FusionLane: Multi-Sensor Fusion for Lane Marking Semantic Segmentation Using Deep Neural Networks

Ruochen Yin, Yong Cheng, Huapeng Wu, Yuntao Song, *Member, IEEE*, Biao Yu, and Runxin Niu

***Abstract*— Effective semantic segmentation of lane marking is crucial for construction of high-precision lane level maps. In recent years, a number of different methods for semantic segmentation of images have been proposed. These methods concentrate mainly on analysis of camera images, due to limitations with the sensor itself, and thus far, the accurate three-dimensional spatial position of the lane marking could not be obtained, which hinders lane level map construction.This article proposes a lane marking semantic segmentation method based on LIDAR and camera image fusion using a deep neural network. In the approach, the object of the semantic segmentation is a bird's-eye view converted from a LIDAR points cloud instead of an image captured by a camera. First, the DeepLabV3+ network image segmentation method is used to segment the image captured by the camera, and the segmentation result is then merged with the point clouds collected by the LIDAR as the input of the proposed network. A long short-term memory (LSTM) structure is added to the neural network to assist the network in semantic segmentation of lane markings by enabling use of time series information. Experiments on datasets containing more than 14,000 images, which were manually labeled and expanded, showed that the proposed method provides accurate semantic segmentation of the bird's-eye view LIDAR points cloud. Consequently, automation of high-precision map construction can be significantly improved. Our code is available at https://github.com/rolandying/FusionLane.**

Manuscript received July 20, 2019; revised March 3, 2020, June 22, 2020, and September 28, 2020; accepted October 6, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFD0701401, Grant 2017YFD0700303, and Grant 2018YFD0700602; in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences under Grant 2017488; in part by the Key Supported Project in the Thirteenth Five-Year Plan of Hefei Institutes of Physical Science, Chinese Academy of Sciences, under Grant KP-2017-35, Grant KP-2017-13, and Grant KP-2019-16; in part by the Independent Research Project of Research Institute of Robotics and Intelligent Manufacturing Innovation, Chinese Academy of Sciences, under Grant C2018005; and in part by the Technological Innovation Project for New Energy and Intelligent Networked Automobile Industry of Anhui Province. The Associate Editor for this article was L. M. Bergasa. *(Corresponding authors: Ruochen Yin; Biao Yu.)*

Ruochen Yin is with the Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China, also with the University of Science and Technology of China, Hefei 230052, China, and also with the Lappeenranta University of Technology (LUT), 53850 Lappeenranta, Finland (e-mail: rrolland@mail.ustc.edu.cn).

Yong Cheng, Yuntao Song, Biao Yu, and Runxin Niu are with the Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China (e-mail: chengyong@ipp.ac.cn; songyt@ipp.ac.cn; byu@hfcas.ac.cn; rxniu@iim.ac.cn).

Huapeng Wu is with the Mechanical Department, School of Energy Systems, Lappeenranta University of Technology (LUT), 53850 Lappeenranta, Finland (e-mail: huapeng.wu@lut.fi).

Digital Object Identifier 10.1109/TITS.2020.3030767

***Index Terms*— Lane marking, semantic segmentation, LIDAR-camera fusion, convolutional neural network, LSTM.**

I. Introduction

HIGH-PRECISION maps play a central role in autonomous driving. Such maps not only provide high-precision positioning based on map matching, but also disclose complex information about roads and pavements as a priori knowledge for unmanned vehicles, for example, lane limits, slope, curvature, heading, etc. High-precision maps can be seen as a complementary element of the perception module of unmanned vehicles, and they help unmanned vehicles focus on other tasks such as detection and tracking of moving obstacles. To enable autonomous operation, the high-precision lane level map must therefore contain accurate lane marking information.

Convolutional neural networks (CNN) have achieved considerable success in the area of image processing, and a number of different CNN-based image semantic segmentation methods have been proposed. Compare with the bounding box, pixel-wise prediction result is more in line with the requirements of high-precision map construction. However, limitations with the camera itself mean that these methods are unable to provide accurate spatial position information about the lane marking.

Some research has used aerial photography for semantic segmentation of lane marking [1]. This method utilizes aerial images from an unmanned aerial vehicle (UAV) for the semantic segmentation of the lane marking. The main advantages of the approach are low cost and high efficiency, but there are also clear disadvantages. For example, the classification accuracy of some structural similar elements is not high and the segmentation edge of different prediction types is not sufficiently accurate, because the aerial photographs contain a lot of irrelevant background information. Moreover, the real space area corresponding to each pixel in the aerial images is much larger than in the images collected by the ground platform.

To address these problems, the method for semantic image segmentation proposed in this article utilizes a bird's-eye view of the road converted from the LIDAR points cloud instead of a image captured by camera. The overall structure of this method is shown in Fig. 2. The proposed method achieves

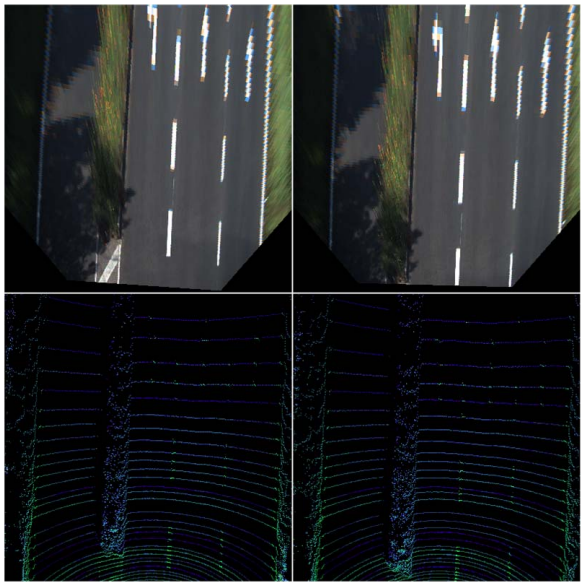

Fig. 1. Comparison of camera and LIDAR data. Two consecutive frames of data in the KITTI dataset collected by a camera (upper row) and LIDAR (lower row). The upper row shows bird's-eye views converted from the front views of the camera. Even if the images are calibrated with the given internal and external parameters of the camera, it can clearly be seen that the camera views are distorted due to the bumpy road surface. The corresponding LIDAR points cloud bird's eye views (lower row) are much more stable. At the same time, the actual physical space corresponding to each pixel becomes larger as the distance from the image sensor increases. Thus, when using camera data, the target details at the top of the image become increasingly blurred. LIDAR points cloud does not have this shortcoming.

