Few-shot Open-set Recognition Using Background as Unknowns

Nan Song∗
nan001@e.ntu.edu.sg
Nanyang Technological University & Alibaba Group
Singapore

Chi Zhang
johnczhang@tencent.com
Tencent
China

Guosheng Lin†
gslin@ntu.edu.sg
Nanyang Technological University
Singapore

Figure 1: a) Close-set classification. Standard classification classifies all inputs to a set of pre-defined training classes. b) Few-shot open-set classification task aims to classify images from seen and unseen classes given only limited training samples of seen classes. We propose to utilize the background regions at the training phase as the pseudo unseen classes to train the classifier that can directly classify seen and unseen images.

ABSTRACT

Few-shot open-set recognition aims to classify both seen and novel images given only limited training data of seen classes. The challenge of this task is that the model is required not only to learn a discriminative classifier to classify the pre-defined classes with few training data but also to reject inputs from unseen classes that never appear at training time. In this paper, we propose to solve the problem from two novel aspects. First, instead of learning the decision boundaries between seen classes, as is done in standard close-set classification, we reserve space for unseen classes, such that images located in these areas are recognized as the unseen classes. Second, to effectively learn such decision boundaries, we propose to utilize the background features from seen classes. As these background regions do not significantly contribute to the decision of close-set classification, it is natural to use them as the pseudo unseen classes for classifier learning. Our extensive experiments show that our proposed method not only outperforms multiple baselines but also sets new state-of-the-art results on three popular benchmarks, namely tieredImageNet, miniImageNet, and Caltech-USCD Birds-200-2011 (CUB).

CCS CONCEPTS

• Computing methodologies → Object recognition; Supervised learning by classification.

KEYWORDS

few-shot learning, open-set recognition, class activation map

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Nan Song, Chi Zhang, and Guosheng Lin. 2022. Few-shot Open-set Recognition Using Background as Unknowns. In Proceedings of the 30th ACM International Conference on Multimedia (MM ’22), October 10–14, 2022, Lisboa, Portugal. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3547933
1 INTRODUCTION

Deep Neural Networks have made great success in computer vision tasks, such as image recognition, image segmentation, text understanding, and object detection [6, 20, 24, 52, 56]. Deep models based on Convolutional Neural Network (CNN) [11, 13, 17, 35] architectures have been proved efficient in excavating representative information from data in many fields.

Supervised learning is one of the most dominant learning approaches for deep models, where the optimization of deep models can be achieved by feeding a large amount of training data to the models and utilizing their labels as the training supervision. However, supervised learning limits the model application to the pre-defined training classes, which is often described as the close-set learning task. It is problematic to directly deploy these models to the real world, as a variety of novel categories that are out of the training classes may exist. As it is impossible to enumerate all the real-world classes for training, the learned model has to recognize inputs from new classes as one of the known classes, as illustrated in Fig. 1 (a), which hampers the applications of deep models in the real world.

Open-set recognition (OSR) is proposed, as a promising direction, to solve this kind of unknown class rejection problem. It aims to accurately classify the known classes and meanwhile detect the unknown classes. Many OSR methods make use of the large amount of training data to estimate the distributions of training classes, such that those out-of-distribution data can be recognized as unseen classes. This has largely solved the open-set recognition problem in many tasks. However, there are still many application scenarios where these algorithms are not applicable. For example, in long-tailed recognition, when the training data of some classes, e.g., rare animals, is not fairly enough, it is difficult to depict the distributions of these classes. Moreover, for tasks that rely on limited data to make predictions, such as few-shot learning, face verification, and person re-identification, these distribution-based methods would also fail. To handle the problems in these special settings, a more challenging task, namely, Few-shot Open-Set Recognition (FSOSR) is proposed [21], which aims to classify inputs from both known classes and unknown classes with only a few labeled training data of seen classes available.

The goal of FSOSR model is twofold: to identify discriminative information from limited training data, and to have the ability to recognize unseen samples. The well-studied few-shot learning literature has well addressed the former problem, hence the main challenge of FSOSR is to detect unseen samples without harming the classification accuracy, which is the focus of this paper. The main difficulty is that the information from training data to use for obtaining discriminative classifiers of seen classes is already very limited. A question is then raised — where can we find the information of unseen classes? In this paper, we argue that the background regions provide valuable information that can be utilized to detect samples from novel classes. In closed-set recognition, a CNN is optimized to identify the discriminative information in the training samples, with the help of activation functions. As a result, only the discriminative parts in the images are kept as the representations of the data for classification, and the backgrounds are therefore ignored. Since these ignored backgrounds do not belong to any of these existing classes, we treat them as the pseudo unseen classes to learn a classifier that can classify both seen and unseen classes meanwhile. Instead of learning the decision boundaries of only seen classes, we reserve the space for unseen classes with the representations from the background, and we can easily conduct open-set classification by detecting samples located in these areas. Our motivation is illustrated in Fig. 1 (b).

