Attention-based Multi-Context Guiding for Few-Shot Semantic Segmentation

Anonymous Author(s)

ABSTRACT

Few-shot learning is a nascent research topic, motivated by the fact that traditional deep learning methods require tremendous annotated data. The scarcity of annotated data becomes even more challenging in semantic segmentation since pixel-level annotation in segmentation task is more labor-intensive. To tackle this issue, we propose Attention-based Multi-Context Guiding (A-MCG) network, which consists of three branches: support branch, query branch, feature fusion branch. A key differentiator of A-MCG is the integration of multi-scale context features between support and query branches, enforcing a better guidance from the support set. In addition, we also adopt a spatial attention mechanism along the fusion branch to highlight context information from several scales, enhancing self-supervision in one-shot learning. To address the fusion problem in multi-shot learning, ConvLSTM is adopted to collaboratively integrate the sequential support features to elevate the final accuracy. Our architecture strikes a state-of-the-art result on unseen classes in a variant of PASCAL VOC12 dataset and performs favorably against the previous work with large gains of 1.1%, 1.4% measured in mIoU in the 1-shot and 5-shot setting respectively.

CCS CONCEPTS

• **Computing methodologies** → **Image segmentation**;

KEYWORDS

few-shot learning, image segmentation

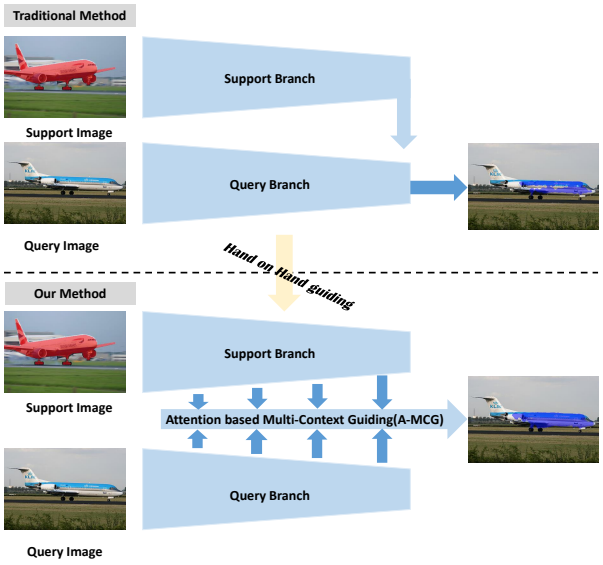

Figure 1: Our motivation. The support image mask is overlaid with ground truth in red, the query image is overlaid with ground truth in blue. The above is traditional method, the below is our method. Our A-MCG network can help the support branch fuse multi-context information to hand on hand guide the query branch.

1 INTRODUCTION

The state-of-the-art of image classification, detection, segmentation techniques have been greatly advanced by convolution neural network (CNN). Although CNN exhibits superior performances in a variety of tasks, it has the key problem of being data hungry. Typically gigantic data with annotations are required for achieving high performance. This issue becomes more severe in pixel-level annotations. In recent years, there emerges a new research thrust which learns new concepts from limited data, known as few-shot learning in the literature [20, 23, 25]. Though widely explored in tasks like image classification, few-shot learning is rarely considered in dense pixel prediction problems.

Most of existing methods in few-shot semantic segmentation are based on the framework as shown in the top panel of Fig. 1. Conceptually, the framework is comprised of support branch and query branch. Support branch provides discriminative support feature to assist the target segmentation, query branch is the feature extractor for target segmentation. This paradigm has following difficulties: (1). *Inefficient support feature utilization.* The feature of support branch is precious, as it determines final category the network will segment. However, most of the previous methods only consider the single output from the end of the network, which do not take full advantage of multi-context feature. (2). *Lack of attention.* As the amount of support and query data is often very small, optimization based on a large amount of data is impossible, some self-supervised methods such as attention mechanism should be introduced to make the network concentrate on our target class. (3). *Inconvenience in multi-shot learning.* Traditional fusion method for multi-shot semantic segmentation is logical or operation, this inflexible approach lacks in exploring the inner common feature between various support images.

To attack above problems in few-shot semantic segmentation, we propose an Attention-based Multi-Context Guiding network (A-MCG) shown in the bottom panel of Fig. 1. Our A-MCG tries to fuse small-to-large scale context information to globally guide the query branch to make the right segmentation decision. A multi-context feature will largely facilitate query branch segmentation based on multiple scales of support feature. In addition, we utilize the Residual Attention Module(RAM) [26] to carry out a self-supervised attention mechanism for further improvement of the segmentation. To deal with multi-shot learning, ConvLSTM [28] is incorporated for better fusing multi-shot support feature.

Our A-MCG network mainly makes the following contributions: (1). We firstly propose a Multi-Context Guiding structure to fuse

the small-to-large scale context features between support branch and query branch to globally guide the query branch segmentation. (2). We introduce a Residual Attention Module[26] in our MCG network to realize the attention mechanism in few-shot learning of segmentation. (3). We embed the ConvLSTM[28] module into the end of our network to better merge the feature map from support set in multi-shot semantic segmentation. (4). Compared with previous methods, our A-MCG reaches state-of-the-art 61.2%, 62.2% measured in mIoU in the 1-shot and 5-shot setting respectively.

