CANet: Class-Agnostic Segmentation Networks with Iterative Refinement and Attentive Few-Shot Learning

Chi Zhang[1], Guosheng Lin[1]*, Fayao Liu[2], Rui Yao[3], Chunhua Shen[2]
[1]Nanyang Technological University, Singapore
[2]The University of Adelaide, Australia
[3]China University of Mining and Technology, China
E-mail: chi007@e.ntu.edu.sg , gslin@ntu.edu.sg

Abstract

Recent progress in semantic segmentation is driven by deep Convolutional Neural Networks and large-scale labeled image datasets. However, data labeling for pixel-wise segmentation is tedious and costly. Moreover, a trained model can only make predictions within a set of pre-defined classes. In this paper, we present CANet, a class-agnostic segmentation network that performs few-shot segmentation on new classes with only a few annotated images available. Our network consists of a two-branch dense comparison module which performs multi-level feature comparison between the support image and the query image, and an iterative optimization module which iteratively refines the predicted results. Furthermore, we introduce an attention mechanism to effectively fuse information from multiple support examples under the setting of k-shot learning. Experiments on PASCAL VOC 2012 show that our method achieves a mean Intersection-over-Union score of 55.4% for 1-shot segmentation and 57.1% for 5-shot segmentation, outperforming state-of-the-art methods by a large margin of 14.6% and 13.2%, respectively.

1. Introduction

Deep Convolutional Neural Networks have made significant breakthroughs in many visual understanding tasks including image classification [13, 9, 30], object detection [27, 8, 26], and semantic segmentation [16, 2, 20]. One crucial reason is the availability of large-scale datasets such as ImageNet [4] that enable the training of deep models. However, data labeling is expensive, particularly for dense prediction tasks, *e.g.*, semantic segmentation and instance segmentation. In contrast to machine learning algorithms, humans are able to segment a new concept from the image easily when only seeing a few examples. The gap between humans and machine learning algorithms motivates the study of few-shot learning that aims to learn a model which can be generalized well to new classes with scarce labeled training data.

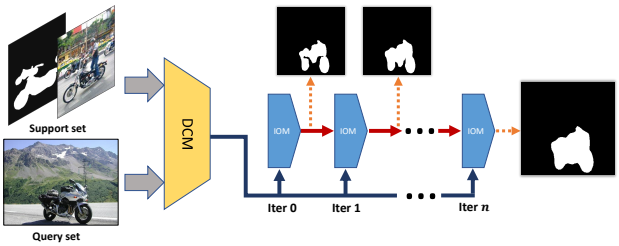

Figure 1 – Overview of our proposed network for 1-shot segmentation. Our framework consists of a dense comparison module (DCM) and an iterative optimization module (IOM). Given only one annotated training image, our network is able to segment test images with new classes and iteratively optimize the results.

In this paper, we undertake the task of few-shot semantic segmentation that only uses a few annotated training images to perform segmentation on new classes. Previous work [29, 24, 5] on this task follows the design of two-branch structure which includes a support branch and a query branch. The support branch aims to extract information from the support set to guide segmentation in the query branch. We also adopt the two-branch design in our framework to solve the few-shot segmentation problem.

Our network includes a two-branch dense comparison module, in which a shared feature extractor extracts representations from the query set and the support set for comparison. The design of the dense comparison module takes inspiration from metric learning [37, 31] on image classification tasks where a distance function evaluates the similarity between images. However, different from image classification where each image has a label, image segmentation needs to make predictions on data with structured representation. It is difficult to directly apply metric learning to dense prediction problems. To solve this, one straight-

*Guosheng Lin is the corresponding author.

forward approach is to make comparisons between all pairs of pixels. However, there are millions of pixels in an image and comparison of all pixel pairs takes enormous computational cost. Instead, we aim to acquire a global representation from the support image for comparison. Here, to only focus on the assigned category, we use global average pooling over the foreground area to filter out irrelevant information. Then the global feature is compared with each location in the query branch, which can be seen as a dense form of the metric learning approach.

Under the few-shot setting, the network should be able to handle new classes that are never seen during training. Thus we aim to mine transferable representations from CNNs for comparison. As is observed in feature visualization literature [39, 38], features in lower layers relate to low-level cues, *e.g.*, edges and colors while features in higher layers relate to object-level concepts such as categories. We focus on middle-level features that may constitute object parts shared by unseen classes. For example, if the CNN learns a feature that relates to *wheel* when the model is trained on the class *car*, such feature may also be useful for feature comparison on new vehicle classes, *e.g.*, *truck* and *bus*. We extract multiple levels of representations in CNNs for dense comparison.

