FSS-1000: A 1000-Class Dataset for Few-Shot Segmentation

Tianhan Wei[1]   Xiang Li[1*] Yau Pun Chen[1*]   Yu-Wing Tai[2]   Chi-Keung Tang[1]
[1]HKUST   [2]Tencent

{tweiab, xlide, ypchen}@connect.ust.hk, yuwingtai@tencent.com, cktang@cs.ust.hk

Abstract

Over the past few years, we have witnessed the success of deep learning in image recognition thanks to the availability of large-scale human-annotated datasets such as PAS-CAL VOC, ImageNet, and COCO. Although these datasets have covered a wide range of object categories, there are still a significant number of objects that are not included. Can we perform the same task without a lot of human annotations? In this paper, we are interested in few-shot object segmentation where the number of annotated training examples are limited to 5 only. To evaluate and validate the performance of our approach, we have built a few-shot segmentation dataset, FSS-1000, which consists of 1000 object classes with pixelwise annotation of ground-truth segmentation. Unique in FSS-1000, our dataset contains significant number of objects that have never been seen or annotated in previous datasets, such as tiny daily objects, merchandise, cartoon characters, logos, etc.

We build our baseline model using standard backbone networks such as VGG-16, ResNet-101, and Inception. To our surprise, we found that training our model from scratch using FSS-1000 achieves comparable and even better results than training with weights pre-trained by ImageNet which is more than 100 times larger than FSS-1000. Both our approach and dataset are simple, effective, and easily extensible to learn segmentation of new object classes given very few annotated training examples. Dataset is available at https://github.com/HKUSTCV/FSS-1000

1. Introduction

Although unprecedented in the number of object categories when first released, contemporary image datasets for training deep neural networks such as PASCAL VOC [5] (19,740 images, 20 classes), ILSVRC [28] (1,281,167 images, 1,000 classes), and COCO [21] (204,721 images, 80 classes) are actually quite limited for visual recognition tasks in the real world: a rough estimate of the number of different objects on the Earth falls in the range of 500,000 to 700,000, following the total number of nouns in the En-

glish language. While the exact total number of visual object categories is smaller than these numbers, these large-scale datasets contribute less than 1% in total. Extending a new object category to existing datasets is a major undertaking because a lot of human annotation effort is required: in ImageNet, the mean number of images in a given class is 650. More importantly, observe that the number of images within each object category in ImageNet for instance can vary significantly, ranging from 1 to 3,047. This inevitably introduces undesirable biases which may have a detrimental effect on important tasks solely relying on pre-trained weights obtained using a dataset that is biased in both the choice of object classes (small number) and images within a given class (uneven distribution). Biases in existing datasets have also been recently reported [9, 20].

Thus, Few-Shot Learning has emerged as an attractive alternative for important computer vision tasks, especially when the given new dataset is very small and dissimilar so relying on the aforementioned pre-trained weights may not work well. Particularly relevant is image segmentation which requires extremely labor-intensive, pixelwise labeling for supervised learning. In few-shot segmentation, given an input consisting of a small support image set with labels (5 in this paper) and a query image set without labels, the learned model should properly segment the query images, even the pertinent objects belong to an object class unseen before.

There is *no* large-scale object dataset for few-shot segmentation. Previous research on few-shot segmentation relies on a manual split of the PASCAL VOC dataset to train and evaluate a new model [30, 24], but only 20 and 80 classes in the PASCAL VOC and COCO datasets respectively contain pixelwise segmentation information. Thus, building a large-scale object segmentation dataset is necessary to extensively and objectively evaluate the performance of our and future few-shot models.

FSS-1000 is the first large-scale dataset for few-shot segmentation with built-in object category hierarchy which emphasizes the number of object *classes* rather than the number of images. FSS-1000 is highly scalable: 10 new images with ground-truth segmentation are all it takes for new object class extension.

Our baseline network architecture is constructed by ap-

---

*Equal contribution.

Dataset	Images	Classes	Classification	Detection	Segmentation	Mean	Stddev
SUN [36]	131,067	3,819	✓	✓	✗	39.22	717.68
ImageNet	3,200,000	5,247	✓	✓	✗	650.02	526.03
Open Image	9,052,839	7,186	✓	✓	✗	1409.62	14429.29
PASCAL VOC 2012	19,740	20	✓	✓	✓	215.90	164.07
MS COCO	204,721	80	✓	✓	✓	4492.13	7487.38
FSS-1000	10,000	1,000	✓	✓	✓	10	**0**

Table 1. Large-scale datasets comparison. Mean and standard deviation are based on the expected number of images in each class.
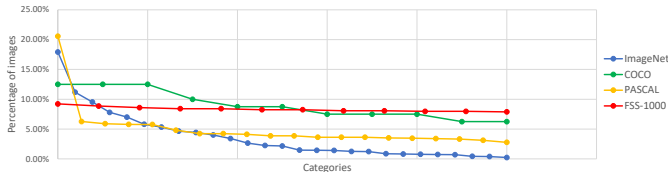

Figure 1. Normalized image distribution. To make these datasets comparable, we normalize each dataset respectively in the total number of images (y-axis) and in the total number of object super-categories (x-axis) such that the area under each curve is 1 to make them comparable. All existing datasets are biased toward a number of object categories except FSS-1000 (red).

pending a decoder module to the relation network [33], which is a simple and elegant deep model effective and originally designed for few-shot image classification only. Reshaping the relation network into a fully-convolutional U-Net architecture [26], our extensive experimental results show that this baseline model trained from scratch on FSS-1000, which is less than 1% of the size of contemporary large-scale datasets, outperforms the model fine-tuned from weights pre-trained on ImageNet/COCO dataset. With its excellent segmentation performance as well as extensibility, FSS-1000 is expected to make a lasting contribution to few-shot image segmentation. Please also refer to the supplemental materials for our extensive experimental results.

