Representative Local Feature Mining for Few-shot Learning

Paper_ID: 1633

Kun Yan[1], Lingbo Liu[2], Jun Hou[3], Ping Wang[1]

[1]Peking University, [2]Sun Yat-Sen University, [3]SenseTime Research

Introduction

- Few-shot learning aims to recognize unseen images of new classes with only a few training examples.
- Most metric-based works rely on the measurement based on global feature representation of images, which is sensitive to background factors due to the scarcity of training data.
- Existing methods based on local features use the information of all local features contain no matter semantical parts or background factors.

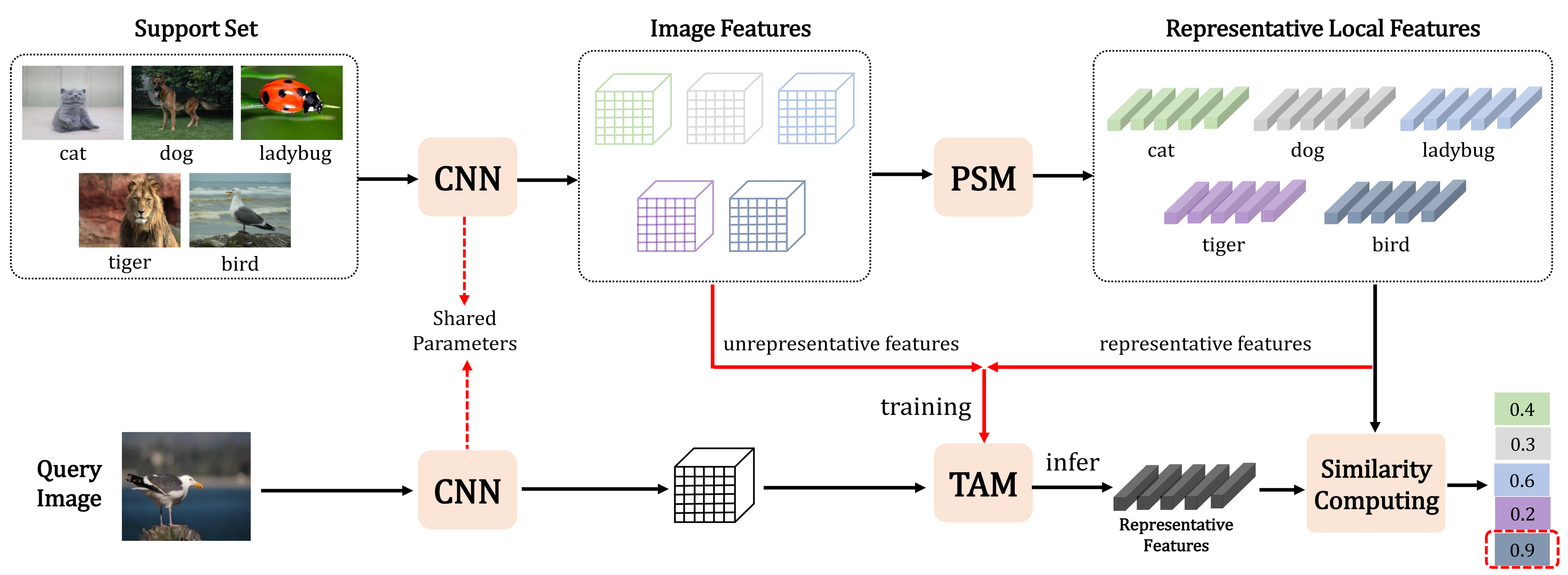
Problem setting up

- In FSL, we are given a base class set and a novel class set. Each class in base class set has sufficient labeled images, while only a few labeled samples are obtained for each class in the novel class set. we adopt the episode-based training scheme to facilitate few-shot learning. In each episode, each classification task is performed on a support set $\mathcal{S}$ and a query set $\mathcal{Q}$.
- In particular, $\mathcal{S}$ follows a N-way K-shot setting. N is the number of classes and K is the number of labeled examples in each class. Note that K is a small integer, such as 1 or 5.
- In training episodes, we optimize our model with $\mathcal{S}/\mathcal{Q}$ sampled from base class set. During the testing episodes, we measure the generalization performance of a model with $\mathcal{S}/\mathcal{Q}$ sampled from novel class set, where labels in $\mathcal{S}$ are known and those in $\mathcal{Q}$ are unknown.

Our Method

- We propose **a "task-specific guided" strategy** to mine local features that are task-specific and representative.
- We develop a **Prototype Selection Module (PSM)** to mine representative local features for labeled images by a loss guided mechanism through a simple image classification task
- We develop a **Task Adaption Module (TAM)** to adapt a binary classifier for unlabeled images based on representative local features from PSM.

