

# Few-shot Semantic Segmentation with Self-supervision from Pseudo-classes

Yiwen Li<sup>1</sup>

kate@robots.ox.ac.uk

Gratianus Wesley Putra Data<sup>1</sup>

gwpd@robots.ox.ac.uk

Yunguan Fu<sup>2,3</sup>

yunguan.fu.18@ucl.ac.uk

Yipeng Hu<sup>2</sup>

yipeng.hu@ucl.ac.uk

Victor Adrian Prisacariu<sup>1</sup>

victor@robots.ox.ac.uk

<sup>1</sup> Active Vision Lab

University of Oxford

Oxford, UK

<sup>2</sup> Centre for Medical Image Computing

and Wellcome/EPSRC Centre for

Interventional and Surgical Sciences,

University College London,

London, UK

<sup>3</sup> InstaDeep,

London, UK

---

## Abstract

Despite the success of deep learning methods for semantic segmentation, few-shot semantic segmentation remains a challenging task due to the limited training data and the generalisation requirement for unseen classes. While recent progress has been particularly encouraging, we discover that existing methods tend to have poor performance in terms of meanIoU when query images contain other semantic classes besides the target class. To address this issue, we propose a novel self-supervised task that generates random pseudo-classes in the background of the query images, providing extra training data that would otherwise be unavailable when predicting individual target classes. To that end, we adopted superpixel segmentation for generating the pseudo-classes. With this extra supervision, we improved the meanIoU performance of the state-of-the-art method by 2.5% and 5.1% on the one-shot tasks, as well as 6.7% and 4.4% on the five-shot tasks, on the PASCAL-5i and COCO benchmarks, respectively.

## 1 Introduction

The goal of semantic segmentation is to classify each pixel in an image into semantically meaningful classes. The data-driven representation learning enabled by deep neural networks has led to accurate and compelling segmentation results [4, 16]. However, most segmentation methods rely on large datasets with full annotations, which are expensive and labour-intensive to collect. Furthermore, the predictable categories are often constrained to those that have been annotated in the dataset. It is, therefore, of great value to reduce the annotation burden by enabling unseen class prediction with only a few training examples. Specifically, given a small number of *support* images with ground truth binary masks of an unseen *target* class, the task is to segment the regions of the same class in unannotated *query* images. This is known as few-shot semantic segmentation (FSS) [21].

While existing methods [23, 24] have achieved significant improvements over fine-tuning and foreground-background segmentation baselines [21], we observed a significant drop in performance when the *query* image contains multiple classes (Figure 5), *i.e.* the more classes (including both training and testing classes defined in the dataset) the image has, the more difficult is it to segment the *target* class precisely.

For example, on the 1-shot PASCAL-5i [6, 24] benchmark, the current state-of-the-art (PFENet [23]) achieved a meanIoU score of 66.1% for *query* images with one class but only 32.7% for *query* images with four classes. A similar trend has also been observed in the 5-shot results and with other benchmarks such as few-shot COCO [24].

We conjecture that the performance drop is caused by the presence of the other non-target classes in the *query* images during training; Here, target class is the class of segmentation interest, whilst in binary segmentation tasks all the non-target classes are labelled as background regardless of their differences. Although the *target-versus-non-target* binary classification paradigm seems appropriate for images with single target class, it may inhibit the model from learning meaningful representations from the *non-target* pixels. Given practical limitations such as finite data and imperfect labels, further discriminating these non-target classes can potentially aid classifying *target* pixels, similar to multi-task learning [9]. This is especially true in few-shot learning, in which data-efficient model updating is sought-after.

Following the above intuition, we propose a novel self-supervised training strategy that generates pseudo-classes in the *query* images. For each episode, a pseudo-class is created by sampling superpixels with high activation from the *query* background. Together with the *query* image, an extra pseudo image-mask pair can be generated to assist training. By guiding the model to discriminate the pseudo-class from the background, this training paradigm enforces the model to distinguish possible non-target classes (those unannotated in the training set) present in the background of the *query* images. As details presented in the remainder of the paper, the proposed method leads to consistent improvement over different architectures and different datasets.

Our contributions are summarised as follows:

- We propose a novel self-supervised task that generates pseudo-classes for few-shot semantic segmentation, which encourage the models to learn a more discriminative feature space during adapting to *novel* target classes.
- We highlight the significant performance drop as the number of classes in *query* images increases for existing methods. To the best of our knowledge, it is the first time this issue is highlighted and investigated.
- We present an extensive set of 1-shot and 5-shot experiments that demonstrate the significant improvement from the proposed method, for both PASCAL-5i and COCO datasets. The performance on *query* images with multiple classes were improved and the imbalance between classes were mitigated.

