Efficient Dense Spatial Pyramid Network for Lane Detection

Yinggang Li¹,² and Xiaobo Lu¹,²,*

¹School of Automation, Southeast University, Nanjing 210096, China; ²Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, 210096, China
*Corresponding author: xblu2013@126.com

Abstract. Lane detection separating the roads and positioning each lane is a significant task in Autonomous-driving field. Most of existing methods take lane detection as a semantic segmentation problem, and the methods based on encoder-decoder networks achieve robust performances than traditional methods with the disadvantages of large amounts of parameters and high computational cost. To improve efficiency and better adapt to adverse environments, we propose an EDSPnet by compiling the efficient dense module of depthwise dilated separable convolution (EDD module) and dense spatial pyramid (DSP) module. The two proposed modules enable our network to make full use of the contextual information from different feature maps when compared to a chain of layers. Different from symmetrical segmentation networks, we concentrate on the encoder block and a simple decoder is designed to match the input resolution. The decoder is complemented by two methods of upsampling: bilinear interpolation and deconvolution, which guarantee the accuracy and efficiency simultaneously. Our method is evaluated on the TuSimple dataset, and the experimental results show that our network has less parameters and is still efficient even compared with the state-of-the-art semantic segmentation networks.

1. Introduction

Currently deep learning has become a widely used technology in the application of computer vision including lane detection. Line detection is a significant and challenging problem in self-driving cars and video surveillance analysis on traffic. Common methods use images as inputs to detect lanes due to low cost. Some traditional methods use colour transformation [9] or Hough transform to extract distinctive features. Visual cues based on stripe-like features and texture characteristic are integrated to separate lane boundary lines [24]. But these approaches have poor robustness especially on occlusion issues. Seokju Lee et al. [8] published a lane and road marking benchmark dataset under different weather conditions. The grid-level annotation is induced to a data layer, which enables the algorithm to integrate independent lane and road marking. Vanishing Point Prediction [15] module that guides robust lane and road marking detection similar to human vision separate the input image into four sections to infer a global geometric context. However, this splendid approach should take input images from a certain perspective. Fabio et al. [17], Yuetong Du et al. [3] and Ze Wang et al. [22] treated the lane detection as a semantic segmentation or instance segmentation problem. These works mainly differ in networks.

In this paper, we build an EDSPnet upon the efficient architecture [10]. Compared with the original network, our improved network has less parameters but with better performances. The important features in EDSPnet are EDD module and DSP module which are designed based on the convolutional factorization principle. The asymmetric convolution [20] decomposes a standard convolution with kernel n×n into two standard convolutions with kernel n×1 and 1×n respectively. The factorization of
standard convolution does not weaken the learning ability but reduces lots of parameters. While the depthwise separable convolution maintain the learning property with lower computational cost than asymmetric convolution. For those fairshaped lanes, a big receptive field provided by dilated convolution [23] is essential for context understanding. Thus, the depthwise separable dilated convolution is used in the DSP module. The bilinear interpolation upsampling and deconvolution are utilized in decoder in consideration of computational cost and less coarse results. Furthermore, we also use feature concatenation to improve the heat map resolution.

The rest part of this paper is organized as follows: Some classical segmentation networks which inspired us are summarized in Sect.2. In Sect.3 we propose an efficient network named EDSPnet whose core parts are EDD module and DSP module. Then, the results produced by different networks will be compared in Sect.4. Finally, a conclusion is drawn in Sect.5.

2. Related Work
Lane detection task is not simply like the object detection problem in which objects’ positions can be approximately described with rectangular bounding boxes. Due to the long and thin shape of a lane, the classical object detection networks are not applicable for lane detection. Thus, it is often cast to a semantic segmentation problem [3, 13, 14, 17, 22]. Our EDSPnet is based on the two different types of semantic segmentation networks.

