Dynamic Extension Nets for Few-shot Semantic Segmentation

Lizhao Liu*[†]
South China University of Technology
Guangzhou, China
selizhaoliu@mail.scut.edu.cn

Junyi Cao*
South China University of Technology
Guangzhou, China
sejaycao@mail.scut.edu.cn

Minqian Liu*
South China University of Technology
Guangzhou, China
csmqliu@mail.scut.edu.cn

Yong Guo*
South China University of Technology
Guangzhou, China
guoyongcs@gmail.com

Qi Chen*
South China University of Technology
Guangzhou, China
chenqi.china@outlook.com

Mingkui Tan[‡]
South China University of Technology
Guangzhou, China
mingkuitan@scut.edu.cn

ABSTRACT

Semantic segmentation requires a large amount of densely annotated data for training and may generalize poorly to novel categories. In real-world applications, we have an urgent need for few-shot semantic segmentation which aims to empower a model to handle unseen object categories with limited data. This task is non-trivial due to several challenges. First, it is difficult to extract the class-relevant information to handle the novel class as only a few samples are available. Second, since the image content can be very complex, the novel class information may be suppressed by the base categories due to limited data. Third, one may easily learn promising base classifiers based on a large amount of training data, but it is non-trivial to exploit the knowledge to train the novel classifiers. More critically, once a novel classifier is built, the output probability space will change. How to maintain the base classifiers and dynamically include the novel classifiers remains an open question. To address the above issues, we propose a Dynamic Extension Network (DENet) in which we dynamically construct and maintain a classifier for the novel class by leveraging the knowledge from the base classes and the information from novel data. More importantly, to overcome the information suppression issue, we design a Guided Attention Module (GAM), which can be plugged into any framework to help learn class-relevant features. Last, rather than directly train the model with limited data, we propose a dynamic extension training algorithm to predict the weights of novel classifiers, which is able to exploit the knowledge of base classifiers by dynamically extending classes during training. The extensive experiments show that our proposed method achieves state-of-the-art performance on the PASCAL-5^i and COCO-20^i datasets. The source code is available at https://github.com/lizhaoliu-Lec/DENet.

CCS CONCEPTS

• **Computing methodologies** → **Image segmentation**; **Scene understanding**; **Neural networks**.

KEYWORDS

Few-shot Semantic Segmentation; Guided Attention; Dynamic Extension Network

ACM Reference Format:
Lizhao Liu, Junyi Cao, Minqian Liu, Yong Guo, Qi Chen, and Mingkui Tan. 2020. Dynamic Extension Nets for Few-shot Semantic Segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413915

1 INTRODUCTION

Semantic segmentation aims to assign a unique label to each pixel of an image, which plays an essential role in many vision-based applications, such as autonomous driving [38, 45], video surveillance [8, 41] and bio-medical image diagnosis [27]. However, it requires a considerable amount of densely annotated training data for each class (called base class) [25] to obtain a promising model. More critically, the model may generalize poorly to novel categories, for which the amount of pixel-level annotated data may be limited or very expensive to obtain [1, 47].

To address the novel class issue in real-world applications, the few-shot semantic segmentation, which seeks to handle unseen object categories with limited data, has gained great attention in the computer vision community [11–14, 28, 32, 33]. Thanks to the representation power of deep neural networks, the last decade has witnessed large progress in few-shot semantic segmentation [24, 26, 32, 33, 37, 40, 42–44]. In general, these methods can be divided into two categories: binary segmentation methods and multi-class segmentation methods. The binary segmentation methods [24, 26, 32, 42–44] focus on finding a novel class in the query image given the support set of the corresponding class. These methods cannot directly handle multiple novel classes. Multi-class segmentation methods [37, 40] thus have been proposed and they seek to estimate a classifier for each novel class for the pixel-level classification. Once a novel classifier is built, the output probability space will change. How to maintain the base classifiers and dynamically include the novel classifiers remains an open question. Moreover, existing methods may still suffer from two limitations.

*Authors contributed equally.
[†]Also with Pazhou Laboratory, Guangzhou, China
[‡]Corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413915

First, it is non-trivial to extract class-relevant features from the few labeled samples of the novel classes. In fact, images may contain the pixels of both the base and novel classes, and thus the image content can be very complex. Due to the limited data, the novel class information may also be suppressed or interfered by base classes, which hampers the recognition of novel classes. Therefore, how to effectively extract features from the few samples of novel classes and alleviate the interference incurred by base classes is an important problem.

Second, how to exploit the learned knowledge from the base classifier to build a promising classifier for a novel class is very challenging. In fact, one can easily learn promising base classifiers based on a large amount of training data, but it is hard to exploit the learned knowledge to train the novel classifiers. Moreover, when we build a new classifier, the output probability space would change and model performance may deteriorate. To address this issue, a lot of efforts have been made to learn a weight generator to predict the weights of classifiers [13, 34]. In this way, by combining both the learned base classifiers and the predicted novel classifiers, these methods are able to recognize both the base and novel classes simultaneously. However, the base classifiers have to be trained before the training of weight generator. As a result, the model cannot be trained in an end-to-end manner, which may hamper the performance. Thus, how to design an end-to-end learning method to effectively exploit the knowledge learned by base classifiers and guide the learning of novel classifiers remains a question.

In this paper, we propose a novel Dynamic Extension Network (DENet) for few-shot semantic segmentation. To address the first limitation, we propose a Guided Attention Module (GAM), which uses the attention mechanism to guide the model to focus on the class-relevant content of the images. In this way, the model is able to capture more discriminative features from the given images and then has better generalization ability for the novel classes. To address the second limitation, we propose a dynamic extension training method that dynamically estimates and incorporates novel classifiers. In this way, we are able to train the model to recognize both the base and novel classes in an end-to-end manner. Experiments on the PASCAL-5^i and COCO-20^i datasets demonstrate the superiority of the proposed method over existing methods.

The contributions of this paper are summarized as follows.

