Robust Lane Detection via Spatial and Temporal Fusion

Siyuan Peng[1], Wangshu Yao[1,2,3] and Yifan Xue[3]

[1] School of Computer Science & Technology, Soochow University, Suzhou, China
[2] School of Soft, Soochow University, Suzhou, China
[3] Collaborative Innovation Center of Novel Software Technology and Industrialization
wshyao@suda.edu.cn

Abstract. Lane detection is a crucial and challenging task in autonomous driving. Most existing detection methods only have good results in common scenes, but they have detected poorly in extreme scenarios such as occlusion and strong illumination. To address this problem, this paper introduces a robust lane detection network based on spatial-temporal fusion (LSTnet) for extreme scenarios like occlusion. LSTnet incorporates a detachable local and global memory component as an external storage unit. Through the fusion, read, and update operations on memory features, the component captures temporal information to compensate for the lack of information in extreme detection scenarios. Additionally, LSTnet uses a memory alignment loss function to guide the memory component to update the memory effectively, so as to obtain temporal consistency between the feature maps outputted by the memory component and the ground truth feature maps. Extensive experiments on two commonly used datasets demonstrate that the network achieves an F1 score of 79.49% on CULane and 97.31% on the TuSimple dataset.

Keywords: Lane Detection, Time Series Model, Memory Network.

1 Introduction

In the past decade, autonomous driving technology has gradually become a research hotspot in the field of computer vision, attracting widespread attention from both academia and industry. To ensure the safe operation of autonomous vehicles, it is crucial for autonomous driving systems to accurately understand the spatial information of lane markings. Therefore, it is very important for autonomous driving systems to quickly obtain the shape and position information of the lane markings from the image captured by the front-facing camera.

In recent years, most research has approached lane detection as a segmentation or detection problem. SCNN[1] uses multi-class classification to segment pixels into lane markings or background, but it may also predict pixels unrelated to lane markings. PointLaneNet[2] predicts lane markings based on anchor points. LaneATT[3] uses anchor lines to extend the feature range of anchors and predicts lane instances through rays. Although these methods have achieved satisfactory detection results, their performance tends to degrade in extreme scenarios, such as occlusion. In such cases, extracting more hidden lane information becomes crucial.

In addressing the limitations of the aforementioned work, we propose a robust lane detection network based on spatial-temporal fusion (LSTnet). The network internally incorporates a detachable local and global memory component to capture temporal information to compensate the lack of temporal information in extreme detection scenarios. Firstly, the component processes temporal feature maps in two ways, sequential and shuffled, to obtain local and global memory features, respectively. Then, through fusing the storage memory feature and the current frame's feature, it reads effective memory features. These effective features are used to supplement the temporal information in the network and simultaneously replace the memory features in the component to achieve the iterative update of the memory feature. Besides, since the model's temporal information is read from the component, a memory alignment loss function is designed to align the original annotated feature map and the fused memory feature map. The proposed method effectively handles some extreme detection scenarios while maintaining high accuracy and good real-time performance.

The main contributions of this paper can be summarized as follows:

1. We introduce a robust lane detection network(LSTnet) by fusing spatial-temporal information, aiming to enhance detection performance in extreme scenarios.
2. We design a memory alignment loss function, which enhances the effectiveness of memory storage in the memory component.
3. Experiment results conducted on TuSimple and CULane datasets demonstrate that LSTnet is superior to most existing models and achieves better performance.

2 Related Works

Since lane detection constitutes a sub-task of autonomous driving, the temporal information from video streams can be effectively integrated into the network. Numerous temporal lane detection methods leverage common temporal models to capture the temporal lane information in video sequences, such as LSTM[4], GRU[5], etc. LaneLSTM[6] uses the LSTM structure to process the encoder's output of images and then places the processed image features into the decoder to obtain the final output prediction feature map. LaneGRU[7] uses two GRUs to handle lane detection tasks—one for extracting temporal information and another as the encoder for the input image sequence. Despite the satisfactory performance of the aforementioned methods on the TuSimple dataset, their detection capabilities are less effective on the CULane[1] dataset. This inadequacy is attributed to the inability of these temporal models to accurately capture all the information conveyed in the images, thereby hindering their ability to handle challenging detection scenarios.

