DENSE DECONVOLUTIONAL NETWORK FOR SEMANTIC SEGMENTATION

Wenbin Yang[1,2,*], *Quan Zhou*[1,2,†], *Jingnan Lu*[1], *Xiaofu Wu*[1], *Suofei Zhang*[3], *and Longin Jan Latecki*[4]

[1]National Engineering Research Center of Communications and Networking,
Nanjing University of Posts & Telecommunications, Nanjing, P.R. China.
[2]State Key Lab. for Novel Software Technology, Nanjing University, P.R. China.
[3]School of Internet of Things, Nanjing University of Posts & Telecommunications, Nanjing, P.R. China.
[4]Department of Computer and Information Sciences, Temple University, Philadelphia, USA.

ABSTRACT

Recently, exploring multiple feature maps from different layers in fully convolutional networks (FCNs) has gained substantial attention to capture context information for semantic segmentation. This paper presents a novel encoder-decoder architecture, called *dense deconvolutional network (DDN)*, for semantic segmentation, where the feature maps of deeper convolutional layers are *densely* upsampled for the shallow deconvolutional layers. The proposed DDN is trainable end-to-end, and allows us to *fully* investigate multiple scale context cues embedded in images. The experimental results show that our DDN outperforms previous FCNs and encoder-decoder networks (EDNs) on PASCAL VOC 2012 dataset.

Index Terms— Semantic Segmentation, Dense Deconvolutional Network, EDNs, FCNs

1. INTRODUCTION

The recent years have witnessed the substantial progress of image semantic segmentation using fully convolutional networks (FCNs) [1, 2, 3, 4]. The representation power of FCNs leads to the successful results: a combination of feature descriptors extracted from FCNs are complementing each other to enhance segmentation performance [1, 3], and this simple off-the-shelf classifiers works very well for dense prediction problems [5, 6, 7, 8]. Although achieving promising results, the FCNs suffer from a couple of critical limitations. Firstly, due to the consecutive pooling or convolution striding at successive layers, the spatial resolution is significantly reduced in feature maps. This invariance to local image transformation may be harmful for dense prediction tasks, where detailed spatial information is often required to delineate object shapes and boundaries [2, 4]. Secondly, the existence of objects tend to be with multiple scales. However, the receptive field of previous FCNs is not adaptive, leading to the problem that objects substantially larger or smaller than the receptive field may be fragmented or incorrectly classified [2, 9, 10].

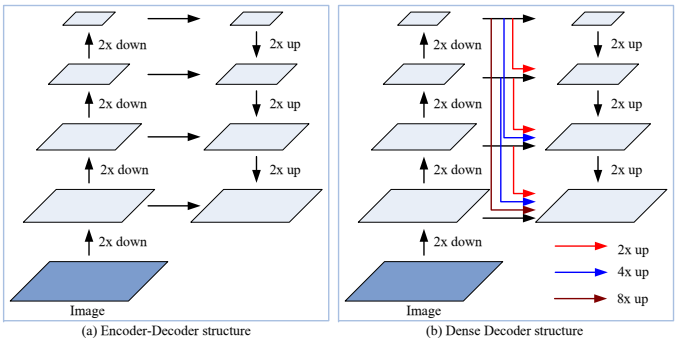

Fig. 1. Comparison of encoder-decoder architecture (a) and our dense deconvolution decoder structure (b) to capture multiple scale context. (Best viewed in color)

In order to overcome these two challenges, the encoder-decoder network (EDN) and its variants are proposed in recent literature [9, 10, 11, 12, 13]. As illustrated in Figure 1(a), the EDN consists of two parts. The encoder gradually reduces the spatial dimension of feature maps, and thus wide scale context cues are more easily captured in the deeper layer output. Conversely, the decoder progressively increases spatial dimension to recover object details using upsampling and deconvolution. For instance, [1, 10] employ deconvolution to learn the upsampling of low resolution feature responses. SegNet [9] reuses the recorded pooling indices from the encoder to upsample feature maps, and learns extra deconvolutional layers to densify the feature responses. Through adding skip connections, U-Net [11] introduces an elegant symmetric network architecture, which concatenates feature maps from the encoder side to the corresponding decoder activations. In [12], the authors employ a Laplacian pyramid reconstruction network, where the shallow feature maps are utilized to successively refine segment boundaries reconstructed from deeper feature maps. More recently, RefineNets [13, 14, 15, 16] have been also demonstrated that the encoder-decoder struc-

*This work was partly supported by NSF IIS-1302164, and NSFC 61881240048, 61401228, 61501247, 61501259, 61671253.

