Few-shot Learning for Segmentation

Chia Dai

CMU-RI-TR-19-35

May 16, 2019

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Martial Hebert, *chair*
Jean Oh
Deva Ramanan
Allison Del Giorno

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my family and friends.

Abstract

Most learning architectures for segmentation task require a significant amount of data and annotations, especially in the task of segmentation, where each pixel is assigned to a class. Few-shot segmentation aims to replace large amount of training data with only a few densely annotated samples. In this paper, we propose a two-branch network, FuseNet, that can few-shot segment an input image, i.e. query image, given one or multiple images of the target domain, i.e. support images. FuseNet preserves the local context around the target domain by masking out the non-target region in the feature space. The network then leverages the cosine similarity between the masked features from the support and the feature from the query as guidance to predict the segmentation mask. In the case of few-shot, we weigh such guidance differently according to their image-level feature similarity with the query. We also explore the quantitative effects of number of support images on Intersection over Union(IoU). Our network achieves the state-of-the-art result on PASCAL VOC 2012 for both one-shot and five-shot semantic segmentation.

Acknowledgments

I want to thank my family and friends for supporting me through my master and enriching my life with all kinds of craziness. MSR has been a journey of knowledge, fun and fulfilling work.

Contents

List of Figures

List of Tables

Chapter 1

FuseNet: Few-shot Learning for Semantic Segmentation using Feature Conditioning

1.1 Introduction

Current breakthroughs in deep neural networks, in particular convolutional neural networks (CNN) have led to a series of model variants such as U-net [22], FCN [19], and Mask R-CNN [11]. These models enabled convenient end-to-end feature extraction and performed exceptionally well on various visual tasks such as classification, recognition, detection and segmentation. However, these models require a large amount of data to learn a new class, which in the real-world translates to significant time and effort of collecting and annotating samples. This problem becomes more serious in the tasks of segmentation, where dense labels are harder to annotate. Moreover, other than the lack of annotations, a bigger issue lies in the long tail distribution of real-world images, where many classes of interest simply do not have sufficient data and constitute the heavy tail.

Several approaches have been proposed to learn from insufficient data and label. One direction is to reduce annotation time by using weak supervision signal such as image-level label [13, 15], bounding boxes [7, 14], and points [2], where the annotator

simply selects a pixel for the object. Even though these algorithms reduce annotation time, they still require a large pool of training samples. Moreover, these methods require some manual fine-tuning and can easily overfit the training samples yet remain insensitive to new incoming data, yielding great performance on fine-tuned classes but still poor result on classes with few training samples. Therefore, we employ one-shot segmentation to directly tackle the issues of limited data, annotations and manual parameter tuning.

The goal of few-shot segmentation is to predict a binary mask of an unseen class given a few pairs of support and query images containing the same unseen class and the binary ground truth masks for the support images. One simple approach is to fine-tune the pre-trained segmentation network. However, such technique is prone to overfitting due to potentially millions of parameter updates. In addition, tuning often resorts to heuristics which can be hard to determine. In contrast, meta-learning [6, 16, 20, 28] abstracts such parameter tuning away by letting the meta-learner network infer model parameters for the learner network. In classification, meta-learning often works well with k nearest neighbor to predict the image-level label without changes to the model parameter given a small set of samples containing the target class.

Inspired by the network structure of meta-learning, [24] proposed OSLSM, a two-branch model adapted from the siamese classification network [16], to perform dense segmentation. The model consists of a segmentation network for predicting a binary mask of the target class from the query, and a conditioning network for generating parameters for the logistic regression layer. Built on the two-branch model, co-FCN [21] modified the conditioning network to generate features instead of parameters to guide the segmentation mask. The assumption is that the target region in the support and query images should have similar appearance, therefore, similar features. However, co-FCN simply element-wise multiply feature maps from two branches to guide the segmentation, leaving the rest for the model to optimize. Moreover, co-FCN only implicitly encoded the ground truth by concatenating the binary mask in the input.