## 2 Related Work

### 2.1 Few-shot Semantic Segmentation

OSLSM [21] first introduced the task of few-shot segmentation and demonstrated the effectiveness of episodic training, compared favourably with a fine-tuning baseline. PLNet [6] extracted class-wise prototypes from the support set and makes predictions based on cosine

similarity between pixels and prototypes. This two-step architecture is later inherited by many in this field.

Based on PLNet, extracting information from the support set was proposed to further improve the performance. For instance, PPNet [14] and RPMM [26] generated multiple prototypes for each class through clustering and Expectation-Maximisation. PGNet [29] and DAN [9] adopted the attention unit to adjust the foreground prototype according to query pixel features. OANet [32] incorporated an objectness segmentation module to calculate the objectness prior. FWB [18] introduced regularisation to suppress support image background activation and boosts the prototype with an ensemble of experts.

Other improvements can be achieved by refining the feature comparison module. In particular, multi-scale comparison were utilised to overcome spatial inconsistency [23, 26, 29, 30]. PANet [24] described a prototype alignment regularisation which encouraged the resulted segmentation model to perform few-shot learning in reverse direction.

However, these methods share the common limitation with binary masks, as discussed, during episodic training: pixels not belonging to the sampled target class in query images are all labelled as the same, background class, disregarding how semantically different/similar they are. In this work, we address this issue by generating pseudo-classes through superpixels, which creates more training classes and enables discrimination of these non-query classes from the background to provide a better training signal.

## 2.2 Superpixel Segmentation

Superpixel segmentation is an over-segmentation method that groups pixels coherently based on handcrafted features. It is widely used in image segmentation task to reduce computational costs [10, 17]. As superpixel provides additional local information, it has been used for few-shot semantic segmentation to compensate for the lack of annotated data. For example, SSL-ALPNet [19] used superpixel-based pseudo-labels to replace manual annotation. An important difference in our use of pseudo-label is to amend the performance drop when multiple classes present in the *query* image. PPNet [14] enriched support prototypes by extracting features from unlabelled support images based on superpixel segmentation. Rather than focusing on the support set, in this work, we sample superpixels to define pseudo-classes only on the background of the query images.

# 3 Method

## 3.1 Task Description

Few-shot semantic segmentation aims to learn a model that can segment novel classes when given only a few annotated examples of these classes. Specifically, denote  $(I, M(c))$  as an image-mask pair, where  $I$  represents an RGB image and  $M(c)$  represents its binary mask of a class  $c$ . The model is trained on a *base* dataset  $\mathcal{D}_{base} = \{(I_i, M_i(c)) \mid c \in \mathcal{C}_{base}\}_{i=1}^N$  and tested on a *novel* dataset  $\mathcal{D}_{novel} = \{(I_j, M_j(c)) \mid c \in \mathcal{C}_{novel}\}_{j=1}^{N'}$ , where *base* classes and *novel* classes are mutually exclusive *i.e.*  $\mathcal{C}^{base} \cap \mathcal{C}^{novel} = \emptyset$ .

Following the benchmark introduced by [21], the evaluation takes place in an episodic manner: a *target* class  $c \in \mathcal{C}_{novel}$  is sampled for each *episode*, consisting of a *support* set  $\mathcal{S} = \{(I_i^s, M_i^s(c))\}_{i=1}^k$  and a *query* image-mask pair  $(I^q, M^q(c))$  of the *target* class. The model is tasked to predict the *query* mask  $M^q(c)$  given the *query* image  $I_q$  and the *support* set  $\mathcal{S}$ .

