On the Texture Bias for Few-Shot CNN Segmentation

Reza Azad[1]   Abdur R. Fayjie[1,2]   Claude Kauffmann[2]   Ismail Ben Ayed[1,2]

Marco Pedersoli[1]   Jose Dolz[1]

[1] *ÉTS Montreal, Canada*

[2] *CRCHUM Montreal, Montreal, Canada*

{rezazad68,fayjie92}@gmail.com   claude.kauffmann.chum@ssss.gouv.qc.ca

{ismail.benayed,marco.pedersoli,jose.dolz}@etsmtl.ca

Abstract

Despite the initial belief that Convolutional Neural Networks (CNNs) are driven by shapes to perform visual recognition tasks, recent evidence suggests that texture bias in CNNs provides higher performing models when learning on large labeled training datasets. This contrasts with the perceptual bias in the human visual cortex, which has a stronger preference towards shape components. Perceptual differences may explain why CNNs achieve human-level performance when large labeled datasets are available, but their performance significantly degrades in low-labeled data scenarios, such as few-shot semantic segmentation. To remove the texture bias in the context of few-shot learning, we propose a novel architecture that integrates a set of Difference of Gaussians (DoG) to attenuate high-frequency local components in the feature space. This produces a set of modified feature maps, whose high-frequency components are diminished at different standard deviation values of the Gaussian distribution in the spatial domain. As this results in multiple feature maps for a single image, we employ a bi-directional convolutional long-short-term-memory to efficiently merge the multi scale-space representations. We perform extensive experiments on three well-known few-shot segmentation benchmarks –Pascal i5, COCO-20i and FSS-1000– and demonstrate that our method outperforms state-of-the-art approaches in two datasets under the same conditions.

1. Introduction

Deep models, and particularly convolutional neural networks (CNNs), have shown an impressive performance in many visual recognition tasks, including semantic segmentation [19]. However, their extreme hunger for labeled training data strongly limits their scalability to novel classes and reduces their applicability to rare categories. Few-shot learning [23, 8] has ap-

peared as an appealing alternative to train deep models in a low-labeled data scenario. In this setting, the model is trained to accommodate for novel categories with only a handful of labeled images, typically known as *support* images. In few-shot segmentation approaches, the learned knowledge from the *support* images is typically fed into a parametric module to guide the segmentation of the unseen images, referred to as *queries*.

Recent works have demonstrated that the CNN bias towards recognizing textures rather than shapes introduces several benefits under the standard learning paradigm [9, 2], which contrasts with the inductive bias found in the human visual cortex, that is driven by shapes [13]. This does not represent a problem when training and testing classes are drawn from the same distribution in large-labeled datasets. Nevertheless, in low-labeled data regime, the difference on perceptual biases poses difficulties to CNNs to mimic human performance, particularly if there exists a distributional shift between training and testing classes, such as in the few-shot learning scenario [24].

Thus, we argue that attenuating high-frequency local components in the feature space yields to a better generalization to unseen classes in the context of few-shot semantic segmentation. Our motivation is inspired by the findings in [9], who showed that CNNs have a strong texture inductive bias that limits their ability to leverage useful low-frequency (e.g, shape) information. Although they show that the representational power of CNN can be improved if CNNs are forced to use shape information (by modifying input images), how to design efficient algorithms that allow CNNs to meaningfully use low-frequency information remains an open problem. We tackle this issue by proposing a novel architecture (Fig. 1) which integrates a set of difference of gaussians (DOGs) [20] on the feature representations. At each scale-space of the DOGs, the original high-frequency signals are attenuated differ-

ently, according to the standard deviation values, σ, employed to model the Gaussian distribution in the spatial domain, which results in multiple versions of the feature maps for a single image. The DoG at two near standard deviations (σ_1 and σ_2) will smooth out the features, reducing textural information that the feature extractor may have propagated. Then, following the literature on few-shot segmentation, we generate class representative prototypes from the learned representations, with the difference that in our setting we have multiple prototypes per image, i.e., one at each scale-space of the DOG. Thus, for each query image, our model produces an ensemble of segmentations, each one associated with a prototype. To generate the final prediction, we cast the problem into a sequential segmentation task, where each segmentation on the ensemble represents a time-point. To efficiently fuse temporal, i.e., multiple segmentation masks, and spatial features we resort to a Bi-directional convolutional long-short-term memory (BConvLSTM) [29], which bidirectionally encourages information exchange between LSTM units. Furthermore, in the k-shot setting, our approach learns a parametric fusion of the different support images by jointly analyzing their contribution.

Our contributions can be summarized as follows: (1) we propose to reduce the texture bias in CNNs in the few-shot segmentation task by attenuating high-frequency local components on the feature space, (2) to merge the multiple segmentations produced at different scale-space representations we reformulate the problem as a sequential segmentation task and employ a bi-directional convLSTM to efficiently fuse all the information, and (3) we report very competitive results on few-shot segmentation across several public benchmarks, outperforming most recent literature while keeping a light architecture.