As there exist variances in appearance within the same category, objects from the same class may only share a few similar features. Dense feature comparison is not enough to guide segmentation of the whole object area. Nevertheless, this gives an important clue of where the object is. In semi-automatic segmentation literature, weak annotations are given for class-agnostic segmentation, *e.g.*, interactive segmentation with click or scribble annotations [36, 14] and instance segmentation with bounding box or extreme point priors [10, 21]. Transferable knowledge to locate the object region is learned in the training process. Inspired by semi-automatic segmentation tasks, we hope to gradually differentiate the objects from the background given the dense comparison results as priors. We propose an iterative optimization module (IOM) that learns to iteratively refine the predicted results. The refinement is performed in a recurrent form that the dense comparison result and the predicted masks are sent to an IOM for optimization, and the output is sent to the next IOM recurrently. After a few iterations of refinement, our dense comparison module is able to generate fine-grained segmentation maps. Inside each IOM, we adopt residual connections to efficiently incorporate the predicted masks in the last iteration step. Fig. 1 shows an overview of our network for one-shot segmentation.

Previous methods for k-shot segmentation is based on the 1-shot model. They use non-learnable fusion methods to fuse individual 1-shot results, *e.g.*, averaging 1-shot predictions or intermediate features. Instead, we adopt an attention mechanism to effectively fuse information from multiple support examples.

To further reduce the labeling efforts for few-shot segmentation, we explore a new test setting: our model uses the bounding box annotated support set to perform segmentation in the query image. We conduct comprehensive experiments on the PASCAL VOC 2012 dataset and COCO dataset to validate the effectiveness of our network. Main contributions of this paper are summarized as follows.

- We develop a novel two-branch dense comparison module which effectively exploits multiple levels of feature representations from CNNs to make dense feature comparison.
- We propose an iterative optimization module to refine predicted results in an iterative manner. The ability of iterative refinement can be generalized to unseen classes with few-shot learning for generating fine-grained maps.
- We adopt an attention mechanism to effectively fuse information from multiple support examples in the k-shot setting, which outperforms non-learnable fusion methods of 1-shot results.
- We demonstrate that given support set with weak annotations, *i.e.*, bounding boxes, our model can still achieve comparable performance to the result with expensive pixel-level annotated support set, which further reduces the labeling efforts of new classes for few-shot segmentation significantly.
- Experiments on the PASCAL VOC 2012 dataset show that our method achieves a mean Intersection-over-Union score of 55.4% for 1-shot segmentation and 57.1% for 5-shot segmentation, which significantly outperform state-of-the-art results by 14.6% and 13.2%, respectively.

2. Related Work

Few-shot learning. Few-shot learning aims to learn transferable knowledge that can be generalized to new classes with scarce labeled training data. There exist many formulations on few-shot classification, including recurrent neural network with memories [28, 23], learning to fine-tune models [6, 25], network parameter prediction [1, 35], and metric learning [31, 37, 11]. Metric learning based methods achieve state-of-the-art performance in the few-shot classification tasks and they have the trait of being fast and predicting in a feed-forward manner. Our work is most related to Relation Network [37]. Relation Network meta-learns a deep distance metric to compare images and compute the similarity score for classification. The network consists of an embedding module which generates the representations of the images and a relation module that compares the embeddings and outputs a similarity score. Both modules are in the form of convolutional operations. The

dense comparison module in our network can be seen as an extension of Relation Network in a dense form to tackle the task of segmentation.

Few-shot semantic segmentation. Fully supervised semantic segmentation is the task of classifying each pixel in an image to a set of pre-defined categories [16, 2, 20, 15, 17]. Few-shot semantic segmentation, on the other hand, aims to generalize the segmentation ability to any new categories with only a few annotated examples. Previous work on few-shot semantic segmentation employs two-branch structures. Shaban *et al*. [29] first adopt few-shot learning on semantic segmentation. The support branch directly predicts the weights of the last layer in the query branch for segmentation. In [24], the support branch generates an embedding which is fused to the query branch as additional features. Our network also follows the two-branch design. However, different from previous work where two branches have different structures, the two branches in our network share the same backbone network. The models in previous methods focus on the 1-shot setting, and when extending 1-shot to k-shot, they apply 1-shot method independently to each support example and use non-learnable fusion methods to fuse individual predicted results at the image level or feature level. For example, Shaban *et al*. [29] propose to use logic OR operation to fuse individual predicted masks and Rakelly *et al*. [24] average the embedding in the support branch generated by different support examples. Instead, we adopt a learnable method through an attention mechanism to effectively fuse information from multiple support examples.