More specifically, we first initialize a class-mean classifier, where the prototypes, i.e., the weight vectors corresponding to different classes, are obtained by averaging embeddings of support samples from each class. We then randomly initialize a set of prototypes that correspond to unseen classes. Next, we use the background features as well as the features of support data to finetune the whole classifier. The labels of the background features are set as the nearest prototypes that correspond to unseen classes.

To accurately locate the background regions given limited training data, we further develop a Progressive Class Activation Mapping (ProCAM) module. Compared with the standard CAM [37], the Pro-CAM iteratively mines the class-related regions in images, which can better separate the foreground and background areas in the low-show case as shown in Fig. 2. We empirically show that our proposed ProCAM can not only generate better prototypes of unseen classes, but also regularize the learning of close-set classification.

Extensive experiments on multiple benchmark datasets have been conducted to verify the effectiveness of our proposed method. Our main contributions in this work can be summarized as follows.

- We solve the few-shot open-set recognition task as a multi-class classification problem, where a set of additional classes is added as unseen classes.
To obtain the data used for training the classifier, we propose to utilize the background regions of images from the seen classes.

We develop a Progressive CAM (ProCAM) module to iteratively separate the class-related foreground regions and the background regions.

Experiments on the tieredImageNet, CUB, and miniImageNet datasets show that our method significantly outperforms the baselines and sets new state-of-the-art performance with remarkable advantages.

2 RELATED WORK

2.1 Few-shot learning

Few-shot learning focuses on training models with insufficient training samples to gain the ability of classifying unseen images [5, 12, 23, 25, 36, 37, 41, 45, 47, 49–51, 53–55]. There are great diversities shown up in the literature of few-shot learning. The two main streams in the few-shot learning literature are optimization-based methods [4, 7, 18, 34, 37] and metric-based methods [12, 36, 43, 46, 47, 50].

Optimization-based methods target at finding good initial parameters and adapting these parameters to new tasks effectively with low shot supports. Ravi and Larochelle [32] optimize the model parameters using LSTM in few-shot setting. Finn et al. [7] proposed Model-Agnostic Meta-learning (MAML) where a meta-learner is designed to adapt parameters for cross-tasks. TaskNorm [4] proposed a normalization technique for meta-learning tasks to fuse different normalization values for efficient optimization.

Our model is more related to metric-based methods, which aim to distinguish different classes of instances with distance metrics in their feature space. MatchNet [43] introduces cosine similarity to measure data distance and compute the similarity scores. ProtoNet [36] proposes the negative L2 distance to measure the difference between the class prototypes and the query vector. Deep-EMD [50] applies Earth Mover’s Distance (EMD) as their distance metric. Recent methods transform the class prototypes to task-specific prototypes with task-adaptive weights [19] or transformer [47]. The limitation of the existing FSL methods is that it only guarantees the good performance under the closed-set settings. The query set and the support set are in the same classes. FSOSR extends the FSL task with the unseen detection capability. We inject the additional background classes into conventional FSL methods and get a better detection capability.

2.2 Open-set recognition

Open-set recognition addresses the classification model with the detection capability of unseen samples [2, 15, 16, 28, 38–40, 48]. It aims to reject unseen samples using the open-set classifier and classify the seen samples. The model cannot see the training samples for open-set data, and can only use the training data from the close-set.

Bendale et al. [2] train a close-set classification network first and then endow the trained network with open-set discrimination using Extreme Value Theory. Gopenmax [8] synthesizes fake open data during training. Neal et al. [28] propose to generate counterfactual images for the unknown class and train an (N+1)-class classifier to recognize the N known classes and 1 unknown class. Zhou et al. [58] also introduced the (N+1)-way classifier to solve the open-set recognition problem, but they synthesize fake open data using manifold mixup [42] techniques, while we utilize image background. Recently, GAN-based methods gain popularity [1, 8, 16, 28, 29, 39]. C2AE [29] and CGDL [39] use all the training samples to train an autoencoder and detect unseen samples by the large reconstruction error. Perera et al. [31] propose reconstructing the image along the channel dimension. The reconstruction of the unseen class samples is easy to fail which will lead to a lower classification probability for the unseen class. The normal OSR methods all require rich training data to train the unseen sample detector [2, 28, 29, 31, 39], which is hard to train in the FSL situation and getting poor performance.