Symmetrical encode-decoder: These networks [1, 18, 19] are often easy to understand and have clear architectures. They usually use stacked convolutions at different stages and convolution with strides which are greater than 1 or pooling to do downsampling operations. The decoders are often reversed replicas of the encoders or are slightly modified in comparison to the encoders. The higher resolution feature maps often have abundant and original information. In order to benefit from this character, pooling indices [1], concatenation [19] and elementwise addition [4, 6] are utilized in the upsampling stages.

Module based architecture: Our work is mainly inspired by this structure. Efficient dense modules with asymmetric convolution in [10], efficient spatial pyramid module in [9] and shuffle unit in [11] are the fundamental modules in constructing the architectures. Instead of similar encoder and decoder, their work focused more on the compression. A lightweight decoder, for example, a bilinear interpolation [10], is then followed to match the input resolution. Similar to stacking dilated convolutions, these modules may have larger receptive fields as the networks go deeper. Convolution factorization has lower computational cost while maintaining the efficiency. Thus, group convolution, depthwise convolution and atrous convolution are the key operations in modules.

3. Network Architecture
Our complete network architecture is described in Figure 1. An intuition is that, for a semantic segmentation task whose input and output are equal in resolution, strong downsampling will lose spatial and contextual information. While to do convolutional operations on high resolution feature maps brings large computational cost. Thus, we downsample the input early in the network with a downsampling block [10]. A downsampling block has two branches. One is a 3×3 standard convolution with stride 2 and the other is a 2×2 max-pooling with stride 2. The results are then concatenated. This guarantees a balanced harvest in efficiency and information preservation. Stacked and multiple convolution layers enable the network learn more features. Meanwhile, the number of parameters is getting larger especially when the channels rapidly increase. Therefore, we follow the dense connectivity [7] of EDD and DSP modules respectively in different resolutions. The sketch map is shown in Figure 2. It maintains few parameters and strong learning ability even the network goes deeper. For the decoder, the bilinear interpolation is firstly used to upsample the feature maps by a factor of 4. Because it does not provide any parameter. However, simple interpolation results in coarse segmentation. Therefore, we concatenate the result with the output which is obtained by the first downsampling block for feature fusion. A deconvolution layer is employed to enable the decoder to learn more parameters.
3.1. **EDD Module**

Usually, the earlier networks constructed by consecutive convolutional layers without skip connections propagate the learned features in a straight way. This has many constraints in learning complex and miscellaneous features. In our EDD module, the initial input and the learned feature map are concatenated as the input for the next one. Additionally, a new module learns new features. We first use a 1x1 convolution to reduce the input channel for low computational cost because the dense connectivity has resulted in multistep features. A depthwise convolution [2] and a depthwise dilated separable convolution are then followed. Compared to the standard convolution, they produce less parameters and are still effective. The parameters of two pairs of asymmetric convolution are computed as:

$$K \cdot 1 \cdot M \cdot N \cdot 2 \cdot 2$$  \hfill (1)

In our EDD module, the parameters are:

$$K \cdot K \cdot M \cdot 1 + K \cdot K \cdot M \cdot 1 + 1 \cdot 1 \cdot M \cdot N$$  \hfill (2)

Where K is the kernel size, M denotes the number of input channels and N indicates the number of output channels. We set K to 3, M to 40 and N to 40 in this module. Our module in this part reduces parameters by 8 times. The clear structure is described in Figure 3.

3.2. **DSP Module**

Occlusion, blurs and dashed shape of lanes require big receptive fields to aggregate more contextual information. Instead of replicating the EDD module in the second block, we substitute it with a DSP module which is modified upon the efficient spatial pyramid module in [12]. We add a “channel split” operation to separate the input into 4 branches after the channel contraction. Each branch is a depthwise dilated separable convolution with different dilation rate. The paralleled branches construct...
a spatial pyramid of feature maps with different receptive fields. Considering that the feature map is already in small size, a receptive field of $33 \times 33$ is large enough for learning contextual representations. We also use hierarchical feature fusion [9] to avoid gridding artifacts instead of simple concatenation. The structure and concrete parameters are shown in Figure 4.