2. Related Work

Few-shot segmentation. Pioneer works on few-shot semantic segmentation [26, 5, 22] incorporated two independent branches: a conditional branch that generates the prototypes (e.g., embedding) from the *support* set, and a segmentation branch, which takes the learned prototypes and the *query* image as input and produces the segmentation masks. More recently, researchers have unified these dual-branch architectures into a single branch network which can derive better guidance features with the addition of a masked average pooling layer [42, 27, 34, 21]. For example, a similarity guidance module is integrated in [42] to recalibrate the query feature map based on a similarity score between the representative prototype and each spatial location on the query features. In [27], authors present

an approach to generate the weights of the final segmentation layer for the novel classes via imprinting. Other works interchange support and query images for prototype alignment regularization [34] or to concurrently make predictions [17] with the goal of achieving better generalization. Nguyen et al. [21] integrated a regularization that estimates feature relevance by encouraging jointly high-feature activations on the foreground and low-feature activations on the background. Deep attention has also been exploited to learn attention weights between support and query images for further label propagation [11, 40]. More recently, some researchers have adopted graph CNNs to establish more robust correspondences between support and query images and enrich the prototype representation [18, 33]. Our work differs from previous approaches from a motivation and methodological perspective. While most of the current literature focuses on learning better prototypical representations or iteratively refining these, we approach this problem under the perspective of reducing the inductive texture bias of CNNs. Thus, from a methodological point of view, our approach is the first attempt to integrate a pyramidal set of DoG to address the problem of texture bias.

Semantic segmentation with conv LSTM. Conv Long-Short Term Memory (LSTM) was presented in [35] to address the limitations of LSTMs in tasks such as semantic segmentation, where the learned intermediate representations of the input images must preserve the spatial information. Particularly, convLSTM addresses this by integrating a convolution operator in the state-to-state and input-to-state transitions. In the context of image segmentation on 3-dimensional data, e.g. videos or medical imaging, convLSTMs are integrated to encode the spatial-temporal relationships between frames or slices [4, 31, 29, 41]. If only 2D images are available instead, an alternative is to leverage convLSTM for multi-level feature fusion [14, 1]. Li et al. [14] employed convLSTM units to progressively refine the segmentation masks from high-level to low-level features. In [1], features derived from the skip connections in the encoding path of UNet [25] were non-linearly fused with their corresponding features in the decoding path by employing a bi-directional convLSTM, instead of a simple concatenation. In a related work, Hu et al. [11] employ a ConvLSTM to merge multiple segmentations in a k-shot scenario ($k > 1$), where each segmentation is generated from a different support image. This differs from our work, where our goal is to fuse the segmentations from a single support image ($k = 1$) derived from multiple scale-space representations. Furthermore, we use a bidirectional ConvLSTM to foster the exchange
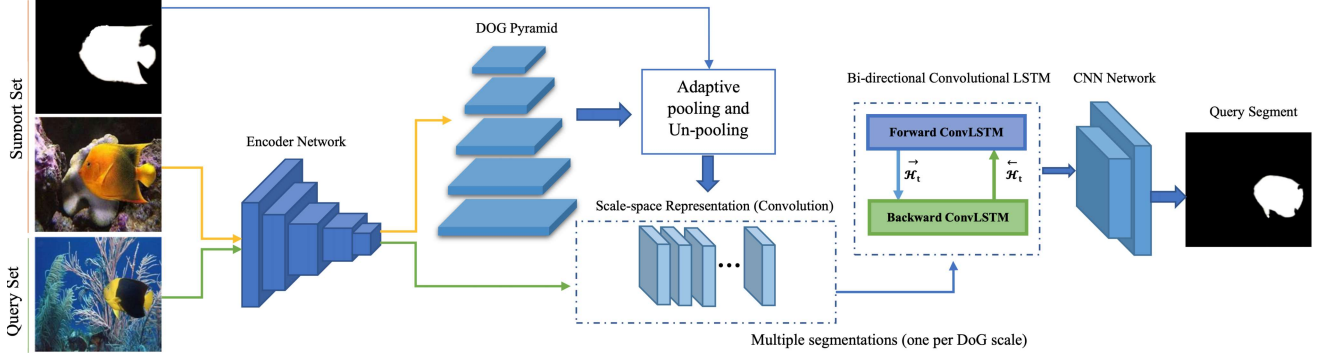
Figure 1. Overview of the proposed method (DoG-LSTM) for few-shot segmentation. It first applies a pyramid of difference of gaussians (DoG) on the learned *support* features to attenuate high-frequency local components on the feature space. To perform segmentation on a *query* image, the multiple scale-space *support* representations are combined with the *query* features, and later fed as input to a bi-directional convLSTM. The convLSTM merges the information from multiple representations and generates the final *query* segmentation map.

of information between the forward and backward path of each recurrent module.

3. Methodology

3.1. Problem Formulation

Following the standard notation and set-up in few-shot semantic segmentation, we define three datasets: a training set $D_{train} = \{(X_i^t, Y_i^t)\}_{i=1}^{N_{train}}$, a support set $D_{support} = \{(X_i^s, Y_i^s)\}_{i=1}^{N_{support}}$, and a test set $D_{test} = \{(X_i^q)\}_{i=1}^{N_{test}}$. In this setting, $X_i \in \mathbb{R}^{H \times W \times 3}$ denotes an RGB image, with H and W being the height and the width of the image, respectively, and $Y_i \in \{0,1\}^{H \times W}$ is its corresponding pixel-level mask. Furthermore, each set contains N images. The classes, denoted as $c \in C$, are shared among the support and test set, and are disjoint with the training set, i.e., $\{C_{train}\} \cap \{C_{support}\} = \emptyset$.

