VIL-100: A New Dataset and A Baseline Model for Video Instance Lane Detection

Yujun Zhang[1]*, Lei Zhu[2]*, Wei Feng[1]†, Huazhu Fu[3], Mingqian Wang[1],
Qingxia Li[4], Cheng Li[1], and Song Wang[1,5]
[1] Tianjin University, [2] University of Cambridge, [3] Inception Institute of Artificial Intelligence,
[4] Automotive Data of China (Tianjin) Co., Ltd, [5] University of South Carolina

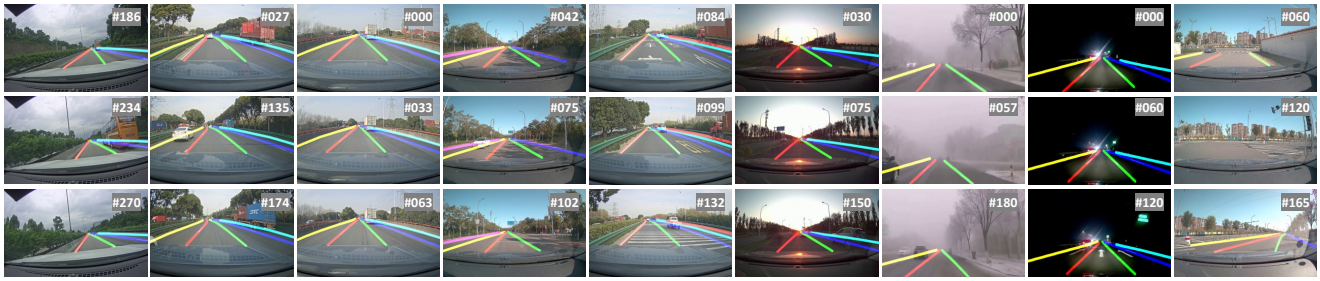


Figure 1: Our proposed Video Instance Lane Detection (VIL-100) dataset contains different real traffic scenarios, and provides the high-quality instance-level lane annotations.

Abstract

Lane detection plays a key role in autonomous driving. While car cameras always take streaming videos on the way, current lane detection works mainly focus on individual images (frames) by ignoring dynamics along the video. In this work, we collect a new video instance lane detection (VIL-100) dataset, which contains 100 videos with in total 10,000 frames, acquired from different real traffic scenarios. All the frames in each video are manually annotated to a high-quality instance-level lane annotation, and a set of frame-level and video-level metrics are included for quantitative performance evaluation. Moreover, we propose a new baseline model, named multi-level memory aggregation network (MMA-Net), for video instance lane detection. In our approach, the representation of current frame is enhanced by attentively aggregating both local and global memory features from other frames. Experiments on the new collected dataset show that the proposed MMA-Net outperforms state-of-the-art lane detection methods and video object segmentation methods. We release our dataset and code at `https://github.com/yujun0-0/MMA-Net`.

1. Introduction

In recent years, autonomous driving has received numerous attention in both academy and industry. One of the most fundamental and challenging task is lane detection in traffic scene understanding. Lane detection assists car driving and could be used in advanced driving assistance system (ADAS) [28, 27, 42]. However, accurately detecting lanes in real traffic scenarios is very challenging, due to many harsh scenarios, e.g., severe occlusion, bad weather conditions, dim or dazzle light.

With the advancement of deep learning, lane detection has achieved significant progress in recent years by annotating and training on large-scale real data [25, 37, 14, 30, 33]. However, most of the existing methods are focused on image-based lane detection, while in autonomous driving, car camera always collects streaming videos. It is highly desirable to extend deep-learning based lane detection from image level to video level since the latter can leverage temporal consistency to resolve many in-frame ambiguities, such as occlusion, lane damage etc. The major obstacle for this extension is the lack of a video dataset with appropriate annotations, both of which are essential for deep network training. Existing lane datasets (e.g., TuSimple [43], Culane [33], ApolloScape [15] and BDD100K [49]) either support only image-level detection or lack temporal instance-level labels. However, according to the United Nations Regulation No.157 [1] for autonomous and connected vehicles, continuous instance-level lane detection in videos is indispensable for regular/emergency lane change, trajectory planning, autonomous navigation, etc.

To address above issues, in this work, **we first collect a new video instance lane detection (VIL-100) dataset** (see

*Yujun Zhang and Lei Zhu are the joint first authors of this work.
†Wei Feng (wfeng@ieee.org) is the corresponding author.

Figure 1 for examples). It contains 100 videos with 10,000 frames, covering 10 common line-types, multiple lane instances, various driving scenarios, different weather and lighting conditions. All the video frames are carefully annotated with high-quality instance-level lane masks, which could facilitate the community to explore further in this field. Second, **we develop a new baseline model, named multi-level memory aggregation network (MMA-Net).** Our MMA-Net leverages both local memory and global memory information to enhance multiple CNN features extracted from the current key frame. To be specific, we take past frames of the original video to form a local memory and past frames of a shuffled ordered video as a global memory, and the current video frame as the query is segmented using features extracted from video frames of both local memory and global memory. A local and global memory aggregation (LGMA) module is devised to attentively aggregate these multi-level memory features, and then all CNN features are integrated together to produce the video instance lane detection results. Finally, **we present a comprehensive evaluation of 10 state-of-the-art models on our VIL-100 dataset**, making it the most complete video instance-level lane detection benchmark. Results show that our MMA-Net significantly outperforms existing methods, including single-image lane detectors [25, 37, 14, 30, 33], and video instance object segmentation methods [18, 50, 32, 24, 44].

