IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 February 2024, accepted 29 February 2024, date of publication 7 March 2024, date of current version 21 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3373203

RESEARCH ARTICLE

Enhanced SCNN-Based Hybrid Spatial-Temporal Lane Detection Model for Intelligent Transportation Systems

JINGANG LI[1], CHENXU MA[1], YONGHUA HAN[1], HAIBO MU[2], AND LURONG JIANG[1]
[1]School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China
[2]Hangzhou Hikvision Digital Technology Company Ltd., Hangzhou 310018, China

Corresponding author: Yonghua Han (han_yong_huahan@zstu.edu.cn)

This work was supported in part by Zhejiang Sci-Tech University 2021 National University Students Innovation and Entrepreneurship Training Program, China, under Grant 11120032662125; and in part by the Key Research and Development Program of Zhejiang Province under Grant 2022C03136.

ABSTRACT Accurate and timely lane detection is imperative for the seamless operation of autonomous driving systems. In this study, leveraging the gradual variation of lane features within a defined range of width and length, we introduce an enhanced Spatial-Temporal Recurrent Neural Network (SCNN) framework. This framework serves as the cornerstone of an innovative hybrid spatial-temporal model for lane detection, which is tailored to address the prevalent issues of substandard detection performance and insufficient real-time processing in intricate scenarios, such as those involving lane erosion and inconsistent lighting conditions, which often challenge conventional models. With the foundational understanding that lanes manifest as continuous lines, we employ a temporal sequence of lane imagery as the input to our model, thereby ensuring a rich provision of feature information. The model adopts an encoder-decoder structure and integrates a Spatial-Temporal Recurrent Neural Network module for the extraction of interrelated information from the image sequence. The model culminates in the output of the lane detection results for the terminal frame. The proposed lane detection model exhibits a commendable synthesis of accuracy and real-time efficiency, attaining an *Accuracy* of 97.87%, an F_1-score of 0.943, and a *FPS* of 19.342 on the tvtLANE dataset and an *Accuracy* of 98.21%, an F_1-score of 0.957 on the Tusimple dataset. These metrics signify a superior performance over a majority of the current lane detection methods.

INDEX TERMS Computer vision, deep learning, lane detection.

I. INTRODUCTION

Recent advancements in autonomous driving technologies have attracted global interest due to their extensive application potential and their capacity to conserve both human and material resources. A critical component of this technology is the analysis and processing of lane markings, as captured by vehicular cameras and interpreted through algorithms. This process is integral to the functioning of autonomous driving systems, as it enables vehicles to ascertain their current position and intended trajectory, thereby augmenting both traffic safety and efficiency. Furthermore, this technology is essential for intelligent traffic management and driver assistance systems, providing indispensable, real-time support for urban traffic planning and driving safety. Within the realm of lane detection, deep learning algorithms have emerged as a novel approach. When compared to traditional machine learning techniques, deep learning demonstrates enhanced feature extraction capabilities and improved resistance to noise, which allows it to adapt to diverse and complex driving conditions with superior performance. Consequently, it has become a popular subject of research.

As scientific knowledge progresses, deep learning-based lane detection methods for automated or assisted driving

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

are increasingly becoming a focal point of research. In the scientific literature, studies [1], [2], [3] have designed deep learning models to facilitate pixel-level image segmentation. This technique discerns whether each pixel in an image corresponds to a lane or the background, thus providing comprehensive lane detection information suitable for intricate scenarios, albeit at the cost of increased computational resources. Studies [4], [5], [6] have adopted a row-based detection approach, whereby the image is segmented into horizontal sections, and deep learning models identify the presence of lane markings by pinpointing candidate points within each section. Although this method is straightforward and computationally efficient, its performance is suboptimal for curved and diagonal lane markings, and it is less capable in the detection of complex scenes. Studies [7], [8], [9] have utilized an anchor-based strategy, wherein deep learning models leverage anchors to deduce regression-based relative coordinates, enhancing the detection accuracy for lane curvature. This approach is versatile and can accommodate different shapes and orientations of lane markings, but it necessitates the adjustment of anchor box sizes based on the dimensions and angles of the lane markings, which complicates the model.

The aforementioned text delineates various deep learning-based lane detection methodologies. Nonetheless, lane detection in practical applications encounters numerous challenges. Factors such as inconsistent lighting, shadows, glare, worn lane markings, road obstructions, and low image contrast elevate the requirements for real-time and precise detection of lane markings. Specifically, real-world road conditions are highly variable, and many publicly available datasets are limited to particular scenarios. For instance, the Tusimple dataset [10] primarily features images from highways under clear weather conditions, which are relatively easy to detect. In contrast, actual driving often involves a multitude of complex scenarios, including rural, urban, and highway environments. To more accurately reflect real-world conditions, some researchers have introduced datasets that encompass various complex scenarios, such as the tvtLANE dataset proposed by Zou et al. [11], which includes rural, urban, and highway sections, thereby addressing the issue of limited sample diversity. Additionally, to enhance model robustness, some researchers have amalgamated different datasets. For example, Khan et al. [12] trained deep learning models using a combination of the Udacity Machine Learning Nanodegree Project Dataset [13] and the Cracks and Potholes in Road Images Dataset [14], aiming to effectively detect lane markings under adverse weather and lighting conditions.

To surmount the challenges inherent in lane detection, researchers have conducted comprehensive studies and proposed a suite of innovative solutions, although these methods have inherent limitations. Perumal et al. [15] tackled the issue of unstructured roads with their LaneScanNET, which showed promise in conditions such as shadows, fog, and dust; however, it was not specifically tailored for scenarios with uneven lighting. Bhandari et al. [16] introduced the deep learning model Deeplane, which facilitated lane detection in the absence of lane markings, yet its detection accuracy and efficacy in complex scenarios like shadows require further enhancement. Oğuz et al. [17] employed a one-dimensional deep learning technique to classify pixel intensity distributions for lane marking detection, overcoming low recognition rates in the presence of shadows and road imperfections, but it was not sufficiently adept at detecting curved paths. Zhao et al. [18] combined fast connections, gradient maps, and WGAN features to devise Ripple-GAN, which exhibited strong performance in multi-lane and complex road conditions, although its detection capabilities were compromised on roads that were completely or partially obstructed, such as those in dimly lit areas. In 2023, Dewangan and Sahu [19] addressed adverse conditions such as nighttime, rain, and fog by extracting texture features based on the Local Vector Pattern (LVP) and employing an optimized Deep Convolutional Neural Network (DCNN) for road and lane classification, introducing an FS-MS technique to refine the model outcomes. Nonetheless, this methodology did not account for scenarios involving damaged lane markings. Zhang and Zhong [20] integrated a spatial attention mechanism and proposed the deep learning model CRLaneNet, based on Catmull-Rom curves, which proved effective in crowded and low-light environments but was less successful in handling curved and intersecting lanes. Despite technological progress in specific environments and conditions, lane detection algorithms continue to require adaptation to a broader range of extreme and diverse scenarios to satisfy the practical demands of automated or assisted driving systems.

Contemporary methodologies in the field of lane detection have predominantly focused on the analysis of individual image frames, thereby not fully capitalizing on the prior contextual information available across sequential frames. This limited approach has been shown to reduce the accuracy of lane detection within complex environments, particularly when faced with challenges such as variable lighting conditions, missing lane markings, and visual obstructions. Lane markings, which are typically continuous solid or dashed lines, possess a clear temporal correlation in terms of their location. To address this, hybrid spatial-temporal lane detection methods have been introduced as an efficacious solution. These methods utilize the combined feature information from preceding frames together with the current frame within a temporal sequence to deduce the positional attributes of lane markings in the present frame. In scenarios marked by damaged lane markings, occlusion by vehicles, and non-uniform lighting, such models draw upon the extensive lane feature information from previous frames to bolster the detection process in the current frame, thereby yielding improved detection results.

In fields such as video content analysis and sequence prediction, Spatial-Temporal Recurrent Neural Networks (ST-RNNs) have been established as a powerful tool capable of discerning complex dependencies inherent in time-series data (as detailed in [21], [22], and [23]). Leveraging the successful deployment of ST-RNNs in these areas, this study investigates their potential application in lane detection, with a specific focus on dynamic traffic scenarios and the analysis of continuous video frames. In this vein, Zou et al. [11] introduced an innovative network architecture that amalgamates Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs), resulting in a notable enhancement in detection performance. However, the method lacks exploration of higher precision in complex environments and does not fully capitalize on prior lane line information. Building upon this foundation, Dong et al. [24] utilized the SCNN [25] to delve deeper into the geometric features of lane markings and achieved an uplift in model performance by substituting Conv-LSTM with ConvGRU within the RNN framework. However, the method lacks a balance between real-time performance and accuracy. Liu and Gao [26] proposed a lane detection algorithm predicated on the fusion of multi-frame information that integrates CNNs with Conv-LSTM and DenseNet, creating a network adept at deep semantic extraction that shows promising results under challenging conditions such as shadows, wireless environments, and nighttime settings, yet it does not fully exploit the predictive utility of lane marking priors. Moreover, Gupta and Choudhary [27] proposed an unsupervised spatial-temporal incremental clustering algorithm for dynamic lane detection. This method effectively integrates time-series information into the curve fitting process for lane lines, specifically addressing the challenges posed by highly dynamic traffic scenarios. Nevertheless, the approach falls short in providing an in-depth discussion of complex environmental factors, such as vehicle occlusion. Although these investigations have highlighted the substantial potential for augmenting lane detection through the synthesis of spatial and temporal data, they concurrently point to the limited robustness of algorithms when contending with complex scenes that include damaged lane markings, uneven lighting, and vehicle occlusions, signaling a need for more resilient solutions.

In light of the research outlined above, this manuscript introduces a hybrid spatial-temporal model for lane detection predicated on an enhanced SCNN architecture. This model is specifically designed to tackle formidable challenges such as damaged lane markings, vehicle occlusions, and inconsistent lighting, achieving greater accuracy and real-time performance, thereby advancing the state of the art.

The primary contributions of this study are twofold:

(1) Acknowledging the significance of shape, color, texture, and the temporal interplay of features across sequential frames, we propose a novel hybrid spatial-temporal sequence-to-sequence deep neural network, incorporating an encoder-decoder framework. This approach is aimed at addressing the difficulties inherent in precisely and expeditiously detecting lane markings from single images under challenging conditions such as lane marking degradation, vehicular occlusion, and disparate lighting.

