PPEDNet: Pyramid Pooling Encoder-Decoder Network for Real-Time Semantic Segmentation

Zhentao Tan[1,2], Bin Liu[1,2](✉), and Nenghai Yu[1,2]

[1] Key Laboratory of Electromagnetic Space Information,
Chinese Academy of Sciences, Hefei, China
[2] School of Information Science and Technology,
University of Science and Technology of China, Hefei, China
flowice@ustc.edu.cn

Abstract. Image semantic segmentation is a fundamental problem and plays an important role in computer vision and artificial intelligence. Recent deep neural networks have improved the accuracy of semantic segmentation significantly. Meanwhile, the number of network parameters and floating point operations have also increased notably. The real-world applications not only have high requirements on the segmentation accuracy, but also demand real-time processing. In this paper, we propose a pyramid pooling encoder-decoder network named PPEDNet for both better accuracy and faster processing speed. Our encoder network is based on VGG16 and discards the fully connected layers due to their huge amounts of parameters. To extract context feature efficiently, we design a pyramid pooling architecture. The decoder is a trainable convolutional network for upsampling the output of the encoder, and fine-tuning the segmentation details. Our method is evaluated on CamVid dataset, achieving 7.214% mIOU accuracy improvement while reducing 17.9% of the parameters compared with the state-of-the-art algorithm.

Keywords: Semantic segmentation · Pyramid pooling · Real-time

1 Introduction

Image semantic segmentation is to divide an image into several regions with each region having the same semantic implication. Over the years, researchers have proposed many powerful algorithms, which can be roughly grouped into two categories: traditional approaches and deep convolutional neural network (DCNN) based approaches. Traditional approaches rely on low-level vision cues, such as Normalized cut [24]. They are not suitable for complex scenes due to their limited performance. By contrast, recent approaches have achieved remarkable success by applying deep convolutional neural network to this pixel-level labeling task [6,18,19,30]. DCNN is applied to classification tasks in the early days, such as handwritten digit recognition, image classification and object detection. Recently, the availability of large scale well annotated datasets and

Y. Zhao et al. (Eds.): ICIG 2017, Part I, LNCS 10666, pp. 328–339, 2017.
https://doi.org/10.1007/978-3-319-71607-7_29

computationally-powerful machines have pushed forward the development of deep convolutional neural network. Moreover, it has been widely proved that the well-trained DCNN models [12,25,26] pretrained on these large scale datasets can be transferred to other vision tasks, like image semantic segmentation. For better performance, deeper and larger convolutional neural networks are explored [15,21,29], requiring more computing resources and inference time.

However, the real-world applications such as augmented reality wearables, self-driving vehicles and other automatic devices have a strong demand for image semantic segmentation algorithms that can process in real-time. Taking self-driving vehicles as an example, the complex traffic environment requires autopilot system can deal with emergency timely and effectively. Obviously, the existing architectures can not meet this requirement [7,10,21,29]. To solve this problem, several neural networks have been proposed to balance the segment accuracy and inference time, such as SegNet [1] and ENET [23]. These networks pay more attention to complex segmentation tasks such as in road and indoor scenes, and achieve a fast segmentation speed at the cost of accuracy.

In this paper, we propose a new convolutional neural network architecture which achieves higher segmentation accuracy and faster inference speed. Our network is primarily motivated by the road scene dataset [4] which requires modeling both appearance and shape, understanding the context between different classes such as the road surface and the side-walk. The main contributions of this paper can be summarized as follows: (1) we propose a new DCNN-based network architecture which reduces the model size notably; (2)we explore a well-designed pyramid pooling architecture to extract contextual information; (3) we build a practical system for semantic segmentation which outperforms existing approaches with similar processing speed [1,19,23].

The rest of this paper is organized in the following order. In Sect. 2 we review the related work about image semantic segmentation. In Sect. 3 we introduce the pyramid pooling encoder-decoder network architecture and discuss the advantages of this architecture. In Sect. 4, we evaluate our network empirically and compare it with other networks. Finally, we give a conclusion on our work in Sect. 5.