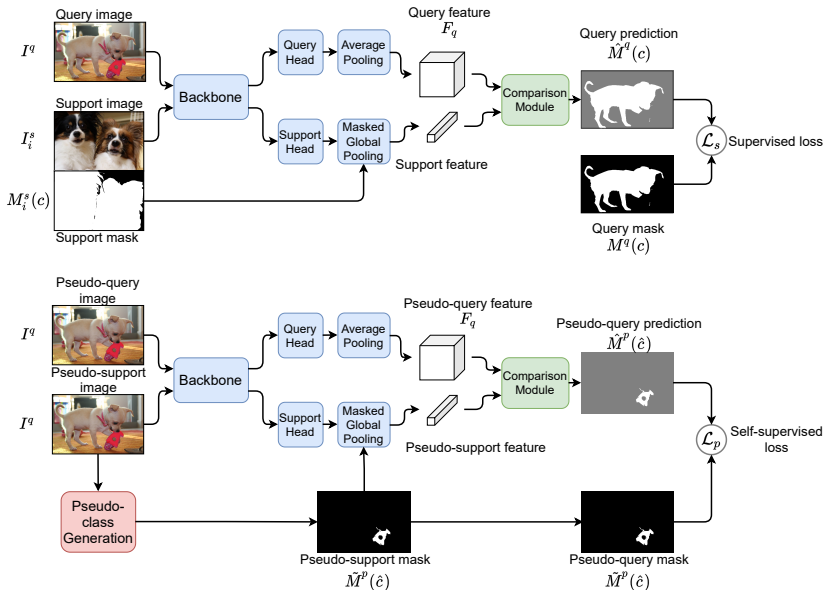


Figure 1: An illustration of training data flow. The upper diagram demonstrates the supervised pathway which predicts the query mask, conditioned on the given support example. The lower diagram demonstrates the self-supervised pathway which generates a pseudo-class and predicts the generated mask, conditioned on the pseudo-support example. Networks are shared between two pathways during training.

### 3.2 Self-supervision from Pseudo-classes

Most state-of-the-art few-shot semantic segmentation methods follow the episodic training paradigm. For each iteration, an *episode*, consisting a *support* set  $\mathcal{S} = \{(I_i^s, M_i^s(c))\}_{i=1}^k$  and a *query* image-mask pair  $(I^q, M^q(c))$ , where the *target* class  $c \in C_{base}$  is sampled from the *base* dataset  $\mathcal{D}_{base}$ .

As shown in Figure 1 both *query* and *support* images are encoded by the feature extractor into feature maps. The *target* class prototype is then derived through global average pooling all foreground pixels of the support feature maps. The comparison module makes the final prediction based on the input *query* feature maps and *target* class prototype.

The model is optimised to reduce  $\mathcal{L}(M^q(c), \hat{M}^q(c))$ , the cross entropy loss between the



Figure 2: Step-by-step breakdown of pseudo-mask generation for self-supervision. (a) Input query image. (b) Generate superpixels in the query image. (c) Remove superpixels that coincide with the original class ground truth. (d) Randomly sample superpixel with high corresponding feature activation, and convert it into a binary mask.

predicted *query* mask  $\hat{M}^q(c)$  and the ground-truth *query* mask  $M^q(c)$ . Denote  $\mathcal{L}(M, \hat{M})$  as the spatially averaged cross-entropy loss between ground-truth  $M$  and prediction  $\hat{M}$ :

$$\mathcal{L}(M, \hat{M}) = -\frac{1}{WH} \sum_{x=1}^W \sum_{y=1}^H [M_{(x,y)} \log \hat{M}_{(x,y)} + (1 - M_{(x,y)}) \log(1 - \hat{M}_{(x,y)})], \quad (1)$$

where  $M_{(x,y)}$  represents the value of mask  $M$  at pixel  $(x, y)$ .

Under this training paradigm, *non-target*-class objects with different semantic meanings in the *query* image are all regarded as the same class – background. The purposed self-supervision task aims to better utilise the information from the background pixels.

First, we apply a superpixel segmentation method (such as Felzenszwalb *et al.* [21]) to  $I^q$ . For each superpixel, denote its region as the pseudo-class  $\hat{c}$  and the binary mask representing the region as the pseudo-mask  $M^p(\hat{c}) \in \{0, 1\}^{H \times W}$ . Second, we refine each pseudo-mask  $M^p(\hat{c})$  with the aid of the ground-truth mask  $M^q(c)$  by removing pixels of the *target* class  $c$ ,

$$\tilde{M}^p(\hat{c}) = M^p(\hat{c}) \odot (1 - M^q(c)), \quad (2)$$

where  $\odot$  denote the element-wise multiplication. Then, a class activation score  $s(\hat{c})$  is calculated for each pseudo-class  $\hat{c}$  by averaging the extracted query feature  $F^q$  over the pseudo-mask foreground and channels, specifically:

$$s(\hat{c}) = \frac{\sum_{x=1}^W \sum_{y=1}^H \sum_{z=1}^d F_{(x,y,z)}^q \tilde{M}_{(x,y)}^p(\hat{c})}{\sum_{x=1}^W \sum_{y=1}^H \sum_{z=1}^d \tilde{M}_{(x,y)}^p(\hat{c})}. \quad (3)$$

Lastly, a pseudo-class  $\hat{c}$  is randomly selected among the ones having the top-5 scores, so that the selected pseudo-class is more likely to be non-background while keeping diversity between pseudo-classes. The corresponding refined mask is denoted as  $\tilde{M}^p(\hat{c})$ .

