden state of the LSTM layer. The variables E^r and E^c represent predictions for the existence of row and column anchors, respectively, while P^r and P^c are the predictions for the localization branch [41].

The loss functions are identical to those in the original formulation [41], incorporating classification and structural losses. The main distinction between our proposed method and the baseline lies in our predictions leveraging spatiotemporal information from the LSTM layer, which helps maintain lane continuity across frames and potentially enhances overall accuracy in cases of occlusion.

Unlike traditional classification tasks, where classes are treated as independent and unrelated entities, lane detection involves classes that possess an inherent ordinal relationship. This means that adjacent classes represent closely related values, forming a continuum. By considering these ordinal relationships, the model can better capture the spatial relationships of lane coordinates and provide more precise predictions. The base classification loss based on the outputs of localization branch is defined as [41]:

$$\begin{aligned} L_{cls} = &\sum_{i=1}^{N^r_{lane}} \sum_{j=1}^{N_{row}} L_{CE}(P^r_{i,j}, onehot(T^r_{cls\_i,j})) \\ &+ \sum_{m=1}^{N^c_{lane}} \sum_{n=1}^{N_{col}} L_{CE}(P^c_{m,n,}, onehot(T^c_{cls\_m,n})) \end{aligned} \tag{9}$$

The base loss function employs the cross-entropy loss L_{CE} to measure the error between the actual anchor locations T^r_{cls} , T^c_{cls} and the predicted outcomes P^r , P^c . This loss is calculated for both row anchors (the first term) and column anchors (the second term), aggregating contributions from all relevant lanes and anchors.

To further improve the learning process, the expectation loss $L_{\exp}$ is utilized. This loss aligns the model's expected predictions with ground truths by computing a weighted sum of predicted probabilities for each class, similar to a voting mechanism. Minimizing this loss helps the model adjust its predictions, bringing the expected outputs closer to actual lane positions and enhancing localization accuracy. The expectation loss is defined as [41]:

$$\text{prob}^c_{i,j} = \text{softmax}(\text{P}^c_{i,j}) \tag{10}$$

$$\text{prob}^r_{i,j} = \text{softmax}(\text{P}^r_{i,j}) \tag{11}$$

$$Exp^r_{i,j} = \sum_{k=1}^{N^r_{\dim}} \text{Prob}^r_{i,j}[k].k \tag{12}$$

$$Exp^c_{m,n} = \sum_{l=1}^{N^c_{\dim}} \text{Prob}^c_{m,n}[l].l \tag{13}$$

$$\begin{aligned} L_{\exp} = &\sum_{i=1}^{N^r_{lane}} \sum_{j=1}^{c} L_1(Exp^r_{i,j}, T^r_{cls\_i,j}) + \\ &\sum_{m=1}^{N^c_{lane}} \sum_{n=1}^{N_{col}} L_1(Exp^c_{m,n}, T^c_{cls\_m,n}) \end{aligned} \tag{14}$$

Here, $L_{\exp}$ defines the expectation loss in the localization branch; P^r and P^c represent the row and column anchor predictions, respectively; and T^r_{cls} and T^c_{cls} are the

learning targets of the localization branch. Furthermore, $L_1(.)$ denotes the L1 loss function.

The existence loss L_{ext} is designed to determine the presence or absence of lanes. Using cross-entropy, it compares predicted lane existence with ground truth, ensuring the model not only predicts correct lane positions but also accurately identifies whether a lane exists at a given location. This component is essential for addressing scenarios where lanes may be absent or partially obstructed. The existence loss is defined as [41]:

$$L_{ext} = \sum_{i=1}^{N_{lane}^{r}} \sum_{j=1}^{N_{row}} L_{CE}(E_{i,j}^{r}, onehot(T_{ext\_i,j}^{r})) + \sum_{m=1}^{N_{lane}^{c}} \sum_{n=1}^{N_{col}} L_{CE}(E_{m,n}^{c}, onehot(T_{ext\_m,n}^{c})) \quad (15)$$

Here, L_{ext} defines the existence loss; L_{CE} denotes cross entropy function; and T_{ext}^{r} , T_{ext}^{c} represent the row and column learning targets in the existence branch, respectively.