Table 1: Problem Formulation Notations.

notation	meaning
I_S^i	i-th image in support set
$Y_S^i(l)$	i-th image-binary mask for class l
$S = \{(I_S^i, Y_S^i(l))\}_{i=1}^k$	support set
I_q	image in query set
L_{train}	train label set
L_{test}	test label set

2 RELATED WORK

Semantic Segmentation. During the early period, the CNN is only employed in the classification tasks, most of them[15, 24] are composed of convolution layers and fully connected layers. Fully Convolutional Network(FCN)[16] firstly applies CNN for the task of image semantic segmentation. FCN's key contribution is building "fully convolutional" network that takes an input of arbitrary size and produces correspondingly-sized output with efficient inference and learning.

In Deeplab[5], Dilated Convolutions are introduced as an alternative to CNN pooling layers in deep part to capture larger context without reducing the image resolution. A module named Atrous Spatial Pyramid Pooling (ASPP) is also included in Deeplab where parallel Dilated Convolution layers with different rates capture multi-scale information. Our method is also illuminated by the multi-context fusion pattern of ASPP and merges the multi-context information by borrowing the multi-scale context from support branch and query branch.

Attention Mechanism. In this paper, we mainly talk about two types of attention mechanism: (1). *Spatial Attention* such as Residual Attention Module(RAM)[26]. Inside each Attention Module, an Hourglass-like[18] bottom-up and top-down feedforward structure is used for generating attention map. (2). *Channel Attention* like SENet[13]. "Squeeze-and-Excitation"(SE) block is designed that adaptively recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels. We mainly attempt to employ these two types of attention mechanism into our MCG architecture.

Few-Shot Learning in Semantic Segmentation. The first work in few-shot semantic segmentation is OSLSM[21]. They proposed the basic paradigm in few-shot segmentation. Support branch and query branch are constructed by VGG[22] to supervise training. However, the structure is not fully convolutional, which leads to inefficient utilization of spatial information. Later co-FCN[19] turns both of the support and query branches into FCN architecture, but the exploration of multi-scale context is not thorough as our method.

On the other side, OSVOS[4] tries to solve the task of semi-supervised video object segmentation. OSVOS is also based on a fully-convolutional network architecture and transfers generic semantic information to the task of foreground segmentation. OSVOS shows the effectiveness of fine-tuning for video object segmentation, but fine-tuning for every test video is too time-consuming.

Convolutional Long Short-Term Memory. Long Short-Term Memory(LSTM) [12] is proposed as a special RNN structure to model long-range dependencies in various previous studies. However, LSTM is not suitable for handling spatiotemporal data because the input-to-state and state-to-state are all full connections thus no spatial information is encoded. To tackle this problem, ConvLSTM[28] is put forward by using a convolution operator in the state-to-state and input-to-state transitions.

In our network, ConvLSTM works as a memory unit to capture and integrate the previous support set feature for better multi-shot learning in semantic segmentation. This method gives us more interpretability and can better smoothen our k-shot learning result compared with traditional fusion method.

3 OUR METHOD

3.1 Problem Formulation

We follow the paradigm and notations in [21], which are detailed in Table 1. The target is to learn a model $f(I_q, S)$ that, when given a support set S and query image I_q, predicts a binary mask $\hat{M_q}$ for the semantic class l. The f function is parameterized by neural networks of support branch and query branch.

During training, the algorithm has access to a large set of image-mask pairs $D = \{(I^j, Y^j)\}_{j=1}^N$ where $Y^j \in L_{train}^{H \times W}$ is the binary mask for training image I^j. At testing, the query images are only annotated for new(unseen) semantic classes i.e. $L_{train} \cap L_{test} = \phi$, which leads us to divide the PASCAL VOC12 dataset like Table 2. This is the key difference between one-shot learning for segmentation and traditional segmentation, what we really care about is the segmentation performance on unseen data. Similar to the extension from one-shot learning to k-shot learning in classification task, k-shot learning can also be applied in semantic segmentation. In OSLSM[21], k-shot learning results are fused by a logical OR operation between the k binary masks.

3.2 Attention Mechanism Review

Residual Attention Module(RAM) is first proposed by Wang *et al.* [26] for image classification. We here review the structure of RAM in Fig. 3. The original RAM proceeds many ablation studies for its setting, we directly use the explored optimal structure. In our paper, we mainly explore the attention location in our MCG module in Sec. 4.

