Neuron-level Selective Context Aggregation for Scene Segmentation

Zhenhua Wang*
Hebrew University of Jerusalem
zhenua.wang@mail.huji.ac.il

Fanglin Gu*
Shandong University
gfl699468@mail.sdu.edu.cn

Dani Lischinski
Hebrew University of Jerusalem
danix@mail.huji.ac.il

Daniel Cohen-Or
Tel Aviv University
dcor@tau.ac.il

Changhe Tu
Shandong University
chtu@sdu.edu.cn

Baoquan Chen
Shandong University
baoquan@sdu.edu.cn

Abstract

Contextual information provides important cues for disambiguating visually similar pixels in scene segmentation. In this paper, we introduce a neuron-level Selective Context Aggregation (SCA) module for scene segmentation, comprised of a contextual dependency predictor and a context aggregation operator. The dependency predictor is implicitly trained to infer contextual dependencies between different image regions. The context aggregation operator augments local representations with global context, which is aggregated selectively at each neuron according to its on-the-fly predicted dependencies. The proposed mechanism enables data-driven inference of contextual dependencies, and facilitates context-aware feature learning. The proposed method improves strong baselines built upon VGG16 on challenging scene segmentation datasets, which demonstrates its effectiveness in modeling context information.

1. Introduction

Scene segmentation is a long standing fundamental problem in computer vision, which aims to associate a semantic object category label with each pixel in an image of a scene. It has long been recognized that taking into account the semantic context between image regions can significantly improve the performance of scene segmentation algorithms [8, 14, 21, 26].

As is the case in many other areas of computer vision, the state-of-the-art in scene segmentation has been transformed through the use of deep neural networks. Long et al. [13] proposed a fully convolutional network which can be trained end-to-end to predict a dense labeling map, capable of extracting coarse object shapes. This approach was extended by Noh et al. [15] by introducing multiple deconvolutional layers to gradually capture the finer details. In these pure CNN-based methods, the representations of neurons in the convolutional feature maps are extracted from limited receptive fields, capturing only local information.

To address this problem, Liu et al. [12] propose to use the average feature of a layer as global context to augment the local features at each neuron. Yu and Koltun [28] introduce dilated convolutions to systematically aggregate multi-scale contextual information without losing resolution. Zhao et al. [30] exploit global context information by different region-based context aggregation through pyramid pooling. Although these methods improve the segmentation accuracy, they aggregate context in a predefined manner, which does not depend on the input image. Thus, they lack internal mechanisms for handling non-uniform dependencies among different image regions.

In another line of research, Shuai et al. [24] proposed Directed Acyclic Graph Recurrent Neural Networks (DAG-RNNs) to leverage contextual information, thereby enhancing the representation ability of convolutional features. Their approach has demonstrated significant advances over the state-of-the-art on three challenging datasets. However, due to the gradient vanishing problem of RNNs [1, 16], the contextual information can only be propagated to nearby neurons, thus the long-range dependencies cannot be well captured. In fact, Shuai et al. [23] report that using DAG-RNNs induced from 8-connected grids, significantly outperforms the 4-connected case. Moreover, due to the usage of
RNNs, the hidden vectors of DAG-RNNs must be sequentially updated, making it less efficient than pure CNN-based methods.

In this work, we propose a new approach for selectively aggregating contextual information at each neuron, and doing so in an input-dependent manner, by introducing a novel context aggregation module. This module takes as input the feature map of a convolutional layer, and outputs a context-aware feature map. Specifically, this module consists of two main components: a contextual dependency predictor and a context aggregation operator. The dependency predictor is an auxiliary network that infers a coefficient representing the degree of dependency between each pair of neurons. Note that these coefficients are input-dependent. The context aggregator densely connects each output neuron to all the input neurons according to its predicted coefficients, yielding feature maps with enhanced context representation ability. These feature maps are then fed to a deconvolutional network to predict a semantic label map (see Fig. 1). We refer to the proposed module as a neuron-level Selective Aggregation (SCA) module, since each of its output neurons uses a different set of weights to aggregate contextual features from the input neurons.

In summary, our work makes three main contributions:

- We present a novel context aggregation module which selectively injects global context information into local representations, based on image-specific dependencies among neurons in the deep convolutional feature map.