As shown in Figure 1, the additional image-mask pair  $(I^q, \tilde{M}^p(\hat{c}))$  serves as both *pseudo-support* and *pseudo-query*, forming a *pseudo support-query* pair for  $I^q$  in the current training iteration. Specifically, the self-supervised task is defined as: given the *pseudo-support* example  $(I^q, \tilde{M}^p(\hat{c}))$ , the model is required to predict the same mask  $\tilde{M}^p(\hat{c})$  for the pseudo-class  $\hat{c}$ . The corresponding prediction is denoted as  $\hat{M}^p(\hat{c}) \in [0, 1]^{H \times W}$ . Similarly, we compute a cross-entropy loss  $\mathcal{L}(\tilde{M}^p(\hat{c}), \hat{M}^p(\hat{c}))$ , for this self-supervision.

The total loss is therefore a weighted sum of the supervised loss with the target class  $c$  and the self-supervised loss with the pseudo-class  $\hat{c}$ ,

$$\mathcal{L}_{\text{total}} = \mathcal{L}(M^q(c), \hat{M}^q(c)) + \alpha \mathcal{L}(\tilde{M}^p(\hat{c}), \hat{M}^p(\hat{c})), \quad (4)$$

where  $\alpha$  is a scaling coefficient to control the contribution of the self-supervised loss towards the overall objective.

## 4 Experiments

### 4.1 Implementation Details

We evaluated our methods on the PASCAL-5i [22] (which consists of PASCAL VOC 2012 [6] with extended annotations from SDS [8]) and COCO [23] datasets. Each dataset has 4 configurations (folds) of base-novel class splits, following OLSM [22] and FWB [18] protocol.

We chose PFENet [23], the current state-of-the-art, as our baseline architecture. We also followed the training and evaluation setting of the original paper [23] while removing the train-free prior generation module. Felzenszwalb *et al.* [4] was used for superpixel segmentation with scale set to 100, sigma to 0.8 and min\_size to 200. The derived superpixel segmentations were further refined by the training (base) class masks. The hyper-parameter  $\alpha$  was set to 0.5 in equation (4). We report the meanIoU metric for evaluation, which is the intersection-over-union (IoU) averaged over classes. For 5-shot evaluation, the individual 1-shot models were used without fine-tuning and we averaged the resulting prototype vectors of supports before feeding them to the comparison module. For brevity, the prefix “SS-” denotes models trained with the proposed self-supervision.