Our proposed network, FuseNet, reduces the model redundancy by combining two networks, one from conditioning branch and the other segmentation branch, into one shared base network similar to hard parameter sharing in [16]. With fewer parameters,

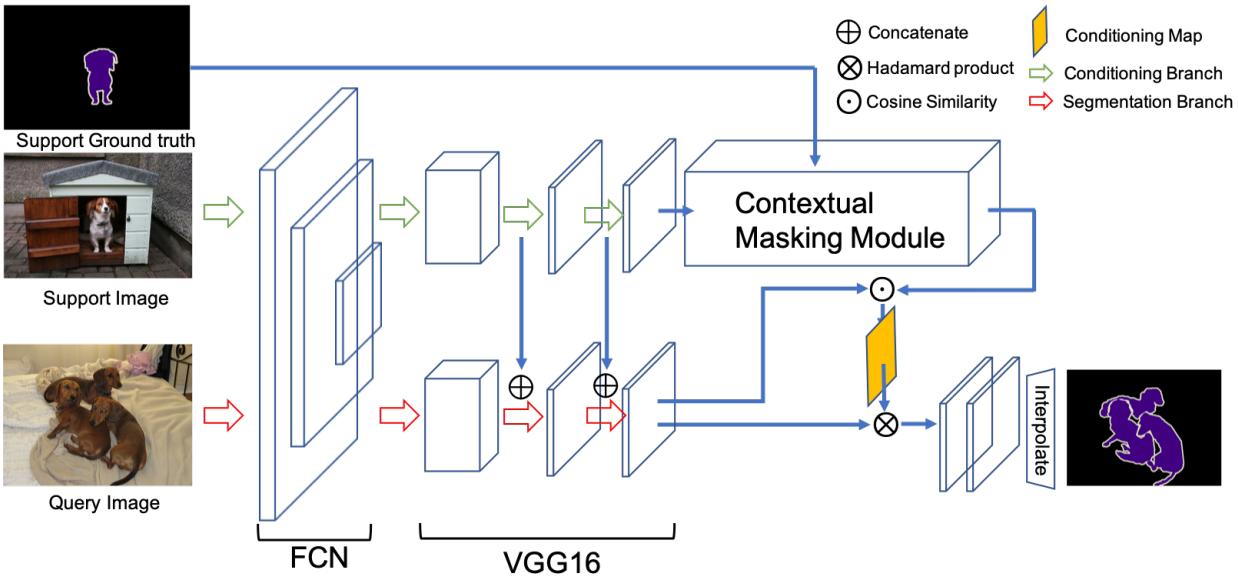

Figure 1.1: FuseNet: overall simplified architecture for one-shot segmentation, consisting of segmentation and conditioning branches

the network is less proned to overfitting and the training converges faster. In addition, we learn the target class feature by desigining a contextual masking module for extracting a shallow linear representation of the target region. Inspired by [18], we enforce the explicit feature conditioning by filtering with the target class feature, similar to a 1×1 convolution, instead of the indirect guidance and optimization in OSLSM and co-FCN [21, 24]. We use the same network for testing on one-shot and few-shot settings. In the case of few-shot, we fuse multiple support features by exploiting their global feature similarity with the query feature. Our overall architecture is shown in Figure 1.1, with a more detailed illustration of contextual masking and few-shot fusion technique in Figure 1.2. Our network achieves a mean IoU of 46.4% on PASCAL VOC 2012 4-fold cross-validation for one-shot segmentation and 50.2% for five-shot segmentation, both outperforming the current state-of-the-art result.

1.2 Related Work

Semantic segmentation aims to classify each pixel of an image to a pre-defined class. The dense annotation can be used as an interpretable feature for downstream applications such as path planning and scene understanding. Fully Convolutional Network (FCN) [19] is a major building block for segmentation and uses only convolutional layers to preserve relative pixel positions in features of all scales. By discarding fully-connected layers, FCN reduces model parameters, accommodates input images of arbitrary size, and becomes more robust due to less sensitivity to overfitting. Subsequently, U-Net [22], Chen [4] and DeepLabv3+ [5] used the fully-convolutional concept to improve IoU on various segmentation datasets, including PASCAL [8]. Additionally, He [11] and Hariharan [10] proposed to simultaneously perform object detection and segmentation in the bounding box, both within a unified network.