2.3 Few-shot open-set recognition

FSOSR is recently proposed to address the open set recognition tasks in the few-shot inputs scenario [14, 21]. PEELER [21] proposes the FSOSR task and solves the problem using the meta-learning framework and Mahalanobis distance to classify the known samples. They introduce an open-set loss using a negative entropy loss to maximize the entropy for seen classes. Snatch [14] uses the pre-train model with the transformer from FEAT [47] and needs to reuse the transformer to remap all the known and unknown query samples and a set of prototypes to a new space to calculate the distance. Our method is a metric-based FSOSR approach which is the same as PEELER and Snatch, but we do not need the transformer module. PEELER [21] uses Mahalanobis distance. It needs to calculate the time-consuming covariance matrix. We follow MatchNet [43] to use cosine distance which is simple and effective.

FSOSR is different from the Generalized zero-shot learning [9, 22] (GZSL). Although both do not have support samples for new classes, GZSL provides the semantic information for the target classes. FSOSR targets rejecting unseen new classes, the class information is unknown. As for GZSL, it is known information with class-level semantic description and target to classify target class samples to the correct classes.

3 PROBLEM SET-UP

FSOSR aims to design a machine learning algorithm that can reject unseen classes from only a few seen training examples and classify the seen classes samples correctly. The typical setup for FSOSR follows episodic training and testing schema where batched tasks are sampled for training or testing. Specifically, an N-way K-shot FSOSR problem contains the support set and the query set for each episode.

For support set \( \mathcal{D}^S = \{x_i^S, y_i^S\}_{i=1}^{NK} \), where \( x_i \in X_S \) and \( y_i \in Y_S \). The data from the support set \( X_S \) are from the classes seen with the label \( Y_S \). The query set for FSOSR is different from FSL, \( \mathcal{D}^U = \mathcal{D}^K \cup \mathcal{D}^U \), where \( \mathcal{D}^K \) is the known query set and \( \mathcal{D}^U \) is the unknown query set. The known query set are samples from the seen classes with the same label space \( Y_S \), \( \mathcal{D}^K = \{x_i^K, y_i^K\}_{i=1}^{NQ} \), \( x_i^K \in X_K \) and \( y_i^K \in Y_S \). \( \mathcal{Q} \) is the number of queries in each query class. The unknown query is sample from the unseen class with label space \( Y_U \), \( \mathcal{D}^U = \{x_i^U, y_i^U\}_{i=1}^{NU} \), \( x_i^U \in X_U \), \( y_i^U \in Y_U \). There is no overlap for the seen labels and unseen \( Y_S \cap Y_U = \emptyset \).

\(^1\)We interchangeably use seen and known, and unseen and unknown in this paper.
An algorithm fits model \( f(x) \) is defined as \( f(x) = W^T \Phi(x) \), where \( \Phi(\cdot) : \mathbb{R}^D \rightarrow \mathbb{R}^d \) is the embedding function and \( W \in \mathbb{R}^{dxN} \) is the Fully Connected (FC) layer to classify seen classes. We denote the classifier for class \( i \) as \( w_i : W = [w_1, \ldots, w_N] \).

4 METHOD

In this section, we introduce our framework for few-shot Open-set Recognition. We first describe our extra background classifier method for clustering the unknown samples and separate the known classes in Section 4.1. Then we present our proposed background feature as unknowns schema to train the extra background classifiers in Section 4.2. The overview of the entire training pipeline is shown in Fig. 3.