The RAM actually utilizes a two-scale(down sample 2 times, then up sample 2 times) Hourglass structure to construct a soft attention mask M(x). In the original ResNet[11], residual learning

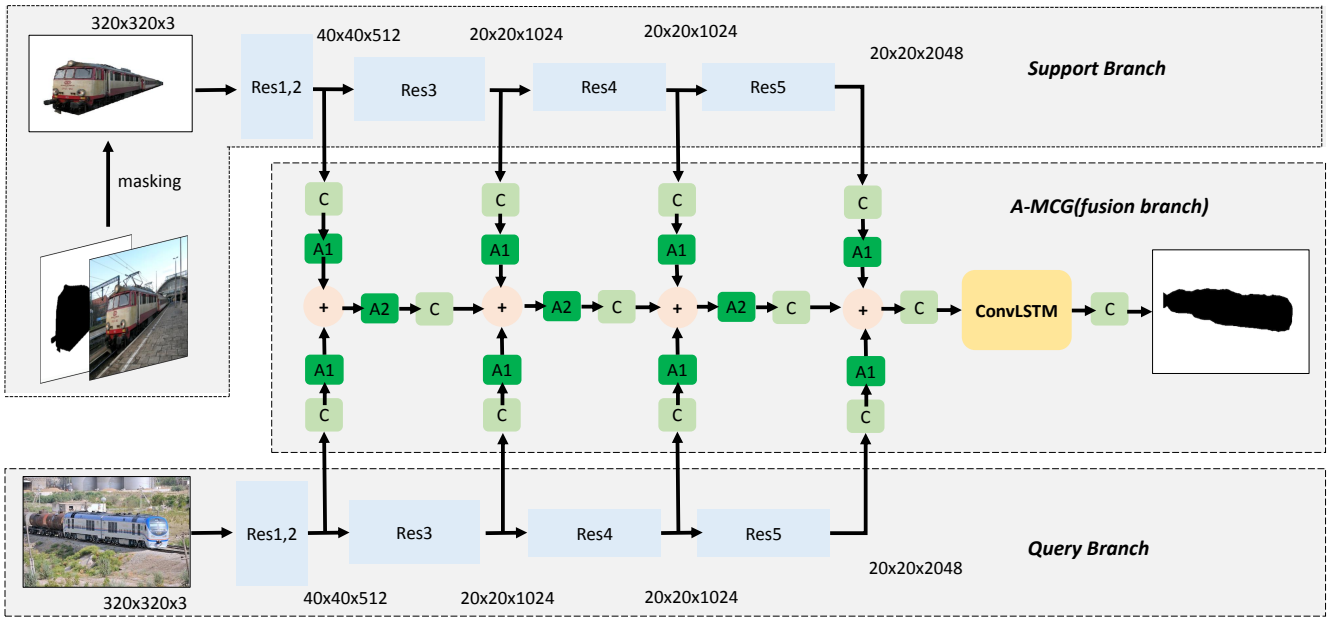

Figure 2: Attention-based Multi-Context Guiding Network Architecture for One-Shot Segmentation. It includes three parts: (1). support branch. (2). query branch. (3). A-MCG module. Res1, 2, 3, 4, 5 represent different block in ResNet. C is 1×1 convolution unit, stride=2 convolution is employed when the image size becomes smaller. We design two exclusive location settings of attention mechanism: A1, A2. The details of the attention architecture are illustrated in Fig. 3.

is formulated as:

$$H_{i,c} = x + F_{i,c}(x) \tag{1}$$

where $F_{i,c}(x)$ approximates the residual function, i ranges over all spatial positions and $c \in \{1, ..., C\}$ is the index of the channel.

In RAM, the attention module is modified as:

$$H_{i,c} = (1 + M_{i,c}(x)) * F_{i,c}(x) \tag{2}$$

M(x) ranges from [0,1] because of the sigmoid function. With M(x) approximating 0, H(x) will approximate original features F(x). The key of RAM lies in M(x), which works as feature selectors that enhance good features and suppress noises from trunk features. This characteristic of RAM is particularly important for few-shot learning cases.

Except that, we also explored the SE block[13], which is a typical channel attention structure. Our later ablation study in Sec. 4 shows that SE block fails to compete with RAM under the condition of same parameters.

3.3 Attention-based MCG

We propose an Attention-based Multi-Context Guiding network (A-MCG) illustrated in Fig. 2. Our A-MCG network is composed of three parts: (1). support branch. (2). query branch. (3). A-MCG(fusion) branch. The backbones of support branch, query branch are ResNet101. We elaborate our network as input image size 320×320, and their output feature map size is also marked in Fig. 2. Notably, the convolutions in Res-4, Res-lass5 blocks are equipped with Dilated Convolution[5] whose dilated rate=2. Therefore, the feature map size no long decreases after Res-3, but the receptive field continues to be enlarged due to Dilated Convolution.

Our A-MCG module tries to utilize multi-context feature from support branch to globally improve the query branch segmentation. We attempt two types of attention location pattern. In the following, we denote the b_s^i as feature maps after Res-i block in support branch, b_q^i as feature maps after Res-i block in query branch, C as naive convolution(without ReLU[17], BN), H as our attention function, F_i as the mixed features after Res-i block of support branch and query branch.

Therefore, we mainly come up with three variants.

(1). Multi-context guiding module:

$$F_{i+1} = C\{C(b_s^i) + C(b_q^i) + F_i\} \tag{3}$$

Multi-context information from both branches are mixed by convolution operation. Notably, convolution here doesn't include ReLU, BN.

(2). Multi-context guiding with separate attention:

$$F_{i+1} = C\{H(C(b_s^i)) + H(C(b_q^i)) + F_i\} \tag{4}$$

which corresponds to A1 in Fig. 2.

Attention mechanism is employed separately both in query branch and support branch.