- We propose an attention-based few-shot semantic segmentation method that exploits a Guided Attention Module (GAM) to estimate the weights of novel classifiers. To extract class-relevant features, GAM uses the support segmentation mask to attend the support image and thus effectively alleviates the interference from other classes (*e.g.*, base classes).
- We propose a dynamic extension training method that dynamically estimates and incorporates novel classifiers. In this way, we are able to train both the base and novel classifiers in an end-to-end manner.
- Extensive experiments on PASCAL-5^i and COCO-20^i demonstrate the superiority of the proposed method over existing methods for both 1-shot and 5-shot semantic segmentation.

2 RELATED WORK

2.1 Semantic Segmentation

Semantic segmentation has achieved considerable progress in recent years [2–4, 19, 25, 36, 46]. Long et al. [25] first employ deep CNNs to segmentation and propose Fully Convolutional Network (FCN) to considerably improve the segmentation result. To cope with complex scenes, Zhao et al. [46] and Hou et al. [19] adopt pooling strategies to leverage the global context. DeepLabs [2–4] combine the atrous convolution with spatial pyramid pooling [46] and propose an Atrous Spatial Pyramid Pooling (ASPP) module. To further improve the performance, many methods seek to use neural architecture search techniques [15, 16, 48] to automatically find good semantic segmentation models [23]. Compared with these methods that require abundant labeled data, our approach is able to recognize novel classes with only a few labeled samples without any retraining or fine-tuning processes.

2.2 Few-shot Learning

Few-shot learning [9] requires only a few samples to learn a new concept. The methods that are most related to our work fall into two groups: meta-learning methods [10, 31] and metric learning methods [13, 20, 28, 35, 39]. Meta-learning methods exploit different tasks [10, 31] sampled from a large amount of base class data to simulate the inference scenarios. In this way, they can yield promising generalization performance for novel samples. Metric learning approaches [20, 35, 39] attempt to learn the feature representations where the features of the same object are closer than different ones. Qi et al. [28] explore the connections between metric learning and softmax layer, and propose weight imprinting to extend the classifier. Gidaris and Komodakis [13] propose a two-stage training algorithm to acquire a base classifier and a weight generator to dynamically extend the model. Different from this method, our dynamic extension training algorithm optimizes both the base classifiers and the weight estimator in an end-to-end manner.

2.3 Few-shot Semantic Segmentation

Most efforts have been made to improve the performance of few-shot semantic segmentation, including binary segmentation methods [24, 26, 42–44] and multi-class segmentation methods [37, 40]. Zhang et al. [42] propose a graph attention unit that captures the correspondence between the support and the query to aid the query image segmentation. Liu et al. [24] propose a cross-reference network which uses a cross-reference mechanism to generate the reinforced feature representations by comparing the co-occurrent features in the support and the query. Wang et al. [40] obtain the prototypes for different classes from the support set and use them to recognize the corresponding classes in the query image. Tian et al. [37] propose to acquire the weights of novel classes with a closed-form solution for better generalization. Moreover, Rakelly et al. [30] consider few-shot segmentation under few pixel supervision setting. However, these methods still suffer from extracting class-related features from scarce labeled data, and it is difficult for them to use the base class knowledge to help learn the classifiers of novel classes.

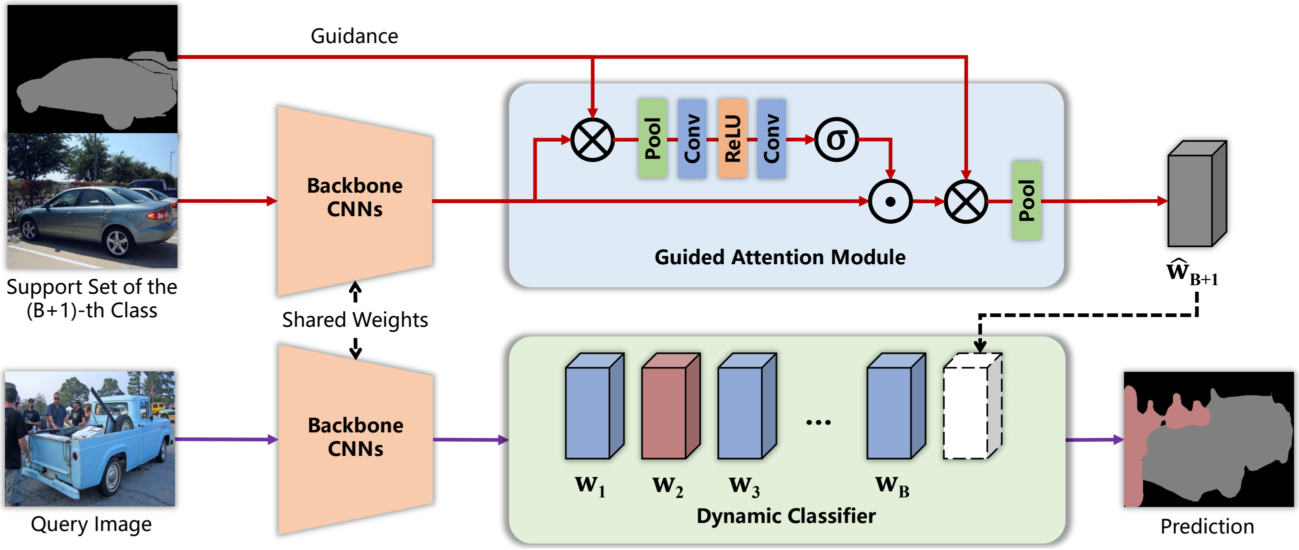

Figure 1: Overview of Dynamic Extension Network. Given B base classes, we term the novel class as the $(B+1)$-th class. Given the support set of the $(B+1)$-th class, we first extract the feature maps using Backbone CNNs and then feed them into a Guided Attention Module (GAM) to estimate the weight $\hat{\mathbf{w}}_{B+1}$ of the $(B+1)$-th class with the attention mechanism. In this sense, GAM is able to extract class-relevant information. The weights of B base classes are preserved by Dynamic Classifier (DC) after trained on base classes. Last, we add the estimated weight $\hat{\mathbf{w}}_{B+1}$ into DC to distinguish both the base and novel classes.