(2) We propose an enhanced SCNN configuration that maintains accuracy while expediting the real-time feature extraction for elongated, linear objects. A novel block-wise flow strategy is devised for SCNN in each direction, supplanting the traditional layer-by-layer approach. This strategy not only abbreviates the SCNN processing time but also enables efficient, lightweight information transfer across various directions for spatially consistent lane markings. An replacement strategy is adopted to prevent information loss that can occur when superimposed values exceed unity during the binarization process in SCNN's flow of information, thereby promoting the propagation of features. Lastly, we employ depthwise separable convolutional kernels in lieu of conventional convolutional kernels, significantly diminishing the parameter count.

Furthermore, this technology is essential for intelligent traffic management and driver assistance systems, providing indispensable, real-time support for urban traffic planning and driving safety. The specific challenges of lane erosion and inconsistent lighting conditions present significant hurdles for current lane detection models, necessitating advancements that can adeptly handle such complexities. Our proposed model aims to fill this gap by leveraging spatial-temporal information to enhance detection accuracy in challenging scenarios, thereby supporting the broader application of autonomous driving technologies in intelligent transportation systems. This not only contributes to the academic discourse on lane detection but also offers practical insights into developing more resilient and adaptable autonomous driving solutions that can navigate the intricacies of real-world driving conditions.

The manuscript is structured as follows: Section II meticulously delineates the methodology adopted for lane detection. The discourse commences with an exposition of image preprocessing techniques, subsequently advancing to the conceptualization of a bespoke deep learning model crafted to address the multifaceted challenges inherent in lane detection. This model comprises a tripartite architecture: the foundational backbone network, an enhanced SCNN, and the ST-RNN. Section III delves into the experimental framework and its intricacies. It initiates with a delineation of the datasets employed, namely TuSimple and tvtLANE, progressing to a detailed account of the experimental milieu, the hyperparameters selected, and the benchmarks for evaluation. What follows is a suite of four targeted experiments, encompassing an exploration of backbone network selection, the information flow strategies adopted by enhanced SCNN, the refinement of the SCNN's sliding step size, and an analysis of the feature map channel dimensions pre- and post-enhanced SCNN integration. The efficacy of the proposed model in lane detection is rigorously assessed through the employment of confusion matrices, P-R curves, analyses of real-time performance, and a various means of visualisation.Concluding the paper, Section IV amalgamates and distills the principal scholarly contributions, concurrently acknowledging the research's constraints and positing avenues for future inquiry.

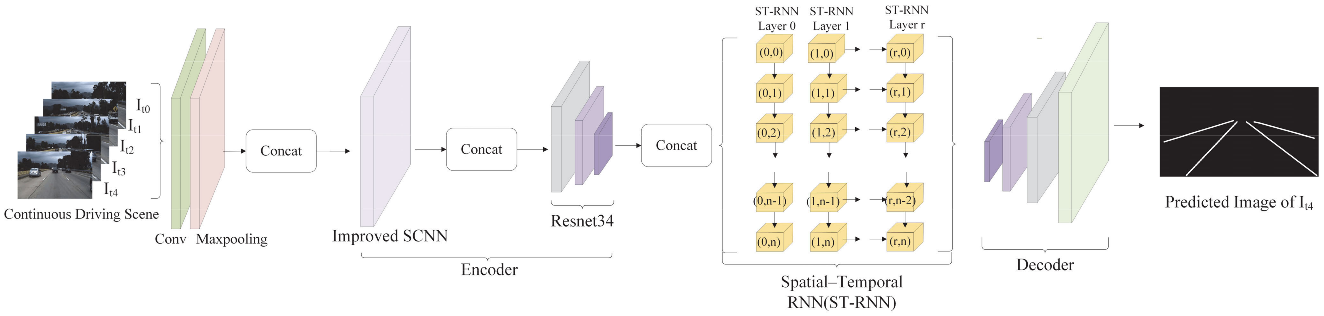


FIGURE 1. Overall model.

II. METHODOLOGY

A. GENERAL NETWORK ARCHITECTURE

This investigation conceptualizes lane detection as a binary semantic segmentation challenge. The semantic segmentation model proposed herein, delineated in Fig. 1, encapsulates four integral components: image preprocessing, the encoder, the Spatial-Temporal Recurrent Neural Network (ST-RNN), and the decoder. The preprocessing of images commences with the extraction of salient features from the input image, informed by the distinctive attributes of lane markings, and concurrently reduces the image dimensions to bolster real-time processing capabilities while preserving essential lane marking details. This step is further optimized by employing advanced dimensionality reduction techniques, ensuring a balance between computational efficiency and the preservation of crucial information. The encoder's role is to transmute the input image into refined semantic information. In this phase, our approach is particularly attentive to the geometric configuration of lane markings by harnessing prior contextual knowledge, which is then complemented by the integration of ResNet34 to augment the extraction of lane marking texture and chromatic data. The use of ResNet34 is instrumental in enhancing the model's ability to extract intricate features, thereby significantly improving the accuracy of semantic interpretation. The ST-RNN is engineered to distill the temporal associative information from a chronologically arranged series of lane marking images. This component adeptly captures the dynamic changes in lane markings over time, providing a temporal depth to the analysis. The decoder's primary function is to transcribe the high-level semantic data pertaining to lane markings, as ascertained by the encoder, into pixel-level semantic information. It meticulously reconstructs the lane information, ensuring a high degree of accuracy in the final detection results. The culmination of the model's output is the lane detection result for the terminal time frame in the sequenced progression of lane markings. This comprehensive approach underscores the model's capability to not only detect but also precisely delineate lane markings in real-time scenarios.

Algorithm 1 Algorithm of the Proposed Method

```
Input: ConsecutiveImages [5] (each image shape: 128 × 256×3)
Output: LogProbabilities
DataTensor ← ConvertImagesToTensor(ConsecutiveImages)

// Initial feature extraction using a convolutional layer
FeatureList ← []
for each Image in DataTensor do
    Feature ← ApplyConvolutionalLayer(Image)
    Append Feature to FeatureList

// Preprocess features for SCNN
SCNN_Input ← StackFeatures(FeatureList)
SCNN_Input ← ApplyMaxPooling(SCNN_Input)
SCNN_Output ← ApplySCNN(SCNN_Input)
Conv_Input ← ApplyUnPooling(SCNN_Output)

// Prepare features for convLSTM
ConvLSTM_Features ← []
for i from 0 to 4 do
    ChannelFeature ← SliceChannels(Conv_Input, i)
    for each Layer in ResNet34Layers do
        ChannelFeature ← ApplyLayerAndMax
        Pooling(ChannelFeature)
    Append ChannelFeature to ConvLSTM_Features

// Apply convLSTM and upsample
ConvLSTM_Input ← ConcatenateFeatures(ConvLSTM_Features)
ConvLSTM_Output ← ApplyConvLSTM(ConvLSTM_Input)
UpsampledOutput←SequentiallyApplyConvAndUnPooling(ConvLSTM
_Output)

// Calculate final output probabilities
LogProbabilities ← CalculateLogSoftmax(UpsampledOutput)
// Calculate final output probabilities
```

B. IMAGE PREPROCESSING

Given the operating conditions of vehicle velocities spanning from 0 to 120 km/h and a camera frame rate of at least 25 FPS, the locations of lane markings across the quintet of images fed into the model over a consistent temporal interval remain largely invariant. This suggests that while the feature information extracted from each frame is akin, it is not identical, thereby allowing for reciprocal enhancement of information across frames. In accordance with the methodologies delineated in [11] and [24], a sequence of five contiguous images is selected for model input.

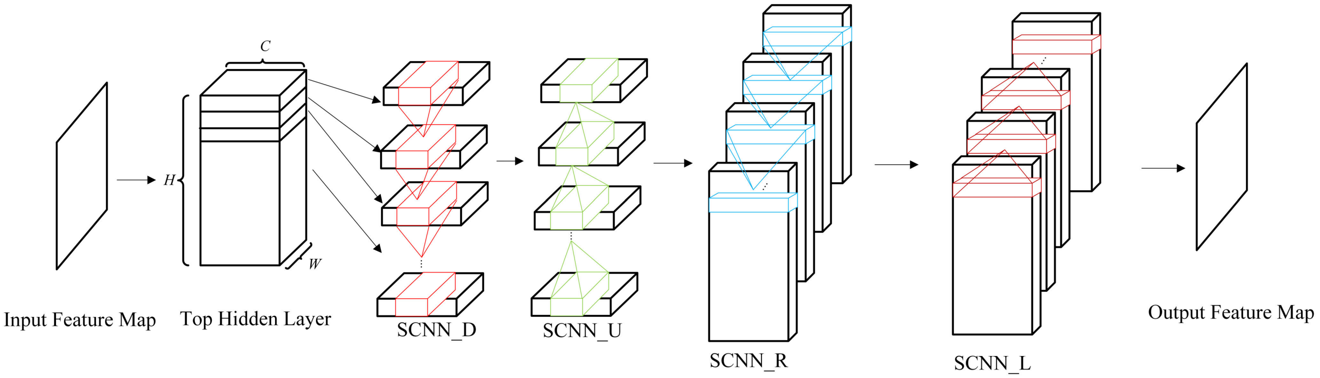


FIGURE 2. Enhanced SCNN. Through the improvement of this module, the prediction and training speed of ours neural network model has been significantly enhanced.

Accounting for the established knowledge that lane markings—irrespective of being white or yellow—exhibit pixel values in their RGB channels that are either maximal or proximate to maximal, we apply convolution and max pooling operations to these five images. This approach not only facilitates preliminary feature extraction but also contracts the image scale, consequently economizing on the computational demands for the ensuing enhanced SCNN.

C. ENHANCED SCNN

The SCNN initiates by horizontally dissecting the input image into layers, executing convolutions sequentially from top to bottom and vice versa, succeeded by vertical segmentation and convolutions from left to right and the reverse. The SCNN transcends the conventional layer-by-layer convolution, adopting a slice-by-slice methodology. However, the substantial dimensions of acquired lane marking images and the profusion of slicing can drastically decelerate the flow of information, thus impinging upon the model's real-time efficacy.