To address these aforementioned issues, methods such as MT-Net[8] and VIL-100[9] adopt a memory network instead of common temporal models. They iteratively read and update the externally stored memory to obtain feature information from temporal inputs. In addition, they use an encoder-decoder structure to process the input image, convert it into a feature map with lane spatial information, and then fuse the spatial information with the temporal information read from the memory network. Ultimately, the decoded and fused memory features serve as the final output.

3 Network

3.1 Overview

The LSTnet follows the structure of LaneMP[10]. On this basis, as illustrated in **Fig. 1**, we introduce a detachable memory component to capture lane features of images along the temporal dimension. Specifically, within the memory component, the Local-Global Memory Fusion Module (LGMF) is employed to merge local and global memory features and the Memory Read and Update Module (MRU) combines the current frame features with the fused memory features to read out relevant features. The relevant feature is then incorporated into the encoder to serve as a complement of temporal information for the memory component. Finally, the local and global memory features are updated by fusing the memory features read by MRU with the current frame features. In addition, a memory alignment loss function is designed to guide the storage and read of memories better. The KL loss between the original annotated feature map and the effectively fused memory feature map is used to align the final memory feature with the original image feature map.

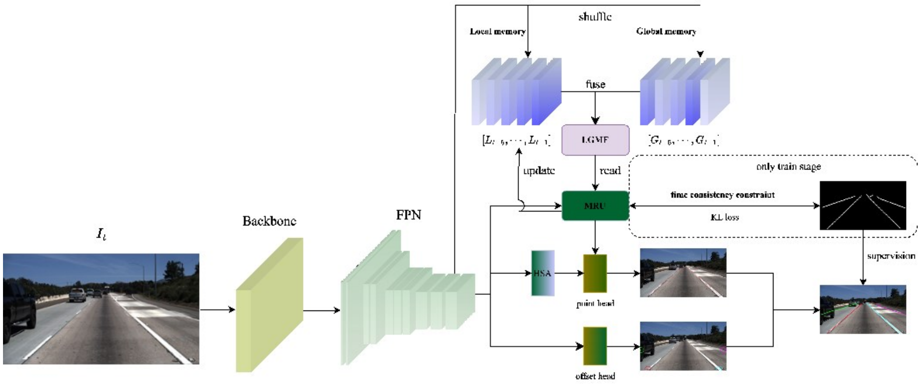


Fig. 1. The structure of LSTnet

3.2 LGMF module

The memory component obtains local memory features and global memory features through sequential and random shuffle. In contrast to traditional methods, this memory component directly utilizes the five saved features from the highest of FPN layers as coarse-grained input sequence features, rather than the original image sequence. The retained local features L are denoted as $[L_{t-5},\cdots,L_{t-1}]$ and global features G are denoted as $[G_{t-5},\cdots,G_{t-1}]$, where $\mathrm{L_i} \in \mathrm{R}^{\mathrm{C\times H\times W}}$, $\mathrm{G_i} \in \mathrm{R}^{\mathrm{C\times H\times W}}$, C is the number of channels, H and W are the feature map size, and $\mathrm{G_i}$ is shuffled randomly by the order of $\mathrm{L_i}$.

