

VH-HFCN based Parking Slot and Lane Markings Segmentation on Panoramic Surround View

Yan Wu, Tao Yang, Junqiao Zhao*, Linting Guan and Wei Jiang

Abstract—The automatic parking is being massively developed by car manufacturers and providers. Until now, there are two problems with the automatic parking. First, there is no openly-available segmentation labels of parking slot on panoramic surround view (PSV) dataset. Second, how to detect parking slot and road structure robustly. Therefore, in this paper, we build up a public PSV dataset. At the same time, we proposed a highly fused convolutional network (HFCN) based segmentation method for parking slot and lane markings based on the PSV dataset. A surround-view image is made of four calibrated images captured from four fisheye cameras. We collect and label more than 4,200 surround view images for this task, which contain various illuminated scenes of different types of parking slots. A VH-HFCN network is proposed, which adopts an HFCN as the base, with an extra efficient VH-stage for better segmenting various markings. The VH-stage consists of two independent linear convolution paths with vertical and horizontal convolution kernels respectively. This modification enables the network to robustly and precisely extract linear features. We evaluated our model on the PSV dataset and the results showed outstanding performance in ground markings segmentation. Based on the segmented markings, parking slots and lanes are acquired by skeletonization, hough line transform and line arrangement.

I. INTRODUCTION

An autonomous car is a vehicle that is capable of sensing its environment and driving without manual intervention. In the past decade, autonomous driving has entered a period of rapid development, especially in detecting surroundings with varieties of sensors, such as cameras, LiDAR. However, as a branch of autonomous driving, efficient and highly automatic parking has still been a difficult problem. There already exist many advanced parking assistance systems, but most of them are in a semi-manufactured stage or only suitable for a very limited parking lot. The critical problem for automatic parking is the detection of parking slot and auxiliary lane markings. Existing detection methods are mostly based on hand-crafted computer vision processing by extracting line features and adding artificial rules on them. Parking slot and lane markings detected in this way are vulnerable to illumination change and fading, thus cannot be applied robustly and fully automatically.

In recent years, deep learning-based approaches have achieved great success in image recognition, object detection, and image segmentation [1]. The outstanding performance

of deep learning makes it the preferred choice for sensing algorithm in autonomous driving. Detecting parking slot and lane markings with deep network models have become a new direction. However, the lack of openly-available parking slot labels based on PSV is the biggest obstacle.

In this paper, we build up a PSV dataset and propose a deep convolutional network to segment parking slot and lane markings. With the segmented ground markings, followed by post-processing of skeletonization, hough line transform, and line arrangement, we can robustly extract read-to-use parking slots and lanes for autonomous parking. The main contributions of this paper are listed as follow:

- We collected more than 4,200 images of PSV, which contained scenes of various illumination conditions. We labeled these images so that we could train a deep neural network for parking slot and lane markings segmentation.
- We proposed an efficient and precise VH-HFCN model to segment the parking slot and lane markings around the vehicle on our PSV dataset.

II. RELATED WORK

Automatic parking is an emerging industry in recent years. In 2008, Yu-Chih Liu et al. staged a PSV to obtain the complete perspective around the car [2]. After then, Mengmeng Yu et al. used surround cameras to obtain a 360-degree surround view in the parking guide [3]. Recently, Chunxiang Wang [4], Jae Kyu Suhr [5], Linshen LI [6] used a PSV to realize the parking slot detection in automatic parking. Apart from the parking slots, autonomous parking also requires robust LKA during planning, which can hardly be satisfied based on mid-to-far range front view cameras. Traditional lane detection methods are mostly based on the line detection, line fitting for lane markings and multi-frame fusion methods [7]. Similarly, robust and generic lane detection methods are also required, especially in the surrounding of the vehicle. In 2014, Jihun Kim used a convolutional neural network and random sampling method to acquire robust lanes [8]. Subsequently, Baidu IDL proposed a deep convolutional network model for lane detection [9]. KAIST proposed VPGNet for detecting lanes and road signs with their labeled dataset [10]. However, these lane markings detection methods were all based on the images captured by the front monocular camera, which are only useful in the urban road. In 2016, Ford proposed to detect lane markings on both sides of the vehicle, but this approach has a limited camera perspective [11].

