Stacked Deconvolutional Network for Semantic Segmentation

Jun Fu, Jing Liu, *Member, IEEE*, Yuhang Wang, and Hanqing Lu, *Senior Member, IEEE*

Abstract—Recent progress in semantic segmentation has been driven by improving the spatial resolution under Fully Convolutional Networks (FCNs). To address this problem, we propose a Stacked Deconvolutional Network (SDN) for semantic segmentation. In SDN, multiple shallow deconvolutional networks, which are called as SDN units, are stacked one by one to integrate contextual information and guarantee the fine recovery of localization information. Meanwhile, inter-unit and intra-unit connections are designed to assist network training and enhance feature fusion since the connections improve the flow of information and gradient propagation throughout the network. Besides, hierarchical supervision is applied during the upsampling process of each SDN unit, which guarantees the discrimination of feature representations and benefits the network optimization. We carry out comprehensive experiments and achieve the new state-of-the-art results on three datasets, including PASCAL VOC 2012, CamVid, GATECH. In particular, our best model without CRF post-processing achieves an intersection-over-union score of 86.6% in the test set.

Index Terms—Semantic Segmentation, Deconvolutional Neural Network, Dense Connection, Hierarchical Supervision

✦

1 INTRODUCTION

SEMANTIC image segmentation has been one of the most important fields in computer vision, which is to predict the category of individual pixels in an image. Recently, Deep Convolutional Neural Networks (DCNNs) [1] have strong learning ability to obtain high-level semantic features, and make remarkable advances in computer vision, including image classification [2], [3], [4], object detection [5], [6] and keypoint prediction [7], [8]. For semantic segmentation tasks, DCNNs based methods mainly utilize the architecture of Full Convolutional Networks (FCNs) [9] which usually adopt a certain pretrained classification network and output a probability map per class for an arbitrary-sized input. However, the classification network with downsampling operations sacrifices the spatial resolution of feature maps to obtain the invariance to image transformations. The resolution reduction results in poor object delineation and small spurious regions in segmentation outputs.

Many approaches have been proposed to solve the above problems. One way is to apply dilated convolutions [10], [11], [12], [13]. This type of solutions are to upsample the filters with different dilation factors, while the number of filter parameters per position remains unchanged. Accordingly, the receptive fields are enlarged and the larger contextual information is captured without losing the spatial resolution. However those methods output a coarse subsampling feature maps which still lose details of object delineation. Moreover, many methods [11], [12], [14], [15] incorporate multi-scale or global feature maps to capture contextual information effectively and fit objects at multiple scales. Although this strategy further improves the receptive field and captures multi-scale information, it still outputs

low-resolution feature maps which impede the generation of detailed boundaries.

Another type of methods are to recover the spatial resolution by an upsampling or deconvolutional path [16], [17], [18], [19], which can generate high-resolution feature maps for dense prediction. By learning the upsampling process, the low resolution of the feature maps can be restored to the input resolution for pixel-wise classification, which is useful for accurate boundary localization. In the above upsampling solutions, the deconvolutional and unpooling layers are appended with a symmetric structure of the corresponding convolutional and pooling layers. Consequently, the parameter size of the new network is twice as large as the original convolutional structure. Furthermore, the deconvolutional networks are usually built by simply stacking layers, leading to the degradation problem when the depth increases [4]. To make the model easy to convergence, Wang et al. [16] use the VGG16 network [3] as pretrained weights to obtain better initial parameters of deconvolutional network, and Noh et al. [18] use a two-stage training strategy on single object images and multi-object images, respectively. Due to the difficulties on network optimization, neither of the previous networks can be extended directly to the deeper models, e.g., RestNet [4] and DenseNet [20], resulting in their limited learning ability.

To overcome the above issues, in this paper, we propose a Stacked Deconvolutional Network (SDN) for semantic image segmentation. SDN is a deeper deconvolutional network but easier to optimize compared with most deconvolutional solutions. Instead of building a single deep encoder-decoder network, we design an efficient shallow deconvolutional network (called as SDN unit), and stack multiple SDN units one by one with dense connections, thus making the proposed SDN capture more contextual information and easily optimized. The designing details of the proposed architecture are shown in Fig. 1. A SDN unit is an encoder-decoder network. The encoder module

• *J. Fu, J. Liu (corresponding author), Y. Wang and H. Lu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing 100190, China.*
E-mail: {jun.fu,jliu,yuhang.wang,luhq}@nlpr.ia.ac.cn

is operated as a downsampling process, aiming to exploit multi-scale features and capture contextual information as much as possible, while the decoder module is operated as an upsampling process to recover the spatial resolution, aiming for accurate boundary localization. To benefit from the impressive performance of the pretrained deep model on ImageNet [21], we adopt DenseNet-161 [20] without its last downsampling operations as the encoder module of the first SDN unit, while other encoder and decoder modules consist of some simple downsampling blocks and upsampling blocks, respectively. Typically, each downsampling (upsampling) block includes a max-pooling layer (deconvolutional layer), several convolutional layers, and a compression layer.

As the number of the stacked SDN units increases, the difficulty on model training becomes a major problem. Two strategies have been taken to ameliorate the situation. First, hierarchical supervision is added to upsampling blocks of each SDN unit. Specifically, the compression layers are mapped to pixel-wise labeling maps by a classification layer. With this structure, the network could be learned in a more refined way with no loss on discrimination ability. Second, we import intra-unit and inter-unit connections to help the network optimization. The connections are shortcut paths from early layers to later layers, and they are beneficial to the flow of information and gradient propagation throughout the network. With in an upsampling or a downsampling block of a given SDN unit, the intra-unit connections is a short-range dense connection which is the direct link from the inputs of previous convolutional layers to the ones of back convolutional layers. The inter-unit connection is a long-range skip connection between certain two SDN units. Considering the different intentions for information transmission, there are two forms of inter-unit connections. One is to link decoder and encoder modules between any two adjacent SDN units to promote the network optimization. The other is to connect the multi-scale feature maps from the encoder module of the first SDN unit to the corresponding decoder modules of each SDN unit, thus maintaining the low-level details for the high-resolution prediction. With the above two strategies, our proposed SDN can be trained efficiently and effectively, and achieves impressive performance on three popular benchmarks including PASCAL VOC 2012 dataset [22], CamVid dataset [23] and GATECH dataset [24]. In particular, a Mean IoU score of 86.6% without CRF post-processing on PASCAL VOC 2012 test set.

Our main contributions can be summarized as follows:

- We propose a novel network (Stacked Deconvolutional Network) for semantic segmentation, which stacks multiple shallow deconvolutional networks to capture multi-scale context, and guarantee the fine recovery of localization information.
- Our proposed SDN adopt intra-unit and inter-unit connections to enhance the flow of information and gradients throughout the network. In particular, inter-unit connections make the multi-scale information across different units efficient to reuse.
- The hierarchical supervision for each SDN unit results in better optimization of the proposed network, since early layers of the network can obtain more

gradient feedback. Besides, it guarantees the discrimination of the feature maps from the upsampling process of each unit.
- Extensive experimental evaluations demonstrate that the proposed SDN model achieves the new state-of-the-art performance on all the three benchmarks.

The remainder of this paper is organized as follows: In Section II, we review the related work about DCNNs based methods for semantic image segmentation and scan the basic architecture of DenseNet [20]. In Section III, we will introduce our proposed Stacked Deconvolutional Network, including the designing details of a single SDN unit, the connections stacking multiple SDN units and the multi-scale hierarchical supervision. To verify the effectiveness of our work, the experimental evaluations and necessary analysis are presented in Section IV. Finally, we summarize our work in Section V.

