Pattern Recognition Letters 147 (2021) 93–99

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

SML: Semantic meta-learning for few-shot semantic segmentation☆☆

Ayyappa Kumar Pambala, Titir Dutta, Soma Biswas*

Department of Electrical Engineering, Indian Institute of Science, Bangalore, Karnataka 560012, India

ARTICLE INFO

Article history:
Received 30 November 2020
Revised 7 March 2021
Accepted 28 March 2021
Available online 20 April 2021

Keywords:
Few-shot learning
Semantic segmentation
Attributes

ABSTRACT

The significant amount of training data required for training Convolutional Neural Networks has become a bottleneck for applications like semantic segmentation. Few-shot semantic segmentation algorithms address this problem, with an aim to achieve good performance in the low-data regime, with few annotated training images. Recent approaches based on class-prototypes computed from available training data have achieved immense success for this task. In this work, we propose a novel meta-learning framework, Semantic Meta-Learning (SML), which incorporates class level semantic descriptions in the generated prototypes for this problem. In addition, we propose to use the well-established technique, ridge regression, to not only bring in the class-level semantic information, but also to effectively utilise the information available from multiple images present in the training data for prototype computation. This has a simple closed-form solution, and thus can be implemented easily and efficiently. Extensive experiments on the benchmark PASCAL-5i dataset under different experimental settings demonstrate the effectiveness of the proposed framework.

1. Introduction

Image segmentation is one of the fundamental problems in the field of computer vision. Traditional supervised segmentation methods [1–3] give impressive results when large amounts of annotated data are available. However, this requirement of labeled training data for segmentation task is quite difficult, since annotating each and every pixel for huge amount of image data is highly expensive and cumbersome. On the other hand, humans can identify any novel concept very easily even with very few examples of the same. Few-shot semantic segmentation [4–6] addresses this problem, by working in the very low data regime, utilizing few annotated images from each class.

Meta-learning or learning-to-learn approaches have achieved very good performance for the problem of few shot learning [7,8], and also for the segmentation application [5,9–11]. Training meta-learning algorithms constitute two stages of learning: (1) base-learner, which learns to predict an individual task at the episode level and (2) meta-learner, which learns to generalize by learning across a large number of training tasks/episodes. Significant amount of research has been done along these lines, but recently, the approaches based on computing class representatives or class prototypes have been very successful [4,12].

In this work, we propose a novel meta-learning framework, termed as Semantic Meta-Learning or **SML**, to address the few-shot semantic segmentation task by utilizing class-specific semantic information. The proposed SML approach is based on computing the prototypes corresponding to each class in the training data. But, in contrast to other prototype-based meta-learning approaches in literature [4,9], SML does not compute the class prototypes as the average representation of all the visual feature embeddings. Instead, they are learnt by incorporating the semantic knowledge of the particular class (obtained automatically from the class names) into the visual information obtained from the images by the base-learner. In addition, we propose to utilise the visual feature embeddings obtained from multiple training images of the same class individually while computing the class prototypes, instead of averaging them. Both these steps effectively bring the visual embeddings of the same class images closer to one another, and also maintain semantically meaningful intra-class distances between the class prototypes, even when few training images per class are available. Inspired by the seminal work in Bertinetto et al. [13], these objectives are achieved through learning a linear function between the visual feature embeddings and the semantic information or attributes using standard ridge regression method. Thus, in the proposed SML framework, this linear function has a closed-form solution to compute the prototypes, which makes the computation very efficient. Extensive experiments on the benchmark PASCAL-5i dataset with different experimental settings show that the proposed SML framework is effective for few-shot semantic segmentation task. SML also performs favorably with respect to

☆ Handle by Associate Editor Jie Zou.

* Corresponding author.
E-mail address: somabiswas@iisc.ac.in (S. Biswas).

https://doi.org/10.1016/j.patrec.2021.03.036
0167-8655/

the state-of-the-art, even for weaker annotations. Thus, the contributions of this work are as follows:

(1) We propose a novel end-to-end meta-learing framework, SML, which can effectively integrate the attribute information and the visual feature embeddings in the meta-learner to generalize to new classes during testing.
(2) SML also utilises multiple images in the training data more effectively for computing better class-prototypes.
(3) Extensive evaluation on the PASCAL-5i dataset shows that the proposed SML framework performs better or comparable to the state-of-the-art.

2. Related work

Few-shot learning is a very active area of research in the field of computer vision, and several approaches have been proposed. In this section, we give pointers to the relevant meta-learning based approaches in literature which addresses few-shot classification and semantic segmentation.

2.1. Few-shot learning

Meta-learning based few-shot learning approaches can be broadly divided into (a) *metric learning* based approaches like [7,8], (b) *optimization* based approaches like [13–15], etc. The goal of these approaches is to leverage the support set images to learn to classify the images in the query set. Matching network [7] learns using soft weighted nearest neighbour scores obtained from the support images. In prototypical network [8], support class means are evaluated as the class-prototypes and the classification is performed on the query examples using Euclidean distance measure. Relation network [16] learns to find relations between small number of images at each episode, which contains both support and query images. Recently, in differentiable solver [13], the network is learned using ridge regression. Though the proposed method is inspired from [13], we propose to additionally fuse the semantic information of the classes for the goal of few-shot semantic segmentation.