![Figure 3. EDD module. Numbers in the parentheses are input and output channels.](image)

![Figure 4. DSP module. W and k are respectively input and output dimensions.](image)

Batch normalization and nonlinear activation function Prelu are adopted after each convolution layer in our network.

4. Experiments

We evaluate our method on two challenging dataset. One is TuSimple [21] which contains 3626 for training and 2782 for testing, the other is CULane [15] which is consisted of 88880 for training set, 9675 for validation set and 34680 for test set.

4.1. TuSimple

As we cast lane detection to a segmentation task and the labels in TuSimple are simply provided in the form of discrete coordinates. We fill the lanes with a width of 5 pixels in training set and 10 pixels in test set. All training and testing images are resized at $512 \times 256$ resolution. We train the network using Adam optimizer with a learning rate $5 \times 10^{-4}$ and an end learning rate $1 \times 10^{-4}$ with a polynomial decay whose exponent is set to 0.9, along with batch size of 12. All dropout rates are set to 0.1. The dilation rates in dense EDD modules are 1, 1, 1, 2 and 2. Considering the unbalanced positive and negative samples, we refer to the definition of class weighting scheme in [16].

| Method           | F-measure(%) | Parameters(M) | Inference time (ms) |
|------------------|--------------|---------------|---------------------|
| EDAnet           | 53.8         | 0.68          | 17.8                |
| Enet             | 36.6         | 0.37          | 26.3                |
| LaneNet(Vgg16)   | 54.4         | 21.7          | 42.1                |
| Segnet           | 53.1         | 29.5          | 53.6                |
| EDSPNet(ours)    | **54.8**     | 0.45          | 18.0                |

Predictions correctly cast to lane class are viewed as true positives (TP), wrongly predicted are viewed as false positives (FP) and missed lane labels are viewed as false negatives (FN). We use F-measure $= \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \beta^2 \times \text{Recall}}$ as final evaluation index, where $\text{Precision} = \frac{TP}{TP + FP}$, $\text{Recall} =$
We set $\beta$ to 1 corresponding to harmonic mean. All results were tested on NVIDIA GeForce GTX 1080 GPU. The LaneNet (Vgg16) is only the binary branch we extract from [5] for fair comparison.

It is noting that the semantic segmentation labels are created based on the known coordinates in TuSimple. Consequently, the predicted lanes have wider width than ground truths, which leads to low precision. Generally, our EDSPnet and [10] demonstrate the superiority in inference speed. In terms of network lightness, our method is only inferior to [16] but achieve big improvements on F-measure and speed.

In Figure 5, we can obviously find that EDSPnet and [10] have superior performance than [16]. Locally, the input image in the first row has no clearly colored lane lines. Our method gives almost perfect result with less positive predictions. In row 2, we can see that there is a large gap in the dashed line near the view while we are still able to restore the blank. Predicting marginal lane lines in the third row is a very challenging task for all methods. However, we can find that they all show excellent robustness when there is not too much occlusion.

\[
\frac{TP}{TP+FN}
\]

**Figure 5.** Predictions of some examples generated by different networks

4.2. CULane

The backbone of our network remains same as described in Fig.1. For better adapt to the labels in the dataset, we add another branch to the network.

**Figure 6.** EDSPnet for specific lane detection dataset. ‘FC’ and ‘c’ denotes fully connected layer and output channels respectively.
Table 2. Testing results on CULane dataset

| Method    | Normal (%) | Crowded (%) | Night (%) | Shadow (%) | Arrow (%) | Curve (%) | Dazzle light (%) | No line (%) |
|-----------|------------|-------------|-----------|------------|-----------|-----------|------------------|-------------|
| EDAnet    | 93.1       | 75.6        | 67.7      | 72.2       | 83.8      | 71.1      | 70.0             | 51.3        |
| EDSPnet   | 93.5       | 77.6        | 70.8      | 78.5       | 85.0      | 74.5      | 71.4             | 53.1        |

Different from the pixel-wise evaluation in TuSimple, we calculate the intersection-over-union (IoU) between predictions and groundtruths. When the result is larger than 0.3, we view it as true positive. Only [10] which has comparable computational cost with our method is tested. The iteration number sets to 80K and other parameters remain same as in Sect.4.1. Only F-measure is shown in Table 2. From the results, we can see that our approach shows greater robustness as the environments get adverse.