This work is supported by the National Natural Science Foundation of China (No. U1764261), the Natural Science Foundation of Shanghai (No.kz170020173571) and the Fundamental Research Funds for the Central Universities (No. 22120170232, No. 22120180095).

All authors are with the Department of Computer Science and Technology, School of Electronics and Information Engineering, Tongji University, Shanghai, 201804, China zhaojunqiao@tongji.edu.cn

Lane markings and parking slots present linear structures. Thus we propose to use semantic segmentation model to segment target parking slot and lane markings. Long, et al. proposed Fully Convolutional Network (FCN), in 2015. [12]. After then, deep networks based on this end-to-end learning methods have become the defacto of image semantic segmentation. Stacked Deconvolutional Network [13] and SegNet [14] with a Encoder-Decoder Architecture performed well on public segmentation datasets. Other improved methods include: using a more efficient benchmarking classification models, such as ResNet [15], DenseNet [16]; optimizing network-specific modules like dilated convolution [17], zoom-out features [18]; combining the traditional segmentation model CRF [19]. At CVPR 2017, Zhao et al. proposed a PSPNet model, achieving state-of-the-art performance on most public datasets [20]. Behrendt introduced a new lane markings segmentation network from automatically generated labels [21].

In prior work, based on FCN, we put forward efficient segmentation networks FCCN [22] and HFCN [23]. We proposed the cost function method to train our network. Our approach was inspired initially by lane markings segmentation task, and we successfully evaluated it on public benchmarks. In this paper, we made further improvements to implement a prominent network for parking slot and lane markings segmentation.

III. DATASET

A. Image Collection

Current visual parking lot detection is mostly based on PSV, which has two main benefits: 1) achieve a complete view around the car without blind spot; 2) with a perspective of looking down, the parking slot and lane markings no longer distort and metric-based measurement becomes possible. However, PSV datasets which can be used for deep learning are rare. Therefore, this paper builds up a PSV dataset using TiEV (Tongji Intelligent Electronic Vehicle) based on roads in Jiading campus of Tongji University¹.

We collected a total of 4249 PSV images. As to the distribution of training, validation, and test, we refer to other common datasets, by the ratio of 6: 1: 3. In detail, the numbers of images for training, validation, and test are 2550, 425, and 1274. Respectively, each part contains images captured in different scenes.

The original image of the surround view is captured by four cameras mounted in the front, rear, left, and right of the car body, each holds a resolution of 640x480. All the cameras have a wide viewing angle of 180 degrees. Because of heavy distortion, original images captured from the camera should be calibrated and undistorted. A car-centered coordinate frame will then be determined to warp the undistorted images onto the ground. Fig. 1 shows the results of undistortion and warp perspective of the original image.

After undistortion and warp perspective operation, we get four top-down view images that can be used for stitching.

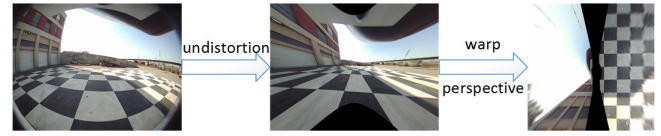


Fig. 1. Left image is the original image captured from the fisheye camera, the middle image is undistorted with internal parameters, the right image is the result of warp perspective transform.

select four masks appropriately for each of the corresponding top-down views. Then, we stitch four top-down views with these masks to generate a PSV image. Fig. 2 shows the four original images and the final PSV image, which holds a resolution of 1000 by 1000. One pixel indicates 1 cm by 1 cm in car frame, so an area of 10 meters by 10 meters is covered.