The purpose of few-shot learning is to train a neural network $f_\theta(\cdot)$ on the training set D_{train} to have the ability to segment a novel class $c \notin C_{train}$ on the test set D_{test}, based on k references from the support set $D_{support}$. To reproduce this mechanism during training, the network is trained on D_{train} following the episodic paradigm [32]. Specifically, assuming a *c*-way *k*-shot learning task, each episode is generated by sampling: (1) a support training set $D_{train}^{\mathcal{S}} = \{(X_s^t, Y_s^t(c))\}_{s=1}^k \subset D_{train}$ for each class c, where $Y_s^t(c)$ is the binary mask for the class c corresponding to the image X_s^t and (2) a query set $D_{train}^{\mathcal{Q}} = \{X_q^t, Y_q^t(c)\} \subset D_{train}$, where X_q^t is the query image and $Y_q^t(c)$ its corresponding binary mask for the class c. The input of the model is composed of the support training set and the query image, $f_\theta(D_{train}^{\mathcal{S}}, X_q^t)$, which are employed to estimate the segmentation mask for the class c in the query image, $\hat{Y}_q^t(c)$. Then, the neural network parameters θ are optimized

by employing an objective function between $Y_q^t(c)$ and $\hat{Y}_q^t(c)$[1]. During the testing phase, the model $f_\theta(\cdot)$ is evaluated on the test set D_{test} given k images from the support set $D_{support}$.

3.2. Removing Texture Bias

Recent findings suggest that perceptual bias on CNNs do not correlate with those in the human visual cortex [9], which may limit the performance of these models in low-labeled data scenarios [24]. Inspired by this, we propose to reduce the texture bias of CNNs in the context of few-shot segmentation. To achieve this, we integrate a set of difference of gaussians (DoGs) [20] into the learned feature space to attenuate high-frequency local components, i.e., texture. First, we use a CNN to encode the input images into the latent space, resulting in $F_s \in \mathbb{R}^{W' \times H' \times M}$ and $F_q \in \mathbb{R}^{W' \times H' \times M}$ for the support and query samples. The variables W', H' and M represent the width, height and feature dimensionality on the latent space, respectively. To encode the high-frequency information during training, we apply a DoG on each channel $m \in M$ of the feature map from the support samples F_s, which can be formulated as:

$$
\begin{aligned}
G_s &= \Gamma_{\sigma_1, \sigma_2}(F_s) \\
&= (F_s^m * \frac{1}{2\pi\sigma_2^2} \exp^{-\frac{x^2+y^2}{2\sigma_2}}) - \\
&\quad (F_s^m * \frac{1}{2\pi\sigma_1^2} \exp^{-\frac{x^2+y^2}{2\sigma_1}}), \forall m \in M \quad (1)
\end{aligned}
$$

where σ_1 and σ_2 are $(\sigma_2 > \sigma_1)$ are the variance of the Gaussian filters, x and y represent the spatial position in the encoded feature space and $*$ denotes the convolution operator. To encode different frequency information

---

[1]Typically the standard cross-entropy loss function is employed in the few-shot segmentation literature.

we apply a pyramid of DoGs with increasing σ values, similar to [20]. This results in L level representations ($L = 4$) for each support sample (See Fig. 2), where the novel feature maps at each level ($l \in L$) can be denoted as $G_s^l \in \mathbb{R}^{W' \times H' \times M}$.

Support images can contain cluttered background, as well as multiple object categories. Thus, we need to find a representative embedding f_s that corresponds exclusively to the target class. Since we have L feature representations, each of them encoding different high-frequency local components, we generate L prototypes per class. To obtain the class prototypes, the novel encoded feature maps at each scale G_s^l are averaged over the known foreground regions in the support mask $Y_s(c)$. Thus, at each level we can estimate f_s^l as:

$$f_s^l = \frac{1}{|\tilde{Y}_s(c)|} \sum_{i=1}^{W'H'} G_s^l \tilde{Y}_s(c) \tag{2}$$

where the support mask $Y_s(c)$ is down-sampled to $\tilde{Y}_s(c) \in \{0,1\}^{H' \times W'}$ to match the spatial resolution of the feature maps G_s^l and $|\tilde{Y}_s(c)| = \sum_i \tilde{Y}_{s,i}(c)$ is the number of foreground locations in $\tilde{Y}_s(c)$. Then, each prototype is unpooled to the same spatial resolution as the query features F_q and the upsampled prototypes are convolved with F_q. We then define the scale-space representation (SSR), which will serve as input signal of the BConvLSTM. This representation can be formulated as a convolution operation between the class representative feature maps at each scale-space and the feature maps derived from the query image:

$$SSR = \{BN(\psi_s^l * F_q)\}, \forall l \in L \tag{3}$$

where ψ_s^l are the upsampled prototypes f_s^l, and BN denotes a batch normalization layer.

3.3. Encoding Scale-Space Representation

Fusion of the query features F_q with the multi-scale class representations from the support features ψ_s^l produces L joint feature maps, one at each scale-space representation. While logical or average operations may be a straightforward solution to obtain a unique representation, they fail to exploit the inner relationship between sequential scale-space representations. To efficiently solve this, we reformulate the problem as a sequential task, and integrate a bidirectional convolutional long short term memory (BConvLSTM) [29] on the output of the CNN architecture (Fig. 2). Even though LSTM have been proposed to deal with sequential problems, this sequential processing strategy may fail to explicitly encode the spatial correlation, since they use full connections in input-to-state and state-to-state transitions. To overcome this limitation, ConvLSTM was

proposed in [35], which leverages convolution operations into input-to-state and state-to-state transitions instead. Specifically, three gating functions are calculated in the ConvLSTM, which are defined as:

$$i_t = \sigma(\mathbf{W}_{xi} * \mathcal{X}_t + \mathbf{W}_{hi} * \mathbf{H}_{t-1} + \mathbf{W}_{ci} \circ \mathbf{C}_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(\mathbf{W}_{xf} * \mathcal{X}_t + \mathbf{W}_{hf} * \mathbf{H}_{t-1} + \mathbf{W}_{cf} \circ \mathbf{C}_{t-1} + b_f) \tag{5}$$

$$o_t = \sigma(\mathbf{W}_{xo} * \mathcal{X}_t + \mathbf{W}_{ho} * \mathbf{H}_{t-1} + \mathbf{W}_{co} \circ \mathbf{C}_{t-1} + b_0) \tag{6}$$

where $\mathcal{X}_t$ and $\mathbf{H}_t$ denote the input (i.e., SSR in eq. (3)) and hidden state at time t, respectively, and b is used to represent the bias term in each state. Similarly, $\mathbf{W}_x$, $\mathbf{W}_h$ and $\mathbf{W}_c$ represent the set of learnable parameters. Last, '$\circ$' denotes the Hadamard product. The LSTM module generates a new proposal for the cell state by looking at the previous $\mathbf{H}$ and current $\mathcal{X}$, resulting in:

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_{xc} * \mathcal{X}_t + \mathbf{W}_{hc} * \mathbf{H}_{t-1} + b_c) \tag{7}$$

Now we linearly combine the newly generated proposal $\tilde{\mathbf{C}}_t$ with the previous state $\mathbf{C}_{t-1}$ to generate the final cell state in the recurrent model:

$$\mathbf{C}_t = f_t \circ \mathbf{C}_{t-1} + i_t \circ \tilde{\mathbf{C}}_t \tag{8}$$

Finally, the new hidden state $\mathbf{H}$ can be estimated as:

$$\mathbf{H}_t = o_t \circ \tanh(\mathbf{C}_t) \tag{9}$$

Inspired by [29], we employ in this work a BConvLSTM to encode the different scale-space representations (SSR) at the output of the convolutional network (Fig. 2). The bidirectional modules with forward and backward paths allow to strength the spatio-temporal information exchanges between the two sides, facilitating the memorization of both past and future sequences. This contrasts with the standard convLSTM, where only the dependencies on the forward direction are employed for the predictions. Thus, the output prediction for a query image X^q is given at the output of the BConvLSTM, which is defined as:

$$\hat{Y}^q = \tanh(\mathbf{W}_y^{\overrightarrow{\mathbf{H}}} * \overrightarrow{\mathbf{H}} + \mathbf{W}_y^{\overleftarrow{\mathbf{H}}} * \overleftarrow{\mathbf{H}} + b) \tag{10}$$

where $\overrightarrow{\mathbf{H}}$ and $\overleftarrow{\mathbf{H}}$ represent the hidden states of the forward and backward convLSTM units, respectively, and b is the bias term. Last, the output of the BConvLSTM is passed through a series of convolutions, followed by upsampling and batch normalization layers to produce the final segmentation masks in the original input image resolution.
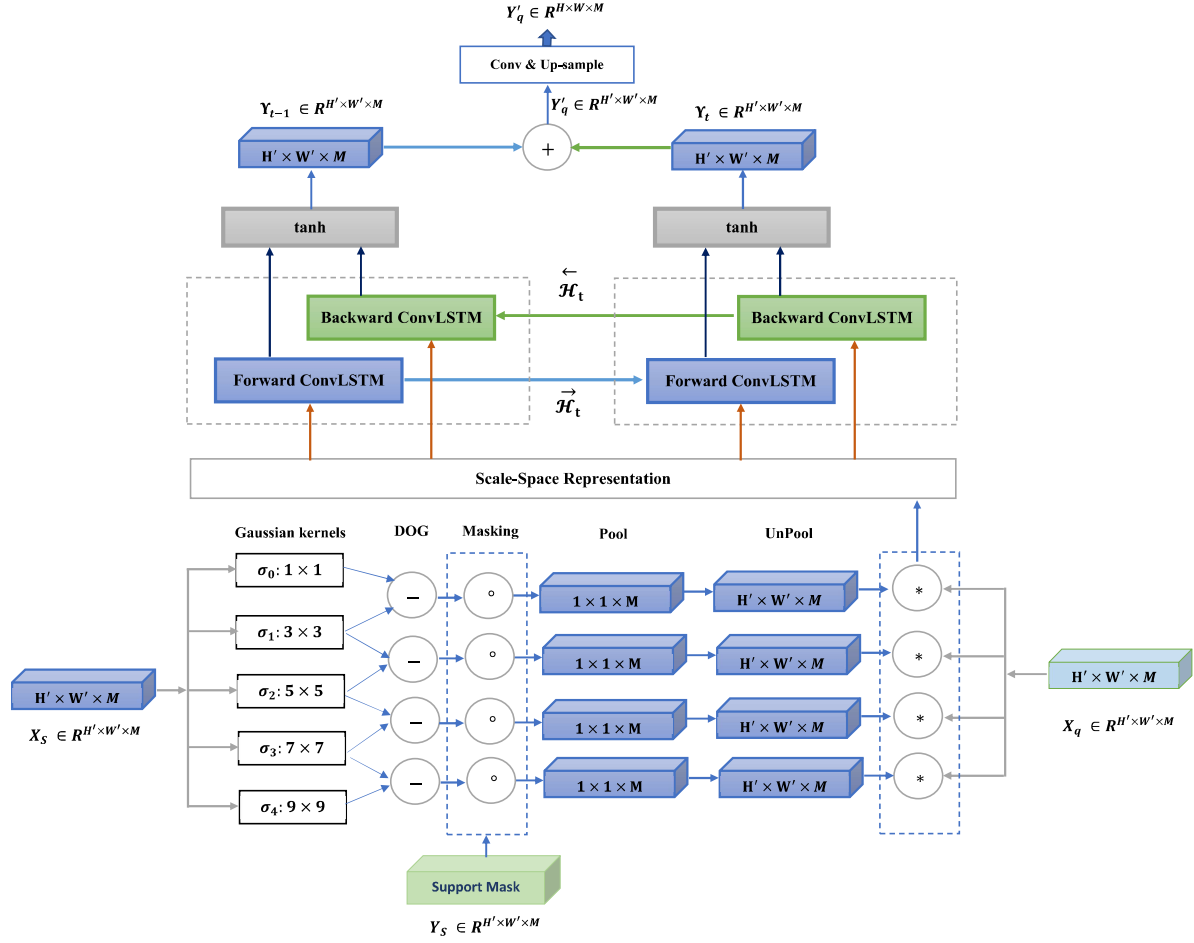
Figure 2. The scale-space encoding block in the proposed method.