2.2. Few-shot segmentation

Some of the older approaches in the literature for few-shot segmentation follow the strategy in Vinyals et al. [7] to learn the network parameters from the support images and perform pixel wise classification task on the query images. Co-FCN ([10]) learns to segment the query images by fusing feature information of support images. A metric learning based prototype learning is used in Dong and Xing [9] for few-shot segmentation. Weight imprinting mechanism of new classes using adaptive masked proxies is used in Siam and Oreshkin [6]. CANet [5] uses iterative optimization solution and MetaSegNet [12] uses a differentiable optimization solver for the segmentation task. Recently, prototypical network [8] for few-shot classification is adapted in Wang et al. [4], Dong and Xing [9] to perform the segmentation task. PANet [4] uses masked average pooling to obtain the prototypes, and also proposes a reverse alignment regularizer to learn the prototypes using a swapped-setting of support and query images. PPNet [17] and PMM [18] use multiple class prototypes with unlabelled data for the segmentation task.

Our method is inspired from [4] to obtain better class prototypes by incorporating the semantic class-descriptions (using the class names), into the base-learners to learn the segmentation task. In essence, the base-learner exploits both visual information and semantic/attribute knowledge to obtain better prototypes and also to efficiently cluster the features of images of the same class.

3. Problem definition

Here, we discuss the proposed SML framework for the task of few-shot semantic segmentation. First, we describe the notations and problem statement. Let the training data be denoted as $\mathcal{D}_{base} = \{\mathbf{I}, \mathbf{M}\}$, where, $\mathbf{M}$ represents the ground truth mask corresponding to the input image $\mathbf{I}$. Let $\mathcal{C}_{base}$ be the set of base classes for which the annotated data is available. During testing, the goal is to segment images from a set of novel classes $\mathcal{C}_{novel}$, given only few images from each class. Let the number of images available per class during testing be denoted by K, and K is usually less than or equal to 5. The testing classes do not overlap with the training ones, i.e. $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \phi$.

To address this task, a meta-learning based approach is adopted, where we attempt to imitate the testing scenario as closely as possible during training. The entire training process is split into sub-tasks or episodes, referred to as episode-based training [7,8]. Each task or episode $\mathcal{E} = (\mathcal{S}, \mathcal{Q})$ consists of a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. To emulate the testing scenario, each episode is constructed to be a C-way, K-shot segmentation task, i.e., given a support set of K images and their ground truth masks from each of C classes, the goal is to segment the images in the query set containing objects from one of these C classes. Thus, $\mathcal{S}$ contains randomly selected C classes from the set $\mathcal{C}_{base}$ with K images from each class, i.e., $\mathcal{S} = \bigcup_{c=1}^{C}\{\mathbf{I}_{i,c}^{s}, \mathbf{M}_{i,c}^{s}\}_{i=1}^{K}$. Similarly, the query set $\mathcal{Q} = \{\mathbf{I}_{j}^{q}, \mathbf{M}_{j}^{q}\}_{j=1}^{N_q}$ contains N_q number of query images from the classes present in its support set. The set of images in the support and query sets in every episode are strictly non-overlapping. The network is trained by sampling several episodes in a meta-learning fashion.

In this work, segmentation is performed by computing semantically meaningful prototypes for each class. Given the class names present in the support (or query) set, any existing pre-trained language model, such as FastText [19] or Word2Vec [20] can be utilised to obtain the semantic or attribute vectors automatically corresponding to those object classes. Let $\mathbf{a}_c \in \mathbb{R}^{d_a}$ denote the attribute vector corresponding to the cth class. All image pixels in the support and query set which belong to the cth class will have the same semantic representation. Note that, this information can be easily obtained given just the class names in $\mathcal{D}_{base}$, and does not require any additional information. Attributes are widely used in applications like ZSL [21,22], but it has been relatively less explored for few-shot learning.

3.1. Main idea of SML framework

An illustration of the SML framework for few-shot segmentation is shown in Fig. 1. The proposed framework is learnt in a meta-learning fashion consisting of several episodes, and Fig. 1 illustrates the base learner for 1-way, 2-shot segmentation scenario. It has two main modules: (1) feature-extractor module and (2) attribute-injector module. For each episode, given the support images, their visual features are extracted using the feature extractor network. Their masks are used to obtain the foreground and background regions, which are in turn used to compute the average foreground (ϕ_{cow}) and background (ϕ_{bg}) feature embeddings for each support image by average mask pooling. The class names in the support set is used by the attribute injector module to incorporate the semantic information into the base learner. Specifically, this is achieved by using a linear function (h_W) to effectively combine the information from the attribute vectors and the visual features, towards the goal to obtain better class prototypes ($h_W(.)$). These final prototypes of the class ($h_W(\mathbf{a}_{cow})$) and the background ($h_W(\mathbf{a}_{bg})$) are used to perform segmentation on the images in the query set. The loss computed between the query prediction and its ground

