Replicating Few-shot Image Segmentation with Prototype Alignment

Bryan Carlo Miguel
bryancarlomiguel@gmail.com
Concordia University
Montreal, Quebec, Canada

Abstract

Traiditonal deep CNNs require extensive labeled datasets and are difficult to generalize to unseen object categories. This paper introduces PANet, which uses few-shot segmentation along with a novel technique, the prototype alignment network, to better utilize information from the support set. PANet learns class-specific prototype representations from a few support images within an embedded space, which then matches pixels over the query images from the learned prototypes. As such, PANet learns high-quality prototypes representative for each semantic class. Using prototype alignment network regularization (PAR), we obtain a better alignment between the prototypes learnt from the support set and the query images, resulting in better generalization on few-shot segmentation.

1 Introduction

As stated in the paper, image segmentation has significantly advanced using multiple CNN-based architectures such as FCN, SegNet, DeepLab, and PSPNet. These models, however, require large amounts of image data with pixel-level annotations. Additionally, they also generalize poorly to unseen classes. Few-shot segmentation is a technique used to alleviate these problems. In short, most few-shot segmentation methods learn from a **support set** containing images and their ground-truth (actual) segmentation mask. This learnt knowledge is then used on the **query set** where the 'actual' segmentation occurs. The main drawback is that the knowledge-extraction and segmentation processes are done 'all at once' and are not differentiated from each other. This could be problematic since the *segmentation model representation* and the *semantic features of the supports* share the same latent or embedding space. An optimization of parameters in the segmentation model could, for example, interfere with parameters in the model responsible for feature extraction.

1.1 Approaches

The paper proposes two novel approaches:

The first one being the separation of the aforementioned *knowledge-extraction* and *segmentation* processes. The original paper formally calls them **prototype extraction** and **non-parametric metric learning** respectively. During prototype extraction, differentiable feature vectors are generated for each class using a limited number of annotated support images. During non-parametric metric learning, we perform segmentation on the query set after the prototypes have been established. We use non-parametric metric learning to reduce the additional classification parameters, which prevents overfitting and improves generalization.

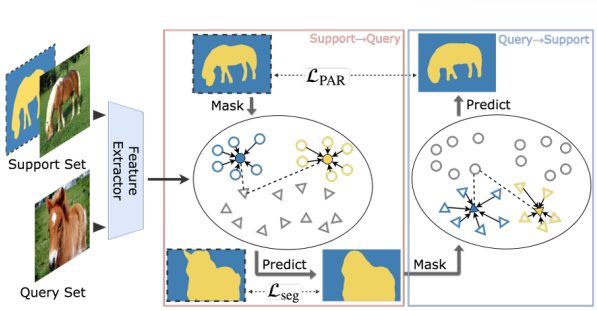

Figure 1: Overview of the proposed few-shot segmentation process. Segmentation masks with dashed borders indicate ground-truth annotations. First, we map support (circles) and query (triangles) images into their latent representation. Prototypes are then learnt from each class (blue for background, yellow for horse). Segmenatation over the query is then performed by matching its features the nearest prototype within the latent space. The 'forward' and 'reverse' few-shot segmentation passes are represented by the red and blue boxes respectively. This represents the proposed prototype alignment regularization (PAR).

The second approach is to perform **few-shot segmentation in the reverse direction** after the original pass. In other words, the query set and their respective masks are considered as a new support set, and is used to segment the previously used support set. This technique, formally called **Prototype Alignment Regularization (PAR)**, acts as a regularization technique to supervise the few-shot learning process, allowing the model to generate more consistent prototypes between the support and query sets, allowing for better generalization and performance.

1.2 Network Introduction

The paper introduces the **Prototype Alignment Network** (PANet) to tackle the challenges with few-shot segmentation and implement the approaches described beforehand. It consists of two main parts: the feature extractor and the prototype based segmentation model. First, PANet embeds foreground objects into different prototypes using the feature extractor. Then, each pixel of the query image is labelled by by referring to the class-specific prototypes nearest to its embedding representation. Furthermore, it uses the aforementioned prototype alignment regularization technique, finding that this, indeed, better aligns the prototypes generated from the query set aligns with those of the support set. This regularization technique is only present during model training.

Additionaly, it is noted that the models learns relatively well with weaker annotations such as bounding boxes and scribbles instead of a segmentation mask. The results of using these weaker annotations can be seen further in the experiments section.

