Supplementary Material of Learning Lightweight Lane Detection CNNs by Self Attention Distillation

Yuenan Hou[1], **Zheng Ma**[2], **Chunxiao Liu**[2], **and Chen Change Loy**[3†]
[1]The Chinese University of Hong Kong [2]SenseTime Group Limited [3]Nanyang Technological University
hy117@ie.cuhk.edu.hk, {mazheng, liuchunxiao}@sensetime.com, ccloy@ntu.edu.sg

1. Details of Architecture

Table 1 summarizes the architecture of the lane existence prediction branch for ENet-SAD, ResNet-18-SAD and ResNet-34-SAD. As to ResNet-18-SAD and ResNet-34-SAD, we also use dilated convolution [10] to replace the original convolution layers in the last two blocks for ResNet-18 [1] and ResNet-34 [1].

Table 1. The architecture of the the lane existence prediction branch. Assuming the input is $3 \times 288 \times 800$. Note that the output size is $c \times h \times w$ before "Flatten", where c, h and w denote channel, height and width, respectively. The number in the bracket besides the layer name is the parameter for that layer. For instance, the four numbers besides dilated convolution denote kernel size, stride, padding and dilated rate, respectively.

Layer Name	Output Size
Dilated Convolution (3, 1, 4, 4)	$32 \times 36 \times 100$
Batch Normalization	$32 \times 36 \times 100$
Relu	$32 \times 36 \times 100$
Spatial Dropout (0.1)	$32 \times 36 \times 100$
Convolution (1, 1)	$5 \times 36 \times 100$
Spatial SoftMax	$5 \times 36 \times 100$
Average Pooling	$5 \times 18 \times 50$
Flatten	4500
Fully Connected	128
Relu	128
Fully Connected	4
Sigmoid	4

2. Lane Post-processing in CULane

For CULane, in the inference stage, we feed the image into the ENet model. Then the multi-channel probability maps and the lane existence vector are obtained. Following [7, 3], the final output is obtained as follows: First, we use a 9×9 kernel to smooth the probability maps. Then, for each lane whose existence probability is larger than 0.5, we search the corresponding probability map every 20 rows for the position with the highest probability value. Finally, we use cubic splines to connect these positions to get the final output. The process improves the final lane prediction results as it removes noises in the probability maps. The process is depicted in Figure 1. Here, we differentiate different lane instances with different colors.

3. More Qualitative Results in Lane Detection

Figures 2 and 3 depict the qualitative results of different algorithms on TuSimple [9], CULane [7] and BDD100K [11]. As can be seen in Fig. 2, ENet-SAD can detect lanes more precisely than ENet [8] in TuSimple and CUlane. Besides, the detection of ENet-SAD [5] is less affected by the irrelevant objects on the road compared with SCNN [7]. As can be seen in Fig. 3, the output probability maps of ENet-SAD are more compact and contain less noise compared with those of SCNN in poor light conditions. Compared with conventional knowledge distillation methods [2, 4], SAD is more memory-efficient since it does not require a teacher model. Besides, ENet-SAD can also be applied to much larger lane detection datasets, *e.g.*, ApolloScape dataset [6].

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[3] Yuenan Hou. Agnostic lane detection. *arXiv preprint arXiv:1905.03704*, 2019.

[4] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning to steer by mimicking features from heterogeneous auxiliary networks. *arXiv preprint arXiv:1811.02759*, 2018.

[5] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. *arXiv preprint arXiv:1908.00821*, 2019.

[6] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory

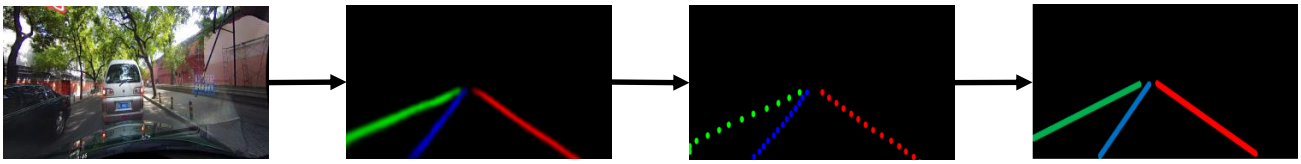

Figure 1. The process of obtaining lanes from probability maps on the CULane dataset. From left to right: original image, probability map, extracted lane points and final lane prediction.

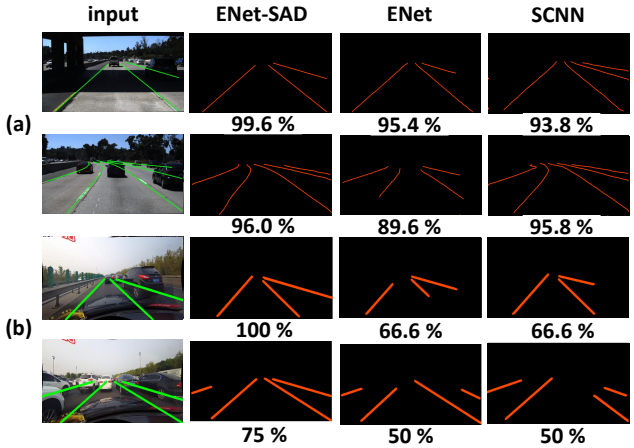

Figure 2. Performance of different algorithms on (a) TuSimple and (b) CULane testing sets. The number below each image denotes the accuracy. Ground-truth lanes are drawn on the input image.

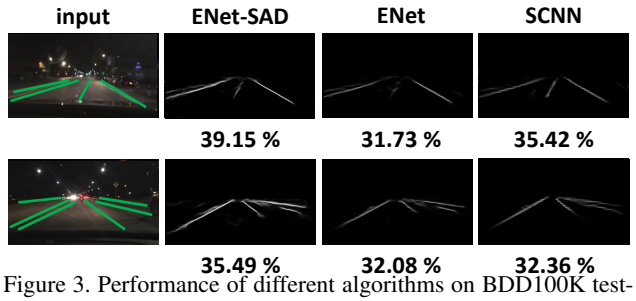

Figure 3. Performance of different algorithms on BDD100K testing set. We visualize the probability maps to better showcase the effect of adding self attention distillation. The brightness of the pixel indicates the probability of this pixel belonging to lanes. The number below each image denotes the pixel accuracy of lanes. Ground-truth lanes are drawn on the input image.

prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6120–6127, 2019.

[7] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial CNN for traffic scene understanding. In *Association for the Advancement of Artificial Intelligence*, 2018.

[8] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[9] TuSimple. http://benchmark.tusimple.ai/#/t/1. Accessed: 2018-09-08.

[10] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016.

[11] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.