Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Weakly Supervised Semantic Segmentation Using Superpixel Pooling Network

Suha Kwak,[1,2] **Seunghoon Hong,**[2] **Bohyung Han**[2]
[1]Department of Information and Communication Engineering, DGIST, Korea
[2]Department of Computer Science and Engineering, POSTECH, Korea
skwak@dgist.ac.kr, {maga33, bhhan}@postech.ac.kr

Abstract

We propose a weakly supervised semantic segmentation algorithm based on deep neural networks, which relies on image-level class labels only. The proposed algorithm alternates between generating segmentation annotations and learning a semantic segmentation network using the generated annotations. A key determinant of success in this framework is the capability to construct reliable initial annotations given image-level labels only. To this end, we propose Superpixel Pooling Network (SPN), which utilizes superpixel segmentation of input image as a pooling layout to reflect low-level image structure for learning and inferring semantic segmentation. The initial annotations generated by SPN are then used to learn another neural network that estimates pixel-wise semantic labels. The architecture of the segmentation network decouples semantic segmentation task into classification and segmentation so that the network learns class-agnostic shape prior from the noisy annotations. It turns out that both networks are critical to improve semantic segmentation accuracy. The proposed algorithm achieves outstanding performance in weakly supervised semantic segmentation task compared to existing techniques on the challenging PASCAL VOC 2012 segmentation benchmark.

Introduction

Semantic segmentation is a computer vision task that assigns a semantic label (*e.g.*, object class) to every pixel in an image. This task is particularly challenging when objects involve substantial appearance variations due to changes in pose, scale and illumination, or objects boundaries are distracted by occlusion and background clutter. Recently, Deep Neural Networks (DNNs) have been extensively studied to tackle this problem, and some algorithms demonstrate impressive performance (Long, Shelhamer, and Darrell 2015; Noh, Hong, and Han 2015; Chen et al. 2015; Vemulapalli et al. 2016; Lin et al. 2016b; Qi 2016)

However, the data-hungry nature of DNNs restricts their applications to semantic segmentation in an uncontrolled and realistic environment. Training DNNs for semantic segmentation demands a large number of pixel-level segmentation annotations. However, the collection of large-scale annotations is dreadfully labor-intensive and it is difficult to maintain good quality of labels in terms of accuracy and consistency. For this reason, existing datasets often suffer from lack of annotated examples and class diversity, and it is not straightforward to learn DNNs for semantic segmentation that can handle various object classes in real world images.

Weakly supervised approaches for semantic segmentation have been proposed to alleviate the challenge (Dai, He, and Sun 2015; Papandreou et al. 2015; Pinheiro and Collobert 2015; Pathak, Krähenbühl, and Darrell 2015). Instead of pixel-level annotations, they make use of weaker annotations such as bounding boxes (Dai, He, and Sun 2015; Papandreou et al. 2015), scribbles (Lin et al. 2016a), and image-level class labels (Pinheiro and Collobert 2015; Pathak, Krähenbühl, and Darrell 2015), all of which are much less expensive to obtain than pixelwise segmentation annotations and readily available in various large-scale datasets such as PASCAL VOC (Everingham et al. 2010) and ImageNet (Russakovsky et al. 2015). The main challenge in training a model based on weak supervision is the step to generate pixelwise label maps from incomplete information through self-supervision. The most popular choice for this task is to employ discriminative objectives for the identification of local image regions relevant to each semantic category. However, although such strategy is useful to roughly localize objects, it often concentrates on small discriminative parts of an object and is not sufficient to cover entire object areas; it leads to poor segmentation performance compared to fully supervised approaches.

We believe that weakly supervised semantic segmentation can be improved by considering low-level structures of individual images and shape information commonly observed in multiple images. We realize this idea using two DNNs with an iterative optimization procedure. Specifically, our approach alternates between 1) generating segmentation annotations and 2) learning a semantic segmentation network using the generated annotations, where the learned network is in turn used to generate annotations for the next round. The success of this framework relies on the capability to generate dependable initial annotations given image-level class labels only. For the purpose, we propose Superpixel Pooling Network (SPN), which employs superpixel segmentation as a pooling layout to reflect low-level image structure for learning and inferring semantic segmentation in a weakly

supervised setting. The initial annotations generated by SPN are then given to learn DecoupledNet (Hong, Noh, and Han 2015), which is the second network in our approach for the final semantic segmentation. By decoupling semantic segmentation into classification and segmentation tasks, DecoupledNet learns class-agnostic shape prior from the initial annotations effectively. Both SPN and DecoupledNet substantially improve segmentation accuracy according to our observation. Our framework outperforms the existing state-of-the-art techniques in weakly supervised semantic segmentation with only a single round training, and its performance is further improved by additional rounds, where annotations are generated by DecoupledNet of the previous iterations. The contribution of our approach is three-fold:

- We propose a novel DNN architecture to exploit superpixels as a pooling layout for generating segmentation annotations with image-level labels only. The proposed SPN is naturally combined with existing DecoupledNet for weakly supervised semantic segmentation.
- To construct robust initial segmentation labels for training, we introduce techniques for segmentation label sanitization and reliable image identification, which are useful to improve the final segmentation performance.
- The proposed algorithm demonstrates impressive performance, and achieves the state-of-the-art accuracy compared to existing approaches with significant margins.

Related Work

The state-of-the-art algorithms on semantic segmentation rely on DNN (Long, Shelhamer, and Darrell 2015; Noh, Hong, and Han 2015; Zheng et al. 2015; Chen et al. 2015; Badrinarayanan, Handa, and Cipolla 2015). These algorithms are built with a classification network pre-trained on a large-scale image collection (Russakovsky et al. 2015). The standard framework of semantic segmentation is to train the encoder and/or decoder networks for pixel-level classification based on ground-truth segmentation masks. Long et al. (Long, Shelhamer, and Darrell 2015) proposed an efficient end-to-end learning framework based on Fully-Convolutional Network (FCN). Later approaches have improved the FCN-style architecture by applying fully-connected CRF (Chen et al. 2015) as post-processing or integrating it as a network component (Zheng et al. 2015). Other alternatives have built deep decoding networks based on deconvolution network (Noh, Hong, and Han 2015; Badrinarayanan, Handa, and Cipolla 2015) to preserve accurate object boundaries. Although these approaches have achieved substantial improvement in performance, there is a critical bottleneck in collecting a large amount of segmentation annotations to learn their DNNs, which limits their practicality for semantic segmentation in the wild.

Weakly supervised approaches (Pinheiro and Collobert 2015; Papandreou et al. 2015; Pathak et al. 2015; Hong, Noh, and Han 2015) have been proposed to resolve the data deficiency issues. In this setting, models for semantic segmentation is trained with only image-level labels (Pinheiro and Collobert 2015; Papandreou et al. 2015; Pathak et al. 2015), scribbles (Lin et al. 2016a), or bounding box (Dai, He, and Sun 2015). To build an association between coarse labels and pixel-level fine annotations, they often employ auxiliary objectives such as a classification loss to obtain coarse localization of semantic categories, which are often refined by Multiple Instance Learning (MIL) (Pinheiro and Collobert 2015; Pathak et al. 2015) or Expectation-Maximization (EM) (Papandreou et al. 2015). However, these approaches tend to localize only few discriminative parts of object because of the missing supervision on segmentation, and perform much worse than fully-supervised methods. To mitigate this issue, (Hong et al. 2016) proposed a transfer-learning approach that exploits segmentation annotations of other object classes, and (Pinheiro and Collobert 2015) investigated the use of various shape priors such as superpixels and object proposals. Also, (Pathak, Krähenbühl, and Darrell 2015) showed that even a very little supervision such as one-bit information about object size improves segmentation performance significantly.

Our approach attempts to bridge the gap between discriminative localization and shape estimation, which is attained by the use of superpixel as a unit for shape estimation (SPN) and integration of a shared segmentation network across multiple object classes (DecoupledNet).

Superpixel Pooling Network

This section describes the architecture of SPN including the details of the superpixel pooling layer, and the learning strategy for SPN. It also discusses how to compute initial segmentation annotations of training images using SPN.

Architecture

SPN is composed of three parts: 1) a feature encoder f_{enc}, 2) an upsampling module composed of a feature upsampler f_{ups} and the superpixel pooling layer, and 3) two classification modules that classify feature vectors obtained from the encoder and the upsampling module. The entire network is learned with the two separate classification losses computed by the last component. Overall architecture of SPN is illustrated in Fig. 1.