†Quan Zhou is the corresponding author.

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:58:02 UTC from IEEE Xplore. Restrictions apply.

ture is very effective on several semantic segmentation benchmarks [17, 18]. However, the EDNs still suffer from the following shortcomings. Although pooling indices are recorded to perform upsampling, the remaining elements are padded as zero, which may induce noise to the forthcoming deconvolution. In addition, the EDNs have a pre-defined fixed structure, where the skip connections are only constructed between the convolutional layer and its deconvolution counterpart, resulting in the fact that contextual cues are not fully investigated.

In this paper, we introduce a novel encoder-decoder architecture based on U-net [11] for semantic segmentation, called *dense deconvolution network (DDN)*. As illustrated in Figure 1(b), our DDN harvests and concatenates different scale representations in encoder to form the feature representation in decoder, which carries both local and global context information. Specifically, for one specific deconvolution feature representation, it is first concatenated with feature maps *densely* upsampled from deeper convolutional layers in encoder (denoted as colored arrows in Figure 1(b)), and then supplemented with its counterpart (denoted as black arrows in Figure 1(b)). Compared with traditional EDNs, the proposed DDN has two major contributions:

- Instead of using switch variables to record pooling indices [9, 10], the feature maps of encoder are concatenated to the counterpart of decoder, avoiding the unnecessary padding operation.
- The stacked feature maps are complementing each other, allowing us to *fully* explore multiple scale contextual information embedded in images.

We evaluated our DDN on popular benchmark [17], and the experimental results show the superior performance of our DDN for semantic segmentation without any postprocessing.

2. THE APPROACH

This section first elaborates on the architecture details of our DDN, and then introduces the learning scheme of our DDN.

2.1. The architecture detail of DDN

Figure 2 shows the detailed architecture of the entire DDN. Similar to previous EDNs [9, 10, 11], our DDN is also composed of two parts: convolutional network and deconvolutional network, which include five basic components: convolution, rectified linear unit (ReLU), duplication, max pooling and deconvolution. The convolutional network corresponds to feature extractor that transforms the input image to multiple scale dimensional feature representation. On the contrary, the deconvolutional networks delineate shape boundaries that output object segmentation from the convolution feature maps produced from convolutional network. The final output of our DDN is a 21-dimensional probability map with the same size

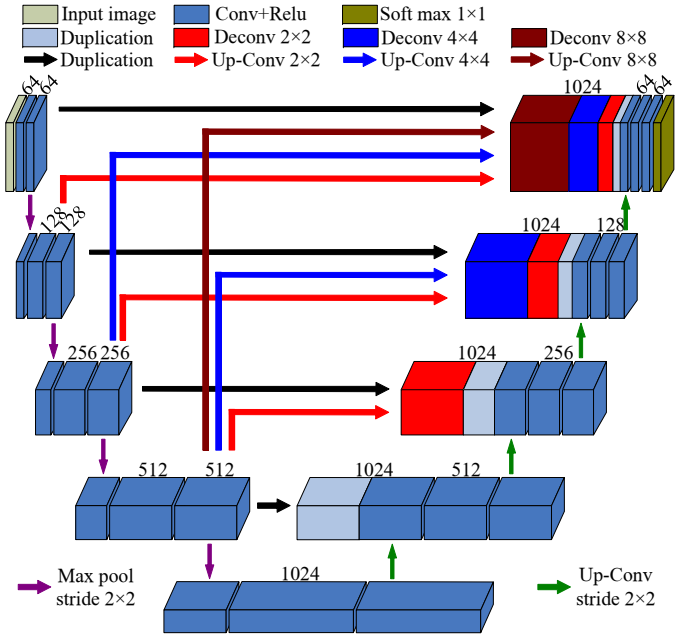

Fig. 2. Overall architecture of the proposed DDN. On top of the network based on U-net [11], we construct dense skip connections from encoder to decoder, producing the delineated segmentation map of an input image. Note the numbers of feature channels are marked on the top of each convolutional and deconvolutional layer. (Best viewed in color)

of input image, indicating probability of each pixel belonging to one of the predefined 20 classes or to the background.