3 PROBLEM DEFINITION

Notation. Throughout the paper, we use the following notations. We use C^b and C^n to denote the set of base classes and the set of novel classes, respectively. Note that there is no overlap between these two sets, *i.e.*, $C^b \cap C^n = \varnothing$.

Few-shot semantic segmentation aims to predict the segmentation mask for the novel (*i.e.*, previously unseen) classes with very few labeled data. Formally, a few-shot learning problem can be formulated as an N-way K-shot recognition problem. Specifically, given N novel classes, there are only K samples for each class and we take them as the support set $\mathcal{S}$ to predict the segmentation mask of a query image $\mathbf{I}^q$ in the query set $\mathcal{Q}$. The support set is a set of image-annotation pairs, *i.e.*, $\mathcal{S} = \{(\mathbf{I}_{c,k}, \mathbf{M}_{c,k})\}$, where $\mathbf{I}_{c,k}$ and $\mathbf{M}_{c,k}$ denote the image and the segmentation mask of the k-th sample in the c-th class, with $k = 1, 2, \cdots, K$ and $c = 1, 2, \cdots, N$. The query set $\mathcal{Q}$ includes the segmentation mask during training and excludes it during testing. We train the model with the samples from C^b and evaluate it on the samples from C^n. Formally, a few-shot semantic segmentation model can be formulated by

$$\hat{\mathbf{M}}^q = f\left(\mathcal{S}, \mathbf{I}^q\right), \tag{1}$$

where $\hat{\mathbf{M}}^q$ is the predicted segmentation probability map of $\mathbf{I}^q$ and f denotes the few-shot segmentation model.

This problem, however, is very challenging to solve. **First**, it is non-trivial to effectively extract class-relevant features from only a few novel samples. Meanwhile, since the image content can be very complex, the novel class information may be suppressed by the base categories. **Second**, when learning the novel classifiers, it is difficult to exploit the knowledge of base classifiers that are trained on a large amount of data.

4 DYNAMIC EXTENSION NETWORK

In this paper, we propose a Dynamic Extension Network (DENet) that accurately estimates the weights of novel classifiers and dynamically extends the few-shot semantic segmentation model. We show the overall scheme of the proposed method in Figure 1.

To effectively extract the information from the few labeled data, we propose a Guided Attention Module (GAM) that uses the support segmentation mask to attend the support image. Based on the attended features, we are able to accurately estimate the weights for a specific novel class. To exploit the learned knowledge from base classifiers, we propose a dynamic extension training method that jointly trains both the base classifiers and the estimated novel classifiers. In this sense, we are able to train DENet in an end-to-end manner. More critically, based on the base and novel classifiers, we are able to recognize both the base and novel classes during inference. The training algorithm is shown in Algorithm 1.

4.1 Weight Estimation with Guided Attention

Due to the limited data of novel classes, the novel class information may be suppressed/interfered by the base classes and thus the novel classifiers become hard to learn. To address this issue, an intuitive way is to extract the class-relevant information to alleviate the interference incurred by the base classes. To this end, we propose a Guided Attention Module (GAM) as the attention-based weight estimator $G(\cdot, \cdot)$ that takes both the support image and segmentation mask as inputs to estimate the weights of novel classifiers.

We first detail the weight estimation process. Given the support image $\mathbf{I}_c^s$ of the c-th novel class, we first use a CNN model to extract features $\mathbf{F}_c^s$. Then, we take both the features $\mathbf{F}_c^s$ and the corresponding segmentation mask $\mathbf{M}_c^s$ to estimate the weights $\hat{\mathbf{w}}_c$

Algorithm 1: Dynamic extension training algorithm.

```
Input: Support set $\mathcal{S}$, query set $\mathcal{Q}$, learning rate $\eta$.
Initialize the model parameters $\theta$ for DENet.
for each iteration do
    Obtain $\left(\mathbf{I}_c^s, \mathbf{M}_c^s\right)$ from $\mathcal{S}$ and extract features $\mathbf{F}_c^s$.
    Estimate the weights $\hat{\mathbf{w}}_c$ by Eqn. (2).
    Record the $c$-th classifier $\mathbf{w}_c' \leftarrow \mathbf{w}_c$.
    Replace the $c$-th classifier $\mathbf{w}_c \leftarrow \hat{\mathbf{w}}_c$.
    Obtain $\left(\mathbf{I}^q, \mathbf{M}^q\right)$ from $\mathcal{Q}$ and extract features $\mathbf{F}^q$.
    Compute the probability map $\hat{\mathbf{M}}^q$ by Eqn. (7).
    Compute the loss $\mathcal{L}$ by Eqn. (8).
    Update the model by $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$.
    Restore the $c$-th classifier $\mathbf{w}_c \leftarrow \mathbf{w}_c'$.
end
```

Algorithm 2: Support-based Prediction method.

```
Input: Support set $\mathcal{S}$, query set $\mathcal{Q}$.
// Construct the dynamic classifier based on $\mathcal{S}$
for $c = 1, ..., N$ do
    for $k = 1, ..., K$ do
        Obtain $(\mathbf{I}_{c,k}^s, \mathbf{M}_{c,k}^s)$ from $\mathcal{S}_i$ and extract features $\mathbf{F}_{c,k}^s$.
    end
    Estimate the weights $\hat{\mathbf{w}}_c$ by Eqn. (6).
    Extend the dynamic classifier by Eqn. (10).
end
// Predict the segmentation masks for the data in $\mathcal{Q}$
for $i = 1, \cdots, |\mathcal{Q}|$ do
    Obtain $\mathbf{I}_i^q$ from $\mathcal{Q}$ and extract features $\mathbf{F}_i^q$.
    Compute the probability map $\hat{\mathbf{M}}_i^q$ by Eqn. (7).
end
```

of the considered class by

$$\hat{\mathbf{w}}_c = G\left(\mathbf{F}_c^s, \mathbf{M}_c^s\right). \tag{2}$$

As depicted in Figure 1, the operation of GAM can be divided into three steps. **First**, given the support feature map $\mathbf{F}_c^s \in \mathbb{R}^{d \times h \times w}$ and corresponding support mask $\mathbf{M}_c^s \in \mathbb{R}^{1 \times h \times w}$, our model fuses $\mathbf{F}_c^s$ with $\mathbf{M}_c^s$ by an element-wise multiplication. In this sense, we obtain the attention vector $\mathbf{g} \in \mathbb{R}^{d \times 1 \times 1}$ using the following formula:

$$\mathbf{g} = \sigma(h(\mathrm{Pool}(\mathbf{F}_c^s \otimes \mathbf{M}_c^s))), \tag{3}$$

where $\otimes$ denotes the element-wise multiplication, h represents the convolutional network and σ denotes the activation function. **Second**, to allow our model to focus on class-relevant information of the features, we suppress the irrelevant features by using the attention vector g. The function can be defined as

$$\mathbf{F}_c^{s\prime} = \mathbf{F}_c^s \odot \mathbf{g}, \tag{4}$$

where $\odot$ is the channel-wise multiplication. **Finally**, we estimate the weights $\hat{\mathbf{w}}_c \in \mathbb{R}^{d \times 1 \times 1}$ for class c by

$$\hat{\mathbf{w}}_c = \mathrm{Pool}(\mathbf{F}_c^{s\prime} \otimes \mathbf{M}_c^s), \tag{5}$$

where $\otimes$ is the element-wise multiplication.

***K*-shot Setting.** For the K-shot setting where more than one sample is available for the new category, we process K support samples of the given category independently to acquire K estimated weights. Then, we obtain the final estimated weight for the new class by taking the average of them. Note that previous methods [42, 43] may introduce extra parameters or operations for the K-shot setting. Different from these methods, we only take the average of the weights estimated from G over K support images

$$\hat{\mathbf{w}}_c = \frac{1}{K} \sum_{k=1}^{K} G(\mathbf{F}_{c,k}^s, \mathbf{M}_{c,k}^s). \tag{6}$$

4.2 Dynamic Extension Training Algorithm

In this paper, we propose a dynamic extension training method that jointly trains the base classifiers and the estimated novel classifiers in an end-to-end manner. In this sense, the proposed method is able to effectively exploit the knowledge learned from the base classifiers. We show the training method in Algorithm 1.

To learn a good DENet, we build the training method to mimic the real-world scenario that there is a large amount of training data of base classes and few data of novel classes. Specifically, at each iteration, we randomly pick a "fake" novel category from the base categories. For convenience, we use $|C^b|$ to denote the number of base classes and $|C^n|$ to denote the number of novel classes. It is worth noting that the novel classes are the "real" novel ones that do not come from the set of base classes during inference. In this way, there are $(|C^b|+|C^n|)$ classes in total. Assume that the c-th class is chosen to be the "fake" novel one, we first estimate the weights $\hat{\mathbf{w}}_c$ and use it to replace the original weights by $\mathbf{w}_c \leftarrow \hat{\mathbf{w}}_c$. Based on the extracted features $\mathbf{F}^q$ of a query image $\mathbf{I}^q$, the probability of a pixel that is located at the position (i, j) and belongs to the c-th class can be computed by

$$(\hat{\mathbf{M}}^q)_c^{(i,j)} = \frac{\exp\left((\mathbf{F}^q)^{(i,j)} \cdot \mathbf{w}_c\right)}{\sum_{l \in C^b \cup C^n} \exp\left((\mathbf{F}^q)^{(i,j)} \cdot \mathbf{w}_l\right)}, \ c = 1, \cdots, |C^b|+|C^n|. \tag{7}$$

Based on the predicted probability of each pixel, we minimize the weighted cross-entropy loss over all pixels of an image to jointly train the base classifiers and novel classifiers. Due to the imbalance between the base and novel classes, we introduce a weighting factor λ_c to reflect the importance of the pixels that belong to the novel classes. Note that we will discuss the effect of different λ in Section 6.2. Thus, the training loss of DENet can be written as

$$\mathcal{L} = -\sum_{i,j} \sum_{c \in C^b \cup C^n} \lambda_c \cdot \mathbb{1}\left[(\mathbf{M}^q)^{(i,j)} = c\right] \log(\hat{\mathbf{M}}^q)_c^{(i,j)}. \tag{8}$$

Here, $\mathbb{1}[A]$ is an indicator function, where $\mathbb{1}[A] = 1$ if A is true and $\mathbb{1}[A] = 0$ if A is false. The weighting factor λ_c can be computed by

$$\lambda_c = \begin{cases} 1 & c \in C^b \\ \lambda, & c \in C^n \end{cases}. \tag{9}$$

Note that we record the replaced classifier before replacing its weights. In each iteration, after updating the parameters, we restore the c-th classifier with the recorded one to continue to learn its weights.

4.3 Inference Method

We consider two kinds of inference methods of DENet. First, given a support set and a query image, we typically apply a **Support-based Prediction** method to extend the classifier with the support set and predict the segmentation mask of the query image. Second, when the query image belongs to the previously encountered novel class, we are able to further derive a **Support-free Prediction** method to directly predict the query segmentation mask without the need for any support set.

Support-based Prediction. The details are shown in Algorithm 2. Given a support set $\mathcal{S}$, DENet first extracts the corresponding feature $\mathbf{F}^s_{c,k}$ for each image and then estimates the weight $\hat{\mathbf{w}}_c$ of novel classifier c based on the extracted features. In this way, we obtain a dynamic classifier that contains the weights of novel classifier c. After that, we incorporate the estimated classifier into the set of originally existing classifiers $\mathcal{W}$:

$$\mathcal{W} \leftarrow \mathcal{W} \cup \{\hat{\mathbf{w}}_c\}. \tag{10}$$

In this sense, we are able to dynamically extend the classifiers and thus recognize both the base and novel classes during inference.

Support-free Prediction. DENet takes the support set (even with a single sample) as inputs to estimate the weights of a novel classifier. Once obtaining the novel classifier, we no longer need to predict a new classifier and we just use the previously estimated one in the following predictions for different query images belonging to this class. Since the prediction does not rely on the support set after a very first prediction, we term this method support-free prediction. The proposed support-free prediction method does not estimate the classifier. Thus, the inference can be much more efficient than support-based methods. We will demonstrate this in Section 6.4.