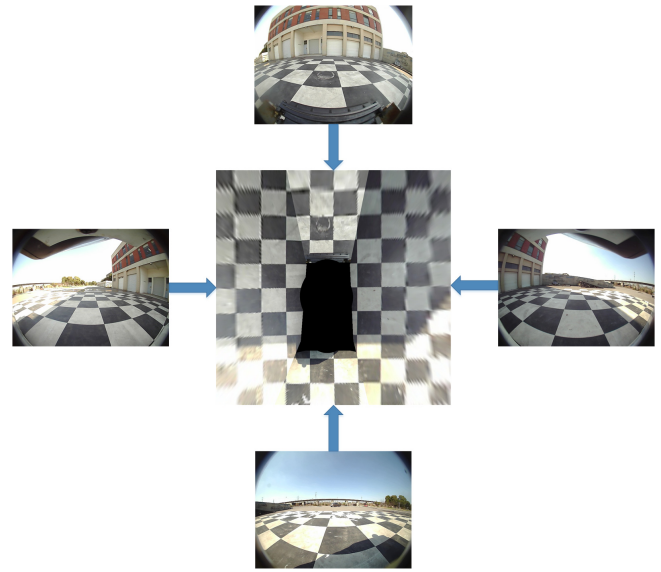


Fig. 2. A PSV image is generated by four original fisheye image.

We collected the PSV dataset containing a variety of surrounding environment, including indoor, outdoor, strong lighting, shadows and so on. Both fuzzy and clear parking slots and lane markings were collected. For parking slots, we collected horizontal, vertical and diagonal shaped slots, also including different colors, like yellow and white. Fig. 3 shows examples of different scenes and types. Since lines hold only a small area in PSV images, the ratio of line segments targeted for identification is low. We calculate a mean ratio 2.49% of all images.

B. Dataset Labeling

There are five types of segmentation classes in our PSV dataset, which are parking slot, solid white lane, white dashed lane, yellow solid lane and yellow dashed lane. When labeling the images, we connect all corner points of the parking slot or lane markings to generate a closed area as the segmentation target. However, the labeling of the dashed lane is special. We choose to label the dashed lane as a whole