In short, the paper claims that the network has the following core contributions:

- A simple but performant newtwork for few-shot segmentation (PANet). Using metric learning on latent-space prototypes, the model remains small when compared to other networks using a parametric classification architecture.
- A regularization technique (PAR) that allows better prototype alignment which results in better generalization and performance.
- A network that trains well even with weaker annotations.

2 Implementation methodology

2.1 Background Information

The process of replicating the experiments and metrics of the original paper began with understanding some of the underlying concepts of the network in greater detail.

Semantic Segmentation Figure 2 shows different types of image segmentation. Our model performs *semantic segmentation*, meaning it only separates objects within the image into different classes. It does not differentiate between objects of the same class. Semantic image segmentation can be seen in the top right image of Figure 2.

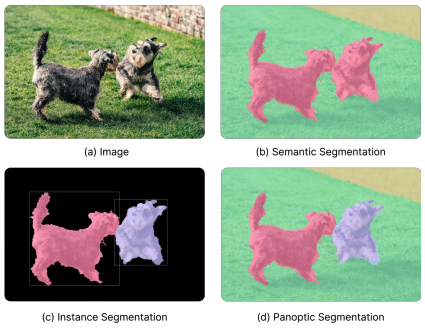

Figure 2: Image Segmentation Types

Few-shot Learning and Segmentation Few-shot learning utilizes knowledge from a set of examples (support set) to be able to better perform predictions on the target images. Few-shot learning and few-shot segmentation are closely related, with both incorporating support images to be able to better perform on target images. However, few-shot learning is associated with image-level classification, while few-shot segmentation is associated with pixel-level prediction.

2.2 Literature Review

Some of the first works that involved few-shot learning were *Matching Networks* [5] and *Prototypical Networks* [4], where classification was framed as a *metric learning problem*. Instead of classifying inputs, the network learns a distance function that minimizes the distances between items of the same class, while maximizing the distances of items accross different classes. *Matching Networks* used an attention mechanism to better align learned information from the support set, while *Prototypical Networks* built on top of this by representing each class as a prototype, a latent space embedding, to improve classification.

OSLSM [2] extended this concept to dense prediction tasks. It used prediction at the pixel level to perform semantic image segmentation using a few shot learning. *SG-One* [3] further improved this methodology by implementing similarity-based guidance during segmentation. This produced more robust similarity maps between the support and the query sets, allowing better performance on segmentation tasks. Finally, *PANet* [1] further builds on top of SG-One, introducing process separation and prototype alignment regularization (discussed beforehand) to improve segmentation performance even in challenging scenarios.

2.3 Challenges

The most notable challenges while trying to replicate PANet and its results are:

- Difficulties in Data Preparation
- Computational Resource Constraints

Difficulties in Data Preparation In addition to learning the core theoretical concepts of few-shot image segmentation learning, I also had to figure out how to pre-process and represent the ground-truth segmentation masks provided in the dataset. In addition to that, I had to figure out how to properly pass this data into the segmentation model.

Figure 3: Examples images with different annotation types.
Left contains *dense* annotations, middle contains *scribble* annotations, while right contains *bounding-box* annotations

In addition to the ground-truth masks provided by the dataset, the paper also uses weakly-annotated data such as scribbles and bounding boxes (see Figure 3). Having deal with three types of annotations complicates the already complex setup required with just one annotation.

Computational Resource Constraints The other challenge was dealing with the computational costs related to training and testing the network. Despite being a relatively simple network and having fewer parameters compared to its predecessor networks, the

Method	1-shot					5-shot					Δ	#Params
	split-1	split-2	split-3	split-4	Mean	split-1	split-2	split-3	split-4	Mean		
OSLSM	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	43.9	3.1	272.6M
co-FCN	36.7	50.6	44.9	32.4	41.1	37.5	50.0	44.1	33.9	41.4	0.3	34.2M
SG-One	40.2	58.4	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1	0.8	19.0M
PANet-init	30.8	40.7	38.3	31.4	35.3	41.6	52.7	51.6	40.8	46.7	11.4	14.7M
PANet	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7	7.6	14.7M
PANet	41.8	57.7	49.8	41.2	47.6	65.3	52.2	64.7	47.5	57.4	9.8	14.7M

Table 1: Results of 1-shot and 5-shot segmentation on the PASCAL VOC 2012 dataset using the mean-IoU metric. Values in gray represent values obtained from the original paper. Δ represents the difference between in the mean-IoU metric between 1-shot and 5-shot training.

size of the dataset and the number of various configuration - among other things - made the network very costly and time-consuming to train. This made it so that performing experiments to validate the correctness of the setup was expensive and time-consuming.