5 EXPERIMENTS

5.1 Datasets and Evaluation Metrics

Datasets. We evaluate the proposed method on two benchmark datasets, namely PASCAL-5^i [32] and COCO-20^i [26]. The PASCAL-5^i dataset is constructed from PASCAL VOC 2012 [7] that is extended by SBD [17] annotations. There are 20 categories in PASCAL VOC in total. These categories are evenly split into 4 folds and each fold contains 5 classes. We train the model on 3 folds and test the model on 1 fold in a cross-validation manner. During testing, we randomly sampled 1000 support-query pairs for evaluation.

The COCO-20^i dataset is constructed from the MS COCO 2014 dataset [22]. There are 80 classes in total, which are split into 4 folds, with each containing 20 classes as in [26]. We train and test the model with the same protocol as on the PASCAL-5^i dataset.

Evaluation Metrics. For a fair comparison, we follow previous methods [26, 32] to compute the mean Intersection over Union (mIoU) as the evaluation metric. For each test fold, mIoU is the average of the Intersection over Union of different novel classes. After obtaining the mIoU score of the four test folds, we compute the average of them as the overall performance of the model. We report the mIoU of each fold and their average score in the experiments.

5.2 Implementation Details

Following [42, 43], we adopt the dilated ResNet-50 [18] and an ASPP [2] module as the backbone CNNs. Specifically, the layers after the block3 of ResNet50 are removed and ASPP is applied directly upon the features from the block3. The ResNet-50 is pre-trained on ImageNet [5]. All the convolutional operations after the block3 of ResNet-50 generate features of 256 channels. We do not train the parameters of ResNet50. The nonlinear function of GAM is sigmoid. Eventually, the bilinear interpolation is applied to the output of DENet to produce the segmentation probability map with the same spatial size as the ground truth. All the experiments are conducted on PyTorch.

DENet and the re-implemented models are trained under the 1-way 1-shot setting with a batch size of 8. We train the models for 150k and 200k iterations on PASCAL-5^i and COCO-20^i respectively. During training, we use an SGD optimizer with the momentum of 0.9. The learning rate is set to 0.0025 with the weight decay set to 0.005. The λ in Eqn. (9) is set to 1.0 by default. We only use random horizontal flipping for data augmentation. The images used for both training and testing are resized to 321×321.

5.3 Quantitative Comparisons

In this section, we compare DENet with state-of-the-art methods and compare their quantitative results. We conduct experiments on N-way K-shot settings as in [32, 40], where $N = \{1, 2\}$ and $K = \{1, 5\}$. In each N-way task, the query image contains at least 1 class from the N novel classes for recognition. The quantitative comparisons on both the PASCAL-5^i and COCO-20^i datasets are shown Table 1 and Table 2, respectively.

Results on PASCAL-5^i. From Table 1, DENet reaches the highest mIoU under all $\{1, 2\}$-way and $\{1, 5\}$-shot settings. In particular, DENet outperforms CANet [43] by 2.37% under the 1-way 1-shot setting and 1.86% under the 1-way 5-shot setting respectively. It is worth noting that DENet under the 1-way 1-shot setting outperforms FWB [26] under the 1-way 5-shot setting. These results imply that DENet has better generalization ability on novel classes than state-of-the-arts. In the more challenging 2-way settings, although the number of novel classes presented in one support set increases (*i.e.*, N increases), DENet still achieves state-of-the-art performance.

Results on COCO-20^i. Table 2 reports the comparison results with state-of-the-arts on COCO-20^i. This is a more challenging dataset since it has more novel classes than PASCAL-5^i, *i.e.*, 5 *v.s.* 20. From Table 2, DENet consistently achieves great improvements under all $\{1, 2\}$-way and $\{1, 5\}$-shot settings compared with previous methods, often by a large margin.

5.4 Qualitative Results

We also compare the qualitative results of different methods under the 1-way setting and the 2-way setting in Figure 2 and Figure 3, respectively. For fair comparisons, we set 1-shot in all experiments.

From Figure 2 and Figure 3, we draw several conclusions. **First**, DENet is able to produce the full mask that contains both the base classes and novel classes (*e.g.*, the novel class car and the base class person shown in Figure 2 row 1). This indicates that our model has effectively preserved the ability to recognize the base classes. In contrast, the previous methods only separate the novel class from other classes. **Second**, the quality of the novel masks produced by our model is often better than that produced by other methods. In Figure 2 row 1, our method clearly separates the person and the car.

Table 1: Performance comparison with state-of-the-art models on the PASCAL-5^i dataset in terms of mIoU (%). The bold number indicates the best result. "*" denotes our re-implementation. "–" denotes that the results are not reported.

N-Way	Model	1-shot					5-shot				
		fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean
1	OSLSM [32]	33.60	55.30	40.90	33.50	40.80	35.90	58.10	42.70	39.10	43.90
	Co-FCN [29]	36.70	50.60	44.90	32.40	41.10	37.50	50.00	44.10	33.90	41.38
	PL [6]	–	–	–	–	42.70	–	–	–	–	43.70
	AMP [33]	41.90	50.20	46.70	34.70	43.40	41.80	55.50	50.30	39.90	46.90
	AMP* [33]	31.06	44.78	44.51	31.61	37.99	35.60	51.40	50.84	36.84	43.67
	SG-One [44]	40.20	58.40	48.40	38.40	46.30	41.90	58.60	48.60	39.40	47.10
	PANet [40]	42.30	58.00	51.10	41.20	48.10	51.80	64.60	59.80	46.50	55.70
	FWB [26]	51.30	64.49	56.71	**52.24**	56.19	54.84	67.38	62.16	**55.30**	59.92
	CANet [43]	52.50	65.90	51.30	51.90	55.40	55.50	67.80	51.90	53.20	57.10
	CANet* [43]	**56.34**	69.12	55.65	49.72	57.71	**56.99**	70.22	57.40	49.79	58.60
	PGNet [42]	56.00	66.90	50.60	50.40	56.00	54.90	67.40	51.80	53.00	56.80
	PGNet* [42]	52.68	66.45	50.68	49.53	54.84	28.09	65.41	52.08	48.44	48.51
	CRNet [24]	–	–	–	–	55.70	–	–	–	–	58.80
	FSS-1000 [21]	–	–	–	–	–	50.61	70.29	58.43	55.08	58.60
	DENet	55.74	**69.69**	**63.62**	51.26	**60.08**	54.72	**70.99**	**64.51**	51.63	**60.46**
2	PANet [40]	–	–	–	–	45.10	–	–	–	–	53.10
	CANet* [43]	**44.44**	49.65	42.06	37.51	43.42	**46.74**	52.12	44.85	38.42	45.53
	PGNet* [42]	39.23	53.11	14.21	36.17	35.68	14.71	43.30	37.88	21.60	29.37
	MetaSeg [37]	–	–	–	–	–	–	–	43.30	–	–
	DENet	44.26	**61.94**	**55.74**	**46.73**	**52.17**	45.53	**62.13**	**58.68**	**48.14**	**53.62**