Algorithm 2 Algorithm of the Enhanced SCNN(in UP_to DOWN Direction

```
Input: SCNN_Input (shape: [64 × 5, 64, 128])
Output: SCNN_Output
for i from 0 to (SCNN_Input.height - 8) step 4 do
    FeatureMap_f1 ← ExtractRows(input_scnn, i, i + 8)
    ConvFeatureMap_f2 ←ApplyConvolution(FeatureMap_f1, stride=2)
    SCNN_Input ← ReplaceRows(SCNN_Input, i + 4,
ConvFeatureMap_f2)
end for
SCNN_Output ←SCNN_Input
```

In consideration of the fact that lane markings possess a defined area and exhibit color uniformity along both horizontal and vertical planes, we introduce an optimized SCNN. This advancement retains the precision of lane detection while ameliorating real-time performance. The optimized SCNN transitions from a layer-by-layer information flow to a block-by-block paradigm, i.e., progressing in designated sliding strides. We define the sliding stride for each direction as $s = (s_1, s_2)$, where s_1 and s_2 represent the sliding strides for vertical and horizontal directions, respectively. The processed image is then channeled into the Top Hidden Layer as depicted in Fig. 2, ensuing in an information flow that follows the up-to-down, down-to-up, left-to-right, and right-to-left trajectories, correlating with SCNN_D, SCNN_U, SCNN_R, and SCNN_L in Fig. 2, respectively. It is inferred that the SCNN operates with $s = 1$, whereas the optimized SCNN functions with $s > 1$. The pseudocode for the enhanced SCNN is provided above.

(1) delineates the formula for the up-to-down direction in the optimized SCNN, with analogous formulations for other directions omitted for brevity.

$$x'_{i+4:i+7} = ReLU(SCNN\_D(x_{i:i+7}, K)) \quad i = 1, 5, 9, \ldots, H-7 \tag{1}$$

In (1), the variable x sans superscript denotes the feature map prior to the onset of information flow, while x' signifies the feature map that has been updated in the course of information flow. The symbol ':' designates the range of values, with the adjacent numerals specifying the respective lower and upper boundaries. The indices i represent the slide of horizontal slice $s_1 = 4$, respectively. ReLU stands as the activation function, and $SCNN\_D$ indicates the convolutional operation involving the kernel K and the input $x_{i:i+7}$, with a convolutional stride set at (2,1).

The traditional SCNN employs a concatenation strategy during information flow, convolving upper layer data with a convolutional kernel and amalgamating the outcome with the subsequent layer. Our refined SCNN adopts a replacement strategy, concurrently selecting both upper and lower layers for a convolution with a stride of (2,1), directly substituting the result into the lower layer. This not only propels real-time performance but also, by expanding the convolutional scope of feature information, aids in capturing a more extensive array of contextual data, thus bolstering feature representation. Moreover, the elimination of repetitive concatenation operations mitigates the potential for information loss that can arise when the cumulative values exceed unity during the binarization process, fostering a more effective feature flow. Additionally, in light of the gradual variation of lane marking

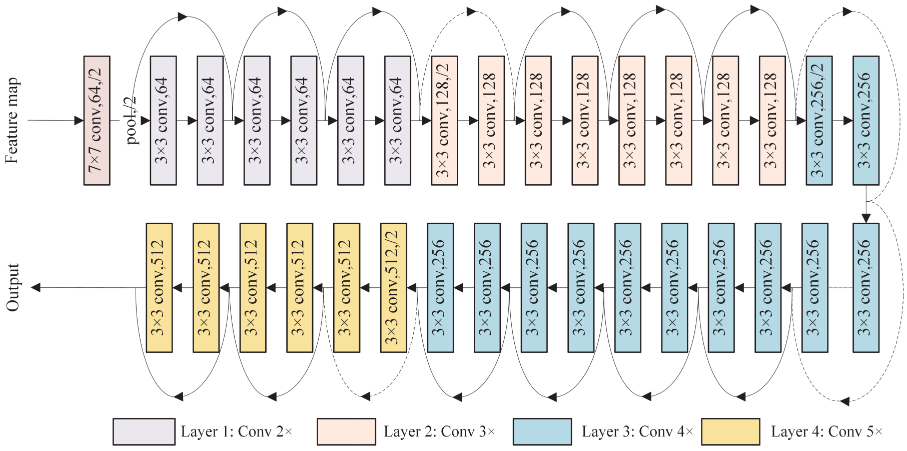
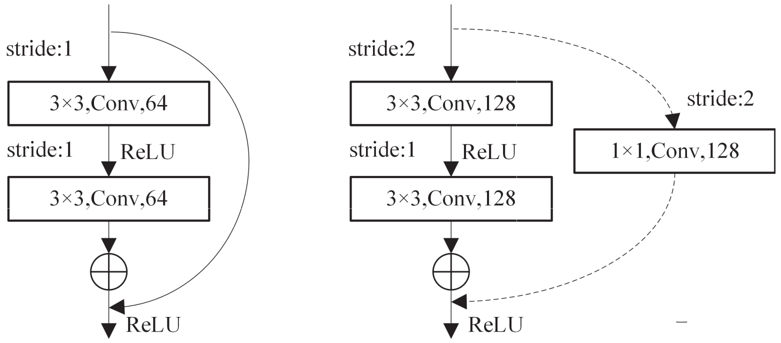


FIGURE 3. ResNet34.



(a)

(b)

FIGURE 4. (a) Solid residual structure (b) Dashed residual structure.

features within a specified width and length range and the inherent fault tolerance in spatial feature extraction, we substitute the standard convolution with depthwise separable convolution, which, while safeguarding accuracy, enhances the real-time processing capability.

D. BACKBONE NETWORK

The backbone network is predicated on the ResNet34 architecture. ResNet34 is a streamlined residual network adept at extracting spatial color and texture features, comprising a mere 34 layers. To facilitate connectivity with the impending ST-RNN layer dedicated to temporal feature extraction, the fully connected layer is omitted, preserving only the initial 33 convolutional layers. ResNet34 is organized into 4 Layers, each consisting of a series of identical BasicBlocks. Fig. 3 exhibits the structural schematic of ResNet34, wherein BasicBlocks of uniform color coalesce into a Layer, linked by residual edges between successive BasicBlocks.

In Fig. 3, divergence in the channel count of images entering and exiting a residual edge is denoted by a dotted line, while congruence is indicated by a solid line. Fig. 4 presents the structural schematic for each variant of residual edge upon its inaugural appearance in Fig. 3.

In Fig. 4(a), the channels of images interfacing with the residual edge are equivalent; the principal segment comprises a duo of 3 × 3 convolutions in sequence, subsequently superimposed with the residual edge to procure the output. Conversely, in Fig. 4(b), where the channel counts differ, the principal segment consists of a pair of 3 × 3 convolutions in series, with the residual edge undergoing an ancillary 1 × 1 convolution prior to amalgamation with the principal segment to align the channel count of the resultant image.

ResNet34 serves to distill lane marking information pertaining to texture and color. The combined utilization of SCNN and ResNet34 constitutes the encoding segment of the model, ensuring the exhaustive extraction of both global features, such as the geometric contour and hue of lane markings, and local features, including textural details. The ST-RNN captures correlation information for each image in a lane sequence over continuous time. The decoding layer is primarily responsible for converting high-level semantic information of lanes extracted in the encodinglayer into pixel-level semantic information. The output of the final model represents the detection result for the last frame of the lane sequence over continuous time.

E. SPATIAL-TEMPORAL RNN (ST-RNN)

A paramount challenge in lane detection lies in the accurate recognition and tracking of dynamic shifts in vehicular trajectories across a series of video frames. Lane markings are distinguished by their inherent spatial-temporal continuity within natural settings: spatially, they are generally characterized by smooth and unbroken lines; temporally, their positions and configurations undergo gradual transformations in response to vehicular motion, often exhibiting a degree of regularity. To harness this spatial-temporal data comprehensively, this investigation adopts a Convolutional Long Short-Term Memory (Conv-LSTM) architecture as the cornerstone of the Spatial-Temporal Recurrent Neural Network (ST-RNN) module, tasked with discerning the dynamic attributes of lane markings through consecutive video frames.

The Conv-LSTM framework integrates the robust spatial feature extraction prowess of Convolutional Neural Networks (CNNs) with the sequential temporal analysis capabilities inherent to Long Short-Term Memory (LSTM) networks. This synergistic combination is particularly apt for addressing the spatial-temporal complexities in lane detection. Within the Conv-LSTM paradigm, convolutional operations supplant the fully connected mechanisms typical of conventional LSTM, thereby enabling the model to adeptly parse spatial features of lane markings per frame, all while preserving the integrity of temporal data. Furthermore, the gated controls within Conv-LSTM judiciously orchestrate the timing for the integration of novel input features and the expurgation of obsolete data, thus safeguarding the uniformity and coherence of lane marking features throughout the video sequence.

The architecture's suitability for lane detection is underscored by its capacity to not only interpret intricate spatial details within individual frames but also to monitor and appraise temporal shifts in lane marking characteristics across a succession of frames. Fig. 5 elucidates the operational principles of Conv-LSTM, where the convolutional gates encode both spatial and temporal variations of lane markings. This mechanism empowers the ST-RNN module to effectively distill the sequential consistency and uniformity of lane markings, thereby enhancing the precision of lane detection predictions.

As illustrated in Fig. 5, the flow of information is modulated by the forget gate's sigmoid unit, which adjudicates the preservation or elimination of data from X_t and H_{t-1}, with residual data proceeding to the input gate. The sigmoid layer ascertains which data segments necessitate updates, and the tanh layer formulates novel cell state data for subsequent updates. The model's final output is derived by amalgamating the output gate's sigmoid-filtered information with the tanh-processed memory cell data.

Conv-LSTM's tripartite gate mechanism—encompassing the forget gate (f_t), input gate (i_t), and output gate (O_t)—orchestrates the flow from memory cells (C_t). These cells not only encapsulate the characteristics of the current input but also exert control over the transmission of antecedent information via the forget and input gates. This design endows Conv-LSTM with the facility to more efficaciously marshal and harness antecedent data when processing sequential datasets, thus capturing the sequence's long-term dependencies with greater acuity. The relational dynamics of information transfer are encapsulated by (2)-(4):

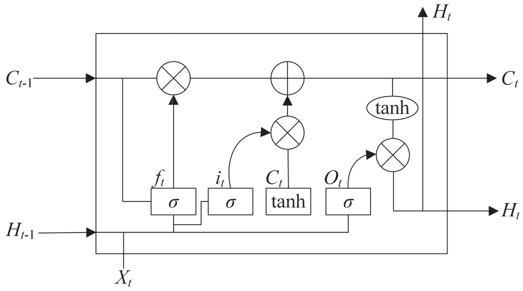


FIGURE 5. Conv-LSTM.