(3). Multi-context guiding with share attention:

$$F_{i+1} = C\{H(C(b_s^i) + C(b_q^i) + F_i)\} \tag{5}$$

which corresponds to A2 in Fig. 2.

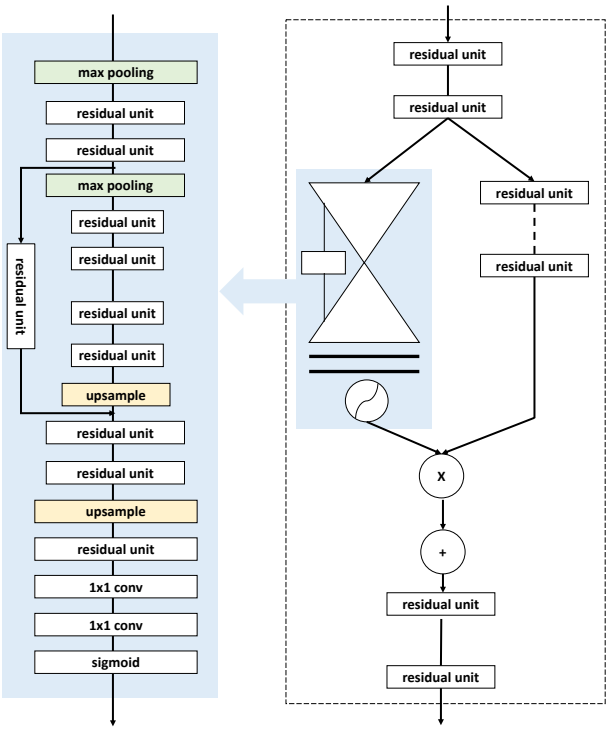

Figure 3: Residual Attention Module(RAM). Left side is an hourglass-like Soft Mask Branch, followed by a softmax operation. Right side is the whole structure of RAM. "residual unit" is a typical bottleneck structure which is detailed in ResNet [11].

Attention mechanism is applied after the fusion of query branch and support branch.

3.4 Convolutional LSTM for k-shot learning

In previous work, we mainly focus on the circumstance of 1-shot learning in semantic segmentation. How shall we deal with k-shot learning? In OSLSM[21], k-shot learning results are fused by a logical OR operation. However, this straightforward process is unexplainable and fails to utilize the inner relationship between sequential support images.

To better solve this multi-shot learning problem, we attempt to embed ConvLSTM[28] at the end of the fusion branch as illustrated in Fig. 2. At last, a 1×1 convolution will be appended to generate segmentation probability map.

ConvLSTM is firstly applied to the precipitation nowcasting task. The key idea of ConvLSTM is to implement all operations, including state-to-state and input-to-state transitions, with kernel-based convolutions. The inner feature of the sequential support image mask can be sustained by ConvLSTM. We here adopt a popular LSTM variant with "peephole connections"[8]. In detail, the three gating functions in ConvLSTM are calculated according to the equations below:

$$i_t = \sigma(\boldsymbol{W}_{x,i} \otimes \boldsymbol{X}_t + \boldsymbol{W}_{h,i} \otimes \boldsymbol{H}_{t-1} + \boldsymbol{W}_{c,i} \otimes \boldsymbol{C}_{t-1}) \quad (6)$$

$$o_t = \sigma(\boldsymbol{W}_{x,o} \otimes \boldsymbol{X}_t + \boldsymbol{W}_{h,o} \otimes \boldsymbol{H}_{t-1} + \boldsymbol{W}_{c,o} \otimes \boldsymbol{C}_t) \quad (7)$$

$$f_t = \sigma(\boldsymbol{W}_{x,f} \otimes \boldsymbol{X}_t + \boldsymbol{W}_{h,f} \otimes \boldsymbol{H}_{t-1} + \boldsymbol{W}_{c,f} \otimes \boldsymbol{C}_{t-1}) \quad (8)$$

where we let $\boldsymbol{X}_t, \boldsymbol{H}_t$ be the input/hidden state at time t respectively. $\otimes$ represents spatio-temporal convolution operator.

Investigating previous $\boldsymbol{H}$ and current $\boldsymbol{X}$, the recurrent model synthesizes a new proposal for the cell state, namely

$$\tilde{\boldsymbol{C}_t} = tanh(\boldsymbol{W}_{x,c} \otimes \boldsymbol{X}_t + \boldsymbol{W}_{h,c} \otimes \boldsymbol{H}_{t-1}) \quad (9)$$

The final cell state is obtained by linearly fusing the new proposal $\tilde{\boldsymbol{C}_t}$ and previous state $\boldsymbol{C}_{t-1}$:

$$\boldsymbol{C}_t = f_t \odot \boldsymbol{C}_{t-1} + i_t \odot \tilde{\boldsymbol{C}_t} \quad (10)$$

where $\odot$ denotes the Hadamard product. To continue the recurrent process, it also renders a filtered new H:

$$\boldsymbol{H}_t = o_t \odot tanh(\boldsymbol{C}_t) \quad (11)$$

In our previous structure, the network is trained on one-shot support set. Once ConvLSTM is imported into our framework, it will enable us to train with k-shot support set. For every batch(if batch size=1), one query image and k support image masks will be fed into our neural network.