3.4. k-shot Segmentation

To fuse the information from several support images ($k > 1$), most previous works estimate the class prototype ψ by simply taking the average of the representation vectors among k samples (non-parametric approach) [26, 42, 27]. Nevertheless, this strategy assumes that each k sample has equal importance, and thus fails to provide a robust category representation when dealing with noisy or corrupted samples. To deal with this limitation, we propose to use a non-linear parametric method to further improve the model performance on the k-shot setting. The key idea is to first generate the embedded representation between the query and each k support samples and then apply BConvLSTM on these representations to get the final representation in a non-linear parametric fashion. Moving k-shot setting inside the scale-space representation gives the BConvLSTM more freedom to generate better representations using various samples.

3.5. Weakly-supervised Few-shot Segmentation

To push further the idea of training with very few supervision, we explore the performance of our method when other forms different than full-supervision , i.e., full pixel-level masks, are available. Particularly, we investigate bounding box annotations, which are less time-consuming to obtain than exhaustive segmentation masks. In this context, we relax the support mask by considering all the area inside the bounding box as the foreground. We show in the experiments that, compared to pixel-level annotations, our model achieves very competitive results by employing sparse support annotations.

4. Experiments

In this section, we present the datasets employed to evaluate our method and the experimental setting in our experiments. We then report the results compared to state-of-the-art few-shot segmentation approaches, demonstrating the benefits of our method.

4.1. Datasets

We perform extensive evaluations on three few-shot semantic segmentation benchmarks, i.e., PASCAL-5^i, FSS-1000 and COCO, following standard procedures in the literature. Details are given in *Supplemental materials*.

4.2. Experimental Set-up

4.2.1 Network and implementation details.

We employ VGG [28] and ResNet-101 pre-trained on ImageNet as the backbones for feature extractor. The proposed model is trained end-to-end by using Adam [12] for 50K episodes with a batch size of 5. The initial learning rate is set to 10^{-4} and reduced by 10^{-1} at every 10K iterations. The work is carried out using one NVidia Titan X GPU. The code is written in Keras with tensorflow as backend and the code is publicly available at `https://github.com/rezazad68/fewshot-segmentation`

4.2.2 Evaluation protocol.

To evaluate the performance of the few-shot segmentation models, we employ the average IoU over all classes (mIoU). As pointed out in [40], the mIoU is a better metric, compared to background-foreground IoU (FB-IoU), in the context of few-shot semantic segmentation for several reasons. First, if a given image contains very small objects, the model may completely fail to segment those objects. Nevertheless, the background IoU can still be very high, which misleads information about the real performance of the model. And second, FB-IoU is more suitable for binary segmentation problems, such as video or foreground segmentation, while our purpose is on semantic segmentation.

4.3. Results

4.3.1 Comparison with state-of-the-art.

Comparison of the proposed model with state-of-the-art methods on the FSS-1000 and PASCAL-5^i datasets is reported in Tables 1 and 2, respectively[2]. Results in Table 1 show that the proposed model outperforms the state-of-the-art methods in both 1-shot and 5-shot settings employing the same backbone, i.e., VGG. Particularly, in the 1-shot task, our method achieves a significant improvement of 5.5% over the second best performing model. In the case of 5-shot learning, we found that fusing the segmentations from the different supports in a non-parametric way brings nearly 1% of improvement with respect to the 1-shot setting. Nevertheless, combining the 5 support segmentations in a parametric

---

[2]Results on COCO are given in Supplemental Material.

---

fashion, i.e., with BConvLSTM, increases the mIoU by 2.5%. It is noteworthy to mention that the method in DAN [33] uses ResNet-101 as backbone, which might explain the differences between the different methods.

Table 1. Results of 1-way 1-shot and 1-way 5-shot segmentation on the FSS-1000 data set employing the mean Intersection Over Union (mIoU) metric. Best results in bold.