Weakly-supervised segmentation uses image classification label and rough boundary scribble as annotations in training to segment images. Due to significant cost in acquiring dense annotation, Huang [13] proposed to train the network to segment by starting from discriminative region and growing progressively. Furthermore, Zhou [32] and Zhang [29, 30] showed how the convolutional layer identifies the region for object of interest by localizing its feature from image-level label. Our contextual masking and few-shot fusion technique are inspired by similar local feature correspondence and global feature exploitation. Lin et al. [17] presented scribble line annotation methodology and the network learns to match the unlabeled pixel to the closest scribble line pixels in spectral distance. Tang et al. [27] extended the scribble line matching by designing the normalized cut loss to optimize for consistency across all pixels within the mask as it grows and shrinks, as opposed to growing from the seed with cross-entropy loss.

The few-shot learning problem aims to solve the general pattern recognition by a few labeled examples, defined as the support set, through learning based approach. The difficulty lies in the generalization to the unseen domain while maintaining the accuracy, which are typically a tradeoff. The discriminative approach learns to differentiate the pre-defined domain from the target domain. Siamese network [16] learns the differences by learning the embedding space to maximize inter-class distance, optimized through energy loss. In contrast, Vinyals [28] focused on recognizing the

target class by augmenting memory into the neural network and jointly optimize for the support set embedding via metric learning. Similarly, Annadani [1] explicitly model the semantic relation as attributes in the embedding space for zero-shot learning. Finn et al. [9] used meta-learning to optimize the manifold of model parameter evolution and learned the more transferable internal representation across classes.

In few-shot dense segmentation, OSLSM [24] designed a two-branch network, where one branch regress the parameter weights for the other branch to segment the query images. Similar to auto fine-tuning in OSLSM, the more recent co-FCN [21] also employed the two-branch setup, differing in that co-FCN uses spatial feature extracted from the support images to directly guide the query feature. Zhang [31] proposed similar idea to our network without the exploitation of image-level feature, which is commonly used as coarse guidance for identifying the target region [13].

1.3 Problem Setup

The goal of one-shot and few-shot segmentation is to produce a binary segmentation map, $\hat{M}_{query}$, of the target region in an unseen query image given one or multiple images containing the target class, called the support set, along with their corresponding ground truth masks. Let a set of classes in a dataset be, C. We partition C into a set of training classes, C_{train}, and a set of testing classes, C_{test}, where the two sets are disjoint. We define a sample as a tuple of an image and a binary mask, (I, M), where M contains only one class. We then construct a training set, D_{train}, out of the set of tuples whose mask class is in C_{train} and similarly for a testing set, D_{test}, whose mask class is in C_{test}. In training, for each iteration, we sample one tuple as our query and another tuple of the same class as our support from D_{train}. Similarly in testing, we sample one tuple as our query and k tuples of the same class to form the support set, where k represents the number of shots. We use uniform sampling for all procedures.