2 Related Work

With the development of convolutional neural networks, image semantic segmentation has achieved unprecedented performance recently. Fully convolutional neural network (FCN) [19] was the first algorithm used in PASCAL VOC 2012 segmentation tasks [9]. This method was based on VGG16 and changed its fully connected layers to convolutional ones. Pre-trained on ImageNet [2], FCN can extract the features of object efficiently and outperform all previous methods. This successful attempt encouraged other researchers to exploit deep network architecture for better segmentation results.

The direct prediction of FCN based methods are usually in low resolution. To obtain high resolution predictions, many recent methods focus on refining

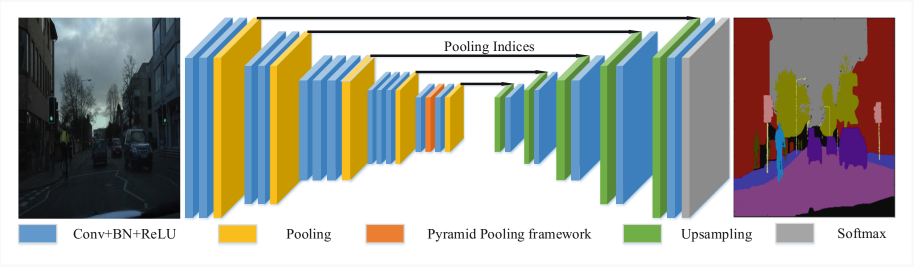

Fig. 1. The architecture of pyramid pooling encoder-decoder network

the low resolution predictions. DeepLab-CRF [6] performed bilinear upsampling of the score map to the input image size and applied the fully connected conditional random fields [14] to refine the object boundary. The work in [21] trained deconvolutional layers to upsample the low resolution predictions. CRF-RNN [30] applied a recurrent neural network to replace conditional random fields for end-to-end training. To reduce the computation time, Liu et al. [18] and Lin et al. [15] both designed an efficient approximate inference algorithm for fully connected CRF models. The network proposed in [5] extracted the edge feature maps and applied a discriminatively trained domain transform so as to combine it with the score maps from FCN. The networks such as [16,22,28] used context information for finer segmentation results. These networks achieve high score in image semantic segmentation challenges like PASCAL VOC 2012 [9], but can not meet the requirement of real-world application because of their large network architectures.

Unlike employing the whole CNNs directly, SegNet [1] discarded the fully connected layers of VGG16, so as to reduce the number of parameters. Furthermore, this network only stored the max-pooling indices in the encoder to its corresponding upsampling layers. As a result, SegNet had a great performance both on segmentation accuracy and processing speed. Another network, named ENET [23], focused on real-time image segmentation and chose to pre-train it's own encoder network in ImageNet classification task to avoid overlarge network architecture. Tested on an NVIDIA Titan X GPU, ENET achieved the fastest implementation speed, more than 100 fps. However, the high segmentation speed was built at the sacrifice of segmentation accuracy.

3 Network Architecture

Our network architecture PPEDNet (Pyramid Pooling Encoder-Decoder Network) is shown in Fig. 1. It consists of a large encoder network, a corresponding small decoder network followed by a pixel-wise classification layer. The encoder corresponds to the feature extractor that transforms the input image to multidimensional feature representation, whereas the decoder is a shape generator that produces segmentation result from the features extracted from the encoder. Following the process of image segmentation, we first present the encoder network and then the decoder network.