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \tag{3}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \tag{4}$$

Herein, σ signifies the sigmoid activation function, $\circ$ represents the element-wise product of matrices, and $*$ indicates convolutional operations. This empirical framework utilizes convolutional operations for feature extraction, while gate structures, comprised of sigmoid layers and point-wise multiplication maneuvers, selectively filter the information.

III. EXPERIMENTS

A. TVTLANE DATASET

The tvtLANE dataset [11], a derivative of the Tusimple dataset, was utilized for the experiments. Detailed information on the training and testing sets is delineated in Table 1. The tvtLANE dataset is composed of two components: highway sections and rural road segments. The highway sections are derived from the Tusimple dataset, while the rural road segments are gathered from rural areas in China. Each image within the dataset is standardized to a resolution of 128 × 256 pixels. The training set comprises 9548 labeled images, which underwent a fourfold data augmentation process. The testing set consists of 1268 labeled images. To augment the dataset, the 13th image from each Tusimple sequence was labeled, and an additional 1148 rural road images were incorporated to broaden the dataset's coverage. The dataset was organized into 19383 sequences, with each sequence containing images from five consecutive time points.

In acknowledgment of the proportionality between driving speed and the distance covered within identical time intervals, the training set was sampled at intervals of 1, 2, and 3, while a consistent interval of 1 was maintained for the testing set

TABLE 1. The training set and test set of tvtLANE.

Part	Including	Labeled Frames	Labeled Images	Sequence Number	Purpose
Trainset	Highway	13th, 20th	29,008	19096	Training models
	Rural area	13th, 20th	9,184		
Testset	Test #1	13th, 20th	540	287	Testing performance

TABLE 2. Sampling methods for the tvtLANE training set.

Sample Stride	Train Sample Frames	Labeled Ground Truth
1	1th,4th,7th,10th,13th	13th
2	5th,7th,9th,11th,13th	
3	9th,10th,11th,12th,13th	
1	8th,11th,14th,17th,20th	20th
2	12th,14th,16th,18th,20th	
3	16th,17th,18th,19th,20th	

to preserve lane continuity. The sampling strategy for the tvtLANE training set is explicated in Table 2. To detect continuous scenes, the model acquires knowledge of a sequence of lane marking image features during a single training iteration, wherein each sequence comprises

5 images. Utilizing information from the first 4 images, the model aids in learning the lane marking positions depicted in the final image, which is the sole image containing labels. Each row in the figure represents a distinct sampling interval, capturing a series of lane marking images and their corresponding labels at consecutive time intervals.

The tvtLANE dataset encompasses a spectrum of diverse scenarios as illustrated in the Fig. 6: with the former four rows representing the antecedent four frames, the fifth row corresponding to the incumbent frame, and the sixth row depicting the Ground Truth pertinent to the current frame. Scene 1 epitomizes the prototypical and most facilely identifiable scenario—a straight roadway under clement skies. Scene 2 replicates the serpentine thoroughfares characteristic of alpine passes or intricate junction turns, whereas Scene 3 illustrates a curvilinear segment of roadway enshrouded in shadows, notwithstanding the presence of vehicular obstructions. Scenes 4 through 7 are representative of a variety of conditions encompassing low luminance with unilateral shadowing, low luminance beset by vehicular occlusions and bilateral shadows, hyperexposure, and the complex interplay of light during the waning hours of the day, respectively. Scene 8 reflects the widespread issue of lane degradation, particularly prevalent on rural and suburban roads. Scene 9 exhibits instances of vehicular occlusion under low light conditions. Scene 10 presents a straight segment of roadway under the cloak of shadows, while Scene 11 delineates a curvaceous section of the road under low light conditions.

TABLE 3. Hyperparameters of the experiment.

Hyperparameters	Value
Epoch	10
Init_lr	1e-3
Optimizer	Adam
Learning Rate Schedule	Step Decay
The weight of *WBCE*	0.02:1.02

B. EXPERIMENTAL ENVIRONMENT

To ensure a fair comparison across all experiments, we maintained a consistent experimental environment. Every test was conducted on an Ubuntu 20.04 operating system with identical hardware specifications, including an Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60GHz, an RTX 3090 GPU, 90GB of RAM, and 50GB of storage. The use of a uniform platform mitigates any performance variations due to system differences, thereby isolating the impact of the experimental variables under study.

C. EXPERIMENTAL HYPERPARAMETERS

The hyperparameters, as enumerated in Table 3, were standardized across all experiments to ensure comparability. The selection of these hyperparameters was grounded in established practices within the field and previous empirical findings that suggest their suitability for the models and datasets employed in our research. A fixed epoch count of 10, an initial learning rate of 1e-3 with a step decay schedule, and the Adam optimizer were used consistently unless the investigation required specific alterations to these parameters. Such changes, if any, were part of a controlled approach to evaluate their influence on the model's performance. To enhance the model's generalization capabilities, data augmentation techniques such as random cropping, rotation, and flipping were incorporated into the training process. Additionally, the training process involved a detailed learning rate adjustment strategy, including the setting of an initial learning rate and applying a step decay schedule to modulate the learning rate effectively.

We took additional measures to promote the reproducibility of our experiments, including the use of a specific deep learning framework, PyTorch 1.10.0, and CUDA version 11.3, to eliminate variability due to software differences. Detailed documentation of the experimental procedures, along with the explicit declaration of software and hardware environments, enables other researchers to replicate our setup and validate our findings.

D. EVALUATION INDEX AND LOSS FUNCTION

The experiment undertook a segmentation performance evaluation on the lane dataset, employing *Precision*, *Recall*, F_1-*score*, and *Accuracy* as metrics to appraise the model's segmentation efficacy. *Precision* quantifies the ratio of true positives within the positively detected sample set, while *Recall* measures the fraction of true positives against the

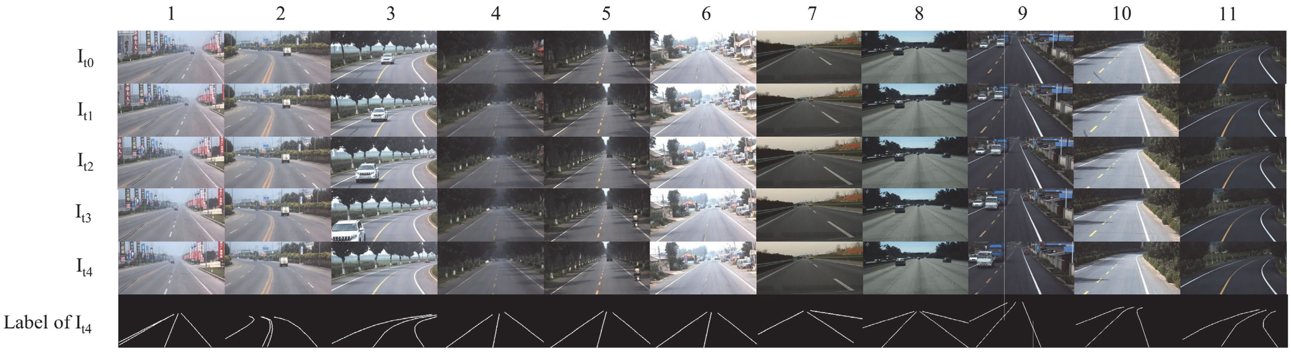


FIGURE 6. Scenarios with different continuous time series on the tvtLANE dataset.

total positive detections. *TP* (*True Positive*) corresponds to the aggregate of pixel points accurately identified as positive, *TN* (*True Negative*) to the sum of pixel points correctly classified as negative, *FP* (*False Positive*) to the sum of pixel points falsely detected as positive, and *FN* (*False Negative*) to the sum of pixel points erroneously classified as negative. The calculation formulas for *Precision*, *Recall*, F_1-*score*, and *Accuracy* are shown as (5)-(8). The selection of *Precision*, *Recall*, F_1-*score*, and *Accuracy* as metrics for evaluating lane detection models is grounded in their ability to provide a holistic understanding of model performance. *Precision* is critical for assessing the model's accuracy in predicting lane markings, minimizing false positives, which is paramount in avoiding misguidance in autonomous driving scenarios. *Recall* measures the model's ability to detect all relevant instances of lane markings, ensuring that the vehicle can recognize and follow the correct path. The F_1-score is employed as a harmonic mean of precision and recall, offering a balanced metric that accounts for the trade-off between identifying lane markings accurately and ensuring no relevant markings are missed. Lastly, *Accuracy* serves as an overarching measure of the model's performance across all predictions, giving a straightforward indication of its effectiveness in lane detection tasks. Collectively, these metrics rigorously evaluate the model's capability to perform with high reliability and precision, essential qualities for autonomous driving systems.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F_1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

$$Accurary = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

For this study, the weighted binary cross-entropy loss function was adopted, as delineated in (9).

$$L = -\frac{1}{N}\sum_{i=1}^{N}[\omega \cdot y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \tag{9}$$

Here, y_i signifies the actual label for the i^{th} sample, p_i denotes the predicted label for the i^{th} sample, and N represents the total sample count. Given the stark disparity between the pixel counts of lane markings and background regions, and the imbalanced nature of the positive and negative samples, a weight ratio $\omega = 0.02{:}1.02$ was judiciously chosen.

Additionally, given the high demand for real-time performance in lane detection, the *Time* metric was used to denote the detection duration for a single image, thus evaluating the model's inference speed.

E. EXPERIMENTAL RESULTS ON TVTLANE DATASET

The experiments were designed to probe the influence of SCNN enhancements and the selection of backbone networks on the lane detection model's performance. Prior research [24] has verified the beneficial impact of incorporating SCNN; therefore, our experimental framework centered around five pivotal variables: the backbone network choice,the method of inputting sequential images into SCNN, SCNN's information flow strategy, the sliding stride size post-SCNN enhancement, and the channel dimensions of feature maps pre- and post-enhanced SCNN input. By systematically varying these parameters, our objective was to meticulously assess their individual and collective effects on model accuracy and real-time execution. The experimental outcomes are presented in Table 4. No. 1-No. 5 utilized Segnet as the backbone network, whereas the remaining configurations adopted ResNet, altering the five critical factors to determine their influence on accuracy and temporal efficiency. To ensure the fairness and comparability of our experiments, we standardized the training process by setting the number of epochs to 10 for each experimental condition.

The findings underscore that models with ResNet as the backbone network generally outperformed those with Segnet in terms of *Accuracy* and F_1-*score*. Notably, the ResNet_$s = (8{,}16)$_64 configuration from No. 6 achieved an exemplary *Accuracy* of 95.4819% and an F_1-*score* of 0.8562. In the realm of *Recall*, the Segnet_$s = (1{,}1)$_64 configuration reached a peak of 0.9996, highlighting its remarkable ability to retrieve positives. The highest *Precision* was recorded in

TABLE 4. Experimental results on tvtLANE dataset.