2. Related Works

Lane detection datasets. Large-scale datasets are important for deep learning methods. Several public datasets, such as Caltech-Lanes [2], TuSimple [43], Culane [33], ApolloScape [15] and BDD100K [49], have been used for lane detection study. Table 1 provides a comparison of VIL-100 and other public available datasets from different perspectives. Caltech-Lanes only contains 1,224 images and is usually not used for training deep networks, while TuSimple and Culane provide large-scale image data with instance-level lane annotations. However, we are interested in video instance lane detection in this paper, for which both TuSimple and Culane are not applicable. BDD100K and ApolloScape are two large-scale video datasets for driving. However, these two datasets do not provide annotations of lane instances – on each frame, multiple lanes of the same kind are not separated and annotated with one label. Lane-instance detection is important for regular/emergency lane change, trajectory planning, autonomous navigation in autonomous driving, and we provide video-level lane-instance annotations on our collected VIL-100 dataset. In our VIL-100, we increase six lanes annotated at a time by including more complex scenes. In addition, we annotate the relative location of each lane to the camera-mounted vehicle in VIL-100 and such location information was not annotated on any previous datasets.

Lane detection. Early lane detection methods mostly relied on hand-crafted features, such as color [41, 45], edge [9, 22, 27] and texture [23]. Recently, the use of deep neural networks [39, 16, 30] has significantly boosted the lane detection performance. In VPGNet [20], vanishing points were employed to guide a multi-task network training for lane detection. SCNN [33] specifically considers the thin-and-long shape of lanes by passing message between adjacent rows and columns at a feature layer. SAD [14] and inter-region affinity KD [13] further adopt the knowledge distillation to improve lane detection. PolyLaneNet [3] formulates the instance-level lane detection as a polynomial-regression problem, and UFSA [37] provides ultra-fast lane detection by dividing the image into grids and then scanning grids for lane locations. Recently, GAN [10] and Transformer [25] are also used for detecting lanes. Different from the above methods that detect lanes from individual images, this paper addresses video lane detection, for which we propose a new VIL-100 dataset and a baseline MMA-NET method.

Video object segmentation. General-purpose video object segmentation (VOS) methods can also be adapted for video lane detection. Existing VOS methods can be divided into two categories: zero-shot methods and one-shot methods. They differ in that the latter requires the true segmentation on the first frame while the former does not. For zero-shot VOS, traditional methods are usually based on heuristic cues of motion patterns [19, 31], proposals [21, 36] and saliency [8, 47]. Recent deep-learning based methods include the two-stream FCN [7, 17] that integrates the target appearance and motion features. Recurrent networks [34, 44] are also used for video segmentation by considering both spatial and temporal consistency. For the one-shot VOS, earlier methods usually compute classical optical flow [29, 11, 46] for inter-frame label propagation. Recent deep-network based methods [4, 38, 32, 18, 50] include GAM [18], which integrates a generative model of the foreground and background appearance to avoid online fine-tuning. TVOS [50] suggests a deep-learning based approach for inter-frame label propagation by combining the historical frames and annotation of the first frame. STM [32] uses the memory network to adaptively select multiple historical frames for helping the segmentation on the current frame. STM exhibits superior performance on many available tasks and we take it as the baseline to develop our proposed MMA network.

3. Our Dataset

3.1. Data Collection and Split

VIL-100 dataset consists of 100 videos, 100 frames per video, in total 10,000 frames. The fps rate of all the videos

Table 1: Comparisons of our dataset and existing lane detection datasets. '#Frames' column shows the number of annotated-frames and the total number of frames. While TuSimple provides a video dataset, it only annotates the last frame of each video and supports image-level lane detection.

Dataset	Lane detecion on	Size			Diversity					
		#Videos	#Frames	Average Length	Instance-level Annotation	Maximum #Lanes	Lane Location	Line-type	Scenario	Resolution
Caltech Lanes 2008 [2]	Video	4	1224/ 1224	-	✓	4	-	-	light traffic, day	640 × 480
TuSimple 2017 [43]	Image	6.4K	6.4K/ 128K	1s	✓	5	-	-	light traffic, day	1280 × 720
Culane 2017 [33]	Image	-	133K/ 133K	-	✓	4	-	-	multi-weather, multi-traffic scene, day & night	1640 × 590
ApolloScape 2019 [15]	Video	235	115K/ 115k	16s	✗	-	-	13	multi-weather, multi-traffic scene, day & night	3384 × 2710
BDD100K 2020 [49]	Video	100K	100K/ 120M	40s	✗	-	-	11	multi-weather, multi-traffic scene, day & night	1280 × 720
VIL-100(ours) 2021	Video	100	10K/ 10K	10s	✓	6	8	10	multi-weather, multi-traffic scene, day & night	640 × 368 ∼ 1920 × 1080

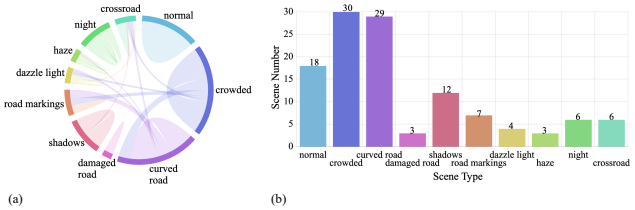


Figure 2: (a) Co-occurrence of different scenarios. (b) Scenario statistics of the proposed VIL-100.

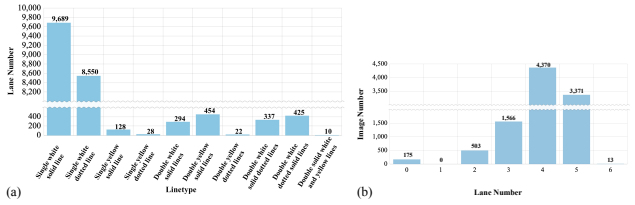


Figure 3: (a) Distributions of 10 line-types in our VIL-100. (b) Video frame statistics of the number of annotated lanes in our VIL-100.

is 10, by down-sampling from 30fps videos. Among these 100 videos, 97 are collected by monocular forward-facing camera mounted near the rear-view mirror. The remaining 3 videos are collected from Internet and they are taken in hazy weather, which increases the complexity and reality of the collected dataset. We consider 10 typical scenarios in data collection: normal, crowded, curved road, damaged road, shadows, road markings, dazzle light, haze, night, crossroad. Other than the normal one, the latter nine of them usually bring more challenges to lane detection. We split the dataset to training set and test set according to the ratio of 8:2, and all 10 scenarios are presented in both training and test sets. This can facilitate the consistent use of and fair comparison of different methods on our dataset.