The encoder of SPN computes a convolutional feature map $\mathbf{z} = f_{\text{enc}}(\mathbf{x})$ of input image $\mathbf{x}$. As a part of the encoder, we adopt the VGG16 network (Simonyan and Zisserman 2015) pre-trained on ImageNet excluding its fully-connected layers. The parameters of the VGG16 network is fixed throughout. To adapt the encoder to the target task, we add an additional convolutional layer, which is learned from scratch, on the top of the convolutional layers of the VGG16 network. The encoder is thus fully convolutional and outputs a feature map that contains a spatial information.

Given the convolutional feature map $\mathbf{z}$, a straightforward way to estimate object area is to classify every spatial location of the feature map—akin to the sliding-window method in object detection. To learn such classifier jointly with other network parameters, global average pooling (Zhou et al. 2016) or global max pooling (Oquab et al. 2015) is applied to the feature map to convert it into a single feature vector, which is then used to learn classification layers based on classification loss. However, the activation map obtained

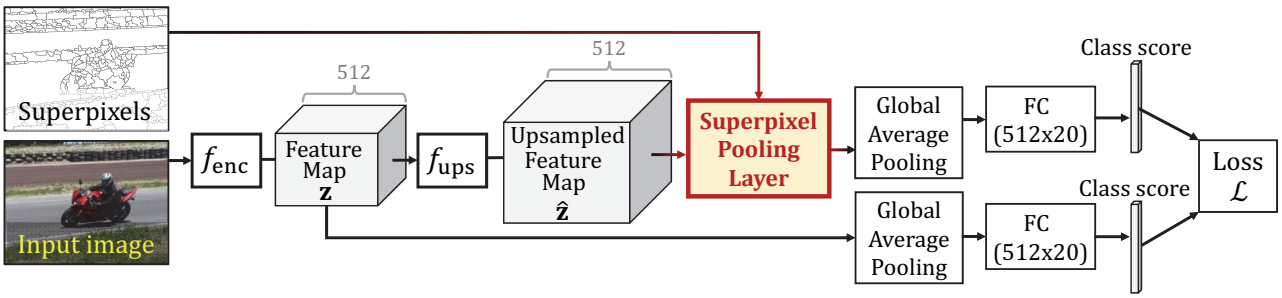

Figure 1: Overall architecture of SPN. SPN takes two inputs for inference: an image and its superpixel map. Given an input image, our network extracts high-resolution feature maps using encoder f_{enc} followed by several upsampling layers f_{ups}, and the superpixel pooling layer aggregates features inside of each superpixel by exploiting an input superpixel map as pooling layout. Then relevance of superpixels to semantic categories is obtained by training a model with discriminative loss in a similar way to (Zhou et al. 2016). Note that we put additional branch of global average pooling for regularization, which prevents undesirable training noises introduced by superpixels.

by this approach is not sufficient for semantic segmentation since it assigns high scores to only few discriminative parts of an object and its resolution is too low to recover object shape accurately.

We design the Superpixel-Pooling (SP) layer to resolve the above issue. Unlike traditional pooling layers, the pooling layout of the SP layer is not pre-defined but determined by superpixels of the input image. Through the SP layer, we can aggregate feature vectors spatially aligned with superpixels by average-pooling. The output of the SP layer then becomes an $N \times K$ matrix, where N means the number of superpixels in the input image and K indicates the number of channels in the feature map ($K = 512$ in the current SPN architecture). Analogous to (Zhou et al. 2016), the $N \times K$ superpixel features are then averaged over superpixels to build a single $1 \times K$ vector, which will be classified by the following fully-connected layer to compute and backpropagate the classification loss.

The simplest way to utilize the SP layer is to directly connect the feature map and the SP layer, but this is not appropriate since the resolution of the feature map is too low; a feature map location may be associated with a large number of superpixels, which leads to overly-smoothed feature vectors of superpixels. We thus add a non-linear upsampling module f_{ups} between the feature map and the SP layer. This module consists of two deconvolution layers (Long, Shelhamer, and Darrell 2015) and one unpooling layer (Zeiler and Fergus 2014) followed by another two deconvolution layers. A batch normalization layer (Ioffe and Szegedy 2015) and a rectified linear unit (Nair and Hinton 2010) are attached after every deconvolution layer. We employ a shared pooling switch (Noh, Hong, and Han 2015) between the last pooling layer of the encoder and the unpooling layer, which is known to be useful to reconstruct object structure in the semantic segmentation scenario. All the parameters of the upsampling module are trained from scratch.