To be able to perform rapid prototyping, I consistently used Google Colab's A100 GPU to validate the experiment setups, and to train and test the models. In addition to the challenge of porting my whole setup to be able to run on Colab, I also had to carefully ensure that I used my credits only when necessary. In the end, I went through roughly 200 credits to be able to replicate the original paper's experimental results.

3 Experiments

3.1 Setup

Dataset We follow the evaluation algorithm from the original paper against the PASCAL VOC 2012 dataset. To accommodate bounding-box and scribble annotations, the dataset has been slightly modified. This is also described in the original paper.

The 20 categories in the PASCAL VOC dataset are evenly split into 4-splits, each containing 5 classes/categories. Models are trained on 3 of the 4 splits, with the last one being used to test the model in a cross-validation pattern. During testing, we have five runs in which we perform 1,000 iterations each, averaging the results.

Evaluation Metrics We use the **mean-IoU** and **binary-IoU** metrics to evaluate the model.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Mean-IoU for each foreground class and averages it over all other classes. It measures how well the model segments each class and averages the performance across all classes. **Binary-IoU** treats all object categories as one foreground class and averages the IoU of the foreground and background. Evaluates overall object detection whenever class distinction isn't critical. As such, we mainly use the mean-IoU because it correctly distinguishes between foreground categories, and therefore more accurately reflects the model's performance.

Hyperparameters Similar to the original paper, we use a VGG-16 network as the feature extractor initialized with pretrained weights on ILSVRC. Input images are resized to (417, 417) and augmented using random horizontal flipping. The model is trained end-to-end by SGD with the momentum of 0.9 for 30,000 iterations. The learning rate is initialized to 1e-3 and reduced by 0.1 every 10,000 iterations. The weight decay is 0.0005 and the batch size is 1.

As mentioned above, we will be training the model on 3 of the 4 splits. We will vary our training configuration by training in 1-shot or 5-shot (1 and 5 supporting images respectively), and by training with and without PAR. We will then evaluate the model on the remainder split, using the same amount of supporting images during training, with different types of annotations. Dense annotations are the 'default', while 'bounding-box' and 'scribble' are considered weaker annotations. We will examine and analyze the results of all the above configurations below.

3.2 Comparison with Other Networks

Baselines We set a baseline model which is initialized with weights pre-trained on ILSVRC but not further trained on PASCAL VOC, which is denoted as PANet-init. Additionally, metrics obtained from the original paper are colored in **gray** and are displayed here for easier comparison.

Method	1-shot	5-shot	Δ
FG-BG	55.0	-	-
Fine-tuning	55.1	55.6	0.5
OSLSM	61.3	61.5	0.2
co-FCN	60.1	60.2	0.1
PL	61.2	62.3	1.1
A-MCG	61.2	62.2	1.0
SG-One	63.9	65.9	2.0
PANet-init	58.9	65.7	6.8
PANet	66.5	70.7	4.2
PANet	66.0	71.1	5.1

Table 2: Table2:Results of 1-way 1-shot and 1-way 5-shot segmentation on PASCAL VOC dataset using binary-IoU metric.

PASCAL-5^i Table 1 compares the model with other models agains the PASCAL VOC 2012 dataset in 1-way 1-shot and 1-way 5-shot settings. As noted in the paper, we can see that the proposed