The overall loss function L_T combines the base classification loss, expectation loss, and existence loss, each weighted by specific coefficients. This comprehensive loss enables the model to optimize multiple objectives simultaneously, ensuring it learns the ordinal relationships among classes while accurately predicting lane positions [41].

$$L_T = L_{cls} + \alpha L_{\exp} + \beta L_{ext} \quad (16)$$

where $L_{cls}, L_{\exp}$ and L_{ext} are base classification, expectation and existence losses, respectively and α, β are the loss coefficients

4. Experiments

4.1. Experimental Settings

4.1.1. Datasets

To train and test the proposed lane detection model, we used two widely adopted benchmark datasets: TuSimple [46] and CULane [47], each presenting distinct challenges and scenarios relevant to real-world driving. Table I outlines the key characteristics of the datasets.

The TuSimple dataset comprises 6,408 annotated images captured on U.S. highways. It features structured environments with clear lane markings and high visibility. The dataset includes one-second video clip recorded at 20 frames per second (FPS), with a resolution of 1280×720 pixels. Annotations are provided for every 13th frame, offering labelled lane positions at regular intervals. The dataset consists of daytime scenes under clear weather conditions, making it well-suited for training models in low-occlusion environments.

In contrast, the CULane dataset presents a more challenging and diverse evaluation setting. It comprises 133,235 frames captured across urban and rural areas in China, encompassing a wide variety of real-world driving conditions, including nighttime scenes, rain, snow, shadows, occlusions, and heavy traffic. The dataset also features complex scenarios such as intersections, curved roads, and lane-less sections, making it particularly valuable for evaluating the robustness of lane detection systems in highly dynamic and cluttered environments.

4.1.2. Evaluation Metrics

In lane detection, several evaluation metrics are used to quantitatively assess the performance of the proposed model, providing insights into its accuracy and effectiveness in detecting lane markings under various conditions. The main metrics include accuracy, F1 score, and Top-N accuracy. Accuracy measures the ratio of correctly predicted lane points to the total actual lane points in the dataset, serving as a direct indicator of the model's ability to accurately identify lane positions. The accuracy metric is defined as [5]:

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \quad (17)$$

Here, *Cclip* represents the number of correctly predicted lane points, while *Sclip* denotes the total number of lane points in each clip. Top-N accuracy measures how frequently the true label appears among the top N predictions, making it especially useful in multi-class applications.

The F1 score, which combines precision and recall, offers a balanced assessment of model performance, particularly in cases with imbalanced class distributions. The F1 score is defined as follows [5]:

$$F1 = \frac{2 \times Percision \times \mathrm{Re}call}{Percision + \mathrm{Re}call} \quad (18)$$

$$Percision = \frac{TP}{TP + FP} \quad (19)$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \quad (20)$$

These evaluation methods are crucial for a comprehensive assessment of the lane detection model's performance. Accuracy provides a clear measure of correct predictions, while the F1 score offers a balanced view between precision and recall. Top-N accuracy further enhances the analysis by evaluating the model's ability to identify correct labels among its top predictions. Together, these metrics form a robust framework for assessing the strengths and weaknesses of the proposed lane detection model, ensuring its reliability and effectiveness in real-world applications.

4.1.3. Implementation Details

All models were trained and evaluated using PyTorch on a GeForce 1070TI graphics card. The batch size for training was set to 16, and the models were trained for 30 epochs. We used the Adam optimizer with an initial learning rate of 0.1 for training. The ResNet backbone was initialized using pre-trained weights from ImageNet, whereas the remaining layers; such as the LSTM and

fully connected layers, employed the Xavier initialization method [45]. During training, the entire network was optimized end-to-end, enabling fine-tuning of all components, including the ResNet backbone.