2. Related Work

We first review the relationship and difference between FSS-1000 and modern datasets aiming to solve image segmentation and few-shot classification. Then we review contemporary research on few-shot learning and semantic segmentation and discuss how we relate the few-shot segmentation to previous research.

Large-Scale Datasets When deep learning had started to become a dominating tool for computer vision, the importance of building large-scale datasets was emphasized for training deep networks. The PASCAL VOC [5] was the first to provide a challenging image dataset for object class recognition and semantic segmentation. The latest version VOC2012 contains 20 object classes and 9,993 images with segmentation annotations. Despite the absence of segmentation labels, the Imagenet [4] is built upon the backbone of WordNet and provides image-level labels for 5,247 classes for training, out of which a subset of 1,000 categories are split out to form the ILSVRC [28] dataset. This challenge

has made a significant impact on the rapid progress in visual recognition task and computer vision in recent years. The latest Open Image dataset [17] contains 7,186 trainable distinct object classes for classification and 600 classes for detection, making it the largest existing dataset with object classes and location annotations. Following the PASCAL VOC and ImageNet, the COCO segmentation dataset [21] includes more than 200,000 images with instance-wise semantic segmentation labels. There are 80 object classes and over 1.5 million object instances in COCO dataset.

In this paper, we instead focus on broadening the number of object classes in a segmentation dataset rather than increasing dataset size. Our FSS-1000 consists of 1,000 object classes, wherein each class we label 10 images with binary segmentation annotation. So in total, our dataset contains 10,000 images with pixelwise segmentation labels. We are particularly interested in segmentation due to its obvious benefits: segmentation captures the essential feature of an object without background; instance level segmentation can be ready from segmentation. The structure of our dataset is similar to widely-used datasets for few-shot visual recognition. For example, the Omniglot dataset [18] consists of 1,623 different handwritten characters of 50 different alphabets, which is equivalent to 1,623 object classes with 50 images in each class. The MiniImageNet, first proposed in [35], consists of 60,000 images with 100 classes each having 600 examples. But none of these few-shot learning datasets incorporate dense pixelwise segmentation labels, which is essential in training a deep network model for semantic segmentation.

Few-Shot Learning Recent research in few-shot classification can be classified into 1) learn a good initial condition for the network to be fine-tuned on extremely small training set, as proposed in [8, 25]; 2) rely on memory properties of RNN, introduced in [22, 29]; 3) learn a metric between few-shot samples and queries, as in [2, 10, 18, 16, 33]. We choose to extend the relation network [33] for few-shot segmentation because it is a simple, general and working framework. By concatenating the CNN feature maps between support images and query images, the relation module can consider the hidden relationship between these two sets of images guided by the loss function. In the original relation network, it uses the MSE loss to compare the final probability vector to the ground truth. In this paper, we simply modify the loss to calculate pixelwise differences between the segmentation ground truth and heatmap. In
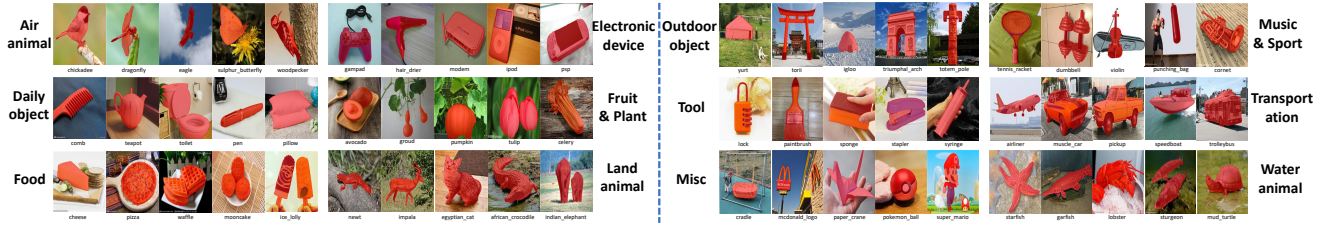
Figure 2. Example images and their corresponding segmentation in FSS-1000. For the 12 super-categories here, 5 examples are shown, where the ground-truth segmentation map is overlaid in red in the corresponding image.