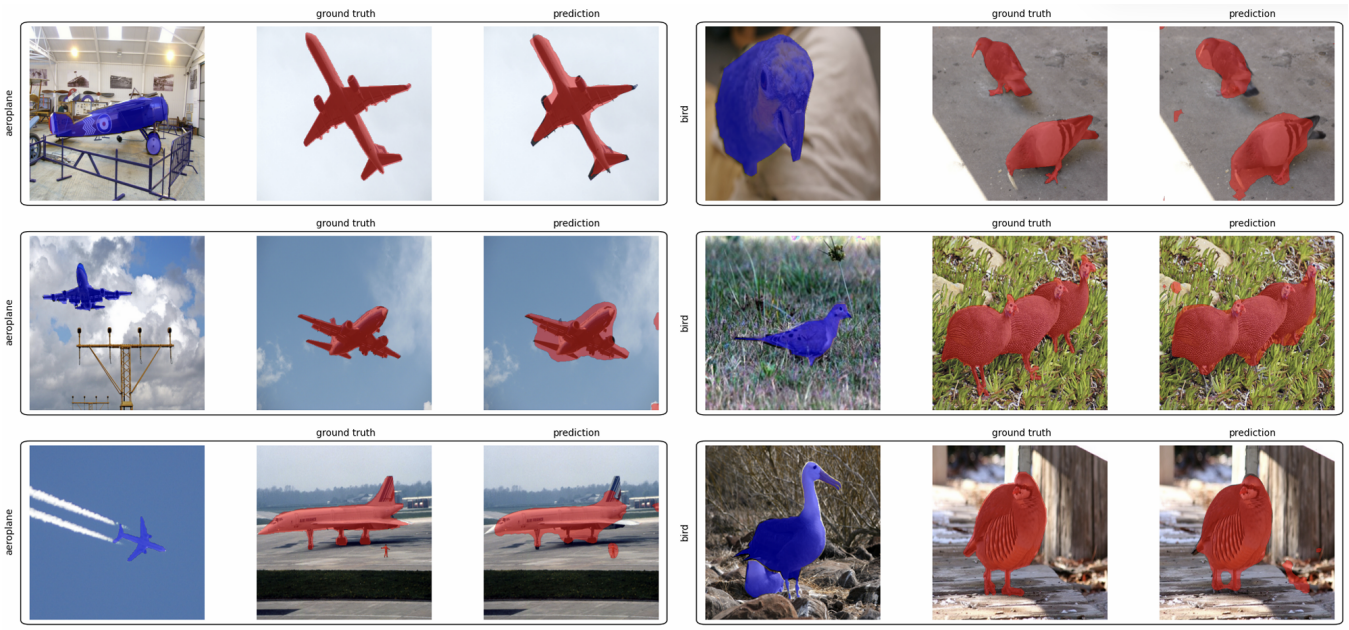

Figure 4: Qualitative results of our model in 1-way 5-shot segmentation on the PASCAL VOC 2012 dataset.

PANet network outperforms previous state of the art models with a Δ of 1.3% and 10.3% for 1- and 5-shot segmentation respectively when compared to the highest performing predecessor, SG-One. Furthermore, we can see that our network also uses less parameters compared to previous networks. Using the binary-IoU method, as shown in table B, out network also receives the highest performance.

Another thing to note is the large Δ between 1-shot and 5-shot learning. Our trained model was able to achieve a performance increase 9.8% when using 5-shot learning as compared to 1-shot. The highest of the predecessor networks being OSLSM, with only 3.1%. This implies that our network is more effectively able to make use of information from the support set, as compared to previous networks. The results of $PANet - init$ also supports this, with a Δ of 11.4%. Desipte not being fine-tuned, being able to draw more information from the support set allows it to rival the previous state-of-the-arts in 5-shot learning.

Qualitative results for 1-way segmentation are showin in Figure 4. Compared to other previous models, PANet does not use any sort of decoder structure or post-processing refinements. Despite that, we get good segmentation masks for unseen classes with only one annotated support image. This further supports the strong learning and generalization abilities of the network.

Another thing worth noting is that segmenation results can occurr with occasional patches far from our desired object (see the edges for the prediced image in Figure 3, row 2, col 1). The paper states that the probabilities for the mask are calculated individually at the pixel-level, but also states that small artefacts like these can be alleviated though some sort of post-processing.

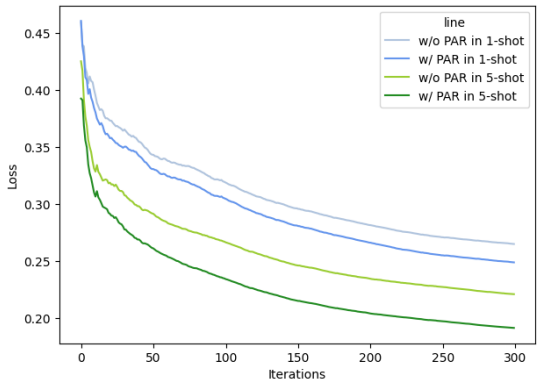

Figure 5: Loss during training for models with and without PAR, trained in 1-shot or 5-shot learning.

3.3 Analysis on PAR

The proposed PAR provides two main advantages over the 'normal' non-regularized networks. These advantages are also described in the original paper.