- We introduce a contextual dependency predictor that learns to infer image-specific dependencies between neurons. The resulting dependency coefficients allow each neuron to selectively aggregate context information from the entire image.

- We apply the proposed network on the scene segmentation task, and demonstrate its effectiveness and potential on two large datasets.

The rest of the paper is organized as follows: Sec. 2 discusses some work related to our method, Sec. 3 introduces the two components of the proposed new module, Sec. 4 gives the implementation details, and, finally, Sec. 5 reports the results of our experiments.

2. Related Works

The semantic segmentation problem has been addressed by a wide variety of methods in recent years. Since visually similar pixels are locally indistinguishable, a major question that arises is how to leverage contextual information for disambiguation. Many methods rely on Probabilistic Graphical Models (PGMs), e.g. Markov Random Fields (MRFs) and Conditional Random Fields (CRFs), to account for context [21, 8, 27, 29, 18]. These methods usually require a pre-segmentation, such as superpixels, from which to extract features, and they are usually inefficient for inference due to the iterative solving of local beliefs.

Due to the recent advances in deep neural networks, the mainstream methods of semantic segmentation usually adopt Convolutional Neural Networks (CNNs) as their basic component. Farabet et al. [6] feed their CNN with a multi-scale pyramid of images, covering a large context. Pinheiro et al. [17] encode long range pixel label dependencies by using a recurrent CNN architecture. Sharma et al. [19] use a recursive neural network to recursively aggregate contextual information from local neighborhoods to the entire image and then propagate the global context back to individual local features. Shuai et al. [22] use a non-parametric model to represent global context, and transfer such information to local features extracted by CNN. Long et al. [13] propose a fully convolutional network to directly predict label map from the input image. Noh et al. [15] extend [13] by introducing multi-layer deconvolution networks.

Liu et al. [12] propose to use the average feature of a layer as global context to augment the features at each neuron. Yu and Koltun [28] introduce dilated convolutions to systematically aggregate multi-scale contextual information without losing resolution. Zhao et al. [30] exploit global context information by different region-based context aggregation through pyramid pooling. Shuai et al. [24] utilize DAG-RNNs to embed context into local features. Different from all these methods, which use an input-independent and fixed manner to aggregate context, our method introduces a dynamic mechanism which aggregates context selectively at each neuron, and does so in an input-dependent manner. Our approach of using an auxiliary network to achieve input-dependent processing is similar in spirit to
spatial transformer networks [7] and deformable convolution networks [5]

3. Our Approach

Contextual information has been widely used in computer vision applications [8, 14, 21, 26]. In general, context can refer to any global information that has a facilitative effect on representation ability. Note, however, that a local representation may be influenced differently by different semantic categories of pixels in its context. For example, the presence of ‘sea’ pixels in an image is a more important clue for classifying nearby sand-colored pixels as ‘beach’, compared to ‘mountain’ pixels, since ‘sea’ and ‘beach’ have higher co-occurrence correlation. However, previous methods [12, 28, 30] usually embed global context into a local representation in a fixed and uniform way, thereby lacking internal mechanisms to effectively account for non-uniform dependencies among different image regions.

To enhance the ability of handling non-uniform contextual information, we propose an input-dependent way to selectively incorporate context into the local representation. Fig. 1 shows an overview of our proposed segmentation network architecture. Given an input image I, the convolution layers gradually extract its abstract features, yielding a high-level feature map X. Next, a neuron-level Selective Context Aggregation (SCA) module is applied onto X to augment it with context information. The SCA module consists of two components. The first component is the contextual dependency predictor network, which takes X as input, and through several hidden layers predicts a matrix A, which consists of the dependency coefficients between all pairs of neurons in X. A dependency coefficient \( a_{ij} \) attempts to predict the extent to which neuron \( x_j \) should be accounted for in the context of neuron \( x_i \). Next, the context aggregation operator uses the predicted dependency matrix A to transform the input feature map X into a context-aware feature map H. Finally, several deconvolutional layers are applied on H to predict the final semantic segmentation label map.