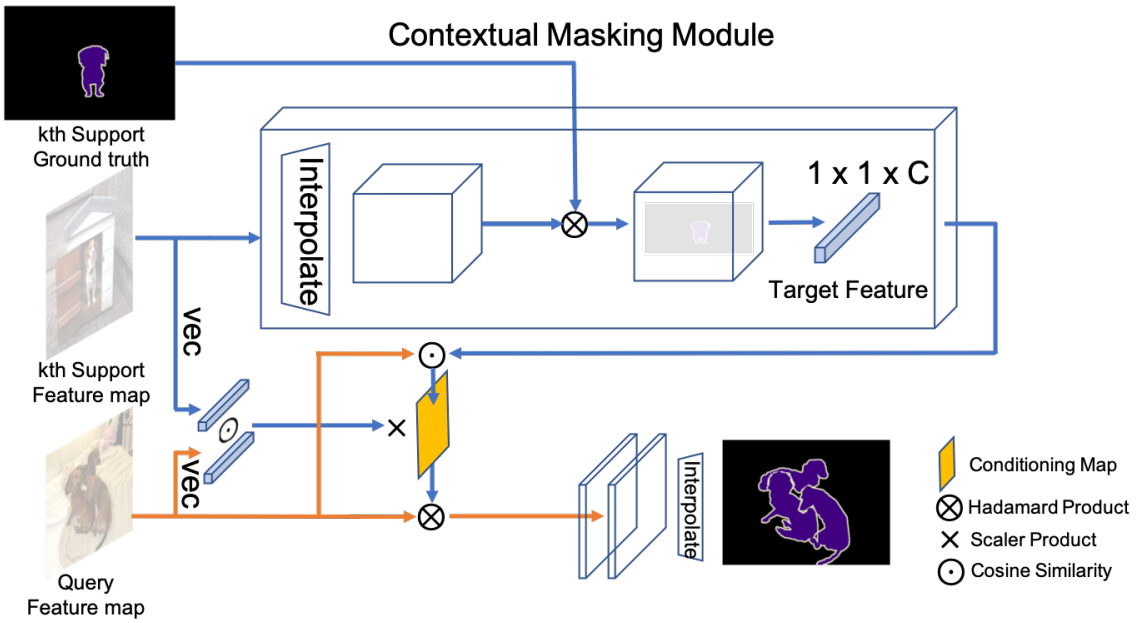

Figure 1.2: Fusion Module: detailed illustration of multiple support images. The support and query feature maps come from FCN and VGG pipeline before, which are not shown. Note that since the normalized fusion weight is 1, i.e. $\hat{s_1} = 1$, we use the same pipeline of few-shot fusion for the one-shot setup as well.

1.4 Proposed Method

Our porposed method is shown in Figure 1.1. We employ the two-branch network [21, 24], consisting of the conditioning branch for extracting the support image feature and the segmentation branch for segmenting the query image.

1.4.1 Shared Network

Initially as the query and the support images are passed into FuseNet, the same fully convolutional network (FCN) [19] produces feature tensors of the same size, which are passed into two separate VGG16 [25] layers, creating two branches. By using one FCN for both branches, we reduce the risk of overfitting and gain the benefit of fewer parameters, along with faster convergence and the ability to use deeper network. Since the query and support come from the same dataset, we hypothesize that it is desirable for both branches to largely share the same network. In addition, inspired by [12, 22, 26], we also concatenate the support feature with the query feature as new channels in the last two layers of VGG16 [25] to add information flow from the conditioning branch to the segmentation branch. We choose feature concatenation over summation because [12] argues that the feature summation would harm the gradient flow. We only use the convolution and ReLU activation without any fully-connected layers to preserve relative pixel location for the contextual masking operation.

1.4.2 Contextual Masking Module

The contextual masking module aims to produce a linear target detector, represented as a target feature in Figure 1.2, from one support image-mask tuple. We design the module to mask the support in feature space in order to preserve the local context around the target region and use as guidance in the later operation. The support feature map, produced by the conditioning branch of VGG16, is passed into the contextual masking module to compute the target feature. Inside the module, we first bilinearly interpolated the support feature map into $w \times h \times c$ feature tensor, where w and h are the width and the height of the input image. We element-wise multiply the mask of the same size with each channel of the feature tensor to produce

the masked tensor. Lastly, we use an average pooling to create a single value for each channel to obtain the target feature of size $1 \times 1 \times c$.

Mathematically, let v_i represents the ith value of the target feature, v, then

$$v_i = \frac{\sum_{x,y} M \otimes F_i}{\sum_{x,y} M}$$

where $\otimes$ is element-wise matrix multiplication, M is the ground truth binary map of size $w \times h$, and F_i is the ith slice of support feature tensor after the bilinear interpolation.