3. Task Description

Suppose that our model is trained on a dataset with the class set $\mathcal{C}_{train}$, our goal is to use the trained model to make the prediction on a different dataset with new classes $\mathcal{C}_{test}$ where only a few annotated examples are available. Intuitively, we train the model to have the ability that for a new class $c \notin \mathcal{C}_{train}$, our model is able to segment the class from the images when only sees a few pictures of this class. Once the model is trained, the parameters are fixed and require no optimization when tested on a new dataset.

We align training and testing with the episodic paradigm [33] to handle the few-shot scenario. Specifically, given a k-shot learning task, each episode is constructed by sampling 1) a support (training) set $\mathcal{S} = \{(x_s^i, y_s^i(c))\}_{i=1}^k$, where $x_s^i \in \mathbb{R}^{H_i \times W_i \times 3}$ is an RGB image and $y_s^i(c) \in \mathbb{R}^{H_i \times W_i}$ is a binary mask for class c in the support image; and 2) a query (test) set $\mathcal{Q} = \{x_q, y_q(c)\}$ where x_q is the query image and $y_q(c)$ is the ground-truth mask for class c in the query image. The input to the model is the support set $\mathcal{S}$ and the query image x_q, and the output is the predicted mask $\hat{y}_q(c)$ for class c in the query image. As there may be multiple classes in one query image x_q, the ground truth query mask is different when a different label c is assigned. Fig. 1 shows an illustration of the task when $k = 1$.

4. Method

We propose a new framework that solves the few-shot semantic segmentation problem. We begin with the illustration of our model in the 1-shot setting first without loss of generality. Our network consists of two modules: the dense comparison module (DCM) and the iterative optimization module (IOM). The DCM performs dense feature comparison between the support example and the query example, while IOM performs iterative refinement of predicted results. Fig. 2 (a) shows an overview of our framework. To generalize our network from 1-shot learning to k-shot learning, we adopt an attention mechanism to fuse information from different support examples. Moreover, we propose a new test setting that uses support images with bounding box annotations for few-shot segmentation, which is described subsequently.

4.1. Dense Comparison Module

We develop a two-branch dense comparison module that densely compares each position in the query image with the support example, as shown in Fig. 2 (b). The module consists of two sub-modules: a feature extractor that extracts representations and a comparison module that performs feature comparison.

Feature extractor. The feature extractor aims to harvest different levels of representations from CNNs for feature matching. We use a ResNet-50 [9] as the backbone of the feature extractor. As done in previous few-shot segmentation work, the backbone model is pre-trained on Imagenet [4]. As is observed in CNN feature visualization literature [39, 38], features in lower layers often relate to low-level cues, *e.g*., edges and colors while features in higher layers relate to object-level concepts such as object categories. In the few-shot scenario, our model should adapt to any unseen classes. Thus we can not assume that a feature corresponding to an unseen category is learned during training. Instead, we focus on middle-level features that may constitute object parts shared by unseen classes. The layers in ResNet are divided into 4 blocks based on the spatial resolution which naturally correspond to 4 different levels of representation. We choose features generated by $block2$ and $block3$ for feature comparison and abandon layers after $block3$. We use dilated convolutions [2] in layers after $block2$ to maintain the spatial resolution of feature maps. All feature maps after $block2$ have a fixed size of 1/8 of the input image. Features after $block2$ and $block3$ are concatenated and encoded to 256 dimensions by 3×3 convolutions. We investigate the choice of features for comparison in Section 5.1.3. Both the support branch and the query branch use