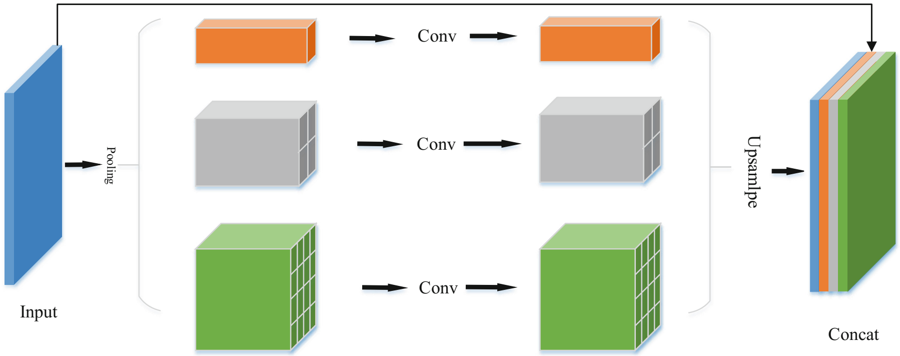

Fig. 2. Pyramid pooling model

3.1 Encoder Network

Feature extraction is the premise and core of pixel-wise classification. Thus, a powerful encoder network is of great importance. Our feature extraction framework is based on VGG16 which is a very successful image classification network. Therefore, we can initialize the training process from weights trained for classification on large datasets [2]. Each *encoder* in the encoder network performs convolution with a filter bank to produce a set of feature maps. They are then batch normalized. Following that, an element-wise rectified-linear non-linearity (ReLU) max $(0, x)$ is performed. Max pooling with a 2×2 window and stride 2 (non-overlapping window) is used to result in a large input image context (spatial window) for each pixel in the feature maps. Different from the original VGG network, we remove the fully connected layers because these layers consume too many parameters. Besides, the most significant change is that we have designed a pyramid pooling framework.

Most convolutional neural networks like FCN [19] and DeepLab [6] only predict each pixel independently, without considering context relationship between each receptive field. This limits the ability of diverse scenes understanding, and networks in [6,19] may usually make mistakes when there exists similar appearance inter class. Although some post-processing methods such as conditional random field, can calculate the pairwise potential and smooth noisy segmentation maps, the complicated inference operations reduce the processing speed severely. Liu et al. [17] tried to learn global context with global average pooling, and yielded encouraging improvement. Compared with the Parsenet [17], pyramid pooling model has a stronger ability of extracting and combining different regional characteristics.

The pyramid pooling model is shown in Fig. 2. The input is high-level feature maps, which in our network is the output of Conv5-1 of VGG16. Then, three different pyramid scales are used to extract different sub-region features from the input feature maps, forming pooled representation for different locations. The outputs of different levels of the pyramid pooling model contain feature maps of different sizes. We use 1×1 convolutional operation after each pyramid pooling layer to adjust the weights of every channel. To concatenate these feature maps with the original one, a direct upsampling operation is used to resize the low-dimensional feature maps, producing the desired feature map via bilinear interpolation. Finally, different levels of feature maps are concatenated as a hybrid multi-scale context input for further convolutional layers. Compared to other multi-scale pooling modules, our pyramid pooling model extracts the multi-scale feature maps in the same high-level maps, and concatenates these new contexts directly without several 3×3 convolutional operations. This makes the raw hybrid multi-scale context with the same resolution, which provides strong evidence for classification.

The number of pyramid pooling levels and sizes can be modified according to how many sub-regional contexts we want to combine. Considering the input images of CamVid dataset with a resolution of 480×360 (Width, Height), the feature maps which are the input of the pyramid pooling model (30×23) are too small to be divided into many levels. So, our pyramid pooling model has three levels with bin sizes of 1×1, 2×2, and 4×4. Furthermore, we note that road scene images could always be divided into three parts: the road in the middle and the buildings on the two sides. A three-level pyramid pooling model with sizes of 1×1, 3×2, and 6×4 should be more reasonable. For convenience, we name these two frameworks as the original pyramid pooling and the attentional pyramid pooling respectively. Inspired by Zhao et al. [29], we choose average pooling as the type of pooling operation. With the pyramid pooling model, our network extracts an effective global context for pixel-level scene parsing.

3.2 Decoder Network

With several layers of max-pooling, low-resolution feature maps from the encoder network have a loss of spatial resolution, which is unbeneficial to segmentation. Thus, we need an appropriate method to upsample these feature maps to dense high-resolution segmentation image with the same size as original input image. Recent work has pursued two directions to address localization challenge. The first approach is to employ information from multiple layers in the network [19] or a super-pixel representation [20] to optimise the edge segmentation. The second approach is to design a trainable decoder network to learn deconvolution [1,21]. These decoder networks are always the mirror of the encoder networks.

