DNS: A multi-scale deconvolution semantic segmentation network for joint detection and segmentation

Ning Feng[1], *Le* Dong[1,*], *Qianni* Zhang[2], *Ning* Zhang[3], *Xi* Wu[4] and *Jianwen* Chen[5]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
[2]School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, E1 4NS, UK
[3]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
[4]School of Computer Science, Chengdu University of Information Technology, Chengdu, 610103, China
[5]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

Abstract. Real-time semantic segmentation has become crucial in many applications such as medical image analysis and autonomous driving. In this paper, we introduce a single semantic segmentation network, called DNS, for joint object detection and segmentation task. We take advantage of multi-scale deconvolution mechanism to perform real time computations. To this goal, down-scale and up-scale streams are utilized to combine the multi-scale features for the final detection and segmentation task. By using the proposed DNS, not only the tradeoff between accuracy and cost but also the balance of detection and segmentation performance are settled. Experimental results for PASCAL VOC datasets show competitive performance for joint object detection and segmentation task.

1 Introduction

Semantic segmentation is a technique to assign semantic or object-class labels to individual pixels in images [1]. It is usually converted into pixel-wise classification problem, and focuses on the connection of semantics and location. The natural step in semantic segmentation is utilizing global information to resolve what while utilizing local information to resolve where [2].

Despite the attention it has received, global context and interplay between labelling and detection of object instance are still the restricted factors of semantic segmentation. Before the Simultaneous Detection and Segmentation (SDS) [3] were proposed, the detection and segmentation tasks are usually treated as individual one. Especially, most of the existing semantic segmentation approaches focus on single inference of net-design for higher accuracy, which are difficult to extend to incorporate other types of tasks. The effectiveness

[*] Corresponding author: ledong@uestc.edu.cn

of these convolutional nets largely depends on the sophisticated model design regarding depth and width, which has to involve many operations and parameters.

For the remarkable progress of recent deep convolutional neural nets, we in this paper resort to DCNN method similarly to manipulate semantic segmentation task. Besides, the problem of using a single network to handle multiple tasks has been repeatedly pursued in the stage of deep learning. In [4] the DCNN is used for recognition, localization and detection, while in [5] DCNN is trained for surface normal estimation, depth estimation and semantic segmentation, and [6] for joint detection, pose estimation and region proposal generation. Importantly, unlike the aforementioned holistic approaches, we are interested in exploiting semantic segmentation in order to improve both object detection and segmentation performance. Here, we take advantage of deconvolution mode with the sharing weight to combine these two tasks. Towards this goal, our convolutional network is trained under the supervision of bounding boxes and segmentation maps.

Meanwhile, real-time semantic segmentation has become crucial in many practical applications and brought with fundamental difficulty of reducing computation for pixel-wise label inference. As the prevalent image detection and segmentation pipelines, the approach remains expensive and relies on a region-based strategy that makes the network architecture inappropriate for semantic segmentation. Learning and inference in our model are efficient as we reason at the detection and segment level. We extensively evaluate the proposed model called ‘DNS’ on the PASCAL detection and segmentation benchmarks. What's more, the proposed method shows a graceful degradation compared with its counterpart.

Our main contributions are summarized below:

1. We focus on building a deep model for joint detection and semantic segmentation with a decent speed. To work out this problem, we introduce a multi-scale deconvolution mechanism which is a direct mode to perform easily. Specifically, we learn a multi-layer deconvolution network, which is composed of down-scale and up-scale stream. In this stream, we combine the multi-scale features instead of using the multi-scale inputs which has been demonstrated that outperforms average- and max-pooling, and can achieve excellent performance.

2. The trained ‘DNS’ network makes it possible to train a single net for multiple task (detection and segmentation). We achieved competitive advantages in PASCAL VOC benchmark. In addition to the trade-off between accuracy and inference cost, you will find that our DNS trained only on the PASCAL VOC dataset settle the balance of detection and segmentation performance.

The paper is organized as follows: Section 2 discusses related work; Section 3 presents our real-time multi-task framework. Finally, we devote to our experiments in Section 4, and Section 5 concludes the paper.

2 Related work

Before we introduce our approach, we now present techniques for both detection and semantic segmentation.