## 4.2 Results

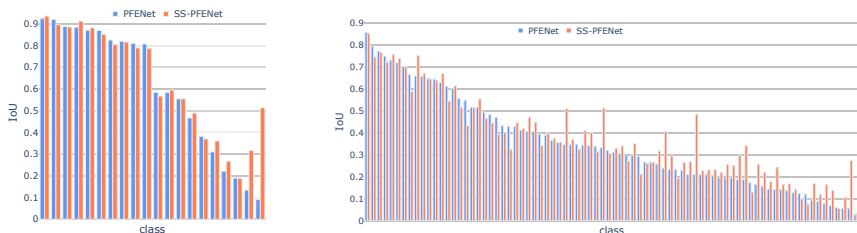
| method      | 1-shot       |              |              |              |              | 5-shot       |              |              |              |              |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|             | fold1        | fold2        | fold3        | fold4        | mean         | fold1        | fold2        | fold3        | fold4        | mean         |
| VGG-16      |              |              |              |              |              |              |              |              |              |              |
| OSLSM [24]  | 33.6         | 55.3         | 40.9         | 33.5         | 40.8         | 37.5         | 50.0         | 44.1         | 33.9         | 41.4         |
| co_FCN [20] | 36.7         | 50.6         | 44.9         | 32.4         | 41.2         | 35.9         | 58.1         | 42.7         | 39.1         | 44.0         |
| AMP [25]    | 41.9         | 50.2         | 46.7         | 34.7         | 43.4         | 41.8         | 55.5         | 50.3         | 39.9         | 46.9         |
| Sg-one [6]  | 40.2         | 58.4         | 48.4         | 38.4         | 46.4         | 41.9         | 58.6         | 48.6         | 39.4         | 47.1         |
| FWBNet [18] | 47.0         | 59.6         | 52.6         | 48.3         | 51.9         | 50.9         | 62.9         | 56.5         | 50.1         | 55.1         |
| RPPM [26]   | 47.1         | 65.8         | 50.6         | 48.5         | 53.0         | 50.0         | 66.5         | 51.9         | 47.6         | 54.0         |
| CRNet [3]   | -            | -            | -            | -            | 55.2         | -            | -            | -            | -            | 58.5         |
| PANet [24]  | 42.3         | 58.0         | 51.1         | 41.3         | 48.2         | 51.8         | <b>64.6</b>  | 59.8         | 46.5         | 55.7         |
| SS-PANet    | <b>49.3</b>  | <b>60.8</b>  | <b>53.9</b>  | <b>45.2</b>  | <b>52.3</b>  | <b>54.5</b>  | 64.5         | <b>62.3</b>  | <b>52.0</b>  | <b>58.3</b>  |
| PFENet [23] | <b>56.9</b>  | <b>68.2</b>  | 54.4         | 52.4         | 58.0         | <b>59.0</b>  | 69.1         | 54.8         | 52.9         | 59.0         |
| SS-PFENet   | 54.5         | 67.4         | <b>63.4</b>  | <b>54.0</b>  | <b>59.8</b>  | 56.9         | <b>70.0</b>  | <b>68.3</b>  | <b>62.1</b>  | <b>64.3</b>  |
| ResNet-50   |              |              |              |              |              |              |              |              |              |              |
| PPNet [14]  | 48.6         | 60.6         | 55.7         | 46.5         | 52.9         | 58.9         | 68.3         | 66.8         | 58.0         | 63.0         |
| RPMM [26]   | 50.2         | 66.9         | 52.6         | 50.7         | 55.1         | 56.3         | 68.3         | 54.5         | 51.0         | 57.5         |
| CANet [6]   | 52.5         | 65.9         | 51.3         | 51.9         | 55.4         | 55.5         | 67.8         | 51.9         | 53.2         | 57.1         |
| CRNet [3]   | -            | -            | -            | -            | 55.7         | -            | -            | -            | -            | 58.8         |
| PGNet [29]  | 56.0         | 66.9         | 50.6         | 50.4         | 56.0         | 57.7         | 68.7         | 52.9         | 54.6         | 58.5         |
| ORG [27]    | 52.6         | 65.8         | 54.7         | 52.1         | 56.3         | 57.2         | 67.8         | 57.5         | 56.2         | 59.7         |
| OANet [6]   | 56.9         | 66.4         | 57.1         | 50.7         | 57.8         | 59.6         | 68.9         | 57.8         | 54.4         | 60.2         |
| ASGNet [10] | 58.8         | 67.9         | 56.8         | 53.7         | 59.3         | 63.66        | 70.6         | 64.2         | 57.4         | 64.0         |
| RePRI [0]   | 60.2         | 67.0         | 61.7         | 47.5         | 59.1         | <b>64.5*</b> | 70.8         | <b>71.7*</b> | 60.3         | 66.8         |
| SCL [28]    | <b>63.0*</b> | <b>70.0*</b> | 56.5         | <b>57.7*</b> | 61.8         | <b>64.5*</b> | 70.9         | 57.3         | 58.7         | 62.9         |
| PFENet [23] | 61.7         | 69.5         | 55.4         | 56.3         | 60.7         | 63.1         | 70.7         | 55.8         | 57.9         | 61.9         |
| SS-PFENet   | 58.9         | 69.9         | <b>66.4*</b> | <b>57.7*</b> | <b>63.2*</b> | 61.4         | <b>75.0*</b> | 70.5         | <b>67.7*</b> | <b>68.6*</b> |

Table 1: meanIoU results on PASCAL-5i [21]. Best performance with the same model and overall are respectively bolded and starred. “SS-” refers to models with self-supervision.

The meanIoU performances per fold for PASCAL-5i 1-shot and 5-shot tasks are presented in Table 1. By incorporating the self-supervision during the training of ResNet-50 PFENet, we achieved 63.2% meanIoU on the PASCAL-5i 1-shot task, improving its baseline performance by 2.5%. For the 5-shot task, we achieved 68.6% meanIoU with a 6.7% absolute improvement. This sets a new state-of-the-art on the PASCAL-5i dataset for both 1-shot and 5-shot. On the COCO benchmark (Table 2), we achieved 37.5% and 41.8% mean-