4.1 Extra background classifier

The purpose of Few-shot Open Set Recognition (OSR) is to separate unknown samples from known samples. The known samples need to do close-set classification in further steps. The difficulty lies in that unknown samples are incorrectly classified as known samples if the classifier only contains known classes. For a CNN classifier with \( N \) classes, it will divide the embedding space into \( N \) parts. The class boundaries are determined by the discriminative information in the classifiers. Unknown samples projecting to the same embedding space will fall into known classes and classify wrongly with such kinds of classifiers. Therefore, we propose to reserve space for unknown classes, which is simple and straightforward by adding extra class classifiers. We get the hint from a special class in segmentation problems, the background class, which may contain a lot of unknown information compared to the foreground known classes. We name our extra class as background class for easy understanding. Let \( W_{bkg} \in \mathbb{R}^{d \times N_{bkg}} \) be the parameter matrix for the additional background classifier. It has the same number of feature channels \( d \) as the known classes classifier \( W_k \in \mathbb{R}^{d \times N_k} \). \( N_k \) is the number of known classes and \( N_{bkg} \) is the number of background classes we pre-set to cluster possible unknown samples. The final FC of our network becomes \( W = [W_k, W_{bkg}] \in \mathbb{R}^{d \times (N_k + N_{bkg})} \).

We employ the episodic training approach [36] to train our model. Each mini-batch of training will mimic an FSOSR task, the classifier weight for each known class \( i \) can be seen as the average embedding for each class \( i \).

\[
    w_i = \frac{1}{N} \sum_{j \in S_i} \Phi(x_j). \tag{1}
\]

We introduced the background features as unknowns method to train our background class classifier using the background features of the support data. The details are provided in Section 4.2.

We use the cosine distance as our distance metric to measure the similarity between the query embedding \( \Phi(X_Q) \) and the final FC of our network \( W \).

\[
    sim(\Phi(X_Q), W) = (\frac{W}{\|W\|_2})^T (\frac{\Phi(X)}{\|\Phi(X)\|_2}). \tag{2}
\]

The highest similarity of \( \Phi(x^{U}) \) to the classifier weight \( w_y \) define the prediction of \( x \) belong to \( y \). The loss can be expressed as:

\[
    L(x, y) = L_{CE} \left( f(x^K), y^K \right) + \lambda \cdot L_{CE} \left( f(x^U), \hat{y} \right), \tag{3}
\]

where \( \hat{y} = \text{argmax} \{ \text{sim}(W_{bkg}, \Phi(x^{U})) \} \) + \( |X| \) is the background class label with the maximum similarity the unknown query belongs to. The second term matches the unknown query to the reserved background classes space.
The ProCAM module is developed to apply CAM on the original image in spatial location \(c\) the discriminative regions for that category. For a given input image \(x\), let \(h_1(a, b)\) represent the activation of the image in spatial location \((a, b)\) in the last convolutional layer before average pooling. The CAM for class \(c\) can be calculated through the following equation:

\[
M_c^0(x) = \sum_{all(a,b)} w^c h_1(a, b),
\]

where \(w^c\) is the classifier weight for class \(c\). We scale \(M_c^0(x)\) to [0, 1] using the Min-Max normalization.

\[
\text{Norm}(M_c^0(x)) = \frac{M_c^0(x) - \min(M_c^0(x))}{\max(M_c^0(x)) - \min(M_c^0(x))},
\]

where \(M_c^0(x)\) is the original feature map. We improve ProCAM using softmax normalization to emphasize the high response regions and squeeze the low response regions to get a better mask for the foreground area.

\[
\text{softmax} M_c(x) = \frac{\exp(M_c^0(x))}{\sum_{all(a,b)} \exp(M_c^0(a,b))},
\]

where \(\text{softmax} M_c(x)\) represents the activation of the softmax layer for class \(c\). Following [14], we use the same network backbone Resnet12 for the known classifier. Figure 4 shows the details of the classifier training. The training input is the support feature and the background feature map \(\text{BKG feature map Bkg class FC}\) of different known classes. We proposed a ProCAM module that utilized multiple normalized class activation maps to mine class-related regions and get better background features.

4.2 Background feature as unknowns

FSOSR lacks open-set data to train the model. The existing OSR method normally uses GAN to generate pseudo-samples for open-set data, which is time-consuming and requires an extra branch of the generator network. We notice that the background of support data usually contains extra information and has a large difference from the foreground known class data. We can use background features as unknowns to train our extra background class and optimize the known classifier. Figure 4 shows the details of the classifier training. The training input is the support feature and the background feature of the support. The target label for the support feature is the known class label. We assign a background class label to the background feature. The backward propagation will only fine-tune the classifiers through the following loss.

\[
L_b(\Phi^\text{bkg}, \Phi) = L_{CE}(\Phi^\text{bkg}, \hat{y}) + \lambda_2 L_{CE}(\Phi^\text{bkg}, \hat{y}).
\]

The background feature can be extracted from the class activation map of different known classes. We proposed a ProCAM module that utilized multiple normalized class activation maps to mine class-related regions and get better background features.