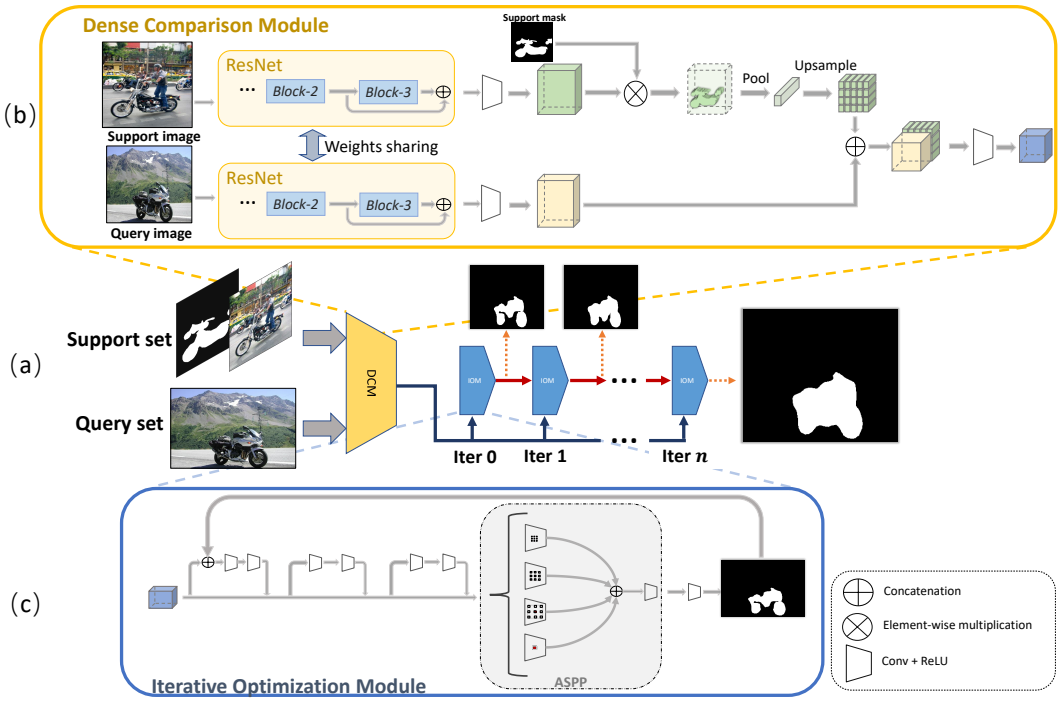

Figure 2 – CANet for 1-shot semantic segmentation. (a) Overview of our network structure. (b) Dense Comparison Module. (c) Iterative Optimization Module.

the same feature extractor. We keep the weights in ResNet fixed during training.

Dense Comparison. As there may be multiple object categories and cluttered backgrounds in the support image, we want to acquire an embedding that only corresponds to the target category for comparison. Here, we use global average pooling over the foreground area to squeeze the feature maps to a feature vector. Global image features turn out to be useful in segmentation tasks [19, 40, 3], which can be easily achieved by global average pooling. In our network, we only average features over the foreground area to filter out irrelevant areas. After we obtain the global feature vector from the support set, we concatenate the vector with all spatial locations in the feature map generated by the query branch. This operation aims to compare all the spatial locations in the query branch to the global feature vector from the support branch. Then, the concatenated feature maps go through another convolutional block with 256 3×3 convolutional filters for comparison.

For efficient implementation, we first bilinearly downsample the binary support mask to the same spatial size of the feature maps and then apply element-wise multiplication with the feature maps. As a result, features belonging to the background area become zero. Then we adopt global sum pooling and divide the resulting vector by the foreground area to obtain the average feature vector. We upsample the vector to the same spatial size of query features and concatenate them for dense comparison.

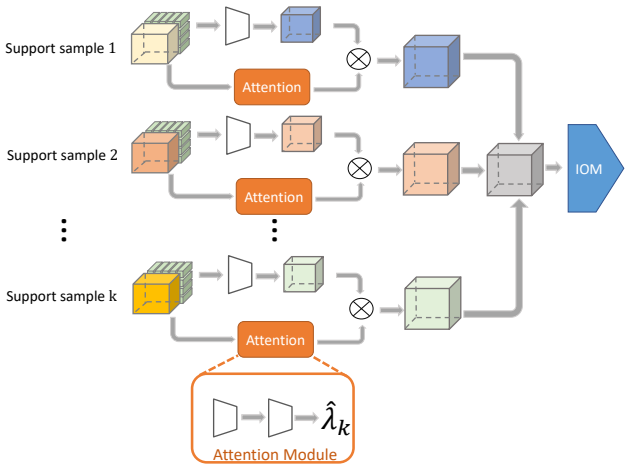

Figure 3 – Attention mechanism for k-shot semantic segmentation. We use the softmax function to normalize the outputs of the attention module from different support examples.