In our method, we propose an asymmetric encoder-decoder network that is different from the one presented in [1]. This is motivated by the idea that the encoder network should extract the appearance and shape features, providing strong evidence for classifier. By contrast, decoder network is only required to restore the resolution of the input feature maps and fine-tune the segmentation details. To explore the influence of decoder network on segmentation performance, we choose SegNet [1] as the baseline, and try three different decoder networks. First, we remove the deconv3-3, deconv4-3 and deconv5-3 convolutional layers in SegNet. Then, only one convolutional layer is retained between two upsampling layers. Further more, we try to replace the convolutional layers with the bottleneck architecture presented in ENET [23] for its success. Different from the original bottleneck, we make some changes which are shown in Fig. 3. Figure 3(a) is the original convolutional block, it consists of three parts, a 3 × 3 convolutional layer, a batch normalization layer and a rectified linear unit. Figure 3(b) is the bottleneck module which has two branches. On the right of the branch includes three convolutional layer: a 1 × 1 projection that reduces the channels of feature maps, a main convolutional layer with 3 × 3 kernels, and a 1 × 1 expansion that resizes the channels. On the left of the branch is a 1 × 1 adjustion matches the number of the channels. The details of the experiment are shown in Sect. 4. As a result, we choose the second approach for the balance of accuracy and speed.

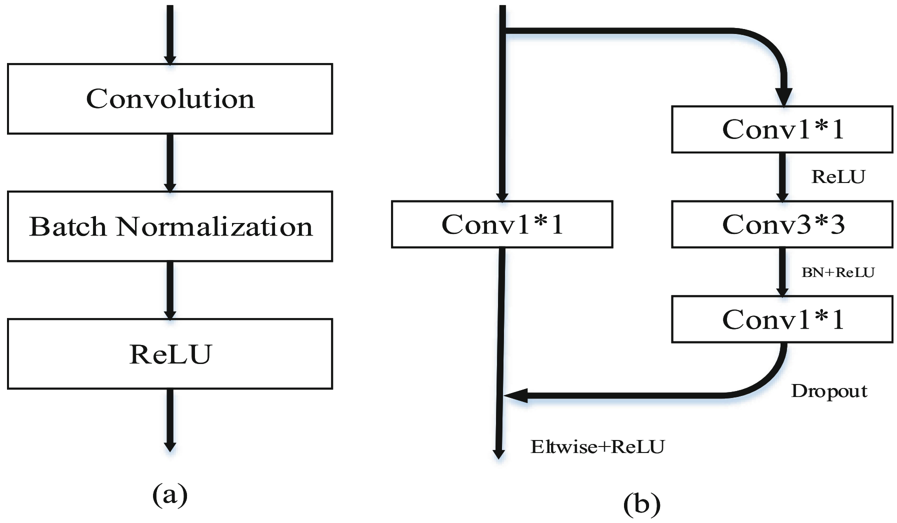

Fig. 3. The comparison of convolutional module. (a) Original convolutional block. (b) Bottleneck module.

Another noteworthy point is that the *decoder* in the decoder network upsamples its input feature maps using the memorized max-pooling indices from the corresponding encoder feature maps. Inspired by SegNet [1], we only store the max-pooling indices, i.e., the location of the maximum feature value in each pooling window is memorized for each encoder feature map. For intuitive comparison, we inference DeconvNet [21] which applies this upsamle technology, and the required memory is reduced greatly (from 1872M to 1174M).

4 Experiments

We conduct experiments on CamVid dataset [4]. This dataset consists of 367 training and 233 testing RGB images at 960 × 720 resolution. There are eleven different classes such as tree, car, building, etc.[1] To reduce the computational requirements, we reshape the image to 480 × 360 before training.

Table 1. Comparison of decoder variants.