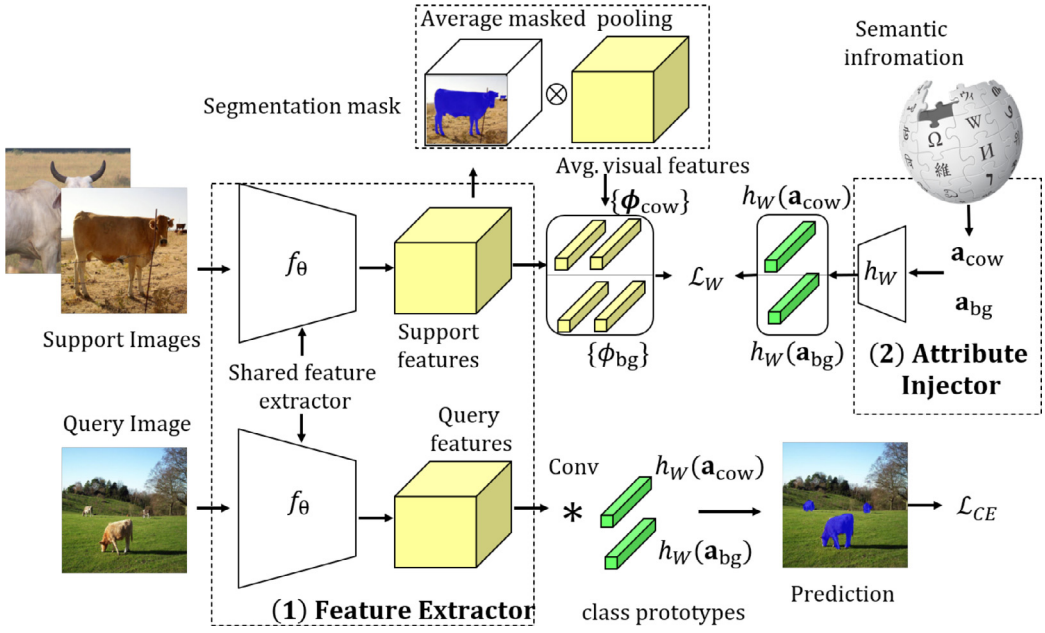


Fig. 1. Illustration of the proposed Semantic Meta-Learning (SML) framework for few-shot semantic segmentation. This illustration is for a single episode for 1-way 2-shot segmentation task. The base-learner learns the class-prototypes by integrating the visual information from the images (using average-masked features ϕ) and attribute information $\mathbf{a}$, which is accomplished using a linear function h_W. These class-prototypes are used to obtain the final prediction for the query set.

truth mask is used to update the base-learner, which constitute the feature extractor and the linear function h_W. The meta-learner is learnt over several training episodes to generalize to segment novel classes with few examples per class.

The main difference between the proposed SML framework and other prototype-based few-shot segmentation approaches like [4,9], etc. is the use of attribute information. We feel that semantic information is extremely beneficial, specially in the low-data regime. Instead of relying solely on the visual information from the images, SML effectively combines it with the semantic information for obtaining improved class prototypes. In addition, the features from multiple support images are not averaged, rather they individually contribute to the prototype computation. This helps to bring the features from the same class closer to one another, thus effectively utilising all the additional support images. The closed form solution of the linear function makes the learning very efficient. Extensive experiments and analysis in Section 5 justifies the effectiveness of this framework. Now, we will discuss the loss functions and the meta-learning based training.

4. Proposed SML framework

Here, we will describe the different components of the SML framework. The proposed meta-learning framework learns over a collection of episodes, and thus gets trained with data of all classes ($\mathcal{C}_{base}$) in the training set. But, the *base learner* learns over a single episode with data from randomly selected C-classes. The base learner consists of a feature extractor and an attribute injector module as described below.

4.1. Feature-extractor module

Given an input image $\mathbf{I}$ from support or query, the feature extraction module f_θ extracts its 3-d feature representation $f_\theta(\mathbf{I}) \in \mathbb{R}^{h\times w\times d}$. Here, $f_\theta(\mathbf{I})$ is the upsampled feature map and has the same height (h) and width (w) as the original image and and d denotes the number of channels of the final convolutional layer. The set of learnable parameters in the feature extractor module is denoted as θ. This 3-d visual feature representation of an image can be considered as the d-dimensional embedding of each of the image pixels, which either belong to the class of interest or the background.

As per the standard meta-learning set-up [23], in each episode, the base learner first processes the data in the support set $\mathcal{S}$, and then computes the losses on the query set $\mathcal{Q}$. Given the support set $\mathcal{S} = \bigcup_{c=1}^{C} \{\mathbf{I}^s_{i,c}, \mathbf{M}^s_{i,c}\}_{i=1}^{K}$, for every image, its 3-d feature representation $f_\theta(\mathbf{I}^s_{i,c})$ is computed using the feature extractor module. This obviously contains the feature representation of both the object (or, the foreground) and the background in the image. The foreground and background features can be separated using the mask $\mathbf{M}^s_{i,c}$ provided with the input image. As in several other works [4,6], we also use average mask-pooling operation to obtain the foreground (or, background) embedding as follows:

$$\phi^s_{i,c} = \frac{1}{|\mathcal{X}|}\sum_{\mathcal{X}} f_\theta(\mathbf{I}^s_{i,c}) \odot \mathbf{1}_{(\mathbf{M}^s_{i,c}==c)}, \text{ for foreground}$$

$$\phi^s_{i,bg} = \frac{1}{|\mathcal{Y}|}\sum_{\mathcal{Y}} f_\theta(\mathbf{I}^s_{i,c}) \odot \mathbf{1}_{(\mathbf{M}^s_{i,c}\neq c)}, \text{ for background} \quad (1)$$

Here, $\odot$ represents the point-wise multiplication along the dimension of d. $\mathcal{X}$ and $\mathcal{Y}$ denote the respective sets of spatial locations in the image $\mathbf{I}^s_{i,c}$ for which the corresponding indicator functions are activated. Thus, $\phi^s_{i,c}$ and $\phi^s_{i,bg} \in \mathbb{R}^d$ can be considered as the mask-pooling of foreground and background features separately in the visual embedding space evaluated as the mean over their spatial spread in the image. We normalize the features, $\phi^s_{i,c} = \frac{\phi^s_{i,c}}{||\phi^s_{i,c}||_2}$ and $\phi_{i,bg} = \frac{\phi^s_{i,bg}}{||\phi^s_{i,bg}||_2}$ in our experiments.