3.2. Annotation

For each video, we place a sequence of points positioned along the center line of each lanes in each frame and store them in json-format files. Points along each lane are stored in one group, which provides the instance-level annotation in our work. We then fit each group of points into a curve by third-order polynomials, and expand it into a lane-region with a certain width. Empirically, on 1,920 × 1,080 frames, we select the width to be 30 pixels. For lower-resolution frames, the width is reduced proportionally. We further annotate each lane as one of the 10 line-types, i.e., single white solid, single white dotted, single yellow solid, single yellow dotted, double white solid, double yellow solid, double yellow dotted, double white solid dotted, double white dotted solid, double solid white and yellow. In each frame, we also assign an number label to reflect its relative position to the ego vehicle, i.e., an even label $2i$ indicates the i-th lane to the right of vehicle while an odd label $2i-1$ indicates the i-th lane to the left of vehicle. In VIL-100, we set $i = 1, 2, 3, 4$ that enables us to annotate as many as eight lanes in a frame.

3.3. Dataset Features and Statistics

While we consider 10 scenarios in data collection, multiple scenarios may co-occur in the same video, or even in the same frame. Actually 17% of the video frames contain multiple scenarios, and Figure 2 (a) shows the frame-level frequency of such co-occurrence of the 10 scenarios in VIL-100. Meanwhile, one scenario may occur only in part of the video. For example, in video, the scenario may change from 'normal' to 'crossroad', and then get back to 'normal' again in the frames corresponding to 'crossroad', there should be no lane detected. Figure 2 (b) shows the total number of frames for each scenario – a frame with co-occurred sce-

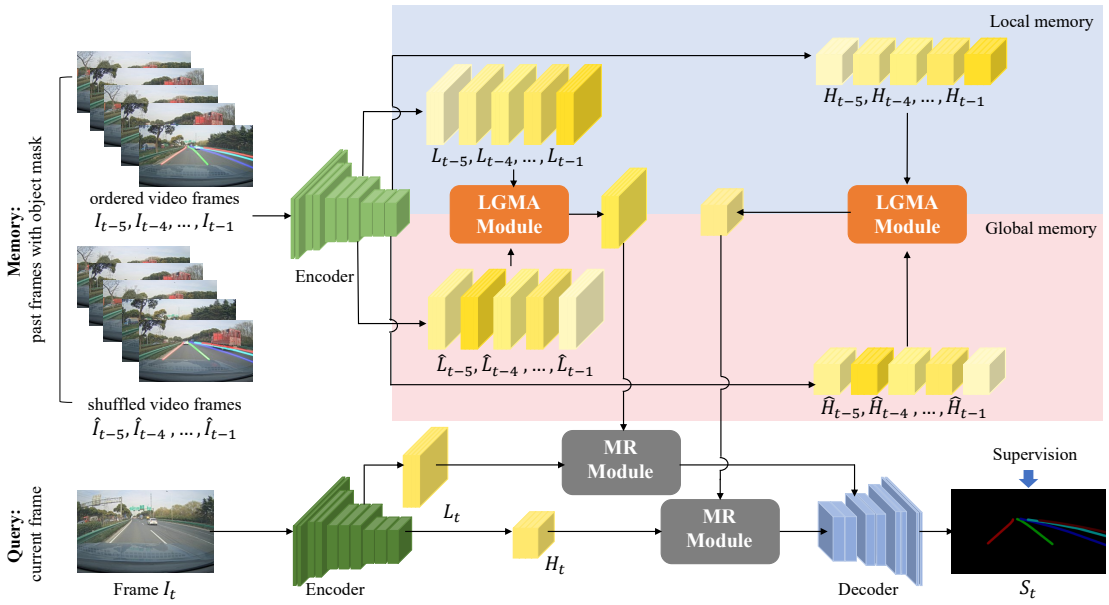


Figure 4: The schematic illustration of our multi-level memory aggregation network (MMA-Net) for video instance lane detection. "LGMA" denotes the local-global memory aggregation module while "MR" module is the memory read module.

narios is counted for all present scenarios.

As shown in Table 1, our VIL-100 contains 10 line-types and provides 6 annotated lanes at most in video frames. Specifically, Figure 3(a) shows the number of annotated lanes for 10 line-types,while Figure 3(b) presents the number of video frames with different annotated lanes, showing that 3,371 video frames have 5 annotated lanes and 13 frames have 6 annotated lanes in our VIL-100.

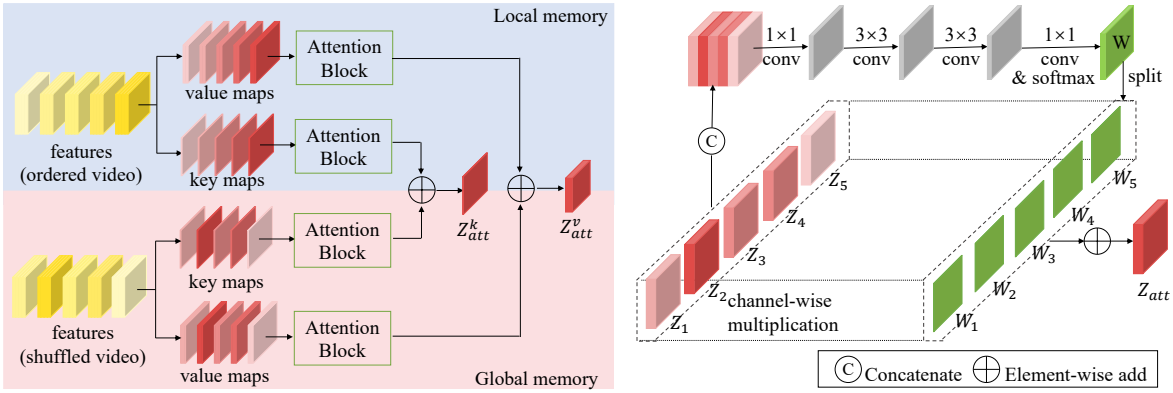
4. Proposed Method

Figure 4 shows the schematic illustration of our multi-level memory aggregation network (MMA-Net) for video instance lane detection. Our motivation is to learn memory-based features to enhance low-level and high-level features of each target video frame for video instance lane detection. The memory features are obtained by integrating a local attentive memory information from the input video and a global attentive memory information from a shuffled video.