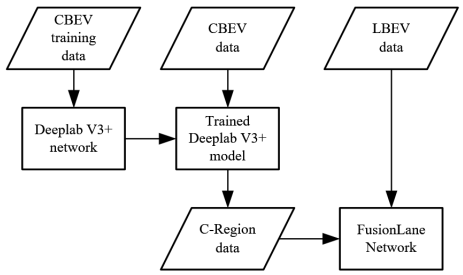

Fig. 2. The overall structure of the method in this article. First, we train the DeepLabV3+ [2] network to achieve semantic segmentation on camera bird's eye view (CBEV) data, we called it as the C-Region. Then, we input the C-Region and LIDAR points cloud bird's eye view (LBEV) data together into the Fusionlane network.

accurate semantic segmentation of LBEV by effectively fusing the data from multiple sensors.

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

- First, to the best of our knowledge,the proposed approach is the first method for lane marking semantic segmentation that utilizes LBEV. This approach has the advantage that the three-dimensional spatial position of each pixel in the LBEV can be accurately obtained, that is to say, accurate position information of each prediction in the semantic segmentation result can be easily obtained, which meets the requirement for high-precision map construction.
- Second, based on the KITTI [4] datasets, a dataset is created that contains more than 14,000 LBEV and CBEV each with manual labeling and extension methods.
- Third, as in practical situations the pixels in the LBEV and CBEV images cannot be perfectly aligned, even after calibration, an encoder module is designed with two input branches instead of a single input with multiple channels. This encoder design allows the network to be independent of the assumption of perfect alignment. Consequently, the result of the CBEV semantic segmentation (hereinafter referred to as the C-Region) can be fused with the LBEV, which enables the segmentation result of the proposed network to have the advantage of both accurate classification from the camera and precise position information from LIDAR. The LSTM [3] structure can help the network achieve better prediction results through the provision of timing information.

The remainder of this article is organized as follows. Section II reviews related works. Section III details the proposed method and network. Section IV compares the proposed method with other methods and analyzes the results. Section V concludes the work by summarizing the advantages of the proposed method and making suggestions for future work.

II. Related Works

A high-precision map is indispensable to autonomous driving. However, construction of high-precision maps is difficult and complicated, and improving the automaticity of the high-precision map construction process and reducing the amount of human participation have long been goals of researchers in the field. In automated map construction, the algorithm not only has to infer semantic information from the input image (the need for scene understanding: process from specific to abstraction) but also be able to make pixel-wise segmentation for each category of target (the need for map construction accuracy: process from abstraction to specific). Semantic image segmentation is able to meet these requirements.

Before deep learning began to be applied to computer vision, researchers generally used Texton Forest [5] or Random Forest [6] method to construct classifiers for semantic segmentation. However, these methods only solved the problem to some extent. Deep learning has revolutionized the field and many computer vision problems, including semantic segmentation, have begun to use the methods based on deep learning frameworks (commonly convolutional neural networks), and the effect achieved far exceeds the traditional method. Therefore, the review of semantic segmentation in this article considers only deep learning frameworks.

A. Semantic Segmentation

At present, all neural networks that have been successfully used for semantic segmentation are derived from the same work utilizing a fully-convolutional neural network (FCN) [7]. In the paper, the authors converted well-known network frameworks such as AlexNet [8], VGG-16 [9], GoogLeNet [10], and ResNet [11] into a fully-convolutional structure replacing the

original fully connected layer in these network frames with the small-scale upsampling layers. Semantic segmentation tasks require the network to have the following two capabilities: the ability to learn multiple scales of features in the image and the ability to accurately restore details of the original image, especially at the edge of the segmentation. To meet these two requirements, researchers have improved the FCN network in the following ways:

1) Encoder Variants: To solve the first problem, some models [12],[13] resize the input for several scales and fuse the features from all the scales. In another approach, Farabet *et al.* [14] transform the input image through a Laplacian pyramid, feed each scale input to a deep convolutional neural network (DCNN) and merge the feature maps from all the scales. Other work employs spatial pyramid pooling to capture context at several ranges [15], [16]. DeepLabv2 [17] proposes atrous spatial pyramid pooling (ASPP), where parallel atrous convolution layers with different rates capture multi-scale information.

2) Decoder Variants: Some researchers adopt an approach that uses various upsampling methods in the decoder module. The aim is mainly to improve the ability of the decoder to restore the details of the original picture. In [7] and [18], deconvolution [19] is employed to learn the upsampling of low resolution feature responses. SegNet [20] reuses the pooling indices from the encoder and learn extra convolutional layers to densify the feature responses. FCN and DeepLabV3+ use the bilinear upsampling in the decoder module and connect the upsampling outputs with the low-level feature from the encoder module. DeepLabv2 used a Conditional Random Field (CRF) to improve the segmentation effect.

B. Semantic Segmentation for Lane Marking

Various approaches have been used in semantic image segmentation for lane marking, for example, Chen *et al.* [2] use the aerial photography for lane marking semantic segmentation. This method can efficiently and quickly complete large-area lane marking semantic segmentation tasks, but there are two shortcomings: surrounding scenes in the image easily interfere with the result; and there is less robustness in areas where the illumination changes significantly. Zou *et al.* [21] combine the ConvLSTM with an encoder-decoder structure DCNN and use the time context information in the lane marking semantic segmentation task. This approach results in some improvement in the segmentation effect, but there are only two classification results, background and lane marking.

III. The Proposed Method

As shown in Figure 2, in order to achieve effective semantic segmentation for LBEV, this article first implements the transfer learning of the DeepLabV3+ network on the CBEV training set, then inputs the C-region and LBEV data into the Fusionlane network for learning. So in this section, we will first describe the preprocessing of the data utilized in our study and then introduce the proposed multi-sensor fusion deep neural network.

