Few-Shot Open-Set Recognition using Meta-Learning

Bo Liu  
UC, San Diego  
boliu@ucsd.edu

Hao Kang  
Wormpex AI Research  
haokheseri@gmail.com

Haoxiang Li  
Wormpex AI Research  
lhxustcer@gmail.com

Gang Hua  
Wormpex AI Research  
ganghua@gmail.com

Nuno Vasconcelos  
UC, San Diego  
nuno@ece.ucsd.edu

Abstract

The problem of open-set recognition is considered. While previous approaches only consider this problem in the context of large-scale classifier training, we seek a unified solution for this and the low-shot classification setting. It is argued that the classic softmax classifier is a poor solution for open-set recognition, since it tends to overfit on the training classes. Randomization is then proposed as a solution to this problem. This suggests the use of meta-learning techniques, commonly used for few-shot classification, for the solution of open-set recognition. A new open set meta learning (PEELER) algorithm is then introduced. This combines the random selection of a set of novel classes per episode, a loss that maximizes the posterior entropy for examples of those classes, and a new metric learning formulation based on the Mahalanobis distance. Experimental results show that PEELER achieves state of the art open set recognition performance for both few-shot and large-scale recognition. On CIFAR and miniImageNet, it achieves substantial gains in seen/unseen class detection AUROC for a given seen-class classification accuracy.

1. Introduction

The introduction of deep convolutional neural networks (CNNs) has catalyzed large advances in computer vision. Most of these advances can be traced back to advances on object recognition, due to the introduction of large scale datasets, such as ImageNet [2], containing many classes and many examples per class. Many modern computer vision architectures are entirely or partially based on recognition networks. In the large scale setting, CNN-based classifiers trained by cross-entropy loss and mini-batch SGD have excellent recognition performance, achieving state of the art results on most recognition benchmarks.

However, variations of the recognition setting can lead to substantial performance degradation. Well known examples include few-shot learning [25, 3, 33, 29], where only a few training examples are available per class, domain adaptation [32], where a classifier trained on a source image domain (e.g. synthetic images) must be deployed to a target domain (e.g. natural images) whose statistics differ from those of the source, long tailed recognition [18], where the number of examples per class is highly unbalanced, or problems with label noise [20, 30]. All these alternative recognition settings stress the robustness, or ability to generalize, of the large-scale classifier.

More recently, there has been interest in endowing CNNs with self-awareness capabilities. At the core, self-awareness implies that, like humans, CNNs should identify what they can do and refuse what they cannot. Several variants of this problem have been proposed, including out-of-distribution detection [9], where the CNN rejects examples outside the training distribution (e.g. images from other domains or adversarial attacks [11]), realistic classification [19], where it rejects examples that it deems too hard to classify, or open-set recognition [1], where the examples to reject are from novel classes, unseen during training. While several techniques have been proposed, a popular approach is to force the CNN to produce high entropy posterior distributions in the rejection regions. Rejections can then be identified as examples that give rise to such distributions.

In this work, we consider open-set recognition. This has been mostly addressed in the large-scale setting, using so-
lutions based on the large-scale classifier. These attempt to recognize the novel classes by post-processing the posterior class distribution [1], defining a “reject” class trained with examples that are either artificially generated [5] or sampled from the training classes [28], or a combination of the two. We seek to generalize open-set recognition beyond the large-scale setting, proposing that it should target a continuum of tasks that includes all the alternative recognition settings. Since training over many settings can be complex, we consider the two at the extremes of this continuum: large-scale and few-shot recognition.

There are three main reasons for this. First, open-set recognition is a challenge under all settings. A recognizer trained in the few-shot regime is not less likely to face unseen classes. An open-set recognition technique that also supports the few-shot setting is thus more useful than the one that does not. Second, few-shot open-set recognition is harder to solve than large-scale open-set recognition, due to the scarcity of labeled data. Hence, the few-shot setting poses a greater challenge to open-set recognition research. Third, like open-set recognition, the main challenge of few-shot recognition is to make accurate decisions for data unseen during training. Since this makes robustness the main trait of few-shot architectures, these architectures are likely to also excel at open-set recognition. Most notably, they are likely to beat the large-scale classifier, which tends not to score highly along the robustness dimension.

The main limitation of the large-scale classifier (trained by cross-entropy) is that it tends to overfit the training classes. Because the ideal embedding for classification maps all examples of each class into a unique point in feature space, the large-scale classifier tends to learn embeddings that are only locally accurate. While the metric structure of the embedding is reflected by the semantic distances in the neighborhood of class representatives (the parameter vectors of the softmax layer), these distances are meaningless away from the latter. In result, large-scale classifier embeddings underperform on tasks that require generalization beyond the training set, such as image retrieval [7], face identification [17], pose invariant recognition [10], person re-identification [38], or few-shot learning [33]. Embeddings derived from the metric learning literature [37] or explicitly designed for problems like few-shot recognition [36] tend to capture the metric structure of the data more broadly throughout the feature space and achieve better performance on these generalization critical tasks. It is thus expected that these embeddings will outperform the large-scale classifier on the open set recognition problem.