Method	mIoU
	1-shot
OSLSM [26]	70.3
co-FCN [22]	71.9
FSS-1000 [16]	73.5
FOMAML [10]	75.2
Baseline	74.2
Baseline+DoG	78.7
Baseline+DoG+BConvLSTM	**80.8**
DAN [33] (ResNet-101)	85.2
	5-shot
OSLSM [26]	73.0
co-FCN [22]	74.3
FSS-1000 [16]	80.1
FOMAML+regularization [10]	80.6
FOMAML+regularization+UHO [10]	82.2
Baseline+DoG+BConvLSTM (non-param)	81.7
Baseline+DoG+BConvLSTM (param)	**83.4**
DAN [33] (ResNet-101)	88.1

We now report in Table 2 an extensive evaluation of all previous works on PASCAL-5^i, the most common benchmark in few-shot semantic segmentation. To make a fair comparison under different feature extractor backbones, we split the table into three groups. The *top* group shows the approaches that rely on VGG-16 as backbone architecture, whereas the methods in the *middle* and *bottom* groups resort to ResNet-50 and ResNet-101 to extract features, respectively. From the reported values, we can observe that the proposed approach outperforms most previous methods, under the same backbone and in both 1- and 5-shot scenarios. Specifically, compared to the second best performing approach based on VGG-16 (i.e., [17]), our method achieves nearly 3% and 2% of improvement in 1- and 5-shot, respectively. Furthermore, our approach achieves the best and second best performance across all the methods in the 1- and 5-shot scenarios, respectively, regardless of the backbone architecture. These quantitative results demonstrate the strong learning and generalization capabilities of the proposed model in both 1- and 5-shot settings.

4.3.2 Qualitative results.

We depict visual results of the proposed method on Pascal5^i in Fig 3. Particularly, the support image-mask pair and the segmentation generated by our method for multiple query images, as well as their corresponding ground truths for several categories are shown. Without

Table 2. Results of 1-way 1-shot and 1-way 5-shot segmentation on PASCAL-5^i data set employing the mean Intersection-Over-Union (mIoU) metric. Best results for each backbone architecture are highlighted in bold. We employ ∇ to denote the difference between 1- and 5-shot settings.

Method		1-shot fold^1	fold^2	fold^3	fold^4	Mean	5-shot fold^1	fold^2	fold^3	fold^4	Mean	∇
					Backbone (VGG 16)							
OSLSM [26]	BMVC'18	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	43.9	3.1
co-FCN [22]	ICLRW'18	36.7	50.6	44.9	32.4	41.1	37.5	50.0	44.1	33.9	41.4	0.3
AMP [27]	ICCV'19	41.9	50.2	46.7	34.7	43.4	41.8	55.5	50.3	39.9	46.9	3.5
PANet [34]	ICCV'19	42.3	58.0	51.1	41.2	48.1	51.8	64.6	**59.8**	46.5	55.7	7.6
FWB[21]	ICCV'19	47.0	59.6	52.6	48.3	51.9	50.9	62.9	56.5	50.1	55.1	3.2
Meta-Seg [3]	IEEE Access'19	42.2	59.6	48.1	44.4	48.6	43.1	62.5	49.9	45.3	50.2	1.6
MDL [6]	COMPSAC'19	39.7	58.3	46.7	36.3	45.3	40.6	58.5	47.7	36.6	45.9	0.6
SG-One [42]	IEEE SMC'20	40.2	58.4	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1	0.8
OS$_{Adv}$ [37]	Inf. Sci.'20	46.9	59.2	49.3	43.4	49.7	47.2	58.8	48.8	47.4	50.6	0.9
ARNet [15]	ICASSP'20	42.6	59.5	50.2	40.2	48.1	43.3	59.8	51.7	41.4	49.1	1.0
CRNet [17]	CVPR'20	-	-	-	-	55.2	-	-	-	-	58.5	3.3
FSS-1000 [16]	CVPR'20	-	-	-	-	-	37.4	60.9	46.6	42.2	56.8	–
RPMM [18]	ECCV'20	47.1	65.8	50.6	48.5	53.0	50.0	66.5	51.9	47.6	54.0	1.0
PFNet [30]	TPAMI'20	**56.9**	**68.2**	54.4	52.4	**58.0**	**59.0**	69.1	54.8	52.9	59.0	1.0
Proposed	-	56.2	66.0	**56.1**	**53.8**	**58.0**	57.5	**70.6**	56.6	**57.7**	**60.6**	2.6
					Backbone (ResNet-50)							
CANet [40]	CVPR'19	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1	1.7
PGNet [39]	ICCV'19	56.0	66.9	50.6	50.4	56.0	57.7	68.7	52.9	54.6	58.5	2.5
LTM [38]	MMM'20	52.8	**69.6**	53.2	52.3	57.0	57.9	**69.9**	56.9	**57.5**	60.6	3.6
CRNet [17]	CVPR'20	-	-	-	-	55.7	-	-	-	-	58.8	2.9
PPNet [18]*	ECCV'20	47.8	58.8	53.8	45.6	51.5	**58.4**	67.8	**64.9**	56.7	**62.0**	10.5
RPMM [36]	ECCV'20	55.2	66.9	52.6	50.7	56.3	56.3	67.3	54.5	51.0	57.3	1.0
PFNet [30]	TPAMI'20	**61.7**	69.5	**55.4**	**56.3**	**60.8**	63.1	70.7	55.8	57.9	61.9	1.1
					Backbone (ResNet-101)							
FWB [21]	ICCV'19	51.3	64.5	56.7	52.2	56.2	54.9	67.4	**62.2**	55.3	59.9	3.7
DAN [33]	ECCV'20	54.7	68.6	**57.8**	51.6	58.2	57.9	69.0	60.1	54.9	60.5	2.3
PFNet [30]	TPAMI'20	**60.5**	**69.4**	54.4	**55.9**	**60.1**	**62.8**	**70.4**	54.9	**57.6**	**61.5**	1.4
Proposed	-	57.0	67.2	56.1	54.3	58.7	57.3	68.5	61.5	56.3	60.9	2.2

* We report the results where no additional unlabeled data is employed.

any post-processing step, the proposed model provides satisfying segmentation results on unseen classes with only one annotated support image. It is noteworthy to highlight that the same support image can be employed to segment multiple query images presenting high appearance variability. For example, our model can successfully segment cats (first row of Fig. 3) when only fractions of the target are shown, such as the head (first column) or even a partial head (third column). Looking at other categories, e.g., bike or table, we observe that the proposed method can also handle objects viewed from a different perspective or presenting different shapes. This illustrates that our model has a strong ability to successfully generalize to unseen classes from only a handful of labeled examples.