3.1. Context Aggregation Operator

Generally, a context aggregation operator aims to encode global contextual information into a local neuron representation. It takes as input a feature map \( X \), and produces a context-aware feature map \( H \) with the same spatial dimensions (but possibly with a different number of channels). To fully exploit the context, the receptive field of this operator should cover the full size image. A naive approach is to use a fully-connected layer in which each output neuron is connected to all the input neurons. To be more specific, let \( x_i \in \mathbb{R}^N \) and \( h_i \in \mathbb{R}^M \) denote the feature vectors of the \( i \)-th neuron in \( X \) and \( H \), respectively. Using a fully-connected layer each neuron in \( H \) is given by:

\[
h_i = \sum_{j=1}^{n} W_{ij} x_j
\]

where \( n \) is the number of input neurons in \( X \), and \( W_{ij} \in \mathbb{R}^{M \times N} \) is the weight matrix.

As can be seen, \( W_{ij} \) is particular to the \( i \)-th input neuron and the \( j \)-th output neuron. Thus, the learned weights are location sensitive, which is undesirable for dense label prediction, as the same category of pixels can appear anywhere in the image. Furthermore, after training, \( W_{ij} \) is fixed during test time, and thus the context aggregation is input-independent. Finally, the fully-connected layer consumes a huge amount of learnable parameters, making it easy to overfit the training data.

To address all these problems, we introduce a new aggregation operator. Our operator first uses two 1 × 1 convolutions to extract two different features, the identity feature and the context feature, for each neuron. The intuition is that the feature used to describe the neuron itself and the feature that describes its contribution to the context of other neurons may be different. Then, the output feature of a neuron is produced by linearly combining its own identity feature with the context features of all the other neurons, according to some input-dependent coefficients. Formally, the computation of the output neuron \( h_i \) is defined as:

\[
h_i = a_{ii} W_d x_i + \frac{1}{\sum_{j=1}^{n} a_{ij} - a_{ii} \sum_{j=1, j \neq i}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij} W_c x_j
\]

where \( a_{ij} \) is a coefficient which controls the amount of information taken from \( x_j \) to \( h_i \), \( W_d \in \mathbb{R}^{M \times N} \) and \( W_c \in \mathbb{R}^{M \times N} \) are the weight matrices for extracting the identity and context features, respectively. In practice, we always set \( a_{ii} \) to 1 so that the output neuron takes all of its identity feature. Fig. 2 illustrates the computation procedure of the proposed operator for a single output neuron on a 1D feature map.
Notably, this new operator has several appealing properties. First of all, the matrices $W_d, W_c$ are shared across neurons, reducing the parameter space from $n^2NM$ (for a fully-connected layer) to $2NM$. Secondly, the operation preserves spatial information, thus it is suitable to be applied on dense prediction tasks. Thirdly, the coefficients $a_{ij}$ are not fixed parameters, and are capable of representing arbitrary dependencies between different pairs of neurons. These dependencies may be manually defined, according to some prior knowledge, or automatically inferred for the specific input $X$. Here we adopt a data-driven paradigm, and introduce an auxiliary contextual dependency predictor (Sec. 3.2) that learns to infer the dependencies among neurons and provide on-the-fly input-dependent predictions for the proposed context aggregation operator.

To allow backpropagation through the proposed operator, we must define the gradients of the loss $L$ with respect to both the input feature map neurons $x_i$ and the coefficients $a_{ij}$. These gradients are given by

$$\frac{\partial L}{\partial x_i} = a_{ii} W_d^T \frac{\partial L}{\partial h_i} + \sum_{k \neq i}^n \sum_{j \neq k} a_{kj} W_c^T \frac{\partial L}{\partial h_k} \forall i \in [1,n]$$

(3)

and

$$\frac{\partial L}{\partial a_{ij}} = \sum \frac{\partial L}{\partial h_i} \odot \frac{\sum_{k \neq i}^n a_{ik}(W_c x_j - W_c x_k)}{(\sum_{k \neq i}^n a_{ik})^2} \forall j \neq i$$

(4)

where $\frac{\partial L}{\partial h_i}$ denotes the loss gradients with respect to $h_i$ received from top layer, $\odot$ denotes the element-wise multiplication, and the outermost sum in (4) is over the feature channel.