Finally, SPN has a branch that directly applies global average pooling to the feature map $\mathbf{z}$ and classifies the aggregated feature vector. This branch aims at preventing the network from being spoiled by the SP layer and keeping discriminative parts with high activation scores.

Forward and Backward Propagations via SP Layer

This section derives forward and backward propagations through the SP layer. Let $\mathbf{p}_i = \{p_i^k\}_{k=1,\dots,K_i}$ be the i-th superpixel of an image, where p_i^k indicates individual pixels belonging to $\mathbf{p}_i$ and K_i denotes the number of pixels in $\mathbf{p}_i$.

The results of the forward propagation through the SP layer are feature vectors, each of which is average-pooled from the area of the associated superpixel. Let $\hat{\mathbf{z}} = f_{\text{ups}}(\mathbf{z})$ be the upsampled feature map, which is the input of the SP layer. The pooled feature vector of the i-th superpixel is then given by

$$\bar{\mathbf{z}}_i = \frac{1}{K_i} \sum_j \sum_k I(p_i^k \in \mathbf{r}^j)\, \hat{\mathbf{z}}^j = \sum_j h_i^j\, \hat{\mathbf{z}}^j, \qquad (1)$$

where $\mathbf{r}^j$ and $\hat{\mathbf{z}}^j$ represent the receptive field and the feature vector of the j-th location in $\hat{\mathbf{z}}$, respectively. $I(p_i^k \in \mathbf{r}^j)$ is an indicator function that is 1 if the center of $\mathbf{r}^j$ is closer to p_i^k than those of any other receptive fields are, and 0 otherwise. The average of indicator functions over the superpixel is denoted by $h_i^j = \sum_k I(p_i^k \in \mathbf{r}^j)/K_i$, which represents how much area of the i-th superpixel is occupied by the receptive field $\mathbf{r}^j$ (*i.e.*, the responsibility of $\mathbf{r}^j$ for the superpixel $\mathbf{p}_i$). Through global average pooling over all superpixels, we obtain a single feature vector for the input image, which is given by

$$\tilde{\mathbf{z}} = \frac{1}{N} \sum_i \bar{\mathbf{z}}_i = \frac{1}{N} \sum_i \sum_j h_i^j\, \hat{\mathbf{z}}^j, \qquad (2)$$

where N indicates the number of superpixels. The image-level feature vector $\tilde{\mathbf{z}}$ is then classified by the fully connected layer following the SP layer, and the result is fed to the loss function $\mathcal{L}$.

In the backward propagation, the gradient of $\hat{\mathbf{z}}^j$ of the input feature map is derived as

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{z}}^j} = \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{z}}} \frac{\partial \tilde{\mathbf{z}}}{\partial \hat{\mathbf{z}}^j} = \frac{1}{N} \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{z}}} \frac{\partial \sum_i \sum_{j'} h_i^{j'} \hat{\mathbf{z}}^{j'}}{\partial \hat{\mathbf{z}}^j} = \frac{1}{N} \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{z}}} \sum_i h_i^j. \tag{3}$$

Note that although individual input images in a mini-batch may have different numbers of superpixels, the forward and backward propagations do not depend on the number of superpixels due to the average pooling over all superpixels per image.

Learning SPN

Loss function. SPN is learned with classification losses as only image-level class labels are available in our setup. Since multiple objects of different classes may appear in an image, given C object classes, our loss function is thus defined as the sum of C binary classification losses as shown in

$$\mathcal{L}(f(\mathbf{x}), \mathbf{y}) = \frac{1}{C} \sum_{c=1}^{C} \left\{ y_c \log \frac{e^{f_c(\mathbf{x})}}{1 + e^{f_c(\mathbf{x})}} + (1 - y_c) \log \frac{1}{1 + e^{f_c(\mathbf{x})}} \right\}, \tag{4}$$

where $f_c(\mathbf{x})$ and $y_c \in \{0, 1\}$ are the network output and the ground-truth label for a single class c, respectively. Note that the SPN outputs two class score vectors at the end of the two branches (Fig. 1), and they are treated independently by the identical loss function.

Multi-scale learning. Objects are depicted with different scales in images. To better model such scale variations of objects, we follow the approach of (Oquab et al. 2015) that randomly resizes images in the input mini-batch during training. Specifically, we first make images square by padding zeros to its shorter dimension, and rescale them randomly to one of the 6 predefined sizes: 250^2, 300^2, 350^2, 400^2, 450^2, and 500^2 pixels. Note that our SPN consists only of convolutional and global pooling layers except the SP layer, so the network is naturally fit to this multi-scale approach if the SP layer could work with images of different sizes. To this end, we compute the responsibilities of the receptive field (*i.e.*, h_i^j's in Eq. (1)) for all 6 possible input resolutions in advance per image. We observed empirically that this multi-scale approach helps to estimate object area more accurately.