More specifically, in the convolutional network, we borrow the network architecture widely used in FCN-based architecture [1, 2]. Unlike previous U-net [11] that employs unpadded convolution, our DDN consists of the repeated padded convolution with two 3×3 filter kernels, each followed by a ReLU and a 2×2 max pooling operation with stride 2 for downsampling. The padding operation introduces small noise before convolution, but will results in the segmentation outputs with the same resolution of input image, which is beneficial for dense estimation problem. Notice at each downsampling step, we double the number of feature channels. Since the spatial information is significantly lost with going deeper of network, it is very hard to recover small objects from convolutional feature maps with lowest resolution. Therefore, the final pooling layers and the following convolutional layers are removed in our convolutional network.

On the other hand, the deconvolutional network contains a series of deconvolutional layers, which include three steps: upsampling, concatenation and convolution. In each deconvolutional layer, the feature maps are first enlarged using a 2×2 upsampling ("up-convolution"), which halves the number of feature channels (denoted as green arrows in Figure 2). In concatenation step, unlike previous EDNs that directly

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:58:02 UTC from IEEE Xplore. Restrictions apply.

duplicate feature maps in encoder [11, 13], our DDN stacks feature representation from deeper convolutional layers (denoted as colored arrows in Figure 2), and the correspondingly counterpart (denoted as black arrows in Figure 2) in the convolutional network. The integrated feature maps allow us to fully explore multiple scale context cues. Notice the feature maps within deeper convolutional layers have to be expanded with different upsampled ratio, resulting in feature representation of equal dimension for stacking. Finally, the concatenated feature maps, carrying both local and global context, are fed into two 3×3 convolutions, each followed by a ReLU. At the final layer, a 1×1 convolution is used to map each 64-component feature vector to the desired number of classes.

2.2. Training DDN

In this section, we first introduce batch normalization, which is widely used for network training. Then a two-stage scheme is adopted to train our network since our DDN is very deep (nearly twice deeper than [1, 2]).

Batch Normalization. According to [19], it is very hard to train a deep neural network due to the internal-covariate-shift problem. Since the parameters of previous layers have been updated, the distributions of filter responses in current layer change in the process of iterative training. This is not beneficial for optimizing our DDN since such changes may be amplified through back propagation across layers. As a result, a batch normalization layer is added to the output of every layer, where the filter responses are normalized to a standard Gaussian distribution, whether in convolutional or deconvolutional network. In the experiments, we observe that the batch normalization is critical to optimize our network, which helps our training algorithm to straggle from poor local optimum.

Two-stage training. Although batch normalization helps us to escape from local optima, the solution space for semantic segmentation is still very large, leading to the disadvantage that the benefit of our DDN might be cancelled. Therefore, we employ a two-stage training scheme to address this issue. In the first stage, we initialize the convolutional network with the weights pre-trained on ILSVRC dataset [20], and the weights in the deconvolutional network with zero-mean Gaussians. In the second stage, we fine-tune the trained network based on PASCAL VOC dataset [17]. In our DNN, a dense pixel-wise soft-max is adopted as objective function to evaluate segmentation estimation with respect to the associated ground truth:

$$p_k(\boldsymbol{x}) = \frac{\exp\{a_k(\boldsymbol{x})\}}{\sum_{k'}^{K} \exp\{a_{k'}(\boldsymbol{x})\}} \tag{1}$$

where K is the total number of object categories, and $a_k(\boldsymbol{x})$ denotes the activation for k^{th} category at the pixel position $\boldsymbol{x}$. Note $p_k(\boldsymbol{x})$ is a approximated maximum function, where $p_k(\boldsymbol{x}) \approx 1$ for the category that has the maximum activation $a_k(\boldsymbol{x})$, and $p_k(\boldsymbol{x}) \approx 0$ for the remaining classes.

