A Pixel-Level Meta-Learner for Weakly Supervised Few-Shot Semantic Segmentation

Yuan-Hao Lee  Fu-En Yang  Yu-Chiang Frank Wang
Graduate Institute of Communication Engineering, National Taiwan University, Taiwan, R.O.C.
ASUS Intelligent Cloud Services, Taiwan, R.O.C.
{r07942074, f07942077, ycwang}@ntu.edu.tw

Abstract

Few-shot semantic segmentation addresses the learning task in which only few images with ground truth pixel-level labels are available for the novel classes of interest. One is typically required to collect a large mount of data (i.e., base classes) with such ground truth information, followed by meta-learning strategies to address the above learning task. When only image-level semantic labels can be observed during both training and testing, it is considered as an even more challenging task of weakly supervised few-shot semantic segmentation. To address this problem, we propose a novel meta-learning framework, which predicts pseudo pixel-level segmentation masks from a limited amount of data and their semantic labels. More importantly, our learning scheme further exploits the produced pixel-level information for query image inputs with segmentation guarantees. Thus, our proposed learning model can be viewed as a pixel-level meta-learner. Through extensive experiments on benchmark datasets, we show that our model achieves satisfactory performances under fully supervised settings, yet performs favorably against state-of-the-art methods under weakly supervised settings.

1. Introduction

Recent advances in deep convolutional neural networks (CNNs) [19] have significantly improved the performances of several computer vision tasks. Among them, semantic segmentation aims at predicting class labels for each pixel in an image, with applications ranging from autonomous driving to medical imaging. With the help of CNNs, state-of-the-art semantic segmentation models including FCN [25], SegNet [1] and DeepLabs [5, 6, 7, 8] have all achieved very promising results and been successfully applied to the above applications. However, these models are generally trained in a fully supervised manner, requiring a huge amount of training data with pixel-level annotation for each category of interest. This substantially limits the scalability and practicality of these models, as collecting densely labeled data would be very time-consuming.

Extended from the learning task of semantic segmentation and few-shot learning, few-shot semantic segmentation considers a more challenging setting in which only a few images are with ground-truth pixel-level labels for the (novel) classes of interest. In order to realize the learning of few-shot segmentation models, existing methods typically adopt meta-learning schemes [13, 30, 34], utilizing support and query images sampled from base categories (i.e., those with a sufficient amount of training data) for performing pixel-wise classification. Recent methods like AMP [38], PANet [46], FWB [27], PFENet [42] and ASGNet [20] choose to extract prototypes from support-set images using their ground truth masks, and expect such prototypes to sufficiently describe the associated semantic category. Nevertheless, these existing methods require pixel-level ground truth labels for each image of the base categories, and can-
not be easily extended to weakly supervised settings.

In order to alleviate the requirement of annotating pixel-level ground truth label information, we consider an even more challenging yet practical setting of weakly supervised few-shot segmentation, which requires only image-level labels collected for images in both base and novel categories, as depicted in Figure 1. As shown in the figure, we aim at inferring pixel-level labels from image-level labels in few-shot settings, followed by a meta-learning scheme enforced at pixel-level for segmentation purposes.

With this goal in mind, we propose a novel learning scheme in this paper, focusing on deriving a pixel-level meta learner for weakly supervised few-shot semantic segmentation. During the meta-training stage, our proposed learning framework observes only image-level labels and utilizes Classification Activation Maps (CAM) [59, 35] for prediction of pseudo pixel-level labels for each input image. While the class label embedding [2, 26] is exploited to bridge the gap between image and pixel-level information, our proposed method does not encounter any information leak since the class labels of both base and novel categories are not present in those of CAM.

Under the guidance of the produced pseudo pixel-level labels, we uniquely reinterpret the original problem as a pixel-wise few-shot classification task. That is, we view each pixel in support/query sets as individual samples, and turn the proposed model into a pixel-level meta learner for few-shot semantic segmentation. As confirmed later by our experiments, our model not only achieves satisfactory performances on standard fully supervised few-shot semantic segmentation tasks, it would perform favorably against several state-of-the-art approaches on benchmark datasets under weakly supervised settings.

The contributions of this work are summarized below:

- We address weakly supervised few-shot semantic segmentation, which requires only image-level labels during both training and testing.