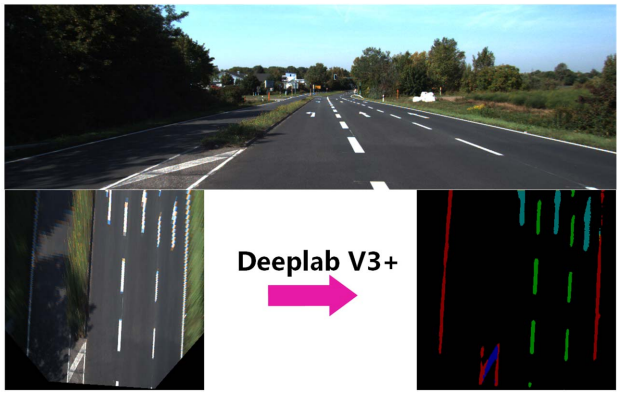

Fig. 3. The semantic segmentation result of CBEV (C-Region) acquisition process.

A. Data Preprocessing

The network contains two input branches – data from the camera and data from LIDAR, and we will refer to input from the C-Region as Branch C and input from the LBEV as Branch L. Data from the camera and LIDAR need to be preprocessed to meet the input requirements of the proposed network.

1) Acquisition of C-Region: First, we transform the front view of the camera into an bird's eye view through inverse perspective mapping. The CBEV is a 400 by 400 pixel image showing an area of 26 meters to 6 meters in front and 10 meters on each side. That is to say, each pixel represents an area of 5 cm by 5 cm in real space. The DeepLabV3+ network trained on our labeled dataset is then used to semantically segment the CBEV to get the input data for Branch C, as show in Fig. 3. It should be noted that, C-Region has been colored into an RGB image for convenience of observation.

2) LBEV Design and Generation: In generation of the LBEV, we intercept the same region of interest as the CBEV for the original points cloud acquired by three-dimensional LIDAR. Based on the height information of the points cloud, the height threshold of the region of interest is between -2 meters and -1 meters (the installation height of the LIDAR device is about 1.73 meters above ground). Here, we are not simply projecting the points cloud into a two-dimensional grayscale image but transforming it into a three-channel bird's eye view, as shown in the second row of Fig. 1. As in the CBEV, each pixel of the LBEV corresponds to a 5cm by 5cm real space.

The value of the first channel of the LBEV corresponds to the intensity of the points spot falling within the grid. The value of the first channel is calculated as follows:

$$F(x, y) = \frac{\sum_1^n i}{n} \times 255 \tag{1}$$

where $F(x, y)$ is the value of the first channel, $i_1, i_2, \ldots, i_n, i \in [0, 1]$ is the reflection intensity value of each point falling within the grid corresponding to the pixel, and n is the number.

The value of the second channel corresponds to the average height information of the points, which is calculated as

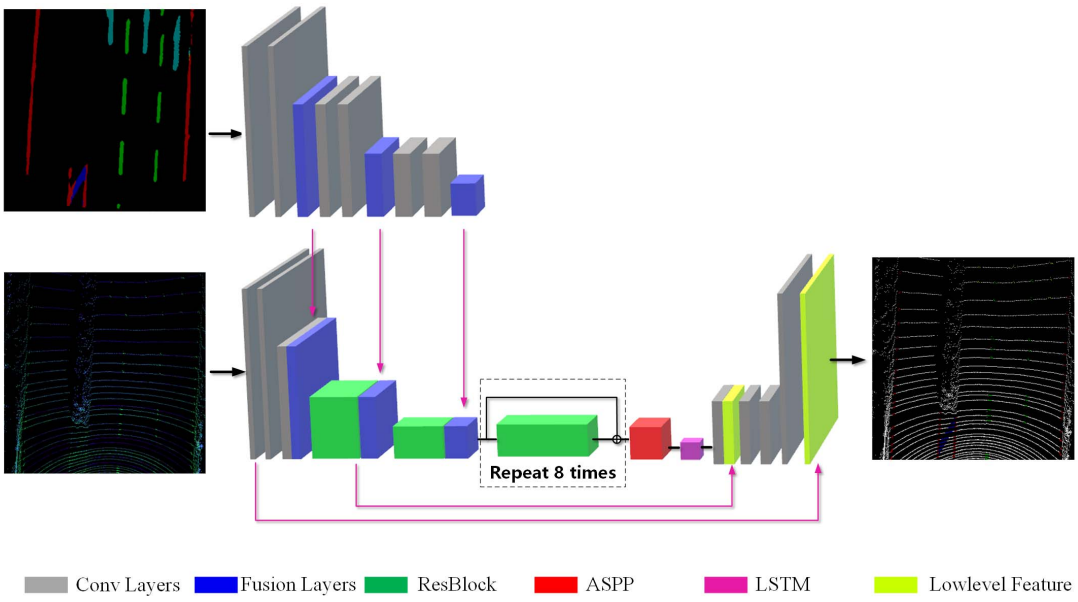

Fig. 4. The network structure diagram, this diagram mainly shows the general structure of the proposed network, the details will be described below.

follows:

$$S(x, y) = \frac{\sum_1^n (h + 2)}{n} \times 255 \tag{2}$$

where $S(x, y)$ is the value of the second channel, $h_1, h_2, \ldots, h_n, h \in [-2, -1]$ is the height value of each laser spot falling within the grid, and n is the number.

The value of the third channel corresponds to the standard deviation of the height value of the points falling within the grid and its eight neighborhoods. and is calculated as follows:

$$T(x, y) = 255 \times \frac{2}{\pi} \times \arctan \sqrt{\frac{\sum_1^n (h - \sum_1^n \frac{h}{n})^2}{n}} \tag{3}$$

At first, we set the statistical range of the standard deviation to a single pixel as with the previous two channels. However, during the experiment, we found that a single pixel in LBEV often only corresponds to one or two LIDAR points, which makes the standard deviation meaningless. So we extend the statistical range to each pixel and its own eight neighborhoods. In formula (3), we use $arctan$ as the normalization function.