In this work, we investigate the open-set performance of a class of popular solutions to the few-shot problem, known as meta-learning (ML) [33]. ML approaches replace the traditional mini-batch CNN training by episodic training. Episodes are generated by randomly sampling a subset of the classes and a subset of examples per class, to produce a support and query set to which a classification loss is applied. This randomizes the classification task for which the embedding is optimized at each ML step, producing a more robust embedding, less overfitted to any particular set of classes. We generalize the idea to open-set recognition, by randomly selecting a set of novel classes per episode, and introducing a loss that maximizes the posterior entropy for examples of these classes. This forces the embedding to better account for unseen classes, producing high-entropy posterior distributions beyond the regions of the target classes.

This solution has at least two strong benefits. First, because it draws on a state of the art approach to few shot learning, it immediately extends open set recognition to few-shot problems. Second, because ML embeddings are robust and open set recognition is about generalization, the performance of the latter improves even in the large-scale setting. This is shown by extensive experiments with various open-set recognition benchmarks, where we demonstrate significant improvements over the state of the art. We also investigate the role of the metric used in the feature space on open-set recognition performance, showing that a particular form of the Mahalanobis distance enables significant gains over the commonly used Euclidean distance.

Overall, the contributions of this work can be summarized as follows:

- A new ML-based formulation of open-set recognition. This generalizes open-set to the few-shot recognition setting.
- A new episodic training procedure, combining the cross-entropy loss and a novel open-set loss to improve open-set performance on both the large-scale and few-shot settings.
- A Gaussian embedding for ML-based open-set recognition.

2. Related Work

Open-Set Recognition: Open-set recognition addresses the classification setting where inference can face samples from classes unseen during training. The goal is to endow the open-set classifier with a mechanism to reject such samples. One of the first deep learning approaches was the work of Scheirer et al. [1], which proposed an extreme value parameter redistribution method for the logits generated by the classifier. Later works considered the problem in either discriminative or generative models. Schlachter et al. [28] proposed an intra-class splitting method, where a closed-set classifier is used to split data into typical and atypical subsets, reformulating open-set recognition as a traditional classification problem. G-OpenMax [5] utilizes a generator trained to synthesize examples from an extra class
that represents all unknown classes. Neal et al. [21] introduced counterfactual image generation, which aims to generate samples that cannot be classified into any of the seen classes, producing an extra class for classifier training.

All these methods reduce the set of unseen classes to one extra class. While open samples can be drawn from different categories and have significant visual differences, they assume that a feature extractor can map them all into a single feature space cluster. While theoretically possible, this is difficult. Instead, we allow a cluster per seen class and label samples that do not fall into these clusters as unseen. We believe this is a more natural way to detect unseen classes.

**Out-of-Distribution:** A similar problem to open-set recognition is to detect out-of-distribution (OOD) examples. Typically [9], this is formulated as the detection of samples that represent all unknown classes. Neal et al. [21] implemented by multiple neural network layers, and a softmax layer with linear mapping that estimates class posterior probabilities with

\[ p(y = k|x; \phi, w_k) = \frac{\exp(w_k^T f_\phi(x))}{\sum_{k'} \exp(w_{k'}^T f_\phi(x))} \]  

where \( \phi \) denotes all the embedding parameters and \( w_k \) is a set of classifier weight vectors. Recent works [29] have combined metric learning with softmax classification, implementing the softmax layer with a distance function

\[ p_\phi(y = k|x) = \frac{\exp(-d(f_\phi(x), \mu_k))}{\sum_{k'} \exp(-d(f_\phi(x), \mu_{k'}))} \]  

where \( \mu_k = E[f_\phi(x)|y = k] \) [29]. The softmax classifier is learned with a training set \( S = \{(x_i^*, y_i^*)\}_{i=1}^{n^*} \), where \( y_i^* \in C^*, \forall i, C^* \) is a set of training image classes, and \( n^* \) the number of training examples. This consists of finding the classifier and embedding parameters that minimize the cross entropy loss

\[ \mathcal{L}_{CE} = \sum_{(x_i^*, y_i^*)} - \log p(y_i^*|x_i^*) \]  

Recognition performance is evaluated on a test set \( T = \{(x_i, y_i)\}_{i=1}^{n^*}, \) where \( y_i \in C^i, \forall i, C^i \) is a set of test classes, and \( n^* \) the number of test examples.