2 RELATED WORK

Recently, DCNNs based methods make great progress in semantic image segmentation, represented by FCNs [9]. FCNs apply a fully convolutional structure and bilinear interpolation to realize pixel-wise prediction, which results in rough edges and object vanishing. Following the fully convolutional structure, many works try to alleviate these problems from the following sides.

Some networks import dilated convolutions and reduce down-sampleing operations, which enable context aggregation and retain more spatial information. Deeplab [10] and DilatedNet [13] adopt dilated convolutions to enlarge receptive fields and capture larger contextual information without losing resolution. [25] employs hybrid dilated convolutions to further enlarge receptive fields of the network. Further on, Dai et al. [26] proposes a deformable convolution to adjust the receptive fields according to the scale of objects. The above works remove some downsampling operations for increasing the resolution of outputs, which helps to generate detailed object delineation.

Some networks explore multi-scale or global features to capture contextual information for performance improvement. ParseNet [14] employs global pooling operations to extract image-level information. Deeplabv2 [12] proposes atrous spatial pyramid pooling (ASPP) to embed contextual information, which consists of parallel dilated convolutions with different dilated rates. PSPNet [15] designs a pyramid pooling module to collect the effective contextual prior, containing information with different scales. Deeplabv3 [11] proposes an augment ASPP module with image-level features to further capture global context.

Besides, some networks employ deconvolution operations in upsampling path, and thus they are able to realize high resolution prediction and obtain refined object delineation. OA-Seg [16] and SegNet [17] apply unpooling operations to unsample the low-resolution features and learn deconvolutional layers to improve the upsampling process. U-Net [27] exploits multi-level features by skip connections in a deconvolutional network. RefineNet [28] further refines coarse semantic features with fine-grained low-level features in a multi-path refined architecture. In human pose estimation tasks, Newell et al. [8] stacks encoder-decoder
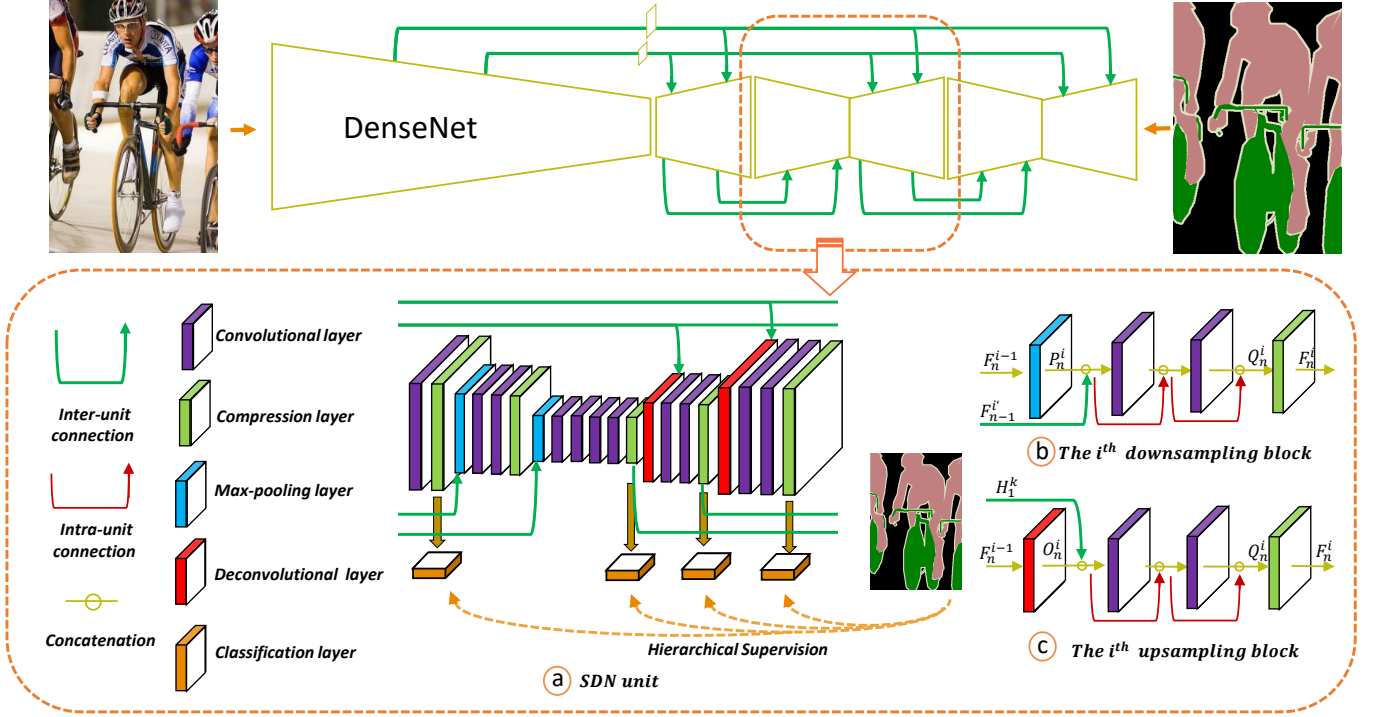
Fig. 1. Overall architecture of our approach. The upper part indicates the structure of proposed stacked deconvolutional network (SDN), the lower part indicates the detailed structure of the SDN unit (a), the downsampling block (b), the upsampling block (c). (Best viewed in color.)

architecture to capture multi-scale information, where the nearest neighbor interpolation is employed in its upsampling path. DCDN [29], which is our previous work, stacks many small deconvolutional networks for enhancing the learning ability of the network. However, the performances of the above deconvolutional solutions are limited by their difficulties on model optimization or simple network design.

In this work, we stack multiple shallow deconvolutional networks one by one to improve accurate boundary localization. In contrary to the work [8], we adopt pretrained classification network to encode images and deconvolutional layers to generate more refined recovery of the spatial resolution, and explore dense connection structure and hierarchical supervision in our network for easier network optimization. Meanwhile, we expand our previous work DCDN [29] by redesigning deconvolutional network with intra-unit and inter-unit connections, introducing hierarchical supervision, and inheriting the ImageNet [21] pre-trained network. All these designs make our network produce more discriminative features and easy to optimize.

Our proposed SDN imports dense connections to make the stacked deep model easy to optimize. This is inspired by the work of DenseNet [20]. Accordingly, we will overview the basic idea of DenseNet in the following.

DenseNet adopts dense connections to avoid the vanishing gradient problem and improve the flow of information, and it mainly consists of dense blocks and transition layers. The input of each convolutional layer within a dense block is the concatenation of all feature outputs of its previous layers at a given resolution. Consider x_l is the output of the lth layer in a dense block, x_l can be computed as follows:

$$x_l = H_l \left([x_0, x_1, \ldots, x_{l-1}] \right) \tag{1}$$

where $[x_0, x_1, \ldots, x_{l-1}]$ stands for the concatenation of the feature maps $x_0, x_1, \ldots, x_{l-1}$, and x_0 is the inputs of the dense block. Meanwhile, H_l is defined as a composite function of operations: BN, ReLU, a 1×1 convolution operation followed by BN, ReLU, a 3×3 convolution operation. Each dense block is followed by a transition layer, which do convolution and pooling to change the number and the size of feature maps. Finally, a softmax classifier is attached to make prediction.

3 OUR APPROACH

3.1 Overview

We propose a new framework called as Stacked Deconvolutional Network (SDN), which aims to capture more contextual information and recover high-resolution prediction progressively by stacking multiple shallow deconvolutional networks one by one.