4.3.3 Impact of the multiple scale-space representation fusion strategy

Logical operations, such as OR, have been typically employed to fuse features from different support images

in k-shot segmentation. Even though these operations are straightforward, their result is hard to interpret and they fail to efficiently model the relation between sequential data. Thus, in addition to the results in Table 1, we show in Table 3 the impact of fusing the multiple scale-space representations by both the simple average operation or an additional convolutional layer. Particularly, employing a convolutional layer to combine multiple scale-space representations brings nearly 1% of improvement compared to the simple average. On the other hand, if we integrate the proposed strategy, the performance is improved by 3% and 2%, respectively. These results demonstrate that our fusion achieves better few-shot segmentation performance.

4.3.4 Weakly supervised performance.

We further evaluate the proposed model with weaker forms of annotations, e.g., bounding boxes. As reported in Table 4, our method achieves comparable perfor-
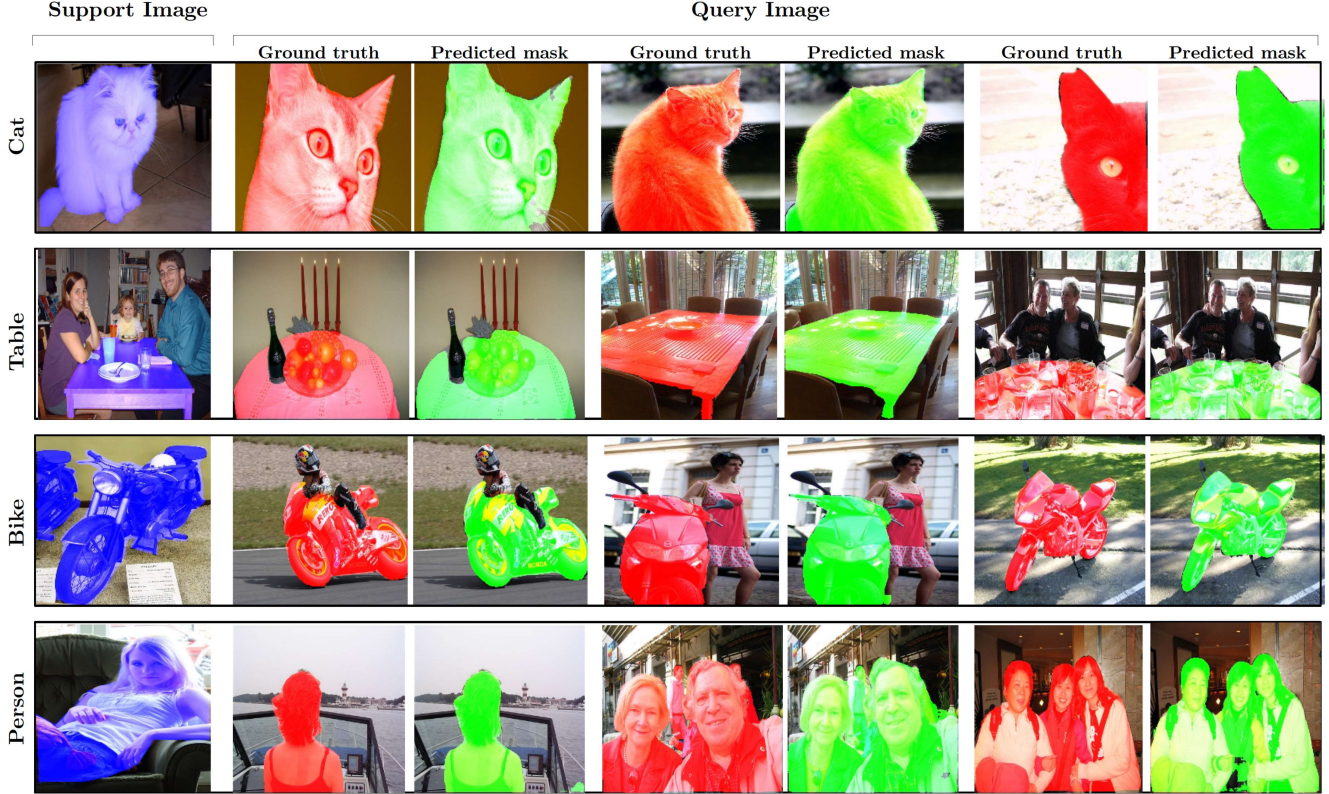
Figure 3. Visual results on Pascal-5^i in 1-way 1-shot setting using the proposed method. The support set, as well as predictions on several query images with corresponding ground truths are shown.

Table 3. Results of 1-way 1-shot segmentation on the FSS-1000 dataset with different fusion strategies to combine multiple scale-space representations. Best results in bold.

Method	mIoU
	1-shot
Average	77.6
CNN layer	78.7
Proposed (BConvLSTM)	**80.8**

Table 4. Full supervision vs weak-supervision performance in the 1-shot scenario. Type of supervision in brackets.