OSLSM [30], the authors proposed a two-branch network to solve few-shot segmentation. The network is quite complex, and their training set was limited to the PASCAL VOC dataset with only 20 object classes. Consequently, their feature extractor may suffer severe bias making it hard to be generalized to other objects. The guided network [24] can also suffer the same limitation on their dataset choice. Though point annotation can be used to guide the training of few-shot segmentation, the sparse annotation can seriously hamper accuracy.

Semantic Image Segmentation Previous research exploiting CNN to make dense prediction often relied on patchwise training [3, 6, 23] and pre- and post-processing of superpixels [6, 11]. In [31] the authors first proposed a simple and elegant fully convolutional network (FCN) to solve semantic segmentation. Notably, this is the first work which was trained end-to-end on a fully convolutional network for dense pixel prediction, which showed that the last layer feature maps from a good backbone network such as VGG-16 contain sufficient foreground features which can be decoded by the upsampling network to produce segmentation results. Intuitively, that is also the guiding principle behind our modification on relation network architecture. Though modern network architectures [12, 14, 19] achieve high accuracy in the COCO challenge by adding complex network modules and branches, these models cannot be adapted easily to segment new classes with few training examples.

3. FSS-1000

Recent few-shot datasets [18, 35] support few-shot classification but there is no large-scale few-shot segmentation dataset. In this section, we first introduce the details of data collection and annotation, then discuss the properties of FSS-1000. Table 1 and Figure 1 compare FSS-1000 with existing popular datasets. FSS-1000 targets at solving general objects few-shot segmentation problem. So datasets only focusing on sub-domain object categories in the world (e.g. handwritten characters, human faces and road scenes) are not included in the comparison.

3.1. Data Collection

Object Classes We first referred to the classes in ILSVRC [28] in our choice of object categories for FSS-1000. Consequently, FSS-1000 has 584 classes out of

its 1,000 classes overlap with the classes in the ILSVRC dataset. We find ILSVRC dataset heavily biases toward animals, both in terms of the distribution of categories and number of images. Therefore, we fill in the other 486 by new classes unseen in any existing datasets. Specifically, we include more daily objects so that network models trained on FSS-1000 can learn from diverse artificial and man-made objects/features in addition to natural and organic objects/features where the latter was emphasized by existing large-scale datasets. Our diverse 1,000 object classes are further arranged in a hierarchy to be detailed in section 3.2.

Raw Images To avoid bias, the raw images were retrieved by querying object keywords on three different Internet search engines, namely, Google, Bing and Yahoo. We downloaded the first 100 results returned (or less if less than 100 images were returned) from a given search engine. No special criteria or assumption was used to select the candidates, however, due to the bias of Internet search engines, a large number of the images returned contain a single object photographed with sharp focus. In the final step, we intentionally included some images with a relatively small object, multiple objects or other objects in the background, and out-of-focus objects as well to balance the easy and hard examples of the dataset.

Images with aspect ratio larger than 2 or smaller than 0.5 were excluded. Since all images and their segmentation maps were to be resized to 224×224, bad aspect ratio would destroy important geometric properties after the resize operation. For the same reason, images with height or width less than 224 pixels were discarded because they would trigger upsampling which would affect the image quality after resizing.

Pixelwise Segmentation Annotation We used Photoshop's "quick selection" tool which allows users to loosely select an object automatically, and refined or corrected the selected area to produce the desired segmentation. Figure 2 shows example images overlaid with their corresponding segmentation maps in FSS-1000.

3.2. Properties

This section summarizes the three desirable properties of FSS-1000:

Scalability To extend FSS-1000 to include a new class, all it takes are 10 images with pixelwise binary segmen-

Bottom level	chiwawa	chickadee	sturgeon	comb	tulip	pizza	torii	barbell	petri_dish	pickup	poker	vacuum
Middle level	dog	bird	fish		flower			gym		car		
Top level	land animal	air animal	water animal	daily object	fruit & plant	food	outdoor object	music & sport	tool	transport-ation	misc	electronic device

Figure 3. Hierarchy of FSS-1000. Arrow represents "is a subclass of" relationship.

tation labels for the new class. This is significantly easier than other datasets such as PASCAL VOC and COCO. First, the mean number of images in a given class is much larger than 10 in these datasets. Second, in these large-scale datasets the object classes need to be first pre-defined. In other words, using data structure as an analogy, existing large-scale datasets are analogous to static array whereas FSS-1000 a dynamic linked list where new classes can be extended easily. Thus we believe binary annotation is a better annotation strategy in few-shot learning datasets, since it allows easy expansion of new object classes without concerning old object classes that have already been annotated.

Hierarchy Figure 3 shows examples of one sub-category for each given super-category in the dataset to illustrate the hierarchical structure of FSS-1000. The object classes are arranged hierarchically following a 3-level structure, while not every bottom-level subclass has a middle-level superclass. The top of the object hierarchy consists of 12 supercategories while the bottom contains the 1,000 classes as the leaf nodes. Note that this is strictly not a tree structure because a given class may belong to more than one superclass (e.g., an apple is both "fruit" and "food").