Model	GA/%	CAA/%	mIoU/%	IM/MB	MS/MB	IS/FPS
SegNet	87.462	69.531	**56.984**	1038	112.4	13
SegNet-1	**87.492**	**71.116**	56.893	987	92.1	14.3
SegNet-2	87.445	70.956	56.89	**873**	71.3	**16.8**
SegNet-3	69.993	58.958	39.834	1235	**59.4**	11.3

4.1 Decoder Variant

We train three different decoder variants described in Sect. 3 on CamVid dataset. Inspired by SegNet [1], the encoder weights are initialized by VGG16 model pretrained on ImageNet classification challenge, the decoder weights are initialized using the technique in [11]. All the variants are trained using stochastic gradient descent (SGD) [3] with a fixed learning rate of 0.001 and momentum of 0.9. Before each epoch, the training set is shuffled and each mini-batch (4 images) is then picked in order to ensure that each image is used only once in each epoch. The weighted cross-entropy loss is used as loss function for training the network. There is a need to balance the weights since there is too many differences between the number of each class in the set. The balance strategy is named median frequency balancing [8], which is assigned to calculate the ratio of the median of class frequency on the entire training set, implying larger classes have smaller weights while smaller classes have higher weights. To compare the three different decoder variants quantitatively, we use six commonly used performance measures: global accuracy (GA) measures the percentage of pixels that are correctly classified in the entire dataset, class average accuracy (CAA) is the predictive accuracy over all classes, mean intersection over union (mIoU) is a more stringent metric than class average accuracy since it penalizes false positive predictions, inference memory (IM) is the memory requirement to segment images, model size (MS) means the number of parameters and inference speed (IS) tests the segmentation efficiency.

The experiment results are illustrated in Table 1. There are three decoder variants named SegNet-1, SegNet-2 and SegNet-3. SegNet-1 removes the deconv3-3, deconv4-3 and deconv5-3 convolutional layers. SegNet-2 only retains

[1] The twelfth class contains unlabeled data, which is ignored while training.

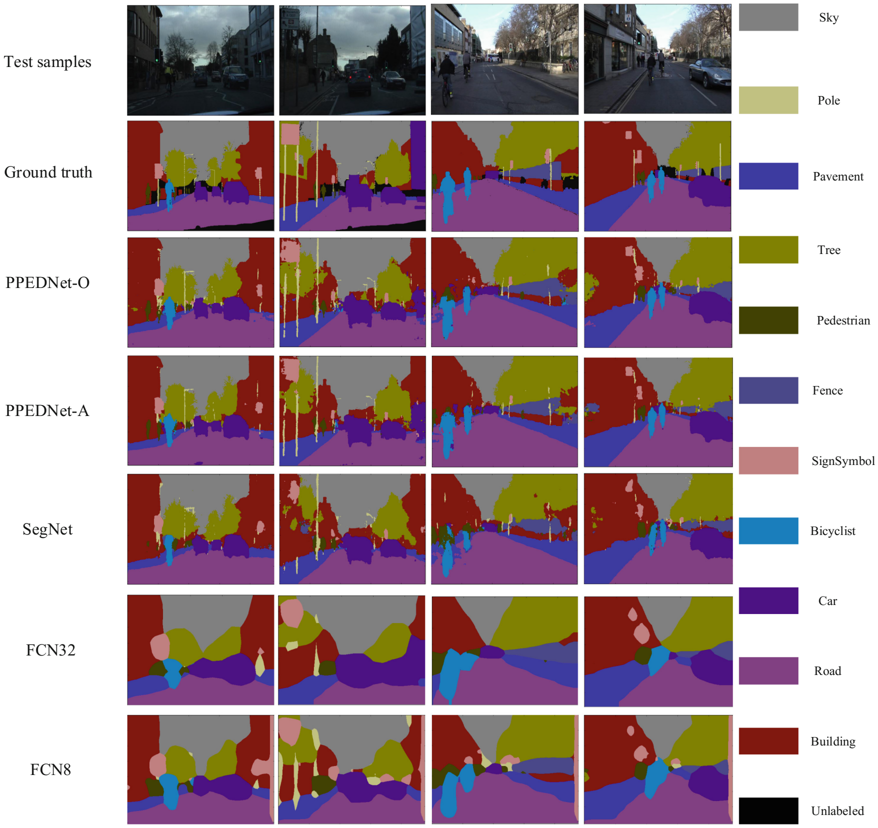

Fig. 4. Results on CamVid day and dusk test/val samples.