4.2. Attribute-injector module

In some of the recent successful approaches like [4], the image embeddings are solely utilised to obtain the class-representatives or prototypes. We propose to augment this information with the semantic knowledge for better generalization to unseen classes

with few examples during testing. Here, the semantic information is given by the attribute vectors, $\mathbf{a}_c \in \mathbb{R}^{d_a}$, which are the class-name embeddings for the cth class. These attributes can be automatically extracted from a pre-trained *FastText* [19] or *Word2Vec* [20] language model. Here, d_a denotes the attribute dimension. Given these semantic attributes of the classes in the training data, the attribute injector module incorporates this information to compute better class prototypes for the segmentation task. Here, we use a linear function for the attribute injector as:

$$h_W(\mathbf{a}_c) = \mathbf{W}\mathbf{a_c} \tag{2}$$

where, $\mathbf{W} \in \mathbb{R}^{d \times d_a}$ is the weight matrix containing all the trainable parameters. Similarly, we also obtain the latent space representation of the background of the images as $h_W(\mathbf{a}_{bg}) = \mathbf{W}\mathbf{a}_{bg}$; where $\mathbf{a}_{bg}$ is the FastText or Word2Vec embedding of the word *background*, and thus is the same for all images.

In order to effectively incorporate the semantic information with the embeddings obtained from the images using Eq. (1) in the latent space, we use the standard ridge regression [13,24]. Thus, the loss function for learning $\mathbf{W}$ can be expressed as

$$\mathcal{L}_W = ||\Phi - \mathbf{W}\mathbf{A}||_2^2 + \lambda||\mathbf{W}||_2^2 \tag{3}$$

where, $\Phi = \left[\{\phi^s_{i,c}|\phi_{i,bg}\}_{i=1}^K\right]_{c=1}^C \in \mathbb{R}^{d \times 2|\mathcal{S}|}$ and $\mathbf{A} = \left[\{\mathbf{a}^s_{i,c}|\mathbf{a}^s_{i,bg}\}_{i=1}^K\right]_{c=1}^C \in \mathbb{R}^{d_a \times 2|\mathcal{S}|}$. λ is a learnable hyper-parameter which is set experimentally to balance the L2-regularizer on the parameters $\mathbf{W}$. The optimum set of parameters of the attribute injector module $\mathbf{W}$ is obtained by minimizing this loss function which has a closed-form solution given by

$$\mathbf{W} = \Phi\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I}_{d_a})^{-1} \tag{4}$$

where, $\mathbf{I}_{d_a}$ is an identity matrix of dimension d_a.

Learning the base-learner parameters: The goal is to leverage the query set images and its corresponding ground truth masks to learn the parameters of the feature extractor module and the linear function parameters $\mathbf{W}$, such that during testing, semantic segmentation can be performed using only few images per class. In each episode, given the support images with the corresponding masks from C-classes, Φ and $\mathbf{A}$ can be computed, from which the linear function parameters $\mathbf{W}$ are learnt as discussed before. Note that, for each class c, the images corresponding to this class will have different feature representations ($\phi^s_{i,c}$) because of the intra-class variability, but they will all have the same semantic representation corresponding to the class name $\mathbf{a}_c$ for both support and query. This unique semantic representation of the cth class is appropriately combined with the image information to get a better class prototype given by $h_W(\mathbf{a}_c) = \mathbf{W}\mathbf{a}_c$.

Now, given an image from the query set $\mathbf{I}^q_j$, we first compute its visual representation by passing it through the feature extractor module $f_\theta(\mathbf{I}^q_j)$ and the feature corresponding to each pixel is normalized. Next, the normalized feature representation $f_\theta(\mathbf{I}^q_j)$ is convolved with the normalized class-prototypes for all the C-classes selected in that episode as follows:

$$\mathbf{S}^q_{j;c} = f_\theta(\mathbf{I}^q_j) * h_W(\mathbf{a}_c), \text{ for } c = \{1, \ldots, C\} \cup \{bg\} \tag{5}$$

Here, $*$ stands for convolution operation and $h_W(\mathbf{a}_c) = \mathbf{W}\mathbf{a}_c$ is the class-prototype of the cth class. We have used the same notations for the normalized representations for simplicity. Thus, $\mathbf{S}^q_{j;c} \in \mathbb{R}^{h \times w}$ denotes the 2-d mask of the query image with the class-similarities, which is finally used for classification. Here, in addition to the foreground classes, similarity with the background class prototype bg is also computed. To segment the query image, we perform pixel-wise classification of the query feature $\mathbf{S}^q_{j;c}$. For this, we compute the logit-score for the pixel at location (m, n) as the probability of the evaluated similarity feature $\mathbf{S}^q_{j;c}$ to belong to class c as

$$p(\mathbf{S}^q_{j;c}(m,n)) = \frac{exp(\alpha\mathbf{S}^q_{j;c}(m,n) + \beta)}{\sum_{c \in \{1,\ldots,C\} \cup \{bg\}} exp(\alpha\mathbf{S}^q_{j;c}(m,n) + \beta)} \tag{6}$$

where α, β are the scaling and bias parameters, respectively. We learn the feature extractor module of the base learner by minimizing the cross-entropy loss function

$$\mathcal{L}_{CE} = \sum_{(m,n) \in \mathbf{I}^q_j} -\log p(\mathbf{S}^q_{j;c}(m,n)) \tag{7}$$