4.3 ProCAM

The ProCAM module is developed to apply CAM on the original support features multiple times to figure out the previously uncovered known class regions. The maps found multiple times can be superimposed to get more complete background area. This section introduces details on how to generate multiple class activation maps. A class activation map for a specific category will highlight the discriminative regions for that category. For a given input image with labeled class \(c\), let \(h_0(a, b)\) represent the activation of the image in spatial location \((a, b)\) in the last convolutional layer before average pooling. The CAM for class \(c\) can be calculated through the following equation:

\[
M_c^0(x) = \sum_{all(a,b)} w^c h_0(a, b),
\]

where \(w^c\) is the classifier weight for class \(c\). We scale \(M_c^0(x)\) to [0, 1] using the Min-Max normalization.

\[
\text{Norm}(M_c^0(x)) = \frac{M_c^0(x) - \min(M_c^0(x))}{\max(M_c^0(x)) - \min(M_c^0(x))},
\]

The first background feature map is getting through \(h_1 = h_0 \times (1 - \text{Norm}(M_c^0(x)))\). We regard the first background as the input of the CAM equation again, the CAM function will highlight more discriminative regions to the class \(c\).

\[
M_c^1(x) = \sum_{all(a,b)} w^c h_1(a, b),
\]

By repeating this operation for \(\tau\) times can give us \(\tau\) different CAMs \([M_c^0, M_c^1, \cdots, M_c^\tau]\). We sum all these CAMs up and normalize them again to get the final ProCAM.

\[
\text{ProCAM} = \text{Norm}(\sum_{i=1}^{\tau} M_c^i(x)) - \text{Norm}(M_c^0(x)).
\]

The final background feature map \(h_{bkg} = h_0(1 - \text{ProCAM})\), where \(h_0\) is the original feature map. We improve ProCAM using softmax normalization to emphasize the high response regions and squeeze the low response regions to get a better mask for the foreground area.

\[
\text{softmax} M_c(x) = \frac{\exp(M_c^0(x))}{\sum_{all(a,b)} \exp(M_c^0(a,b))},
\]


5 EXPERIMENT

5.1 Datasets

\textit{miniImageNet} [43]: It is a popular benchmark for few-shot learning with 100 classes and 600 images in each class. [21] proposed to use this dataset for the FSOSR task. Following the conventional splitting [43] of meta-training(64), meta-validation(16), and meta-testing(20) for FSL, the few-shot open-set problem samples training episodes from the 64 meta-training classes, and testing episodes from the 20 meta-testing classes.

\textit{tieredImageNet} [33]: It is another popular FSL benchmark with more classes, which includes 608 classes from 34 super-classes. We follow [14] to split training, validation, and testing into 351, 97, and 160, respectively.

\textit{Caltech-UCSD Birds-200-2011} [44]: CUB is a fine-grained bird classification dataset with 200 categories and in total 11,788 images. Under the 5-way FSOSR setting, we randomly sampled 5 classes as the closes set and 5 classes as the open sets from 100 meta-training classes to form the training episodes. The validation episodes and testing episodes were sampled from the rest 50 meta-validation classes and 50 meta-testing classes.

5.2 Implementation Details

Following [14], we use the same network backbone Resnet12 for all datasets, where the last layer before the classifier contains 640 dimensions. We built our project using PyTorch [30]. The model was pre-trained with batch size of 128 for 500 epochs and optimized using SGD with momentum. The pre-training learning rate is adjusted using cosine annealing from 0.1 to 0 in a cycle of 100 epochs. Standard data augmentation with random crops, random color jitters, and random horizontal flips are applied during pre-training. Manifold-mixup in conjunction with self-supervision [27] is also
Algorithm 1

Background feature as unknowns. \( r \) is the number of multiple CAMs; \( y_k \) and \( \hat{y}_k \) indicate the ground truth label and the network predictions for the known class, respectively; \( y_{bkg} \) and \( \hat{y}_{bkg} \) indicates the ground truth label and the network predictions for the background class, respectively; \( L(\cdot) \) is the cross-entropy loss function. \( \lambda \) is the weight parameter

Require: Support datasets \( D_{train}^S \), pre-trained model \( R \), a randomly initialized FC for background classes \( W_{bkg} \).