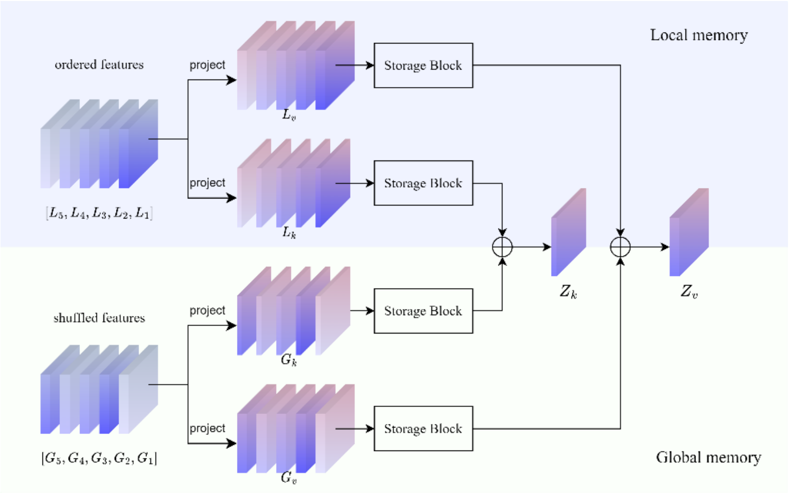


Fig. 2. LGMF module

LSTnet employs the following fusion operations for local memory features and global memory features, as illustrated in **Fig. 2**. Firstly, a 3 × 3 convolution operation with channel numbers C_k and C_v is simultaneously applied to extract key and value maps for local and global memory features. This operation enhances the expressive capability of local and global memory features. The key map L_k , G_k are $[L^k_{t-5}, \cdots, L^k_{t-1}]$, $[G^k_{t-5}, \cdots, G^k_{t-1}]$ with $L^k_i \in R^{C_k \times H \times W}$ and $G^k_i \in R^{C_k \times H \times W}$. The value map L_v , G_v are $[L^v_{t-5}, \cdots, L^v_{t-1}]$, $[G^v_{t-5}, \cdots, G^v_{t-1}]$, with $L^v_i \in R^{C_v \times H \times W}$ and $G^v_i \in R^{C_v \times H \times W}$. Then, after the corresponding key and value maps are obtained, the LGMF module inputs the local memory key map L_k into one Storage Block and the global memory key map G_k into another Storage Block. The output key map of the LGMF module (Z^k_s) is then obtained by summing the outputs of these two Storage Blocks. Simultaneously, L_v and G_v are input into two other distinct Storage Blocks, and summing the output of the two Storage Blocks is the output value map of the LGMF module (Z^v_s). The specific representations of the LGMF module's output key and value maps are provided in Equation (1) and Equation (2), where f_s represents the Storage Block:

$$Z^k_s = f_s \left(L^k_{t-5}, \cdots \right. \qquad \cdots \tag{1}$$

$$Z^v_s = f_s \left(L^v_{t-5}, \cdots \right. \qquad \cdots \tag{2}$$

Fig. 3 illustrates the process of Storage Block aggregating information from the input feature map sequence $(P_{t-5}, \cdots, P_{t-1})$, where input feature map sequence can be either key or value maps from **Fig. 2** (i.e., L_k, G_k, L_v, G_v). Firstly, the Storage Block(SB) concatenates the input feature map sequence, getting $P \in R^{T \times C \times H \times W}$, where C is C_k when the input is the key map and C is C_v when the input is the value map. T represents the length of the input sequence and is set to 5 in this paper. Subsequently, the Storage

Block implements an attention map W with T channels, which includes a 1 × 1 convolutional operation, two consecutive 3 × 3 convolutional operations, and a 1 × 1 convolutional operation. This attention map is used to store the temporal information in the input sequence. Finally, the Block multiplies each channel of W by the input feature map sequence to generate the final output feature map of the Storage Block ($\mathrm{Z_s}$). Therefore, $\mathrm{Z_s}$ can be expressed as: $\mathrm{Z_s} = \sum_{\mathrm{i}=1}^{\mathrm{T}}(\mathrm{W_i} \otimes \mathrm{P_i})$. Here, $\mathrm{P_i}$ is the i-th input feature map to the Storage Block, $\mathrm{W_i}$ is the i-th channel on the temporal dimension of W and $\otimes$ represents the channel-wise multiplication operation between $\mathrm{W_i}$ and $\mathrm{P_i}$.