As illustrated in Fig. 1, three (but not limited) deconvolutional networks, called as *SDN units*, are piled up from end to end, and the intra-unit connections and the inter-unit connections are jointly employed. Such a connected structure of the network enables efficient backward gradient propagation, and effective detailed boundaries restoration for full resolution labelling prediction. In order to obtain high-quality semantic representation, we adopt DenseNet-161, pre-trained on ImageNet [21], as the encoder module of the first deconvolutional network. A typical structure of a SDN unit is shown in Fig. 1(a), which consists of

two *downsampling blocks* and two *upsampling blocks*. Each downsampling (upsampling) block starts with a pooling (deconvolutional) layer, crosses a few of convolutional layers and ends with a compression layer. Besides, the hierarchical supervision for multi-level feature maps are jointly employed during each upsampling process to guarantee discrimination of the output prediction. In the inference step, we only use the highest-resolution results of the last unit as the final prediction. In the following subsections, we will elaborate the designing details of each SDN unit, intra-unit and inter-unit connections, and the hierarchical supervision.

3.2 Deconvolutional Network (SDN unit)

We design a shallow deconvolutional network referred to as a SDN unit to capture contextual information and refine poor object delineation, whose structure is illustrated in Fig. 1(a). A SDN unit has an encoder module and a corresponding decoder module. In a encoder module, we stack two downsampling blocks to enlarge the receptive fields of the network, resulting in the low resolution of the feature maps. In a decoder module, upsampling blocks are used twice to achieve a more refined reconstruction of the feature maps.

For a given SDN unit, its encoder module takes the outputs of its previous unit as inputs and produces low-resolution feature maps with larger receptive fields. Here, we employ the downsampling blocks twice, resulting in $\frac{1}{16}$ spatial resolution of the input image. The structure of downsampling block is shown in Fig. 1(b). A downsampling block consists of a max-pooling layer and 2 (or more) convolutional layers, and a compression layer. First, the feature map F_n^{i-1} from the $(i-1)^{th}$ block is fed into the max-pooling layer in the ith downsampling block of the nth unit and sub-sampled by a factor of 2 to produce a new feature map P_n^i. Second, behind the max-pooling layer, we cascade 2 convolutional layers with intra-unit connections. Concretely, the input of the convolutional layer is the concatenation of the input and output of its previous convolutional layer. Such a densely connected structure is beneficial to feature reuse, i.e., the multi-scale appearance of objects can be better captured to obtain effective semantic segmentation. The densely connected structure takes the feature map P_n^i and the output $F_{n-1}^{i'}$ of the i'^{th} block in the $(n-1)^{th}$ unit as inputs and outputs an feature map Q_n^i, where the i'^{th} block is the backward nearest block to the ith block in the same resolution. However, intra-unit connections bring the linear growth in the channel number of the feature maps, resulting in too much GPU memory demanding. Finally, in order to decrease the computational cost and the memory demanding, we employ a compression layer, which performs convolutions with fewer filters to reduce the channel number of the feature map Q_n^i, to generate an feature map F_n^i. In short, the ith downsampling block takes two feature maps F_n^{i-1} and $F_{n-1}^{i'}$ as its inputs, and outputs an new feature map F_n^i, the operations can be summarized as follows:

$$
\begin{aligned}
P_n^i &= Max(F_n^{i-1}), \\
Q_n^i &= Trans([P_n^i, F_{n-1}^{i'}]), \\
F_n^i &= Comp(Q_n^i).
\end{aligned}
\tag{2}
$$

where $[P_n^i, F_{n-1}^{i'}]$ stands for the concatenation of the feature maps P_n^i and $F_{n-1}^{i'}$. $Max(\cdot)$ denotes a max-pooling operation. $Trans(\cdot)$ denotes a transformation function of the densely connected structure, in which two sequences of BN, ReLU, a 3×3 convolution, dropout, and concatenation operations are performed. $Comp(\cdot)$ refers to a 3×3 convolution operation. It should be noted that our encoder module of the first SDN unit employs full convolutional DenseNet-161 to obtain high-semantic features, while the other SDN units adopt the encoder module described above.

In the decoder module, we apply upsampling blocks to progressively upsample feature maps to larger resolution. The upsampling blocks are also used twice to enlarge resolution back to $\frac{1}{4}$ spatial resolution of the input image. An upsampling block consists of a deconvolutional layer and several convolutional layers and a compression layer. As shown in Fig. 1(c), in the ith upsampling block, we first apply deconvolutional operation on the output F_n^{i-1} of the $(i-1)^{th}$ block and produce a high-resolution feature map O_n^i. Then the feature map O_n^i is concatenated with the feature map H_1^k from the kth block in the encoder module of the first SDN unit, where the kth block have the same resolution with O_n^i. Finally, the concatenated feature maps are fed to the subsequent convolutional layers and compression layer, which adopt the similar connection structure as the downsampling block. The output F_n^i of the ith upsampling block can be computed as follow:

$$
\begin{aligned}
O_n^i &= Deconv(F_n^{i-1}), \\
Q_n^i &= Trans([O_n^i, H_1^k]), \\
F_n^i &= Comp(Q_n^i).
\end{aligned}
\tag{3}
$$

where $Deconv(\cdot)$ refers to a deconvolutional operation. For the last unsamlping block of the last SDN unit, we abandon its compression layer for better prediction. Note that, our highest resolution of SDN unit is set to a quarter of input images. One important reason for this design is that we can reduce GPU memory usage of a single SDN unit to stack more units.

Contrary to traditional deconvolutional networks [16], [17], [18], which have difficulty in network training and require additional aids, our deconvolutional network achieves great improvements in network training. First, our deconvolutional network is shallow encoder-decoder framework, which only includes two simple downsampling and upsampling blocks. Then, intra-unit connections are performed between convolutional layers, and this enables effective backward propagation of the gradients through a SDN unit. All these design improves end-to-end training of all network blocks.

3.3 Densely Connecting SDN units

To enhance the learning ability of network, we stack some shallow SDN units as introduced in above subsection into a very deep model. Meanwhile, the inter-unit connections are imported to make the multi-scale information across different units efficient to reuse. As shown in Fig. 1, there are two types of inter-unit skip connections in the proposed framework. One is between any two adjacent SDN units, and the other is a kind of skip connections from the first
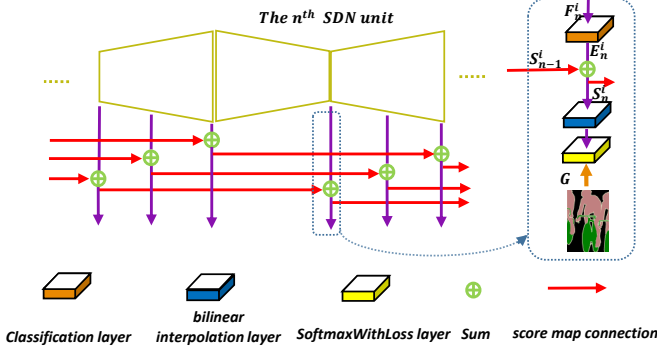
Fig. 2. Hierarchical supervision with score map connections during upsampling process. (Best viewed in color.)

SDN units to others. In the following, we will introduce them in detail.