Representative methods [7-11] consider semantic segmentation task as simultaneous detection and segmentation (SDS), which in introduced in [7]. Semantic segmentation has recently witnessed rapid progress, but many leading methods are unable to identify object instances. To encourage the research on this problem, a Multi-task Network Cascades (MNCs) [11] for instance-aware semantic segmentation is proposed. This model consists of three networks, respectively differentiating instances, estimating masks, and categorizing objects. Although Multi-scale CNNs and their variants have made striking success for modelling the global scene structure for an image, they are limited in labelling fine-grained local structures like pixels and patches, since spatial contexts might be blindly mixed up without customizing their scales. Convolutional Feature Masking [10], connected Markov random field models [8], and Mask R-CNN [9] are also designed to address the issue of contexts of object labels.

In recent years, neural networks are driving advances in semantic segmentation, in which each pixel is labelled with the class of its enclosing region. Most of the convolutional versions of existing networks obtain precise segmentation from fixed-sized inputs in a particular dataset. These works [12, 3, 13], bring together DCNN methods and traditional computer vision algorithms for addressing pixel-wise segmentation problem.

Through the use of contextual information, 'deep CRFs' [14] is proposed to improve semantic segmentation, by combining the strengths of deep CNNs to learn powerful feature representations, with Conditional Random Fields (CRFs) which can capture contextual relation modelling. This method avoids repeated inference, and so is computationally tractable.

Incorporating multi-scale features in fully convolutional neural networks (FCNs) [2] has been a key element to achieving state-of-the-art performance on semantic segmentation. One common way to extract multi-scale features is to feed multiple resized input images to a shared deep network and then merge the resulting features for pixel-wise classification.
FCNs uses large receptive field and many pooling layers, both of which cause blurring and low spatial resolution in the deep layers. As a result, FCNs tends to produce segmentations that are poorly localized around object boundaries. Using a color-based CRF on top the FCN prediction [13] is one way that attempts to address this issue in post-processing steps.
Although post-processing the output of FCN with a fully-connected CRF can increase segmentation accuracy near object boundaries, mean-field inference in fully-connected CRF model is expensive in terms of both memory and CPU time. To this end, a task-specific edge detection model [15] using CNNs and a discriminatively trained domain transform is proposed. This domain transform can equivalently be seen as a recurrent neural network (RNN), and it is a special case of the recently proposed RNN with gated recurrent units.

In addition, using CRF on FCN require additional parameters and low-level features that are difficult to tune and integrate into the original network architecture. To overcome these problem, a Boundary Neural Field (BNF) [16] is proposed. It is a global energy model integrating FCN predictions with boundary cues. Further, some steer DNN architectures, like decoupled DNN [1], Deconvolution net [17] are designed to make precise per-pixel label prediction tasks.

While the discrete CRF is a natural _t for labelling tasks of semantic segmentation, a new end-to-end trainable deep network, referred to as Gaussian Mean Field (GMF) network [18], whose layers perform mean field inference over a Gaussian CRF, is proposed. The Gaussian CRF is composed of three sub-networks: a CNN-based unary network for generating unary potentials, a CNN-based pairwise network for generating pairwise potentials, and a GMF network for performing Gaussian CRF inference. This method outperforms various recent semantic segmentation approaches that combine CNNs with CRF models.

Meanwhile, some similar works set out to deal with both detection and semantic segmentation jointly [19-22]. Yao and Fidler [19] propose a traditional approach with high-order potentials to holistic scene understanding that reasons jointly about regions, location, class and spatial extent of objects. Fidler and Mottaghi [20] focus on how semantic segmentation can help object detection, and their model blends between the detector and the segmentation model, by boosting object hypotheses on the segments. Both of these two method neglect the strength of DCNN. Teichmann and Weber [21] introduce an approach (MultiNet) to joint classification, detection and semantic segmentation via a unified CNN architecture where the encoder is shared amongst the three tasks. However, the MultiNet trained and evaluated on KITTI dataset is limited, and mainly designed for autonomous driving. Kokkinos [22] introduce a convolutional neural network (CNN) that jointly handles low-, mid-, and high-level vision tasks in a unified architecture that is trained end-to-end. It is necessary to point out that UberNet initializes from a network that was trained with M-SCOCO data, which needs more dataset than our DNS.