3.2. Contextual Dependency Predictor

The contextual dependency predictor (CDP) aims to estimate the dependencies between neurons of the input feature maps. More precisely, given a pair of neurons $(x_i, x_j)$, the goal is to predict the coefficient that characterizes the dependency extracted from $x_j$ should be given when aggregating the context of neuron $x_i$. Since each pair may be assigned a different weight, the global context information is encoded selectively into each neuron’s local representation.

The dependency function between neurons may be modeled by a neural network. Typically, Siamese networks, which are popular for tasks that involve finding inherent relationships between two inputs, can be used to learn such a function. Training Siamese networks requires paired inputs and their corresponding ground truths. However, our input feature map is not organized as neuron pairs, and, more importantly, we do not have well-defined ground truth for the dependencies among such pairs.

Our approach is thus to learn the dependency function implicitly, by training the CDP jointly with the context aggregation operator (as well as the rest of the network) to minimize the final segmentation loss. The CDP should take a feature map $X$ with $n$ neurons as input, and output an $n \times n$ matrix $A$, in which each element $a_{ij}$ represents the dependency coefficient between neuron $x_i$ and $x_j$, for the context aggregation operator.

To achieve this goal, we propose a new network architecture described below; also see Figure 3. Firstly, several $1 \times 1$ convolutional layers are applied on $X$ to map the representation of each neuron into a higher-level feature space. Intuitively, the purpose of this mapping is to prevent the network from simply computing linear correlations between the neurons; instead, the network is able to explore higher-level semantic relationships that may prove useful for the final semantic segmentation.

In order to efficiently predict dependencies for all neuron pairs, we use a simple trick to construct a large feature map in which each location stores a concatenation of the mapped features of a particular neuron pair. Concretely, we reshape the 2D feature map to a 1D feature vector, expand it to two 2D square feature maps using horizontal and vertical replication, respectively, and then concatenate these two replicated feature maps through the depth channel. Using the resulting large feature map, we simply apply one more $1 \times 1$ convolutional layer to produce the dependency matrix.

According to Eq. (4), the CDP can receive gradients from upper layers which enable it to be jointly trained with the main network without using extra supervision. It learns how to selectively incorporate context information, by producing appropriated dependency coefficients for different pairs of neurons, to help minimize the overall segmentation cost during training.

4. Implementation Details

4.1. Network Architecture

The convolution layers of our network are adopted from the ImageNet pre-trained VGG16 network, i.e., all the layers before the 5-th pooling layer. Following [28], we remove the last two pooling and striding layers, and replace all subsequent convolutions with dilated convolutions of appropriate factors to make the last convolution layer produce a feature map that is 4 times larger.

For the proposed SCA module, the dimensions of $W_c$ and $W_d$ in the context aggregation operator are both set to be $512 \times 512$. The contextual dependency predictor uses three $1 \times 1 \times 128$ convolution layers to extract features, and uses a $1 \times 1 \times 1$ convolution layer on the concatenated features to predict dependent coefficients.

The deconvolution module is similar to that of a FCN [13]. Concretely, we convert the first fully-connected layer of VGG16 to a $7 \times 7 \times 4096$ convolution layer, and convert the second one to a $1 \times 1 \times 4096$ convolution layer. Then, a $1 \times 1 \times \#\text{classes}$ convolution is applied to predict
scores for all classes in the dataset at each location. Finally, a bilinear upsampling layer is applied to resize the coarse score map to have the same resolution as the input.

4.2. Loss Re-weighting

The class distributions in scene labeling datasets are usually highly unbalanced, thus, it is common to increase the weights of rare classes during training to improve their accuracies [6, 23]. In our work, we simply adopt the re-weighting strategy proposed in [23]. Specifically, the weight for class $c_i$ is defined as $w_i = 2^{\lceil\log_{10}(\eta/f_i)\rceil}$ where $f_i$ is the frequency of $c_i$ and $\eta$ is a dataset dependent scalar defined by the $85% / 15%$ rule on frequent/rare classes.

4.3. Training and Testing

For training, since every component of the proposed segmentation network is differentiable, errors can be backpropagated to all network layers and parameters, making the network trainable using any gradient-based optimization method. In practice, we use min-batch gradient descent with momentum. During training, we use the “poly” learning rate policy, where current learning rate equals to the base learning rate multiplied by $(1 - \frac{iter}{maxiter})^{0.9}$. The base learning rate for the bottom convolution module is set to $10^{-3}$, while it is set to $10^{-2}$ for both the SCA and the de-convolution modules. Due to the GPU memory limitations, we set the batch size to 3. The number of training epochs is set to 50. To augment training data, similarly to most of the previous methods, we adopt random horizontal flipping for all datasets. At test time, we simply do a forward pass of the test image to obtain the estimated label map.