Generating Initial Annotations with SPN

Superpixel-pooled class activation map. SPN assigns a feature vector to each superpixel through the SP layer as described in Eq. (1). During inference, the feature vector of each superpixel is first given to the fully-connected classification layer following the SP layer, and the class scores of the individual superpixels are computed. As a result, we obtain a tensor in $\mathbb{R}^{W \times H \times C}$, where W and H are the width and height of the input image, respectively, and each channel corresponds to an activation map for the associated class. As in training, an input image is rescaled to the 6 predefined sizes, and 6 activation tensors are computed accordingly. Finally, the 6 tensors are aggregated by max-pooling. We refer to the aggregated tensor as Superpixel-Pooled Class Activation Map (SP-CAM).

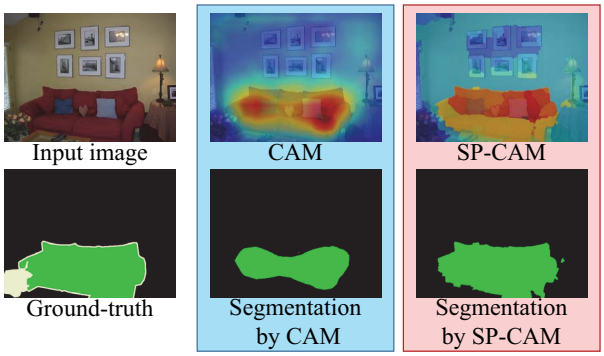

Figure 2: Qualitative comparison of CAM (Zhou et al. 2016) and our SP-CAM with initial segmentation results obtained by thresholding them. CAM is computed by classifying every location of the feature map $\mathbf{z}$ with the fully-connected layer directly connected to $\mathbf{z}$. CAM could not cover the entire object area due to its localized activations on limited resolution. Class activations in SP-CAM preserves object shape more accurately by employing superpixel as a unit for shape estimation.

SP-CAM is motivated by Class Activation Map (CAM) of (Zhou et al. 2016), but has a critical advantage. Unlike CAM, where each location of the feature map is activated independently, SP-CAM assigns class activations to individual superpixels, which allows to generate class activation scores in the original resolution with image structures preserved. This property of SP-CAM is particularly useful for semantic segmentation as illustrated in Fig. 2.

Generating initial annotation with SP-CAM. We obtain initial segmentation annotations of training images through their SP-CAMs. The activation map of each class in SP-CAM is thresholded by 50% of the maximum score of the map; in other words, pixels whose activation scores are below the threshold are ignored. Furthermore, we disregard the activation maps of the classes unmatched with the image-level ground-truth labels. We call this step *annotation sanitization*. Note that using image-level labels is not unfair since our goal is not the inference of segmentation labels but the construction of segmentation annotations of training images given the labels for image-level classification. The segmentation annotation of the SP-CAM is then given by searching for the label with the maximum activation score in every pixel. If the activation scores of a pixel are below a predefined threshold in all the C channels, it is considered as background. An example of initial annotation is illustrated in Fig. 2.

Iterative Learning of DecoupledNet

The output of SPN already produces decent segmentation results thanks to the use of superpixel as a unit for shape estimation. However, SPN is still not free from the limitation

of discriminative learning; it tends to exaggerate the class scores of discriminative superpixels, thus some parts of an object could be lost when initial annotations are generated by the procedure described above. A solution to resolve this issue would be to get hints from annotations of other images, and this idea can be realized by learning class-agnostic segmentation knowledge from a number of initial segmentation annotations. We adopt DecoupledNet (Hong, Noh, and Han 2015) as the network that learns such segmentation knowledge from the initial annotations and refine them iteratively.

DecoupledNet decomposes the semantic segmentation problem into two separate tasks of classification and segmentation, and is composed of two decoupled networks to handle the two tasks. Following (Hong, Noh, and Han 2015), the classification network of DecoupledNet is trained with image-level labels and serves as a feature encoder in our framework. Unlike the original setting, however, its segmentation network is now trained with generated noisy annotations instead of ground-truth segmentations. Since the binary segmentation network is shared across different object classes, DecoupledNet can learn class-agnostic segmentation knowledge from annotations of multiple object classes, which allows it to generate more accurate annotations for the next round of training.