Instance FSS-1000 dataset supports instance-level segmentation with instance segmentation labels in 758 out of the 1,000 classes in the dataset, which are significantly more classes than PASCAL VOC and MS COCO. One major difference between our dataset and PASCAL VOC / MS COCO instance level segmentation is that our dataset only annotates one type of objects in one image, despite there may be other object categories appearing in the background. We annotate at most 10 instances in a single image, which follows the same instance annotation principle adopted by COCO.

4. Methodology

4.1. Problem Formulation

In few-shot learning, the train-test split is on object categories. In both training and testing , the input is divided into two sets, namely, the support set and the query set. The support set consists of samples with annotation, while the query set contains samples without annotation. In few-shot classification, the support set usually includes C classes and K training examples. This setting is defined as C-way-K-shot classification [7, 33]. In few-shot segmentation,
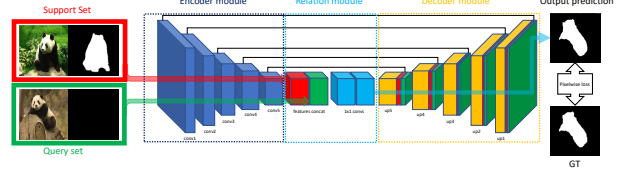

Figure 4. Our baseline network architecture using VGG-16 as backbone. The relation module is adapted from [33] where a decoder module is appended to produce the segmentation map. Both support and query features are concatenated to the decoder module via skip connection. More details of this standard architecture are available in supplemental materials.

we adopt this notation but extend the query output to be per-pixel classification of the query image, rather than a single class label. Specifically, in few-shot segmentation, the input-output pair is given by (X, Y), where

$$L = \left\{ l_{(i,j)}; l \in \{1, 2, ..., C\} \right\}$$
$$X = \left\{ (I_s, L_s, I_q); s \in \{1, 2, ..., K\} \right\}$$
$$Y = \left\{ y_{(i,j)} | I_q; y \in \{1, 2, ..., C\} \right\}$$

$l_{(i,j)}$ is the ground-truth class label and $y_{(i,j)}$ represents the predicted class label for pixel (i, j) in a given image. I_s is the 3-channel RGB support image. For each support input X with image and label pair (I_s, L_s), the model predicts a pixelwise classification map over query image I_q. Following the annotation strategy of FSS-1000, we set $C = 2$ and only focus on few-shot binary segmentation problem in this paper. However, a general C-way-K-shot segmentation could be solved by a union of C binary segmentation tasks.

4.2. Network Architecture

Pipeline Our network consists of three sub-modules: an encoder module E_θ, a relation module R_ϕ and a decoder module D_ω. For a given input X to the network, the encoder E_θ encodes the support and query images respectively into feature maps $E_\theta(I_s)$ and $E_\theta(I_q)$. For K-shot forwarding, we perform element-wise sum over the depth channels of support feature maps, so that the encoder module always produces support feature maps of the same depth regardless of the size of the support set.

The support and query feature maps are then combined in the relation module R_ϕ. We choose channel-wise concatenation as the combination operation, while other choices such as parameter regression and nearest neighbors

are possible and discussed in [24]. The relation module generates coarse segmentation results in low-resolution based on the concatenated feature maps. Finally, the coarse result is fed into the decoder module to restore the prediction map to the same resolution of the input. Figure 4 shows the entire workflow. In summary, the output is defined by

$$Y = D_\omega(R_\phi(\sum_{s=1}^{K} E_\theta(I_s), E_\theta(I_q))).$$

Loss function   We use the cross entropy loss between the query prediction output and the ground-truth annotation to train our model. Specifically, under our binary few-shot segmentation setting, binary cross entropy (BCE) loss is adopted to optimize the parameters in the network:

$$\theta^*, \phi^*, \omega^* =$$
$$\underset{\theta,\phi,\omega}{\operatorname{argmin}} \sum_i \sum_j -L_{(i,j)} \log y_{(i,j)} + (1 - L_{(i,j)}) \log(1 - y_{(i,j)})$$
(1)

Mean square error (MSE) is also a widely used objective function for semantic segmentation task. Different from BCE loss, MSE models the problem as regression to the target output. Our experiments show that BCE and MSE loss achieve similar performance under our network setting.

4.3. Network Module Details

One can design his/her own or choose any popular feature extraction backbone such as VGG-16 [32], ResNet [13] and Inception [34] as the encoder module inside the network. The support and query features compose the combined feature map whose depth is twice the channel number of the last-layer output of the encoder. The relation module utilizes two 1×1 convolutional layers on the combined feature map to embed the relationship between the support features and query features. The decoder module is designed according to the number of downscale operations in the encoder module, which applies equivalent upsample blocks to restore the resolution back to the original input. In each upsample block stands a nearest neighbor upsampling layer and a convolutional layer. Skip connection is adopted between encoder and decoder feature maps, following the scheme proposed by U-Net [26]. We find it helpful to produce fine details in segmentation when information in the encoder feature maps are fused to the decoder module by channel-wise concatenation. ReLU activation is applied throughout the deep network except for the last layer's activation where Sigmoid is used in order to scale the output to a suitable range to calculate cross-entropy loss. More detail parameters of our architecture are provided in the supplemental materials.