The LSTM module consists of a single layer with a hidden size of 128, processing sequences of length 225 with 8-dimensional input vectors derived from the ResNet output. The input frames are resized to 288×800 pixels for the ResNet backbone. The ResNet output is taken from the output of main layer4, followed by a 1×1 convolution that reduces the number of channels to 8, yielding a feature map of shape (B, 8, 9, 25), where B is the batch size. This feature map is reshaped into a sequence of shape (B, 225, 8) (flattening spatial dimensions 9×25 into 225) and passed to the LSTM. The LSTM produces an output of shape (B, 225, 128), which is then flattened and fed into lane existence and lane localization branches.

The lane existence and lane localization branches employ two fully connected layers to transform the flattened LSTM output into the target dimensions by reducing feature size.

4.2. Results

This section presents the results of the proposed model for lane detection using two datasets: TuSimple and CULane. In the experiments, two models were utilized for spatial feature extraction: ResNet18 and ResNet34. Fig. 2 illustrates the F1 score for the test data of the TuSimple dataset across various training epochs. The figure displays the results of both ResNet18 and ResNet34 feature extractors for comparison. The key observations are as follows:

- **Initial Performance:** The proposed model with ResNet18 has a higher initial F1 score than ResNet34, indicating better lane detection performance in the early training stages for the shallower model.
- **Improvement Rate:** Both models show rapid improvement, but ResNet34 surpasses ResNet18 after the initial epochs. By epoch 5, the proposed model with ResNet34 matches and slightly outperforms ResNet18. After epoch 5, ResNet34 exhibits a faster growth rate.
- **Final Performance**: The proposed model with ResNet34 achieves a final F1 score of 0.975, outperforming the proposed model with ResNet18, which scored 0.9699. This small yet significant difference highlights the superiority of the deeper ResNet34 architecture when combined with LSTM for lane detection tasks.

Fig. 3 depicts the test loss curves of the proposed model for the TuSimple dataset over 30 epochs. The figure illustrates the loss curves for both ResNet18 and ResNet34 feature extractors for comparison. Both models exhibit steep decreases in loss during the initial epochs before stabilizing, suggesting effective convergence. Notably, the ResNet34-based architecture achieves a lower optimization loss compared to ResNet18.

To compare the effect of adding the LSTM layer in the proposed model, we also trained the UFLD [38] model using the TuSimple dataset in a similar condition to the proposed model. Fig. 4 shows the F1 score of the UFLD method for the test data of the TuSimple dataset across various training epochs. Similar to the proposed method, the figure displays the results for both ResNet18 and ResNet34 spatial feature extractors.

Similar to the proposed method, UFLD with ResNet18 starts with a higher initial F1 score compared to ResNet34, indicating that the shallower model is able to achieve a better balance between precision and recall in the early stages of training. Both ResNet18 and ResNet34 shows significant improvement in F1 scores throughout the training process. Although ResNet34 starts with a lower F1 score, it surpasses ResNet18 in later epochs and achieved a slightly higher final score.

Tabe I. Key characteristics of the TuSimple and CULane datasets

Dataset	Total Frames	Train Frames	Validation Frames	Test Frame	Resolution (pixels)	No. of Lanes	Environment
Tusimple	6408	3268	358	2782	1280x720	<5	highway
CULane	133235	88880	9675	34680	1640x590	<4	urban and highway

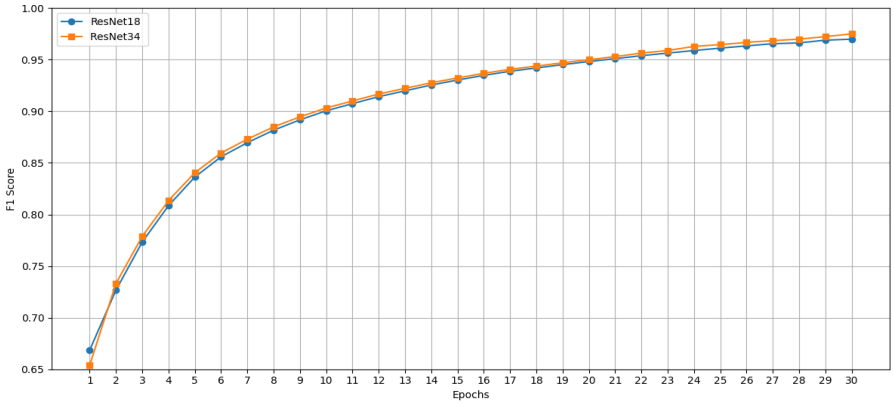

Fig. 2. F1 score for the test data of the TuSimple dataset across various training epochs.