| method      | 1-shot       |              |              |              |              | 5-shot       |              |              |              |              |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|             | fold1        | fold2        | fold3        | fold4        | mean         | fold1        | fold2        | fold3        | fold4        | mean         |
| VGG-16      |              |              |              |              |              |              |              |              |              |              |
| FWBNet [13] | 18.4         | 16.7         | 19.6         | 25.4         | 20.0         | 20.9         | 19.2         | 21.9         | 28.4         | 22.6         |
| PANet [24]  | 28.7         | 21.2         | 19.1         | 14.8         | 21.0         | 39.4         | 28.3         | 28.2         | 22.7         | 29.7         |
| SS-PANet    | <b>29.8</b>  | <b>21.2</b>  | <b>26.5</b>  | <b>28.5</b>  | <b>26.2</b>  | <b>36.7</b>  | <b>41.0</b>  | <b>37.6</b>  | <b>35.6</b>  | <b>37.7</b>  |
| PFENet [23] | 34.3         | 33.0         | 32.3         | 30.1         | 32.4         | 38.5         | 38.6         | 38.2         | 34.3         | 37.4         |
| SS-PFENet   | <b>35.6*</b> | <b>39.2*</b> | <b>37.6*</b> | <b>37.3*</b> | <b>37.5*</b> | <b>40.4*</b> | <b>45.8*</b> | <b>40.3*</b> | <b>40.7*</b> | <b>41.8*</b> |

Table 2: meanIoU results on COCO [6]. Best performance with the same model and overall are respectively bolded and starred. “SS-” refers to models with self-supervision.



(a) PASCAL-5i

(b) COCO

Figure 3: Class IoUs reached by PFENet with and without superpixel self-supervision. X-axis corresponds to different classes. A larger version with class names annotated is available in the supplementary material.

IoU for 1-shot and 5-shot using VGG-16 PFENet and set the new state-of-the-art from our knowledge.

Additionally, we present the IoU values per class in Figure 3 for PFENet with and without self-supervision on both PASCAL-5i and COCO. Most of the improvement came from previously under-performing classes (i.e person, dining table, potted plant, etc.), and thus led to a more class-wise balanced performance.

The meanIoU performances on images with the different number of classes are plotted in Figure 5. Here the number of classes is defined by the ground-truth labels. Therefore, for PASCAL and COCO, the maximum available numbers of classes in a query image are respectively 20 and 80. From Figure 5(a) and Figure 5(c), one can observe that the proposed method consistently improved the performance when multiple classes were presented in the query image over both PASCAL-5i and COCO dataset. This explains the higher relative improvement in performance achieved on COCO (11.8%) compare to PASCAL-5i (3.1%) when the same backbone (VGG-16) was adopted.

Figure 4 shows some qualitative examples achieved by ResNet-50 PFENet with and without the purposed self-supervision. When multiple objects exist in the query image, PFENet succeeds in detecting them from the background but fails to filter out non-target-class objects. This failure in discriminating class differences has been rectified by the proposed self-supervised task which is likely to increase class diversities due to introducing pseudo-classes.

To further understand the impact of the proposed self-supervised task, we used t-SNE [15] to visualise the foreground embedding from query images in Figure 6. Each dot corresponds to the global average pooled query foreground feature from a testing episode, and different colours represent different classes. The point clouds of different classes tend to be better

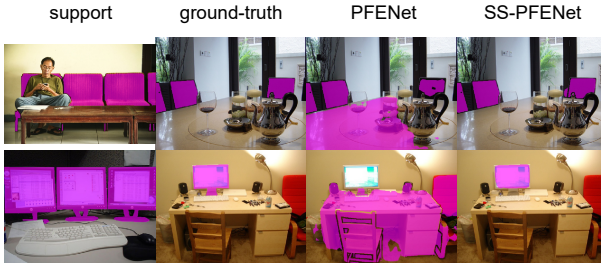


Figure 4: Qualitative results of ResNet-50 PFENet with and without the proposed supervision. More examples are available in the supplementary materials.

disentangled, which qualitatively supports our theory and explains the observed results.

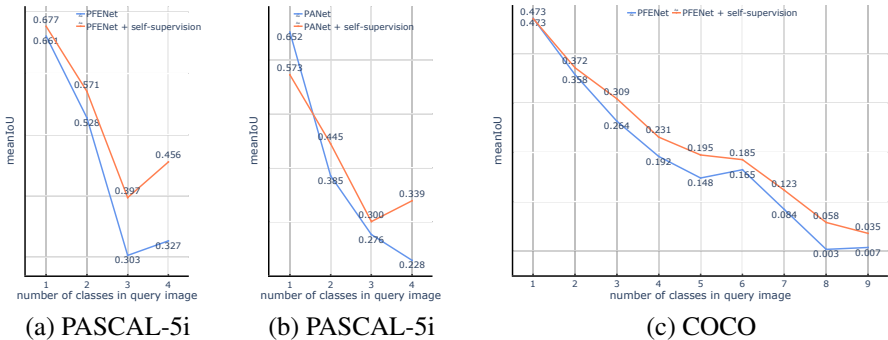


Figure 5: meanIoUs reached with and without superpixel self-supervision when different number of classes occur in the query image.