The skip connections between any two adjacent SDN units are used to promote the flows of high-level semantic information and improve the optimization of encoder modules. For a given SDN unit, its encoder module exploit the intermediate features of the decoder module in its previous unit by a shortcut path. Specifically, the feature map P_n^i from the ith block of the nth unit is concatenated with the feature map $F_{n-1}^{i'}$ from the i'^{th} block of the $(n-1)^{th}$ unit. Such a concatenation operation is shown in Fig. 1(b). We adopt the skip connection twice according to the number of downsamlping blocks. With such skip connections, the gradients can be directly propagated to the previous unit, thus promoting the optimization of network. Besides, the fusion of features from adjacent units would efficiently capture multi-scale information.

Meanwhile, we adopt the skip connections from the first SDN unit to others to fuse low-level representations and high-level semantic features, resulting in refined object segmentation edges. As illustrated in Fig. 1, we feed the low-level visual features from the kth block of the first encoder module into a convolutional layer, where the convolution operations generate a feature map H_1^k. Then the feature map H_1^k is concatenated with the outputs of the deconvolutional layer in the corresponding resolution. Here, we combine the features from the first encoder module with the upsampling features from each decoder module. This is because mid-level representations in first encoder module have more spatial visual information. Such inter-unit connections contribute to detailed boundaries for high-resolution prediction.

3.4 Hierarchical Supervision

Deeper networks lead to better performance. However, the difficulty in training deeper network becomes a major problem. We stack multiple shallow deconvolutional networks with random initialization, which leads to additional optimization difficulty. According to the previous design, inter-unit and intra-unit connections are used to assist training. In order to alleviate this problems further, we add hierarchical supervision in each SDN unit. Specifically, the output F_n^i of the ith upsampling block is fed to a pixel-wise classification

layer to obtain a feature map E_n^i with channel C, where C is the number of possible labels. Then E_n^i is upsampled to match the size of the input image with bilinear interpolation, and finally supervised with pixel-wise groundtruth. In this way, the hierarchical supervision helps optimize the learning process. The pixel-wise cross-entropy loss is applied to all predictions in the network. For the ith block with a pixel-wise cross-entropy loss $\mathcal{L}^i$, the loss is computed as follows:

$$\mathcal{L}^i = \ell(Bi(E_n^i), G), E_n^i = Classify(F_n^i) \qquad (4)$$

where ℓ denotes a cross-entropy loss function, $Classify(\cdot)$ denotes a classifier with a 3×3 convolution operation, $Bi(\cdot)$ denotes a bilinear interpolation operation, and G refers to a ground truth map.

In order to further improve the performance of the network, we enhance score map fusion before bilinear interpolation in the same resolution, depicted in Fig. 2. In particular, for the nth SDN unit, the output E_n^i of the classification layer are fused with the features S_{n-1}^i in the $(n-1)^{th}$ unit by element-wise sum operation to produce an new fused feature S_n^i, and then the fused feature S_n^i is upsampled with bilinear interpolation, and also supervised with pixel-wise groundtruth. Therefor, we can compute the loss $\mathcal{L}^i$ as follows:

$$\begin{aligned} \mathcal{L}^i &= \ell(Bi(S_n^i), G), \\ S_n^i &= E_n^i \oplus S_{n-1}^i \end{aligned} \qquad (5)$$

where $\oplus$ denotes element-wise sum. Such a same-scale sum fusion of score maps enhances information flows and improves segmentation results.

In the testing phase, we only use the highest-resolution result of the last unit as the final prediction. It can be found that, with the hierarchical supervision, the feature maps guarantee the discrimination and restrain the noise during upsampling process.

3.5 Implementation Details

In each SDN unit, we stack 4 convolutional layers in the block of the lowest resolution for better global perspective, while the other blocks have 2 convolutional layers. The convolutional layers in a block is composed of BN, ReLU, and a 3×3 convolution operation followed by dropout with the probability of 0.2. The filter numbers of the convolutional layers are all set to 48. In two downsampling blocks, the compression layers are 3×3 convolution operations and their filter numbers are set to 768 and 1024 respectively. Max-pooling with 2×2 window and stride 2 is preformed. Meanwhile, in two upsampling blocks, the compression layers also are 3×3 convolution operations and their filter number are set to 768 and 576 respectively. Upsampling operation can be done by a 4×4 deconvolutional operation with stride 2. The convolutional layer in inter-unit connections from the first SDN unit to others is composed of BN, ReLU, and a 3×3 convolution operation. We employ the full convolutional DenseNet-161 network, pre-trained on ImageNet, as the first encoder module. Meanwhile, the last downsampling operations is removed and dilation convolutions is used.

The proposed SDN is implemented with Caffe [30]. Similar to [10], we optimize it using the "ploy" learning rate policy, with batchsize of 10. We set power to 0.9, momentum to 0.9 and weight decay to 0.0005. We apply data augmentation in the training step. Here, random crops of 320×320 are used, and horizontal flip is also applied. In the inference step, we pad images with mean value to make the length divisible by 16 before feeding full images into the network.

4 EXPERIMENTS

To show the effectiveness of our approach, we carry out comprehensive experiments on PASCAL VOC 2012 dataset [22], CamVid dataset [23] and GATECH dataset [24]. Moreover, we perform a series of ablation evaluations to inspect the impact of various components on PASCAL VOC 2012 dataset. Experimental results demonstrate our proposed SDN achieves new state-of-the-art performance on 3 datasets.

4.1 Dataset and Evaluation Metrics

4.1.1 Dataset

PASCAL VOC 2012 : The dataset has 1,464 images for training, 1,449 images for validation and 1,456 images for testing, which involves 20 foreground object classes and one background class. Meanwhile, we augment the training set with extra labeled PASCAL VOC images provided by Semantic Boundaries Dataset [31], resulting in 10,582 images for training.

CamVid: The dataset is a street scene understanding dataset which consists of 5 video sequences. Following [17], we split the dataset into 367 training images, 100 validation images, and 233 test images. The resolution of each image is 360×480 and all images belong to 11 semantic categories. Compared with PASCAL VOC 2012 dataset, CamVid dataset has more strong spatial-relationship among different categories.

GATECH: The dataset is a large video set of outdoor scenes which consists of 63 videos with 12241 frames for training and 38 videos with 7071 frames for testing. The dataset is labeled with 8 semantic classes which are sky, ground, solid, porous, cars, humans, vertical mix, and main mix.

4.1.2 Evaluation Metrics

We only report the results of Mean IoU on PASCAL VOC 2012 dataset for the common convention, while we evaluate other datasets with Global Avg and Mean IoU.

- Global Avg: Percentage of correctly classified pixels over the whole dataset. i.e. $\frac{\sum_i t_{ii}}{\sum_i T_i}$.
- Mean IoU: Ratio of correctly classified pixels in a class over the union set of pixels predicted to this class and groundtruth, and then averaged over all classes. i.e. $\frac{1}{N}\sum_i \frac{t_{ii}}{T_i + \sum_j t_{ji} - t_{ii}}$.

where N is the number of semantic classes, and T_i is the total number of pixels in class i, while t_{ij} indicates the number of pixels which belong to class i and predicted to class j.