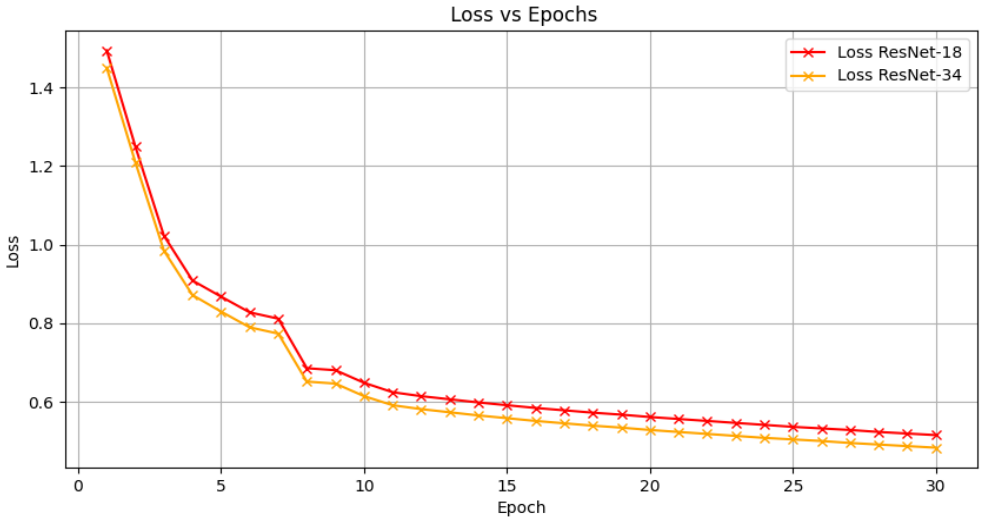

Fig. 3. The test loss curves of the proposed model for the TuSimple dataset over 30 training epochs.

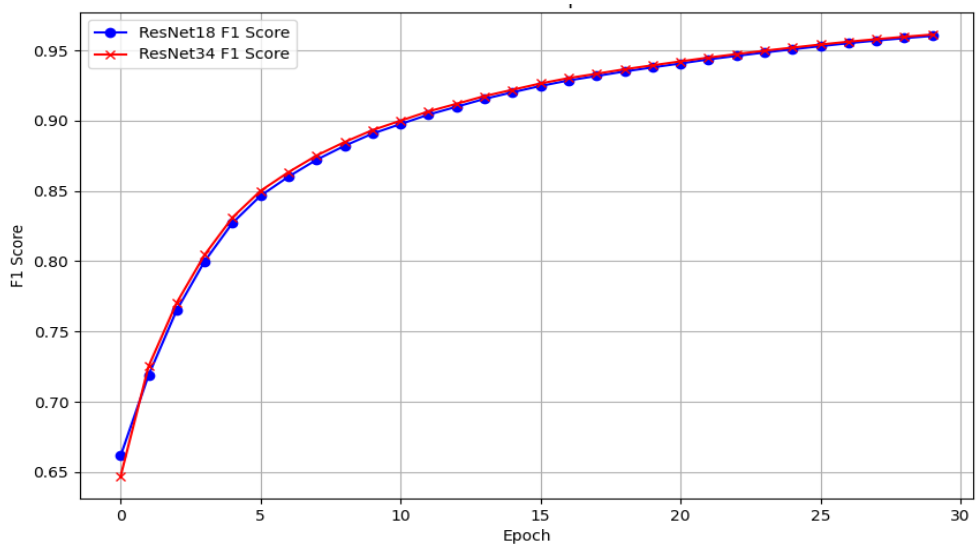

Fig. 4. The F1 score of UFLD method for the test data of the TuSimple dataset across various training epochs