one convolutional layer between two upsampling layers. SegNet-3 is based on SegNet-2 and replaces the convolutional layers with the bottleneck architecture in the decoder. Compared with the baseline (SegNet), SegNet-2 reduces 34.6% model size, accelerates more than 3 fps and has only 0.094% mIoU loss. It is notable that when it comes to global accuracy and class average accuracy, SegNet-1 even has better performance than the baseline. This suggests that some parameters in decoder layers are redundant for upsample. However, the huge successful bottleneck architecture in ENET [23] has a poorer performance, reduce 17.15% mIoU, 17.47% global accuracy and 10.57% class average accuracy. Further more, this architecture pluses feature map with the lower one, which infinitely increasing the required inference memory, thus probably making slower inference speed.

Table 2. The comparison of accuracy among different classes.

Model	Sky	Building	Pole	Road	Pavement	Tree	SignSymbol	Fence	Car	Pedestrian	Bicyclist
SegNet	93.11	87.82	35.15	94.21	86.07	81.16	56.07	39.41	80.84	71.90	39.10
ENET	**95.1**	74.7	35.4	95.1	86.7	77.8	51.0	51.7	82.4	67.2	34.1
ReSeg	93.0	86.8	35.6	**98.0**	87.3	84.7	48.6	20.9	**87.3**	63.3	43.5
FCN32	90.27	74.39	10.22	74.05	84.65	80.73	40.47	**54.83**	80.83	53.03	46.45
FCN8	89.24	71.85	25.66	87.90	82.59	82.31	65.58	52.22	79.61	**74.90**	55.72
PPED-O	94.10	88.51	**48.37**	96.84	**87.79**	**87.93**	**57.89**	52.00	83.17	72.17	60.42
PPED-A	94.66	**90.73**	40.30	97.17	87.68	86.53	53.27	50.87	81.41	73.01	**62.05**

Table 3. Quantitative comparison of semantic segmentation on the CamVid test set when trained on its original train set.

Model	GA/%	CAA/%	mIoU/%	IM/MB	MS/MB	IS/FPS
SegNet	87.462	69.531	56.984	1038	112.4	13
FCN32	77.535	62.720	46.441	1271	512.4	12.1
FCN8	80.94	69.78	50.50	1290	512.5	11.7
PPED-O	89.956	**75.382**	63.294	884	92.3	**16.4**
PPED-A	**90.310**	74.334	**64.198**	884	92.3	16.2

4.2 Comparison

We compare our network with ENET [23], ReSeg [27], FCN × 32 and FCN × 8 [19] on the test set for their fine segment accuracy and inference speed. All the compared network architectures are trained on the original CamVid train set with 367 RGB images. The objective is to understand the performance of these architectures when trained on the same dataset. We add batch normalization [13] layers after each convolutional layer in order to end-to-end train the network. To provide a controlled benchmark we use the same SGD solver [3] with a fixed momentum of 0.9, the learning rate is unfixed for different convergence speed between these networks. A mini-batch size of 4 is set to ensure all architectures can be trained on an NVIDIA Titan X GPU and dropout of 0.5 is added for some deeper convolutional layers to prevent overfitting. VGG16 pre-trained model parameters are used in order to accelerate convergence and other layers' weights are initialized using the technique in He et al. [11]. There is no limit for maximum epoch, and all architectures are trained until no further performance increase is observed.