5. Conclusion

We have introduced our EDSPnet for lane segmentation based on aggregating the advantages of dense connectivity and spatial pyramid. Two principles guide our design of the network. One is convolution factorization and the other is dilated convolution pyramid which provides features with multi-scale receptive fields. They collaboratively result in light network and less memory requirement. Cooperation between parameter-free bilinear interpolation upsampling and deconvolution demonstrates its effectiveness. Our network learns less parameters but still efficient. The pixel-wise and lane-wise evaluations on different dataset have demonstrated the superiority of our method.

Even through our main idea is to take advantage of full contextual information to solve occlusion. The method still has difficulty in this problem and produces not very accurate predictions. This is also because our labels are only approximately estimated labels. Besides, some lanes far from the camera are provided with entangled predictions. This may be improved by perspective transformation.

6. References

[1] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), 2481-2495.
[2] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).
[3] Du, Y., Gu, X., Liu, J., & He, L. (2018). Multiple Encoder-Decoders Net for Lane Detection.
[4] Gamal, M., Siam, M., & Abdel-Razek, M. (2018). Shuffleseg: Real-time semantic segmentation network. arXiv preprint arXiv:1803.03816.
[5] github.com/MaybeShewill-CV/lanenet-lane-detection
[6] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Le, Q. V. (2019). Searching for mobilenetv3. arXiv preprint arXiv:1905.02244.
[7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
[8] Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., ... & So Kweon, I. (2017). Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1947-1955).
[9] Li, M., Li, Y., & Jiang, M. (2018). Lane Detection Based on Connection of Various Feature Extraction Methods. Advances in Multimedia, 2018.
[10] Lo, S. Y., Hang, H. M., Chan, S. W., & Lin, J. J. (2018). Efficient dense modules of asymmetric convolution for real-time semantic segmentation. arXiv preprint arXiv:1809.06323.
[11] Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer
Vision (ECCV) (pp. 116-131).

[12] Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., & Hajishirzi, H. (2018). Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 552-568).

[13] Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., & Van Gool, L. (2018, June). Towards end-to-end lane detection: an instance segmentation approach. In 2018 IEEE Intelligent Vehicles Symposium (IV) (pp. 286-291). IEEE.

[14] Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., & Van Gool, L. (2017). Fast scene understanding for autonomous driving. arXiv preprint arXiv:1708.02550.

[15] Pan, X., Shi, J., Luo, P., Wang, X., & Tang, X. (2018, April). Spatial as deep: Spatial cnn for traffic scene understanding. In Thirty-Second AAAI Conference on Artificial Intelligence.

[16] Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147.

[17] Pizzati, F., Allodi, M., Barrera, A., & García, F. (2019). Lane Detection and Classification using Cascaded CNNs. arXiv preprint arXiv:1907.01294.

[18] Romera, E., Alvarez, J. M., Bergasa, L. M., & Arroyo, R. (2017). Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Transactions on Intelligent Transportation Systems, 19(1), 263-272.

[19] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[20] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[21] TuSimple dataset. http://benchmark.tusimple.ai/

[22] Wang, Z., Ren, W., & Qiu, Q. (2018). LaneNet: Real-Time Lane Detection Networks for Autonomous Driving. arXiv preprint arXiv:1807.01726.

[23] Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

[24] Zhou, S., Jiang, Y., Xi, J., Gong, J., Xiong, G., & Chen, H. (2010, June). A novel lane detection based on geometrical model and gabor filter. In 2010 IEEE Intelligent Vehicles Symposium (pp. 59-64). IEEE.