The above steps provide the input data for Branch L.

B. The Proposed Network

In this article, we propose an encoder-decoder network model which can learn from the visual image and LIDAR points cloud features, and add the LSTM structure to the network to assist the semantic segmentation of the lane marking through timing information. The Fusionlane network basic structure is shown in Fig. 4. First, the result of semantic segmentation of CBEV (C-Region) is put into an input branch of the network. After convolution, the output feature map of the convolution layer with *convolution stride* $= 2$ is fused with the LBEV. Then, the feature map obtained after multiple convolutions is input to the LSTM module as timing information and transmitted to the next moment. Finally, a decoder module is used to restore the feature map output from the LSTM module to the same size as the original image by two times bilinear upsampling. Low level features from the encoder are fused during the upsampling process, which enables the decoder to better recover the details of the image.

Given the excellent performance[1] of DeepLabV3+ network on the PASCAL VOC 2012 semantic segmentation benchmark[24] and the good performance of the Xception network, we chose a modified Xception network as the backbone network for the proposed network. Here, we denote $outputstride$ as the ratio of input image spatial resolution to the final output resolution.

1) Encoder Module: As shown in Fig. 5, two branches are used in the encoder module to perform convolution operations on the LBEV and C-Region. In the convolving process of C-Region, the output feature map is transmitted to the corresponding position of Branch L when its size has been compressed to half of the input. At the beginning, it wasn't clear to us how large the feature map should be for the fusion operation. But in the end we decided to give the choice to the network itself. Consequently, there is a fusion operation each time the feature map is compressed, and the network can then learn the best fusion strategy from the data. In this way, the network can learn the classification information from the Branch C.

From the Fig. 5, it can be seen that the ratio of the number of the feature maps channels from two branches is 1: 3 in each fusion. This is to match the ratio between the original input C-Region and LBEV. In the Xception network, the ordinary convolutional layers are replaced by the depthwise separable convolutional layers, which greatly reduces the computational complexity of the network and improves performance to some

[1] http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=6

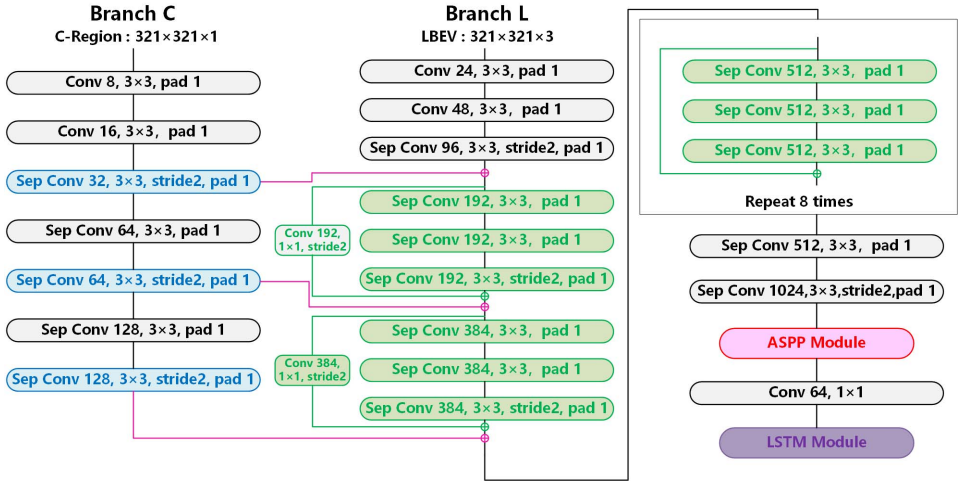

Fig. 5. The specific structure of the encoder module, the convolutional layers of different colors in the figure correspond to the different structures in Fig. 2. Limited by the size of the GPU memory, the images will be randomly cut to a size of 321 × 321 during the training.

extent [22]. Having passed through this module, the size of the output feature map is 21 × 21 × 1024.

In the proposed method, we made two major modifications: first, replace the maxpooling by a convolutional layer with a step size equal to two, second, perform batch normalization operations after each convolutional layer.

2) ASPP Module: The full name of ASPP is Atrous Spatial Pyramid Pooling[17]. By paralleling multiple atrous convolutional layers with different atrous rates, ASPP module can help the network effectively captures multi-scale information. Just the same with DeepLabv3+, ASPP module in our network consists of one 1 × 1 convolution and three 3 × 3 convolutions with $atrous\ rates = (6, 12, 18)$, and the image-level features. Each of them contains 256 channels, and after the layers are connected in series, they are compressed to a thickness of 64 channels using a 1 × 1 convolutional layer, and the feather map is then entered into the LSTM module.

3) LSTM Module: In a real driving scenario, the acquired data of the sensor is continuous in time. Consequently, the data can be input into the recurrent neural network (RNN) to help the network perform the classification task better. Specifically, an LSTM module is employed, which generally outperforms the traditional RNN model as it has the ability to forget unimportant information and remember essential features. This module can also reduce the negative impact on the network of errors in the C-Region. However, traditional full-connection LSTM is not only time and computationally expensive, but it also cannot describe local features in the image, so a three-layer convolutional LSTM (ConvLSTM) [23] is applied in the proposed network, as shown in Fig. 6.

The calculation process in a ConvLSTM cell can be formulated as:

$$\begin{aligned} i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\ o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_{t-1} + b_o) \\ \mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\ \mathcal{H}_t &= o_t \circ \mathcal{C}_t \end{aligned} \tag{4}$$

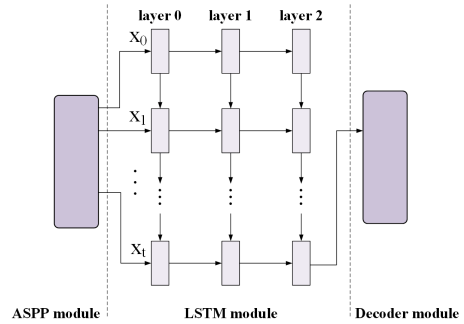

Fig. 6. The LSTM module, we also tuning the number of layers that should be included in the LSTM module, which was finally determined to be a one-layer structure.