No.	Backbone Network	Enhanced SCNN		Channels	*Accuracy*	F_1-*score*	*Recall*	*Precision*	*Latency*
		Method	s						
1	Segnet	concatenation	(1,1)	64	95.0940%	0.7170	0.9996	0.8350	0.4453 s
2	Segnet	replacement	(1,1)	128	94.5164%	0.8073	0.6773	0.9992	0.0629 s
3	Segnet	replacement	(1,1)	64	93.6453%	0.7654	0.6204	0.9987	0.2066 s
4	Segnet	replacement	(8,16)	128	93.4064%	0.7526	0.6040	0.9982	0.0267 s
5	ResNet	replacement	(1,1)	64	95.4819%	0.8562	0.7490	0.9993	0.1223 s
6	ResNet	replacement	(8,16)	128	94.5499%	0.8098	0.6809	0.9990	0.0574 s
7	ResNet	replacement	(4,8)	128	95.0725%	0.8391	0.7231	0.9994	0.0579 s
8(proposed)	**ResNet**	**replacement**	**(4,8)**	**64**	**95.2209%**	**0.8433**	**0.7294**	**0.9992**	**0.0517 s**
9	ResNet	replacement	(4,8)	48	95.0804%	0.8342	0.7160	0.9992	0.0558 s
10	ResNet	replacement	(4,8)	80	95.0449%	0.8365	0.7193	0.9992	0.0595 s

No. 8's ResNet_s = (4,8)_128 configuration, at 0.9994. These insights suggest that Segnet may be preferable when the focus is on minimizing false negatives, while ResNet is more adept at delivering high-precision detection. Our proposed model is Resnet_ $s = (4,8)\_64$ of No. 8, which can balance *Precision* and *Recall* better, in addition, *latency* is less.

Following the comprehensive evaluation of the proposed models based on a consistent number of training epochs, model No. 8 emerged as the optimal candidate in terms of performance. In order to enable the model to learn feature distributions in a wider range of challenging scenarios, we introduce the Tusimple dataset and retrain the proposed model. After the initial training phase on the Tusimple dataset, the obtained model weights served as a robust starting point for transfer learning, effectively functioning as pre-trained parameters. Subsequent retraining of model No. 8 on the tvtLANE dataset resulted in marked improvements in various performance indicators. When evaluated against the tvtLANE dataset, the model demonstrated exceptional *Accuracy*, achieving 97.87%, along with an F_1-*score* of 0.943, a *Precision* of 0.897, and a *Recall* of 0.994. This enhancement in performance is attributable to the transfer learning process, which allowed the model to develop a deeper understanding of the feature distribution specific to the tvtLANE dataset, thereby improving its precision in detecting lane markers within the pertinent road segments.

F. SELECTION OF BACKBONE NETWORKS

The backbone networks under scrutiny were Segnet and ResNet34, evaluated for their respective performances in lane detection. Qualitative analysis involved monitoring the variation in model *Accuracy* and F_1-*score* across epochs for the 12 experimental groups, as documented in Table 4 and illustrated in Fig. 7 and Fig. 8.Fig. 7 depicts a trajectory of ascending Accuracy over the training epochs before reaching a plateau. Particularly, No. 6, 9, and 11 achieved superior final accuracy levels,with No. 1 and 6 demonstrating rapid convergence. This trend suggests that models with ResNet as the backbone tend to excel in *Accuracy*.

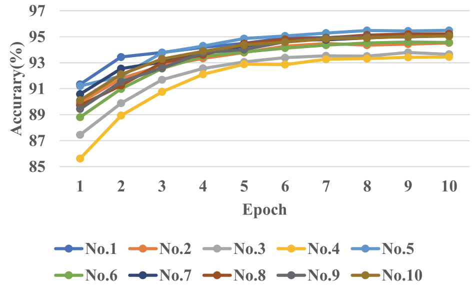


FIGURE 7. Curve of accuracy as a function of epoch.

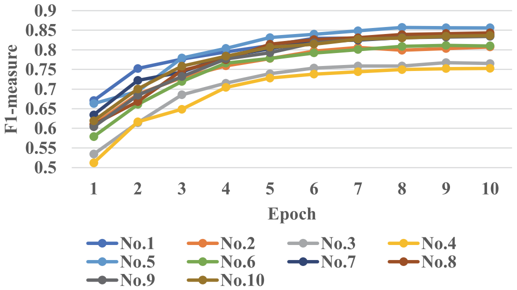


FIGURE 8. Curve of F_1-score as a function of epoch.

Similarly, the evolution of F_1-*score* in Fig. 8 mirrors that of *Accuracy*, with an initial increase followed by stabilization over the course of training epochs. Once more, No. 6, No. 9, and No. 11 attained commendable final F_1-*score*, with No. 1 and No. 6 showing swift convergence rates. This reinforces the superiority of ResNet-backed models in performance.

The advantage of ResNet can be ascribed to its innovative residual connections, which mitigate the vanishing and exploding gradient issues in deep networks. ResNet maintains robust performance even within deeper architectures and can capture a richser spectrum of abstract features, thus amplifying model performance. Consequently, for lane detection tasks that hinge on advanced feature representations, ResNet as the backbone network is often the superior choice, particularly for applications where detection accuracy is paramount.

Nonetheless, when compared to ResNet's benefits, Segnet's streamlined architecture offers an advantage in inference speed due to its elimination of redundant computations in residual connections, significantly curtailing inference time. The experimental data indicates that Segnet-based models approximately double the inference speed of their ResNet counterparts, providing a notable edge for real-time lane detection systems that necessitate swift responses. Hence, Segnet emerges as the more apt option when immediacy is the primary concern.

G. THE METHODOLOGY OF SEQUENTIAL IMAGE INPUT TO THE ENHANCED SCNN

In this study, we evaluated two distinct sequence image input methodologies: pre-stacking prior to input and direct input. The efficacy of these methods was comparatively analyzed in experiments delineated as No. 1 and No.5. Specifically, No. 1 implemented the pre-stacking technique, whereas No. 5 utilized the direct input approach. The results of No. 1 revealed an *Accuracy* of 95.094% and a *Recall* of 0.9996, with F_1-*score* and *Precision* measured at 0.7170 and 0.8350, respectively. By contrast, No. 5 reported an *Accuracy* of 94.2032%, a *Recall* of 0.9632, with the F_1-*score* and *Precision* being 0.7491 and 0.8772, accordingly. Despite a slight reduction in *Accuracy* and *Recall* by 0.89% and 3.64%, respectively, for the pre-stacking method, it demonstrated a notable enhancement in the F_1-*score* and *Precision* by approximately 3.21% and 4.22%. These findings suggest that the pre-stacking input method, while compromising on overall classification accuracy, offers improved performance in terms of the trade-off between false positives and false negatives.

Latency, the measure of temporal cost, was comparable between the two input strategies. The *latency* was 0.4453 s for No. 1 and 0.4293 s for No. 5, with a negligible disparity of merely 0.016 s. The direct input strategy necessitated the processing of five images, each with a channel count of 'c', whereas the pre-stacking approach involved a single image with a channel count of '$5c$'. Batch processing capabilities allowed for concurrent processing of the five images, thereby equalizing the processing time for both input methods. Given that the channel count did not exceed 1000—a threshold beyond which inference times typically increase—the time required for both methodologies was essentially identical.

In conclusion, the pre-stacking input method, despite a modest reduction in *Accuracy* and *Recall*, resulted in gains in F_1-*score* and *Precision*, which may render it more suitable for applications requiring heightened detection accuracy. The negligible difference in temporal overhead between the approaches ensures that users can employ the pre-stacking method without significant concerns about computational delay. This presents a versatile option for the implementation of lane detection algorithms within the constraints of embedded systems or in scenarios demanding real-time computation.

H. INFORMATION FLOW STRATEGIES ADOPTED BY ENHANCED SCNN

In this study, we evaluated two distinct information flow strategies within the Spatial-Temporal Recurrent Neural Network (SCNN): concatenation and replacement. These strategies were compared in No. 1 and No. 3, with No. 1 implementing the concatenation strategy and No. 3 utilizing the replacement strategy. The concatenation approach yielded an *Accuracy* of 95.094% and a *Recall* of 0.9996, accompanied by an F_1-*score* of 0.7170 and a *Precision* of 0.8350. Conversely, the replacement strategy resulted in a marginally reduced *Accuracy* of 94.567% and *Recall* of 0.9912 but demonstrated enhanced *Precision* and F_1-*score* values of 0.8512 and 0.7335, respectively. This suggests that while the replacement strategy may slightly compromise classification *Accuracy*, it offers superior performance in mitigating false positives and false negatives, which is crucial in reducing error rates for lane detection tasks. The capacity to minimize such errors is particularly important; false positives can lead to unwarranted vehicle maneuvers, and false negatives may fail to identify potential hazards, both of which elevate the risk of traffic incidents. The replacement strategy's advantage lies in its improved error trade-off, thereby refining the model's discriminative ability for lane detection.

Moreover, the replacement strategy is time-efficient, requiring only 0.223 s for inference, in contrast to the 0.445 s necessary for the concatenation strategy. This substantial time reduction can be attributed to the fewer operations involved in the replacement strategy's information flow process. Theoretically, the temporal cost for the replacement strategy encompasses convolution and replacement operations, whereas the concatenation strategy involves convolution and concatenation operations. With a sliding stride of 1 and an input feature map size of 64×128 for SCNN, the replacement strategy necessitates 32 convolutions and replacements in the top-down process, compared to the 64 convolutions and 63 concatenations required by the concatenation strategy. The reduced operational count in the replacement strategy, which is approximately half that of its counterpart, accounts for its expedited processing time. The high real-time performance of the replacement strategy is vital for lane detection tasks, where swift and precise responses are essential for maintaining vehicular safety. Conversely, the concatenation strategy, with its more complex transmission process, may result in prolonged inference times, rendering it less suitable for high real-time demand scenarios.

In summary, the replacement strategy outperforms the concatenation strategy in achieving a balance between accuracy and real-time efficiency for lane detection,thus offering significant benefits for the advancement of lane detection systems.