3. EXPERIMENTS

3.1. Implementation Details

Dataset. We evaluate our DDN on PASCAL VOC 2012 dataset [17], which is a very popular benchmark for semantic segmentation. This dataset contains 21 object categories (20 foreground categories and one additional background class). The original dataset includes 1,464 (train), 1,449 (val), and 1,456 (test) images for training, validation, and testing, respectively, where the images of training and validation set have per pixel-level annotations. For training, we use the extra augmented segmentation annotations from [21], which includes 10582 training and validation images. The remaining 1456 test images are used to evaluate the performance of our DDN. Following [1, 4, 10], the performance is measured in terms of mean pixel intersection-over-union (mIOU) averaged across all 21 categories. Note that only the augmented images are used to train our DDN, the performance can be further improved using Microsoft COCO dataset as well as some state-of-the-art approaches [22, 23] do.

Baselines. To show the advantages of our approach, we selected 4 state-of-the-art models as baselines. Experimental results of some baseline models are produced using default parameter settings given by the authors, while others are directly taken from the literature. All the baselines are divided into two categories: (1) FCN-based models, including FCN-8s [1] and DeepLab [2]; (2) EDN-based models, including LDN [10] and SegNet [9].

Parameter settings. The entire DDN is implemented based on Caffe framework [24]. The input images and the corresponding pixel-wised annotated ground truth are used to train the network using the stochastic gradient descent algorithm [25]. In order to make full use of the GPU memory, we favor a large batch size (set as 14) to train batch normalization parameters, where initial learning rate, momentum and weight decay are set to 0.001, 0.99 and 0.0005, respectively.

Learning rate policy. Following [2, 26], we employ a "poly" learning rate policy where the initial learning rate is multiplied by $(1 - \frac{iter}{max_iter})^{power}$ with $power = 0.9$.

3.2. Evaluation Results on Pascal VOC

In Table 1, we compare our results on the testing set with previous works. Compared to FCN-based [1, 2] and EDN-based [9, 10] architecture, our DDN achieves best performance with 74.4% mIOU accuracy, and best scores on 14 out of the 20 classes. It is intriguing that our approach is superior to the existing methods [2, 10] that employ CRF as post-processing to further improve performance. This indicates our DDN is able to capture wide scale context information.

Figure 3 shows the qualitative results on the PASCAL VOC 2012 validation set. It is evident that, compared with baseline models, our method produces more detailed segmentation outputs with accurate object shapes and boundaries,

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:58:02 UTC from IEEE Xplore. Restrictions apply.

Table 1. Individual category results on the PASCAL VOC 2012 test set in terms of mIOU scores. The bold number indicates the best performance among all approaches for each category.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIOU
FCN-8s [1]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab [2]	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	**59.7**	82.2	50.4	73.1	63.7	71.6
SegNet [9]	74.5	30.6	61.4	50.8	49.8	76.2	64.3	69.7	23.8	60.8	54.7	62.0	66.4	70.2	74.1	37.5	63.7	40.6	67.8	53.0	59.1
LDN [10]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	**86.1**	28.5	77.0	62.0	79.0	**80.3**	83.6	80.2	58.8	**83.4**	**54.3**	**80.7**	65.0	72.5
Ours	**90.1**	**55.0**	**88.4**	**68.1**	**69.4**	**88.0**	**82.1**	84.8	**32.3**	**78.2**	**64.1**	**80.9**	79.3	**86.1**	**81.5**	58.3	82.1	53.2	77.1	**69.8**	**74.4**