Inspired by Wang et al. [4], we also interchange the samples in support $\mathcal{S}$ and query $\mathcal{Q}$ set to enhance the segmentation performance. Thus, we learn another set of parameters $\tilde{\mathbf{W}}$ on the newly-constructed $\mathcal{S}$ and $\mathcal{Q}$ using (4). Following similar steps as shown above, we compute the reverse alignment loss $\mathcal{L}_R$ as in Wang et al. [4] which is given by,

$$\mathcal{L}_R = \sum_{(m,n) \in \mathbf{I}^s} -\log p(\mathbf{S}^s_c(m,n)) \tag{8}$$

Note that subscripts are removed for the similarity predictions and for the images to avoid clutter. Thus, the final objective used to learn the feature extractor is given by

$$\min_\theta \sum_{(\mathcal{S},\mathcal{Q})} \left(\mathcal{L}_{CE} + \mathcal{L}_R\right) \tag{9}$$

This completes the learning of the base-learner for a single episode. To summarize, in each episode, first, the attribute injector module parameters $\mathbf{W}$ are learnt to generate class prototypes using the previously learnt base-learner, which are used to further fine-tune the feature extractor parameters θ by computing the pixel wise cross-entropy loss. Using several such training episodes, the meta-learner learns to generalize and segment images containing novel objects during testing. The algorithm used for training SML is given in Algorithm 1 .[1]

Algorithm 1: Algorithm for training *SML*.

Input: $\mathcal{D}_{base} = \{\mathbf{I}, \mathbf{M}\}$, where, $\mathbf{I}$: set of input images and $\mathbf{M}$: set of corresponding ground truth masks;
$\mathbf{a}$: attributes;
β: learning rate;
E: total number of episodes for which the meta-learner is trained, where each episode is given by $\{\mathcal{S}_e, \mathcal{Q}_e\}_{e=1}^E \in \mathcal{D}_{base}$

Output: θ^*: Learnt parameters of the feature extractor.
Initialize:
Pre-trained (from *Imagenet*) weights of the feature extractor f_θ.
for $e = 1, 2, \ldots, E$ **do**
  Compute prototype matrix Φ as in (3).
  Load attribute matrix $\mathbf{A}$.
  % Learning base-learner
  Learn $\mathbf{W} = \Phi\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I}_{d_a})^{-1}$ as in (4).
  Compute new class prototypes: $h_{\mathbf{W}}(a_c) = \mathbf{W}a_c$.
  Predict segmentation mask on the $\mathcal{Q}$.
  Compute segmentation loss $\mathcal{L}_{CE}$ as in (7).
  Interchange $\mathcal{S}$ and $\mathcal{Q}$; and compute reverse alignment loss $\mathcal{L}_R$as (8).
  Compute gradient: $g_\theta = \nabla_\theta(\mathcal{L}_{CE} + \mathcal{L}_R)$.
  Update: $\theta \leftarrow \theta - \beta$ Adam(θ, g_θ).
end

[1] Code is available at: https://github.com/ayyappa1993/SML.

Prediction: Once the meta-learner is trained using all the data over a number of training episodes, it can be used to perform segmentation on novel images in an episode, $\mathcal{E} = (\mathcal{S}, \mathcal{Q})$ which contains images from $\mathcal{C}_{novel}$. As in training, first, $\mathbf{W}$ is computed using the visual image features of the support set and the attributes of the classes present in the testing set. Using this, the class prototypes are obtained. Finally, the prediction on the features extracted from the query image $\mathbf{I}_j^q \in \mathcal{Q}$ is performed as,

$$\hat{\mathbf{S}}_j^q(m,n) = \underset{c\in\{1,\ldots,C\}\cup\{bg\}}{\arg\max} \ \alpha \mathbf{S}_{j,c}^q(m,n) + \beta. \tag{10}$$

5. Experiments

In this section, we describe the experiments performed to evaluate the effectiveness of the proposed SML framework.

5.1. Dataset and evaluation metric

We use PASCAL-5i [25] and COCO-20i [29] datasets to evaluate the model performance.

PASCAL-5i is derived from PASCAL VOC 2012 [30] and SBD [31]. This dataset contains 20 classes. All the 20-classes are divided into 4 splits, with 5-categories per split. As is the standard practice [25], the proposed SML model is trained on 3 splits and evaluated on the 4th split. COCO-20i is derived from MS-COCO dataset [29] which contains 80 classes divided into 4 splits following [28].