3 Detection and semantic segmentation with DNS

In this paper, we propose an efficient and effective semantic segmentation architecture, called DNS, to jointly reason about object detection and semantic segmentation. Figure 1 presents DNS architecture. In this DNS, we add convolutional feature layers to the end of the truncated base net. These layers decrease in size progressively and allow predictions of detections at multiple scales. For object detection task, it is performed by a single convolutional layer that predicts the class and the coordinates of bounding box in the feature maps of the upscale stream. Similarly, we in the segmentation task upscale all the activations of the upscale stream and concatenate them to predict the pixel labels and produce segmentation maps.

In figure 1, the trained DNS is composed of two parts i.e. Convolution and Deconvolution networks. Firstly, the input image is pre-processed by a convolution network to produce a map with high-level features. We employ ResNet-50 or VGG-16 base net for convolutional part. Taking VGG-16 as the example, the convolution network discards the fully-connected softmax layers of VGG-16. We call this layer Conv5-3, following the deconvolution part. This part consists of Down-Scale and Up-Scale stream.

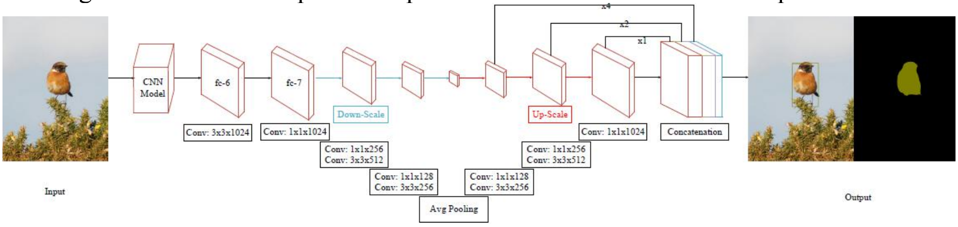

Fig. 1. The DNS architecture, which performs semantic segmentation with fully convolutional network. We adopt deconvolution layers to build the segmentation maps. Multi-upscale feature maps are utilized to make pixel prediction.

In the Down-Scale stream, we respectively use 3 × 3 and 1 × 1 convolutions to get the so-called 'fc-6' and 'fc-7' layer. Given the features produced by fc-7, we employ the similar block in each which includes the 1 × 1 and 3 × 3 convolutional layer as discussed in [23] followed by a pooling layer to produce more precise prediction. With three convolution block, we in the Up-Scale stream apply the same deconvolution pattern in [24] to skip the connections. This skip connection in our DNS is used likewise to prevent the gradient from affecting the backbone network too violently and ensure the stability of the network. With three deconvolution, the feature maps are concatenated in order to predict subsequently precise object masks and segmentation maps.

In the training stage, given training data annotated with bounding boxes and segmentation maps, we design the loss function which is simply the sum of two loss functions of these two task. Our training objective is expressed as:

$$\delta(w_0,det,seg) = \mathrm{B}_{base}(w_0) + L(w_0,w_{det},w_{seg}) \tag{1}$$

In equation (1), we use det and seg to the index two task; w_0 denotes the weights of the base net B_{base}, w_{det} and w_{seg} are task specific weights; B_{base} is the loss function of the base

CNN model(ResNet-50 or VGG-16); $L(w_0,w_{det},w_{seg})$ is the task specific loss function. This task-specific loss is written as follows:

$$L\left(w_0,w_{det},w_{seg}\right) = \frac{1}{N}\sum_{i-1}^{N}\left(L\left(f_{det}^{i}(w_0,w_{det}),y_{det}^{i}\right) + L\left(f_{seg}^{i}\left(w_0,w_{seg}\right),y_{seg}^{i}\right)\right) \qquad (2)$$

where we use i to index training samples, denote by $f_{det,seg}^{i}$, $y_{det,seg}^{i}$ the task-specific network prediction and ground truth at the i-th example respectively, by $w_{det,seg}$ the task-specific network parameters.

To implement the detection operation, we follow the similar approach proposed in [23]. The objective function of detection task is to minimize error between ground-truth bounding boxes and the input image with anchor boxes. For segmentation task, the loss function is the cross-entropy between predicted and target class distribution of pixels. Specifically, we use a 1×1convolutional operation with 64 channels to map each layer of the upscale-stream to an intermediate representation. After this, each layer is up-scaled to the size of the last layer using bilinear interpolation and all maps are concatenated. This representation is mapped to c feature maps, where c is the number of classes, by using 3×3 convolutions to predict posterior class probabilities.