Method	mIoU	
	FSS-1000	PASCAL
Proposed (Pixels)	80.8	58.0
Proposed (Bounding boxes)	78.2	56.4
PANet [34] (Bounding boxes)	–	45.1

mance to full supervision when bounding boxes are available in the support set of novel categories. Furthermore, compared to the very recent PANet architecture [34] our model brings 10% of performance gain in the context of weak supervision. This suggests that our model is able to deal efficiently with noise introduced by bounding box annotations, which ultimately results in more representative class prototypes that approach those obtained by pixel-level annotations.

5. Conclusions

We have presented a novel segmentation network that tackles the challenging problem of few-shot learning from the perspective of reducing the inductive texture bias on CNNs. This contrasts with most prior literature, which focuses on explicitly enhancing the prototypes representation. Particularly, the proposed model presents two novel contributions. First, we integrated a pyramid of Difference of Gaussians to attenuate high-frequency local components in the feature space. Second, to merge information at multiple scale-space representations we reformulated the problem as a sequential task and resorted to bi-directional convolutional LSTMs. For evaluation purposes, we have compared the proposed method to prior work, and performed ablations on important elements of our model on public few-shot segmentation benchmarks. Results demonstrated that the proposed model outperforms most prior methods while maintaining a light architecture, achieving a new state-of-the-art performance on several few-shot semantic segmentation settings.

References

[1] Reza Azad, Maryam Asadi-Aghbolaghi, Mahmood Fathy, and Sergio Escalera. Bi-directional convlstm u-net with densley connected convolutions. In *ICCV Workshops*, pages 0–0, 2019.

[2] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *ICLR*, 2019.

[3] Zhiying Cao, Tengfei Zhang, Wenhui Diao, Yue Zhang, Xiaode Lyu, Kun Fu, and Xian Sun. Metaseg: A generalized meta-learning framework for multi-class few-shot semantic segmentation. *IEEE Access*, 7:166109–166121, 2019.

[4] Jianxu Chen, Lin Yang, Yizhe Zhang, Mark Alber, and Danny Z Chen. Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation. In *Advances in neural information processing systems*, pages 3036–3044, 2016.

[5] Nanqing Dong and Eric Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.

[6] Zihao Dong, Ruixun Zhang, Xiuli Shao, and Hongyu Zhou. Multi-scale discriminative location-aware network for few-shot semantic segmentation. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 42–47. IEEE, 2019.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[8] 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.

[9] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

[10] Sean M Hendryx, Andrew B Leach, Paul D Hein, and Clayton T Morrison. Meta-learning initializations for image segmentation. *arXiv preprint arXiv:1912.06290*, 2019.

[11] Tao Hu, Pengwan Yang, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees GM Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8441–8448, 2019.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

[14] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, pages pp. 5745–5753, 2018.

[15] Rusheng Li, Hanhui Liu, Yuesheng Zhu, and Zhiqiang Bai. Arnet: Attention-based refinement network for few-shot semantic segmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2238–2242. IEEE, 2020.

[16] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. FSS-1000: A 1000-class dataset for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2869–2878, 2020.

[17] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. CRNet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4173, 2020.

[18] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, 2020.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages pp. 3431–3440, 2015.

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[21] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *ICCV*, pages pp. 622–631, 2019.

[22] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *ICLR Workshop*, 2018.

[23] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[24] Sam Ringer, Will Williams, Tom Ash, Remi Francis, and David MacLeod. Texture bias of CNNs limits few-shot classification performance. *arXiv preprint arXiv:1910.08519*, 2019.

[25] 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, 2015.

[26] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2018.

[27] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. AMP: Adaptive masked proxies for few-shot segmentation. In *ICCV*, pages pp. 5249–5258, 2019.

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

[29] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper ConvLSTM for video salient object detection. In *ECCV*, pages pp. 715–731, 2018.

[30] Z Tian, H Zhao, M Shu, Z Yang, R Li, and J Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[31] Sepehr Valipour, Mennatullah Siam, Martin Jagersand, and Nilanjan Ray. Recurrent fully convolutional networks for video segmentation. In *WACV*, pages pp. 29–36, 2017.

[32] 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.

[33] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *ECCV*, 2020.

[34] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. PANet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, pages pp. 9197–9206, 2019.

[35] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[36] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, 2020.

[37] Guangchao Yang, Dongmei Niu, Caiming Zhang, and Xiuyang Zhao. Recognizing novel patterns via adversarial learning for one-shot semantic segmentation. *Information Sciences*, 2020.

[38] Yuwei Yang, Fanman Meng, Hongliang Li, Qingbo Wu, Xiaolong Xu, and Shuai Chen. A new local transformation module for few-shot segmentation. In *International Conference on Multimedia Modeling*, pages 76–87. Springer, 2020.

[39] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *ICCV*, pages pp. 9587–9595, 2019.

[40] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, pages pp. 5217–5226, 2019.

[41] Dongqing Zhang, Ilknur Icke, Belma Dogdas, Sarayu Parimal, Smita Sampath, Joseph Forbes, Ansuman Bagchi, Chih-Liang Chin, and Antong Chen. A multi-level convolutional LSTM model for the segmentation of left ventricle myocardium in infarcted porcine cine mr images. In *IEEE International Symposium on Biomedical Imaging (ISBI 2018)*, pages pp. 470–473, 2018.

[42] 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.

[43] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Objectness-aware one-shot semantic segmentation. *arXiv preprint arXiv:2004.02945*, 2020.