Our MMA-Net starts by randomly shuffling an ordered video index sequence $\{1, ..., T\}$ of the input video (T frames) to obtain a shuffled index sequence, which is then utilized to generating a shuffled video by taking all corresponding video frames of the input video based on the shuffled video index sequence. To detect lane regions of a target video frame (i.e., I_t of Figure 4), we then take five past frames ($\{I_{t-5}, I_{t-4}, ..., I_{t-1}\}$) of the original video and five past frames ($\{\widehat{I}_{t-5}, \widehat{I}_{t-4}, ..., \widehat{I}_{t-1}\}$) of the shuffled video as the inputs. Then, we pass each video frame to a CNN encoder consisting of four CNN layers to obtain a high-level feature map (H) and a low-level feature map (L). By doing so, we can construct a local memory (denoted as $\mathcal{M}_l$) by storing five low-level features and five high-level features from $\{I_{t-5}, I_{t-4}, ..., I_{t-1}\}$, and form a global memory (denoted as $\mathcal{M}_g$) to contain five low-level features and five high-level features from $\{\widehat{I}_{t-5}, \widehat{I}_{t-4}, ..., \widehat{I}_{t-1}\}$. After that, we develop a local-global memory aggregation (LGMA) module to integrate all low-level features at $\mathcal{M}_l$ and $\mathcal{M}_g$, and another LGMA module to fuse all high-level features at $\mathcal{M}_l$ and $\mathcal{M}_g$. We use L_{ma} and H_{ma} to denote output features of two LGMA modules. Then, we pass L_{ma} and the low-level features L_t of the target video frame I_t to a memory read (MR) module for enhancing L_t by computing their non-local similarities. We also refine the high-level features H_t of the target video frame I_t by passing it and H_{ma} into another MR module. Finally, we follow other memory networks [32] to adopt a U-Net decoder to progressively fuse features at different CNN layers and predict a video instance lane detection map for the target video frame I_t.

4.1. Local and Global Memory Aggregation Module

Existing memory networks [32, 5, 26, 40, 48] utilized a regular sampling on every N frames to include close and distant frames, but all sampled frames are ordered, and extracted features may depend so much on temporal information. In contrast, we devise a local-global memory aggregation (LGMA) module to utilize five frames from a shuffled video in global memory to remove the temporal order and enhance the global semantic information for detecting lanes. More importantly, due to varied contents of differ-



(a) LGMA Module (b) Attention Block

Figure 5: Schematic illustration of (a) our local and global memory aggregation (LGMA) module, and (b) our attention block. These input five features of LGMA module can be low-level features or high-level features; see Figure 4. And these input features $\{Z_1, Z_2, \ldots, Z_5\}$ of our attention block can be these five key maps or five value maps of our LGMA module.

ent video frames, memory features from different frames should have varied contributions for helping the current video frame to identify the background objects. Hence, we leverage an attention mechanism to learn attention maps to automatically assign different weights on both local memory features and global memory features.

Figure 5 (a) shows the schematic illustration of the developed LGMA module, which takes five features from the input video and five features from the shuffled video. We first follow the original memory network [32] to extract a pair of key and value maps by applying two 3×3 convolutional layers on each input feature map. Then, we pass key maps of the local memory to an attention block for a weighted average on them, fuse key maps of the global memory by passing them into another block, add these features obtained from two attention block to produce a output key map (denoted as $\mathbf{Z}^{\mathrm{k}}_{att}$) of the LGMA module. Meanwhile, we generate of the output value map (denoted as $\mathbf{Z}^{\mathrm{v}}_{att}$) of our LGMA module by adding these generated features of two attention blocks, which aim to aggregate the value maps of the local memory and the global memory, respectively. Mathematically, the output key map $\mathbf{Z}^{\mathrm{k}}_{att}$ and the output value map $\mathbf{Z}^{\mathrm{v}}_{att}$ of our LGMA module are computed as:

$$\begin{aligned}\mathbf{Z}^{\mathrm{k}}_{\mathbf{att}} &= f_{att}(\mathbf{k}^{\mathbf{L}}_{\mathbf{1}}, \mathbf{k}^{\mathbf{L}}_{\mathbf{2}}, \ldots, \mathbf{k}^{\mathbf{L}}_{\mathbf{5}}) + f_{att}(\mathbf{k}^{\mathbf{G}}_{\mathbf{1}}, \mathbf{k}^{\mathbf{G}}_{\mathbf{2}}, \ldots, \mathbf{k}^{\mathbf{G}}_{\mathbf{5}}) \,, \\ \mathbf{Z}^{\mathrm{v}}_{\mathbf{att}} &= f_{att}(\mathbf{v}^{\mathbf{L}}_{\mathbf{1}}, \mathbf{v}^{\mathbf{L}}_{\mathbf{2}}, \ldots, \mathbf{v}^{\mathbf{L}}_{\mathbf{5}}) + f_{att}(\mathbf{v}^{\mathbf{G}}_{\mathbf{1}}, \mathbf{v}^{\mathbf{G}}_{\mathbf{2}}, \ldots, \mathbf{v}^{\mathbf{G}}_{\mathbf{5}}) \,,\end{aligned} \tag{1}$$