The results in Table 2 show the accuracy of each class that belongs to different architectures. ENET performance is provided by [23] while the data of ReSeg is provided by Visin [27]. Our network have two versions which are introduced in Sect. 3. PPED-O uses the original pyramid pooling framework, and PPED-A is with the attentional pyramid pooling framework. Both our two networks use the second decoder variant described in Sect. 3 as their decoder. It is clear that our networks get the highest accuracy in six of the categories. Some classes like pole and bicyclist on which SegNet and ENET get a poor segmentation result, are much better segmented by our networks. Different pyramid pooling models also influence the segmentation accuracy. Attention pyramid pooling outperforms in five classes which exactly distribute on the left, middle and right of an image. It is notable that particular designed pooling focuses on these three regions, making finer segmentation of some objects like pedestrian and road. But for other classes, especially someone which is between buildings and the road, original pyramid pooling has better performance. Table 3 shows more performance metrics, and our network outperforms existing state-of-the-art algorithms in all the metrics. Our model obtain a 17.9% less of parameters than SegNet, and accelerates implementation speed more than 3 images per second. For more detailed comparison, attention pyramid pooling has better performance in global accuracy and mIoU. But compared in class average accuracy, original pyramid pooling is better. The qualitative comparisons of our network predictions with other deep architectures can be seen in Fig. 4. It is clear that our proposed architecture has a stronger ability to segment smaller classes in road scenes.

5 Conclusion

We propose a novel neural network architecture for complex scene image semantic segmentation. The main motivation is the need of an efficient method for road scene understanding which works well in terms of both accuracy and computational time. For this objective, we propose a pyramid pooling encoder-decoder network architecture. We compare it with others in the metrics of mean of intersection over union (mIoU), inference memory, model size and particularly processing speed. The experimental results reveal both 7.214% mIoU improvement and more than 3 fps acceleration compared with existing state-of-the-art algorithm, SegNet [1].

Acknowledgement. This work is supported by the National Natural Science Foundation of China (Grant No. 61371192), the Key Laboratory Foundation of the Chinese Academy of Sciences (CXJJ-17S044) and the Fundamental Research Funds for the Central Universities (WK2100330002).

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
2. Berg, A., Deng, J., Fei-Fei, L.: Large scale visual recognition challenge (ILSVRC) (2010). http://www.image-net.org/challenges/LSVRC
3. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G. (eds.) Proceedings of COMPSTAT 2010, pp. 177–186. Physica-Verlag, Heidelberg (2010). https://doi.org/10.1007/978-3-7908-2604-3_16
4. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: a high-definition ground truth database. Pattern Recogn. Lett. **30**(2), 88–97 (2009)

5. Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4545–4554 (2016)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062 (2014)
7. Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3640–3649 (2016)
8. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658 (2015)
9. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
10. Ghiasi, G., Fowlkes, C.C.: Laplacian pyramid reconstruction and refinement for semantic segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 519–534. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_32
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
13. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
14. Lafferty, J., McCallum, A., Pereira, F., et al.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML, vol. 1, pp. 282–289 (2001)
15. Lin, G., Shen, C., van den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3194–3203 (2016)
16. Lin, G., Shen, C., van den Hengel, A., Reid, I.: Exploring context with deep structured models for semantic segmentation. arXiv preprint arXiv:1603.03183 (2016)
17. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: looking wider to see better. arXiv preprint arXiv:1506.04579 (2015)
18. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1377–1385 (2015)
19. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
20. Mostajabi, M., Yadollahpour, P., Shakhnarovich, G.: Feedforward semantic segmentation with zoom-out features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3376–3385 (2015)
21. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528 (2015)

22. Pan, T., Wang, B., Ding, G., Yong, J.H.: Fully convolutional neural networks with full-scale-features for semantic segmentation (2017)
23. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: a deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
27. Visin, F., Ciccone, M., Romero, A., Kastner, K., Cho, K., Bengio, Y., Matteucci, M., Courville, A.: ReSeg: a recurrent neural network-based model for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 41–48 (2016)
28. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.: Understanding convolution for semantic segmentation. arXiv preprint arXiv:1702.08502 (2017)
29. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. arXiv preprint arXiv:1612.01105 (2016)
30. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1529–1537 (2015)