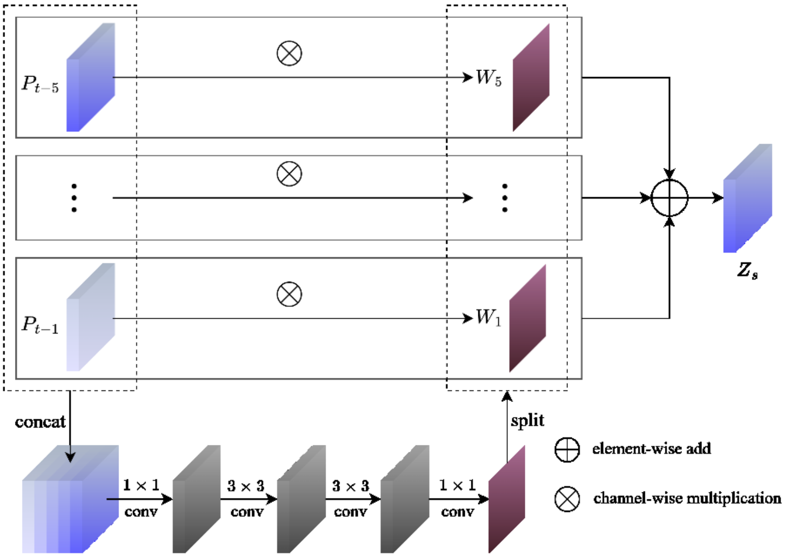


Fig. 3. Storage Block

3.3 MRU module

In Section 3.2, due to sequentially and randomly shuffling the feature maps of FPN layers, LGMF can store local and global features of the input sequence. As shown in **Fig. 4**, to optimize the reading and updating process of memory feature, the MRU module utilizes the current frame's features and the stored memory feature from the LGMF module to obtain the corresponding effective lane memory features. Then, the obtained lane memory feature is used to update the stored memory in the LGMF module. The specific process is outlined as follows:

- Firstly, the MRU module applies two 3 × 3 convolutional layers to the current frame feature map of FPN layer to obtain key map $\mathrm{f_k}$ and value map $\mathrm{f_v}$. The corresponding feature map sizes are $\mathrm{f_k} \in \mathrm{R}^{\mathrm{C_k} \times \mathrm{HW}}$ and $\mathrm{f_v} \in \mathrm{R}^{\mathrm{C_v} \times \mathrm{HW}}$.
- Secondly, a matrix $\mathrm{M_n}$ is obtained by fusing $\mathrm{Z_k}$ with $\mathrm{f_k}$, which preserves the non-local relations between the stored memory information in the LGMF module and the current feature map. The size of $\mathrm{M_n}$ is $\mathrm{R}^{\mathrm{HW} \times \mathrm{HW}}$.

- Thirdly, by merging $\mathrm{M_n}$ with $\mathrm{Z_v}$, the LGMF module obtains the stored effective memory feature $\mathrm{M_m}$. Subsequently, $\mathrm{M_m}$ is concatenated with $\mathrm{f_v}$ along the channel dimension to obtain $\mathrm{M_o}$, which serves as a joint representation of the memory feature for the memory component and input feature maps.
- Finally, the joint memory feature $\mathrm{M_o}$ is aligned with the original annotated feature map to obtain $\mathrm{M_t}$ through a 1×1 convolutional operation to accelerate the learning speed of the memory component, as detailed in Section 3.4. $\mathrm{M_t}$ is then obtained through another 1×1 convolutional operation to obtain $\mathrm{M_u}$, which is used to supplement temporal information and simultaneously update the input sequence of the LGMF module.