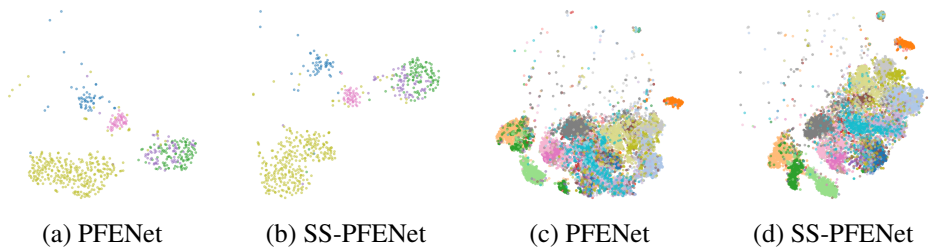


Figure 6: Visual comparison between feature spaces from PFENet trained with and without the proposed self-supervision. (a) and (b) were trained on PASCAL-5i [24] excluding *fold2* classes, (c) and (d) were trained on COCO excluding *fold3* classes.

## 4.3 Ablation Studies

### 4.3.1 Model Architecture

To test the applicability of the proposed self-supervision across different architectures, further experiments were performed on PANet [24], an intrinsically different architecture that



| method  | PFENet | SS-PFENet |      |             |              |
|---------|--------|-----------|------|-------------|--------------|
|         |        | gridding  | SLIC | HED         | Felzenszwalb |
| meanIoU | 60.7   | 57.5      | 62.9 | <b>63.3</b> | 63.2         |

Table 3: Performance of ResNet-50 PFENet on PASCAL-5i 1-shot task with self-supervision using different superpixel segmentation algorithms.

| method                    | PFENet | SS-PFENet |             |       |
|---------------------------|--------|-----------|-------------|-------|
|                           |        | all       | top-5       | top-1 |
| pseudo-class choice range |        |           |             |       |
| meanIoU                   | 60.7   | 58.5      | <b>63.2</b> | 63.0  |

Table 4: Performance of ResNet50-PFENet on PASCAL-5i task with self-supervision when choosing pseudo-class from superpixels with the top-n activation scores.

adopts cosine similarity to compare both foreground and background prototypes with each pixel and make predictions. As shown in Table 1 and Table 2, the proposed method achieved 4.1% and 2.6% absolute improvement on PASCAL-5i 1-shot and 5-shot tasks as well as 5.2% and 8.0% absolute improvement on COCO 1-shot and 5-shot tasks. The consistent improvement on multi-class query images was also observed across architectures as shown in Figure 5(b).

#### 4.3.2 Pseudo-class Generation

We compared the following three other different pseudo-class generation processes: 1) Gridding: divide the image into  $10 \times 10$  grids evenly, 2) SLIC [10]: segment image through iterative clustering, setting compactness to 10 and  $n\_segments$  to 100, 3) HED contour detector [11]: leverage deep learning supervision to detect edges. The meanIoU on the PASCAL-5i 1-shot task are reported in Table 3. Gridding leads to a significant drop of performance to 57.5%, lower than being without self-supervision. The possible reason may be the presence of multiple classes together with the background in the same grid, making the pseudo-classes misleading. We chose to use the second-best algorithm Felzenszwalb instead of HED because HED was pre-trained on the SDS dataset [12], part of our test dataset, which may cause potential information leakage.

We also compared three different pseudo-class sampling strategies: 1) select the superpixel with the highest activation score calculated in Eq. 3, 2) sample a superpixel with the top-5 activation scores, 3) sample a superpixel from the entire image randomly. The meanIoUs on the PASCAL-5i 1-shot task are reported in Table 4, with the "top-5" strategy performing the best. This is understandable as on the one hand, top-5 provides more pseudo-classes than top-1 for training; on the other hand, selecting a random superpixel without ranking may result in over-segmenting, for example, a background region, e.g. sky, to be different to other background regions, e.g. other superpixels of the sky, thus misguiding the model.