In each round of our algorithm, DecoupledNet is learned from generated annotations, which are provided by SPN at the first round and by DecoupledNet trained in the previous iteration from the second round. We learn the segmentation network of DecoupledNet from scratch at every round to avoid annotation biases of the previous round. Note that only a subset of reliable annotations are used to learn the network to minimize undesirable effects by incomplete and noisy annotations. To this end, we define the reliability of an annotation based on *the degree of scatter*, which is the ratio of the squared perimeter to the area of segmentations in the annotation. Only a subset of least scattered annotations are then selected for training, while assuming that less scattered annotations are more reliable. Note that DecoupledNet is well suited to this *reliable subset selection* since the network has shown superior performance even with only a limited number of segmentation annotations given in training. We observed empirically that DecoupledNet learned with a subset of reliable annotations consistently outperforms its counterpart learned with all annotations. The trained DecoupledNet is in turn used to generates annotations for the next round. As in the case of SPN, the generated annotations are sanitized by image-level labels to remove irrelevant or unreliable segments in the annotations.

The above procedure is repeated until the predefined number of iterations is reached. The DecoupledNet trained at the final round is considered as our model for semantic segmentation.

Experiments

This section describes implementation details, and demonstrates the effectiveness of our approach in PASCAL VOC 2012 segmentation benchmark (Everingham et al. 2010) with comparisons to the-state-of-the-art techniques for weakly supervised semantic segmentation. As an evaluation metric, we adopt segmentation accuracy defined by intersection over union between ground-truth and predicted segmentation.

Implementation Details

Both of SPN and DecoupledNet are trained on PASCAL VOC 2012 dataset (Everingham et al. 2010). Besides the provided image sets for the semantic segmentation task, we employ additional images used in (Hariharan et al. 2011) to enlarge training set. In total, 10,582 images are used to train the networks, and the validation set of 1,449 images is kept for evaluating our approach. Superpixels of the images are computed by (Zitnick and Dollár 2014).

SPN is implemented in Torch7 (Collobert, Kavukcuoglu, and Farabet 2011). The network parameters are optimized by Adam (Ba and Kingma 2015) with an initial learning rate of 0.001. The optimization converges after approximately 9K iterations with mini-batches of 12 images. The training procedure takes about 3 hours on a single Nvidia TITAN X GPU with 12Gb RAM in our experiment.

We learn DecoupledNet for two rounds. When learning DecoupledNets, we follow the original setup described in (Hong, Noh, and Han 2015), and the number of iterations is set to 9K for both rounds. As a training set, top 300 most reliable annotations are selected per class at each round.

Evaluation on PASCAL VOC Benchmark

Analysis of generated annotation. We first analyze and evaluate annotations generated by our algorithm at each round. Details of the results are summarized in Fig. 3. It turns out that both of the sanitization and reliable subset selection are essential to improve the quality of annotations and in turn learn a better segmentation network.

We also compare SPN with CAM (Zhou et al. 2016) in terms of the annotation quality. Without reliable subset selection, annotations generated by SPN achieves 43.8% in average accuracy while those by CAM shows 30.5%. This result demonstrates the effectiveness of the superpixel pooling layer of SPN.

Comparison to other methods. Our algorithm is compared with up-to-date weakly supervised approaches. In addition to our two segmentation networks (Ours:Rnd1, Ours:Rnd2), we also evaluate SPN as a semantic segmentation network (Ours:SPN). SPN predicts semantic segmentation in the same way as it generates initial annotations, but makes use of its classification scores instead of image labels to ignore irrelevant classes in its segmentation result. Specifically, SPN disregards object classes whose activation scores averaged over all superpixels are below 0. We also report the accuracy of the DecoupledNet trained with all available ground-truth annotations (Upper-bound) as the upper-bound of our algorithm.