We unroll this procedure in Fig. 4 for better understanding the k-shot fusion process. k-shot support image masks enter ConvLSTM in turn. ConvLSTM plays a critical role in summarizing the total features of the k-shot support image masks.

For better mixing the feature from the support set in k-shot learning, a function loss is designed as follows:

$$L = -\frac{1}{ks^2} \sum_{i=0}^{k} \sum_{m=0,n=0}^{s} Y_{m,n} log X_{m,n} \quad (12)$$

where Y is binary label, X means the neural network output probability, k is shot number, s represents image size.

This loss enforces our A-MCG module to function well on **every** support set image rather than only supervises the segmentation of single support image.

4 EXPERIMENTAL RESULT

4.1 Training details

We implement our code based on the tensorflow framework[1]. Specially, a scaffold framework named tensorpack[27] is used for quickly setting up our experiment. All our models are trained by Stochastic Gradient Descent(SGD)[3] solver with learning rate=1e-4, momentum=0.99 on one Nvidia Titan XP GPU. To fully fill GPU memory, we set the batch size 12. The weights of support branch and query branch are initialized with ImageNet[6] pre-trained weights. For the weight initialization of A-MCG module, Xavier initialization[9] is adopted. All the images in the support and query branch are resized to 320×320. No further augmentation is employed except the image resizing. For Batch Normalization(BN)[14], we employ current batch statistics at training and use the moving average statistics of BN during validation time.

We use the cross-entropy loss as the object function for training the network. The loss is summed up over all the pixels in a mini-batch.

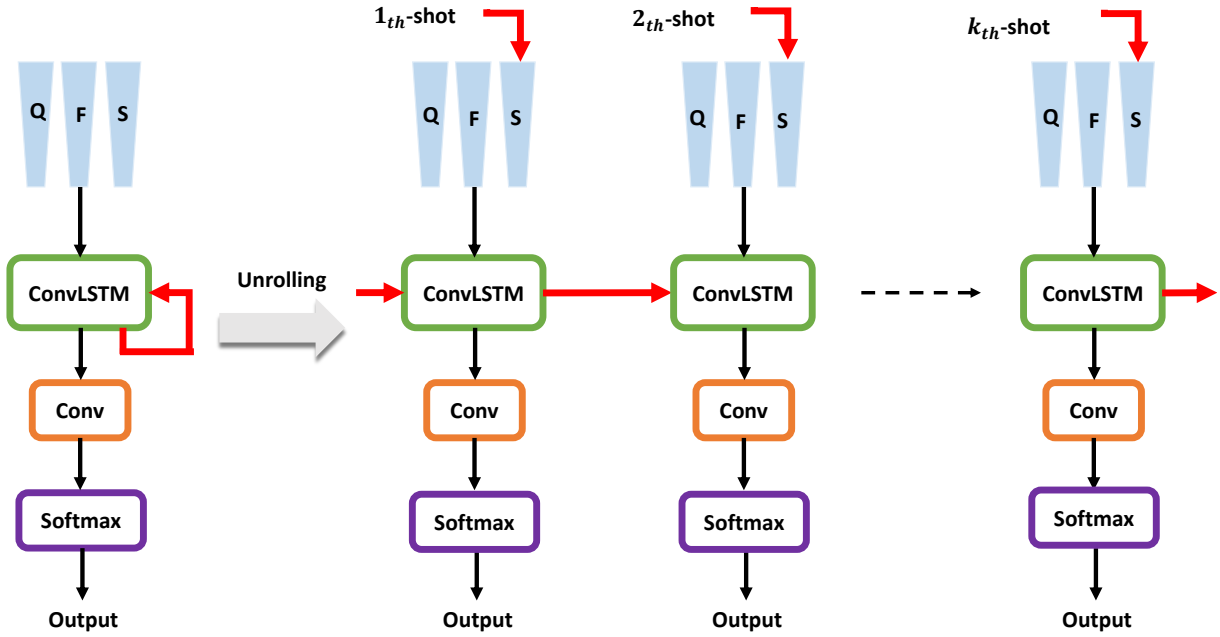

Figure 4: Unrolling view of ConvLSTM. Q, F, S represents query branch, fusion branch, support branch accordingly. Best viewed in color.

When we experiment with ConvLSTM for k-shot learning, we set k=5 by default. The max batch size can only be 6 because every time k support image masks will be fed into the support branch. Layer Normalization[2] is utilized in our ConvLSTM for speeding up convergence.

4.2 Dataset and Metric

Dataset: We utilize dataset PASCAL-5^i [21] to conduct our experiment. This dataset is originated from PASCAL VOC12[7] and extended annotations from SDS[10]. The set of 20 classes in PASCAL VOC12 is divided into four sub-datasets as indicated in Table 2. Three sub-datasets are used as the training label-set L_{train}, the left one sub-dataset is utilized for test label-set L_{test}.

The training set D_{train} is composed of all image-mask pairs from PASCAL VOC12 and SDS training sets that include at least one pixel in the segmentation mask from the label-set L_{train}. The masks in D_{train} are modified into binary masks by setting pixels whose semantic class are not in L_{train} as background class l_ϕ. The test set D_{test} is from PASCAL VOC12 and SDS validation sets, and the processing procedure for test set D_{test} is similar with training set D_{train}. Our evaluation mIoU is the average of 5 sub-dataset mIoUs. For a fair comparison with [21], we take the same random seed and sample N=1000 examples for testing each of our models.