In ConvLSTM, the full-connection between each gate is replaced by a convolution operation. In the above formulas, '*' and '∘' denote the convolution operation and the Hadamard product, respectively. $\mathcal{C}_t$, i_t, f_t and o_t represent the cell, input, forget and output gates. $\mathcal{C}_t$, $\mathcal{H}_t$, $\mathcal{C}_{t-1}$ and $\mathcal{H}_{t-1}$ represent the memory and output activations at time t and $t-1$, respectively. W_{xi} is the weight matrix of the input $\mathcal{X}_t$ to the input gate, b_i is the bias of the input gate. The meaning of other W and b can be inferred from the above rule. σ represents the sigmoid operation and $tanh$ represents the hyperbolic tangent non-linearities.

4) Decoder Module: In the decoder module, the feature image output by the LSTM module is restored to the same size as the original image after two times of bilinear upsampling. First, we only fuse the low-level feature from the encoder module during the first upsampling process. However, the results showed that the network cannot accurately recover the details of the original image. So we directly merge the original input image into the second upsampling process after a 1 × 1 convolution operation, which greatly improved detail recovery of the image, as shown in Fig. 7. It should be noted that in our decoder module, all low-level features come from Branch L.

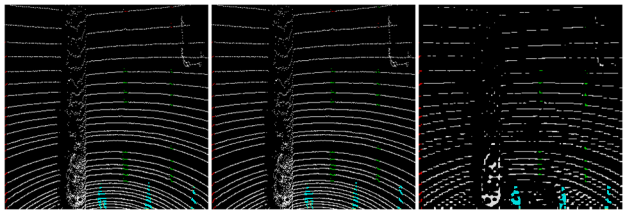

Fig. 7. Comparison of different decoder structures. Left: The ground-truth. Middle: The decoder with two times of low-level feature fusion. Right: The decoder with only one low-level feature fusion.

5) Training Strategy: In the proposed method, we first used Momentum[25] optimizer; however, it was found to be unsuitable because of a lack of reliable initialization parameters. Moreover, Momentum showed insufficient convergence performance when used on our dataset during training. Consequently, it was decided to use the ADAM [26] optimizer instead. In the tuning process, we mainly consider the following aspects:

- In the network structure, we tested the encoder module with different numbers of ResBlocks and the LSTM module with different numbers of ConvLSTM layers.
- The time step was tuned in the LSTM module and the batch size of the input data and the learning rate and its decay during the training.
- To overcome the imbalance among the classes, a weighted cross-entropy was chosen as the loss function and for tuning the weights of the different samples.

IV. Experiment and Results

First, the datasets utilized in this work are introduced and the experiments used to verify the validity and accuracy of the proposed method described. Then the experiments are conducted to verify the validity and accuracy of the proposed method. Then, the performance of the proposed method on the datasets is compared with state-of-the-art methods in semantic segmentation..

A. Datasets

Datasets were constructed based on the KITTI dataset, because KITTI contains synchronous and continuous images and point cloud data of the road. 436 LBEV images and the corresponding CBEV images were labelled manually.

It should be noted that we take the 81th to 148th images as the testing set. The images were rotated 20 times both clockwise and counterclockwise, one degree each time, to give datasets with 14720 labeled LBEV and CBEV images each. the rotated images were used as the training set, which had $362 \times 40 = 14480$ images, and the original 362 images were used as the validation set. During training, we will test the model on the validation set when each epoch is finished. With Tensorboard-a tool provided by Tensorflow [27]-we can track the performance of the model on the training and validation sets in real time, so as to adjust the training strategy in time.

In our datasets,the CBEV images are divided into six different parts: Background, Solid Line, Dotted Line, Stop Line, Arrow and Prohibited Area. The LBEV images have an additional category called Other Point, as shown in Fig. 8. The labeled CBEV images were used to train the DeepLabV3+ based on the cityscapes pretrained model[2].

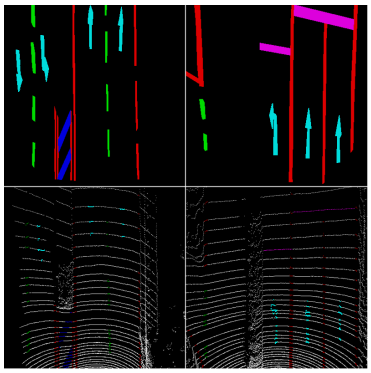

Fig. 8. Example of labeled images. The upper row shows CBEV images and the lower row LBEV images. The black area is the background, the red area is the solid line, the green area is the dotted line, the purple area is the stop line, the steel blue area is the arrows and the blue area is the prohibited area. The LBEV images include an additional white area representing the category Other Point.

TABLE I
MIoUs of of the Two Optimizers on the Testing Set

Optimizer	Momentum	ADAM
MIoU(%)	43.58	67.43

The labeled LBEV images and the C-Region predicted by DeepLabV3+ were then used to train our own network.

B. Experimental Platform

The experiments were implemented on a computer with an Intel Core i7-8700@3.2GHz, 32GB RAM and one NVIDIA TITAN-X (Pscal) GPU.

C. Transfer Learning for DeepLabv3+

As mentioned above, we trained DeepLabV3+ on our CBEV training set based on a pretrained model. DeepLabV3+ is developed by Google, it is one of the most advanced model of image semantic segmentation and achieved great success on many benchmarks. During the training process,it was found that the Momentum optimizer in DeepLabV3+ could not converge well on the training set, so a decision was taken to use the ADAM optimizer instead. Fig. 9 and Table I shows the final losses on our CBEV training set and the Mean Intersection over Union (MIoU) on the testing set of these two different models, respectively. It can be seen that the DeepLabV3+ network with an Adam optimizer can achieve better semantic segmentation results for the CBEV. So, the required C-region was taken from the prediction results of the improved DeepLabV3+ network.