4.2. Iterative Optimization Module

As there exist variances in appearance within the same category, dense comparison can only match a part of the object, which may not be sufficiently powerful to accurately segment the whole object in the image. We observe that the initial prediction is an important clue about the rough position of the objects. We propose an iterative optimization module to optimize the predicted results iteratively. The structure is shown in Fig. 2 (c). The module's input is

the feature maps generated by the dense comparison module and predicted masks from the last iteration. Directly concatenating feature maps with predicted masks as additional channels causes mismatch to the feature distribution as there is no predicted mask for the first forward pass. Instead, we propose to incorporate the predicted masks in a residual form:

$$M_t = x + F(x, y_{t-1}), \tag{1}$$

where x is the output feature of the dense comparison module; y_{t-1} is the predicted masks from the last iteration step, and M_t is the output of the residual block. Function $F(\cdot)$ is the concatenation of feature x and predicted masks y_{t-1}, followed by two 3×3 convolution blocks with 256 filters. Then we add two vanilla residual blocks with the same number of convolutional filters. On top of that, we use Atrous Spatial Pyramid Pooling module (ASPP) proposed in Deeplab V3 [3] to capture multi-scale information. The module consists of four parallel branches that include three 3×3 convolutions with atrous rates of 6, 12, and 18 respectively and a 1×1 convolution. The 1×1 convolution is operated on the image-level feature which is achieved by global average pooling. Then the resulting vector is bilinearly upsampled to the original spatial size. The output features from 4 branches are concatenated and fused by another 1×1 convolution with 256 filters. Finally, we use 1×1 convolution to generate the final masks which include a background mask and a foreground mask. We use a softmax function to normalize the score in each location, which outputs the confidence maps of the foreground and the background. The confidence maps are then fed to the next IOM for optimization. Our final result is achieved by bilinearly upsampling the confidence map to the same spatial size of the query image and classifying each location according to the confidence maps. At the training time, to avoid the iterative optimization module over-fitting the predicted masks, we alternatively use predicted masks in the last epoch and empty masks as the input to IOM. The predicted masks y_{t-1} is reset to empty masks with a probability of p_r. This can be seen as dropout of the whole mask, an extension of the standard dropout [32]. In comparison to previous iterative refinement methods in segmentation literature [14, 34, 22], our method integrates the refinement scheme into the model with residual connection so that the whole model could run in a feed-forward manner and is trained end-to-end.

4.3. Attention Mechanism for k-shot Segmentation

In order to efficiently merge information in the k-shot setting, we use an attention mechanism to fuse the comparison results generated by different support examples. Specifically, we add an attention module parallel to the dense comparison convolution in DCM (see Fig. 3). The attention branch consists of two convolutional blocks. The first one has 256 3×3 filters, followed by 3×3 max pooling. The second one has one 3×3 convolution followed by a global average pooling. The result from the attention branch serves as the weight λ. Then, the weights from all support examples are normalized by a softmax function:

$$\hat{\lambda}_i = \frac{e^{\lambda_i}}{\sum_{j=1}^{k} e^{\lambda_j}}. \tag{2}$$

The final output is the weighted sum of features generated by different support samples.

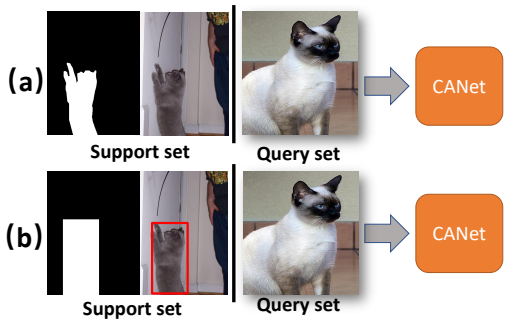

Figure 4 – (a) CANet with pixel-wise annotated support set. (b) CANet with bounding box annotated support set.

4.4. Bounding Box Annotations

As the essence of our dense comparison module is to densely compare each location in the query image to the global representation provided by the support example, we explore a new form of support set annotation that uses bounding boxes. Compared with pixel-wise annotations, the bounding box annotation uses a rectangular box to denote the object area, which is often used in object detection tasks. Labeling bounding box annotations is much cheaper than pixel-wise labeling. We relax the support set by treating the whole bounding box area as the foreground. We test our model under this setting to evaluate the capability of our framework. The comparison of the two test settings is shown in Fig. 4.

5. Experiments

To evaluate the performance of our proposed method, we conduct extensive experiments on the PASCAL VOC 2012 dataset and COCO dataset. Our network is trained end-to-end. The loss function is the mean of cross-entropy loss over all spatial locations in the output map. Our network is trained using SGD for 200 epochs with the PyTorch library on Nvidia Tesla P100 GPUs. We set the learning rate to 0.0025 and set probability p_r to 0.7. We use a mini-batch of 4 episodes for training on PASCAL-5^i and 8 on COCO. At inference time, we iteratively optimize the predicted results for 4 times after the initial prediction.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
OSLSM[29]	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	43.9
CANet	**52.5**	**65.9**	**51.3**	**51.9**	**55.4**	**55.5**	**67.8**	**51.9**	**53.2**	**57.1**

(a) 1-shot and 5-shot results under the meanIoU evaluation metric.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
OSLSM [29]	-	-	-	-	61.3	-	-	-	-	61.5
co-FCN [24]	-	-	-	-	60.1	-	-	-	-	60.2
PL [5]	-	-	-	-	61.2	-	-	-	-	62.3
CANet	**71.0**	**76.7**	**54.0**	**67.2**	**66.2**	**74.2**	**80.3**	**57.0**	**66.8**	**69.6**

(b) 1-shot and 5-shot results under the FB-IoU evaluation metric.