Metric: To compare the quantitative performance of the different models, mean intersection over union(mIoU) over two classes is used for our benchmark evaluation. For binary segmentation in our work, we firstly calculate the 2×2 confusion matrix, then compute the according IoU_l as $\frac{tp_l}{tp_l+fp_l+fn_l}$. tp_l is the number of true positives for class l, fp_l is the number of false positives for class l and fn_l is the number of false negatives for class l. The final mIoU is its average over the set of classes.

Table 2: PASCAL-i^5 group information. The top table displays 4 groups of label and their semantic classes. The bottom table shows 4 sub-datasets and their training, validation components.

label set index	Semantic Classes
0	aeroplane, bicycle, bird, boat, bottle
1	bus, car, cat, chair, cow
2	diningtable, dog, horse, motorbike, person
3	potted plant, sheep, sofa, train, tv/monitor

sub-dataset	train label set	val label set
0	1,2,3	0
1	0,2,3	1
2	0,1,3	2
3	0,1,2	3

4.3 Ablation Study

Baseline. Our method is mostly compared with OSLSM[21] and co-FCN[19]. Both of them utilize the VGG[22] as basic model. Different from them, we adopt ResNet101[11] as our basic model, for ResNet101 owns much less parameter than VGG16, thus it is less

Table 3: Ablation Study for fusion width. The experiment is conducted on PASCAL-i^5 sub-dataset 0.

fusion width	1-shot	#params(M)
64	63.4	85.6
128	63.3	86.1
256	63.6	87.2
512	63.6	89.8
1024	63.2	96.1

Table 4: Ablation Study for multi-context pattern. The experiment is conducted on PASCAL-i^5 sub-dataset 0.

Method	1-shot	#params(M)
context-2345	63.6	87.2
context-45	63.2	86.7
context-5	61.2	86.1

prone to over-fitting. Besides, ResNet also enables larger batch size training in our architecture.

After removing the fully connected layers in the end, our ResNet101 baseline becomes a fully-convolutional structure. Support branch and query branch are fused by element-wise Add between the Res-5 output of them, followed by a naive convolution.

MCG: We explore several factors of our Multi-Context Guiding(MCG) architecture such as (1). *fusion width.* Fusion width is the channel number in the MCG branch, all the features in the support branch and query branch will be transformed into features with width of fusion width. Different settings of fusion width are stated in Table 3. (2). *multi-context pattern*, we try to explore what kind of context combination is better for the few-shot learning in Table 4. The number after "context" is the feature we will adopt for fusion. For example, context-45 means that only features from Res-4, Res-5 are used for fusion. For convenience, we only proceed the ablation study in the sub-dataset 0 in this part.

From Table 3, we can conclude that when the fusion width is too small such as 64 ,128, the one-shot mIoU is 0.2% lower than width=256. Meanwhile, larger fusion width like 1024 will make the mIoU worse due to over-fitting. Taking into consideration of the balance between mIoU and parameter cost, we choose fusion width=256 as our default setting. The latter ablation study will also adopt the default value.

As for the multi-context pattern in Table 4, more level context fusion often leads to a better result. Context-2345 outperforms context-5 nearly 2.4% mIoU. This shows that our multi-context guiding strategy works as our motivation. Multi-context information fusion from both support branch and query branch could efficiently "support" the query branch's segmentation.

Attention Mechanism. We set two variants about the attention module: (1). *Spatial Attention.* Residual Attention Module(RAM) is applied here as a representative method. (2). *Channel Attention.* SENet[13] is explored in our ablation study. At the same time, two attention location patterns "separate" and "share" are also explored.

Table 5: Ablation Study for Attention Mechanism. ChannelAttention means SENet Block, SpatialAttention means Residual Attention Block. "sep" denotes the support branch and query branch adopt separate attention. "share" represents the support branch and query branch share the same attention.

Method	1-shot	#params(M)
MCG	63.3	87.2
MCG-ChannelAttention-sep[1]	63.6	89.6
MCG-ChannelAttention-share[2]	61.7	89.8
MCG-SpatialAttention-sep	63.3	93.3
MCG-SpatialAttention-share	65.8	89.5

[1] For fair comparison with SpatialAttention method, we change the fusion width to 428 to make #param nearly the same.
[2] For fair comparison with SpatialAttention method, we change the fusion width to 480 to make #param nearly the same.

"sep" denotes the support branch and query branch adopt separate attention. "share" represents the support branch and query branch share the same attention.

As shown in Table 5. We can find that Spatial Attention works much better than Channel Attention under the circumstance of same parameters. We conclude that spatial information is more useful in dense pixel task like image segmentation, while channel information plays a more important role in classification task.

It can be obviously figured out that sharing the same attention is basically better than separate attention. We speculate that for support branch, the input has been already masked so that it does not need attention mechanism, while sharing attention mechanism will make the query branch pay more attention to the support branch's input mask.