4 Experiments

We now present various experiments conducted on the Pascal VOC 2007 and 2012 datasets, for which both bounding box annotations and segmentation maps are available. Section 4.1 presents the datasets and the metrics in more details; Section 4.2 presents technical details which is important to make our work reproducible. The last section discusses the inference speed in the network architecture.

Our experiment has two objectives. The first one is to explore how DNS architecture addresses the two individual tasks. The second objective is how to settle the balance of detection and segmentation task in one single net (DNS). In order to examine this, we compare to the results in three aspects. Firstly, we contrast with the prevalent models only for the single detection task. Then the primary models merely used in segmentation task are compared. Finally, we contrast five representative approaches designed for joint detection and segmentation task.

4.1 Experimental setup

Datasets and Metrics: We use the PASCAL VOC07, and VOC12 datasets. All images in the VOC datasets are annotated with ground truth bounding boxes of objects. Both VOC07 and VOC12 consist of 20 foreground object classes and one background class. The VOC07 dataset is divided into 2 subsets, train-val (5011 images) and test (4952 images). The PASCAL VOC12-train subset contains 5717 images annotated for detection and 1464 of them have segmentation ground truth as well, while VOC12-val has 5823 images for detection and 1449 images for segmentation. We train our DNS on different subsets which consist of 'voc07-trainval-seg', 'voc12-train-seg', 'voc12-val', and 'voc12-val-seg'. In these four subsets, 'voc07-trainval-seg' subset includes 5011 segmentation images in PASCAL VOC07, while 'voc12-train-seg' 1464 segmentation images, 'voc12-val' 5823 images, and 'voc12-val-seg' 1449 segmentation images in PASCAL VOC12.

Optimization: Our DNS is coded in Python and TensorFlow. The experiments were conducted on a Tesla K40c GPU with 11439M memory. In all experiments, we use the Adam algorithm instead of SGD, with a mini-batch size of 32 images. The initial learning rate is set to 10^{-4} and decreased twice during training by a factor 10. We also use a weight decay parameter of 5×10^{-4}. As already mentioned, we use ResNet-50 as a feature extractor, 512 feature maps for each layer in down-scale and up-scale streams, 64 channels for intermediate representations in the segmentation branches. We evaluate our proposed methods on the PASCAL VOC 2007 and 2012 test set. We also compare our test set results with other competing methods.

4.2 Experimental evaluation

4.2.1 Individual object detection.

We start by verifying that diverse training PASCAL VOC dataset make much difference in our DNS. In the experiment, the default image size used to train our DNS is 300×300. The comparison of detection performance on different training PASCAL VOC datasets are reported in the Table 1. Table 2 and 3 respectively present the comparison of detection accuracy between DNS and state-of-the-art models on PASCAL VOC07 and VOC12.

Table 1. A Comparison of detection performance on Pascal VOC 2007 and 2012 test set. The model was trained on different VOC training datasets.

Network trained on different subsets	Base net	07mAP	12mAP
'voc07-trainval-seg', 'voc12-train-seg', 'voc12-val-seg'	ResNet-50	60.1	65.8
'voc07-trainval-seg', 'voc12-train-seg'	ResNet-50	62.8	68.4
'voc12-train-seg', 'voc12-val'	ResNet-50	64.2	70.6
'voc07-trainval-seg', 'voc12-train-seg', 'voc12-val'	**ResNet-50**	**69.0**	**73.3**
'voc07-trainval-seg', 'voc12-train-seg', 'voc12-val-seg'	VGG-16	63.4	67.6
'voc07-trainval-seg', 'voc12-train-seg'	VGG-16	65.9	69.4
'voc12-train-seg', 'voc12-val'	VGG-16	66.8	70.9
'voc07-trainval-seg', 'voc12-train-seg', 'voc12-val'	VGG-16	68.2	72.3

As shown in Table 1, our DNS result achieves the best results on the joint subset of voc07-trainval-seg, voc12-train-seg, and voc12-val, with 69.0 on 07mAP column and 73.3 on 12mAP column respectively. Note that the result on voc07-trainval-seg, voc12-train-seg, and voc12-val subset of ResNet-50 base net is better than each subset of VGG-16 base net. However, all the rest subset results on 07mAP and 12mAP column of base net of VGG-16 are better than that of ResNet-50.