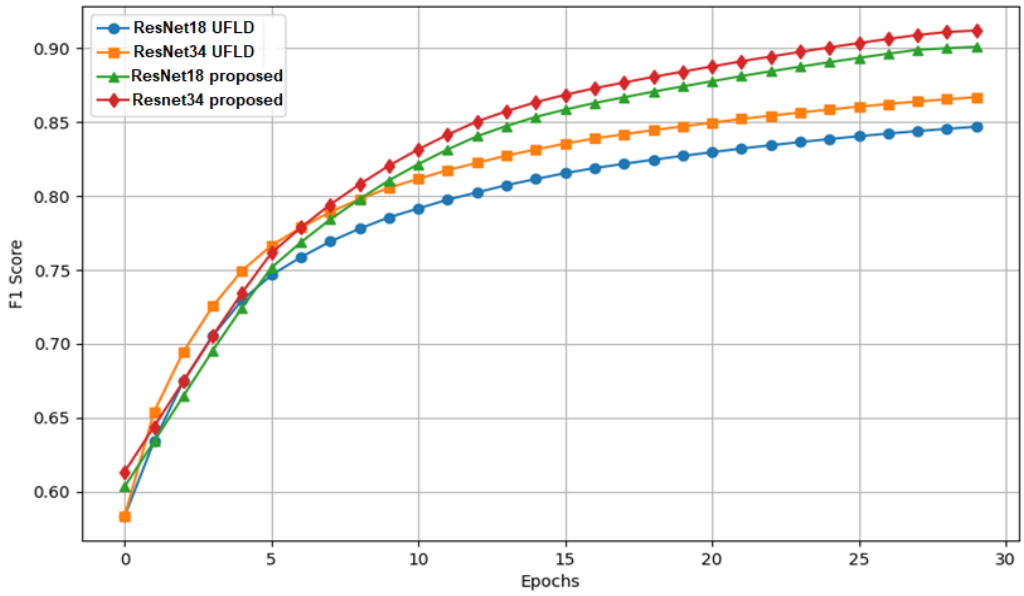

Fig. 5. The F1 score of proposed and UFLD methods for lane detection in dataset with various occlusion scenarios.

A comparison of the results in Figs. 2 and 3 indicates that adding the LSTM layer slightly improves the performance of the proposed model when compared to the UFLD model. Additionally, while the proposed method using ResNet18 shows strong initial performance, the proposed method using ResNet34 achieves higher accuracy after more extended training epochs due to its deeper architecture. The inclusion of LSTM in the proposed model also leads to a slight enhancement in the F1 score for lane detection.

To show the effectiveness of the proposed model, we also tested the proposed model with videos with challenging occluded lane scenarios from the TuSimple and CULane datasets and compared the results with the results of UFLD method. Fig. 5 shows the F1 score of proposed and UFLD methods for lane detection in dataset with various occlusion scenarios.

Fig. 6 shows the lane detection results for two sample video frames in low visibility scenarios using UFLD method. Fig. 7 also illustrates the results of applying the proposed method on video frame with less visibility and occluded scenarios.

The results of Figs. 5 to 7 clearly demonstrate that the proposed model significantly outperforms the UFLD model in dataset with occluded scenarios.

Fig. 6. Sample results of the UFLD model in occlusions and low visibility scenarios.

4.3. Comparative Study

We compared the results of the proposed algorithm with other methods using TuSimple test set. Table II shows the results of the proposed and various lane detection models on TuSimple test set. The results of Table II are summarized as follows:

- The proposed methods utilizing ResNet18 and ResNet34 demonstrate impressive execution times of 4.1ms and 4.3ms per frame, respectively, making them highly efficient compared to most methods such as SCNN, which takes 133.5ms.
- The accuracy of the proposed method using ResNet34 reaches **70.95%**, which is significantly higher than the accuracy of LaneNet (**38.96%**) and SCNN (**53.96%**). This indicates that the proposed model is more effective in correctly identifying lane markings in various conditions.
- The F1 Score is a crucial metric for evaluating the balance between precision and recall in lane detection tasks. The proposed method using ResNet34 achieves an F1 Score of **96.20%**, outperforming all other methods listed.

It is important to note that although the proposed method marginally outperforms the UFLD method on the Tusimple dataset, which has fewer occlusion scenarios, the proposed model significantly outperforms the UFLD model in datasets with occluded scenarios, as shown in Figs. 5 to 7.