**Closed vs. open set classification.** Under the traditional definition of classification, the sets of training and test
classes are identical, i.e. \( \mathbb{C}^s = \mathbb{C}^t \). This is denoted as closed set classification. Recently, there has been interest in an alternative open set classification setting, where \( \mathbb{C}^t = \mathbb{C}^s \cup \mathbb{C}^u \). In this case, the classes in \( \mathbb{C}^s \) are denoted as seen (during training) and the classes in \( \mathbb{C}^u \) as unseen.

**Large-scale vs few-shot recognition.** In the large-scale recognition setting, the number \( n^s \) of training examples is quite high, reaching the millions for datasets like ImageNet. On the contrary, for few-shot recognition, this number is quite low, usually less than twenty examples per class. A few-shot problem with \( K \) training examples per class is commonly known as K-shot recognition. Note that the number \( n^s \) of test examples plays no role in differentiating these two settings. Since these examples are only used for performance evaluation, the test set should have the same cardinality under the two settings. As shown in Table 1, different combinations of properties in the scale and coverage of the training data define different classification paradigms. The few-shot open-set setting is largely unexplored in the literature and the focus of this paper.

### 3.2. Meta-learning

Meta-learning (ML) addresses the problem of “learning to learn”. In this case, a meta-learner learns a learning algorithm by inspection of many learning problems. For this, the meta-learner relies on a meta training set \( \mathbb{M}_S = \{(S^i_t, T^i_t)\}_{i=1}^{N^C} \), where \((S^i_t, T^i_t)\) are the training and test set of the \( i^{th} \) learning problem and \( N^s \) the number of learning problems used for training; and a meta test set \( \mathbb{M}_T = \{(S^i_s, T^i_s)\}_{i=1}^{N^t} \), where \((S^i_s, T^i_s)\) are the training and test set of the \( i^{th} \) test learning problem and \( N^t \) is the number of learning problems used for testing. Given \( \mathbb{M}_S \), the meta-learner learns how to map a pair \((S, T)\) into an algorithm that leverages \( S \) to optimally solve \( T \).

The procedures are as follows. At meta iteration \( i \), a meta-model \( h \) is initialized with the one produced by the previous meta-iteration. Two steps are then performed. First, the meta-learning algorithm performs the mapping

\[
h' = \mathcal{M}(h, S^i_s)
\]

(4) to produce an estimate \( h' \) of the optimal model for training set \( S^i_s \). The test set \( T^i_t \) is then used to find the model

\[
h^* = \arg \min_h \sum_{(x_k, y_k) \in T^i_t} L[y_k, h'(x_k)]
\]

(5) for a suitable loss function \( L \), e.g. cross-entropy, using a suitable optimization procedure, e.g. back-propagation. The resulting \( h^* \) is finally returned as the optimal meta-model for meta-iteration \( i \). During testing, the final meta-model \( h^* \) and a training set \( S^i_t \) from the meta test set \( \mathbb{M}_T \) are used by the meta-learner to produce a new model

\[
h'' = \mathcal{M}(h^*, S^i_t)
\]

(6) whose performance is evaluated with \( T^i_t \).

**ML for few-shot recognition.** While different approaches have been proposed to apply ML to few shot recognition, in this work we adopt the procedure introduced by the popular prototypical network architecture [29], which is the foundation of various other approaches. In this context, ML is mostly a randomization procedure. The pairs \((S^i_t, T^i_t)\) are denoted episodes, the training sets \( S^i_t \) are denoted support sets and the test sets \( T^i_t \) query sets. The meta-training set \( \mathbb{M}_S \) is generated by sampling train and test classes. In particular, the training set \( S^i_t \) of the \( i^{th} \) episode is obtained by sampling \( N \) classes from the set of classes of the low-shot problem, and \( K \)-examples per class. This defines a set of \( K \)-shot learning problems and is known as the \( N \)-way \( K \) shot problem. The model \( h \) is the softmax classifier of (2), the meta-learning mapping of (4) implements the Gaussian mean maximum likelihood estimator

\[
\mu_k = \frac{1}{|P_{i,k}|} \sum_{x_j \in P_{i,k}} f_\phi(x_j)
\]

(7) where \( P_{i,k} = \{x_j \in S^i_t | y_j = k\} \) is the set of support samples from class \( k \), and back-propagation is used in (5) to update the embedding \( f_\phi \).