Table 2. Comparison of detection performance on Pascal VOC 2007 test set. The models where trained on 'voc07-trainval-seg', 'voc12-train-seg', and 'voc12-val'.

Network	DNS	R-FCN[25]	Faster RCNN[26]	YOLO[27]	SSD[23]
Base net	ResNet-50	ResNet-101	ResNet-101	YOLO net	VGG-16
mAP	**69.0**	80.5	76.4	63.4	68.0
Aero	72.1	79.9	79.8		73.4
Bike	75.3	87.2	80.7		77.5
Bird	65.9	81.5	76.2		64.1
Boat	61.4	72.0	68.3		59.0
Bottle	30.6	69.8	55.9		38.9
Bus	77.7	86.8	85.1		75.2
Car	78.0	88.5	85.3		80.8
Cat	84.2	89.8	89.8		78.5
Chair	48.8	67.0	56.7		46.0
Cow	75.4	88.1	87.8		67.8

Table	67.0	74.5	69.4		69.2
Dog	77.5	89.8	88.3		76.6
Horse	81.6	90.6	88.9		82.1
Mbike	75.4	79.9	80.9		77.0
Person	71.2	81.2	78.4		72.5
Plant	41.0	53.7	41.7		41.2
Sheep	71.9	81.8	78.6		64.2
Sofa	70.8	81.5	79.8		69.1
Train	82.2	85.9	85.3		78.0
Tv	71.5	79.9	72.0		68.5

In Table 2, the results show that detection performance of DNS outperforms SSD and YOLO with a 69.0 mAP, while being a real time detector. Our DNS performs 5.6% better than YOLO and 1.0% than SSD for detection on Pascal VOC 2007 test set. We further improve the results by training for detection achieving 73.3 mAP on Pascal VOC 2012 test dataset in Table 3.

Table 3. Comparison of detection performance on Pascal VOC 2012 test set. The models where trained on 'voc07-trainval-seg', 'voc12-train-seg', and 'voc12-val'.

Network	DNS	R-FCN[25]	Faster RCNN[26]	YOLO[27]	SSD[23]
Base net	ResNet-50	ResNet-101	ResNet-101	YOLO net	VGG-16
mAP	**73.3**	77.6	73.8	57.9	72.4
Aero	82.6	86.9	86.5	77.0	85.6
Bike	80.4	83.4	81.6	67.2	80.1
Bird	75.8	81.5	77.2	57.7	70.5
Boat	55.9	63.8	58.0	38.3	57.6
Bottle	49.4	62.4	51.0	22.7	46.2
Bus	72.8	81.6	78.6	68.3	79.4
Car	77.1	81.1	76.6	55.9	76.1
Cat	90.5	93.1	93.2	81.4	89.2
Chair	57.2	58.0	48.6	36.2	53.0
Cow	79.6	83.8	80.4	60.8	77.0
Table	59.7	60.8	59.0	48.5	60.8
Dog	88.9	92.7	92.1	77.2	87.0
Horse	83.6	86.0	85.3	72.3	83.1
Mbike	82.0	84.6	84.8	71.3	82.3
Person	78.2	84.4	80.7	63.5	79.4
Plant	54.5	59.0	48.1	28.9	45.9
Sheep	76.5	80.8	77.3	52.2	75.9
Sofa	69.0	68.6	66.5	54.8	69.5
Train	83.7	86.1	84.7	73.9	81.9
Tv	68.4	72.9	65.6	50.8	67.5

As Table 3 shows, detection on VOC12 improves by more than 4.3% on VOC07. Meanwhile, our detection results are similar to Faster RCNN on VOC12, which is also better than YOLO and SSD. We argue than this result could be competitive even though it is still 3.3% less than R-FCN.

4.2.2 Individual semantic segmentation

The second task that we have tried is semantic segmentation. Even though a broad range of techniques designed for this problem, we compare to the state-of-the-art methods. Table 4 presents the comparison of different segmentation evaluation results on PASCAL VOC12 test.