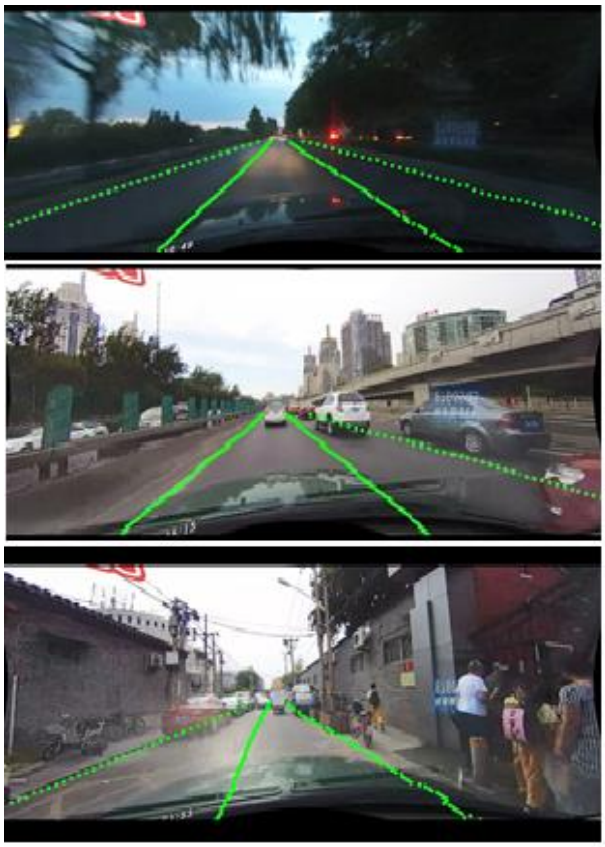

Fig. 7. Sample results of the proposed model in occlusions and low visibility scenarios.

5. Conclusion

In this paper, we proposed a novel spatiotemporal lane detection approach that combines ResNet-based spatial feature extraction with LSTM-based temporal modelling, integrated within UFLD's hybrid anchor-driven ordinal classification framework. This design enables the model to capture sequential dependencies and retain contextual information across frames, significantly improving detection performance in challenging real-world conditions such as occlusion, low visibility, and complex road environments. Our method preserves the efficiency of the original UFLD while enhancing its robustness. The lightweight ResNet-18 + LSTM configuration achieves an impressive processing rate of 244 FPS, confirming its

suitability for real-time deployment in autonomous driving and ADAS systems. This balance of speed and accuracy allows for responsive and reliable lane detection, even under adverse environmental conditions. The broader implications of our findings highlight the potential for integration into automotive-grade embedded systems, where low-latency and high-reliability perception modules are essential. Improved occlusion handling can contribute to safer navigation and reduced lane-keeping errors, particularly in urban and congested traffic scenarios. As the future work, we aim to extend this work through multi-camera setups and multi-sensor fusion, incorporating data from LiDAR or radar to enhance spatial understanding. Furthermore, exploring adaptive temporal modelling—where the temporal window adjusts dynamically based on scene complexity—could further boost performance in fast-changing environments.

Tabe II. The comparison of the results of proposed model with other methods on TuSimple test set

Method	Time/Frame (ms)	Accuracy	F1 Score
LaneNet [48]	19.0	96.38	94.8
SCNN[18]	133.5	96.53	95.97
SAD[22]	13.4	96.64	95.92
UFLD using ResNet18 [41]	3.2	95.50	96.05
UFLD using ResNet34 [41]	3.9	95.53	96.13
Method of [5] using ResNet18	2.4	95.82	95.10
Method of [5] using ResNet18	3.6	96.10	95.93
Proposed Method using ResNet18	4.1	95.60	96.08
Proposed Method using ResNet34	4.3	95.70	96.20

6. References

[1] R. Asgarian Dehkordi, H. Khosravi and A. Ahmadyfard, "Vehicle Dimensions and Speed Estimation using Camera Calibration Based on Recognition of a Number of Common Cars by VGG Network," Tabriz *Journal of Electrical Engineering*, vol. 50, no. 2, pp. 777-788, 2020.