where $f_{att}(\cdot)$ denotes an attention block to attentively integrate memory features. $\{\mathbf{k}^{\mathbf{L}}_{\mathbf{1}}, \mathbf{k}^{\mathbf{L}}_{\mathbf{2}}, \ldots, \mathbf{k}^{\mathbf{L}}_{\mathbf{5}}\}$ and $\{\mathbf{v}^{\mathbf{L}}_{\mathbf{1}}, \mathbf{v}^{\mathbf{L}}_{\mathbf{2}}, \ldots, \mathbf{v}^{\mathbf{L}}_{\mathbf{5}}\}$ denote key maps and value maps of five input features of the local memory. $\{\mathbf{k}^{\mathbf{G}}_{\mathbf{1}}, \mathbf{k}^{\mathbf{G}}_{\mathbf{2}}, \ldots, \mathbf{k}^{\mathbf{G}}_{\mathbf{5}}\}$ and $\{\mathbf{v}^{\mathbf{G}}_{\mathbf{1}}, \mathbf{v}^{\mathbf{G}}_{\mathbf{2}}, \ldots, \mathbf{v}^{\mathbf{G}}_{\mathbf{5}}\}$ are key maps and value maps of five input features of the global memory. As shown in our framework of Figure 4, we pass these low-level features of both the local memory and the global memory into a LGMA module to aggregate them for generating a pair of key map and value map (denoted as $L(\mathbf{Z}^{\mathrm{k}}_{att})$ and $L(\mathbf{Z}^{\mathrm{v}}_{att})$). Also, another LGMA module is devised to aggregate the high-level features of both the local memory and the global memory, and these two output key and value maps are denoted as $G(\mathbf{Z}^{\mathrm{k}}_{att})$ and $G(\mathbf{Z}^{\mathrm{v}}_{att})$; see Figure 4.

Attention block. Figure 5 (b) shows the developed attention block to attentively integrate input five feature maps $\{Z_1, Z_2, \ldots, Z_5\}$, which can be the five key maps or five value maps features of Figure 5 (a). Specifically, we first concatenate five input maps and then utilize a 1×1 convolutional layer, two successive 3×3 convolutional layer, a 1×1 convolutional layer, and a Softmax function on the concatenated feature map to produce an attention map W with five feature channels. Then, we multiply each channel of W with these input five maps, and then we add these multiplication results together to produce an output map (Z_{att}) of the attention block. Hence, Z_{att} is computed as

$$Z_{att} = \sum_{i=1}^{5} (\mathbf{W}_i \bigotimes \mathbf{Z_i}) \,, \tag{2}$$

where $\{Z_1, Z_2, \ldots, Z_5\}$ denotes all five input maps of our attention block, and they can be five key maps or five value maps of LGMA module; see Figure 5 (a). W_i is the i-th channel of the attention map W. $\bigotimes$ is the multiplication of W_i and Z_i.

4.2. Implementation Details

Memory read module. Following the original memory network [32], we also develop a memory read (MR) module to retrieve the relevant memory information (i.e, the key and value map of our LGMA module; see Figure 5 (a)) for the query frame (i.e., the target video frame I_t of Figure 4). Specifically, we first apply two 3×3 convolutional layers on features of the query frame I_t to obtain a pair of key map

and value map, Then, the MR module first obtains an non-local affinity matrix by computing similarities between all pixels of the output key map of our LGMA module and the key map of I_t. After that, we multiple the affinity matrix with the output value map of our LGMA module, and then concatenate the multiplication result with the value map of I_t to produce the output features of the MR module.

Decoder. As shown in Figure 4, our network employs two memory read (MR) modules to read the corresponding attentive memory features to enhance features at the 3rd CNN layer and the 4-th CNN layer. After that, the decoder of our network takes the output features of two MR modules to predict the instance-level lane detection result of the target video frame I_t. To do so, we first compress the output features of two MR modules to have 256 channels by a convolutional layer and a residual block. Then, three refinement blocks (see [32] for the details of the refinement block) are employed to gradually fuse two compressed feature maps and these two encoder features at the first two CNN layer, and each refinement block upscales the spatial resolution by a factor of two. Finally, we follow [32] to produce the video instance-level lane detection result from the output features of the last refinement block.

Our training procedure. Like [32], we first train the feature extraction backbone (i.e., encoder of Figure 4) of our network to extract features for each video frame. Specifically, we take the past two frames (only from the input video) of the current video frame (query frame) to construct a local memory, and then employ a memory read (MR) module to read the local memory feature for producing a instance-level lane detection result of the query frame. After that, we take five past frames from the input video and five past frames of a shuffled video of the current video frame (query frame), and the encoder trained in the first training stage to obtain their feature maps, and then follow the network pipeline of Figure 4 to predict an instance-level lane detection result of the target video frame to train our network. In these two training stages, we empirically add a cross entropy loss and a IoU loss to compute the loss of the predicted instance-level lane map and the corresponding ground truth for training.

Training parameters. We implement our MMA-Net using Pytorch and train our network on a NVIDIA GTX 2080Ti. In the first training stage, we initialize the feature extraction backbone by using pre-trained ResNet-50 [12], and employ Adam optimizer with a learning rate of 10^{-5}, a momentum value of 0.9, and a weight decay of 5×10^{-4}. In the second training stage, stochastic gradient descent optimizer is employed to optimize the whole network with the learning rate as 10^{-3}, a momentum value of 0.9, a weight decay of 10^{-6}, and a mini-batch size of 1. The first training stage takes about 14 hours with 100 epochs while the second training stages takes about 7 hours with 50 epochs.

5. Experiments

Dataset. Currently, there is no benchmark dataset dedicated for training video instance lane detection by annotating instance-level lanes of all frames in videos. With our VIL-100, we test the video instance lane detection performance of our network and compared methods.