On the other hand, we demonstrate some images' feature map visualization and feature map histogram in Fig. 5. From the visualization, we can qualitatively observe that the feature map becomes more focusing on the target segmentation objects. As for the feature map histogram, we can quantitatively discovery that the histogram peak move towards small value. We owe this observation to the fact that the Spatial Attention enhances good features and suppresses noises from trunk features.

ConvLSTM for k-shot learning. We mainly contrast two loss variants in ConvLSTM. (1). *1-loss ConvLSTM.* Only output from the last shot is supervised. (2). *5-loss ConvLSTM.* Every output from ConvLSTM is supervised. As indicated in Table 6, the ConvLSTM highly boosts our mIOU both in 1-shot and 5-shot learning. 1-shot result on 1-loss ConvLSTM is 0.7% lower than our baseline, we speculate that 1-loss ConvLSTM fails to fully supervise single-shot learning. Interestingly, both the 1-shot and 5-shot result on 5-loss LSTM outperform our baseline, which sufficiently validates our motivation.

Furthermore, we also conduct k-shot learning where k ranges from 1 to 10 in Fig. 6. k-loss ConvLSTM fully surpasses the traditional logical or method in all shot number range. When $k \leq 4$, the performance of 1-loss ConvLSTM is less than 5-loss ConvLSTM,

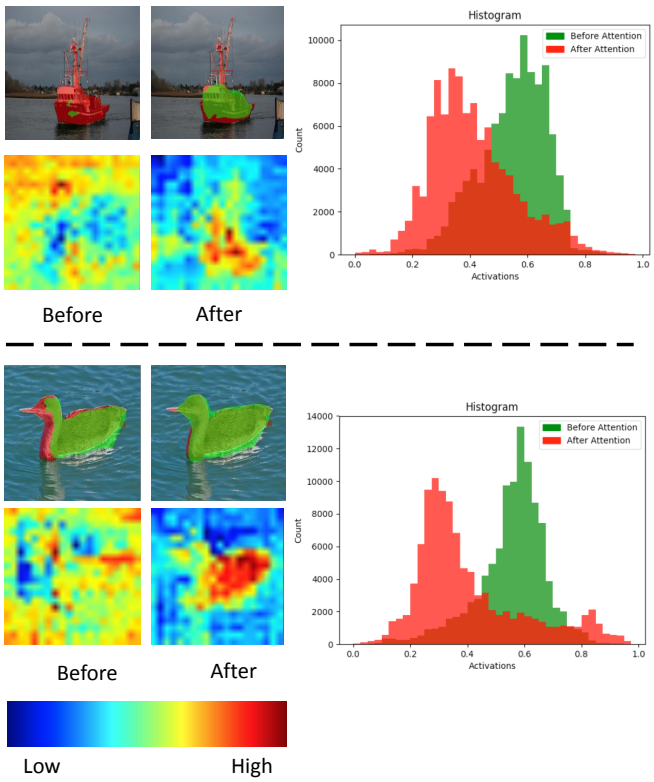

Figure 5: Attention Mechanism Visualization. Two images are demonstrated for the comparison between "before attention" and "after attention". The image is overlaid with predicted mask in green. The according histogram of the feature map is displayed in the right side. The feature map's activation value is normalized to 1. The support image is ignored. Best viewed in color.

Table 6: Ablation Study for loss function in ConvLSTM. Baseline is our A-MCG module, we mainly compare the difference between 1-loss ConvLSTM and 5-loss LSTM. The experiment is conducted on PASCAL-i^5 sub-dataset 0.

Method	1-shot	5-shot	#params(M)
baseline	65.8	66.2	89.5
1-loss ConvLSTM	65.1	67.5	90.8
5-loss ConvLSTM	66.1	67.9	90.8

while partially larger than our baseline. This proves that our 5-loss ConvLSTM better integrates multi-shot support features than traditional method.

Final result. As shown in Table 7, our A-MCG architecture could outperform nearly 61.2% in 1-shot mIoU, 62.2% in 5-shot mIoU. Based on our baseline, we continue applying the MCG, attention mechanism, ConvLSTM, reach a new state-of-the-art result on PASCAL-i^5 dataset in the end.

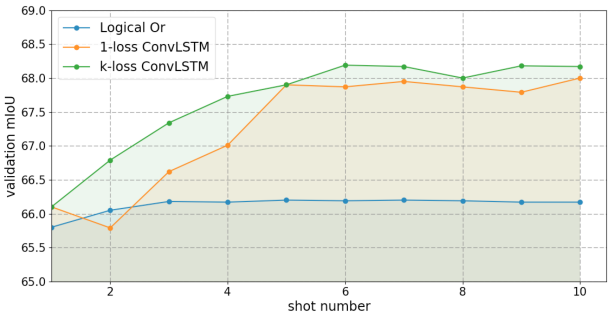

Figure 6: The relationship between shot number and validation mIoU, we mainly compare among three multi-shot learning fusion strategies: (1). Logical Or. (2). 1-loss ConvLSTM. (3). k-loss ConvLSTM(k=5 in our experiment). The experiment is conducted on PASCAL-i^5 sub-dataset 0.

Table 7: Result on PASCAL-i^5 Dataset. All results are computed by taking the average of the 5 sub-datasets in PASCAL-i^5. The 5-shot result is obtained by logic or fusion except the method with ConvLSTM.