[2] A. Eslami Mehdi Abadi, F. Mohanna, "Investigation of the noisy image edge detection based on the GWO algorithm," Tabriz *Journal of Electrical Engineering* 2024.

[3] N.S. Aminuddin, M.M. Ibrahim, N.M. Ali, S.A. Radzi, W.H.M. Saad, and A.M. Darsono, "A new approach to highway lane detection by using Hough transform technique", *Journal of Information and Communication Technology*, vol. 16, no. 2, pp.244-260, 2017.

[4] T. Ganokratanaa, M. Ketcham, and S. Sathienpong, "Real-time lane detection for driving system using image processing based on edge detection and Hough transform," In International Conference on Digital Information and Communication Technology and its Applications, 2013.

[5] S. Nie, G. Zhang, L. Yun, and S. Liu, "A Faster and Lightweight Lane Detection Method in Complex Scenarios. *Electronics*," vol. 13, no. 13, p.2486, 2024.

[6] Qin, Z., P. Zhang, and X. Li, "Ultra fast deep lane detection with hybrid anchor driven ordinal classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2555-2568, 2022.

[7] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection", *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.

[8] T.-Y. Sun, S.-J. Tsai and V. Chan, "HSI color model based lanemarking detection", In IEEE Intelligent Transportation Systems Conference (ITSC), 2006, pp. 1168–1172.

[9] B. Yu and A. K. Jain, "Lane boundary detection using a multiresolution hough transform", In IEEE International Conference on Image Processing, vol. 2, 1997, pp. 748–751.

[10] Y. Wang, D. Shen, and E. K. Teoh, "Lane detection using spline model", *Pattern Recognition Letter*, vol. 21, no. 8, pp. 677–689, 2000.

[11] M. Aly, "Real time detection of lane markers in urban streets," In IEEE Intelligent Vehicles Symposium, 2008, pp. 7–12.

[12] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using b-snake", *Image and Vision Computing*, vol. 22, no. 4, pp. 269–280, 2004.

[13] Z. Kim, "Robust lane detection and tracking in challenging scenarios", *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, 2008.

[14] P. Krahenb ¨ uhl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," In Advances in Neural Information Processing Systems, 2011, pp. 109–117.

[15] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection", In *IEEE Intelligent Vehicles Symposium*, 1995, pp. 54–59.

[16] J. P. Gonzalez and U. Ozguner, "Lane detection using histogram based segmentation and decision trees," In IEEE Intelligent Transportation Systems Conference (ITSC), 2000, pp. 346–351.

[17] H.M. Mandalia and M.D.D. Salvucci, "Using support vector machines for lane-change detection," In Proc. Hum. Factors Ergon. Soc. Annu. Meet., vol. 49, no. 22, 2005, pp. 1965–1969.

[18] S. Lee, J. Kim, J. Shin Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. Seok Hong, S.-H. Han, and I. So Kweon, "VPGNET: Vanishing point guided network for lane and road marking detection and recognition," In International Conference on Computer Vision, Oct. 2017, pp. 1947–1955.

[19] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tng, "Spatial as deep: Spatial CNN for traffic scene understanding," In AAAI, 2018, pp. 7276–7284.

[20] Z. M. Chng, J. M. H. Lew, and J. A. Lee, "Roneld: Robust neural network output enhancement for active lane detection", In 25th International Conference on Pattern Recognition (ICPR), IEEE, 2020, pp. 6842-6849.