Table 4. Semantic segmentation evaluation results on PASCAL VOC 2012 test set.

Method	Mean IOU(%)
SDS[7]	51.6
FCN-8s[2]	62.2
DeconvNet[17]	69.6
DeepLab-v2[28]	79.7
PSPNet[29]	82.6
RefineNet[30]	83.4
DeepLab-v3[31]	85.7
DNS	**69.8**

From Table 4, we can see that our DNS yields 69.8 on the mean IOU metric evaluation, which improves over SDS (the primogenitor of simultaneous detection and segmentation) by a large 18.2% margin, while improving 7.6% with respect to FCN-8s. The performance of DeconvNet is competitive to our DNS, about 0.2% less than ours. Note that RefineNet and DeepLab-v3 are the most representative works in semantic segmentation which focus on segmentation thus neglecting detection, which is also the goal of our pursuit.

4.2.3 Joint detection and semantic segmentation

Motivated by the empirical results in the previous paragraphs, we have explored the ability of how DNS architecture addresses the two individual tasks. Now for the joint task, we contrast to five representative methods, which similarly make use of a single net to address joint detection and segmentation (final goal) in Table 5. We also shows the visualization results of detection and segmentation on PASCAL VOC 2007 test images in figure 2.

From table 5, we can observe that very little of works address this two task jointly. Among the rest method which conduct on PASCAL VOC Dataset, our DNS is largely better than SDS, Holistic Scene Understanding and segDPM method. What's important is that our DNS achieves second good performance for joint detection and segmentation task.

Table 5. Comparison of Jointly Detection and Segmentation Evaluation.

Method	Dataset	Mean mAP (%)	Mean IOU (%)
SDS[7]	VOC12Det+Seg	53.9	51.6
HSU[19]	MSRC-21Det+VOC10 Seg	49.3	31.2
segDPM[20]	VOC10Det+Seg	61.4	34.8
DAG[32]	VOC07Det+VOC10Seg	67.09	72.07
UberNet[22]	VOC07Det+VOC12Seg	78.8	71.1
DNS	VOC12Det+Seg	73.3	69.8

Compared with DAG which is a little bit less than our performance, the detection performance of DNS achieves 73.3 with about 6.2 mAP growth, while segmentation performance 69.8 with about 2.3 mean IOU decline. The main reason is that the object detection architecture of DAG is based on the Faster-RCNN, which is a little bit better than ours, and the semantic segmentation architecture of DAG is based on FCN, that is a little bit worse than ours. Especially the accuracy of DAG drops (both mAP and mean IOU) significantly after the adversarial perturbations are added.

Compared with UberNet which achieves the best performance, in conjunction with Faster-RCNN, UberNet makes use of the VOC 2007 dataset to fine-tune the MS-COCO pretrained network for detection task. For segmentation task, UberNet deviates from the Deeplab-FOV architecture by using linear operations on top of skip layers to reach the similar result of DeepLab-v2. In spite of this, the mAP accuracy of our DNS is about 5.5 under UberNet, while the mean IOU accuracy of DNS just 1.3 under UberNet.

4.3 Speed comparison

To strike the balance between accuracy and inference cost, we report speed comparison to other state-of-the-art pipelines in figure 3. Our approach is the most accurate among these five detectors working 24 frames per second (FPS) and in the setting close to real time (19 FPS), it can provide the real-time detections among the counterparts, while also providing semantic segmentation mask.