In comparison, OSLSM [24] masks out background from the support image at input. Such early masking leaves only the target region to extract feature from and creates an unnatural RGB image without using boundary information. In co-FCN [21], they concatenate the mask with the RGB image and let the network optimize without any use of explicit masking for focusing on area of attention. In contrast, the target feature we generate from average pooling is an explicit linear class detector. The vector encodes not only the object region information but also the local context around the region. We inject the direct guidance by using the target feature to compute the conditioning map for the query.

1.4.3 Few-Shot Fusion

To extend to few-shot segmentation, previous work like co-FCN [21] proposed to simply average the support features and OSLSM [24] proposed to take the union of all output binary masks due to high precision and low recall, i.e. very few false positives in each output mask. These two fusion heuristics effectively weigh all support images equally and do not consider any coupling or correlation between the query and the support. Instead of fusion by heuristics, we design a few-shot fusion technique that couples the query with each support by their feature distance measured by cosine similarity. We combine multiple target features, each from one support tuple, by a normalized weighted sum. Each target feature is a vector representation of the target region from a support image. We exploit the global similarity between the query and each support to weigh the corresponding target feature accordingly. Given a support

set and a query image, we define the normalized fusion weight for each target feature as the support score.

The support score is computed from global feature maps and represents the image level similarity in appearance between the query and the support. We normalize the score over all supports and compute the weighted sum of all target features as our final target feature. Benefited from the feature concatenation in VGG16 module, we avoid the issue of numerical instability in computing cosine similarity for high dimensional features.

Suppose we have K support tuples, to obtain the weighted sum of all K target features, we couple the query and the support by computing the global feature cosine similarity, i.e.

$$s_j = \frac{f^{query} \cdot f_j^{support}}{||f^{query}|| \cdot ||f_j^{support}||}$$

$$\hat{s_j} = \frac{s_j}{\sum_{j=1}^{K} s_j}$$

where $\hat{s_j}$ is the support score for jth target feature given the support set $(f_1^{support} ... f_K^{support})$, f^{query} is the vectorized query feature map from VGG, and $f_j^{support}$ is the jth vectorized support feature map. Since $\hat{s_j}$ is normalized, we use the same pipeline for one-shot as well, where $\hat{s_j} = 1$. The weighted sum of all K target features is simply

$$\hat{v} = \sum_{j=1}^{K} \hat{s_j} \cdot v_j$$

The underlying assumption is that the targets in images of similar appearance, potentially from similar perspectives and background, would also be visually more similar. Our few-shot fusion technique enables the network to more heavily condition the query feature map with globally similar image from the support set.

1.4.4 Conditioning Map

Given the fused target feature, we proceed to generate a conditioning map, which represents the pixel-level similarity map between the query feature and the target class. In a way, the conditioning map serves as a heat map or attention map to

Dataset	Test classes
PASCAL-A	aeroplane, bicycle, bird, boat, bottle
PASCAL-B	bus, car, cat, chair, cow
PASCAL-C	dining table, dog, horse, motorbike, person
PASCAL-D	potted plant, sheep, sofa, train, tv/monitor

Table 1.1: 4-fold cross validation for PASCAL VOC 2012

down weigh the irrelevant feature region and activate the target region in the query. The process of generating conditioning map is analogous to reversing the contextual masking, where we want to use the target feature to generate a "soft mask" for the query feature.

First, we bilinearly interpolate the query feature to $w' \times h' \times c$ feature tensor. To compute the conditioning map, let the target feature be $\hat{v}$, and F^{query} be the query feature tensor. Then the conditioning map is

$$C_{x,y} = \frac{\hat{v} \cdot F^{query}_{x,y}}{||\hat{v}|| \cdot ||F^{query}_{x,y}||}$$

where $F^{query}_{x,y}$ is a $1 \times 1 \times c$ feature vector across all channels at pixel (x, y), and $C_{x,y}$ is the similarity score to the target class at query position (x, y).