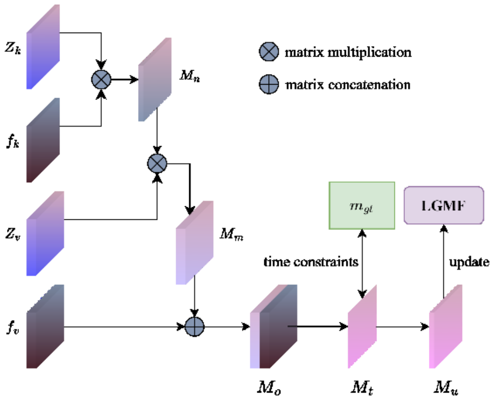


Fig. 4. MRU module

3.4 Time Consistency Constraint

Inspired by TGC-NET[11], to ensure the stability of the memorized feature within the memory component and learn coarse-grained features, we introduce a memory alignment loss function. The loss function is designed to enhance the memory component's ability to represent temporal information, ensuring that the MRU module reads the correct memory feature to achieve the decoder's coarse-grained predictions. The effect of time consistency constraint is measured by Kullback-Leibler(KL) divergence as following function:

$$L_t = KL\left(M_t, resize(M_{gt})\right) \tag{3}$$

As illustrated in **Fig. 4**, $\mathrm{M_t}$ represents the feature map used for temporal alignment in the MRU module, and $\mathrm{M_{gt}}$ represents the ground truth feature map. Due to the size discrepancy between $\mathrm{M_t}$ and $\mathrm{M_{gt}}$, it is necessary to adjust the size of $\mathrm{M_{gt}}$ to match that of $\mathrm{M_t}$. As TuSimple and CULane datasets annotate lanes in the form of points, we use the following scale-up operation to process the annotated point, as shown in Equation (4) :

$$x = \frac{X}{W/w}, y = \frac{Y}{H/h} \tag{4}$$

where H and W are the size of input, h and w are the size of M_t, X and Y are the coordinates of the annotated points in the dataset, and x and y are the coordinates of the points after modifying the feature map's size.

The overall loss function for the entire model, as shown in Equation (5). L_s is the loss function of LaneMP[10], representing spatial information. α and β represent the weights assigned to spatial and temporal information, respectively. Adjusting these coefficients allows the network to achieve a balanced consideration between the spatial information and the temporal information.

$$\begin{gathered} L = \alpha L_s + \beta L_t \\ \text{s.t. } \alpha + \beta = 1 \end{gathered} \tag{5}$$

4 Experiments Settings

4.1 Datasets and Evaluation Metrics

Experiments are conducted on two datasets as follows:

CULane: the F1 score is typically used to represent model performance. Initially, a continuous line area with a uniform width is rendered based on the predicted discrete points. Subsequently, the Intersection over Union (IoU) between the predicted and actual areas is calculated. Lanes with an IoU ≥ 0.5 are identified as true positives (TP), while other lanes are classified as false positives (FP) or false negatives (FN).

TuSimple: a predicted point is considered correct only if it is within a distance of 20 pixels from the corresponding ground truth point. To align with the CULane standard, a lane is considered a true positive (TP) only if the accuracy of the predicted points exceeds 85%.

4.2 Implementation Details

The model adopts the ResNet architecture as the backbone, resulting in two distinct versions of LSTnet denoted as LSTnet-S and LSTnet-M. Based on the image sizes in the CULane and TuSimple datasets, the input image is initially cropped to a size of 800 × 320 to get as much relevant data as possible. For model optimization, the Adam optimizer and poly learning rate decay are employed, with an initial learning rate set to 0.001. The model is trained for 300 and 80 epochs on the TuSimple and CULane datasets, respectively, with a batch size of 32 per GPU. Data augmentation techniques, including random scaling, cropping, horizontal flipping, random rotation, and color transformations, are applied during the training phase. Both training and testing processes are conducted on Tesla-A100 GPUs.

5 Experiment Results

5.1 Results in TuSimple

Table 1 validates the effectiveness of LSTnet on the TuSimple dataset, with an F1 value of 97.31 and FPS(Frames Per Second) of 57. This addition enables the model to handle challenging detection scenarios effectively, consequently reducing the number of false negatives.