Ensure: A trained FC for background classes \( W_{bkg} \) and known classes \( W_k \).

1. while not done do
2. \( W_k \) \leftarrow Learn FC layer upon \( R \) with \( D_{train}^S \)
3. \( h_{imp}(x^S) \) \leftarrow Get the original feature map upon \( R \) with \( D_{train}^S \)
4. for \( i \) in range(\( r \)) do
5. \( cam\_all = [] \) \leftarrow An empty list to store all CAM
6. \( M^f(x^S) \) \leftarrow Get CAM from \( h_{imp}(x^S) \) and \( W_k \) with eq. 5;
7. \( Norm(M^f(x^S)) \) \leftarrow Get normalized CAM using eq. 6;
8. \( cam\_all.append(Norm(M^f(x^S))) \)
9. \( h_{imp}(x^S) = h_{imp}(1 - Norm(M^f(x^S))) \) \leftarrow Update \( h_{imp} \) to bkg feature map
10. end for
11. \( ProCAM \) \leftarrow Get normalized multiple CAM using eq. 9;
12. \( h_{bkg} = h_0(1 - ProCAM) \)
13. \( \{y_k, \hat{y}_{bkg}\} \) \leftarrow Make predictions for \( \{h_0(x^S), h_{bkg}\} \) using \( [W_k, W_{bkg}] \)
14. \( loss3 \) \leftarrow Compute loss with \( L_3(y_k, \hat{y}_k) \).
15. \( loss4 \) \leftarrow Compute loss with \( L_4(\hat{y}_{bkg}, y_{bkg}) \).
16. \( loss = loss3 + \lambda loss4 \) \leftarrow get total loss.
17. \( [W_k, W_{bkg}] \) \leftarrow Update classifier with SGD
18. end while

Table 1: Ablation study on different components for our method on 5-shot TieredImageNet dataset

| Model              | meta learning | cam | ft_all_proto | ProCAM | acc  | auroc |
|--------------------|---------------|-----|--------------|--------|------|-------|
| base               |               |     |              |        | 82.53| 78.54 |
| meta_cos           |               | ✓   |              |        | 84.59| 80.95 |
| meta_cam           |               | ✓   | ✓            |        | 85.08| 82.41 |
| meta_cam_ft        | ✓             | ✓   | ✓            |        | 85.67| 82.45 |
| meta_cam_ft_Norm   | ✓             | ✓   | ✓            | ✓      | 86.34| 82.76 |

Table 2: Comparison of different initialization methods to finetune extra background class classifier for 5-shot classification. Our model with random initialization to finetune 20 epochs for the extra background classifier has the best performance.

| init_method | f.t. epochs | acc  | auroc |
|-------------|-------------|------|-------|
| random      | epd10       | 83.05| 76.15 |
| random      | epd20       | 83.73| 77.44 |
| random      | epd30       | 83.60| 77.48 |
| random      | epd50       | 83.47| 77.28 |
| random      | epd100      | 83.26| 76.47 |
| avg         | epd10       | 76.35| 73.71 |
| avg         | epd20       | 76.49| 75.50 |
| avg         | epd30       | 76.69| 75.48 |
| avg         | epd50       | 76.65| 74.08 |
| avg         | epd100      | 76.36| 76.48 |
| global      | epd10       | 75.51| 75.29 |
| global      | epd20       | 74.86| 75.56 |
| global      | epd30       | 75.00| 75.90 |
| global      | epd50       | 74.04| 75.56 |
| global      | epd100      | 73.99| 75.52 |

Figure 5: Analysis accuracy(%) and auroc(%) on different number of ProCAM and extra background class. The best number of repeating ProCAM is 4 times for both accuracy and AUROC on 1-shot. The highest AUROC for 5-shot is having 2 extra background classes. Increasing the number of extra background classes does not affect the accuracy too much.
Table 3: Comparison with the State-of-the-art methods. Average closed-set classification accuracies (%) and average unknown detection AUROCs (%) over 600 episodes for MiniImageNet and TieredImageNet. PEELER* is quoted from the paper, which has a ResNet-18 backbone.