5. Experiments

We conduct experiments to evaluate the practicability of FSS-1000 and the performance of our method under few-

Method	MeanIoU
VGG-16-BCEloss	**80.12%**
VGG-16-MSEloss	79.66%
ResNet-101-BCEloss	79.43%
ResNet-101-MSEloss	79.12%
InceptionV3-BCEloss	79.02%
InceptionV3-MSEloss	79.22%

Table 2. Different network settings to explore the best settings for our network architecture.

Method	MeanIoU
OSLSM-1shot [30]	70.29%
OSLSM-5shot	73.02%
Guided Network-1shot [24]	71.94%
Guided Network-5shot	74.27%
Ours-1shot	73.47%
Ours-5shot	**80.12%**

Table 3. Different few-shot segmentation networks trained and tested on FSS-1000.

Method	PASCAL-5^0	PASCAL-5^1	PASCAL-5^2	PASCAL-5^3	Mean
OSLSM	34.23%	57.92%	43.20%	37.79%	43.29%
GN	33.12%	58.91%	44.26%	39.91%	44.05%
Ours	37.44%	60.94%	46.55%	42.23%	46.79%
Ours*	**50.61%**	**70.29%**	**58.43%**	**55.08%**	**58.60%**

Table 4. Comparison of different models on PASCAL-5^i. GN is Guided Network and Ours* is our model trained on FSS-1000. All models are using 5-shot setting.

shot learning settings. We evaluate models with the same network architecture but trained on different datasets to show that FSS-1000 is the best choice for few-shot segmentation task. Different support sets and their influence on query results will be discussed. Finally we illustrate that models trained on FSS-1000 are capable to generalize the few-shot segmentation knowledge to new unseen classes.

The metric we use is the intersection-over-union (IoU) of positive labels in a binary segmentation map. IoU is a standard metric and widely adopted in evaluating image segmentation methods.

All the networks are implemented in PyTorch. We use Adam solver [15] to optimize the parameters. The learning rate is initially set to 10^{-3} (10^{-4} for fine-tuning) and halved for every $50,000$ episodes. We train all the networks for $500,000$ episodes.

Network setting   To explore the best settings for our network, we train different models using a combination of different backbones and loss functions on FSS-1000. Table 2 tabulates the respective performance on VGG-16, ResNet-101 and InceptionNet as backbone, and BCE and MSE as loss function. Based on the result, we choose VGG-16 as feature extractor and use BCE loss in our model throughout the experimental section.

5.1. Benchmarks

5.1.1   FSS-1000

We train OSLSM and Guided Network on FSS-1000 to provide benchmarks and justify our dataset. Table 3 shows that our adapted relation network achieves the best results on

No.	ImageNet	fsPASCAL	fsCOCO	FSS	MeanIoU
I	✓	✓			66.45%
II	✓		✓		71.34%
III	✓			✓	79.30%
IV				✓	80.68%
V		✓		✓	81.97%
VI			✓	✓	82.66%

Table 5. Comparison of models trained on different datasets. Each model (row) shows the training stages, e.g., model I uses the pre-trained weights from ImageNet then fine-tuned on fsPASCAL. All learning rates are initially set to 10^{-4} except the model trained without using ImageNet pre-trained weights, which is set to 10^{-3}.
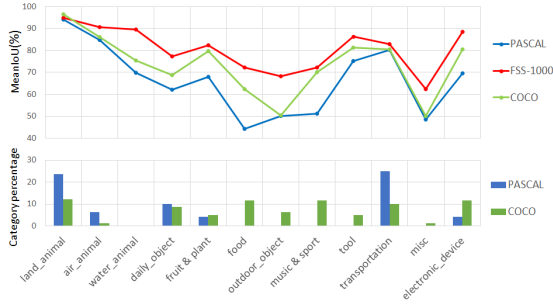

Figure 5. MeanIoU of superclasses in FSS-1000 tested with models trained on fsPASCAL, fsCOCO and FSS-1000. Bars at the bottom indicate the percentage of the number of categories overlapping with FSS-1000 in the corresponding dataset.

FSS-1000. Moreover, ours is the only model whose 5-shot training boosts the accuracy by over 10% compared to the 1-shot case. We believe that embedding multiple support images at the input end of the network and encouraging the feature extractor to consider correlation between multiple support images and the query image is the appropriate way to design k-shot ($k > 1$) segmentation network, rather than simply combining 1-shot prediction [30] or merging high-level features of multiple supports [24].

5.1.2 PASCAL-5^i

To compare with previous few-shot methods, we train and test our network on PASCAL-5^i [30]. Table 4 shows that our method marginally outperforms the others. Most importantly, we train our network merely on FSS-1000 without fine-tuning on PASCAL-5^i to avoid any potential overfitting, and this model achieves much better results compared to models trained on PASCAL-5^i, which justify the effectiveness of FSS-1000.