(a) SDN$_{M1}$

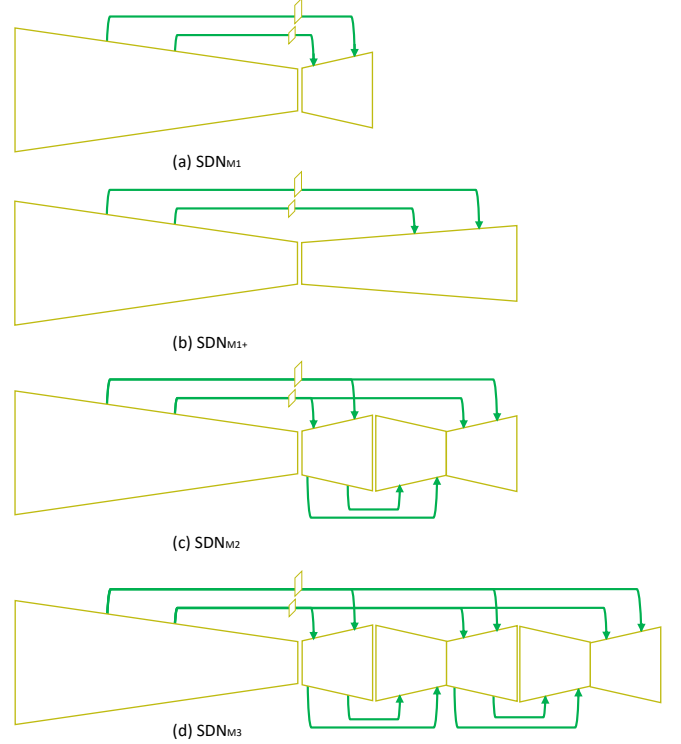(b) SDN$_{M1+}$

(c) SDN$_{M2}$

(d) SDN$_{M3}$

Fig. 3. Different stacked SDN structure. (Best viewed in color.)

4.2 Results on PASCAL VOC 2012 dataset

In this subsection, we first verify the effectiveness of each component in our approach, including the effects of stacking multiple SDN units, stacked network design and hierarchical supervision. Then we report the experimental results on PASCAL VOC 2012 test set. Here, we set the initial learning rate to 0.00025, and further apply data augmentation by randomly transforming the input images with 5 scales {0.6, 0.8, 1, 1.2, 1.4}, and 5 aspect rations {0.7, 0.85, 1, 1.15, 1.3}.

4.2.1 Stacking multiple SDN units

We stack multiple SDN units to refine the segmentation maps. The key of the stacked architecture is that each unit can capture more contextual information and recover the high-resolution features. To verify this intuition, we gradually increase the number of the SDN units and test the performance respectively. As shown in Fig. 3, we refer to the network stacking k SDN units as SDN$_{Mk}$, and the number of the SDN units is up to 3.

The results are shown in Table 1, it is observed that the performance consistently increases with the growth of SDN unit number. Specially, the performance of SDN$_{M1}$ network is only 78.2%. When we increase SDN unit number from 1 to 3, the performance improves from 78.2% to 79.2% to 79.9%. Meanwhile, we show some predicted semantic maps under different networks in Fig. 4. With the growth of SDN unit number, the object edges are ameliorated (the first and fourth column), and the object discriminate is enhanced (the second and third column). These factors bring improvement of the semantic maps. The noticeable trend indicates that, with increasing number of the stacked units, the model ben-
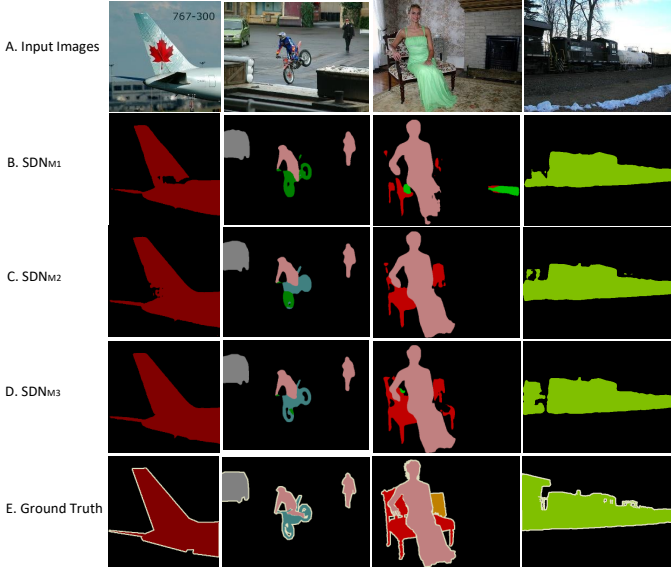
Fig. 4. Results on PASCAL VOC 2012 val set. For every column we list input images (A), the semantic segmentation results of SDN_{M1} network (B), SDN_{M2} network (C), SDN_{M3} network (D), and Ground Truth (E).

efits from the deeper network. Moreover, stacking multiple SDN units makes a coarse-to-fine prediction process, thus improving the network performance. However, it also leads to more computational stress and GPU memory demand, thus we stack up to 3 SDN units as more units bring too slight improvements.

TABLE 1
The performance comparison between different networks on PASCAL VOC 2012 val set.

	SDN_{M1}	SDN_{M2}	SDN_{M3}	$\text{SDN}_{M1}+$
Depth	169	185	201	185
Parameters (M)	84.9	161.7	238.5	161.7
Mean IoU	78.2	79.2	79.9	78.6

4.2.2 Stacked network design

From the experiments above, the network depth and its parameter size increase with the number of stacked units. We want to explore that the improvements are mainly brought by more parameters or stacked network design. To address this problem, we design a single encoder-decoder network, named as $\text{SDN}_{M1}+$, which has the same network depth and the size of parameters as the network stacking two SDN units, i.e., SDN_{M2}. $\text{SDN}_{M1}+$ also employ DenseNet-161 as the encoder module, meanwhile, the decoder module consists of 6 trivial convolutional layers and 2 upsampling blocks, the number of layers in upsampling block are set to 11 and 7 respectively. The results for the performance comparison between $\text{SDN}_{M1}+$ and SDN_{M2} are listed in Table 1. From the results we can see that the performance of SDN_{M2} is 0.6 percent higher than $\text{SDN}_{M1}+$. The comparison indicates that such a stacked network design, which captures

more contextual information, improves the performance of the network.

4.2.3 Hierarchical supervision

To further alleviate the difficulty on optimization, we add hierarchical supervision to the upsampling process in each unit. In order to explore the impact of hierarchical supervision, we take SDN_{M1} as an example to test the performance of different supervision, Specifically, we modify SDN_{M1} network by adding supervision at different upsampling scales, and refer corresponding settings to as SDN_{M1_1}, SDN_{M1_2}, SDN_{M1} respectively. Here, we denote by *up_ratio* the ratio of the input image spatial resolution to the output resolution of the block. In SDN_{M1} network, we add supervision at *up_ratio* = 16, 8, 4. And for SDN_{M1_2} network, we add supervision at *up_ratio* = 8, 4. In SDN_{M1_1} network, we only add supervision at *up_ratio* = 4. In the inference step, we output the prediction at different *up_ratio*.

Results are shown in Table 2, we can find that, in SDN_{M1} network, the performance at *up_ratio* = 4 is 2.3 percent higher than the performance at *up_ratio* = 16, it is clear that the high-resolution prediction, which use more low-level visual features, performs better than the low-resolution prediction during upsampling process of the unit. Meanwhile, the performance keep increasing with the number of the supervision. the performance of SDN_{M1_2} is 0.5 percent higher than SDN_{M1_1}, and there is modest improvement in performance from 78.0 to 78.2 when we add supervision at all *up_ratio*. The results prove the effectiveness of hierarchical supervision, and we employ hierarchical supervision during upsampling process of each unit.