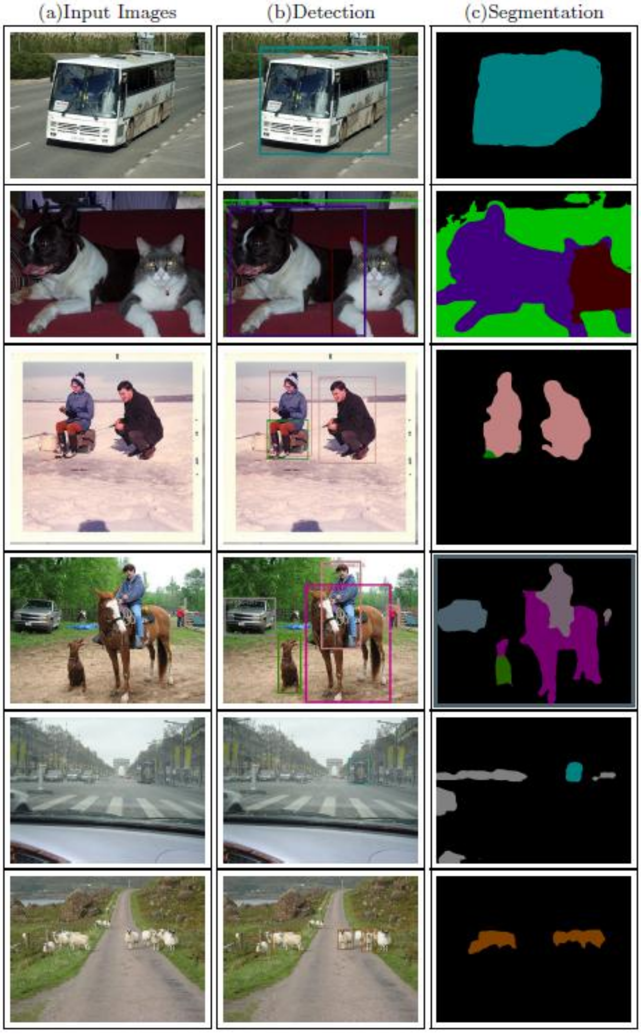

Fig. 2. Example results of several PASCAL VOC 2007 test images based on DNS models. (a) Input Images. (b) Detection results in VOC07. (c) Segmentation results in VOC07.

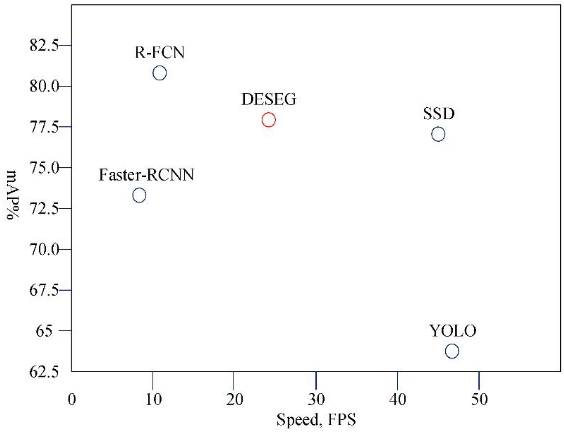

Fig. 3. Speed comparison. X-coordinate is their speed, in FPS. Y-axis is the detection accuracy of measured in mAP.

5 Conclusions

This paper adapts a deep deconvolution semantic segmentation model (DNS) to handle both detection and segmentation task. Experiments on PASCAL VOC dataset have shown that: (1) Our DNS network based on weight-sharing is advantageous to both detection and segmentation task. (2) Merging the down-scale and up-scale features not only improves the performance over deconvolution baselines, but also allows us to fast the detection speed. (3) Our network demonstrates the competitive performance in PASCAL VOC detection and segmentation benchmark.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61772114, in part by the Fundamental Research Funds for the Central Universities (No. ZYGX2016J077), and in part by the program of China Scholarships Council (No. 201606070055).

References

1. Hong S, Noh H and Han B 2015 Decoupled deep neural network for semi-supervised semantic segmentation *In: Advances in Neural Information Processing Systems*
2. Long J, Shelhamer E and Darrell T 2014 Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39** 640-51
3. Hariharan B, Arbelaez P and Girshick R 2015 Hypercolumns for object segmentation and _ne-grained localization *In: The IEEE Conference on Computer Vision and Pattern Recognition*
4. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R and LeCun Y 2014 Overfeat: Integrated recognition, localization and detection using convolutional networks *Preprint arXiv:1312.6229*