5. Experiments

Similarly to most previous works, we report results on three different evaluation metrics: Per-Pixel Accuracy (PPA), Class Average Accuracy (CAA) and mean Intersection over Union (mIoU).

| $K_f$ | PPA  | CAA  | mIoU |
|-------|------|------|------|
| 0     | 85.9%| 54.2%| 38.9%|
| 1     | 86.0%| 54.1%| 39.5%|
| 2     | 86.6%| 52.7%| 38.7%|
| 3     | 86.3%| 54.2%| 39.2%|
| 4     | 85.5%| 53.6%| 38.0%|

Table 1. Results of using different $K_f$ for the contextual dependency predictor on SIFT Flow. $K_f$ is set to 128.

4999/5104 images are used for training/testing. These images originally come from Pascal VOC 2010 dataset, and are augmented with pixel-wise annotations of 540 classes. Following [14], we only consider the most frequent 59 classes in the evaluation. To enable batch-based training, we rescale and pad each image to be $448 \times 448$.

COCO Stuff [2] dataset contains 10000 images in various resolutions, in which 9000/1000 images are used for training/testing. These images originally come from the Microsoft COCO dataset [10] with pixel-wise annotations of 80 Thing categories. They are further augmented with 91 Stuff categories. In total, there are 171 categories in this dataset.

5.1. Hyper-parameters Study

To be fair, we conduct the hyper-parameter study on a third dataset, SIFT Flow [11]. This dataset is smaller compared to PASCAL Context and COCO Stuff, making it faster for tuning. It contains 2488/200 images for training/testing with a resolution of $256 \times 256$, and are annotated with 33 classes.

Specifically, we study two hyper-parameters of the attention predictor: the number $K_i$ of convolution layers for feature extraction and the number $K_f$ of output feature channels of convolution in each layer. The results are shown in Tab. 1 and Tab. 2, respectively. We note that, for overall performance, $K_i = 3$ and $K_f = 512$ perform slightly better than other settings. In the following experiments, we will use these two values as the default setting.
Table 2. Results of using different $K_f$ for the contextual dependency predictor on SIFT Flow. $K_f$ is set to 3.

| $K_f$ | PPA  | CAA  | mIoU |
|-------|------|------|------|
| 128   | 86.1%| 55.0%| 40.0%|
| 256   | 86.9%| 54.5%| 40.2%|
| 512   | 87.0%| 54.3%| 40.7%|

5.2. Ablation Study

To investigate the usefulness and effectiveness of the SCA module, we compare our method with two baselines that use different configurations of the SCA module. Note that, for a fair comparison, all these baselines use the same convolution and deconvolution modules as our proposed network.

- **baseline-no** We remove the dependency predictor and set $a_{ij} = 1, \forall j = i$ and $a_{ij} = 0, \forall j \neq i$. In this case, the SCA module does not embed any context information into the local neuron representations, degenerating to a $1 \times 1$ convolution layer.

- **baseline-ave** We remove the dependency predictor and set $a_{ij} = 1, \forall i,j \in [1,n]$. In this case, the SCA module embeds the averaged context features into local neuron representations. This is similar to the global context module proposed in [12], where the globally pooled feature is concatenated with the local feature everywhere.

The results on both PASCAL Context and COCO Stuff are shown in Tab. 3. As can be seen, baseline-no performs the worst on most metrics as it does not explicitly embed context into local neuron representation. Although the use of dilated convolution can help to expand the theoretical receptive fields of neurons to aggregate context, the empirical receptive field that affects the neurons may still be small [12]. With the use of globally averaged features as context, the baseline-ave expands its receptive field to the entire image, thus improving the performance of baseline-no. Our method further improves baseline-ave on all the metrics on both datasets by a significant margin. Since all the settings are the same except for the way of aggregating context, it demonstrates the effectiveness of the proposed neuron-level selective context aggregation. The proposed SCA module successfully models neuron-dependent contextual information, by exploring the dependencies between neurons, to facilitate context-aware feature learning.

