Efficient and Lightweight Convolutional Neural Network for Lane Mark and Road Segmentation

Guan-Ting Lin and Jiun-In Guo
Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, ROC
E-mail :{ ilovevictor0424, jiguoccu}@gmail.com

***Abstract*—Semantic segmentation is one of an important task in computer vision that takes a great part in the perception needs of intelligent autonomous vehicles. ConvNets excel at this task, as they can be adaptively trained end-to-end to yield a set of robust hierarchies of features. The proposed key method is to reduce the unnecessary weights to build an efficient and lightweight network to acquire high accuracy on lane mark and road segmentation at pixel level. The proposed fully convolutional neural network achieves 360x480@28 fps and 97.6% accuracy on our in-house pixel-based hand-annotated lane mark and road datasets. All our models and results are trained and evaluated on an NVIDIA GTX 1080 GPU device.**

I. INTRODUCTION

One of a complex and challenging tasks for autonomous vehicles application is drivable lane location and road boundaries detection, which includes the localization of the road and the determination of the relative position of the vehicle on the road. Deep convolutional neural networks have led to a series of breakthroughs for computer vision task. Deep networks naturally integrate not only locational but also semantic level features in an end-to-end multilayer fashion. Cascading the extracted features with a trainable classifier, it can form a powerful system for many related tasks.

Semantic image segmentation, the task of assigning a set of predefined class labels to image pixels, is an important tool for modeling the complex relationships of the semantic entities usually found in street scenes. It is an important part of modern autonomous driving systems, as a pixel-wise understanding of the surrounding scene is crucial to navigation and action planning.

In this paper, we propose an efficient and light-weighted architecture that achieves fast and accurate lane mark and road segmentation without the need for additional post-processing steps and without the limitations imposed by pre-trained architectures. This paper makes the following contributions: (i) We have proposed a VGG-like architecture that can perform 360x480@28 fps on modern GPU (NVIDIA GTX1080) devices. (ii) Without any pre- and post-processing, we can achieve robust and efficient results on lane mark and road region segmentation. (iii) We have reached comparable performance in accuracy compared to other VGG-like architecture [1] with 1.86 times faster.

II. RELATED WORKS

ConvNets were initially designed for image classification challenges, which aims to predict single object categories from images. Long et al. [2] (FCN) firstly adapted pre-trained classification networks (e.g. VGG16 [3] or Alexnet [4]) to perform end-to-end full-image semantic segmentation training and inference by making them fully convolutional and upsampling the output feature maps. SegNet [1] proposed symmetrical encoder-decoder architecture and replaced the deconvolution in FCN to reduce the weights of network and get the upsampled features with unpooling layers by using the indices of the encoder's max-pooling blocks.

III. NETWORK ARCHITECTURE FOR SEGMENTATION

The proposed base convolution neural network architecture is inspired by VGG [3] model for robust feature extraction and a symmetrical decoder like [1] to map the feature map into full resolution (please see *Figure 1*). The proposed network has seventeen convolutional layers and five pooling layers. Instead of stacking six layers in the high-resolution stages like that in [3], we simply adopt the two convolutional layers followed individually by a pooling layer for down sampling. In the low-resolution stages, we use two convolutional layers to encode the features and a pooling layer to enlarge the receptive of the network. For the decoding stages, we use un-pooling indices to up-sample the feature maps and convolution to reformulate the output response.

Different from normal batch normalization [5] operations, we remove the re-mapping transformation; simply normalize the feature maps into a zero-centered range. This step can speed up the proposed network while still remain good performance. Every convolutional layer is followed by a normalization layer and a rectified linear unit.

IV. EXPERIMENTS

A. Data

We have made a pixel wise database that contains lane marks and road, as shown in *Figure 2* for annotation example. We have collected total 3163 frames from real driving environment, and split it into training and testing dataset by the ratio of 9:1. All the training and the testing results are based on this dataset.

B. Training and inference

We train and evaluate the proposed model on the input image of the resolution of 360 by 480. All training and inference results are perform on GTX 1080 GPU device.

We train our model from scratch based on the caffe [6] framework. Initializing the weights as that in [8], we use SGD with a mini-batch size of 3 images for the restriction of GPU memory capacity. The learning rate starts from 0.001 and is divided by 10 for every 1k iterations, and the models are trained for up to 40k iterations. We use a weight decay of 0.0005 and a momentum of 0.9. We do not use dropout [7] in the training procedure.