Evaluation metrics. To quantitatively compare different methods, we first employ six widely-used image-level metrics, including three region-based metrics and three line-based metrics. Three region-based metrics [33, 6] are mIoU, F1(IoU>0.5) (denoted as $\text{F1}^{0.5}$), and F1(IoU>0.8) (denoted as $\text{F1}^{0.8}$), while three line-based metrics are Accuracy, FP, and FN. Apart from image-level metrics [43], we also introduce video-level metrics to consider the temporal stability of the segmentation results for further quantitatively comparing different methods. Video-level metrics are $\mathcal{M}_\mathcal{J}$, $\mathcal{O}_\mathcal{J}$, $\mathcal{M}_\mathcal{F}$, $\mathcal{O}_\mathcal{F}$, and $\mathcal{M}_\mathcal{T}$; please refer to [35] for definitions of these video-level metrics. In general, a better video instance lane detection method shall have larger mIoU, $\text{F1}^{0.5}$, $\text{F1}^{0.8}$, and accuracy scores, as well as smaller FP and FN scores. According to [35], a better video instance segmentation method has larger scores for all video-based metrics.

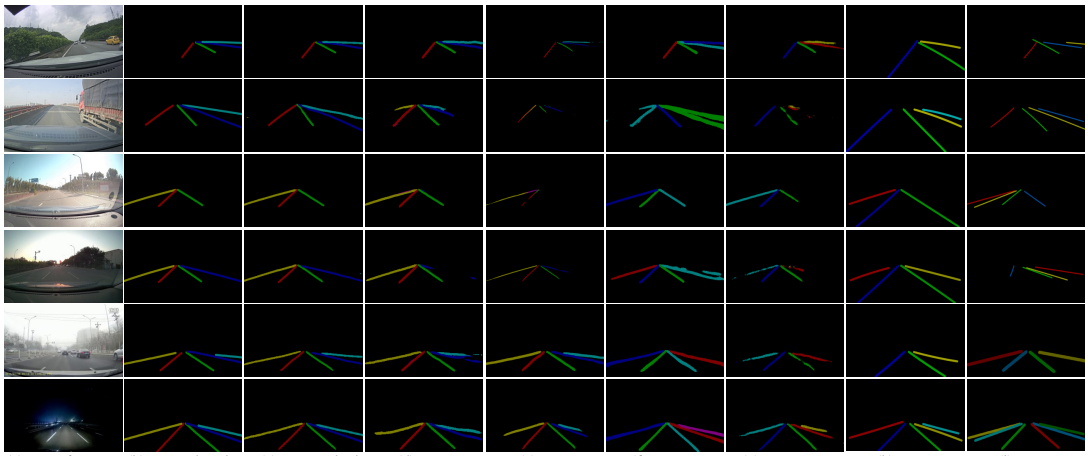
Comparative methods. To evaluate the effectiveness of the developed video instance lane detection method, we compare it against 10 state-of-the-art methods, including LaneNet [30], SCNN [33], ENet-SAD [14], UFSA [37], LSTR [25], GAM [18], RVOS [44], STM [32], AFB-URR [24] and TVOS [50]. Among them, LaneNet, SCNN, ENet-SAD, UFSA, and LSTR are image-level lane detection methods, while GAM, RVOS, STM, AFB-URR and TVOS are instance-level video object detection. Since our work focuses on video instance lane detection, we do not include video binary segmentation methods (e.g., video salient object detection, video shadow detection) for comparisons. For all comparing methods, we use their public implementations, and re-train these methods on our VIL-100 dataset for a fair comparison.

5.1. Comparisons with State-of-the-art Methods

Quantitative comparisons. Table 2 reports six image-level quantitative results of our network and all compared methods. Basically, we can observe that lane detection methods have a better performance on line-based metrics, since they often utilize line-related information (e.g., shape and direction) to infer the lines. By contrast, the VOS methods formulate the lane detection task as a region-based segmentation with abjectness constraint and thus perform better on region-based metrics. Specifically, LaneNet has the best mIoU score of 0.633, STM has the best $\text{F1}^{0.5}$ of 0.756,

Table 2: Quantitative comparisons of our network and state-of-the-art methods in terms of image-based metrics.

Methods	Year	Region			Line		
		mIoU↑	$F1^{0.5}$ ↑	$F1^{0.8}$ ↑	Accuracy ↑	FP ↓	FN ↓
LaneNet [30]	2018	0.633	0.721	0.222	0.858	0.122	0.207
SCNN [33]	2018	0.517	0.491	0.134	0.907	0.128	0.110
ENet-SAD [14]	2019	0.616	0.755	0.205	0.886	0.170	0.152
UFSA [37]	2020	0.465	0.310	0.068	0.852	0.115	0.215
LSTR [25]	2021	0.573	0.703	0.131	0.884	0.163	0.148
GAM [18]	2019	0.602	0.703	0.316	0.855	0.241	0.212
RVOS [44]	2019	0.294	0.519	0.182	0.909	0.610	0.119
STM [32]	2019	0.597	0.756	0.327	0.902	0.228	0.129
AFB-URR [24]	2020	0.515	0.600	0.127	0.846	0.255	0.222
TVOS [50]	2020	0.157	0.240	0.037	0.461	0.582	0.621
MMA-Net (Ours)	2021	**0.705**	**0.839**	**0.458**	**0.910**	**0.111**	**0.105**



(a) Input frames (b) Ground truths (c) Our method (d) STM (e) GAM (f) LaneNet (g) ENet-SAD (h) UFSA (i) LSTR

Figure 6: Visual comparison of video instance lane detection maps produced by our network (3rd column) and state-of-the-art methods (4-th to 9-th columns) against ground truths (2nd column). Please refer to supp. material for more comparisons.

Table 3: Quantitative comparisons of our network and state-of-the-art methods in terms of video-based metrics.