TABLE 2
The performance comparison between different supervision on PASCAL VOC 2012 val set.

	up_ratio	SDN_{M1_1}	SDN_{M1_2}	SDN_{M1}
	16	–	–	75.9
Mean IoU	8	–	77.1	77.7
	4	77.5	78.0	78.2

We also explore the effects of score map fusion, depicted in Fig. 2. It is to compare the performance between networks with or without score map fusion. To this end, we stack two SDN units without score map connection, and refer to the corresponding network as $\text{SDN}_{M2}-$. The results are shown in Table 3, the performance of SDN_{M2} is 0.4 percent higher than $\text{SDN}_{M2}-$. From the results we can see that the score map fusion is benefit for the segmentation result.

TABLE 3
The performance comparison networks with or without score map connections on PASCAL VOC 2012 val set.

	SDN_{M2}	$\text{SDN}_{M2}-$
Mean IoU	79.2	78.8

4.2.4 Some improvement strategies

In the section, detailed evaluations are performed on PAS-CAL VOC 2012 val dataset. Here, we adopt several steps to improve segmentation performance further based on the SDN_{M2} network: (1) UP: We further restore high resolution features by cascading a upsampling block, and we refer to the network as SDN_{M2*}. (2) MS_Flip: We average the segmentation probability maps from 5 image scales {0.5, 0.8, 1, 1.2, 1.4} as well as their mirrors for inference. (3) COCO: Many state-of-the-art models adopt Microsoft COCO dataset [32] for the better performance. In order to compare with these models, we also pretrain the model of SDN_{M2*} on MS-COCO dataset. We evaluate how each of these factors affects val set performance in Table 4.

TABLE 4
The performance comparison between different measures on PASCAL VOC 2012 val set.

Up	MS_Flip	COCO	Mean IoU
			79.2
✓			79.6
✓	✓		80.7
✓	✓	✓	84.8

Increasing an upsampling block gives 0.4% gain, and segmentation map fusion brings another 1.2% improvement. Moreover, when we pretrain the model of SDN_{M2*} on MS-COCO dataset, the performance attains 84.8%. Compared with current well-known method Deeplabv3 [11](82.7% pre-trained on MS-COCO), our method of the SDN_{M2*} outperforms theirs by 2.1%, which shows the strength of our proposed SDN network.

4.2.5 Comparing with the state-of-the-art methods

We further compare our method with the state-of-the-art methods on PASCAL VOC 2012 test set. Here, based on two settings, i.e., with or without pre-trained with MS-COCO dataset, we fine tune our the model of SDN_{M3} on PASCAL VOC 2012 trainval set, and submit our test results to the official evaluation server. Results are shown in Table 5. In the two settings, our method both outperforms all the other methods. Trained with only PSACAL VOC 2012 data, we achieve a Mean IoU score of 83.5% [1]. When we pretrain the model of SDN_{M3} on MS-COCO dataset, it reaches 86.6% [2] in Mean IoU. Specifically, our model outperforms the RefineNet [28] by 2.4%, and RefineNet employs deep encoder-decoder structure and demonstrates outstanding performance on several semantic segmentation dataset. Meanwhile, our model also outperforms the PSPNet [28] by 1.2% and Deeplabv3 [11] by 0.9%, they both adopt multi-scale or global features to capture contextual information. These comparisons indicate that our proposed SDN network can more effectively capture contextual information and generate accurate boundary localization.

4.3 Results on CamVid dataset

In this subsection, we carry out experiments on the CamVid [23] dataset to further evaluate the effectiveness of our method. The experimental settings are as follows: we apply SDN_{M2*} network, and train the network with 367 training images, and test it with 233 test images. The initial learning rate is changed to 5e-5. Meanwhile, we also adopt data augmentation to reduce overfitting by multi-scale translation. Note that we compare SDN with the previous state-of-the-art methods on two settings, i.e., initializing the network with or without the model pretrained on PASCAL VOC 2012 [22].

Following [17], [29], [41], Mean IoU and Global Avg are employed to evaluate our method on this dataset. We compare our method with previous ones in Table 6. Results show that our method on two settings both outperform other methods. The model, without pretrained on PASCAL VOC 2012(marked by SDN), achieves 69.6% in Mean IoU and 91.7% in Global Avg. Particularly, the classes, including *Car*, *Sign*, *Pedestrian*, *Bicyclist*, have a major boost in performance. When we initialize the network pretrained on PASCAL VOC dataset (marked by SDN+), the performance further improves by 2.2 percent in Mean IoU and 1 percent in Global Avg, and here nine out of eleven object categories achieve best performance in Mean IoU. Note that, the CamVid dataset sampling in video frames contains temporal information, and some works [39], [40] mine temporal information to aid segmentation results. Our method still outperforms these works by a relative large margin. Besides, the spatio-temporal information of the dataset is complementary to our method and could bring additional improvements.

Some test images along with ground truth and our predicted semantic maps are shown in Fig. 6. We can find that our network can well sketch multi-scale appearance of objects, including large-scale objects,i.e., building and car, and shape objects i.e., poles and pedestrians. All the results on the this dataset show that our network can capture more contextual information and learn better spatial-relationship.

4.4 Results on GATECH dataset

In order to further verify the generalization of our models, we evaluate our network on GATECH [24] dataset, which is much larger than CamVid dataset, but has a lot of noisy annotations. We employ the SDN_{M2*} network with the same training strategy on CamVid. We also compare our models with previous state-of-the-art methods on two settings, i.e., initializing the network with (marked by SDN) or without (marked by SDN+) the model pretrained on PASCAL VOC 2012 [22].

Results are shown in Table 7, we can find that our method on two settings both outperforms other methods. SDN yields 53.5% in mean IoU and 84.6% in Global Avg. and SDN+ improves the result significantly, which gives 1.7% gain in Global Avg and 2.4% in Mean IoU. Specially, our model, without using any temporal, performs better than the models [39], [43] which exploit spatio-temporal

1. http://host.robots.ox.ac.uk:8080/anonymous/Z9RDVZ.html
2. http://host.robots.ox.ac.uk:8080/anonymous/GRWV3B.html

TABLE 5
Experimental results on PASCAL VOC 2012 test set.

| Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
Only using VOC data																					
FCN [9]	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 [12]	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
CRF-RNN [33]	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
DeconvNet [18]	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
GCRF [34]	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2
DPN [35]	87.7	59.4	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0	74.1
Piecewise [36]	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	75.3
ResNet38 [37]	94.4	72.9	**94.9**	68.8	**78.4**	90.6	**90.0**	92.1	40.1	90.4	71.7	89.9	93.7	**91.0**	89.1	71.3	90.7	61.3	87.7	**78.1**	82.5
PSPNet [15]	91.8	71.9	94.7	71.2	75.8	95.2	89.9	95.9	39.3	90.7	71.7	90.5	**94.5**	88.8	89.6	**72.8**	89.6	**64.0**	85.1	76.3	82.6
SDN	**96.2**	**73.9**	94.0	**74.1**	76.1	**96.7**	89.9	**96.2**	**44.1**	**92.6**	**72.3**	**91.2**	94.1	89.2	**89.7**	71.2	**93.0**	59.0	**88.4**	76.5	**83.5**
Using VOC+COCO data																					
CRF-RNN [33]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
Dilation8 [13]	91.7	39.6	87.8	63.1	71.8	89.7	82.9	89.8	37.2	84.0	63.0	83.3	89.0	83.8	85.1	56.8	87.6	56.0	80.2	64.7	75.3
DPN [35]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Piecewise [36]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
DeepLab [12]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
RefineNet [28]	95.0	73.2	93.5	78.1	84.8	95.6	89.8	94.1	43.7	92.0	77.2	90.8	93.4	88.6	88.1	70.1	92.9	64.3	87.7	78.8	84.2
ResNet38 [37]	96.2	75.2	95.4	74.4	81.7	93.7	89.9	92.5	48.2	92.0	79.9	90.1	95.5	91.8	91.2	73.0	90.5	65.4	88.7	80.6	84.9
PSPNet [15]	95.8	72.7	95.0	78.9	84.4	94.7	**92.0**	95.7	43.1	91.0	**80.3**	91.3	96.3	**92.3**	90.1	71.5	**94.4**	66.9	88.8	82.0	85.4
DeepLabv3 [11]	96.4	76.6	92.7	77.8	**87.6**	96.7	90.2	95.4	47.5	93.4	76.3	91.4	**97.2**	91.0	**92.1**	71.3	90.9	**68.9**	**90.8**	79.3	85.7
SDN+	**96.9**	**78.6**	**96.0**	**79.6**	84.1	**97.1**	91.9	**96.6**	**48.5**	**94.3**	78.9	**93.6**	95.5	92.1	91.1	**75.0**	93.8	64.8	89.0	**84.6**	**86.6**
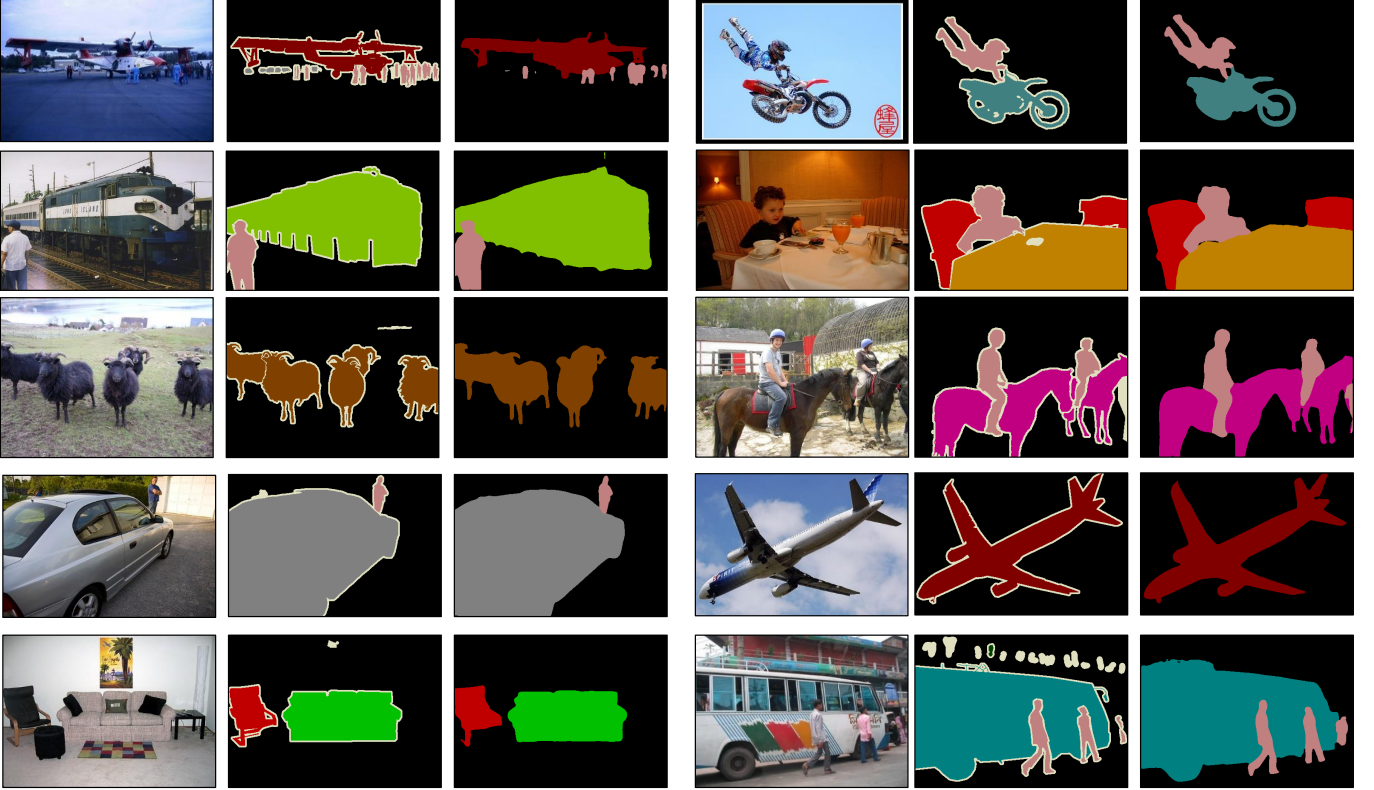
Fig. 5. Results on PASCAL VOC 2012 dataset. The images in each row from left to right are: (1) input image (2) groundtruth (3) semantic segmentation result.
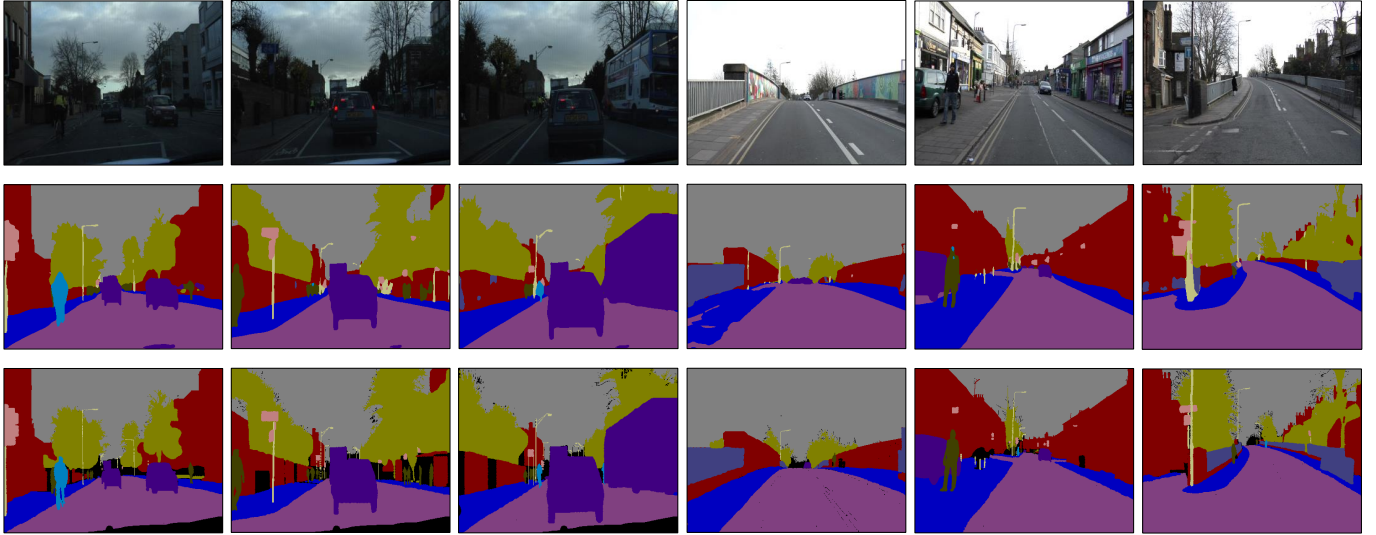
Fig. 6. Results on CamVid dataset. The images in each column from top to down are: (1) input image (2) semantic segmentation result (3) groundtruth.