| Model          | miniImageNet 5-way | tieredImageNet 5-way |
|----------------|--------------------|----------------------|
|                | 1-shot             | 5-shot               | 1-shot              | 5-shot              |
|                | Acc    | AUROC | Acc    | AUROC | Acc    | AUROC | Acc    | AUROC |
| ProtoNet [36]  | 64.01  | 51.81 | 80.09  | 60.39 | 68.26  | 60.73 | 83.40  | 64.96 |
| FEAT [47]      | 67.92  | 57.01 | 82.02  | 63.18 | 70.52  | 63.54 | 84.74  | 70.74 |
| ASY [3]        | 61.64  | 66.43 | 82.37  | 74.45 | 66.28  | 71.06 | 82.53  | 78.54 |
| NN [15]        | 63.82  | 56.96 | 80.12  | 63.43 | 67.73  | 62.70 | 83.43  | 69.77 |
| OpenMax [2]    | 63.69  | 62.64 | 80.56  | 62.27 | 68.28  | 60.13 | 83.48  | 65.51 |
| baseline       | 63.18  | 70.22 | 80.09  | 75.97 | 68.48  | 72.98 | 83.52  | 80.07 |
| PEELER* [21]   | 58.31  | 61.66 | 75.08  | 69.85 | –      | –     | –      | –     |
| PEELER [21]    | 65.86  | 60.57 | 80.61  | 67.35 | 69.51  | 65.20 | 84.10  | 73.27 |
| SnaTCHer [14]  | 67.02  | 68.27 | 82.02  | 77.42 | 70.52  | 74.28 | 84.74  | 82.02 |
| Ours_ProCAM    | 67.78  | 71.41 | 83.45  | 78.19 | 68.16  | 75.35 | 86.34  | 82.76 |
| Ours_ProCAMsm  | 67.86  | 71.09 | 83.66  | 77.51 | 68.82  | 75.55 | 85.64  | 82.77 |

Table 4: Comparison with the State-of-the-art methods. Average closed-set classification accuracies (%) and average unknown detection AUROCs (%) over 600 episodes for CUB.

| Model          | CUB 5-way | 1-shot | 5-shot |
|----------------|-----------|--------|--------|
|                | Acc    | AUROC | Acc    | AUROC |
| baseline       | 61.24  | 68.43 | 76.23  | 75.09 |
| ProtoNet [36]  | 57.31  | 60.52 | 73.19  | 64.55 |
| PEELER [21]    | 59.42  | 58.63 | 78.42  | 66.04 |
| SnaTCHer [14]  | 57.98  | 64.55 | 77.05  | 71.05 |
| Ours_ProCAM    | 65.88  | 75.88 | 81.14  | 83.70 |
| Ours_ProCAMsm  | 68.34  | 76.05 | 82.22  | 82.34 |

Figure 6: T-SNE [26] visualization of data embeddings and classifiers before (left) and after (right) fine-tune with background features as unknowns. Crosses with different colors represent query data points from different known classes. The pink plus represents unknown query. The blue squares and triangles indicate the classifier prototypes before and after fine-tuning, respectively. Our module reserve the space for the background classifier as shown in the green area.

5.3 Analysis

We evaluate the effectiveness of different initialization methods for the background class in Table 2. For each initialization method, we compare the different number of fine-tune epochs for the background class classifier learning. The random initialization method uses the Kaiming uniform initialization [10]. The ‘avg’ method initializes the background classifier weight with the average background feature maps. The ‘global’ method uses a global FC for background class across all the training and testing episodes. The results show that random initialization for every episode and fine-tuning with 20 epochs have a better result compared to the other methods. Fig. 5 shows the accuracy and AUROC change w.r.t. the number of repeating ProCAMs and the number of additional background classes. There is an increasing trend for accuracy and AUROC when the number of repetitions increases and reaches a maximum at r = 4. Increasing the number of extra background classes does not affect the accuracy too much, but the AUROC changes a lot. The highest AUROC for miniImageNet 5-shot is adding 2 extra background classes.

We also plot the t-SNE figure [26] to see the update of the feature space for the background class. Fig. 6 shows that our proposed background classifier helps reserve space for unknown samples and provides a direction for the unknown class to cluster to the extra background class centers.

We provide a visualization of our ProCAM and softmax ProCAM, and compared them with the original CAM in Fig. 7. Our methods generate more accurate foreground masks, which cover all the important features in the known classes. The background feature based on our mask contains less discriminative information for seen class and can provide good background data to train our extra background classes. The background feature data is also helpful for pushing the boundary of known classes. Softmax ProCAM is similar to the ProCAM results, the background and foreground boundaries are more clear than ProCAM.