5.2. Effect of Pre-training

We compare our network model trained on different datasets to demonstrate the effectiveness of FSS-1000 in few-shot segmentation. Since there are no publicly available few-shot image segmentation datasets, we convert PASCAL VOC 2012 and COCO datasets by setting the desired foreground class label as positive and all others as negative, followed by the identical clean-up stage described in section 3.1 to the binarized labels. Two new datasets are
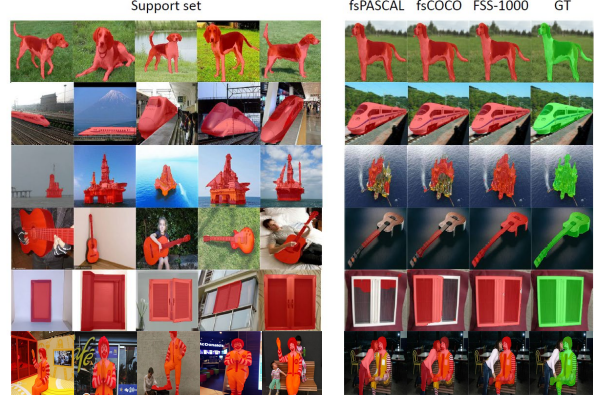

Figure 6. Image results of our baseline model respectively trained on fsPASCAL, fsCOCO and FSS-1000. Support labels and predicted segmentation are overlaid in red in corresponding support images and query images. Ground truth labels for query images are in green. The classes in the first two rows are present in fs-PASCAL and fsCOCO whereas the rest are unique in FSS-1000.
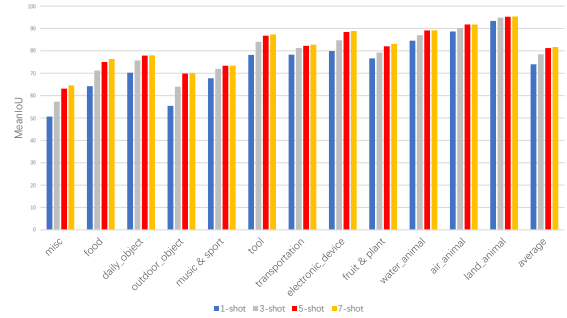

Figure 7. MeanIoU of superclasses in FSS-1000 tested with k-shot models (k = 1,3,5,7).

thus produced: fsPASCAL and fsCOCO. There are respectively 4,318 image and label pairs in 20 object classes in fsPASCAL, and 48,015 image and label pairs in 80 object classes in fsCOCO. All available data in fsPASCAL and fs-COCO are used in training.

For FSS-1000, we build the validation/test set by randomly sampling 20 distinct sub-categories from the 12 super-categories, while the other images and labels are used in training. The train/validation/test split used in the experiments consists of 5,200/2,400/2,400 image and label pairs.

Table 5 tabulates the performance of different models. For each model (row), the ✓marks in sequence indicate the dataset(s) used in pre-training stages with the last mark indicating the dataset used in fine-tuning. Model IV has only one ✓indicating that it is exclusively trained on the dataset.

Using the pre-trained weights from ImageNet, Model III trained on FSS-1000 outperforms respectively the fsPAS-CAL model I and fsCOCO model II by over a large margin of 20% and 10%. Notably, *without* using any pre-trained weights Model IV achieves slightly better results compared to Model III, which substantiate our claim that bias in feature extractor does exist in models pre-trained and/or trained on a dataset unevenly distributed in object categories and
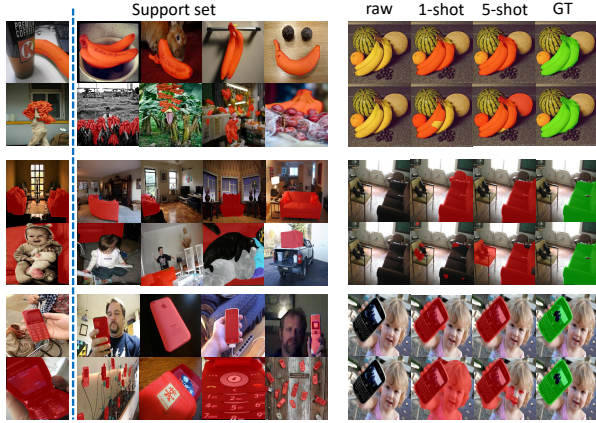
Figure 8. Effect of different support sets. The leftmost support of each row is used to generate 1-shot results. For each class, we show the result of a good support set followed by a bad support set in the next row.

	Human (PS)	Human (GrabCut)	CPU	GPU
Time	180m32s	53m22s	9m13s	**16.9s**
95%+ IOU	100%	71.4%	58.4%	58.4%
90%+ IOU	100%	80.4%	70.4%	70.4%
80%+ IOU	100%	91.0%	87.4%	87.4%
70%+ IOU	100%	95.8%	90.2%	90.2%

Table 6. 500 test images are randomly sampled from FSS-1000 to compare time and accuracy performance of labeling segmentation data between humans and few-shot model.

images within each class.

Interestingly, Model VI pre-trained on COCO and fine-tuned on FSS-1000 achieves the best result, outperforming the model III pre-trained on ILSRVC. We believe this is due to the difference in requirement of feature maps ideal for classification and segmentation task. Intuitively, semantic segmentation requires more accurate low-level features to produce fine details in segmentation map, while classification focuses on high-level features for image understanding.