Table 2: Performance comparison with state-of-the-art models on the COCO-20^i dataset in terms of mIoU (%). The bold number indicates the best result. "*" denotes our re-implementation. "–" denotes that the results are not reported.

N-way	Model	1-shot					5-shot				
		fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean
1	AMP* [33]	25.27	23.15	19.58	21.18	22.30	30.14	28.17	25.83	26.12	27.57
	FWB [26]	16.98	17.98	20.96	28.85	21.19	19.13	21.46	23.93	30.08	23.65
	CANet* [43]	42.21	42.70	37.58	**40.88**	40.84	44.65	43.01	37.54	**42.67**	41.97
	PGNet* [42]	39.54	39.68	33.90	33.49	36.65	42.37	38.87	32.42	36.47	37.53
	DENet	**42.90**	**45.78**	**42.16**	40.22	**42.77**	**45.40**	**44.86**	**41.57**	40.26	**43.02**
2	CANet* [43]	25.90	25.58	24.47	24.05	24.93	31.15	23.93	26.22	27.31	27.15
	PGNet* [42]	23.15	24.51	20.80	19.75	22.05	30.73	23.46	21.22	24.91	25.08
	MetaSeg [37]	–	–	–	–	33.20	–	–	–	–	37.90
	DENet	**38.96**	**39.76**	**38.21**	**37.16**	**38.52**	**41.15**	**42.24**	**40.94**	**39.16**	**40.87**

However, the other two models produce disorganized boundaries. **Third**, DENet well recognizes the objects in query images despite the fact that the objects of the same category in the support set and the query image are different in color, size, and perspective (*e.g.*, the dog and the skateboard in Figure 3). This demonstrates the effectiveness of DENet in complicated scenarios.

6 FURTHER EXPERIMENTS

6.1 Ablation Studies

We conduct the ablation studies to investigate how different modules affect the model performance. Specifically, we evaluate DENet with or without the base classifiers and the GAM. In all experiments, we report the average results of the four test folds on the more challenging COCO-20^i dataset as shown in Table 3.

First, DENet drops its performance by 3.85% and cannot outperform previous methods without the participation of the base classifiers. This decline indicates that preserving base classifiers learned by our end-to-end DENet benefits the learning of novel classes. **Second**, without GAM, the overall performance of DENet is decreased by 1.44%. This manifests that GAM can better extract the task-relevant features to recognize the novel classes. It is worth noting that DENet achieves better results with fewer parameters.

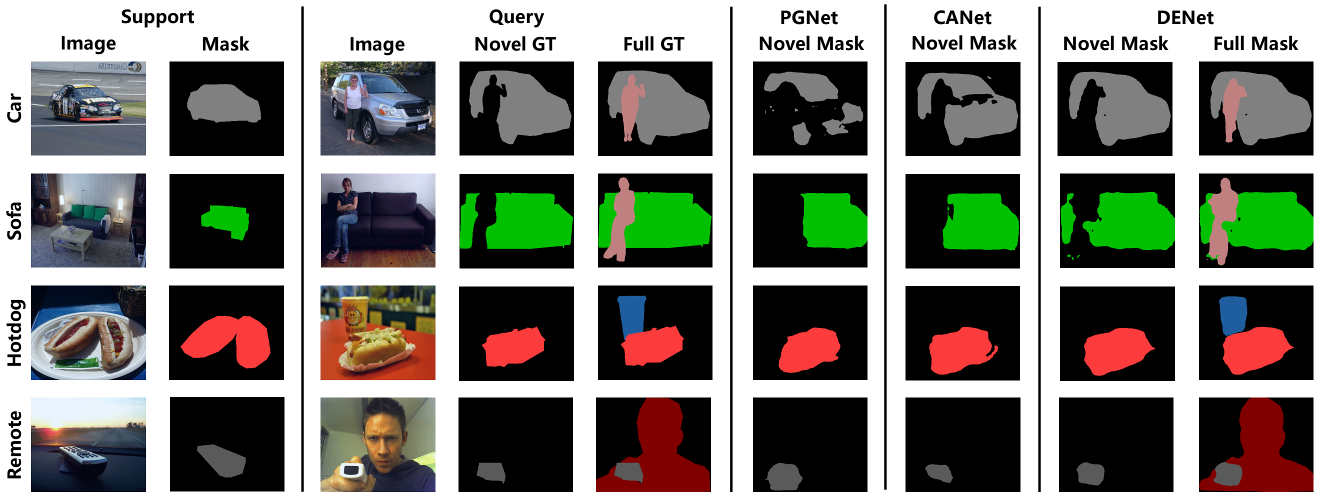

Figure 2: Qualitative results of 1-shot 1-way segmentation on the PASCAL-5^i dataset (the first two rows) and the COCO-20^i dataset (the third and fourth rows). Note that "Novel GT"/"Novel Mask" is the ground-truth/prediction mask that contains only the novel class while the "Full GT"/"Full Mask" includes both base and novel classes.