Table 1 – Results on the PASCAL-5^i dataset. Our proposed method outperforms all previous methods under both evaluation metrics and sets a new state-of-the-art performance (bold).

Evaluation metric. There is a minor difference of evaluation metrics in previous work. Shaban *et al*. [29] measure the per-class foreground Intersection-over-Union (IoU) and use the average IoU over all classes (meanIoU) to report the results. While in [24, 5], they ignore the image categories and calculate the mean of foreground IoU and background IoU over all test images (FB-IoU). We choose the meanIoU evaluation metric for our analysis experiments due to the following reasons: 1) The numbers of test samples in different classes are not balanced (*e.g*., 49 of class *sheep vs.* 378 of class *person*). Ignoring the image categories may lead to a biased result towards the class with more images. Also, we can observe the effectiveness of our model in different classes with the meanIoU evaluation metric. 2) As most objects are small relative to the whole image, even though the model fails to segment any objects, the background IoU can still be very high, thus failing to reflect the capability of the model. 3) Foreground IoU is more often used in binary segmentation literature (*e.g*., video segmentation and interactive segmentation). Nevertheless, we still compare our results with previous work under both evaluation metrics.

5.1. PASCAL-5^i

PASCAL-5^i is a dataset for few-shot semantic segmentation proposed in [29]. It is built on images from PASCAL VOC 2012 and extra annotations from SDS [7]. 20 object categories from PASCAL VOC are evenly divided into 4 splits with three splits for training and one split for testing. At test time, 1000 support-query pairs are randomly sampled in the test split. More details of PASCAL-5^i can be found in [29].

5.1.1 Comparison with the State-of-the-art Methods

We compare our model with the state-of-the-art methods in Table 1. Table 1 (a) shows the results evaluated under the meanIoU evaluation metric and Table 1 (b) shows the results under the FB-IoU metric. For the performance of [29] under the FB-IoU metric, we quote the result reproduced in [24]. *Our model significantly outperforms the state-of-the-art methods under both evaluation metrics*. Particularly, our meanIoU score outperforms the state-of-the-art results by 14.6% for the 1-shot task and 13.2% for the 5-shot task.

Annotation	Result (meanIoU %)
Pixel-wise labels	54.0
Bounding box	52.0

Table 2 – Evaluation with different support set annotations. Our model with bounding box annotated support set can achieve comparable performance to the result with pixel-wise annotations

Qualitative Results. Fig. 5 shows some qualitative examples of our segmentation results. Note that given the same query image, our model is able to segment different classes when different support examples are presented (See the 5th and the 6th examples in Fig. 5).

5.1.2 Experiments on Bounding Box Annotations

We evaluate CANet with the bounding box annotated support set at test time. We acquire bounding box annotations from the PASCAL VOC 2012 dataset and SDS [7]. The support mask is the region inside the bounding box of one instance instead of all instances in a support image. The instance is chosen randomly. As is shown in Table 2, *the performance with bounding box annotated support set is comparable to the result with expensive pixel-level annotated support set*, which means our dense comparison module is able to withstand noise introduced by the background area within the bounding box.

5.1.3 Ablation Study

We implement extensive ablation experiments on the PASCAL-5^i dataset to inspect the effectiveness of different components in our network. All results are average meanIoU over 4 splits on the PASCAL-5^i dataset.

Features for Comparison. In Table 3, we compare our model variants that use different levels of feature in

Figure 5 – Qualitative examples of 1-shot segmentation on the PASCAL-5^i dataset. The first row is query images and support images (right bottom) with ground-truth annotations. The second row is our predicted results. Note that the 5th and the 6th examples have the same query images and our model is able to segment different classes when different support examples are presented.

*block*2	*block*3	*block*4	meanIoU
✓			46.6
	✓		50.8
		✓	48.4
✓	✓		**51.2**
	✓	✓	49.2
✓		✓	49.6
✓	✓	✓	49.5

Table 3 – Ablation experiments on the choice of features in ResNet for comparison. The combination of features after *block2* and *block3* achieves the best result.