I. ENHANCED SCNN SLIDING STEP SIZE 'S'

The effect of the sliding stride s on the SCNN's performance post-improvement was investigated to substantiate the experimental outcomes and enhance their credibility. SCNN models based on Segnet and ResNet architectures were refined, and sliding strides s of (1,1), (4,8), and (8,16) were tested. These modifications were reflected in No. 2 and No. 4, No. 6 and No. 9, and No. 7 and No. 8, respectively. For

s = (1,1), the model demonstrated the highest performance metrics but also the longest inference time; specifically, No. 2 and No. 4 achieved an *Accuracy* of 95.6%, *Recall* of 0.9985, F_1-score of 0.7255, and *Precision* of 0.8432, with an inference time of 0.55 s. As the sliding stride increased to (4,8), a decrease in performance metrics was observed, along with a substantial reduction in inference time to 0.35 s, as evidenced by No. 6 and No. 9. This indicates that a sliding stride of s = (4,8) provides a more optimal balance between performance and temporal efficiency. Further incrementing the sliding stride to (8,16) resulted in a continued decrease in performance metrics, with No. 7 and No. 8 reporting an *Accuracy* of 93.7%, *Recall* of 0.9910, F_1-*score* of 0.7050, and *Precision* of 0.8190, and an inference time of 0.36 s.

These findings suggest that while s = (1,1) offers a slight improvement in performance metrics at the expense of increased inference time, s = (4,8) achieves a favorable balance between the two, and s = (8,16) demonstrates a decline in performance with no significant time efficiency gain. With $s = 1$, convolution operations are computed layer by layer, which, due to separate computation and memory access for each convolutional layer, results in decreased GPU utilization efficiency and, consequently, a longer processing time. Increasing s from (1,1) to (4,8) compromises the inter-layer information correlation, leading to a reduction in Accuracy-related metrics. Further increasing s from (4,8) to (8,16) employs traditional multi-level convolutions with improved GPU efficiency but does not significantly decrease inference time.

In scenarios where high real-time performance is paramount, such as autonomous highway driving, a medium-sized sliding stride (e.g., s = (4,8)) may be preferable to maintain high *Accuracy* while minimizing inference time. Conversely, in complex urban environments where detection *Precision* is crucial, a smaller sliding stride (e.g., s = (1,1)) should be selected to enhance *Accuracy*.

J. THE FEATURE MAP CHANNEL DIMENSIONS PRE- AND POST-ENHANCED SCNN INTEGRATION

This investigation focused on the impact of channel size variations on the performance of the Spatial-Temporal Recurrent Neural Network (SCNN) by evaluating configurations with 48, 64, 80, and 128 channels. The corresponding results are detailed in No. 8, No. 9, No. 10, and No. 11. Specifically, No. 8, with a channel size of 64, yielded an *Accuracy* of 94.8%, a *Recall* of 0.9952, an F_1-*score* of 0.7183, and a *Precision* of 0.8310, with an inference duration of 0.35 s. A reduction in channel size to 48 (No. 9) led to a slight decrease in performance, albeit with a marginal reduction in inference time. Conversely, increasing the channel size to 80 and 128, as explored in No. 10 and No. 11, did not result in notable performance gains but did incur longer inference times due to increased model complexity. These observations suggest that a channel size of 64 is optimal, providing the SCNN with adequate representational capacity for high performance without incurring unnecessary

In essence, No. 8 stood out by combining representational *Accuracy* with the shortest inference time. The channel size of 64 was found to be sufficient for the SCNN to accurately capture the features of the lane images without restricting the model's capacity or adding complexity that might lead to overfitting or increased computation. In summary, a channel size of 64 strikes an optimal balance between inference speed and representational capability, rendering it ideal for lane detection applications.

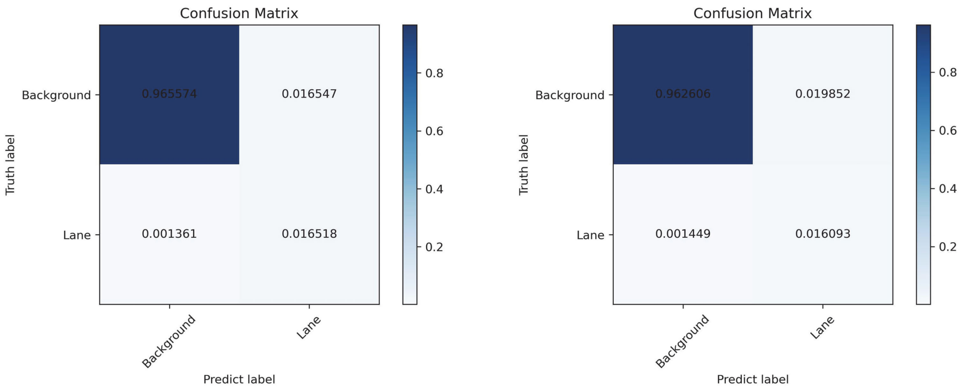
K. CONFUSION MATRIX AND P-R CURVE

In the domain of lane detection, the precise evaluation of a proposed method's performance is critical. The confusion matrix is an established tool for gauging the efficacy of neural networks in discerning various categories, offering a lucid approach to appraise the model's proficiency in identifying lanes (the positive class) versus non-lane regions (the background or negative class).

To elaborate, each column of the confusion matrix aligns with a category as predicted by the model, whereas each row reflects the actual true category. Within the realm of lane detection, attention is predominantly directed towards two categories: lanes and background. Hence, the confusion matrix is categorized into four segments: *True Positive* (*TP*), *False Positives* (*FP*), *True Negative*s (*TN*), and *False Negatives* (*FN*). Here, *TP* denote the accurately detected lanes, and *TN* indicate the correctly identified background. Conversely, *FP* pertain to background erroneously labeled as lanes, and *FN* to lanes that have been overlooked.

For the empirical segment of this research, the confusion matrix is utilized to ascertain the performance of our advanced lane detection model on both the TuSimple and tvtLANE datasets.The TuSimple dataset is an extensively recognized benchmark in lane detection, while the tvtLANE dataset has been expressly compiled and labeled for this investigation, enriching the diversity and complexity of our testing scenarios. The derived confusion matrices from these datasets are depicted in Fig.9. Analysis of these matrices facilitates a profound comprehension of the proposed model's merits and constraints in the lane detection task, in addition to its generalization potential across varied road environments.

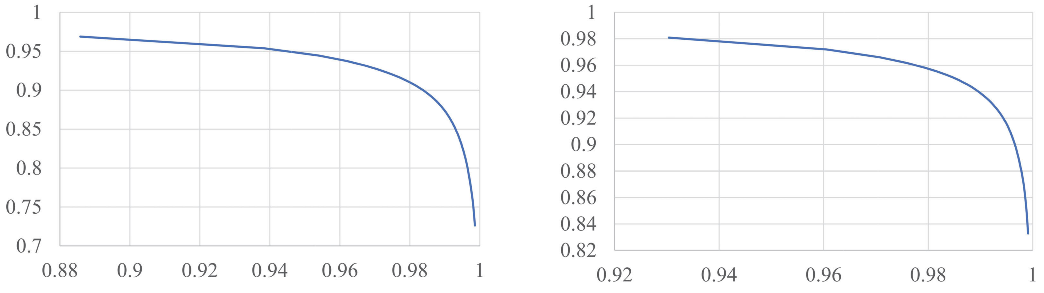
In the domain of lane detection, the *Precision-Recall* (P-R) curve is an invaluable tool that reflects the ability of the model to discern lanes from the surrounding background at various threshold levels. This curve succinctly captures the essential balance between precision—the proportion of true lane detections among all detected lanes—and recall, the proportion of true lane detections out of the lanes present in the dataset, across each threshold. The inherent complexityand variability of driving environments, coupled with the unpredictable nature of road conditions, necessitate a judicious optimization of the threshold to fine-tune the model's performance for lane detection. Our experiments have entailed the plotting of the P-R curve for our proposed hybrid spatial–temporal model for lane detection, utilizing the TuSimple and tvtLANE datasets. These plots are depicted in Fig. 10.



(a) tvtLANE

(b) Tusimple

FIGURE 9. Confusion matrices on different datasets: (a) tvtLANE dataset; (b) Tusimple dataset.



(a) tvtLANE

(b) Tusimple

FIGURE 10. P-R curves on different datasets: (a) tvtLANE dataset; (b) Tusimple dataset.

TABLE 5. Real-time performance analysis.

Method	F_1	*Parameters*	*GFLOPs*	*Latency*	*FPS*
ResNet_*s*=(1,1)_64	0.8562	78.617M	198.789	0.1223	8.177
ResNet_*s*=(4,8)_64	0.8433	**78.389M**	**175.831**	**0.0517**	**19.342**
ResNet_*s*=(8,16)_128	0.8098	78.867M	180.114	0.0574	17.422
ResNet_*s*=(4,8)_128	0.8391	78.867M	179.852	0.0579	17.271

On the tvtLANE dataset, the model's *Recall* hovers around 0.998, signifying its proficiency in detecting the vast majority of true lane markings. Nonetheless, the initial low *Precision* suggests a susceptibility to false positives. In contrast, on the Tusimple dataset, the model sustains a comparably elevated *Recall* without a marked decline. The *Precision* commences at a robust 0.833 and progressively increases, implying that enhancing the model's confidence threshold correlates with a reduction in false positives. These trends affirm the proposed model's efficacy in lane detection.

L. REAL-TIME PERFORMANCE ANALYSIS

In scenarios involving high-speed vehicular travel, the surrounding environment undergoes rapid changes, demanding that lane detection algorithms process high- definition video streams efficiently. It is, therefore, imperative to perform a real-time analysis of lane detection algorithms to evaluate their suitability for use in autonomous or assisted driving systems. The algorithm's efficiency directly influences the response time and reliability of these systems. To ascertain the capability of the lane detection algorithm to perform consistently under dynamic conditions, a real-time analysis was conducted on various methods, as delineated in Table 5.

In real-time analysis, our primary focus is on four metrics: Parameters, *GFLOPs* (Giga Floating Point Operations Per Second), *Latency*, and *FPS* (Frames Per Second). *GFLOPs* is commonly used as a standard for evaluating model computational complexity and performance, with higher *GFLOPs* typically indicating a greater demand for computational resources. *Latency* represents the inference time for a single image, *FPS* is the reciprocal of *latency*, and *Parameters* denote the size of the model parameters.