- We bridge the gap between image and pixel-level labels using classification activation maps with no information leak, which allows prediction of pseudo pixel-level labels in weakly supervised settings.

- We uniquely design a pixel-level meta learner which enforces segmentation consistency across support and query set images during few-shot semantic segmentation. The proposed model can be realized in both fully supervised and weakly supervised settings.

2. Related Works

Semantic Segmentation. The task of semantic segmentation aims at performing pixel-level classification for each image. Most recent methods involve the use of deep convolutional neural networks (e.g., FCN [25]). Following works include SegNet [1], PSPNet [58], U-Net [32], DeepLabs [5, 6, 7, 8] and FastFCN [47]. A notable improvement is achieved by embedding features that contain multi-scale context information by applying dilated convolution [53, 5] or spatial pyramid pooling [58, 6]. These methods, however, are trained in a fully supervised manner and require a huge amount of pixel-level labels, which are time-consuming and laborious to obtain.

Weakly Supervised Semantic Segmentation. To alleviate the need for densely annotated ground truth data, some recent works address semantic segmentation in weak supervision, which utilize multiple-instance learning [43, 44], graph [55, 49, 28] and self-training based [50, 51, 56, 48] techniques. However, such weakly supervised methods cannot be easily extended to few-data or open-set scenarios.

Few-Shot Semantic Segmentation. Few-shot learning aims at learning models which would generalize to categories with only a limited amount of labeled data [12, 11]. Meta-learning [13, 30, 34] has been widely applied for this task, with the core idea of adapting the learning scheme from base to novel categories. For example, metric-based meta-learning algorithms such as ProtoNet [40] and RN [41] learn feature embeddings that exhibit proper distance metrics for classification and generalization. Few-shot semantic segmentation, on the other hand, aims at generalizing the ability of pixel-level classification across categories, while only a limited number of images are with ground truth pixel-level labels. OSLSM [36] is the first proposed method to tackle this problem, leveraging information learned from support-set images and outputs parameters for query image segmentation. PL [9] adopts metric learning methods [40] to extract prototypes of each semantic class, and measures their distances between feature maps of query images; CANet [54] adds an iterative optimization module to refine the predicted results; PFENet [42] generates additional prior masks to enrich the extracted features.

Moreover, PANet [46], FWB [27] and CRNet [23] propose to further leverage information from the support set by performing segmentation in the reversed direction (i.e., segmentation of the support set) for improved model learning. To exploit knowledge from the foreground objects, DAN [45] and SimPropNet [14] introduce attention mechanisms, PMMs [52] and ASGNet [20] employ multiple prototypes for a single category, and PPNet [24] proposes part-aware prototypes to capture fine-grained features. While some of the existing few-shot semantic segmentation methods present results in weak supervision settings (e.g., use of bounding boxes or scribbles [29, 46, 54] as guidance), they cannot produce satisfactory performance with only image-level annotation observed. More recently, [31, 4, 37] follow the few-shot setting using image-level supervision, but they
Still require collection of ground truth pixel-level masks for base-class images during meta-training. As depicted in Figure 2 and Table 1, to the best of our knowledge, we are the first to tackle few-shot semantic segmentation using only image-level annotations during both (meta) training and testing stages.

### 3. Proposed Method

#### 3.1. Notation and Problem Formulation

For the sake of completeness, we define the notations which will be used in this paper. The semantic classes are denoted as $\mathcal{C}$, which are split into two disjoint subsets: base categories $\mathcal{C}_{\text{base}}$ and novel categories $\mathcal{C}_{\text{novel}}$ (i.e., $\mathcal{C}_{\text{base}} \cup \mathcal{C}_{\text{novel}} = \mathcal{C}$ and $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \emptyset$). Note that the novel categories are with only a few samples available during training (typically less than 5 per class). For each input RGB image $I \in \mathbb{R}^{H \times W \times 3}$, we denote its image-level labels as $c \subseteq \mathcal{C}$, and the associated ground truth pixel-level semantic mask as $M \in \{c \cup \emptyset\}^{H \times W}$, where $\emptyset$ indicates the background pixels. For each training episode, we sample a support/query pair from the image dataset $\mathcal{D}_{\text{train}} = \{(S_j, Q_j)\}_{j=1}^{n_{\text{train}}}$ that contain the same set of base categories, while the testing episodes consist of those from $\mathcal{C}_{\text{novel}}$ (i.e., $\mathcal{D}_{\text{test}} = \{(S_j, Q_j)\}_{j=1}^{n_{\text{test}}}$).