Additionally, although LaneLSTM and LaneGRU exhibit high accuracy, their F1 scores show significant gaps compared to other methods. This is attributed to the potential for false negatives when relying solely on temporal models. This further demonstrates that LSTnet fusion of spatial and temporal information is very effective for lane detection.

Table 1. The results on TuSimple

Method	F1	Acc	FP	FN	FPS
SCNN[1]	95.97	96.53	6.17	**1.80**	7.5
UFLDv2[12]	96.16	95.65	3.06	4.61	**312**
LaneATT[3]	96.71	95.57	3.56	3.01	250
Fast-HBNet[13]	-	97.42	**2.26**	2.61	39
Bézier curve[14]	-	95.65	5.10	3.90	150
FOLOLane[15]	-	96.92	4.47	2.28	40
LaneMP-M[10]	96.71	95.82	3.82	2.75	89
ADNet-R34[16]	97.31	96.60	2.83	2.53	-
LaneLSTM[6]	90.40	98.00	-	-	150
LaneGRU[7]	91.24	**98.04**	-	-	-
LSTnet-S(Ours)	97.19	95.53	2.55	3.06	72
LSTnet-M(Ours)	**97.31**	95.63	2.40	2.96	57

5.2 Results in CULane

Table 2 compares the experimental results of the LSTnet model and other models on the CULane dataset and demonstrates the superiority of LSTnet over alternative methods. From **Table 2**, it is evident that the overall performance of the LSTnet model is significantly superior to the other methods. The model exhibits excellent performance in the "Crowded" scene and similarly excels in the "Curve" scene. This is attributed to the introduced memory component, which integrates temporal information into the network. Although the memory component is designed to handle extreme scenarios, the detection of "Normal" scenes is also improved due to the integration of temporal information.

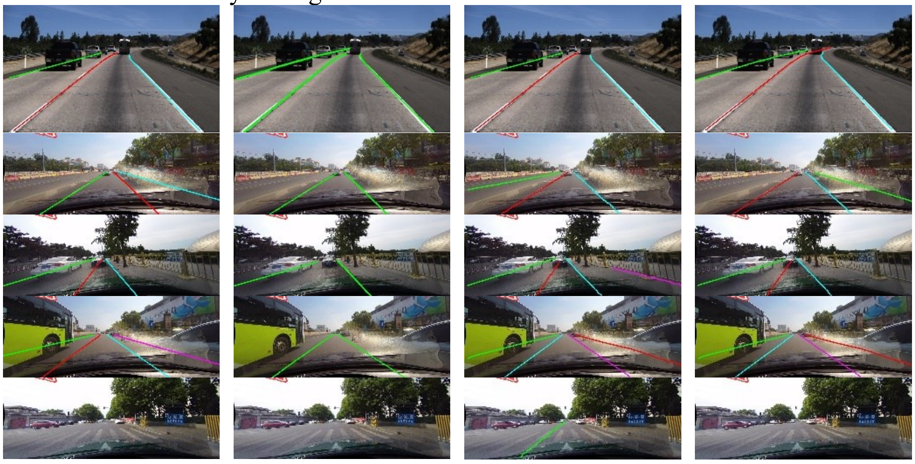
Table 2. The results on CULane

Method	Total	Normal	Crowded	Shadow	Arrow	Curve	Cross	FPS
SCNN[1]	71.60	90.60	69.70	66.90	84.10	64.40	1990	7.5
Fast-HBNet[13]	73.10	91.90	71.60	66.70	85.30	65.10	2306	39
ESAnet[17]	74.20	92.00	73.10	75.10	88.10	68.80	2001	123
PINet[18]	74.40	90.30	72.30	68.40	83.70	65.60	1427	25