TABLE 6
Experimental results on CamVid test set.

Method	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	Mean IoU	Global Avg
SegNet [17]	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	46.4	62.5
DeconvNet [18]	–	–	–	–	–	–	–	–	–	–	–	48.9	85.6
ReSeg [38]	–	–	–	–	–	–	–	–	–	–	–	58.8	88.7
DeepLab-LFOV [10]	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6	–
Bayesian SegNet [19]	–	–	–	–	–	–	–	–	–	–	–	63.1	86.9
Dilation8 [13]	82.6	76.2	89.0	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3	79.0
HDCNN-448+TL [39]	–	–	–	–	–	–	–	–	–	–	–	65.6	90.9
Dilation8+FSO [40]	84.0	77.2	91.3	85.6	49.9	92.5	59.1	37.6	16.9	76.0	57.2	66.1	88.3
FC-DenseNet103 [41]	83.0	77.3	93.0	77.3	43.9	94.5	59.6	37.1	37.8	82.2	50.5	66.9	91.5
G-FRNet [42]	82.5	76.8	92.1	81.8	43.0	94.5	54.6	47.1	33.4	82.3	59.4	68.0	–
DCDN [29]	–	–	–	–	–	–	–	–	–	–	–	68.4	91.4
SDN	84.45	76.9	92.2	88.2	51.6	93.4	62.4	37.0	37.5	77.8	64.4	69.6	91.7
SDN+	**85.2**	**77.5**	92.3	**90.2**	**53.9**	**96.0**	**63.8**	39.8	**38.4**	**85.36**	**66.9**	**71.8**	**92.7**

TABLE 7
Experimental results on GATECH test set.

Method	Temporal Info	Global Avg	Mean IoU
3D-V2V-scratch [43]	Yes	66.7	-
3D-V2V-finetune [43]	Yes	76.0	-
FC-DenseNet103 [41]	No	79.4	-
HDCNN-448+TL [39]	Yes	82.1	48.2
DCDN [29]	No	83.5	49.0
SDN	No	84.6	53.5
SDN+	No	**86.3**	**55.9**

relationships between video frames. All these comparisons confirm the proposed SDN a robust and effective model for coarse-annotation dataset. Some test images along with ground truth and our predicted semantic maps are shown in Fig. 7.

5 CONCLUSION

In this paper, we have presented a stacked deconvolutional network (SDN), a novel deep network architecture for semantic segmentation. We stack multiple SDN units to make network deeper and realize a coarse-to-fine learning process, meanwhile, intra-unit and inter-unit connections and hierarchical supervision are adopted to promote network optimization. The ablation experiments show that those designs effectively capture contextual information and recover the spatial resolution for accurate boundary localization, which benefit network performance. Our best model outperform all previous works on three public benchmarks.
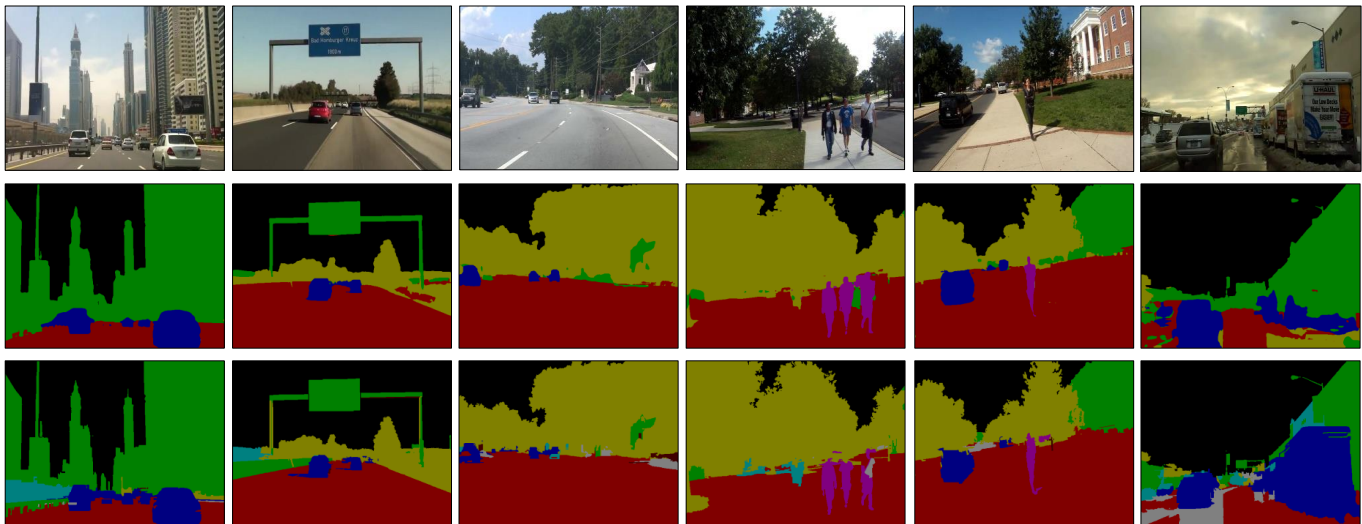
Fig. 7. Results on GATECH dataset. The images in each column from top to down are: (1) input image (2) semantic segmentation result (3) groundtruth.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[7] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *CVPR*, 2014, pp. 1653–1660.

[8] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 483–499.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *ICLR*, 2015.

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[13] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[14] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," in *ICLR*, 2016.

[15] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CVPR*, 2017.

[16] Y. Wang, J. Liu, Y. Li, J. Yan, and H. Lu, "Objectness-aware semantic segmentation," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 307–311.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[18] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015.

[19] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[20] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *CVPR*, 2017.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, pp. 1–42, 2014.

[22] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[23] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[24] S. Raza, M. Grundmann, and I. Essa, "Geometric context from videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3081–3088.

[25] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," *arXiv preprint arXiv:1702.08502*, 2017.

[26] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *arXiv preprint arXiv:1703.06211*, 2017.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[28] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation," in *CVPR*, 2017.

[29] J. Fu, J. Liu, Y. Wang, and H. Lu, "densely connected deconvolutional network for semantic segmentation," in *ICIP*, 2017.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[31] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011, pp. 991–998.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[33] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional random fields as recurrent neural networks," in *ICCV*, 2015.

[34] R. Vemulapalli, O. Tuzel, M.-Y. Liu, and R. Chellapa, "Gaussian conditional random field network for semantic segmentation," in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3224–3233.

[35] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *ICCV*, 2015.

[36] G. Lin, C. Shen, I. Reid, and A. van den Hengel, "Efficient piecewise training of deep structured models for semantic segmentation," in *CVPR*, 2016.

[37] Z. Wu, C. Shen, and A. v. d. Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *arXiv preprint arXiv:1611.10080*, 2016.

[38] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville, "Reseg: A recurrent neural network-based model for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 41–48.

[39] Y. Wang, J. Liu, Y. Li, J. Fu, M. Xu, and H. Lu, "Hierarchically supervised deconvolutional network for semantic video segmentation," *Pattern Recognition*, vol. 64, pp. 437–445, 2017.

[40] A. Kundu, V. Vineet, and V. Koltun, "Feature space optimization for semantic video segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3168–3175.

[41] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," *arXiv preprint arXiv:1611.09326*, 2016.

[42] N. D. B. B. Md Amirul Islam, Mrigank Rochan and Y. Wang, "Gated feedback refinement network for dense image labeling," in *CVPR*, 2017.

[43] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Deep end2end voxel2voxel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 17–24.