5.3. Comparisons with the State of the Art

The results on PASCAL Context dataset presented in Tab. 4. Comparing with previous CNN-based models, our method significantly outperforms all these methods on all metrics except for mIoU on UoA-Context+CRF. Considering RNN-based methods, our method outperforms CRF-RNN by a large margin. For DAG-RNN, we perform better on PPA, and are worse than it on CAA and mIoU. The results on COCO Stuff are presented in Tab. 5. As can be seen, we also outperform all CNN-based methods, and perform better than DAG-RNN on CAA and worse than it on PPA and mIoU. The results on these two challenging datasets demonstrate the effectiveness of the proposed method.

We also note that in DAG-RNN, skip-connections are used to fuse the features of higher layers with low level convolution features, which significantly improves their performance. In our method, to best investigate how the proposed SCA model helps to improve the representation ability of the last convolution features, we do not employ skip-connections in our architecture. Meanwhile, the RNN-based methods have to sequentially update their hidden vectors over time, which make them inappropriate to be parallelized to improve speed. In contrast, our method is a feed-forward network, which can be easily parallelized.

5.4. Qualitative Results

Fig. 4 visualizes the dependency maps for different selected neurons, from which we note that there are two different behaviors of the CDP. Firstly, as shown in Fig. 4(a), the two different selected neurons (indicated using green and red circles) generally have different dependency maps, as they belong to different classes. The CDP has learned to predict a different dependency map for each neuron, enhancing their local representation with different global context. Secondly, as shown in Fig. 4(b), there are also cases where two different neurons may have very similar dependency maps. This may be attributed to the fact that the salient regions, e.g., the birds here, provide an important

![Figure 4](image-url)
context for all other regions. Still, note that context of the green neuron in the grass region includes the surrounding grass regions, while these regions are not important for the semantic interpretation of the red neuron containing the bird’s head.

To better understand how the proposed SCA module helps to improve segmentation accuracy, we show some visual examples from Pascal Context and COCO Stuff in Fig. 5 and Fig. 6, respectively. These examples are also included in the supplementary material with the full legend of label colors. For example, in the top row of Fig. 5 it may be seen that, without considering context, the top part of the bus is mislabeled as ‘car’. The dependency map for a neuron located in that region predicts high dependencies for other neurons in different parts of the bus, which are easier to recognize as such, and thus our method correctly labels the problematic area as ‘bus’. Similarly, in the top row of Fig. 6, part of the playing field is mislabeled as ‘grass’ in the baseline-no result. Neurons in that area are assigned a high dependency with the areas containing the players, which helps our method to correctly label the area as ‘playingfield’.

|                 | PASCAL Context | COCO Stuff |
|----------------|----------------|------------|
|                | PPA | CAA | mIoU | PPA | CAA | mIoU |
| baseline-no    | 71.0% | 51.0% | 39.3% | 59.9% | 41.2% | 27.9% |
| baseline-ave   | 72.1% | 52.1% | 41.1% | 60.8% | 40.5% | 27.6% |
| ours           | 72.8% | 54.4% | 42.0% | 61.6% | 42.5% | 29.1% |

Table 3. Results of ablation study on PASCAL Context dataset and COCO Stuff dataset.
Figure 6. Qualitative results selected in COCO Stuff dataset. We place a yellow circle on a selected region where most pixels inside are misclassified by the baseline-no method while are correctly classified by our method. The leftmost is the dependency mask corresponds to the select region.

Table 4. Comparison with state-of-the-art methods on PASCAL Context dataset.

| Methods            | PPA  | CAA  | mIoU |
|--------------------|------|------|------|
| CFM et al. [4]     | N/A  | N/A  | 31.5%|
| DeepLab et al. [3] | N/A  | N/A  | 37.6%|
| ParseNet et al. [12]| N/A  | N/A  | 40.4%|
| ConvPP-8s et al. [25]| N/A  | N/A  | 41.0%|
| FCN-8s et al. [20] | 67.5%| 52.3%| 39.1%|
| UoA-Context+CRF et al. [9]| 71.5%| 53.9%| 43.3%|
| CRF-RNN et al. [31] | N/A  | N/A  | 39.3%|
| DAG-RNN et al. [24] | 72.7%| 55.3%| 42.6%|
| ours               | 72.8%| 54.4%| 42.0%|