In standard fully supervised $N$-way $K$-shot settings, each support set $S_j = \{(I_i^s, M_i^s)\}_{i=1}^{N \times K}$ contains $N \times K$ image/label example pairs ($K$ pairs from each of $N$ categories), and the query sets are denoted as $Q_j = \{(I_i^q, M_i^q)\}_{i=1}^{n_q}$, where $n_q$ is the number of query images in a single episode. As for the weakly supervised setting considered in this work, only image-level labels are available during training and testing, so that support/query sets are denoted as $S_j = \{(I_i^s, c_i^s)\}_{i=1}^{N \times K}$ and $Q_j = \{(I_i^q, c_i^q)\}_{i=1}^{n_q}$, respectively. Our proposed weakly supervised learning framework is able to produce pseudo masks of each image given only its class label (as depicted in Figure 3), followed by our pixel-level meta-learner for few-shot semantic segmentation (see Figure 4 for the complete framework).

#### 3.2. Pixel-Level Pseudo Label Generation

In the weakly supervised scenario, only the class labels are available for the image data during both training and inference. Thus, we first present a module that is able to generate pseudo pixel-level semantic masks, guiding the following segmentation process. As shown in Figure 3, this pseudo pixel-level label generation can be viewed as the process of semantics-oriented heatmap extraction. As
adopted in previous works like [18], Classification Activation Maps (CAM) [59] have been utilized to localize discriminative regions in images that are informative in classifying image-level labels. In order to produce the heatmap for the input image \( I \) as its pseudo pixel-level labels, we follow [46, 54] and apply a VGG-16 [39] network as our CAM backbone. It is worth noting that, the CAM backbone is pre-trained on a reduced subset of ImageNet [33], in which the images do not belong to either the base or novel categories in our segmentation task. This would minimize possible leakage of semantic information.

With CAM obtained, we extract per-class heatmaps \( T \) of the input image \( I \):

\[
\text{CAM}(I) = T = [T_1, T_2, \ldots, T_{N_{\text{CAM}}}],
\]

where \( T_k \in [0, 1]^{H \times W} \) denotes the heatmap of the \( k \)-th image class in CAM, whereas \( N_{\text{CAM}} \) is the total number of pre-training image categories.

With the above per-class heatmaps observed, we next leverage the word embedding features of each class label as intermediate representations, with the associated similarity indicating the weight for each of the \( N_{\text{CAM}} \) categories. More specifically, as depicted in Figure 3, we extract word embedding features of each class label in CAM and that of the input image. We perform pairwise comparisons between these features to obtain weighting factors \( w_k \) for each CAM category \( c_k \):

\[
w_k = d(\pi(c), \pi(c_k))^{-1},
\]

where \( \pi(\cdot) \) represents the word embedding function. We note that \( d(\cdot, \cdot) \) denotes the distance metric, and we use cosine similarity in our work. Thus, categories that are semantically similar with each other (e.g., \textit{goat/sheep}) would result in a higher weight, and vice versa for those that are dissimilar (e.g., \textit{goat/tree}). As a result, a weighted heatmap \( \bar{T} \) can be obtained by averaging the above \( N_{\text{CAM}} \) heatmaps, which is calculated as

\[
\bar{T} = w \cdot T^\top = \sum_{k=1}^{N_{\text{CAM}}} w_k T_k.
\]

As the CAM heatmaps identify only regions that are helpful in terms of classification, more detailed structural information such as edges and boundaries would not be well described. To this end, we impose a class-agnostic saliency map on the weighted heatmap as a gating mechanism, alleviating the presence of false-positive pixels in the generated pseudo labels. Here we note that, to comply with our weakly supervised setting, the saliency maps are obtained via a network pre-trained with only foreground/background information without any categorical supervision. In other words, minimized leakage of semantic information is also enforced at this stage.