[2]http://download.tensorflow.org/models/deeplabv3_cityscapes_train_2018_02_06.tar.gz

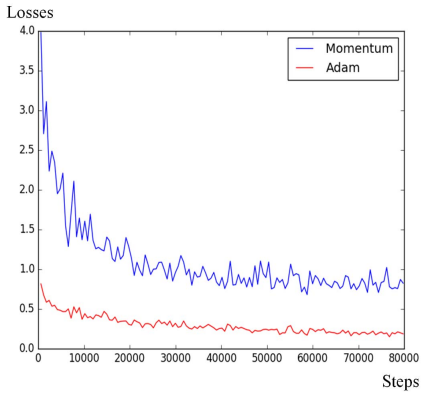

Fig. 9. Losses of the Momentum and ADAM optimizers during training.

D. Semantic Segmentation Performance on LBEV

The method results are compared in two ways. First, we compare the segmentation results of each model intuitively and visually, and then we quantitatively analyze and compare each model in turn. In these experiments, our proposed model is mainly compared with the DeepLabV3+ model which achieved excellent results on the PASCAL VOC 2012 semantic segmentation benchmark.

Specifically, in the experiment we compare the following methods:

- DeepLabV3+: Developed by Google, DeepLabV3+ is one of the most advanced model of image semantic segmentation. LBEV and CBEV semantic segmentation experiments are performed with DeepLabV3+.
- Modified DeepLabV3+: Two modifications are made to DeepLabV3+. First, we apply a 1×1 convolution on the original input LBEV, which is then concatenated with the output of the second upsampling. Second, we replace the Momentum optimizer with the Adam optimizer.
- FusionLane_Without_LSTM: This model does not contain LSTM module but but is otherwise as the model described in Section III.
- FusionLane_FcLSTM: This model deploys a traditional full- connection LSTM after the ASPP module.
- FusionLane: This is the model proposed in this article, namely, the FusionLane model with a ConvLSTM structure.

1) Visually Intuitive Evaluation: The segmentation results on the testing set obtained by the above methods after training are shown in Fig. 10.

For the visually intuitive comparison, we selected a set of segmentation results that contain seven consecutive scenes. The first four rows (from top to bottom in Fig. 10) are the CBEV, C-Region, LBEV and ground-truth of LBEV, respectively. The fifth row is the prediction results of the DeepLabV3+, where it can be seen that the decoder has difficulties restoring the LBEV perfectly, because the original input image is not merged during the upsampling process and many details are lost, resulting in a very low MIoU.

In the sixth line, it can be seen that the Modified DeepLabV3+ can recover the details of the original image quite well, although not perfectly. However, due to the lack of classification information from the C-Region, the network can only rely on the features in the LBEV and is likely to produce incorrect classification results. In this row, for example, the model struggles with predictions of solid and dotted lines in the left half of the images because there is a quite large distance between the two laser lines and it is in an intersection scene where the solid line and the dotted line are very close. At the same time, the model's prediction of the stop line is also poor.

The seventh line is the segmentation result of FusionLane_Without_LSTM. This method can be considered to combine the classification information of the C-Region on the basis of the Modified DeepLabV3+. It performs well in most cases, but mistakes occur when there are serious classification error or blind spot in the C-Region. There happens to be no serious errors in C-Region from the enumerated scene in Fig. 10, but it happens quite often. Some errors in the predictions of the dotted line and stop line are also found.

It was hoped to overcome the problems faced by FusionLane without LSTM by using timing information from the previous and following frames. A traditional full-connection LSTM was employed in the model to form the FusionLane with FcLSTM model. However, as can be seen from the segmentation results in the eighth row, the FcLSTM structure has a negative impact. The model produces many serious errors, such as confusing arrows with dotted lines. The errors are mainly because the feature map is transformed into a one-dimensional tensor when input to the FcLSTM module, which destroys existing local features.

Based on the above results, model improvements were made in three area:

- First, as it is difficult for the network to obtain excellent semantic segmentation results based on the information provided by LBEV alone, the C- Region was introduced into the model.
- Second, error messages and blind spots in the C-Region can have a negative impact on the semantic segmentation results. This problem can be addressed through the inclusion of timing information.
- Third, the FcLSTM module destroys local features in the feature maps, resulting in new errors. Thus, we replaced the FCLSTM module with a ConvLSTM module.

The FusionLane model completed the task very well, as can be seen in the last row of Fig. 10, and it achieved a nearly perfect semantic segmentation result.

2) Quantitative Evaluation: In the quantitative evaluation, we compare the Intersection over Union on each category (IoU), the MIoU and the Pixel Accuracy of different models on the testing set (see Table II).

Pixel Accuracy and MIOU are common evaluation indicators for semantic segmentation and can be calculated:

$$PixelAccuracy = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{5}$$

$$MIOU = \frac{1}{n_c} \sum_i \frac{n_{ii}}{(t_i + \sum_j n_{ji} - n_{ii})} \tag{6}$$