During the real-time analysis, the ResNet_s =(4,8)_64 configuration was highlighted for its ability to achieve an F_1-*score* of 0.8433, which, although slightly lower than that of the (1,1) stride configuration, was deemed acceptable in light of real-time constraints. Remarkably, ResNet_s = (4,8)_64 demonstrated significant advantages in computational efficiency, with 175.831 *GFLOPs* and a *latency* of 0.0517 s, substantially surpassing the 0.1223 s *latency* and

198.789 *GFLOPs* of the (1,1) stride configuration. This indicates that the algorithm represented by ResNet_s = 4,8)_64 requires fewer computational resources during execution, implying better optimization and lower algorithmic complexity. This model achieved the highest frame rate of 19.342 frames per second (*FPS*), outperforming the 8.177 *FPS* rate of the smaller stride setting. These findings reveal that by optimizing the stride and channel count within the SCNN, the ResNet_s = (4,8)_64 configuration considerably improves the model's real-time processing capabilities, which is integral for autonomous driving systems that require swift decision-making. Consequently, ResNet_s = (4,8)_64 shows promise for delivering both high efficiency and robust performance in lane detection tasks.

To conclude, our experimental outcomes indicate that careful optimization of the backbone network and SCNN parameters can significantly enhance the efficacy of lane detection models. In particular, the integration of an optimized SCNN into a ResNet backbone not only preserves accuracy but also satisfies the real-time processing demands essential for the practical deployment of lane detection systems.

M. VISUALIZATION OF DETECTION OUTCOMES

Visualize the 11 diverse scenes contained in the above Fig. 6. By feeding these sequential images from disparate scenarios into a dozen model groups, we have obtained the detection results for scenarios on the tvtLANE dataset, as illustrated in Fig. 11. The representation of these outcomes is such that for each model group, the inaugural row manifests the model's output lanes detected are emblazoned in red against the backdrop of the original input image; the subsequent row reveals the model's binarized output, exclusively containing lanes, where white denotes detected lanes, and black constitutes the background.

These results conspicuously exhibit disparities in lane detection efficacy among the models when handling diverse scenarios. In the rudimentary Scene 1, all twelve model groups evince commendable detection capabilities. However, in the sunny curved scenarios of Scenes 2 and 3, models from No. 1 to No. 5 render relatively rudimentary detection outcomes, with pronounced deficiencies at the convergence points in the distance; conversely, models from Experiments 6, 9, and 10 approximate the Ground Truth more closely, indicating superior detection outcomes. Scenes 4 to 7 challenge the models' resilience to luminance fluctuations, and No. 1, No. 6, No. 10, and No. 11 demonstrate substantial robustness. In Scene 8, characterized by lane degradation, all models save for those from No. 4, No. 5, and No. 12, capably discern the lanes.In scenes 9-12, except for experiments 1, 4, and 10, the performance of the other models is generally good. We particularly focus on the proposed model, i.e., No. 8, which demonstrates effective adaptability to various challenging environments.

An aggregate analysis of detection outcomes across all scenarios enables us to infer that the selection of the backbone network is a pivotal determinant of detection performance. Models from No. 2 to No. 5 generally underperform relative to those from No. 6 to No. 11, and the inferior results of No. 12 can be ascribed to an overabundance of feature map channels introduced into the SCNN post-expansion via 1 × 1 convolution, engendering a dilution of feature information among feature maps and thereby impeding the model's lane detection efficacy. These insights accentuate the significance of judicious backbone network selection and channel number calibration in the engineering of lane detection models to assure proficient and precise detection across variable environments.In the conducted lane detection experiments, beyond the choice of backbone network, additional variables such as the sequential image input technique into the SCNN, the SCNN's information flow paradigm, the magnitude of the sliding step s, and the dimensionality of the feature map channels all markedly sway the detection outcomes. The pre-input stacking approach outperforms in reconciling false positives with non-detections, while the replacement information flow strategy enhances the precision and dependability of the model more effectively. Modulation of the sliding step s indicates that an intermediate step maintains detection accuracy whilst bolstering the model's inferential velocity, particularly apt for scenarios with stringent real-time exigencies.Optimization of feature map channel dimensions suggests that a median channel count can strike a balance between representational capacity and computational expediency, with a 64-channel configuration exhibiting superior performance across diverse scenarios. These revelations emphasize the imperative of a holistic consideration of various elements in the conceptualization of lane detection models to attain optimal detection outcomes in an array of scenarios. For instance, in scenarios with pronounced lighting variability, an information flow strategy with augmented resistance to interference may be requisite, while in contexts demanding high real-time responsiveness, electing an appropriate sliding step s becomes paramount to balance inferential speed with detection precision.

N. VISUALIZATION RESULTS BASED ON CAM

In the domain of lane detection, given the unpredictability of lighting conditions, obstructions, and road surface statuses, more intuitive visualization techniques are necessitated for the analytical assessment and enhancement of detection results. The visualization methodology previously employed, entailing the generation of binary detection images and the concatenation of detected lanes onto the original image, falls short in affording an in-depth comprehension of the model's decision-making processes.To facilitate a more lucid understanding and corroboration of the model's behavior, Class Activation Mapping (CAM) is utilized for a visualization of heightened interpretability. CAM employs a heatmap to accentuate the image regions that the neural network prioritizes during classification decisions, with intensifying hues toward red denoting heightened neural attention. A heatmap that aligns high response zones with actual lane locations

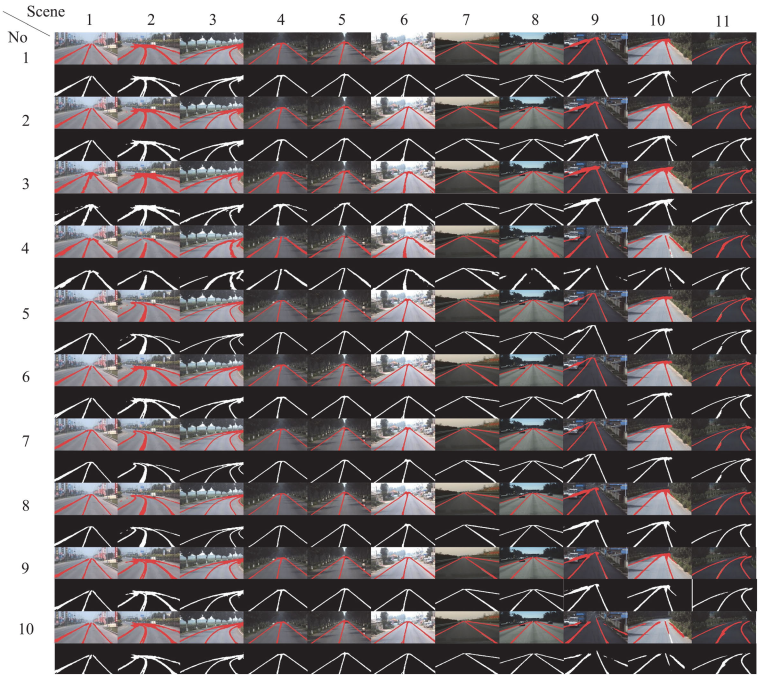
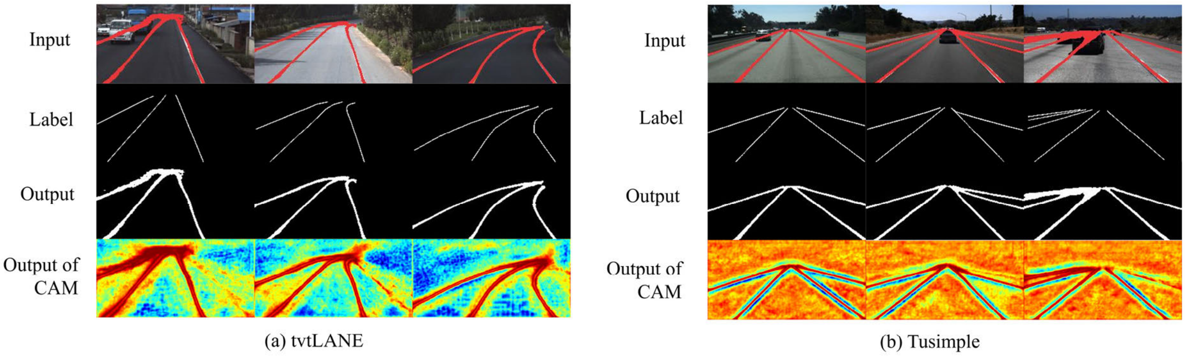


FIGURE 11. The detection results of different scenarios on the tvtLANE dataset.



(a) tvtLANE

(b) Tusimple

FIGURE 12. Visualization results based on CAM on the tvtLANE dataset.

bespeaks the model's robust generalization capacity and reliability. In contrast, high response zones manifesting in non-lane areas indicate susceptibility to occlusions, lighting shifts, and other perturbations. Fig. 12 presents the CAM heatmaps of the proffered model on both the TuSimple and tvtLANE datasets. It can be seen that the pixel values in the lanes show red, while the background parts show yellow or even blue, indicating that the model pays greater attention to the lanes and lower attention to the background.

O. COMPARATIVE ANALYSIS OF VARIOUS LANE DETECTION METHODS ON THE GENERAL DATASET TUSIMPLE

In our rigorous assessment, we benchmarked an array of lane detection algorithms against our proposed model. The selected lane detection models span a spectrum of methodologies, including, but not limited to, Eigenlanes, SCNN, ENet-SAD, RESA, LaneAF, LSTR, FOLOLane, CLRNet, LaneATT, and CondLane, each predicated on distinctive operational principles. For instance, Eigenlanes, CLRNet, and LaneATT employ anchor-based detection; SCNN, ENet-SAD, RESA, and LaneAF engage semantic segmentation; LSTR is predicated on a model-based framework; FOLOLane capitalizes on keypoint estimation; while CondLane executes a row-wise approach to detection.

As delineated in Table 6, our model demonstrates a notable advantage over existing algorithms. Achieving a processing throughput of 19.3 *FPS*, the model attains an exemplary *Accuracy* of 98.21%. When juxtaposed with other models utilizing ResNet and DLA architectures, such as CLRNet with an *Accuracy* of 96.87%, and LaneATT, closely following at 96.83%, our model evidences a substantial *Accuracy* increment of 1.34% and 1.38%, respectively. Moreover, relative to the SCNN, a seminal semantic segmentation-based lane detection approach, our model registers a 1.68% uplift in *Accuracy*. Such results underscore that our proposed model

TABLE 6. Comparison with different algorithms on the Tusimple dataset.