Note that the process of taking cosine similarity resembles a 1×1 convolution with a $1 \times 1 \times c$ filter, i.e. class detector. It condenses the previously interpolated feature tensor back to $w' \times h'$. Once we obtain the conditioning map for the query, we inject direct guidance to the query segmentation by element-wise multiplying the conditioning map with the query feature map from VGG. Finally, the guided feature map goes through two convolutional layers, a bilinear interpolation to resize to $w \times h$, and a probability thresholding to produce a binary segmentation map.

1.5 Implementation

For benchmarking against previous work, we employ VGG16 [25] as our base model for both segmentation and conditioning branch, identical to OSLSM and co-FCN. We use the pre-trained weights from ILSVRC [23] and crop the RGB images from PASCAL VOC 2012 [8] to $224 \times 224 \times 3$. The network uses the same encoding

Methods(1-shot)	PASCAL-A	PASCAL-B	PASCAL-C	PASCAL-D	Mean
1-NN	25.3	44.9	41.7	18.4	32.6
LogReg	26.9	42.9	37.1	18.4	31.4
Siamese	28.1	39.9	31.8	25.8	31.4
Fine-tuning	24.9	38.8	36.5	30.1	32.6
OSLSM	33.6	55.3	40.9	33.5	40.8
co-FCN	36.7	50.6	44.9	32.4	41.1
Ours(concat)	36.8	51.1	43.8	33.1	41.2
Ours(Input-Mask)	37.4	53.2	44.3	33.5	42.1
Ours	**40.3**	**58.6**	**47.9**	**38.7**	**46.4**

Table 1.2: Mean IoU for one-shot semantic segmentation given class partition from Table 1.1

structure as FCN [19] with 3 max pooling layers to downsample to 28×28. In both the segmentation and conditioning branches, we strip the max pooling layer to preserve feature resolution and only use conv4 and conv5 from VGG16. We concatenate support features from conv4 and conv5 with the query features as new channels. In contextual masking module, we employ bilinear interpolation to resize the support feature into 224×224 and follow the procedure described in 1.4.2. We use kernels of size 3×3 for all convolutional layers except the last one before the bilinear interpolation, where 1×1 convolution is used for creating two channels of binary mask, i.e. target class and background. The bilinear interpolation restores the mask size to 224×224 and a probably threshold of 0.5 is applied to obtain the binary mask. The training parameters with SGD are as follows: learning rate of $1e^{-5}$, decay of $1e^{-3}$ and momentum of 0.9. In training, We use batch size of 1, where one sample consists of two tuples of the same class for one iteration. We do not explicitly optimize for conditioning map and only update the network weights in training by back-propagating the cross entropy loss. In testing, the network weights stay fixed while running inference by forward propagating the support set and the query.

Methods(5-shot)	PASCAL-A	PASCAL-B	PASCAL-C	PASCAL-D	Mean
1-NN	34.5	53.0	46.9	25.6	40.0
LogReg	35.9	51.6	44.5	25.6	39.3
OSLSM	35.9	58.1	42.7	39.1	43.9
co-FCN	37.5	50.0	44.1	33.9	41.4
Ours(avg)	40.5	58.9	49.1	39.6	47.0
Ours(union)	43.1	59.6	50.1	40.0	48.2
Ours	**45.3**	**61.6**	**52.1**	**41.7**	**50.2**

Table 1.3: Mean IoU for five-shot semantic segmentation given class partition from Table 1.1

1.6 Experiment

1.6.1 Dataset

We use PASCAL VOC 2012 [8] and follow the protocol from OSLSM [24] for data preparation. We partition a total of 20 classes into 4 folds in alphabetical order, and each fold consists of 5 classes, as shown in Table 1.1. We employ cross-validation and use 1 fold as testing classes, and the rest as training classes. We follow the procedures described in 1.3 to construct D_{train} and D_{test}. We evaluate each fold by averaging the intersection over union of 1000 queries and report the numbers in Table 1.2 for one-shot and Table 1.3 for five-shot. We also include previous works such as co-FCN and OSLSM with the same experiment setup to compare with.