Methods	$\mathcal{M}_\mathcal{J}$ ↑	$\mathcal{O}_\mathcal{J}$ ↑	$\mathcal{M}_\mathcal{F}$ ↑	$\mathcal{O}_\mathcal{F}$ ↑	$\mathcal{M}_\mathcal{T}$ ↑
GAM [18]	0.414	0.203	0.721	0.781	0.568
RVOS [44]	0.251	0.251	0.251	0.251	0.251
STM [32]	0.656	0.626	0.743	0.763	0.656
AFB-URR [24]	0.308	0.251	0.415	0.435	0.362
TVOS [50]	0.255	0.251	0.257	0.256	0.255
MMA-Net (Ours)	**0.679**	**0.735**	**0.848**	**0.873**	**0.764**

and the best $F1^{0.8}$ of 0.327. Regarding Accuracy, FP, and FN, RVOS has the best Accuracy of 0.909; UFSA has the best FP of 0.115; and SCNN has the best FN of 0.110; see Table 2. Compared to these best scores of different metrics, our method has a mIoU improvement of 11.37%, a $F1^{0.5}$ improvement of 10.98%, a $F1^{0.8}$ improvement of 40.06%, a Accuracy improvement of 0.11%, a FP improvement of 3.48%, and a FN improvement of 4.55%.

Moreover, Table 3 summaries video-based metric scores of our network and compared methods. Among results of compared video-based methods, we can find that GAM has the largest $\mathcal{O}_\mathcal{F}$ score (i.e., 0.781), while STM has the best performance of other four video-based metrics. These corresponding best four values of STM are $\mathcal{M}_\mathcal{J}$ of 0.656, $\mathcal{O}_\mathcal{J}$ of 0.626, $\mathcal{M}_\mathcal{F}$ of 0.743, and $\mathcal{M}_\mathcal{T}$ of 0.656. More importantly, our method achieves a further improvement for all five video-based metrics, showing that our method can more accurately segment lanes of different videos. To be specific, our method improves $\mathcal{M}_\mathcal{J}$ from 0.656 to 0.679; $\mathcal{O}_\mathcal{J}$ from 0.626 to 0.735; $\mathcal{M}_\mathcal{F}$ from 0.743 to 0.848; $\mathcal{O}_\mathcal{F}$ from 0.781 to 0.873; and $\mathcal{M}_\mathcal{T}$ from 0.656 to 0.764.

Visual comparisons. Figure 6 visually compares video instance lane detection maps produced by our network and compared methods. Apparently, compared methods neglect some lane regions or wrongly recognized parts of road regions as target lane regions, as shown in 4-th to 9-th columns of Figure 6. Moreover, instance labels of different lanes are also mistakenly detected in video instance

Table 4: Quantitative results of our network with different sampling numbers.

	mIoU ↑	$F1^{0.5}$ ↑	Accuracy ↑	FP ↓	FN ↓
3 frames	0.678	0.816	0.904	0.125	0.116
5 frames (ours)	**0.715**	0.838	0.907	**0.106**	0.111
7 frames	0.705	**0.839**	**0.910**	0.111	**0.105**

Table 5: Quantitative results of our network and constructed baseline networks of ablation study in terms of image-level and video-level metrics.

Measure	Basic	+LM	+GM	+LGM	Ours
mIoU ↑	0.638	0.688	0.670	0.693	**0.705**
$\mathbf{F1}^{0.5}$ ↑	0.758	0.790	0.796	0.822	**0.839**
$\mathbf{F1}^{0.8}$ ↑	0.402	0.425	0.423	0.450	**0.458**
Accuracy ↑	0.857	0.862	0.887	0.897	**0.910**
FP ↓	0.195	0.128	0.150	0.122	**0.111**
FN ↓	0.173	0.163	0.136	0.124	**0.105**
$\mathcal{M}_{\mathcal{T}}$ ↑	0.708	0.706	0.721	0.760	**0.764**
$\mathcal{M}_{\mathcal{J}}$ ↑	0.627	0.632	0.640	0.678	**0.679**
$\mathcal{O}_{\mathcal{J}}$ ↑	0.664	0.676	0.679	0.729	**0.735**
$\mathcal{M}_{\mathcal{F}}$ ↑	0.789	0.781	0.802	0.842	**0.848**
$\mathcal{O}_{\mathcal{F}}$ ↑	0.811	0.795	0.826	0.865	**0.873**

lane detection results of compared methods. On the contrary, our method can more accurately detect lane regions and has correct instance labels for all lanes, and our results are more consistent with the ground truths of Figure 6 (b). In addition, for these challenging cases (i.e, traffic scenes at night or haze weather conditions) at the last two rows, our method also predicts more accurate lane detection maps than competitors, showing the robustness and effectiveness of our video instance lane detection method.

Sampling number. Existing VOS methods usually sampled $3/5/7$ (less than 10) neighboring frames as the input due to the limitation of GPU memory, while memory-based methods (e.g., [30]) required 20 frames in the memory to process a video with 100 frames. Moreover, we also provide an experiment of our network with $3/5/7$ frames in the Table 4, where our method with 5 frames outperforms that with 3 frames significantly, and is comparable with the one employing 7 frames. By balancing the GPU memory and computation consuming, we empirically use 5 frames in our method.

5.2. Ablation Study

Basic network design. Here, we construct four baseline networks. The first one (denoted as "Basic") is to remove the attention mechanism from the local memory, the whole global memory, and the multi-level aggregation mechanism from our network. It means that "Basic" is equal to the original STM but removes the mask initialization of the first video frame. The second one ("+LM") is to add the attention mechanism to the local memory of "Basic" to weighted average local memory features, while the third one ("+GM") is to add the attentive global memory to "Basic". The last baseline network ("+LGM") is to fuse the global memory and the local memory together into "Basic". It means that we remove the multi-level integration mechanism from our network to construct "LGM". Table 5 reports image-based and video-based metric results of our method and compared baseline networks.

Effectiveness of the attention mechanism in memory. As shown in Table 5, "+LM" outperforms "Basic" on all image-level and video-level metrics. It indicates that leveraging the attention mechanism to assign different weights for all memory features, which enables the local memory to extract more discriminative memory features for the query feature refinement, thereby resulting in a superior improving video instance lane detection performance.

Effectiveness of the global memory. "GM" has a better performance of image-based metrics and video-based metrics than "Basic", demonstrating that the global memory has a contribution to the superior performance of our method. Moreover, "LGM" has superior metrics on all metrics over "LM" and "GM". It shows that aggregating the local memory and the global memory can further enhance the query features of the target video frame and thus incurs a superior video instance lane detection performance.