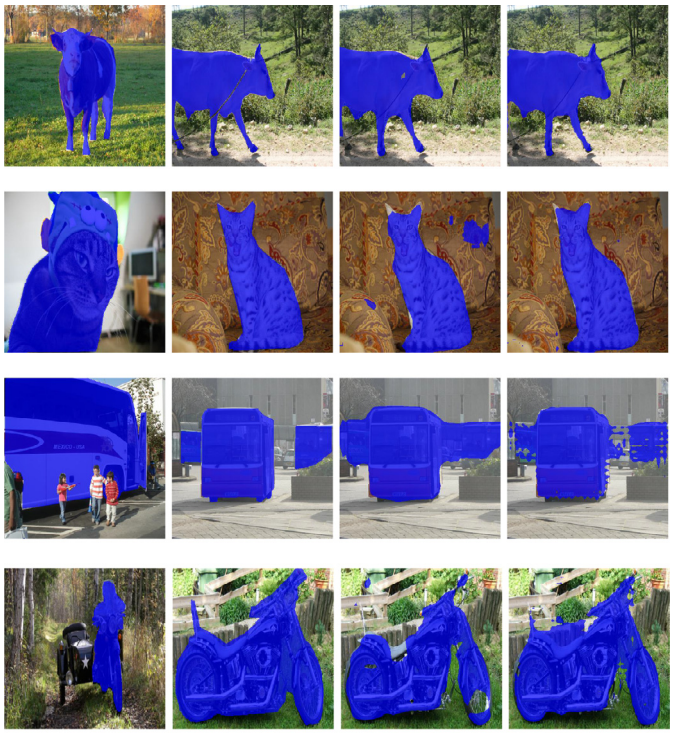
We use two semantic encodings in our work: (1) *Word2vec* [20] is trained on Google News dataset [32], which contains 3-million words; (2) *FastText* [19] is trained on CommonCrawl dataset [33]. We use these pre-trained models for extracting the class-name embeddings in our work.

Two widely-used metrices, Mean-IoU [25,26] and Binay-IoU [9,10,28] are used to report the segmentation performance of the proposed SML framework. Mean-IoU calculates mean of Intersection-over-Union for all the foreground classes. Binary-IoU calculates the Intersection-over-Union by treating all the foreground classes as one class and background class as one class.

5.2. Implementation details

We implement the proposed approach using PyTorch and use TITAN-X 2040, 12 GB GPU for the experiments. We use VGG-16 [34], and ResNet-50 [35] models, pre-trained on ImageNet [36] as the feature extractors to evaluate the effectiveness of SML for different backbones. For VGG-16 and ResNet-50, the image features are extracted from the output of 5th and 4th convolution block, respectively. In our implementation, we modify ResNet-50 as follows: first two residual blocks use convolution with stride 1, last two blocks are designed with dialated convolutions with 2, 4, to increase the receptive field. The number of episodes used to train the proposed SML is 30*k*. For each episode, the time taken by the proposed method with VGG-16 as backbone is 0.21 seconds. The images are resized to 417×417, as followed in PANet. `RandomMirror` augmentation is used in addition to the `ReSize` transformation. SGD-solver with weight decay of 5×10^{-4} and momentum of 0.9 is used for the optimization. We apply initial learning rate of 1×10^{-3} and 1.75×10^{-3} for VGG-16 and ResNet-50 respectively. The learning rate is reduced by a factor of 0.1 after every 10*k* iterations.

The learnable parameter λ in Eq. (4) is initialized to 100. We empirically set $\alpha = 10$ and $\beta = 1$ in (6). We have used pre-trained *Word2Vec* to extract the semantic embedding from class-names in our results and also used *FastText* for additional analysis. To mitigate the sensitivity of the model to random initialization, we repeat the experiments 5 times and report the Mean-IoU.



(a) Support **(b)** GT **(c)** VGG-16 **(d)** ResNet-50

Fig. 2. 1-way 1-shot segmentation results using proposed SML - (a) Support set image; (b) GT: query image with ground-truth mask; (c) and (d) are segmentation results using VGG-16 and ResNet-50 as the feature extractors.

5.3. Comparison with state-of-the-art methods

Here, we compare SML with several recent state-of-the-art methods - CANet [5], PANet [4], PPNet [17], AMP [6], SG-One [26], etc. for different protocols.

1-way 1-shot and 1-way 5-shot results on PASCAL-5i dataset We present the results for 1-way 1-shot and 1-way 5-shot semantic segmentation in Table 1 for both VGG-16 and ResNet-50 backbones as the feature extractor. The Mean-IoU reported in the table for all the other approaches are taken from [4,17]. We observe that for VGG-16, the proposed SML outperforms all the other approaches for both the protocols. The performance improvement compared to the other approaches is significantly more for 5-shot protocol. This implies that SML is able to effectively use all the images in the support set to improve the segmentation performance. For ResNet-50, SML performs at par with the recent PPNet [17]. Though CANet [5] performs better, it is evaluated with multi-scale input unlike the others, as also noted in Liu et al. [17]. We also compare the SML performance with the state-of-the-art methods using Binary-IoU as the evaluation criteria in Table 2. The results for all the other methods are taken from [4,6]. We observe that SML outperforms all other approaches for both the backbones. Specifically, for ResNet-50, it gives an improvement of 0.9% and 2.6% for 1-shot and 5-shot respectively. The segmentation results for few images using SML are given in Figs. 2 and 3 for 1-way 1-shot and 1-way 5-shot protocols respectively. We observe that SML provides good segmentation results even with significant background clutter. In general, ResNet-50 gives better results than VGG-16 backbone, which is also validated by the quantitative results.

1-way 1-shot results on COCO-20i dataset From Table 4, we observe that the proposed SML framework gives significantly better performance as compared to PANet with the same backbone

Table 1
Mean-IoU of the proposed SML and comparison with other state-of-the-art methods for both 1-way 1-shot and 1-way 5-shot protocols on PASCAL-5i data.