3.3. Pixel-Level Meta-Learner for Few-Shot Semantic Segmentation

We now detail our proposed meta-learning scheme for few-shot semantic segmentation. Under the guidance of CAM, the pseudo pixel-level labels obtained in Section 3.2 tend to contain only partial discriminative areas, and may not sufficiently cover the entire foreground object. In order to learn few-shot segmentation models with only such information from pseudo labels, we present a unique pixel-level meta-learning framework, as illustrated in Figure 4. In our pixel-level meta-learner, we utilize DeepLabv3+ [8] as the feature extractor, together with the introduced pseudo label generation and learning modules for pixel-wise classifica-
Figure 4: Architecture of our pixel-level meta-learner for weakly supervised segmentation. During (meta) training, DeepLabv3+ extracts pixel-wise feature maps $F^s$ and $F^q$ for support and query inputs, with the produced pseudo pixel-level labels $\tilde{M}^s$ and $\tilde{M}^q$, respectively. Our meta-learner encoder $E$ transforms the above $F$ into a latent space, in which the pixel-level prototypical loss $\mathcal{L}_{\text{meta}}$ can be calculated based on $\tilde{M}$ for segmentation purposes. During inference (i.e., meta-testing), segmentation $\hat{M}^q$ for the query can be performed by pixel-wise $k$-NN classification using $F^s$ and $\tilde{M}^s$.

It is worth noting that, the DeepLabv3+ is pre-trained on categories not appeared in $C$, and this module remains fixed throughout the meta-learning process. In other words, no semantic information is leaked from its training stage. By removing the final classification layers of DeepLabv3+, we obtain a pixel-wise feature map $F^s \in \mathbb{R}^{H \times W \times d}$ from the support image $I^s$, where $(H, W)$ is the original image dimension and $d$ is the feature channel size.

Next, we label each spatial location in $F^s$ using the generated support pseudo mask $M^s$. By randomly sampling a fixed number of pixels from each background/foreground category, the resulting pixel-wise features are then collected into a set of support pixel features. It is worth noting that, we choose not to use all the labeled pixel features. This is not only because that such pseudo labels might not match the ground truth ones (although not available), this sampling mechanism also makes the meta-learning process more robust against weak labels.

Likewise, a pixel-wise feature map $F^q \in \mathbb{R}^{H \times W \times d}$ of the query image $I^q$ is also obtained via the same DeepLabv3+ feature extractor, resulting in a total of $H \times W$ $d$-dimensional query pixel features. All support and query pixel features are then jointly embedded into a latent space via the learnable encoder $E$. With the support pixel features and pseudo labels obtained, we define the prototypes $p_c$ for each associated category $c \in \{c^s \cup \emptyset\}$ as

$$p_c = \frac{\sum_{l} E(F^s_l)1[\tilde{M}^s_l = c]}{\sum_{l} 1[\tilde{M}^s_l = c]},$$

where $l$ iterates over each pixel, and $1[\cdot]$ is an indicator function which only outputs 1 when the condition holds.

Inspired by [40, 9, 42], we advance the prototypical loss $\mathcal{L}_{\text{meta}}$ as the objective during the meta-training stage, which is calculated by accumulating the distance between each query pixel (with pseudo labels) and its corresponding pixel-level prototype from the support sets:

$$\mathcal{L}_{\text{meta}} = -\sum_{c} \sum_{l} \exp(-d(E(F^q_l), p_c))1[\tilde{M}^q_l = c] \sum_{c} \sum_{l} 1[\tilde{M}^q_l = c].$$