5. Eigen D and Fergus R 2015 Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture *In: The IEEE International Conference on Computer Vision*
6. Gkioxari G, Girshick R and Malik J 2015 Contextual action recognition with r*cnn *In: The IEEE International Conference on Computer Vision*
7. Hariharan B, Arbelaez P, Girshick R and Malik J 2015 Simultaneous detection and segmentation *In: European Conference on Computer Vision*
8. Zhang Z, Fidler S and Urtasun R 2016 Instance-level segmentation for autonomous driving with deep densely connected mrfs *In: The IEEE Conference on Computer Vision and Pattern Recognition*
9. He K, Gkioxari G, Dollr P and Girshick R 2017 Mask r-cnn *In: The IEEE International Conference on Computer Vision*
10. Dai J, He K and Sun J 2015 Convolutional feature masking for joint object and stuff segmentation *In: The IEEE Conference on Computer Vision and Pattern Recognition*
11. Dai J, He K and Sun J 2015 Convolutional feature masking for joint object and stuff segmentation *In: The IEEE Conference on Computer Vision and Pattern Recognition*
12. Mostajabi M, Yadollahpour P and Shakhnarovich G 2015 Convolutional feature masking for joint object and stuff segmentation *In: The IEEE Conference on Computer Vision and Pattern Recognition*
13. Chen C, Papandreou G, Kokkinos I, Murphy K and Yuille L 2015 Semantic image segmentation with deep convolutional nets and fully connected crfs. *In: International Conference on Learning Representations*
14. Lin G, Shen C, Reid I and Hengel A 2016 Efficient piecewise training of deep structured models for semantic segmentation *In: The IEEE Conference on Computer Vision and Pattern Recognition*
15. Chen C, Barron T, Papandreou G, Murphy K and Yuille L 2016 Semantic image segmentation with task-speci_c edge detection using cnns and a discriminatively trained domain transform *In: The IEEE Conference on Computer Vision and Pattern Recognition*
16. Bertasius G, Shi J and Torresani L 2016 Semantic segmentation with boundary neural fields *In: The IEEE Conference on Computer Vision and Pattern Recognition*
17. Noh H, Hong S and Han B 2015 Learning deconvolution network for semantic segmentation *In: The IEEE International Conference on Computer Vision*
18. Vemulapalli R, Tuzel O, Liu Y and Chellappa R 2016 Gaussian conditional random field network for semantic segmentation *In: The IEEE Conference on Computer Vision and Pattern Recognition*
19. Yao J, Fidler S and Urtasun R 2012 Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation *In: The IEEE Conference on Computer Vision and Pattern Recognition*
20. Fidler S, Mottaghi R, Yuille A and Urtasun R 2013 Bottom-up segmentation for top-down detection *In: The IEEE Conference on Computer Vision and Pattern Recognition*
21. Teichmann M, Weber M, Zoellner M, Cipolla R and Urtasun R 2016 Multinet: Real-time joint semantic reasoning for autonomous driving *Preprint arXiv: 1612.07695*
22. Kokkinos I 2017 Training a `universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory *In: The IEEE Conference on Computer Vision and Pattern Recognition*

23. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu Y and Berg C 2016 Ssd: Single shot multibox detector *In: European Conference on Computer Vision*
24. Fu Y, Liu W, Ranga A, Tyagi A and Berg C 2017 Dssd : Deconvolutional single shot detector *Preprint arXiv: 1701.06659*
25. Dai J, Li Y, He K and Sun J 2016 R-fcn: Object detection via region-based fully convolutional networks *In: Advances in Neural Information Processing Systems*
26. Ren S, He K, Girshick R and Sun J 2017 Faster r-cnn: Towards real-time object detection with region proposal networks *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39** 1137-49
27. Redmon J, Divvala S, Girshick R and Farhadi A 2016 You only look once: Unified, real-time object detection *In: The IEEE Conference on Computer Vision and Pattern Recognition*
28. Chen C, Papandreou G, Kokkinos I, Murphy K and Yuille L 2018 Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40** 834-48
29. Zhao H, Shi J, Qi X, Wang X and Jia J 2017 Pyramid scene parsing network *In: The IEEE Conference on Computer Vision and Pattern Recognition*
30. Lin G, Milan A, Shen C and Reid I 2017 Refinenet: Multi-path refinement networks for high-resolution semantic segmentation *In: The IEEE Conference on Computer Vision and Pattern Recognition*
31. Chen C, Papandreou G, Schroff F and Adam H 2017 Rethinking atrous convolution for semantic image segmentation *Preprint arXiv: 1706.05587*
32. Xie C, Wang J, Zhang Z, Zhou Y, Xie L and Yuille A 2017 Adversarial examples for semantic segmentation and object detection *In: The IEEE International Conference on Computer Vision*