Effectiveness of our multi-level mechanism. As shown in the last two columns of Table 5, our method has larger mIoU, $F1^{0.5}$, $F1^{0.8}$, and Accuracy, smaller FP and FN scores, as well as larger video-based metric ($\mathcal{M}_{\mathcal{J}}$, $\mathcal{O}_{\mathcal{J}}$, $\mathcal{M}_{\mathcal{F}}$, $\mathcal{O}_{\mathcal{F}}$, and $\mathcal{M}_{\mathcal{T}}$) scores than "+LGM". It indicates that applying our LGMA modules on low-level and high-level features of the target video frame enable our network to better detect video instance lanes.

6. Conclusion

To facilitate the research on video instance lane detection, we collected a new VIL-100 video dataset with high-quality instance-level lane annotations over all the frames. VIL-100 consists of 100 videos with $10,000$ frames covering various line-types and traffic scenes. Meanwhile, we developed a video instance lane detection network MMA-Net by aggregating local attentive memory information of the input video and global attentive memory information of a shuffled video as a new baseline on VIL-100. Experimental results demonstrated that MMA-Net outperforms state-of-the-art methods by a large margin. We agree that more diverse scenes/viewpoints could enhance the dataset, and we definitely continue to collect more data in our future work.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (Project No. 62072334, 61902275, 61672376, U1803264).

References

[1] UN Regulation No. 157. https://globalautoregs.com/rules/247-automated-lane-keeping-systems-alks, 2021.

[2] Mohamed Aly. Real time detection of lane markers in urban streets. In *IVS*, pages 7–12, 2008.

[3] Massimo Bertozzi and Alberto Broggi. Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, 7(1):62–81, 1998.

[4] Xi Chen, Zuoxin Li, Ye Yuan, Gang Yu, Jianxin Shen, and Donglian Qi. State-aware tracker for real-time video object segmentation. In *CVPR*, pages 9384–9393, 2020.

[5] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *CVPR*, pages 10337–10346, 2020.

[6] Zhihao Chen, Liang Wan, Lei Zhu, Jia Shen, Huazhu Fu, Wennan Liu, and Jing Qin. Triple-cooperative video shadow detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2715–2724, 2021.

[7] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, pages 686–695, 2017.

[8] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014.

[9] Chao Fan, Lilong Hou, Shuai Di, and Jingbo Xu. Research on the lane detection algorithm based on zoning hough transformation. In *Advanced Materials Research*, pages 1862–1866, 2012.

[10] Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *ECCV Workshops*, pages 0–0, 2018.

[11] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Yuenan Hou, Zheng Ma, Chunxiao Liu, Tak-Wai Hui, and Chen Change Loy. Inter-region affinity distillation for road marking segmentation. In *CVPR*, pages 12483–12492, 2020.

[14] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *ICCV*, pages 1013–1021, 2019.

[15] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2702–2719, 2019.

[16] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

[17] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, pages 2117–2126, 2017.

[18] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, pages 8953–8962, 2019.

[19] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, pages 3271–3279, 2015.

[20] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *ICCV*, pages 1947–1955, 2017.

[21] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011.

[22] Yadi Li, Liguo Chen, Haibo Huang, Xiangpeng Li, Wenkui Xu, Liang Zheng, and Jiaqi Huang. Nighttime lane markings recognition based on canny detection and hough transform. In *RCAR*, pages 411–415, 2016.

[23] Zuoquan Li, Huimin Ma, and Zhengyu Liu. Road lane detection with gabor filters. In *ISAI*, pages 436–440, 2016.

[24] Yongqing Liang, Xin Li, Navid Jafari, and Qin Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *NeurIPS*, 2020.

[25] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *WACV*, pages 3694–3702, 2021.

[26] Xinkai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, pages 661–679. Springer, 2020.

[27] Nicolás Madrid and Petr Hurtik. Lane departure warning for mobile devices based on a fuzzy representation of images. *Fuzzy Sets and Systems*, 291(10):144–159, 2016.

[28] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, 2006.

[29] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, pages 743–751, 2016.

[30] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *IVS*, pages 286–291, 2018.

[31] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, pages 1583–1590, 2011.

[32] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019.

[33] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI*, pages 7276–7283, 2018.

[34] Bo Pang, Kaiwen Zha, Hanwen Cao, Chen Shi, and Cewu Lu. Deep rnn framework for visual sequential applications. In *CVPR*, pages 423–432, 2019.

[35] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016.

[36] Federico Perazzi, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, pages 3227–3234, 2015.

[37] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *ECCV*, pages 276–291. Springer, 2020.

[38] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, pages 7406–7415, 2020.

[39] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by back-propagating errors. *Nature*, 99(6088):533–536, 1986.

[40] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, pages 629–645. Springer, 2020.

[41] Tsungying Sun, Shangjeng Tsai, and Vincent Chan. Hsi color model based lane-marking detection. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 1168–1172, 2006.

[42] Jigang Tang, Songbin Li, and Peng Liu. A review of lane detection methods based on deep learning. *Pattern Recognition*, 111:107623, 2021.

[43] TuSimple. http://benchmark.tusimple.ai, 2017.

[44] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro i Nieto. RVOS: End-to-end recurrent network for video object segmentation. In *CVPR*, pages 5277–5286, 2019.

[45] Jun Wang, Tao Mei, Bin Kong, and Hu Wei. An approach of lane detection based on inverse perspective mapping. In *ITSC*, pages 35–38, 2014.

[46] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):985–998, 2018.

[47] Wenguan Wang, Jianbing Shen, Ruigang Yang, and Fatih Porikli. Saliency-aware video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):20–33, 2017.

[48] Ruizheng Wu, Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Memory selection network for video propagation. In *ECCV*, pages 175–190. Springer, 2020.

[49] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2633–2642, 2020.

[50] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *CVPR*, pages 6949–6958, 2020.