¹<http://cs1.tongji.edu.cn/tiev/resource/>

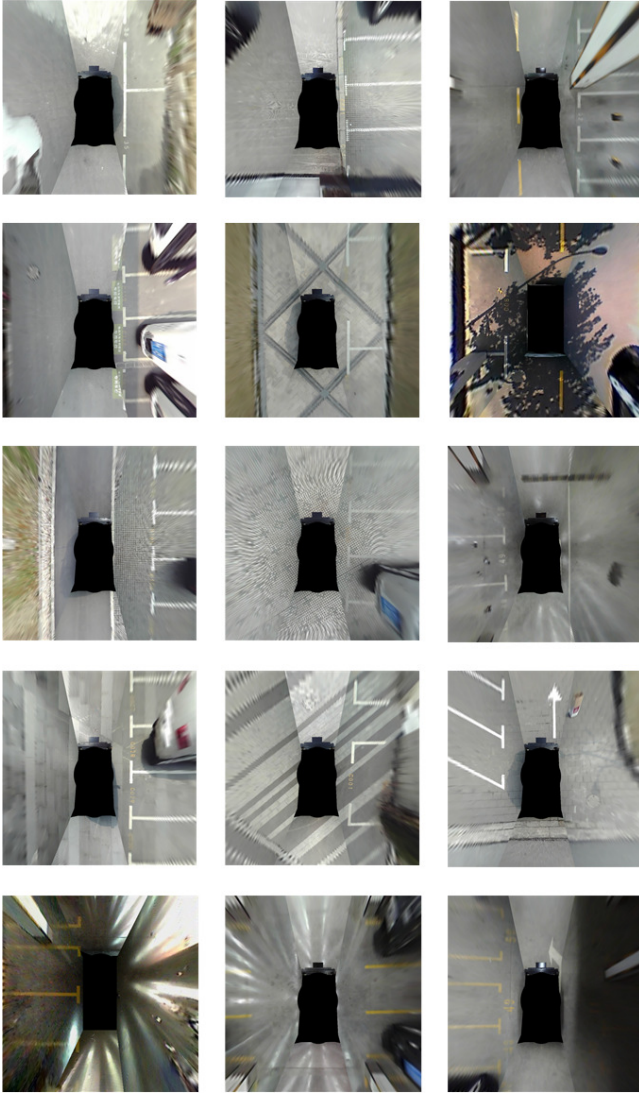


Fig. 3. Some sample images of our PSV dataset in different conditions: 1) first row contains outdoor and indoor scenes; 2) second row are under strong light and shadow; 3) third row shows both clear and fuzzy parking slots; 4) fourth row lists parking slots in horizontal, vertical and diagonal types; 5) fifth row are samples of yellow parking slots.

segmentation area, that is, the blank areas in the middle of the dashed lane are also labeled as a dashed lane area. As shown in Fig. 4, the final labeled lane is an integral area, without appearing segmented. The reason is to preserve the overall information of the lane, rather than segmenting the lane into disjointing blocks.

IV. METHOD

A. Network

We use a deep convolutional network to segment ground markings on our PSV dataset. The proposed VH-HFCN model is illustrated in Fig. 5. Our network is based on HFCN [23], which improves the up-sampling process of the FCN model, divide the up-sampling operation on the feature maps into five steps. After each step, the feature

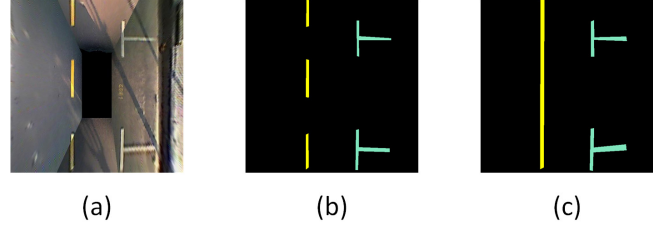


Fig. 4. The dashed line in (a) is labeled as an integral area like (c), rather than several segmented areas like (b).

maps expand to double size in both width and height, as a reverse operation of the previous pooling layer. We then make a combination of each pair of up-sampled and down-sampled feature maps, so as to realize the fusion and reuse of low-level feature information. In addition to this up-sampling path, we add an extra one-step up-sampling operation on each up-sampling layer and acquire a pre-output after each one-step up-sampling. As there exist five up-sampling layers, we shall acquire five pre-output. Finally, we concatenate these pre-outputs, followed by a convolution layer to generate the final output.

In most case, lane markings appear as vertical lines, or with a small angle of deflection, while parking slot is pairs of perpendicular lines. Based on this fact, we add an extra VH-stage before the up-sampling operation to further learn these linear features. This extra stage hardly destroys feature information learned in low layers. Thus other structures like slanting lines can also be well segmented. The VH-stage consists of two independent special convolution paths; each contains five convolution layers and Relu layers. The first path uses a series of vertical convolution kernels with a size of 9×1 . These special convolution kernels are of great help for extracting vertical linear features. Similarly, horizontal convolution kernels with a size of 1×9 are helpful for extracting horizontal linear features. We then sum these two paths and set the result as input to the subsequent up-sampling process. By adding this extra VH-stage, we improved the segmentation performance, assessed by metric of mean Intersection-of-Union(mIoU), of the network by more than five percentage points. We will describe this enhancement in more detail in next section.