Method	Backbone	1-shot					5-shot				
		split-0	split-1	split-2	split-3	Mean	split-0	split-1	split-2	split-3	Mean
OSLM [25]	VGG16	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	43.9
co-FCN [10]	VGG16	36.7	50.6	44.9	32.4	41.1	37.5	50.0	44.1	33.9	33.9
SG-One [26]	VGG16	40.2	58.4	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1
AMP [6]	VGG16	–	–	–	–	43.4	–	–	–	–	46.9
PANet [4]	VGG16	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7
SML (Ours)	VGG16	**43.0**	**59.0**	**51.3**	**41.4**	**48.7**	**52.3**	**64.9**	**61.0**	**50.4**	**57.1**
PANet [4]	RN50	44.0	57.5	50.8	44.0	49.1	55.3	67.2	61.2	53.2	59.2
PPNet [17]	RN50	47.8	58.7	**53.8**	45.6	51.5	**58.3**	**67.8**	**64.8**	**56.7**	**61.9**
CANet [5]	RN50	**52.5**	**65.9**	51.3	**51.9**	**55.4**	55.5	**67.8**	51.9	53.2	57.1
SML (Ours)	RN50	47.4	59.7	**53.8**	44.4	51.3	56.0	**67.8**	62.1	54.0	60.0

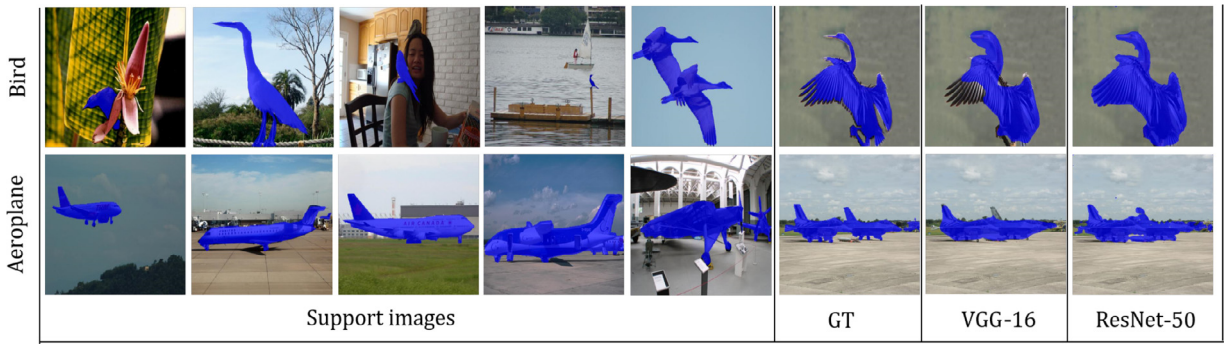


Fig. 3. 1-way 5-shot segmentation results using proposed SML on Pascal-5i dataset. First 5 columns show the support set images with ground-truth masks. Other columns show the query image with ground-truth mask and segmentation results using VGG-16 and ResNet-50 as the feature extractors respectively. Best viewed in color and zoomed.

Table 2
Binary-IoU of proposed SML and comparison with other state-of-the-art for 1-way 1-shot and 1-way 5-shot protocols on PASCAL-5i data.

Method	Backbone	1-shot	5-shot
FG-BG [27]	VGG16	55.0	–
Fine-Tuning [27]	VGG16	55.1	55.6
OSLSM [25]	VGG16	61.3	61.5
co-FCN [10]	VGG16	60.1	60.2
PL [9]	VGG16	61.2	62.3
A-MCG [28]	VGG16	61.2	62.2
SG-One [26]	VGG16	63.9	65.9
AMP [6]	VGG16	62.2	63.8
PANet [4]	VGG16	66.5	70.7
SML (Ours)	VGG16	**66.8**	**71.0**
CANet [5]	RN50	66.2	69.6
SML (Ours)	RN50	**67.1**	**72.2**

Table 3
Mean-IoU performance of proposed SML for 2-way 1-shot task on Pascal-5i dataset.

Method	split-0	split-1	split-2	split-3	Mean
PANet (VGG-16)	–	–	–	–	45.1
SML (VGG-16)	40.5	54.8	47.2	39.5	**45.5**
SML (ResNet-50)	43.2	55.6	49.5	44.4	**48.1**

Table 4
Mean-IoU resutls on COCO-20*i* dataset for 1-way 1 shot experiment using word2vec as attributes.

	Backbone	split-0	split-1	split-2	split-3	mean
PANet	VGG-16	28.7	21.2	19.1	14.8	20.9
SML	VGG16	29.3	24.4	20.1	16.4	**22.6**
SML	ResNet-50	31.9	23.7	20.3	17.1	**23.3**

Table 5
Binary-IoU resutls on COCO-20*i* dataset for 1-way 1 shot experiment using word2vec as attributes.

Method	backbone	Binary-IoU
PANet [4]	VGG-16	59.2
SML	VGG-16	**59.3**
SML	ResNet-50	**59.5**

(VGG-16) in terms of mean-IoU. As expected, the performance of SML improves further when ResNet-50 backbone is used. The proposed approach also performs favorably compared to PANet in terms of Binary-IoU, as shown in Table 5.

5.4. 2-way 1-shot results

From the performance of SML for 2-way 1-shot in Table 3, we observe that SML performs favorably compared to PANet [4] for VGG-16 backbone.