#### 4.3.3 Effect of Novel Class in Training Background

Though there is no label of testing classes available during training, their existences in the training images could still lead to advantage when the sampled *pseudo-classes* overlap with those regions. This ablation study examines the performance of the purposed method without such advantage by masking all pixels belonging to the testing classes in training images so

| method           | fold1       | fold2       | fold3       | fold4       | mean        |
|------------------|-------------|-------------|-------------|-------------|-------------|
| PANet[23]        | 42.3        | 58.0        | 51.1        | 41.3        | 48.2        |
| SS-PANet         | <b>49.3</b> | <b>60.8</b> | 53.9        | <b>45.2</b> | <b>52.3</b> |
| SS-PANet-masked  | 49.1        | 60.7        | <b>54.0</b> | 45.0        | 52.2        |
| PFENet[23]       | <b>61.7</b> | 69.5        | 55.4        | 56.3        | 60.7        |
| SS-PFENet        | 58.9        | <b>69.9</b> | <b>66.4</b> | <b>57.7</b> | <b>63.2</b> |
| SS-PFENet-masked | 58.8        | 69.7        | 65.4        | 57.3        | 62.8        |

Table 5: meanIoU results on PASCAL-5i [23]. All models adopts ResNet-50 as backbone and best performance achieved by same model are bolded. “SS-” refers to models with self-supervision and “-mask” refers to models trained with pixels belonging to novel classes masked.

| $\alpha$ | 0.0  | 0.25 | 0.50        | 0.75 | 1.0  |
|----------|------|------|-------------|------|------|
| meanIoU  | 60.7 | 62.5 | <b>63.2</b> | 61.2 | 59.5 |

Table 6: Performance of ResNet-50 PFENet on PASCAL-5i [23] with self-supervision using different  $\alpha$  values.

these testing class objects will not be part of any pseudo-classes during training. As shown in Table 5, the meanIoU decreased only by 0.1% and 0.4% respectively for PANet and PFENet architecture. This minor decrease indicates the ability of the proposed self-supervision to generalise to completely unseen novel classes.

#### 4.3.4 Effect of $\alpha$ Parameter

We varied the self-supervised loss scaling factor  $\alpha$  and report the performance achieved on PASCAL-5i [23] in Table 6.  $\alpha = 0.5$  appears to be the optimal value. This is reasonable because the pseudo labels generated by superpixel segmentation are inaccurate (over-segmentation) and thus less aligned with our objective than ground-truth annotations.

## 5 Conclusion

In this work, we first raised the issue of the dependency between the few-shot segmentation performance and the inherent complexion of individual *query* images, specifically, the presence of additional classes. We have provided quantitative evidence showing accuracy decreases in segmentation for *novel* classes, as a result of more latent objects appearing in the background of the *target* class of interest. We directly address the lack of background class supervision, by proposing a novel strategy that generates self-supervising pseudo-labels for the non-target classes. A set of extensive experimental results have shown that the proposed approach consistently improved the few-shot segmentation performance on images that contain multiple classes of objects, for different network architectures on multiple datasets. Finally, we showed that different classes were indeed better disentangled in the embedding space using our method. The proposed methodology is not limited by the choice of pseudo-supervision generation method, here we proposed a superpixel-based approach that is simple and effective. Future work will investigate alternatives that, for instance, base on higher-level features to scale better to intra-class variations such as shape and colour variations, and their potentially latent relationship to the model adaptability and predictivity.

## References

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.
- [2] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13979–13988, 2021.
- [3] Xianbin Cao and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. 2020.
- [4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.
- [5] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.
- [6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [7] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.
- [8] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.
- [9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [10] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7334–7344, 2019.
- [11] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021.
- [12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [13] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4173, 2020.
- [14] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 142–158. Springer, 2020.
- [15] Maaten LJPvd and GE Hinton. Visualizing high-dimensional data using t-sne. *J Mach Learn Res*, 9:2579–2605, 2008.

- [16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [17] Branislav Mičušík and Jana Košecká. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 625–632. IEEE, 2009.
- [18] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [19] Cheng Ouyang, Carlo Biffi, Chen Chen, Turkay Kart, Huaqi Qiu, and Daniel Rueckert. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *European Conference on Computer Vision*, pages 762–780. Springer, 2020.
- [20] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018.
- [21] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.
- [22] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [23] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Annals of the History of Computing*, 2020.
- [24] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019.
- [25] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.
- [26] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 763–778. Springer, 2020.
- [27] Xiaowen Ying, Xin Li, and Mooi Choo Chuah. Weakly-supervised object representation learning for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1497–1506, January 2021.
- [28] Bingfeng Zhang, Jimin Xiao, and Terry Qin. Self-guided and cross-guided learning for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8312–8321, 2021.
- [29] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [30] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019.

- [31] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 2020.
- [32] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Objectness-aware one-shot semantic segmentation. *arXiv preprint arXiv:2004.02945*, 2020.