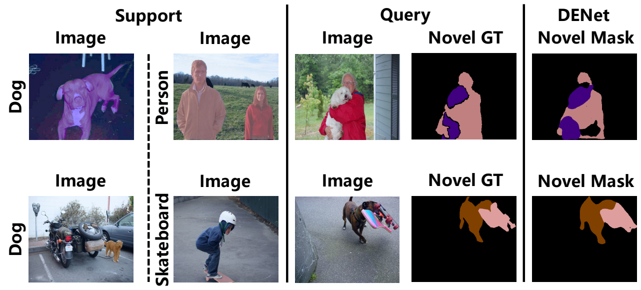

Figure 3: Qualitative results of 1-shot 2-way segmentation on the PASCAL-5^i dataset (the first row) and the COCO-20^i dataset (the second row).

Table 3: Ablation studies of DENet on the COCO-20^i dataset. "$*$" denotes our re-implementation.

Model	#Params (M)	1-shot		
		Base Classifiers	GAM	mIoU (%)
PGNet* [42]	33.94	–	–	36.65
CANet* [43]	34.00	–	–	40.84
DENet	28.89	×	√	38.92
DENet	28.75	√	×	41.33
DENet	28.89	√	√	**42.77**

These experiments convincingly demonstrate the effectiveness of the preserved base classifiers and GAM.

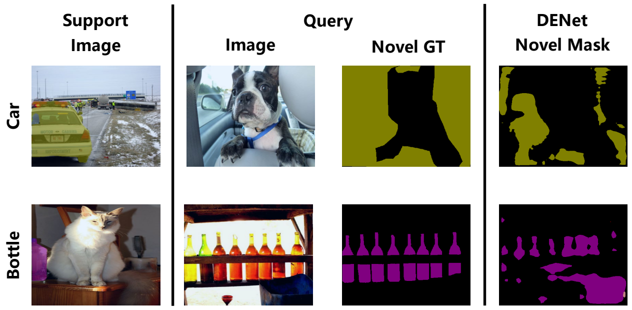

Figure 4: Failure cases of DENet on the COCO-20^i dataset (the first row) and the PASCAL-5^i dataset (the second row).

6.2 Effect of λ in Eqn. (9)

In this experiment, we investigate the effect of the trade-off parameter λ in our objective function described in Eqn. (9). The experiments are conducted on COCO-20^i under the 1-way settings. We evaluate the performance of DENet with $\lambda = 0.1, 1.0, 5.0$, and 10.0 respectively. From Table 4, DENet yields the best result under the 1-shot setting when $\lambda = 1.0$, while $\lambda = 5.0$ brings the best performance under the 5-shot setting. When $\lambda = 0.1$, the performance considerably decreases, which demonstrates that it is important and necessary to sufficiently optimize the weight estimator during training. When $\lambda = 1.0$ or $\lambda = 5.0$, the performance is relatively insensitive in terms of different λ. When $\lambda = 10.0$, the overall training objective is dominated by the optimization of the weight estimator instead of the recognition of base classes. This impairs the learning of base class knowledge and further deteriorates the novel class recognition.

Table 4: The effect of λ on the performance of DENet.

λ		0.1	1.0	5.0	10.0
mIoU (%)	1-shot	26.67	**42.77**	42.35	40.94
	5-shot	27.09	43.02	**44.07**	42.64

Table 5: Comparison of inference time. The bold number indicates the best result. "*" denotes our re-implementation.

Model	Support-free	Time (second)	
		1-shot	5-shot
PGNet* [42]	×	**0.09**	0.21
CANet* [43]	×	0.11	0.35
FWB* [26]	×	0.18	0.48
DENet	×	**0.09**	**0.20**
DENet	√	**0.04**	**0.04**

6.3 Failure Cases of DENet

The failure cases of DENet are shown in Figure 4. We find that the predicted masks of DENet are sometimes discontinuous, potentially because the model lacks the global clues of the query images. These problems are also challenging in the normal semantic segmentation tasks [19], and become harder in the few-shot setting. Furthermore, we empirically find that our method may fail to segment very small objects, *e.g.*, the bottle. The main reason is that it is hard to learn accurate attention for small objects. As a result, the estimated classifier may yield inferior performance on some of the classes.

6.4 Comparisons of Inference Time

In this experiment, we analyze the inference time of DENet in terms of support-based and support-free prediction. We compare DENet with several support-based methods [26, 42, 43]. The inference time (in terms of seconds) is tested on an NVIDIA Titan XP GPU with an input image of size 321 × 321. We take the average of the inference time over 100 runs. From Table 5, the support-based version of DENet has the lowest inference time compared with other considered methods, and the support-free version is even more efficient. This is a considerable advantage in some real-life scenarios with strict time constraints.

6.5 More discussions on DENet

Generalization ability to unseen classes. To further demonstrate the generalization ability of DENet, we take the classes from PASCAL as the base classes to train the model and evaluate it using the novel classes from another dataset, i.e., COCO. Specifically, we directly use the four models trained on each fold of PASCAL-5^i to perform 1-way 1-shot segmentation on 61 classes of COCO, where the overlapping classes are excluded. The results are then averaged across four models. From Table 6, DENet outperforms previous state-of-the-art methods.

More discussions on GAM. Although the GAM module seems simple, it is effective and sufficient to learn the weights of novel classifiers with two convolutions. First, GAM extracts the class-relevant information by exploiting a binary support mask w.r.t. a single class, which is much easier than learning the segmentation masks with multiple classes. Second, once we learn a good backbone CNNs, it is easy for GAM to extract effective information to perform weight estimation. To verify this, we compare the models with different numbers of convolutions in GAM. In practice, we do not observe significant performance improvement when increasing the number of convolutions from 2 to 10 (See Table 7).

Table 6: Experimental results of training on PASCAL and testing on COCO in terms of mIoU (%).

Model		CANet [43]	PGNet [42]	DENet
mIoU (%)	1-shot	34.07	33.96	**34.11**
	5-shot	35.75	31.37	**35.90**

Table 7: The effect of the number of convolutional layers in GAM on COCO-20^i dataset.