LaneATT[3]	75.11	91.17	73.32	69.58	86.62	63.07	1059	250
Bézier curve[14]	75.57	91.59	73.20	76.74	87.16	62.45	**888**	150
UFLDv2[12]	75.90	92.50	74.90	75.30	88.50	70.20	1864	**312**
LaneMP-M[10]	77.11	91.92	76.40	78.24	88.10	72.88	2678	89
CondLane-M[19]	78.74	93.38	77.14	**79.93**	89.89	73.88	1387	152
FOLOLane[15]	78.80	92.70	77.80	79.30	89.00	69.40	1569	40
ADNet-M[16]	78.94	92.90	77.45	79.11	89.90	70.64	1499	77
CANet-M[20]	79.16	93.58	77.88	75.06	90.09	75.54	1176	-
GANet-M[21]	79.39	93.73	77.92	79.49	**90.37**	76.32	1368	127
LSTnet-S(Ours)	78.95	93.28	78.06	78.90	89.99	**76.51**	1291	72
LSTnet-M(Ours)	**79.49**	**93.79**	**78.50**	78.67	89.37	76.10	1338	57

5.3 Qualitative Results

Fig. 5 visualizes the results for LSTnet and other methods. The first row of images is from the TuSimple dataset, while the following four rows are from the CULane dataset. It is noteworthy that in the second and third rows, the LaneMP model exhibits instances of false positives, detecting oncoming lanes and the bottom regions of barriers as lanes. In contrast, LSTnet avoids such occurrences. The detection results show that the performance of the LSTnet surpasses that of other methods. Moreover, the images in the fourth row also indicate that in extreme scenarios such as heavy occlusion, the performance of LSTnet is significantly superior to alternative methods. Additionally, the last row of images demonstrates that in the "Cross" scene, LaneMP tends to predict crosswalks as lanes. In contrast, LSTnet demonstrates effective handling of such situations. This could be attributed to the incorporation of the memory component, which enables the model to effectively distinguish between lanes and crosswalks.



(a) GT lanes (b) LaneATT[14] (c) LaneMP [10] (d) LSTnet-S

Fig. 5. LSTnet and others' qualitative results

5.4 Ablation Study

We conduct ablation experiments using the CULane dataset, employing the LSTnet-S version of the model. As shown in **Table 3**, the weight values of α and β represent the proportions of spatial and temporal information in the entire detection network, respectively. When α = 0.4 and β = 0.6, the model achieves the optimal detection results. The entire table indicates that as the weight value of β increases, the model's performance initially improves but then declines. This trend could be attributed to the possibility that an excessive emphasis on temporal information might lead the model to increase potentially meaningless memory contents, thereby adversely affecting the final detection results.

Table 3. Hyperparameters of total loss

α	β	F1
1	0	77.92
0.5	0.5	78.63
0.4	0.6	**78.95**
0.3	0.7	78.74
0.2	0.8	78.48

6 Conclusion

We introduce a lane detection network named LSTnet, as an enhancement to LaneMP, to integrate spatial and temporal information. LSTnet improves the robustness of lane detection through a detachable memory component comprising the LGMF module and MRU module. Additionally, a memory alignment loss function is proposed to align the feature maps of the original annotations with the fused effective memory features, enhancing the model's ability to represent temporal information. Experimental results on the TuSimple and CULane datasets demonstrate that LSTnet outperforms most existing models in terms of accuracy, particularly excelling in challenging scenarios such as occlusion.

Acknowledgement.

This work was partially supported by Suzhou science and technology plan project (SYG202349), Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial as Deep: Spatial CNN for Traffic Scene Understanding,” *AAAI*, vol. 32, no. 1, pp. 7276–7283, Apr. 2018, doi: 10.1609/aaai.v32i1.12301.

[2] Z. Chen, Q. Liu, and C. Lian, “PointLaneNet: Efficient end-to-end CNNs for Accurate Real-Time Lane Detection,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, pp. 2563–2568. doi: 10.1109/IVS.2019.8813778.

[3] 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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 294–302. doi: 10.1109/CVPR46437.2021.00036.

[4] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.” arXiv, Dec. 11, 2014. doi: 10.48550/arXiv.1412.3555.

[6] 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, doi: 10.1109/TVT.2019.2949603.