Method	1-shot	5-shot	#params(M)
OSLSM[21]	40.8	43.9	276.7
co-FCN(Multi-class)[19]	50.9	50.9	–
co-FCN(Overall)[19]	60.1	60.8	–
Baseline	53.0	54.8	85.1
MCG	55.3	56.5	87.2
A-MCG	57.3	57.8	89.5
A-MCG-ConvLSTM	**61.2**	**62.2**	90.8

Some visualization result can be seen in Fig. 7. We also explore the effect of increasing the support set size as shown in Fig. 8.

5 CONCLUSION

We propose an Attention-based Multi-Context Guiding network (A-MCG) which incorporates multi-level concentrated context. The benefits of our network are in three folds: (1). The shallow part of our network generates low-level semantic features, meanwhile deep part of our network captures high-level semantic features. Context features in equal level are fused by our MCG module, which highly facilitates the support branch to globally "support" the query branch. (2). Spatial Attention is employed along with the whole MCG branch, which makes our network focus on different scales of context information. (3). The import of ConvLSTM enables the network better integrate the feature from support set in multi-shot semantic segmentation. The performance of our model surpasses state-of-the-art methods in few-shot semantic segmentation. In the future, we will exploit few-shot learning in multi-class segmentation at one time.

Figure 7: Some qualitative results of our method for 1-shot learning. Inside each tile, we have the support set at the top and the query image at the bottom. The support is overlaid with the ground truth in red and the query is overlaid with our predicted mask in blue. Best viewed in color.

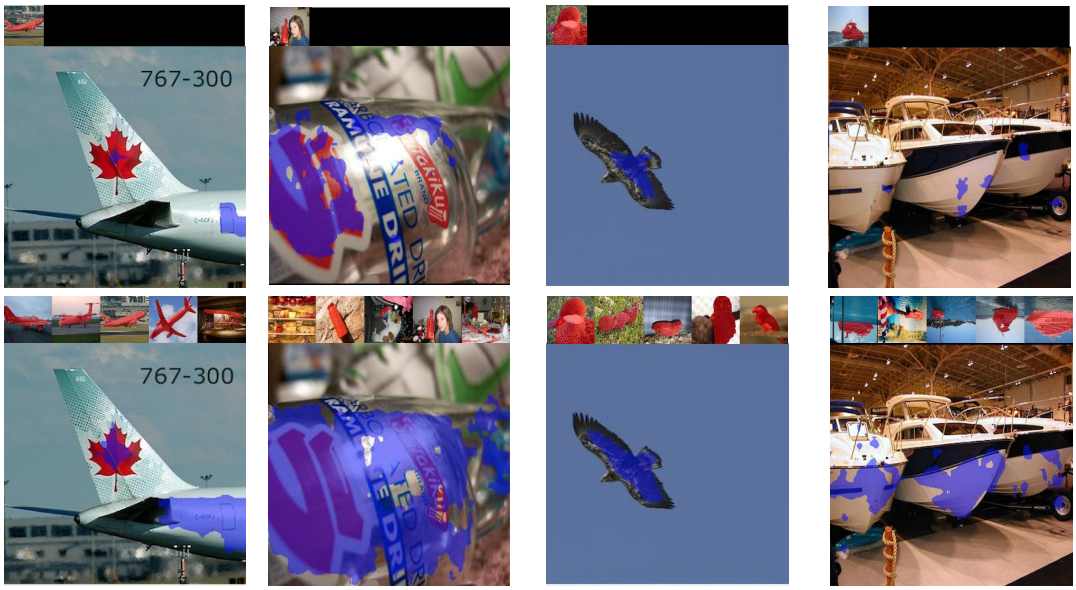

Figure 8: Effect of increasing the size of the support set. Results of 1-shot and 5-shot learning on the same query image are in the first and second rows respectively. The prediction is in blue. Best viewed in color.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A System for Large-Scale Machine Learning.. In *OSDI*, Vol. 16. 265–283.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[3] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. 2017. One-Shot Video Object Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (2018), 834–848.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (2010), 303–338.

[8] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. 2002. Learning Precise Timing with LSTM Recurrent Networks. *Journal of Machine Learning Research* 3, Aug (2002), 115–143.

[9] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*. 249–256.

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

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[13] Jie Hu, Li Shen, and Gang Sun. 2017. Squeeze-and-Excitation Networks. *arXiv preprint arXiv:1709.01507* (2017).

[14] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.

[17] Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *International Conference on Machine Learning*.

[18] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *European Conference on Computer Vision*. Springer, 483–499.

[19] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. 2018. Conditional Networks for Few-Shot Semantic Segmentation. In *International Conference on Learning Representations Workshop Papers*.

[20] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a Model for Few-Shot Learning. In *International Conference on Learning Representations*.

[21] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. 2017. One-Shot Learning for Semantic Segmentation. In *British Machine Vision Conference*.

[22] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[23] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. 2015. Going Deeper with Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[25] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*.

[26] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. 2017. Residual Attention Network for Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[27] Yuxin Wu et al. 2016. Tensorpack. https://github.com/tensorpack/. (2016).

[28] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*.