Method	Backbone	*Accuracy*	*FPS*
Eigenlanes [28]	-	95.62%	-
SCNN [9]	VGG16	96.53%	7.5
ENet-SAD [29]	-	96.64%	75
RESA [30]	ResNet34	96.82%	-
LaneAF [31]	DLA-34	95.62%	-
LSTR [32]	ResNet18	96.18%	420
FOLOLane [33]	ERFNet	96.92%	-
CLRNet [34]	ResNet18	96.84%	-
CLRNet [34]	ResNet34	96.87%	-
CLRNet [34]	ResNet101	96.83%	-
LaneATT [35]	ResNet122	96.10%	26
LaneATT [35]	ResNet18	96.84%	-
LaneATT [35]	ResNet34	96.87%	-
CondLane [36]	ResNet18	95.48%	220
CondLane [36]	ResNet34	95.37%	154
CondLane [36]	ResNet101	96.54%	58
Ours	ResNet50	98.21%	19.3

not only excels in Accuracy metrics but also retains the essential attribute of real-time processing capability.

IV. CONCLUSION

To address the challenges of lane detection in complex scenarios characterized by lane wear, uneven illumination, and vehicle obstructions, which lead to suboptimal detection performance and insufficient real-time capabilities, this study redefines the task of lane detection as a binary semantic segmentation problem. Consequently, a novel hybrid spatial-temporal model for lane detection has been successfully developed and empirically validated. This model is designed to process a temporal sequence of lane imagery, striving to meet the stringent demands for accuracy and real-time processing in lane segmentation within real-world applications.

The proposed model consists of four distinct stages. The initial stage involves preprocessing the input images from sequential time frames, which aims to extract salient features and reduce computational complexity by downscaling and amalgamating the images. Following this, the SCNN is enhanced with an increased sliding step and replacement strategies to expedite the flow of feature information. The ResNet-34 architecture, in conjunction with our enhanced SCNN, constitutes the model's encoder. ResNet-34 is tasked with extracting the lanes' global features, encompassing shape and color characteristics, whereas the enhanced SCNN capitalizes on the inherent continuity of lane markings to extract their shape features. Subsequently, the Conv-LSTM module is employed to instantiate the ST-RNN, which extracts features correlating across the temporal image sequence. The final stage involves the decoder, which translates the high-level semantic information elicited by the encoder into pixel-level semantic information, culminating in precise lane detection. The output of the decoder presents the detection outcome for the last temporal frame in the sequence of lane images. Experimental results corroborate that our proposed model not only demonstrates exceptional robustness but also adeptly balances accuracy with real-time performance in comparison to benchmark models. This research introduces an innovative deep learning model design paradigm for the realm of lane detection.

This manuscript delves into the exploration and examination of lane detection in complex environments, employing convolutional neural networks to process sequential imagery. The method's efficacy is substantiated through comprehensive experimentation. Nonetheless, the current state of lane detection in intricate settings is fraught with challenges and obstacles, indicating considerable potential for enhancement. Firstly, there is an imperative to explore more lightweight network architectures to improve the real-time deployment of lane detection, rendering the model more amenable to embedded systems and edge computing devices, hence providing more efficient solutions for real-life applications in automated or assisted driving. Secondly, further investigation into the instance segmentation of multiple lanes is imperative, aiming to refine the perception and adaptability of the lane detection system. This endeavor has implications that extend beyond the advancement of autonomous driving technology, offering tangible industrial value for sectors such as urban traffic management. By integrating these initiatives within our research framework, we seek to steadfastly propel the industrial application of intelligent transportation systems and self-driving vehicles.

REFERENCES

[1] F. Pizzati, M. Allodi, A. Barrera, and F. García, "Lane detection and classification using cascaded CNNs," in *Proc. Int. Conf. 17th Comput. Aided Syst. Theory (EUROCAST)*, 2020, pp. 95–103.

[2] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 286–291.

[3] D. K. Dewangan and S. P. Sahu, "Road detection using semantic segmentation-based convolutional neural network for intelligent vehicle system," in *Data Engineering and Communication Technology*, 2021, pp. 629–637.

[4] O. Jayasinghe, D. Anhettigama, S. Hemachandra, S. Kariyawasam, R. Rodrigo, and P. Jayasekara, "SwiftLane: Towards fast and efficient lane detection," in *Proc. 20th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2021, pp. 859–864.

[5] 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.

[6] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," in *Proc. 16th Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020, pp. 276–291.

[7] H. Ran, Y. Yin, F. Huang, and X. Bao, "FLAMNet: A flexible line anchor mechanism network for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 12767–12778, Nov. 2023.

[8] Y. Sun, J. Li, X. Xu, and Y. Shi, "Adaptive multi-lane detection based on robust instance segmentation for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 888–899, Jan. 2023.

[9] K. Zhou and R. Zhou, "End-to-end lane detection with one-to-several transformer," 2023, *arXiv:2305.00675*.

[10] TuSimple. (2017). *Tusimple Lane Detection Benchmark*. Accessed: Nov. 2021. [Online]. Available: https://github.com/TuSimple/tusimple-benchmark

[11] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020.

[12] M. A.-M. Khan, M. F. Haque, K. R. Hasan, S. H. Alajmani, M. Baz, M. Masud, and A.-A. Nahid, "LLDNet: A lightweight lane detection approach for autonomous cars using deep learning," *Sensors*, vol. 22, no. 15, p. 5595, Jul. 2022.

[13] MVirgo. (2020). *MLND-Capstone: Lane Detection With Deep Learning—My Capstone Project for Udacity's ML Nanodegree*. [Online]. Available: https://github.com/mvirgo/MLND-Capstone

[14] B. T. Passos, M. Cassaniga, A. M. R. Fernandes, K. B. Medeiros, and E. Comunello. *Cracks and Potholes in Road Images*. [Online]. Available: https://data.mendeley.com/datasets/t576ydh9v8/4

[15] P. Shunmuga Perumal, Y. Wang, M. Sujasree, S. Tulshain, S. Bhutani, M. K. Suriyah, and V. U. K. Raju, "LaneScanNET: A deep-learning approach for simultaneous detection of obstacle-lane states for autonomous driving systems," *Expert Syst. Appl.*, vol. 233, Dec. 2023, Art. no. 120970.

[16] R. Bhandari, A. U. Nambi, V. N. Padmanabhan, and B. Raman, "Driving lane detection on smartphones using deep neural networks," *ACM Trans. Sensor Netw.*, vol. 16, no. 1, pp. 1–22, Jan. 2020.

[17] E. Oğuz, A. Küçükmanisa, R. Duvar, and O. Urhan, "A deep learning based fast lane detection approach," *Chaos, Solitons Fractals*, vol. 155, Feb. 2022, Art. no. 111722.

[18] Y. Zhang, Z. Lu, D. Ma, J.-H. Xue, and Q. Liao, "Ripple-GAN: Lane line detection with ripple lane line detection network and Wasserstein GAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1532–1542, Mar. 2021.

[19] D. K. Dewangan and S. P. Sahu, "Lane detection in intelligent vehicle system using optimal 2-tier deep convolutional neural network," *Multimedia Tools Appl.*, vol. 82, no. 5, pp. 7293–7317, Feb. 2023.

[20] J. Zhang and H. Zhong, "Curve-based lane estimation model with lightweight attention mechanism," *Signal, Image Video Process.*, vol. 17, no. 5, pp. 2637–2643, Jul. 2023.

[21] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 5308–5317.

[22] A. Al-Molegi, M. Jabreel, and B. Ghaleb, "STF-RNN: Space time features-based recurrent neural network for predicting people next location," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–7.

[23] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, "Multi-modality multi-task recurrent neural network for online action detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2667–2682, Sep. 2019.

[24] Y. Dong, S. Patil, B. van Arem, and H. Farah, "A hybrid spatial–temporal deep learning architecture for lane detection," *Comput.-Aided Civil Infrastructure Eng.*, vol. 38, no. 1, pp. 67–86, Jan. 2023.

[25] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018.

[26] J. Liu and Y. Gao, "A multi-frame lane detection method based on deep learning," in *Proc. 6th Int. Conf. Cogn. Syst. Inf. Process. (ICCSIP)*, Suzhou, China, Nov. 2021.

[27] A. Gupta and A. Choudhary, "Real-time lane detection using spatio-temporal incremental clustering," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, Oct. 2017, pp. 1–6.

[28] D. Jin, W. Park, S.-G. Jeong, H. Kwon, and C.-S. Kim, "Eigenlanes: Data-driven lane descriptors for structurally diverse lanes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 17142–17150.

[29] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection CNNs by self attention distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 1013–1021.

[30] 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. 35th AAAI Conf. Artif. Intell.*, Vancouver, BC, Canada, Feb. 2018, pp. 3547–3554.

[31] 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.

[32] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3693–3701.

[33] 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.

[34] 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)*, New Orleans, LA, USA, Jun. 2022, pp. 888–897.

[35] 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.

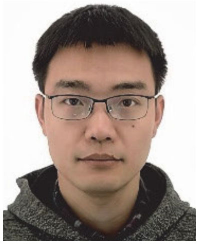
[36] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: A top-to-down lane detection framework based on conditional convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 3753–3762.

JINGANG LI is currently pursuing the bachelor's degree in this field, with research interests focused on deep learning, artificial intelligence, and image processing. He has been enrolled with the Electronic Information Engineering Program, Zhejiang Sci-Tech University, since 2020.

CHENXU MA received the degree from the Electronic Information Engineering Program, Zhejiang Sci-Tech University, in 2020. He is currently pursuing the bachelor's degree. He is passionately exploring the realms of deep learning, artificial intelligence, and image processing within his academic pursuits.

YONGHUA HAN received the B.S. degree in power system and automation and the M.S. degree in power electronics and power transmission from Yanshan University, in 2000 and 2003, respectively, and the Ph.D. degree in agricultural mechanization engineering from Zhejiang University, in 2013. Since 2003, she has been a Teacher with the School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou, China. Her research interests include signal processing and image processing.

HAIBO MU was born in Heilongjiang, China, in June 1979. He received the bachelor's degree in detection and instrumentation from Yanshan University, in 2002. He is currently with Hangzhou Hikvision Digital Technology Company Ltd., mainly engaged in the design and development of video-related products.



LURONG JIANG received the B.S. degree in electronic communication engineering from Zhejiang University, in 2004, the M.S. degree in control theory and control engineering from Hangzhou Dianzi University, in 2009, and the Ph.D. degree in circuits and systems from Zhejiang University, in 2015. Since 2021, he has been an Associate Professor with the School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China. His research interests include signal processing, network science, and wireless sensor networks.