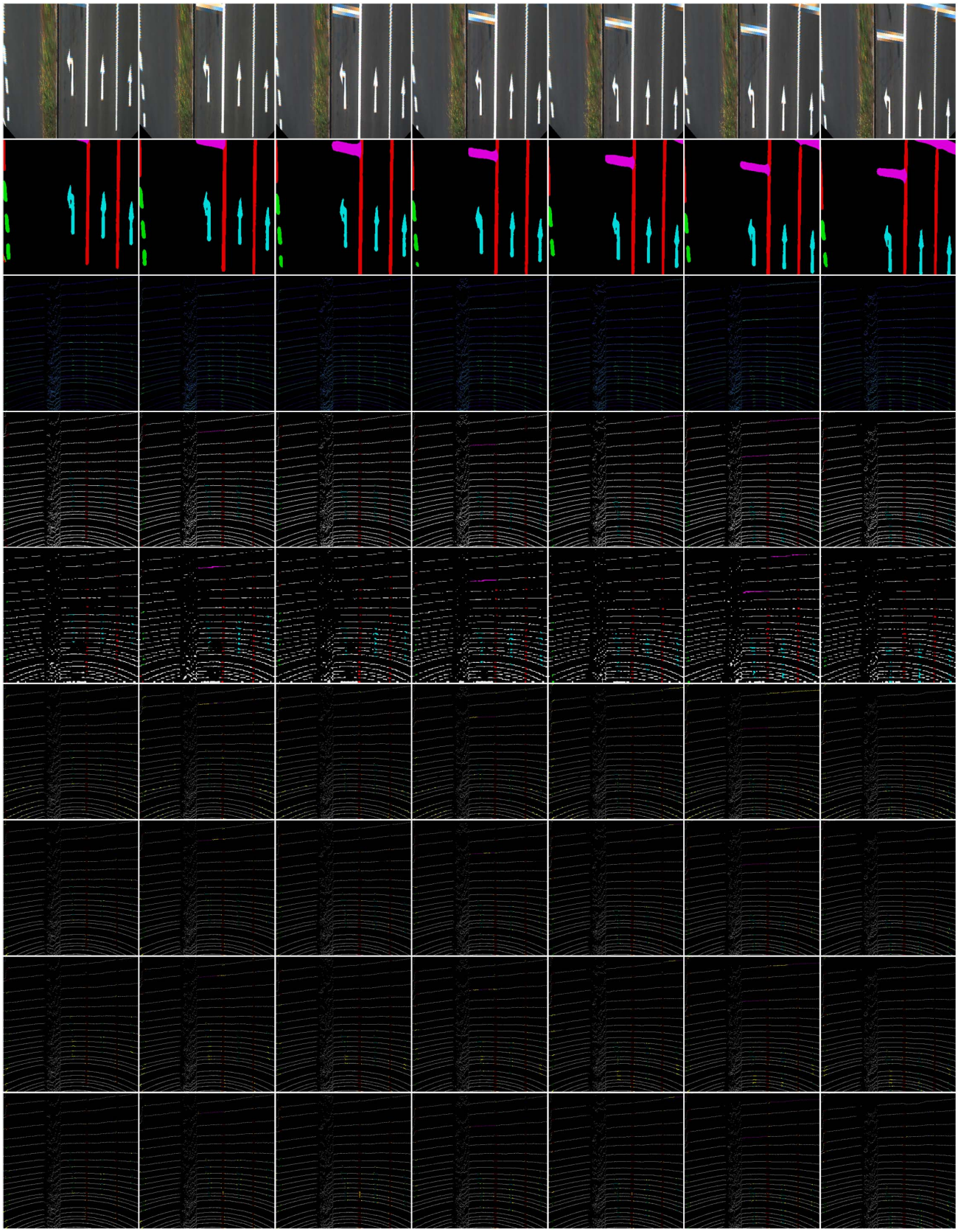

Fig. 10. Raw data and segmentation results of different models in seven consecutive scenarios. First row, the CBEV. Second row, the C-Region obtained from DeepLabV3+. Third row, the LBEV. Fourth row, the ground-truth. Fifth row, DeepLabV3+. Sixth row, the Modified DeepLabV3+. Seventh row, the FusionLane_Without_LSTM. Eighth row, the FusionLane_FcLSTM. Ninth row, the FusionLane. (For the last four columns of images, we reduce the brightness of the correctly classified pixels and highlight the incorrectly classified pixels as the yellow color. The original images are available at https://drive.google.com/open?id=1iKv0c2A6UXud_HzWUPXIPUsqTQhvdcey *here*).

TABLE II
IOU ON EACH CATEGORY, MIOU AND THE PIXEL ACCURACY OF DIFFERENT MODELS

Methods	Background	Solid Line	Dotted Line	Arrow	Prohibited Area	Stop Line	Other Point	MIOU	Pixel Accuracy(%)
DeepLabv3+ (LBEV)	0.9419	0.2587	0.2648	0.2793	0.1915	0.3586	0.2770	0.3674	91.31
DeepLabv3+ (CBEV)	0.9106	0.6287	0.7012	0.5821	0.6935	0.5294	-	0.6743	85.76
Modified DeepLabv3+	0.9989	0.6480	0.6230	0.7106	0.5788	0.3024	0.9654	0.6896	99.59
FusionLane_Without_LSTM	**1.0000**	0.7285	0.7752	**0.7653**	0.7426	0.6574	0.9864	0.8079	99.87
FusionLane_FcLSTM	0.9999	0.7004	0.6192	0.5491	0.6830	0.5629	0.9838	0.7283	99.81
FusionLane	**1.0000**	**0.7477**	**0.7838**	0.7526	**0.7979**	**0.9053**	**0.9867**	**0.8535**	**99.92**

where n_c is the number of classes included in ground truth segmentation, n_{ij} denotes the number of pixels of class i predicted to belong to class j and t_i is the total number of pixels of class i in ground truth segmentation.

From Table II, it can be seen that DeepLabV3+ is unsuitable for LBEV semantic segmentation task. The IoU of all classes except Background are very low. However, this is not because of inadequate training. when modifications were made to the structure, the Modified DeepLabV3+ make a breakthrough under the same training strategy. DeepLabV3+ achieved a much better result in the CBEV semantic segmentation task than the LBEV task. The MIoU is 67.43%, which is considerably better than on the LBEV task, but it is still inadequate. It can be concluded from the data that single-sensor based approaches do not perform well on the studied task.

As this article focuses on the LBEV semantic segmentation task, we replaced the original DeepLabV3+ with the Modified DeepLabV3+ in the comparison.

After merging the classification information of the C-Region, the performance of FusionLane_Without_LSTM is much better than the Modified DeepLabV3+ even if the network is smaller, the MIoU increased almost 12% compared to Modified DeepLabV3+, and FusionLane_Without_LSTM achieved the best IoU score for the Arrow class.

Compared to FusionLane_Without_LSTM, the performance of FusionLane_FcLSTM shows an overall decline, which is a consequence of destroying local features.

From the last row in the table, it can be seen that the FusionLane model achieved the best results for all indicators except the IoU for Arrow, the MIoU is 16.39% higher than that of the Modified DeepLabV3+ and also increased by 4.56% compared with FusionLane_Without_LSTM.