B. Cost Function

In the offline perception competition of the future challenge in China, we adopt the segmentation network FCN to detect lanes, since traditional methods have difficulty in classifying all lane types. However, the network's learning of the target will be weakened due to the small proportion of pixels for the parking slots and lane markings in the image. Therefore, when calculating the cost function, we increase the weights of the target area. What is more, the weights given to the target are not fixed. We dynamically calculate the proportion of each target category in the current image and take the inverse of the proportion as the cost weight. This cost function allows the network to focus more on the

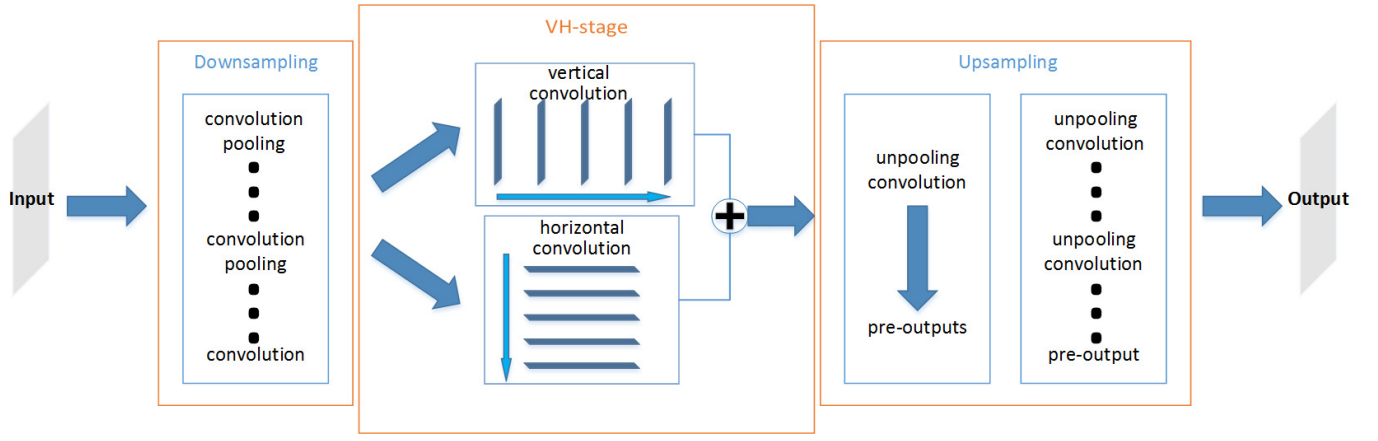


Fig. 5. Our segmentation network consists of three parts: the first down-sampling part for feature extraction; the second VH-stage part for further linear feature extraction, including independent horizontal and vertical convolution path; the last up-sampling part for feature fusion and resolution recover, containing two separate up-sampling way.

learning of the target area. In addition, we add cost functions in five pre-outputs, which is introduced in section IV.A. Thus there are six cost functions when training our segmentation model. The final Loss is calculated according to equation (1), in which $L_{finaloutput}$ is the loss of final output and $L_{preoutputs_i}$ is the loss of i_{th} pre output. λ_i is the weight of each loss, in this paper, λ_i is we achieve the best performance by setting all λ_i to 1.0. With this additional loss, our training can reduce the error erosion during backpropagation.

$$Loss = L_{finaloutput} + \sum \lambda_i * L_{preoutputs_i} \quad (1)$$

Each single loss L is acquired by equation (2), where $H_{i,j}$, $Y_{i,j}$ are predicted class and label in pixel(i, j), $w_{i,j}$ is class weight. Additionally, $w_{i,j}$ is calculated along with the proportion of the target class in the whole image [22].

$$L = \sum_{i=1}^m \sum_{j=1}^n w_{i,j} \|H_{i,j} - Y_{i,j}\|^2 \quad (2)$$

C. Extraction of Parking Slot and Lanes

Though the VH-HFCN model can acquire segmented ground markings, further processing for getting the ready-to-use parking slots and lanes is necessary. In this paper, post-processing after the segmentation consists of three steps: 1) apply morphological skeletonization to extract the central path of each segmented area; 2) hough line transform on the skeleton is added to generate necessary lines; 3) partition and merge similar lines, build up the parking slots and lanes. After the processing, detected parking slots and lanes are used in automatic parking. Though our extraction method is quite robust, it can be improved. We will try continuous model fitting to realize the extraction in our future work.