Method	Result (meanIoU %)
CANet-Init	51.2
CANet-Init + DenseCRF	51.9
CANet	**54.0**

Table 4 – Ablation experiments on the iterative optimization module. CANet-Init denotes the initial prediction from CANet without additional optimization. Our iterative optimization scheme outperforms the baseline models by 2.8% and is more effective in refining the segmentation maps than DenseCRF.

ResNet-50 for feature comparison. In all cases, we encode the features to 256 dimensions before comparison and we do not adopt iterative optimization. We experiment feature comparison with single block and multiple blocks. When single block is used for comparison, *block3* performs the best. When multiple blocks are used for comparison, the combination of *block2* and *block3* achieves the best result. The reason is that *block2* corresponds to relatively low-level cues, which alone is not enough to match object parts. While *block4* corresponds to high-level features, *e.g.*, categories, and incorporates a great number of parameters (2048 channels), which makes it hard to optimize under the few-shot setting. The combination of *block2* and block3 is the best for matching class-agnostic object parts. We also implement experiments with VGG16 as the feature extractor. We choose features of stage 2, 3, and 4 (out of 5). The final multi-scale test result with VGG as the backbone is 54.3%. Compared with the ResNet50 version (55.4%), the performance only drops by 1.1% and still significantly outperforms the state-of-the-art results.

Iterative Optimization Module. To validate the effectiveness of our proposed iterative optimization module, we compare our network with a baseline model that does not employ additional IOM for optimization, *i.e.*, the initial prediction from CANet(CANet-Init). We also compare our iterative optimization scheme with DenseCRF [12], which is a post-processing method widely used in segmentation literature to refine segmentation maps. Table 4 shows the results of different model variants. As is shown, the iterative optimization yields 2.8% improvement over the initial prediction. DenseCRF does not significantly improve the few-shot segmentation prediction. We visualize the results and find that for the predicted masks which successfully locate most of the object region, DenseCRF can effectively improve segmentation results, particularly in the region of object boundaries. However, for failure masks, *e.g.*, false localization of objects, DenseCRF expands false positive regions, which deteriorates the IoU score. Our IOM, on the other hand, can effectively fill the object region and remove irrelevant areas in a learnable way. We visualize the intermediate results of our iterative optimization process in Fig. 6.

Attention *vs.* Feature Fusion *vs.* Mask Fusion. In the k-shot setting, we compare our attention mechanism to several solutions in previous work: 1) Feature-level average fusion. We experiment the method in [24], which is to average the features generated by different support examples. 2) Logic OR fusion for masks. Shaban *et al.* [29] use 1-shot model to make predictions with each support example and use logic OR operation to fuse individual predicted masks. Logic OR operation means that a position is predicted as foreground if any support example predicts it as foreground. 3) Average fusion for masks. Moreover, we also experiment with average operation to fuse individual 1-shot predicted confidence maps. We report the results of CANet with different fusion solutions in Table 5. Our attention mechanism

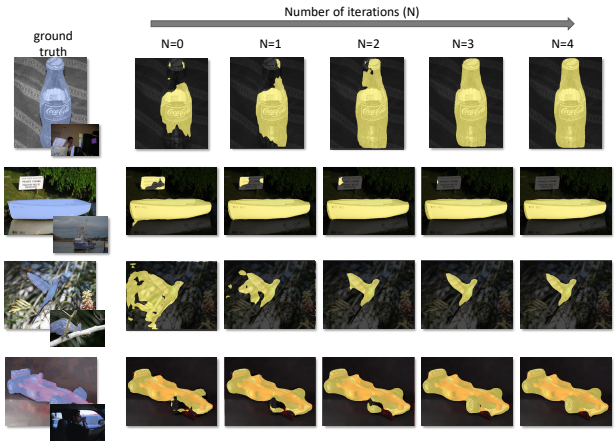

Figure 6 – Visualization of the iterative optimization process. The first column shows the query and support images with ground-truth masks annotated. The rest columns show our iterative optimization results.

Method	Result (meanIoU %)	Increment
1-shot baseline	54.0	0
Feature-Avg	55.0	1.0
Mask-Avg	54.5	0.5
Mask-OR	53.4	-0.6
Attention	**55.8**	**1.8**

Table 5 – Comparison of different 5-shot solutions. Our attention method performs the best and brings the most increment in the mean-IoU score over the 1-shot baseline.

performs the best and brings the most increment over 1-shot baseline. This indicates that a learned attention module can be more effective in fusing information from different support examples than non-learnable fusion methods in feature level or image level. Using logic OR operation to fuse predicted masks does not show improvement over the 1-shot result.