The above data shows that relying on a single kind of sensor, whether camera or LIDAR, cannot give sufficiently accurate semantic segmentation results. Effective fusion of data from different sensors can be considered a viable approach to solving the problem.

An interesting phenomenon can be seen in the data in Table II. In the first row, the IoUs are very low in almost all categories except for the Background class, but the Pixel Accuracy is relatively high at 91.31%. The high Pixel Accuracy but low IoUs can be explained by the small n_{ii} of all the other classes, in addition to the Background class. Consequently, for these classes with a small n_{ii}, a few misclassifications will cause the IoU to drop dramatically. Furthermore, it becomes difficult to improve the IoU when a certain level has been reached. A further consequence for the performance of the models in this class is that the Background class, which occupies most of the LBEV, largely determines the value of Pixel Accuracy, which may explains why DeepLabV3+ (LBEV) has lower MIoU and higher in Pixel Accuracy than DeepLabV3+ (CBEV).

TABLE III
MIOU WITH DIFFERENT TIME STEP VALUES

Time Step	2	3	4	5	6
MIOU(%)	81.73	83.52	**85.35**	81.86	82.14

3) Key Parameter Analysis: The *time step*, which determines how many frames of historical data the network can use to help it predict the current frame, is one of the hyper-parameters having greatest influence on the performance of the network.

When the *time step* is large, the network can review more historical frames, which means there are more historical information. However, this does not mean that the larger the *time step*, the better the prediction of the network, because when the *time step* is excessively large, some data in the history frame is likely to be significantly different from the current data, which will have a negative effect on the network prediction results. To evaluate the effect of *time step* on performance, we therefore conducted a comparative experiment on the network performance with different *time step* values, shown in Table III.

As can be seen from Table III, the MIoU first increased with the increase in *time step*, indicating that historical information has a positive impact on the final result. However, MIoU peaked at *time step* is 4, and further increase in the *time step* decreased MIoU consistent with the previous analysis.

V. CONCLUSION

In this article, we propose a semantic segmentation network for the LIDAR points cloud bird's eye views (LBEV) for the first time. The accuracy of LBEV permits the semantic segmentation results to be directly used to construct a high-precision map. For the task of semantic segmentation of lane marking, the proposed method does not simply turn semantic segmentation into a binary classification task but further subdivides it into a multi- classification task.

As the network structure, we propose a network with a dual-input branch structure to fuse LEBV and C-Region. To address possible misclassification in the C-Region and

its negative impact on network prediction results, an LSTM structure was added to the network. Experiments showed that the proposed method can effectively fuse information from LIDAR and camera images, and the approach achieved excellent results on the LBEV semantic segmentation task.

Future work should investigate transforming the Branch C input from the C-Region to the CBEV to build an end-to-end semantic segmentation network and focus on getting more training data. Such information will enable the construction of high-precision maps.

REFERENCES

[1] S. M. Azimi *et al.*, "Aerial LaneNet: lane marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks," 2018, *arXiv:1803.06904*. [Online]. Available: https://arxiv.org/abs/1803.06904

[2] L. C. Chen *et al.*, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 801–818.

[3] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[5] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, Jan. 2009.

[6] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[10] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[12] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," 2014, *arXiv:1411.4734*. [Online]. Available: http://arxiv.org/abs/1411.4734

[13] G. Lin, C. Shen, A. van dan Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," 2015, *arXiv:1504.01013*. [Online]. Available: http://arxiv.org/abs/1504.01013

[14] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[15] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Oct. 2005, pp. 1458–1465.

[16] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 2169–2178.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[18] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[19] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2018–2025.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[21] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," 2019, *arXiv:1903.02193*. [Online]. Available: http://arxiv.org/abs/1903.02193

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1251–1258.

[23] S. H. I. Xingjian *et al.*, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[24] M. label ref Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserma, "The PASCAL visual object classes challenge a retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2014.

[25] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[27] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: http://arxiv.org/abs/1603.04467

Ruochen Yin received the M.E. degree from the Kunming University of Science and Technology in 2017. He is currently pursuing the joint Ph.D. degree with the University of Science and Technology of China (USTC) and the Lappeenranta University of Technology (LUT). His research interests include intelligent vehicle, machine vision, and mobile robot navigation and localization.

Yong Cheng received the M.E. degree from the Hefei University of Technology, China, in 2011. He is currently an Associate Professor and the Deputy Director of the Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, China. His research interests include industrial robot control and mechanical engineering.

Huapeng Wu received the bachelor's and master's degrees in machine manufacturing automation from the School of Mechanical Engineering, Huazhong University of Science and Technology (HUST), China, in 1986 and 1993, respectively, and the Dr. Sc. (Tech.) degree from the Lappeenranta University of Technology (LUT), Finland, in 2001. He is currently working as an Adjunct Professor and an Associate Professor at LUT in the field of mechatronics, robotics, and artificial intelligences.

Yuntao Song (Member, IEEE) is currently a Professor and the Deputy Director of the Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, China. He is also a Teacher and the Ph.D. Supervisor with the University of Science and Technology of China, Hefei. Since 2012, he has been the Project Leader for the Design of Chinese Fusion Engineering Testing Reactor (CFETR) Machine. Since 2014, he has been the Responsible Officer for the SC200 Superconductor Proton Therapy System. He has been involved in fusion engineering for many years. He has over 100 scientific publications. His current research interest includes many tokamak research fields.

Biao Yu received the B.Sc., M.Sc., and Ph.D. degrees from the School of Mechanical and Automotive Engineering, Hefei University of Technology, in 2007, 2010, and 2013, respectively. He is currently an Associate Professor with the Institute of Applied Technology and the Hefei Institutes of Physical Science at the Chinese Academy of Sciences, China. His research interests include evolutionary computation, intelligent vehicle, and mobile robot navigation and localization.

Runxin Niu is currently a Professor with the Institute of Applied Technology and the Hefei Institutes of Physical Science at the Chinese Academy of Sciences, China. His research interests include intelligent vehicle and mobile robot navigation and localization.