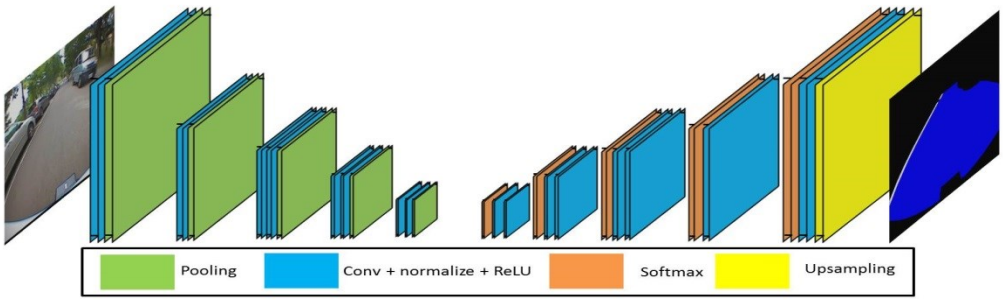

Figure 1: The proposed symmentrical architecture for lane mark and road segmentation, where Input is a RGB image with arbitary size and output is the segmentic image of lane and road.

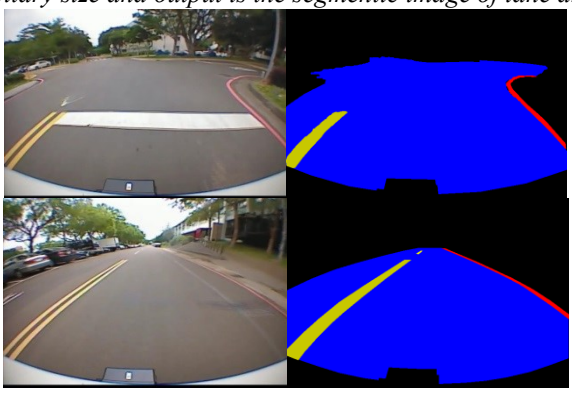

Figure 2: Examples of lane mark and road annotations

C. *Results*

We follow the released training procedure of SegNet [1] as our performance/accuracy baseline. The proposed network performs well in the application of lane mark and road detection. Comparing to the baseline, we have only 50% fewer parameters and 50% fewer multiply and add operations. **Table 1** shows the experimental results. The proposed network can get 97.6% global accuracy, which is 0.2% better than the baseline. We argue the degradation result of baseline coming from the overly deep architecture that makes it hard to learn on our three-class dataset. By reducing the weights in convolutional filters, our model converges very well and performs 1.86 times fast as compared to baseline model.

V. CONCLUSIONS

By reducing the number of weights, we have built a simple and robust convolutional neural network for efficient land mark and road segmentation.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 2481-2459, 2017.

[2] Jonathan Long, Evan Shelhamer, Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[3] K. Simonyan ; A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[4] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2012.

[5] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *ICML*, 2015.

[6] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *arXiv:1408.5093*, 2014.

[7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, Ruslan R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," in *arXiv:1207.0580*, 2012.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

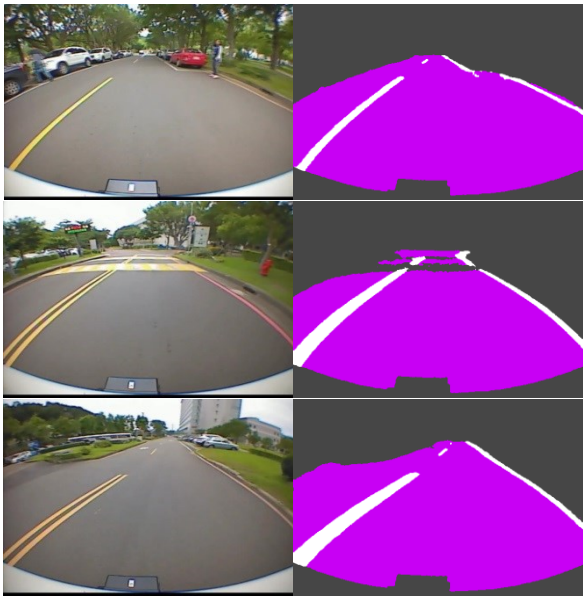

Figure 3: Examples of output on testing data

Table 1: Results of the proposed model on our dataset

Network	Global accuracy	Class average accuracy	Mean iou	Performance (fps)
Baseline	0.97438	0.93582	0.78680	15
The proposed one	**0.97676**	**0.94042**	**0.81137**	**28**