1.6.2 Benchmark Comparison

We use the standard segmentation metric, mean IoU, as computed from $\frac{tp}{tp+fp+fn}$, where tp represents the number of true positive pixels, fp the number of false positive pixels, and fn the number of false negative pixels. The mean cateogry is directly computed by averaging the results of each fold. We include several important one-shot baselines in Table 1.2. In particular, fine-tuning works by only tuning the fully-connected layers, specifically (fc6, fc7, fc8), as mentioned in [3]. Comparing to co-FCN, the second best overall method in one-shot, our method gains an 8% performance increase in PASCAL-B and a 5.3% in mean. Comparing to OSLSM, our

method achieves a 5.6% increase in mean. Since one-shot setting does not utilize our few-shot fusion, we attribute such improvement to the contextual masking module, which produces the class descriptor feature and the conditioning map for segmentation guidance.

From Table 1.2 to Table 1.3, we see a significant boost of IoU in all 4 folds as well as the mean category in five-shot setting. Comparing to the 0.3% improvement in mean from co-FCN, which averages the support features, ours increases from 46.4% to 50.2%, a 3.8% improvement. Surprisingly, in OSLSM, the heuristic of taking union of output masks proves to work well under high precision and low recall, with a 3.1% increase from 40.8%. Instead of averaging or taking the union, we combine our target feature guidance in a principled way. Our few-shot fusion technique utilizes global feature similarity, which provides an even more significant improvement of 3.8% on top of the already better one-shot result.

1.7 Conclusion

We propose FuseNet for few-shot semantic segmentation, where the network learns to segment out regions of target classes with only a few annotated examples. The proposed approach has several advantages over previous work, namely parameter sharing, contextual masking module, and few-shot fusion. We redesign the two-branch network by using a shared fully convolutional network for feature extraction. Such parameter reduction results in less overfitting as well as faster convergence. In contextual masking module, our proposed method encodes the target region and its boundary information into a feature vector. In particular, the linear representation works well under few-shot scenario, where the network has to generalize from the few annotated examples. Lastly, the few-shot fusion weighs each annotated sample according to its global feature similarity with the query. Our network can be optimized end-to-end without any pre-processing or post-processing and achieves the state-of-the-art result on PASCAL VOC 2012 for both one-shot and five-shot semantic segmentation tasks.

Bibliography

[1] Yashas Annadani and Soma Biswas. Preserving semantic relations for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7603–7612, 2018. 1.2

[2] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016. 1.1

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 1.6.2

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 1.2

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 1.2

[6] Ronald Clark, John McCormac, Stefan Leutenegger, and Andrew J Davison. Meta-learning for instance-level data association. In *Neural Information Processing Systems (NIPS)*, 2017. 1.1

[7] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015. 1.1

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 1.2, 1.5, 1.6.1

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 1.2

[10] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 1.2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1.1, 1.2

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1.4.1

[13] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018. 1.1, 1.2

[14] Mostafa S Ibrahim, Arash Vahdat, and William G Macready. Weakly supervised semantic image segmentation with self-correcting networks. *arXiv preprint arXiv:1811.07073*, 2018. 1.1

[15] Longlong Jing, Yucheng Chen, and Yingli Tian. Coarse-to-fine semantic segmentation from image-level labels. *arXiv preprint arXiv:1812.10885*, 2018. 1.1

[16] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 1.1, 1.2

[17] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016. 1.2

[18] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 1.1

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1.1, 1.2, 1.4.1, 1.5

[20] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2554–2563.

JMLR. org, 2017. 1.1

[21] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018. 1.1, 1.2, 1.4, 1.4.2, 1.4.3

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1.1, 1.2, 1.4.1

[23] 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. arxiv: 1409.0575, 2014. 1.5

[24] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 1.1, 1.2, 1.4, 1.4.2, 1.4.3, 1.6.1

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1.4.1, 1.5

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. URL http://arxiv.org/abs/1409.4842. 1.4.1

[27] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1818–1827, 2018. 1.2

[28] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 1.1, 1.2

[29] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, 2018. 1.2

[30] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 597–613, 2018. 1.2

[31] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *arXiv preprint*

arXiv:1810.09091, 2018. 1.2

[32] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 1.2