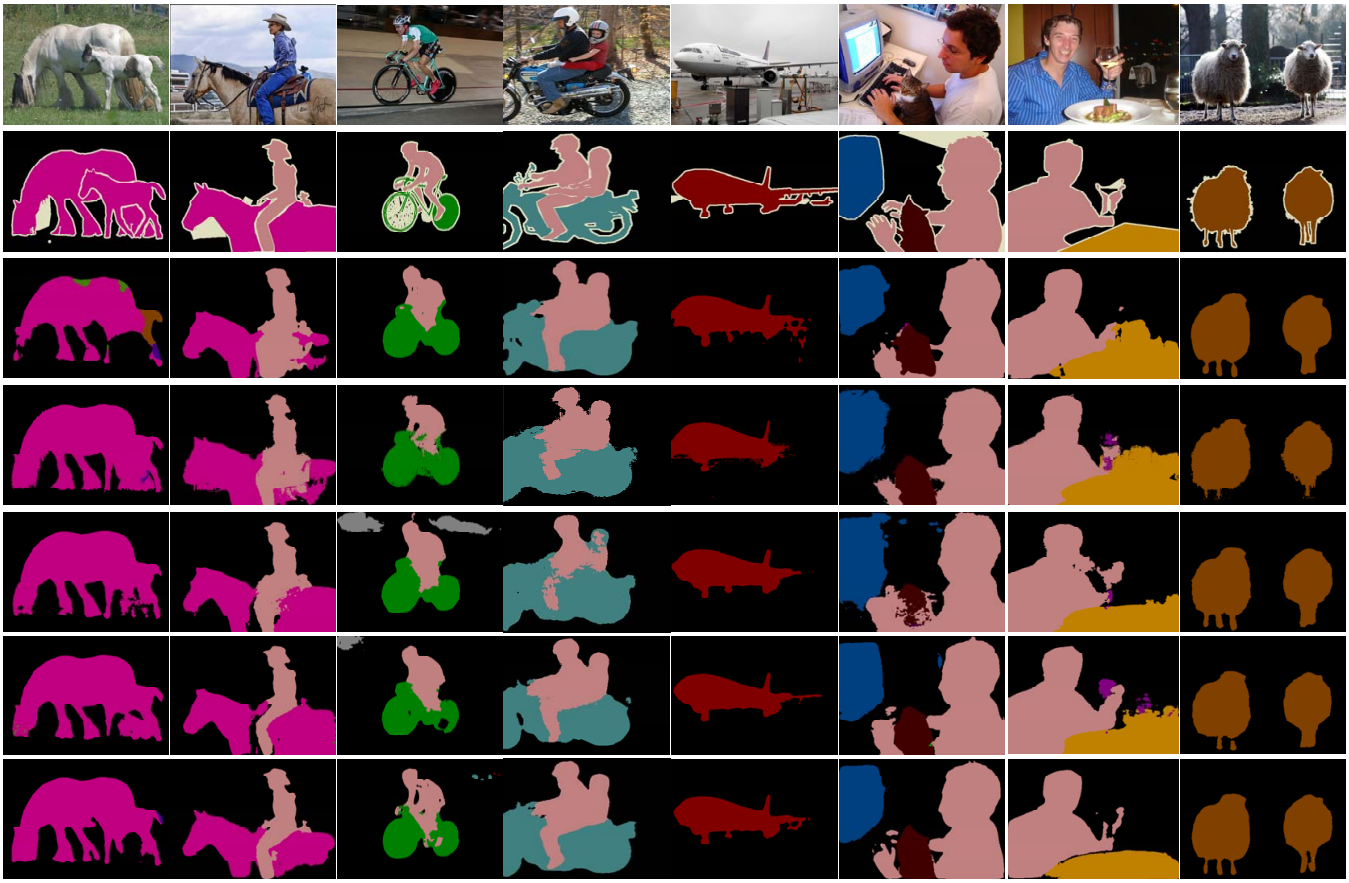

Fig. 3. The visual comparison on PASCAL VOC 2012 val dataset. From top to bottom are original images, the corresponding ground truth, segmentation outputs from FCN-8s [1], DeepLab [2], SegNet [9], LDN [10], and our DDN. (Best viewed in color)

and efficiently prohibits the effect from clutter background.