[7] J. Zhang, T. Deng, F. Yan, and W. Liu, “Lane Detection Model Based on Spatio-Temporal Network With Double Convolutional Gated Recurrent Units,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6666–6678, Jul. 2022, doi: 10.1109/TITS.2021.3060258.

[8] P. Shi, C. Zhang, S. Xu, H. Qi, and X. Chen, “MT-Net: Fast video instance lane detection based on space time memory and template matching,” *Journal of Visual Communication and Image Representation*, vol. 91, p. 103771, Mar. 2023, doi: 10.1016/j.jvcir.2023.103771.

[9] Y. Zhang *et al.*, “VIL-100: A New Dataset and A Baseline Model for Video Instance Lane Detection,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 15661–15670. doi: 10.1109/ICCV48922.2021.01539.

[10] S. Peng, W. Yao, and Y. Xue, “LaneMP: Robust Lane Attention Detection Based on Mutual Perception of Keypoints,” in *Artificial Neural Networks and Machine Learning – ICANN 2023*, vol. 14261, L. Iliadis, A. Papaleonidas, P. Angelov, and C. Jayne, Eds., in Lecture Notes in Computer Science, vol. 14261. , Cham: Springer Nature Switzerland, 2023, pp. 471–483. doi: 10.1007/978-3-031-44198-1_39.

[11] M. Wang, Y. Zhang, W. Feng, L. Zhu, and S. Wang, “Video Instance Lane Detection via Deep Temporal and Geometry Consistency Constraints,” in *Proceedings of the 30th ACM International Conference on Multimedia*, in MM ’22. New York, NY, USA: Association for Computing Machinery, Oct. 2022, pp. 2324–2332. doi: 10.1145/3503161.3547914.

[12] Z. Qin, P. Zhang, and X. Li, “Ultra Fast Deep Lane Detection With Hybrid Anchor Driven Ordinal Classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–14, Jun. 2022, doi: 10.1109/TPAMI.2022.3182097.

[13] G. Pang, B. Zhang, Z. Teng, N. Ma, and J. Fan, “Fast-HBNet: Hybrid Branch Network for Fast Lane Detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15673–15683, Sep. 2022, doi: 10.1109/TITS.2022.3145018.

[14] Z. Feng, S. Guo, X. Tan, K. Xu, M. Wang, and L. Ma, “Rethinking Efficient Lane Detection via Curve Modeling,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (CVPR), New Orleans, LA, USA: IEEE, Jun. 2022, pp. 17041–17049. doi: 10.1109/CVPR52688.2022.01655.

[15] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, "Focus on Local: Detecting Lane Marker from Bottom Up via Key Point," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 14117–14125. doi: 10.1109/CVPR46437.2021.01390.

[16] L. Xiao, X. Li, S. Yang, and W. Yang, "ADNet: Lane Shape Prediction via Anchor Decomposition," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6404–6413. Accessed: Nov. 24, 2023. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Xiao_ADNet_Lane_Shape_Prediction_via_Anchor_Decomposition_ICCV_2023_paper.html

[17] M. Lee, J. Lee, D. Lee, W. Kim, S. Hwang, and S. Lee, "Robust Lane Detection via Expanded Self Attention," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022, pp. 1949–1958. doi: 10.1109/WACV51458.2022.00201.

[18] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key Points Estimation and Point Instance Segmentation Approach for Lane Detection," *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 7, pp. 8949–8958, Jul. 2022, doi: 10.1109/TITS.2021.3088488.

[19] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: a Top-to-down Lane Detection Framework Based on Conditional Convolution," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 3753–3762. doi: 10.1109/ICCV48922.2021.00375.

[20] Z. Yang *et al.*, "CANet: Curved Guide Line Network with Adaptive Decoder for Lane Detection," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece: IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/ICASSP49357.2023.10096282.

[21] J. Wang *et al.*, "A Keypoint-based Global Association Network for Lane Detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 1382–1391. doi: 10.1109/CVPR52688.2022.00145.