Table 5. Comparison with state-of-the-art methods on COCO Stuff dataset.

| Methods            | PPA  | CAA  | mIoU |
|--------------------|------|------|------|
| FCN et al. [2]     | 52.0%| 34.4%| 22.7%|
| DeepLab et al. [3] | 57.8%| 38.1%| 26.9%|
| FCN-8s et al. [20] | 60.4%| 38.5%| 27.2%|
| DAG-RNN et al. [24] | 62.2%| 42.3%| 30.4%|
| ours               | 61.6%| 42.5%| 29.1%|

6. Conclusion

In this paper, we have presented a neuron-level selective context aggregation module for scene segmentation. The key idea is to augment local representation with global context which is selectively aggregated at each neuron according to its own dependencies, which are predicted on-the-fly by an auxiliary network. The strength of our method stems from the fact that the auxiliary network is jointly trained with the main network without extra supervision. It learns to predict appropriated dependency coefficients for different pairs of neurons to selectively aggregate context by minimizing the overall segmentation loss. This mechanism en-
ables data-driven inference of contextual dependencies, and facilitates context-aware feature learning.

Adding global context to local representation does not necessarily help in many cases. It mainly excels in harder cases, where otherwise local context alone might fail. In our current work, we learned general neuron dependencies. In the future, we consider narrowing down and focusing specifically on long-range dependencies, or possibly cross-image dependencies, where we analyze more than a single image in test time. We believe that learning dependency for co-segmentation is a viable research direction to further improve semantic segmentation capabilities.

References

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[2] H. Caesar, J. R. R. Uijlings, and V. Ferrari. COCO-Stuff: thing and stuff classes in context. *CoRR*, abs/1612.03716, 2016.

[3] 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.

[4] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proc. IEEE CVPR*, pages 3992–4000, 2015.

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

[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[7] M. Jaderberg, K. Simonyan, and A. Zisserman. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[8] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011.

[9] G. Lin, C. Shen, A. van den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proc. CVPR*, pages 3194–3203, 2016.

[10] 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 *Proc. ECCV*, pages 740–755. Springer, 2014.

[11] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Proc. IEEE CVPR*, pages 1972–1979. IEEE, 2009.

[12] W. Liu, A. Rabinovich, and A. Berg. ParseNet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[13] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE CVPR*, pages 3431–3440, 2015.

[14] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE CVPR*, pages 891–898, 2014.

[15] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proc. ICCV*, pages 1520–1528, 2015.

[16] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proc. ICML (3)*, volume 28, pages 1310–1318, 2013.

[17] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, pages 82–90, 2014.

[18] A. Roy and S. Todorovic. Scene labeling using beam search under mutex constraints. In *Proc. IEEE CVPR*, pages 1178–1185, 2014.

[19] A. Sharma, O. Tuzel, and M.-Y. Liu. Recursive context propagation network for semantic scene labeling. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2014.

[20] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[21] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Texton-Boost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[22] B. Shuai, G. Wang, Z. Zuo, B. Wang, and L. Zhao. Integrating parametric and non-parametric models for scene labeling. In *Proc. IEEE CVPR*. IEEE, 2015.

[23] B. Shuai, Z. Zuo, B. Wang, and G. Wang. DAG-recurrent neural networks for scene labeling. In *Proc. IEEE CVPR*, pages 3620–3629, 2016.

[24] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Scene segmentation with DAG-recurrent neural networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017.

[25] S. Xie, X. Huang, and Z. Tu. Top-Down Learning for Structured Labeling with Convolutional Pseudopriors, pages 302–317. Springer International Publishing, Cham, 2016.

[26] J. Yang, B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. In *IEEE CVPR*, June 2014.

[27] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proc. IEEE CVPR*, pages 702–709, June 2012.

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

[29] Y. Zhang and T. Chen. Efficient inference for fully-connected CRFs. In *Proc. IEEE CVPR*, pages 582–589. IEEE, 2012.

[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proc. CVPR*, 2017.
[31] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In Proc. IEEE ICCV, pages 1529–1537, 2015.