Overall, models respectively trained on fsPASCAL and fsCOCO produce quite good results in object classes that are included in PASCAL and COCO, or similar to PASCAL and COCO classes. For these classes, sometimes their segmentation results are better in local details compared to the results produced by models trained on FSS-1000 due to more variations in the support training set. However, they respectively fail in classes significantly different from the 20 PASCAL classes and 80 COCO classes. The somewhat limited variation in object categories in existing datasets makes it hard for models trained on them to generalize to more unseen classes under the few-shot setting. On the other hand, models trained on FSS-1000 classes can handle these cases. Quantitative results and qualitative results are shown in Figure 5 and Figure 6 respectively. Among the super categories in FSS-1000, animals are some of the easiest cases due to their relatively obvious structural features and less geometric variations, while daily objects with abundant scale and pose variations make most hard cases.

5.3. Effect of Support Set

We train four different models, using 1, 3, 5 and 7 support images respectively, to study how different number of support images influence the accuracy of few-shot segmentation. Two important observations can be summarized from Figure 7.

First, more support images generally boost the segmentation accuracy because more variations of color, pose, and scale of the object are included. However, the performance increase becomes negligible when more than 5 support images are given. Due to this bottleneck effect, we set up most of the experiments under the 5-shot setting.

Second, the accuracy boost is different among different classes. For easy cases (e.g. animals), the improvement is not obvious because a single support image is enough for the deep network to capture and distinguish strong features of the object. For hard cases (e.g. deformable objects), more support images are essential for the network to learn the complex shapes to make correct segmentation.

Figure 8 demonstrates the effect of support set, which shows that scale and pose of the object to be segmented are the most important characteristics to guide few-shot semantic segmentation on FSS-1000. Since FSS-1000 does not explicitly consider scale variations (future work), a tiny or oversized object in the support set is not a good reference for segmentation. Significant differences in scales can mislead the network to capture wrong feature contents in the query. Besides, significantly different poses in support and query sets can result in bad segmentation results, due to the intrinsic fragility to rotation in CNN features.

5.4. Auto-Labeling on Novel and Unseen Classes

Traditionally a large number of human-annotated images are required to train a deep network for segmenting a new class. Table 6 tabulates the tradeoff in time and accuracy for annotating 500 test images in FSS-1000 by humans (using Photoshop and GrabCut [27] algorithm) and our few-shot segmentation.

With its good accuracy and time tradeoff, despite the current limitations in scale invariance aforementioned, FSS-1000 allows us to automatically segment a novel object category by just providing a few support examples *without* retraining or fine-tuning a given model. We pick a number of very novel classes unseen by FSS-1000, and label 5 images of each class serving as the support set. Figure 9 shows the test results which demonstrates that our model trained on FSS-1000 is capable of generalizing to these unseen classes. More extensive results on novel classes are included in supplementary materials.

For example, *android robot* is an unreal object unseen in FSS-1000. In cartography from satellite images which often come in overlapping image tiles, cartographers need to label only 5 images or tiles and our system can automatically segment the rest, such as recognizing *river* in our exam-
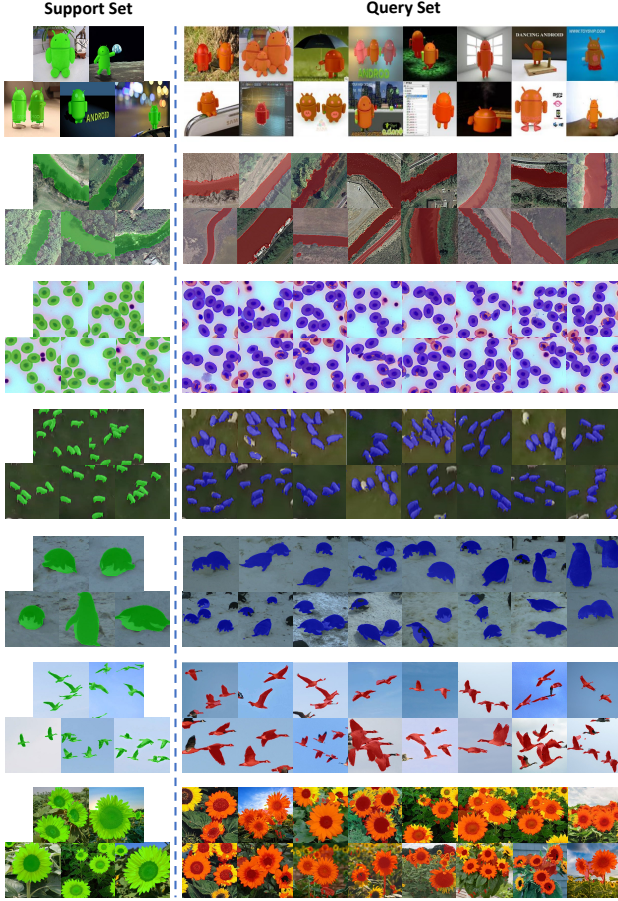
Figure 9. Test results for unseen classes. From top to bottom: *android robot*; the *river* from UC Merced Land Use Dataset [37]; a large *cell* image cropped into patches; herds of *sheep*; *penguin* from Oxford penguin counting dataset [1]; flock of wild *goose*; different images of fields of *sunflower* depict various scales in the presence of occlusion and perspective distortion.