V. EXPERIMENTS

A. Experiments Configuration

All our experiments are implemented on MatConvNet [24], which is a Matlab toolbox for the deep convolutional

network. We used a batch-size of 10, a learning rate of 0.0001, and an epoch of 50 for training. Multiple cost functions used in the experiments all keep a weight of 1.0. The networks are trained on NVIDIA GeForce GTX TITAN X graphic card.

B. Evaluation

Our proposed model evaluation experiments on the PSV dataset can be divided into three stages: 1) acquire the best size of vertical and horizontal convolution kernels; 2) determine the convolution path and the method of combining feature maps in VH-stage; 3) compare segmentation performance of proposed model with FCN, FCCN, and HFCN on PSV dataset. We use Mean IoU as the metric of our segmentation performance.

It is a crucial issue for us to determine the convolution kernel size in VH-stage, since short kernels are not enough to cover full linear features, while long kernels may reduce the efficiency of the network, even reduce the segmentation performance. Table I shows the segmentation results of different convolution kernel sizes in VH-stage. Among the model column, model v3h3 means horizontal kernels with a size of 1x3 and vertical kernels with a size of 3x1; the other models are similar to this. From Table I, we can know that the longer the convolution kernel, the better the segmentation results. However, if we continue to enlarge the kernel after model v9h9, the growth of the result has become negligible, the major metric mIoU even dropped. Hence, Table I shows that the model v9h9 obtains the best performance. It should be oversized convolution kernel with redundant feature information. Therefore, there is no need for us to continue enlarging convolution kernel.

For the methods of feature maps combination after the VH-stage, we tried both concatenation and summing. Besides, we added a third independent convolution path apart from the vertical and horizontal convolution path. Since there exist some slanting lanes in our PSV dataset, we add the third convolution path to retain feature information extracted

TABLE I
RESULTS OF DIFFERENT CONVOLUTION KERNEL SIZES IN VH-STAGE
(MEAN IOU IS THE PERFORMANCE METRIC).

Model	Mean pix.acc(%)	Pixel acc(%)	Mean IoU(%)
v3h3	89.71	95.30	43.80
v5h5	89.59	95.29	44.75
v7h7	88.25	95.32	45.72
v9h9	89.79	96.25	46.51
v11h11	89.87	95.88	46.48

in lower layers. As is shown in Table II, the third convolution path is included in the first model conv+, while only vertical and horizontal path in model concat and sum. We can know that without an additional convolution path we achieve a higher segmentation performance in mIoU, which shows that the VH-stage can learn useful line features. The additional convolution path may result in redundant features. Hence, the segmentation results become worse.

In addition, we can also learn from Table II that summing the feature maps achieves a better performance rather than concatenating them after VH-stage. Two feature maps can be integrated into one feature map with a summing operation, so the summed map can hold integral features of target ground markings, especially parking slot. In other way, features are still separated by a concatenation, which results in a worse performance in later segmentation.

TABLE II
RESULTS OF DIFFERENT STRATEGIES IN VH-STAGE (MEAN IOU IS THE
PERFORMANCE METRIC).

Model	Mean pix.acc(%)	Pixel acc(%)	Mean IoU(%)
conv+	90.96	96.09	45.27
concat	88.27	96.29	45.99
sum	89.79	96.25	46.51

Apart from our proposed network in this paper, we also evaluate FCN, FCCN, and HFCN on our PSV dataset. Table III shows detail results of these models. Compared with other models, we achieve a considerable increase in the segmentation performance, which indicates that our proposed VH-stage strategy is effective. Fig. 6 shows several visual samples with predictions of different models on PSV dataset. We can learn that our proposed model achieves more precise segmentation result. It's worth noting that we achieve the almost perfect dashed lane segmentation in the fourth row, while other models generate bumps in the blank area. This is due to the integral lane features obtained by VH-stage module. As shown in Fig. 7, post-processing is added to the segmented result to generate the ready-to-use parking slots and lanes.