Ablation study. We perform our ablation study on the 5-way 5-shot setting for tieredImageNet to analyze the effectiveness of different components in 1. The base method is the testing result for the pre-train model under FSOSR setting directly. The method ‘meta_cos’ employs the meta-learning and cosine distance and gets a better result than the pre-training model. We apply the vanilla class activation map to the previous ‘meta_cos’ method and we can get higher accuracy with better AUROC for this ‘meta_cam’ model.
The previous CAM method only uses the background feature to fine-tune the background class classifiers, we try to fine-tune all the FCs using both the support feature and the background feature in ‘meta_cam_ft’. The result provides around 0.6% of the increase in the accuracy, which indicates that the background features can help push the boundaries of the known classes and train better background class classifiers. We optimize the fine-tune of all FCs with better background feature maps. The results show an increasing trend in known class accuracy and unknown class AUROC.

5.4 Comparison with the State-of-the-Art Methods

Finally, we conduct comparison of our performance with the state-of-the-art results on three benchmarks: miniImageNet, tieredImageNet, and CUB. We choose the state-of-the-art FSL methods ProtoNet [36], FEAT [47], and ASY [3] as the first group of models. In the second group, we select two OSR methods (NN [15], OpenMax [2]). The third group is our baseline with mete-learning and cosine classifier and the current existing FSOSR methods PEELER and SnaTCher [14, 21]. Because the existing FSL and OSR methods are not designed for FSOSR tasks, we follow SnaTCher [14] to slightly modify them for comparison. We show the detailed numbers for miniImageNet and tieredImageNet in Table 3. The comparison for CUB is in Table 4. Our model has the highest AUROC of all the methods. The accuracy of the known classes is also competitive among all the datasets. In particular, our AUROC outperforms the state-of-the-art result by 3.14% on miniImageNet 5-way 1-shot. Our accuracy increased 1.64% for the same dataset 5-shot. For tieredImageNet, the 1-shot accuracy is slightly lower maybe due to the lack of support background data, but the AUROC still improved around 1.27%. The accuracy and AUROC on CUB outperform the state-of-the-art method by 10.56% and 11.5%, respectively.

6 CONCLUSION

In this paper, we solve the few-shot open-set recognition problems from two aspects. We first introduced extra background class classifiers to reserve space for unknown samples, which use cosine distance to determine the space for each class. Then, we propose a learning paradigm to utilize the background features as unknowns to train the extra background class classifiers and fine-tune the known classes’ FCs. To generate a better mask for the foreground objects and get a better background feature, we propose a ProCAM module to utilize multiple class activation maps. We optimize our ProCAM module using softmax to separate the foreground part and generate clear boundaries of the background. Experiments on three datasets show that our method significantly outperforms the baselines and the state-of-the-art approaches.

ACKNOWLEDGMENTS

This work is supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore. This research is also supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-003), the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE-T2EP20220-0007) and Tier 1 (RG95/20).
[48] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. 2019. Classification-reconstruction learning for open-set recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4016–4025.

[49] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. 2020. DeepEMD: Differentiable Earth Mover’s Distance for Few-Shot Learning. arXiv e-prints (2020).

[50] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. 2020. DeepEMD: Few-Shot Image Classification With Differentiable Earth Mover’s Distance and Structured Classifiers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[51] Chi Zhang, Henghui Ding, Guosheng Lin, Ruibo Li, Changhu Wang, and Chunhua Shen. 2021. Meta Navigator: Search for a Good Adaptation Policy for Few-shot Learning. In IEEE International Conference on Computer Vision (ICCV).

[52] Chi Zhang, Guankai Li, Guosheng Lin, Qingyao Wu, and Rui Yao. 2021. Cycleseg-net: Object co-segmentation with cycle refinement and region correspondence. IEEE Transactions on Image Processing (2021).

[53] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. 2019. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision. 9587–9595.

[54] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. 2019. CANet: Class-Agnostic Segmentation Networks with Iterative Refinement and Attentive Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5217–5226.

[55] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. 2021. Few-Shot Incremental Learning with Continually Evolved Classifiers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[56] Chi Zhang, Rui Yao, and Jinpeng Cai. 2018. Efficient eye typing with 9-direction gaze estimation. Multimedia Tools and Applications 77, 15 (2018), 19679–19696.

[57] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2921–2929.

[58] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. 2021. Learning Placeholders for Open-Set Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4401–4410.