#Layers	2	5	10
mIoU (%)	42.77	42.79	**42.87**

More discussions on Dynamic Classifier. When adding a novel classifier into the dynamic classifier, the novel classifier may introduce a bias, which, however, is relatively small. We jointly train the base and "novel" classifiers by our dynamic extending training algorithm. In this way, the bias can be significantly reduced. To verify this, we compare the segmentation performance on base classes using the models with and without the novel classifier on the PASCAL-5^i dataset. In practice, the novel classifier slightly reduces the mIoU score of base classes from 71.19% to 70.43%, which verifies our statements.

7 CONCLUSION

In this paper, we propose a few-shot semantic segmentation method, called Dynamic Extension Network (DENet), which dynamically constructs and maintains the classifiers for the novel classes. Specifically, we design a Guided Attention Module (GAM) that takes both the support segmentation mask and support image to capture the class-relevant information. To exploit the knowledge learned by base classifiers, we propose a dynamic extension training method that estimates and incorporates novel classifiers dynamically. In this sense, DENet can dynamically extend classes and recognize both base and novel classes during inference. Extensive experiments on the PASCAL-5^i and COCO-20^i datasets demonstrate the superiority of our method over the considered methods.

ACKNOWLEDGMENTS

This work was partially supported by Science and Technology Program of Guangzhou, China under Grants 202007030007, Key-Area Research and Development Program of Guangdong Province 2019B010155001, National Natural Science Foundation of China (NSFC) 61836003 (key project), Guangdong 2017ZT07X183, Fundamental Research Funds for the Central Universities D2191240.

REFERENCES

[1] Jonathan C Balloch, Varun Agrawal, Irfan Essa, and Sonia Chernova. 2018. Unbiasing semantic segmentation for robot perception using synthetic data feature transfer. *arXiv preprint arXiv:1809.03676* (2018).

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (2017), 834–848.

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

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

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

[6] Nanqing Dong and Eric Xing. 2018. Few-shot semantic segmentation with prototype learning.. In *British Machine Vision Conference*, Vol. 3.

[7] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. 2011. The pascal visual object classes challenge 2012 (voc2012) results (2012). In *URL http://www. pascal-network. org/challenges/VOC/voc2011/workshop/index. html.*

[8] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. 2012. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2012), 1915–1929.

[9] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 4 (2006), 594–611.

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

[11] Chuang Gan, Ming Lin, Yi Yang, Gerard de Melo, and Alexander G. Hauptmann. 2016. Concepts Not Alone: Exploring Pairwise Relationships for Zero-Shot Video Activity Recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence.* 3487.

[12] Chuang Gan, Ming Lin, Yi Yang, Yueting Zhuang, and Alexander G. Hauptmann. 2015. Exploring Semantic Inter-Class Relationships (SIR) for Zero-Shot Action Recognition. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence.* 3769–3775.

[13] Spyros Gidaris and Nikos Komodakis. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 4367–4375.

[14] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. 2020. Closed-loop Matters: Dual Regression Networks for Single Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5407–5416.

[15] Yong Guo, Yaofo Chen, Yin Zheng, Peilin Zhao, Jian Chen, Junzhou Huang, and Mingkui Tan. 2020. Breaking the Curse of Space Explosion: Towards Efficient NAS with Curriculum Search. In *Proceedings of the 37th International Conference on Machine Learning.*

[16] Yong Guo, Yin Zheng, Mingkui Tan, Qi Chen, Jian Chen, Peilin Zhao, and Junzhou Huang. 2019. Nat: Neural architecture transformer for accurate and compact architectures. In *Advances in Neural Information Processing Systems.* 737–748.

[17] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. 2011. Semantic contours from inverse detectors. In *Proceedings of the IEEE International Conference on Computer Vision.* 991–998.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 770–778.

[19] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. 2020. Strip Pooling: Rethinking Spatial Pooling for Scene Parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4003–4012.

[20] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2. Lille.

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

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision.* Springer, 740–755.

[23] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. 2019. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 82–92.

[24] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. 2020. CRNet: Cross-Reference Networks for Few-Shot Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4165–4173.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3431–3440.

[26] Khoi Nguyen and Sinisa Todorovic. 2019. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE International Conference on Computer Vision.* 622–631.

[27] Dzung L Pham, Chenyang Xu, and Jerry L Prince. 2000. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering* 2, 1 (2000), 315–337.

[28] Hang Qi, Matthew Brown, and David G Lowe. 2018. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 5822–5830.

[29] 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 Track Proceedings.*

[30] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A Efros, and Sergey Levine. 2018. Few-shot segmentation propagation with guided networks. *arXiv preprint arXiv:1806.07373* (2018).

[31] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *International Conference on Learning Representations.*

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

[33] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. 2019. AMP: Adaptive masked proxies for few-shot segmentation. In *Proceedings of the IEEE International Conference on Computer Vision.* 5249–5258.

[34] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. 2019. AMP: Adaptive masked proxies for few-shot segmentation. In *Proceedings of the IEEE International Conference on Computer Vision.* 5249–5258.

[35] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems.* 4077–4087.

[36] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. 2019. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision.* 5229–5238.

[37] Pinzhuo Tian, Zhangkai Wu, Lei Qi, Lei Wang, Yinghuan Shi, and Yang Gao. 2020. Differentiable Meta-Learning Model for Few-Shot Semantic Segmentation.. In *Association for the Advancement of Artificial Intelligence.* 12087–12094.

[38] Michael Treml, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, et al. 2016. Speeding up semantic segmentation for autonomous driving. In *MLITS, NIPS Workshop*, Vol. 2. 7.

[39] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems.* 3630–3638.

[40] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE International Conference on Computer Vision.* 9197–9206.

[41] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision.* Springer, 20–36.

[42] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. 2019. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision.* 9587–9595.

[43] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. 2019. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 5217–5226.

[44] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. 2020. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics* (2020).

[45] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. 2016. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 669–677.

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2881–2890.

[47] Tongxue Zhou, Su Ruan, and Stéphane Canu. 2019. A review: Deep learning for medical image segmentation using multi-modality fusion. *Array* (2019), 100004.

[48] Barret Zoph and Quoc V Le. 2017. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations.*