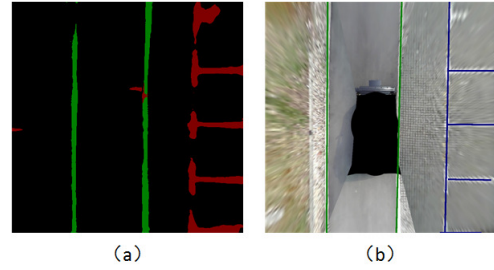


Fig. 7. Parking slots and lanes detection based on segmentation result. (a) segmentation result by VH-HFCN; (b) detected lanes and space parkings.

VI. CONCLUSIONS

In this paper, we introduce a parking slot and lane markings PSV dataset. This dataset comprises surround-view images in varieties of scenes, with multiple types of the parking slot. Based on the labeled PSV dataset, we propose a VH-HFCN segmentation model, which is proved to be efficient in extracting linear feature information from parking slot and lane markings. We use multiple soft cost function to train our model, and achieve a significant performance in the PSV dataset. Compared with other network models, we acquire a significant increase in performance. In addition, we add post-processing to detect the final parking slots and lanes. In the future, we intend to label more PSV images and design a further extended network to improve segmentation performance.

REFERENCES

- [1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [2] Y.-C. Liu, K.-Y. Lin, and Y.-S. Chen, "Birds-eye view vision system for vehicle surrounding monitoring," *Robot vision*, pp. 207–218, 2008.
- [3] M. Yu and G. Ma, "360 surround view system with parking guidance," *SAE International Journal of Commercial Vehicles*, vol. 7, no. 2014-01-0157, pp. 19–24, 2014.
- [4] C. Wang, H. Zhang, M. Yang, X. Wang, L. Ye, and C. Guo, "Automatic parking based on a bird's eye view vision system," *Advances in Mechanical Engineering*, vol. 6, p. 847406, 2014.
- [5] J. K. Suhr and H. G. Jung, "Sensor fusion-based vacant parking slot detection and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 21–36, 2014.
- [6] L. Li, L. Zhang, X. Li, X. Liu, Y. Shen, and L. Xiong, "Vision-based parking-slot detection: A benchmark and a learning-based approach," in *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 2017, pp. 649–654.
- [7] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.
- [8] J. Kim and M. Lee, "Robust lane detection based on convolutional neural network and random sample consensus," in *International Conference on Neural Information Processing*. Springer, 2014, pp. 454–461.
- [9] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue *et al.*, "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1704.01716*, 2015.
- [10] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. S. Hong, S.-H. Han, and I. S. Kweon, "Vpnet: Vanishing point guided network for lane and road marking detection and recognition," *arXiv preprint arXiv:1710.06288*, 2017.
- [11] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "Deepplanes: End-to-end lane position estimation using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 38–45.

TABLE III
QUANTITATIVE RESULTS OF SEGMENTATION PERFORMANCE ON OUR PSV DATASET (%).

Model	background	parking	white-solid	white-dashed	yellow-solid	yellow-dashed	pacc	mIoU
FCN [12]	85.88	13.16	18.42	7.40	23.09	20.32	86.18	28.04
FCCN [22]	92.53	22.50	29.60	11.87	41.21	27.38	92.66	37.51
HFCN [23]	93.87	25.46	36.26	18.97	45.08	26.87	93.97	41.09
Ours	96.22	36.16	39.56	21.46	47.64	38.03	96.25	46.51