During inference (i.e., meta-testing), the labels of the embedded query pixel features are determined by performing pixel-wise $k$-nearest neighbors classification, as depicted by the blue dotted arrows in Figure 4. The final predicted query semantic mask $\hat{M}^q$ is then compared with the ground truth $M^q$ for performance evaluation.
<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Split-0</th>
<th>Split-1</th>
<th>Split-2</th>
<th>Split-3</th>
<th>Mean</th>
<th>∆ Mean</th>
<th>Mean</th>
<th>∆ Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-way</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Co-att [37]</td>
<td>VGG-16</td>
<td>49.5</td>
<td>65.5</td>
<td>50.0</td>
<td>49.2</td>
<td>53.5</td>
<td>—</td>
<td>51.7</td>
<td>—</td>
</tr>
<tr>
<td>AMP [38]</td>
<td>VGG-16</td>
<td>41.9 / 10.6</td>
<td>50.2 / 14.1</td>
<td>46.7 / 7.6</td>
<td>34.7 / 10.9</td>
<td>43.4 / 10.8</td>
<td>32.6</td>
<td>46.9 / 14.7</td>
<td>32.2</td>
</tr>
<tr>
<td>PANet [46]</td>
<td>VGG-16</td>
<td>42.3 / 25.7</td>
<td>58.0 / 33.4</td>
<td>51.1 / 28.8</td>
<td>41.2 / 20.7</td>
<td>48.1 / 27.1</td>
<td>21.0</td>
<td>55.7 / 37.7</td>
<td>18.0</td>
</tr>
<tr>
<td>PFENet [42]</td>
<td>ResNet-50</td>
<td>61.7 / 33.4</td>
<td>69.5 / 42.5</td>
<td>55.4 / 43.6</td>
<td>56.3 / 39.9</td>
<td>60.8 / 39.9</td>
<td>20.9</td>
<td>61.9 / 44.8</td>
<td>17.1</td>
</tr>
<tr>
<td>Ours</td>
<td>VGG-16</td>
<td>38.3 / 36.5</td>
<td>57.6 / 51.7</td>
<td>54.0 / 45.9</td>
<td>40.1 / 35.6</td>
<td>47.5 / 45.5</td>
<td>5.1</td>
<td>50.6 / 45.5</td>
<td>5.1</td>
</tr>
</tbody>
</table>

| 2-way    |          |         |         |         |         |      |        |      |        |
| PANet [46] | VGG-16   | 45.1 / 24.5 | 45.1 / 33.6 | 45.1 / 26.3 | 45.1 / 20.3 | 45.1 / 26.2 | 18.9 | 53.1 / 36.6 | 16.5 |
| Ours     | VGG-16   | 36.5 / 31.5 | 51.8 / 46.7 | 48.5 / 41.4 | 38.9 / 31.2 | 43.9 / 37.7 | 6.2  | 49.3 / 43.0 | 6.3  |

(a) PASCAL-5i

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Split-0</th>
<th>Split-1</th>
<th>Split-2</th>
<th>Split-3</th>
<th>Mean</th>
<th>∆ Mean</th>
<th>Mean</th>
<th>∆ Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-way</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PANet [46]</td>
<td>VGG-16</td>
<td>20.9 / 12.7</td>
<td>20.9 / 8.7</td>
<td>20.9 / 5.9</td>
<td>20.9 / 4.8</td>
<td>20.9 / 8.0</td>
<td>12.9</td>
<td>29.7 / 13.9</td>
<td>15.8</td>
</tr>
<tr>
<td>Ours</td>
<td>VGG-16</td>
<td>26.0 / 24.2</td>
<td>14.5 / 12.9</td>
<td>20.0 / 17.0</td>
<td>18.3 / 14.0</td>
<td>19.7 / 17.0</td>
<td>2.7</td>
<td>27.0 / 17.5</td>
<td>9.5</td>
</tr>
</tbody>
</table>

| 2-way    |          |         |         |         |         |      |        |      |        |
| Ours     | VGG-16   | 18.2 / 17.4 | 12.2 / 9.5 | 9.1 / 10.4 | 6.5 / 7.1 | 11.5 / 11.1 | 0.4  | 14.8 / 11.9 | 2.9  |

(b) MS COCO

Table 2: Performance evaluation on (a) PASCAL-5i and (b) MS COCO in terms of mean-IoU (Mean) and performance difference ∆ due to change of settings. The numbers before and after '/' indicate results under fully and weakly supervised settings, respectively. Note that [37] considers a loosely weakly supervised setting and requires ground truth pixel-level masks during training, while [42] utilizes a stronger backbone (ResNet-50) compared to others (VGG-16).

4. Experiments

Datasets. We follow the evaluation protocol in [36] and conduct experiments on the PASCAL-5i dataset. It contains a total of 20 object categories from the PASCAL VOC 2012 [10] and the extended SDS [15] datasets, which are evenly divided into 4 splits (i = 0, 1, 2, 3). Additionally, we consider the MS COCO 2014 [21] dataset, which contains 80 object categories and thus is more challenging. Following the settings of [46], we divide MS COCO into 4 splits with 20 categories each. For both datasets, three of the splits are used as base (training) classes, with the remaining one as novel (testing) classes in each experiment.