The segmentation results on PASCAL VOC 2012 validation set are quantified and compared in Table 1. SPN by itself outperforms all the previous approaches except MIL using segmentation proposals (MIL+Seg), which is the best one among them. Note that segmentation proposals used in

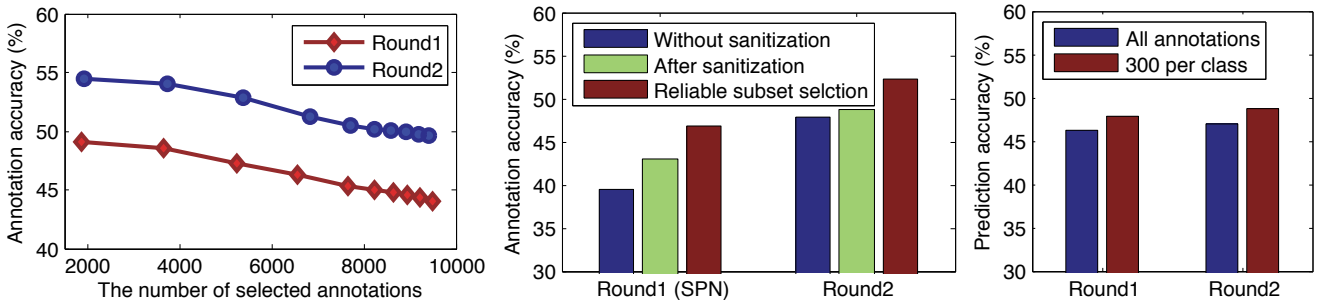

Figure 3: Analysis of our annotation generation technique on PASCAL VOC 2012 segmentation benchmark. **(a) Empirical justification for our reliability metric.** We show the average segmentation accuracy of annotations selected in the decreasing order of reliability for training images. The average accuracy consistently decreases by increasing the number of selected annotations, which indicates that the reliability metric is well correlated with the actual quality of annotation. **(b) Effects of the annotation sanitization and reliable subset selection.** In both rounds, the accuracy of annotations is enhanced by the sanitization, and further improved by selecting 300 most reliable annotations per class in training set. **(c) Effect of the reliable subset selection.** We trained two networks, one with all of the sanitized annotations and the other with 300 most reliable annotations per class, and compare their segmentation prediction accuracy in validation set. The network learned with the reliable subset of annotations was better than its counterpart in both rounds.

Table 1: Accuracy on PASCAL VOC 2012 validation set

	Mean Acc.
Ours:SPN	40.0 %
Ours:Rnd1	48.6 %
Ours:Rnd2	**50.2** %
CCNN (Pathak, Krähenbühl, and Darrell 2015)	35.3 %
EM-Adapt (Papandreou et al. 2015)	33.8 %
MIL+Spx (Pinheiro and Collobert 2015)	36.6 %
MIL+Seg (Pinheiro and Collobert 2015)	42.0 %
Upper-bound (Hong, Noh, and Han 2015)	67.5 %

Table 2: Accuracy on PASCAL VOC 2012 test set

	Mean Acc.
Ours:Rnd2	**46.9** %
CCNN (Pathak, Krähenbühl, and Darrell 2015)	35.6 %
EM-Adapt (Papandreou et al. 2015)	39.6 %
MIL+Spx (Pinheiro and Collobert 2015)	35.8 %
MIL+Seg (Pinheiro and Collobert 2015)	40.6 %
Upper-bound (Hong, Noh, and Han 2015)	66.6 %

MIL+Seg are more expensive and powerful evidences to recover object shapes than superpixels of SPN, and SPN outperforms MIL when it makes use of superpixels (MIL+Spx) as SPN does. Also, with only a single round of DecoupledNet training, our system outperforms MIL+Seg by a large margin, and an additional round further improves the performance. The same tendency is shown in the results on PASCAL VOC 2012 test set as summarized in Table 2.

Qualitative results on PASCAL VOC 2012 validation set are presented in Fig. 4, where results of our system (*Round1* and *Round2*) are compared with those of SPN after sanitization (*Initial annotation*). The figure shows that, over iterations, missing parts are recovered and false alarms are suppressed.

Conclusion

We have proposed a new weakly supervised semantic segmentation algorithm based on Superpixel Pooling Network (SPN). SPN takes advantage of underlying image structure by employing a superpixel map as a pooling layout, which is especially useful for semantic segmentation in a weakly supervised setting. The segmentation results by SPN is then used as pixel-wise segmentation annotations for learning DecoupledNet, which learns class-agnostic segmentation knowledge from the annotations to further improve segmentation results. To alleviate side effects introduced by noisy and incomplete annotations, we also proposed techniques to sanitize the annotations and measure their reliability. The proposed algorithm demonstrated substantially improved performance over previous arts on a challenging benchmark dataset.