Multi-scale evaluation. We also experiment multi-scale evaluation as is commonly done in segmentation literature. Specifically, we re-scale the query image by [0.7, 1, 1.3] and average their predicted results. Multi-scale evaluation brings 1.4% and 1.3% meanIoU improvement in 1-shot and 5-shot settings, respectively.

5.2. COCO

COCO 2014 [18] is a challenging large-scale dataset, which contains 80 object categories. The original dataset contains 82,783 and 40,504 images for training and validation respectively. Directly experimenting on the original dataset is very demanding on time and computation. Instead, we choose a subset of the original dataset to evaluate our model and for further research on this topic. We choose 40 classes for training, 20 for validation and 20 for test, which contain 39,107 (train), 5,895 (validation) and 9,673 (test) samples, respectively. Training images are chosen from the COCO training set, while validation and test images are chosen from the COCO validation set.

Method	MS	Result (meanIoU %)
CANet-Init		42.2
CANet		46.3
CANet	✓	**49.9**

(a) 1-shot results on COCO dataset.

Method	MS	Result (meanIoU %)
Feature-Avg		48.9
Mask-Avg		49.2
Mask-OR		46.2
Attention		49.7
Attention	✓	**51.6**

(b) 5-shot results on COCO dataset.

Table 6 – MeanIoU results on COCO dataset. MS denotes multi-scale evaluation.

For the 1-shot task, we compare our network with the baseline model that does not employ additional iterative optimization (CANet-Init), and for the 5-shot task, we compare our attention mechanism with three non-learnable fusion methods described in Section 5.1.3. The result is shown in Table 6. In the 1-shot setting, our iterative optimization scheme brings 4.1% meanIoU improvement. Multi-scale evaluation shows extra 3.3% increase. In the 5-shot setting, our attention mechanism outperforms all non-learnable methods. Multi-scale evaluation obtains another 1.9% gain.

6. Conclusion

We have presented CANet, a novel class-agnostic segmentation network with few-shot learning. The dense comparison module exploits multiple levels of feature in CNNs to perform dense feature comparison and the iterative optimization module learns to iteratively refines the predicted results. Our attention mechanism for solving the k-shot problem turns out to be more effective than non-learnable methods. Comprehensive experiments show the effectiveness of our framework, and the performance significantly outperforms all previous work.

Acknowledgements

G. Lin's participation was partly supported by the National Research Foundation Singapore under its AI Singapore Programme [AISG-RP-2018-003] and a MOE Tier-1 research grant [RG126/17 (S)]. R. Yao's participation was supported by the National Natural Scientific Foundation of China (NSFC) under Grant No. 61772530. We would like to thank NVIDIA for GPU donation.

References

[1] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *NIPS*, pages 523–531, 2016. 2

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 1, 3

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 4, 5

[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*, pages 248–255, 2009. 1, 3

[5] N. Dong and E. Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018. 1, 6

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

[7] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. Springer, 2014. 6

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988, 2017. 1

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 3

[10] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick. Learning to segment every thing. In *CVPR*, 2018. 2

[11] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 2

[12] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011. 7

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[14] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, pages 3159–3167, 2016. 2, 5

[15] G. Lin, F. Liu, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for dense prediction. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 3

[16] G. Lin, A. Milan, C. Shen, and I. D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, volume 1, page 5, 2017. 1, 3

[17] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, pages 3194–3203, 2016. 3

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 8

[19] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 4

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1, 3

[21] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018. 2

[22] L. McIntosh, N. Maheswaranathan, D. Sussillo, and J. Shlens. Recurrent segmentation for variable computational budgets. In *CVPR*, pages 1648–1657, 2018. 5

[23] T. Munkhdalai and H. Yu. Meta networks. *arXiv preprint arXiv:1703.00837*, 2017. 2

[24] K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine. Conditional networks for few-shot semantic segmentation. In *ICLR Workshop*, 2018. 1, 3, 6, 7

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

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1

[28] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016. 2

[29] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017. 1, 3, 6, 7

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[31] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 1, 2

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 5

[33] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016. 3

[34] X. Wang, S. You, X. Li, and H. Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, pages 1354–1362, 2018. 5

[35] Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, pages 616–634, 2016. 2

[36] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016. 2

[37] F. S. 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. 1, 2

[38] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. 2, 3

[39] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014. 2, 3

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 4