4. CONCLUSION AND FUTURE WORK

This paper describes a DDN model, which explores multi-scale context information for semantic segmentation. Through constructing dense connections from convolutional network to deconvolutional network, our DDN provides a more powerful representation that combines feature maps with different receptive fields, allowing us to fully investigate local and global context cues. The experimental results show that our DDN outperforms recent FCN and EDN-based state-of-the-art networks, and demonstrate that our approach can produce more accurate predictions and delineated segmentation maps on PASCAL VOC 2012 semantic segmentation dataset.

In the future, we are interested in extending our DDN model to perform semantic segmentation in spatio-temporal domain (e.g., video sequence).

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:58:02 UTC from IEEE Xplore. Restrictions apply.

5. REFERENCES

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," *IEEE TPAMI*, vol. 39, no. 4, pp. 640–651, 2017.

[2] Chen Liang-Chieh, Papandreou George, Kokkinos Iasonas, Murphy Kevin, and Yuille Alan, L., "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE TPAMI*, 2018.

[3] Zhao H., Shi J., Qi X., Wang X., and Jia J., Y., "Pyramid scene parsing network," in *CVPR*, 2016, pp. 6230–6239.

[4] Chen Liang-Chieh, Papandreou George, Schroff F., and Adam H., "Rethinking atrous convolution for semantic image segmentation," in *CVPR*, 2017.

[5] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *IJCV*, vol. 81, no. 1, pp. 2–23, 2009.

[6] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun, "Learning hierarchical features for scene labeling," *IEEE TPAMI*, vol. 35, no. 8, pp. 1915–1929, 2013.

[7] Quan Zhou, Baoyu Zheng, Weiping Zhu, and Longin Jan Latecki, "Multi-scale context for scene labeling via flexible segmentation graph," *PR*, vol. 59, pp. 312–324, 2016.

[8] Quan Zhou, Jun Zhu, and Wenyu Liu, "Learning dynamic hybrid markov random field for image labeling," *IEEE TIP*, vol. 22, no. 6, pp. 2219–2232, 2013.

[9] Vijay Badrinarayanan, Kendall Alex, and Cipolla Roberto, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE TPAMI*, vol. 39, no. 12, pp. 2481–2495, 2017.

[10] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015, pp. 1520–1528.

[11] Ronneberger Olaf., Fischer Philipp., and Brox Thomas., "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 225–233.

[12] Ghiasi G. and Fowlkes C., C., "Laplacian reconstruction and refinement for semantic segmentation," *arXiv preprint arXiv:1605.02264*, 2016.

[13] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, 2017, pp. 5168–5177.

[14] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun, "Large kernel matters: Improve semantic segmentation by global convolutional network," in *CVPR*, 2017, pp. 1743–1751.

[15] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *CVPR*, 2017, pp. 3309–3318.

[16] M. A. Islam, M. Rochan, N. D. B. Bruce, and Y. Wang, "Gated feedback refinement network for dense image labeling," in *CVPR*, 2017, pp. 4877–4885.

[17] Everingham Mark, Ali Eslami S., M., Gool Luc, Van, Christopher. K. I. Williams, Winn John, and Zisserman Andrew, "The pascal visual object classes challenge: A retrospective," *IJCV*, vol. 111, no. 1, pp. 98–136, 2015.

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223.

[19] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[21] Bharath Hariharan, Pablo Arbelez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011, pp. 991–998.

[22] Jifeng Dai, Kaiming He, and Jian Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *ICCV*, 2011, pp. 1635–1643.

[23] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan L. Yuille, "Weakly-and semi-supervised learning of a dcnn for semantic image segmentation," in *ICCV*, 2015, pp. 1742–1750.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACMMM*, 2014, pp. 675–678.

[25] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*, 2010, pp. 177–186.

[26] W. B. Yang, Q. Zhou, Y. W. Fan, G. W. Gao, S. S. Wu, W. H. Ou, H. M. Lu, J. Cheng, and L. J. Latecki, "Deep context convolutional neural networks for semantic segmentation," in *CCCV*, 2017, pp. 696–704.

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:58:02 UTC from IEEE Xplore. Restrictions apply.