As expected, the results are significantly better with ResNet-50.

5.5. Additional analysis

Here we further analyze the proposed SML framework for better understanding.

Evaluation with weaker annotations: We experiment with weaker image annotations which are cheaper and easy to obtain, instead of the pixel-by-pixel dense annotations, which is extremely expensive and time-consuming. We observe from Table 6 that even for weaker annotations, the proposed SML performs better than PANet [4].

Evaluation with different semantic embedding: All experiments are reported with *Word2Vec* embedding. We also performed

Table 6
Mean-IoU of proposed SML with different image-annotations. D: Dense, BB: Bounding box, S: Scribble.

Method	1-shot			5-shot		
	D	BB	S	D	BB	S
PANet (VGG-16)	48.1	45.1	44.8	55.7	52.8	54.6
SML (VGG-16)	48.7	45.5	45.0	57.1	53.7	**55.8**
SML (ResNet-50)	51.3	47.9	49.2	60.0	55.1	**57.6**

experiment with *FastText* [19]. For Word2Vec and FastText, we obtain Mean-IoU of $\{51.3, 60.0\}$ and $\{50.9, 59.5\}$ for 1-shot and 5-shot respectively. We observe that Word2Vec performs slightly better compared to FastText.

6. Conclusion

We have proposed a novel Semantic Meta-Learning (SML) framework which utilises the semantic information of object classes for few-shot semantic segmentation. Towards that goal, we introduced a novel attribute-injector module in a traditional meta-learning setting. Extensive experiments with different protocols showed that the proposed framework performs similar or better as compared to the state-of-the-art.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, CVPR, 2015.
[2] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, TPAMI 39 (12) (2017) 2481–2495.
[3] G. Lin, A. Milan, C. Shen, I. Reid, Refinenet: multi-path refinement networks for high-resolution semantic segmentation, CVPR, 2017.
[4] K. Wang, J.H. Liew, Y. Zou, D. Zhou, J. Feng, Panet: few-shot image semantic segmentation with prototype alignment, ICCV, 2019.
[5] C. Zhang, G. Lin, F. Liu, R. Yao, C. Shen, Canet: class-agnostic segmentation networks with iterative refinement and attentive few-shot learning, CVPR, 2019.
[6] M. Siam, B. Oreshkin, Adaptive masked weight imprinting for few-shot segmentation, ICCV, 2019.
[7] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, NIPS, 2016.
[8] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, NIPS, 2017.
[9] N. Dong, E.P. Xing, Few-shot semantic segmentation with prototype learning., BMVC, 2018.
[10] K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, S. Levine, Conditional networks for few-shot semantic segmentation, ICLR, 2018.
[11] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, J. Jia, Prior guided feature enrichment network for few-shot segmentation, TPAMI (2020).
[12] P. Tian, Z. Wu, L. Qi, L. Wang, Y. Shi, Y. Gao, Differentiable meta-learning model for few-shot semantic segmentation., AAAI, 2020.
[13] L. Bertinetto, J.F. Henriques, P.H. Torr, A. Vedaldi, Meta-learning with differentiable closed-form solvers, ICLR, 2019.
[14] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, ICML, 2017.
[15] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, ICLR, 2017.
[16] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, CVPR, 2018.
[17] Y. Liu, X. Zhang, S. Zhang, X. He, Part-aware prototype network for few-shot semantic segmentation, arXiv preprint arXiv:2007.06309 (2020).
[18] Y. Boyu, L. Chang, L. Bohao, J. Jianbin, Q. Ye, Prototype mixture models for few-shot semantic segmentation, ECCV, 2020.
[19] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, T. Mikolov, Fasttext zip: compressing text classification models, arXiv preprint arXiv:1612.03651 (2016).
[20] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, NIPS, 2013.
[21] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, TPAMI 38 (7) (2016) 1425–1438.
[22] A. Pambala, T. Dutta, S. Biswas, Generative model with semantic embedding and integrated classifier for generalized zero-shot learning, WACV, 2020.
[23] R. Vilalta, Y. Drissi, A perspective view and survey of meta-learning, AI 18 (2) (2002) 77–95.
[24] V.K. Verma, P. Rai, A simple exponential family framework for zero-shot learning, ECML, 2017.
[25] A. Shaban, S. Bansal, Z. Liu, I. Essa, B. Boots, One-shot learning for semantic segmentation, arXiv preprint arXiv:1709.03410 (2017).
[26] X. Zhang, Y. Wei, Y. Yang, T.S. Huang, SG-one: similarity guidance network for one-shot semantic segmentation, TC, 2020.
[27] K. Rakelly, E. Shelhamer, T. Darrell, A.A. Efros, S. Levine, Few-shot segmentation propagation with guided networks, arXiv preprint arXiv:1806.07373 2018.
[28] T. Hu, P. Yang, C. Zhang, G. Yu, Y. Mu, C.G. Snoek, Attention-based multi-context guiding for few-shot semantic segmentation, AAAI, 2019.
[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, ECCV, 2014.
[30] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, IJCV 88 (2) (2010) 303–338.
[31] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, J. Malik, Semantic contours from inverse detectors, ICCV, 2011.
[32] X. Wang, Y. Ye, A. Gupta, Zero-shot recognition via semantic embeddings and knowledge graphs, CVPR, 2018.
[33] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, Advances in pre-training distributed word representations, LREC, 2018.
[34] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 2015.
[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, IJCV 115 (3) (2015) 211–252.