Improved network performance The network performs substantially better compared to the same network trained without PAR. We can see in Table 3 that our network had a 0.6% and 3.6% improvement in 1-shot and 5-shot learning respectively when compared to a network trained without PAR. This implies that, with PAR, our model is better able to utilize utilize the knowledge from the supporting set.

Convergence Speedup Another benefit with using PAR is an increased speed in model convergence, all else being kept equal. As indicated by the training loss curve in Figure 5, models trained with PAR converge faster than those without it for both 1-shot and 5-shot segmentation in 1-way.

Method	1-shot	5-shot
PANet w/o PAR	47.2	54.9
PANet	48.1	55.7
PANet w/o PAR	47.0	53.8
PANet	47.6	57.4

Table 3: Evaluation results of PANet trained with and without PAR on PASCAL VOC in mean-IoU metric.

3.4 Analysis on Weak Annotations

In the scenario where obtaining pixel-level annotations for each sample in the dataset may be unrealistic, it would be more reasonable to use weakly-annotated data. The original paper provided *scribble* and *bounding – box* annotations to replicate weakly-annotated data - which were generated from the ground-truth segmentation masks that we have been using beforehand up to this point.

Annotations	1-shot	5-shot
Dense	48.1	55.7
Scribble	44.8	54.6
Bounding box	45.1	52.8
Dense	47.6	57.4
Scribble	36.5	50.1
Bounding box	38.3	47.4

Table 4: Results of using different types of annotations in mean-IoU metric.

In Table 4, we can see that, for 1-shot learning, using a bounding box yielded better performance as comapred to using scribble annotations. However, in 5-shot learning, scribble annotations outperformed bounding-boxes. A potential explanation to this is that due to bounding boxes usually encompass more background information compared so scribble annotations - introducing noise. This noise may not be severely detrimental when having a single support image, but may significantly degrade model performance the more supports are used - as can be seen with 5-shot learning. Despite that, we see that our model is still capable of outperforming previous state of the art models even when forced to use weakly-annotated data.

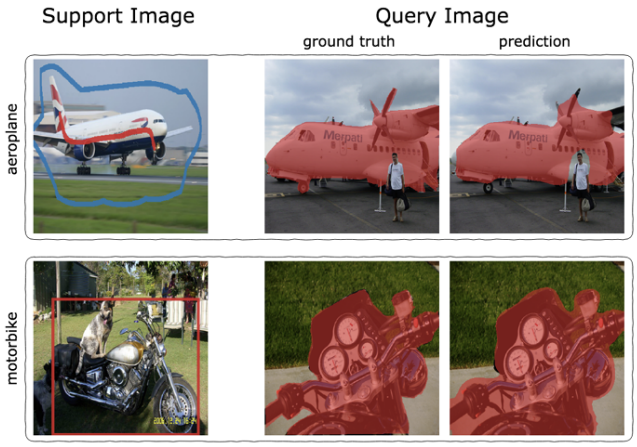

Figure 6: Qualitative results of our model on 1-way 1-shot segmentation using scribble and bounding box annotations. The scribbles are dilated for better visualization

4 Conclusion

The network proposed by the paper and replicated here is a model for semantic image segmentation. PAR enables it to produce robust prototypes whilst also speeding up model convergence. The separation of the feature extraction and segmentation processes allow for better prototypes as well as provide a mechanism for improving feature extraction by swapping out to a more performant network. With this, it is able to outperform previous state-of-the-art networks by a large margin without any decoder or post-processing step, whilst also being much smaller and simpler.

References

[1] K. Wang, J. H. Liew, Y. Zhong, D. Zhou, and J. Feng, *"PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment,"* arXiv preprint arXiv:1908.02490, Aug. 2019.

[2] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, *"One-Shot Learning for Semantic Segmentation,"* arXiv preprint arXiv:1709.03410, Sep. 2017.

[3] X. Zhang, Y. Wei, Y. Yang, and T. Huang, *"SG-One: Similarity Guidance Network for One-Shot Semantic Segmentation,"* arXiv preprint arXiv:1802.09406, Feb. 2018.

[4] J. Snell, K. Swersky, and R. S. Zemel, *"Prototypical Networks for Few-Shot Learning,"* arXiv preprint arXiv:1703.05175, Mar. 2017.

[5] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, *"Matching Networks for One Shot Learning,"* arXiv preprint arXiv:1606.04080, Jun. 2016.