Evaluation Metrics. We follow [36, 57, 46, 54, 27] and apply the mean intersection over union (mean-IoU) as the evaluation metric, which computes separate IoUs for each foreground category, and then averages them along with the background class. Following the protocol in [46], the mean-IoU in each evaluation is calculated by the average of 5 runs, with each run randomly sampling 1,000 episodes from the testing set.

Implementation Details. In our experiments, all images are normalized and reshaped to 129 × 129 pixels. For pseudo pixel-level label generation, we choose the VGG-16 [39] network as the CAM backbone, with weights pre-trained over a reduced subset of ILSVRC 2012 [33] with categories in $C_{base}$ and $C_{novel}$ removed. For semantic labels, we use the word embedding vectors pre-trained on Wikipedia using fastText [3]. We apply DSS [16] to extract saliency maps for the gating mechanism in Fig. 3, which is a model pre-trained over MSRA-B [22] without categorical supervision.

For the segmentation model, we use a DeepLabv3+ [8] pre-trained over irrelevant categories, which remains fixed throughout the meta-learning process. The encoder $E$ is a multilayer perceptron with two hidden layers, and the output dimension (for the latent space) is set to 64. All experiments are implemented by PyTorch, and are run using a single NVIDIA Titan RTX graphics card with 24GB of video memory.

4.1. Comparison with State-of-the-art Methods

Since no previous work was designed to address this weakly supervised few-shot segmentation task, we choose to implement modified versions of state-of-the-art supervised methods by replacing the ground truth masks for training with our generated pseudo labels (i.e., as introduced in Section 3.2). As depicted in Figure 4, our proposed framework can be trained in a fully supervised fashion (i.e., using ground truth pixel-level masks during training). Thus, we include this fully supervised version of our model as the performance upper bounds. We compare our results mainly with methods using the same backbone [37, 38, 46], with split-wise results not reported in the original paper.
the exception of [42] for cross-backbone analysis.

**PASCAL-5i.** We first compare the performances of different methods on PASCAL-5i, with results listed in Table 2a. In the first row of this table, we consider a loosely weakly supervised model of [37] (as illustrated in Figure 2b). For other methods (including ours) in this table, we present results under both fully and weakly supervised settings. While our fully supervised model achieved comparable performance with state-of-the-art methods in the standard 1-way 1-shot setting, our model reported a significant improvement over PANet [46] by 15.3% (42.4% v.s. 27.1%) in the weakly supervised setting. It is worth noting that PFENet [42] is trained using a stronger backbone (ResNet-50), while all other methods (including ours) utilize VGG-16. Nevertheless, our model still outperforms [42] by a considerable margin of 2.5% in the weakly supervised setting. We also observe that the performance drop between the two different settings of our model is significantly less than the others. That is, when only image-level labels (instead of ground truth pixel-level masks) are observed during both training and testing, both PFENet [42] and PANet [46] suffered from a >20% performance drop while only 5.1% was reported by our model.

**MS COCO.** As shown in Table 2b, despite the increased difficulty in few-shot segmentation on MS COCO, our model is able to achieve satisfactory performances under both fully and weakly supervised settings when comparing to [46]. Specifically, we only observe a 2.7% performance drop between the two settings on the 1-way 1-shot task, while that of [46] is 12.9%. The above quantitative results support the use of our propose framework for solving few-shot semantic segmentation, especially when only image-level labels can be observed during both training and testing (i.e., the weakly supervised setting).

**Multi-way Segmentation.** We now show that our model is applicable to the cases when there is more than one foreground object category in an image, which is more challenging since it requires more information to be learned in each episode. In the lower parts of Table 2, we list the performances of different methods under the 2-way setting (i.e., two types of foreground objects exist in an image). It is worth noting that, while most existing methods [36, 38, 54, 27, 17, 42] are designed to tackle only 1-way segmentation, they typically claim such extension can be realized by forward passing $K$ times with additional decision rules. On the contrary, our method can directly produce output labels of a multi-category image as the final classification by a $k$-NN search. As shown in Table 2a, our model performed favorably against previous methods by a margin of 11.5% on the PASCAL-5i 2-way 1-shot task.
the best of our knowledge, we are the first to report results of 2-way tasks for the MS COCO dataset, as shown in the last row of Table 2b.