ple where saliency detection does not work in general. The *cell* example shows the good potential of FSS-1000 in instance segmentation which significantly contributes to cell counting in medical image analysis where, for instance, a patient's health directly correlates to his or her red blood cell count. With the advance of whole slide images (WSI) in which the width and height often exceed 100,000 pixels (and thus many cells to count), using our few-shot segmentation trained on FSS-1000, pathologists only need to label 5 image relevant regions and then the rest of the WSI will be automatically labeled. Although manual corrections for missed or wrong cells may still be necessary given the current accuracy, comparing with exhaustive labeling which requires hours or even days to complete, the potential contribution of FSS-1000 is substantial. Similarly, the related animal examples of *sheep*, *penguin* and wild *goose* show FSS-1000's potential for large-scale instance segmentation. Finally, presently our baseline backbone network is not very robust to scale variance, occlusion and background noises
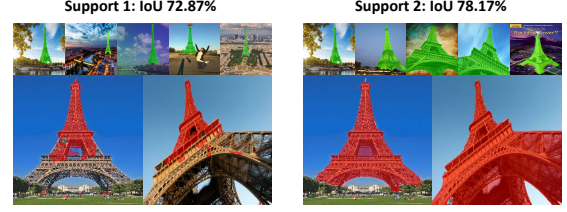

Figure 10. Iterative few-shot segmentation. Left and right show respectively the support sets and results before and after including corrected failure cases in the support set. The whole testing set of *Eiffel Tower* can be found in the supplemental material.

(future work). In *sunflower*, the segmentation results for instances too big or too small (especially for images with depth of field where faraway sunflowers are out of focus) become incomplete or even totally omitted. Despite that, FSS-1000 still reports limited success, while occluded instances are not labeled completely, and the background produces fake positive around the instance, as shown in some of the failure examples here.

5.5. Iterative Few-Shot Segmentation

Our few-shot segmentation can successively benefit from support sets improved easily by including failure cases after correction in each pass. Consider the Eiffel Tower unseen by FSS-1000 in Figure 10 where we manually label 200 images for quantitative evaluation (IoU). The first support set (left) did not have sufficient view and scale variations and did not see clearly the bottom part of the tower which resulted in its incomplete segmentation in some test cases. After mining a few of such hard cases, correcting and including them in the second support set (right), the previous hard cases could now be correctly segmented. While more theoretical studies may be required, few-shot segmentation performed in stages may offer an immediate performance boost.

6. Conclusion

Few-shot learning/segmentation is an emerging attractive alternative, where prediction is made given only a few training examples. However, there is no existing large-scale dataset for few-shot segmentation. In this paper, we address the limitation of existing large-scale datasets in their biases and lack of scalability, and build the first few-shot segmentation dataset FSS-1000 emphasizing class diversity rather than dataset size. We adapt the relation network architecture to few-shot segmentation. This baseline few-shot segmentation model, even trained exclusively on FSS-1000 without using pre-trained weights, achieves higher accuracy than previous methods. We further demonstrated the efficacy and potential of FSS-1000 in large-scale segmentation on totally unseen classes without re-training or fine-tuning, and showed its promise on few-shot instance segmentation and iterative few-shot recognition tasks.

References

[1] C. Arteta, V. Lempitsky, and A. Zisserman. Counting in the wild. In *ECCV*, 2016. 8

[2] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. *NIPS*, 2016. 2

[3] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, 2012. 3

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 2

[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013. 3

[7] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 2006. 4

[8] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2

[9] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 1

[10] I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett. Prototypical networks for few-shot learning. *NIPS*, 2017. 2

[11] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik. Simultaneous detection and segmentation. *ECCV*, 2014. 3

[12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask r-cnn. *ICCV*, 2017. 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016. 5

[14] D. Jifeng, H. Kaiming, and S. Jian. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 3

[15] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ICLR*, 2015. 5

[16] G. R. Koch. Siamese neural networks for one-shot image recognition. In *ICML Workshop*, 2015. 2

[17] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 2

[18] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2, 3

[19] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 3

[20] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *ICCV*, pages 2999–3007, 2017. 1

[21] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft COCO: Common Objects in Context. *ArXiv e-prints*, May 2014. 1, 2

[22] T. Munkhdalai and H. Yu. Meta networks. *ICML*, 2017. 2

[23] P. H. O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014. 3

[24] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine. Few-shot segmentation propagation with guided networks. *ICLR Workshop*, 2018. 1, 3, 5, 6

[25] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2017. 2

[26] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2, 5

[27] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 7

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 1, 2, 3

[29] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 2

[30] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017. 1, 2, 5, 6

[31] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *TPAMI*, 2017. 3

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 5

[33] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2, 4

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015. 5

[35] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS*, 2016. 2, 3

[36] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. Sun database: Exploring a large collection of scene categories. *IJCV*, 2014. 2

[37] Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. *ACM GIS*, 2010. 8