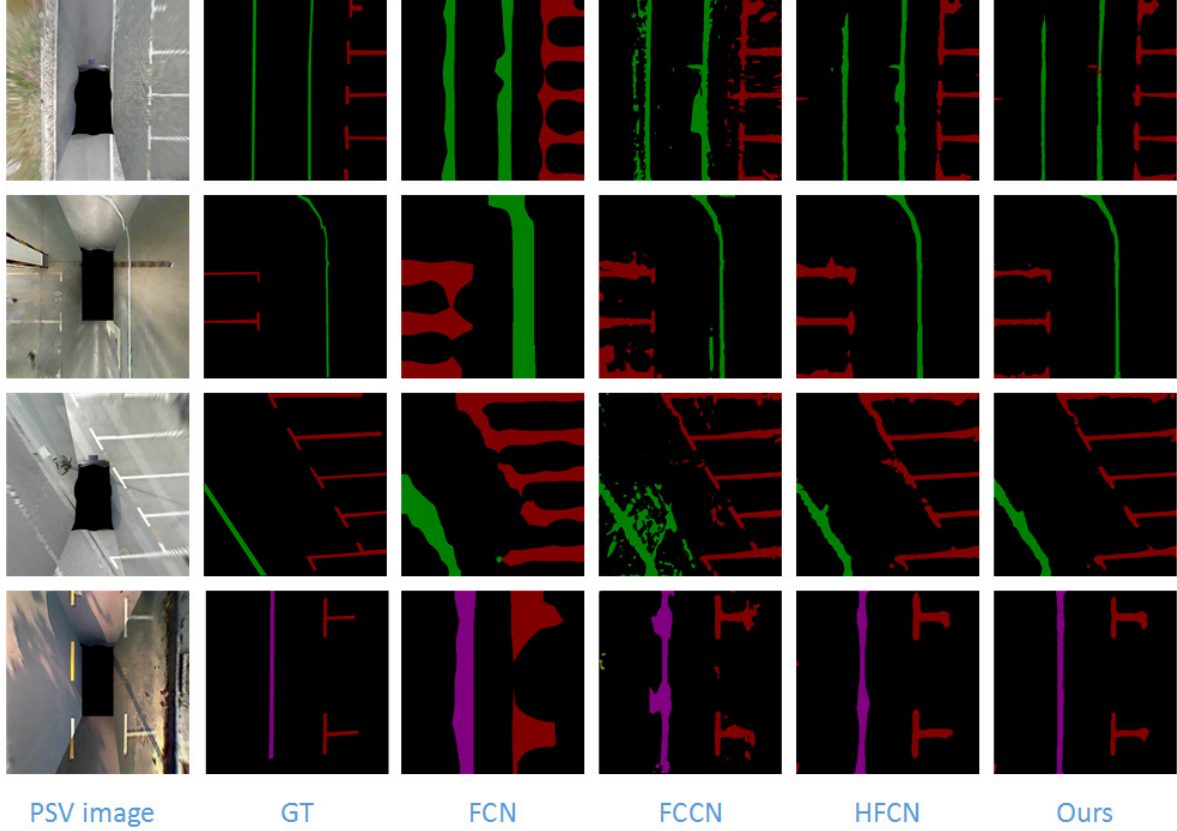


Fig. 6. Some visual results of different models on PSV dataset. We achieve more precise segmentation on parking slot and lane markings.

- [12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [13] J. Fu, J. Liu, Y. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *arXiv preprint arXiv:1708.04943*, 2017.
- [14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.
- [15] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation," *arXiv preprint arXiv:1611.06612*, 2016.
- [16] S. Jégou, M. Drozdal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017 IEEE Conference on. IEEE, 2017, pp. 1175–1183.
- [17] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.
- [18] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3376–3385.
- [19] 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," *arXiv preprint arXiv:1606.00915*, 2016.
- [20] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *arXiv preprint arXiv:1612.01105*, 2016.
- [21] K. Behrendt and J. Witt, "Deep learning lane marker segmentation from automatically generated labels," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 777–782.
- [22] Y. Wu, T. Yang, J. Zhao, L. Guan, and J. Li, "Fully combined convolutional network with soft cost function for traffic scene parsing," in *International Conference on Intelligent Computing*. Springer, 2017, pp. 725–731.
- [23] T. Yang, Y. Wu, J. Zhao, and L. Guan, "Semantic segmentation via highly fused convolutional network with multiple soft cost functions," *arXiv preprint arXiv:1801.01317*, 2018.
- [24] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *ACM International Conference on Multimedia*. ACM, 2015, pp. 689–692.