[21] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, “Resa: Recurrent feature-shift aggregator for lane detection”, In Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 4, 2020, pp. 3547-3554.
[22] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, “Learning lightweight lane detection CNNs by self attention distillation,” In International Conference on Computer Vision (ICCV), 2019, pp. 1013–1021.
[23] Y. Hou, Z. Ma, C. Liu, T.-W. Hui, and C. C. Loy, “Inter-region affinity distillation for road marking segmentation”, In IEEE Conference on Computer Vision and Pattern Recognition, June 2020.
[24] 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 European Conference on Computer Vision, 2020, pp. 1–16.
[25] H. Abualsaud, S. Liu, D. Lu, K. Situ, A. Rangesh, and M. M. Trivedi, “Lane-AF: Robust multi-lane detection with affinity fields,” "Laneaf: Robust multi-lane detection with affinity fields", *IEEE Robotics and Automation Letters,* vol. 6, no. 4, pp. 7477-7484, 2021.
[26] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, “Focus on local: Detecting lane marker from bottom up via key point,” In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2021, pp. 14122-14130.
[27] J. Li, X. Mei, D. Prokhorov, and D. Tao, “Deep neural network for structural prediction and lane detection in traffic scene,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 690–703, 2016.
[28] J. Philion, “Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network,” In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 11 582–11 591.
[29] Y.-C. Hsu, Z. Xu, Z. Kira, and J. Huang, “Learning to cluster for proposal-free instance segmentation,” In International Joint Conference on Neural Networks, 2018, pp. 1–8.
[30] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, “Polylanenet: Lane estimation via deep polynomial regression,” In IEEE *International Conference on Pattern Recognition*, 2021, pp.6150–6156.
[31] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool, “End-to-end lane detection through differentiable least-squares fitting,” In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019.
[32] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, “End-to-end lane shape prediction with transformers,” In Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2021, pp. 3694–3702.
[33] L. Tabelini, R. Berriel, T.M. Paixao, C. Badue, A.F. De Souza, and T. Oliveira-Santos, “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, 2021, pp. 294-302.
[34] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, “Keep your eyes on the lane: Real-time attention-guided lane detection,” In IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1–9.
[35] J. Su, C. Chen, K. Zhang, J. Luo, X. Wei, and X. Wei, “Structure guided lane detection,” *arXiv preprint arXiv:2105.05403*, 2021.
[36] N. Garnett, R. Cohen, T. Pe’er, R. Lahav, and D. Levi, “3D-Lanenet: end-to-end 3D multiple lane detection,” In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2921–2930.
[37] Y. Guo, G. Chen, P. Zhao, W. Zhang, J. Miao, J. Wang, and T. E. Choe, “Gen-lanenet: A generalized and scalable approach for 3D lane detection,” In 16th European Conference on Computer Vision (ECCV), pp. 666-681.
[38] N. Efrat, M. Bluvstein, N. Garnett, D. Levi, S. Oron, and B. E. Shlomo, “Semi-local 3d lane detection and uncertainty estimation,” *arXiv preprint arXiv:2003.05257*, 2020.
[39] N. E. Sela, M. Bluvstein, S. Oron, D. Levi, N. Garnett, and B. E. Shlomo, “3D-lanenet+: Anchor free lane detection using a semilocal representation,” *arXiv preprint arXiv:2011.01535*, 2020.
[40] R. Wang, J. Qin, K. Li, Y. Li, D. Cao, and J. Xu, "Bev-lanedet: An efficient 3D lane detection based on virtual camera via key-points," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 1002-1011 .
[41] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," In *16th European Conference Computer Vision (ECCV) 2020,, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, 2020: Springer, pp. 276-291.
[42] H. Honda and Y. Uchida, "CLRerNet: improving confidence of lane detection with LaneIoU," In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 1176-1185.
[43] S. Sultana, B. Ahmed ,M. Paul, M. R. Islam, and S. Ahmad, "Vision-based robust lane detection and tracking in challenging conditions," *IEEE Access, vol. 11, pp. 67938-67955,* 2023.
[44] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” *arXiv preprint arXiv:1402.1128*, 2014.
[45] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 249–256, 2010.
[46] TuSimple, “Tusimple benchmark,” https://github.com/TuSimple/tusimple-benchmark, accessed November, 2019.
[47] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Spatial As Deep: Spatial CNN for Traffic Scene Understanding, " AAAI Conference on Artificial Intelligence (AAAI), February, 2018.
[48] M. Almehdhar *et al.*, "Deep Learning in the Fast Lane: A Survey on Advanced Intrusion Detection Systems for Intelligent Vehicle Networks," *IEEE Open Journal of Vehicular Technology,* 2024.