Acknowledgement

This work was supported in part by DGIST Faculty Start-up Fund (2016080008), IITP grant (B0101-16-0307; Machine Learning Center, B0101-16-0552; DeepView), and NRF grant (NRF-2011-0031648, Global Frontier R&D Program on Human-Centered Interaction for Coexistence), which are funded by the Korean government (MSIP), NSF CAREER grant IIS-1453651, and ONR grant N00014-13-1-0762.

References

Ba, J., and Kingma, D. P. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Badrinarayanan, V.; Handa, A.; and Cipolla, R. 2015. SegNet: a deep convolutional encoder-decoder architecture

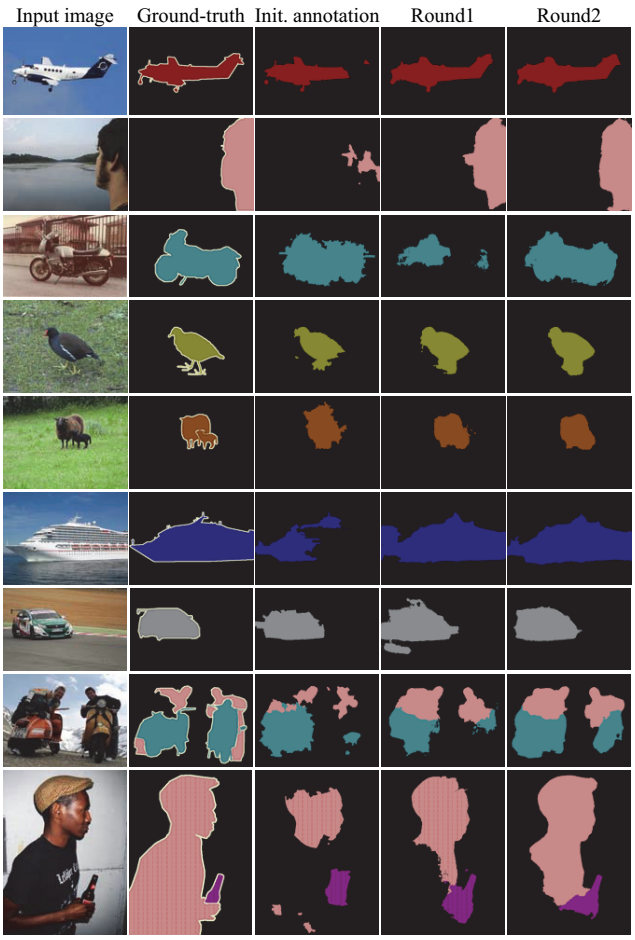

Figure 4: Results on PASCAL VOC 2012 validation set.

for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2015. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*.

Collobert, R.; Kavukcuoglu, K.; and Farabet, C. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

Dai, J.; He, K.; and Sun, J. 2015. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV* 88(2):303–338.

Hariharan, B.; Arbelaez, P.; Bourdev, L.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. In *ICCV*.

Hong, S.; Oh, J.; Lee, H.; and Han, B. 2016. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *CVPR*.

Hong, S.; Noh, H.; and Han, B. 2015. Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J. 2016a. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*.

Lin, G.; Shen, C.; van dan Hengel, A.; and Reid, I. 2016b. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.

Noh, H.; Hong, S.; and Han, B. 2015. Learning deconvolution network for semantic segmentation. In *ICCV*.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2015. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *CVPR*.

Papandreou, G.; Chen, L. C.; Murphy, K. P.; and Yuille, A. L. 2015. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*.

Pathak, D.; Shelhamer, E.; Long, J.; and Darrell, T. 2015. Fully convolutional multi-class multiple instance learning. In *ICLR Workshop*.

Pathak, D.; Krähenbühl, P.; and Darrell, T. 2015. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*.

Pinheiro, P. O., and Collobert, R. 2015. From image-level to pixel-level labeling with convolutional networks. In *CVPR*.

Qi, G.-J. 2016. Hierarchically gated deep networks for semantic segmentation. In *CVPR*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 1–42.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Vemulapalli, R.; Tuzel, O.; Liu, M.-Y.; and Chellapa, R. 2016. Gaussian conditional random field network for semantic segmentation. In *CVPR*.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

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

Zhou, B.; Khosla, A.; A., L.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*.

Zitnick, C. L., and Dollár, P. 2014. Edge boxes: Locating object proposals from edges. In *ECCV*.