4.2. Analysis of Our Proposed Method

Ablation Study. As our meta-learning process is mainly achieved by the episodic learning of encoder $E$, we now design a baseline version such that it directly predicts the output mask by a $k$-NN search on the pixel features encoded by the DeepLabv3+ backbone (i.e., without embedding into the latent space via $E$). As shown in the first row of Table 3, the results were severely degraded with a drop of up to 18%, if the model was not learned via the meta-learning objectives. This further confirms that, the promising performance achieved by our proposed method is not a direct result of using particular strong backbones. Instead, it leverages spatial and structural details captured by the backbone, upon which semantic and category-wise information is reinforced via the meta-learning process.

Additionally, we provide results that are trained using pseudo masks generated without saliency gating (as mentioned in the last step of Section 3.2). As shown in the second row of Table 3, a slight decrease in mean-IoU (less than 5%) was observed, while still outperforming the baseline version by a large margin. Thus, the use of saliency gating as the post-processing step for pseudo pixel-level masks would be preferable but not critical.

Qualitative Analysis. For visual comparisons, we present qualitative results of 1-way 1-shot segmentation on PASCAL-5i dataset in Figure 5. As detailed in Section 3.1, our model is realized in the weakly supervised setting and does not observe ground truth masks of support images during training (i.e., column 1 in Figure 5). Instead, given each image and its image-level label (i.e., class name), we first generate its pseudo mask, which serves as the guidance for our meta-learning framework. As evident in column 2 of Figure 5, our generated pseudo masks do not cover the entire foreground object, but instead contain only discriminative regions that are informative in classifying image-level labels (e.g., muzzle of a horse, or wheels and pedals of a motorcycle). Nevertheless, from this figure, we see that our proposed model is able to predict masks for query images with satisfactory performances (e.g., columns 4 and 6 in Figure 5). It is also worth noting that no post processing is performed on our predicted results.

From the split-wise mean-IoUs listed in Table 2, we see that some data splits would generally suffer from performance drops across different models including ours. In Figure 6, we show failure segmentation example results by our weakly supervised model and the fully supervised PFENet [42]. As evident in this figure, the ground truth images of such categories generally contain small foreground regions (e.g., bottle), or those that cannot be easily distinguished from the background (e.g., chair). For such image categories, training with few-shot samples would not be expected to address semantic segmentation tasks well. This would be the limitation of developing solutions for few-shot semantic segmentation.

5. Conclusion

In this paper, we proposed a unique learning framework for few-shot semantic segmentation in a weakly supervised manner, which only observes image-level labels during both training and testing. Under such weak supervision, our proposed model serves as a pixel-level meta-learner, which produces pseudo segmentation masks for guiding pixel-wise classification in a meta-learning fashion. Our experiments confirmed the effectiveness of our model, which achieved satisfactory results on two benchmark datasets under fully supervised settings, while surpassing the state-of-the-art methods in weakly supervised settings. Finally, we point out the challenge and limitation of the task of few-shot semantic segmentation.

Acknowledgement This work is supported in part by the Ministry of Science and Technology of Taiwan under grants MOST 110-2634-F-002-036 and 110-2221-E-002-121.

Table 3: Ablation study of our model on PASCAL-5i in mean-IoUs. Ours w/o $E$ denotes our framework without meta-learner encoder $E$, while w/o Saliency indicates our model without applying saliency gating to process the generated pseudo masks.
References


[37] Mennatullah Siam, Naren Doraiswamy, Boris N Oreshkin, Hengshuai Yao, and Martin Jagersand. Weakly supervised few-shot object segmentation using co-attention with visual and semantic inputs. In International Joint Conference on Artificial Intelligence (IJCAI), 2020. 2, 3, 6, 7
[50] Jia Xu, Alexander G Schwing, and Raquel Urtasun. Tell me what you see and i will show you where it is. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3190–3197, 2014. 2