Architecture



Prototype Selection Module

- We use the loss change of image classification to distinguishes the importance of each local feature.
- Each local feature is multiplied by a factor $\rho \in [0,1]$ to describe the existence weight of each local feature.
- We define the function to evaluate the importance of a local feature according to the impact on the classification loss: $g(\rho) = |\mathcal{L}(\rho) - \mathcal{L}(0)|$;
- For convenience of calculation, we apply the Taylor expansion to simplify the above formula:

$$\mathcal{L}(x) = \mathcal{L}(\rho) + \frac{\mathcal{L}^{(1)}(\rho)}{1!}(x - \rho) + \cdots + \frac{\mathcal{L}^{(n)}(\rho)}{n!}(x - \rho)^n + R_n(x),$$

the $\mathcal{L}(0)$ is estimated as $\mathcal{L}(\rho) - \rho\mathcal{L}^{(1)}(\rho) + R_1(0)$, the final $g(\rho)$ can be rewritten as:

$$g(\rho) = |\rho\mathcal{L}^{(1)}(\rho) - R_1(0)| \approx |\rho\mathcal{L}^{(1)}(\rho)|$$

Task Adaption Module

- We sample representative and discarded local features from PSM.
- Take representative local features as positive samples, while discarded local features as negative samples to train a binary classifier.
- Use the trained binary classifier mentioned above to mine representative local features for query set.
- During classifier inference, given an image-level feature, it would output a score between 0 to 1 for each local feature.

Results

- Validation of the effectiveness of PSM and TAM

Method	Backbone	Used Modules	5-way 5-shot
Baseline	Conv-64	-	80.83 ± 0.60
Baseline+PSM	Conv-64	+ PSM	82.94 ± 0.56
Baseline+PSM+TAM	Conv-64	+ PSM,TAM	**84.53 ± 0.65**
Baseline	ResNet-18	-	78.92 ± 0.66
Baseline+PSM	ResNet-18	+ PSM	80.13 ± 0.72
Baseline+PSM+TAM	ResNet-18	+ PSM,TAM	**81.21 ± 0.55**

- Time consuming

Method	Backbone	training phase	test phase
ProtoNet	Conv-64	0.394s/iteration	0.264s/iteration
MAML	Conv-64	0.511s/iteration	0.301s/iteration
Our method	Conv-64	0.473s/iteration	0.281s/iteration

- The mean accuracies (%) with a 95% confidence interval on the miniImageNet dataset

Method	Backbone	5-way 1-shot	5-way 5-shot
MAML [20]	Conv-64	48.70 ± 1.75	63.15 ± 0.91
Meta-SGD [21]	Conv-64	50.47 ± 1.87	64.03 ± 0.94
Reptile [22]	Conv-64	47.07 ± 0.26	62.74 ± 0.37
LEO [23]	WRN-28 [24]	61.76 ± 0.08	77.59 ± 0.12
Matching Net [8]	Conv-64	43.56 ± 0.84	55.31 ± 0.73
Prototypical Net [9]	Conv-64	49.42 ± 0.78	68.20 ± 0.66
RelationNet [10]	Conv-64	50.44 ± 0.82	65.32 ± 0.70
GNN [11]	Conv-64	50.33 ± 0.36	66.41 ± 0.63
Baseline++ [19]	Conv-64	48.24 ± 0.75	66.49 ± 0.63
SAML [13]	Conv-64	52.22 ± *	66.34 ± *
DN4 [12]	Conv-64	51.24 ± 0.74	71.02 ± 0.64
STANet-S [14]	Conv-64	53.11 ± 0.60	67.16 ± 0.66
CMT [15]	ResNet-18	62.05 ± 0.55	78.63 ± 0.06
FEAT [25]	Conv-64	55.15 ± *	71.61 ± *
Ours	Conv-64	53.98 ± 0.72	72.13 ± 0.63
Ours	ResNet-18	**62.79 ± 0.67**	**81.21 ± 0.55**

- The mean accuracies (%) with a 95% confidence interval on the CUB dataset

Method	Backbone	5-way 1-shot	5-way 5-shot
MAML [20]	Conv-64	55.92 ± 0.95	72.09 ± 0.76
Matching Net [8]	Conv-64	61.16 ± 0.89	72.86 ± 0.70
Prototypical Net [9]	Conv-64	51.31 ± 0.91	70.77 ± 0.69
RelationNet [10]	Conv-64	62.45 ± 0.98	76.11 ± 0.69
Baseline++ [19]	Conv-64	60.53 ± 0.83	79.34 ± 0.61
SAML [13]	Conv-64	69.33 ± 0.22	81.56 ± 0.15
DN4 [12]	Conv-64	53.15 ± 0.84	81.90 ± 0.60
Ours	Conv-64	**70.13 ± 0.62**	**84.53 ± 0.65**