**Benefits of ML for open-set classification.** The ML procedure is quite similar to the standard mini-batch procedure for training softmax classifiers. Aside from the use of episode training instead of the more popular mini-batch training, there are two main differences. First, \( f_\phi \) is updated by back-propagation from examples in \( T^i_t \) rather than \( S^i_t \). This is advantageous for few-shot learning due to the consistency with meta-testing, which is not the case in the large-scale regime. Second, mini-batches of random examples from all classes are replaced by examples from the subset of \( N \) classes contained in \( S^i_t \) and \( T^i_t \). Randomizing the classification task learned per episode forces the embedding \( f_\phi \) to generalize better to unseen data. This property makes ML a good solution for few-shot learning, where the training data lacks a good coverage of intra-class variations. In this case, large subsets of test samples from the known classes are unseen during training. We propose that the same property makes ML a good candidate for open-set classification, where by definition the classifier must deal with unseen samples from unseen classes. This observation motivates us to design a unified ML-based solution to open-set classification that supports both large-scale and few-shot classification.
4. Meta Learning-based open-set recognition

In this section, we introduce the proposed ML approach to open-set recognition. We first introduce the general procedure and then discuss the specific embedding metric used in our PEELER implementation.

4.1. Open-Set Meta-Learning

As shown in Figure 1, open-set meta-learning (PEELER) relies on a meta training set $S$ and a meta test set $T$. The only difference with respect to standard ML is that the episodes in standard ML are open-set. Hence, PEELER can be addressed with a solution similar to the ML procedure of Section 3.2. While the ML step remains as in (4), the optimization step of (5) becomes

$$h^* = \arg \min_h \left\{ \sum_{(x_k, y_k) \in T_i^u | y_k \in C_i^u} L_c[y_k, h'(x_k)] + \lambda \sum_{(x_k, y_k) \in T_i^u | y_k \in C_i^u} L_o[h'(x_k)] \right\}$$

(8)

where $C_i^u$ is the set of seen (unseen) classes of $T_i^u$, $L_c[\cdot]$ is the classification loss to apply to the seen classes (typically the cross-entropy loss) and $L_o[\cdot]$ is an open-set loss applied to those unseen in $S_i^u$.

**Few-shot open-set recognition.** The few-shot setting requires the sampling of the classes that make up the support and query sets. Similarly to closed-set few-shot recognition, the support set $S_i^u$ of the $i^{th}$ episode is obtained by sampling $N$ classes and $K$-examples per class. This defines the seen classes $C_i^u$. However, the query set $T_i^u$ is composed by the combination of these classes with $M$ additional unseen classes $C_i^u$. These support and query sets are used in (8).

**Large-scale open-set recognition.** In the large-scale setting, the seen classes have plenty of examples and are well trained without any need for resampling. However, resampling of the unseen classes is still advantageous, since it enables an embedding that generalizes better to these classes. In each episode, $M$ classes are randomly sampled from the class label space to form the set of unseen classes $C_i^u$, and the remaining classes are used as seen classes $C_i^*$. Without a support set $S_i^*$, the meta-learning step of (4) is no longer needed. Instead, we rely on the set of seen classes to adjust the model to only classify samples into those classes, i.e. a mapping

$$h' = M(h, C_i^u),$$

(9)

with the loss function of (8) is still applied.

**Open-set loss.** During inference, when faced with samples from unseen classes, the model should not assign a large probability to any class. In this case, a sample can be rejected if the largest class probability $\max_k p_\phi(y = k|x)$ among seen classes is small. To enable this, the learning algorithm should minimize the probabilities on seen classes for samples from $C_i^u$. This can be implemented by maximizing the entropy of seen class probabilities, i.e using the negative entropy

$$L_o[x] = \sum_{k \in C_i^u} p(y = k|x) \log p(y = k|x)$$

(10)

as loss function.

4.2. Gaussian Embedding

While PEELER is a general framework, in this work we propose an implementation based on the prototypical network architecture [29]. A set of class prototypes is first defined and samples with a large distance to the closest class prototype are assigned to the set of unseen classes. For low-shot classification, the class prototypes are the class means $\mu_i$. For large-scale classification, we assume fixed prototypes that are embedded in the network and learned by back-propagation.

Although several distances are discussed in [29], the prototypical network is usually implemented with the Euclidean distance, i.e.,

$$d(f_\phi(x), \mu_k) = (f_\phi(x) - \mu_k)^T (f_\phi(x) - \mu_k).$$

(11)

This implies that the features from each class follow a Gaussian distribution with mean $\mu_i$ and diagonal covariance $\sigma^2 I$, where $\sigma$ is shared by all classes. While sensible for closed-set few-shot learning, where the embedding is learned to produce such feature distributions, this can be very sub-optimal when open-set samples are introduced. Figure 2 illustrates the problem for a setting with three seen classes and two unseen classes. Even though, as shown in Figure 2

Figure 1. The proposed general framework for open-set meta-learning: a meta training set $\{S_i, T\}$ is sampled, where $T$ contains classes not in $S$ as “unseen” classes, and loss in (8) is minimized to obtain $h^*$.
The proposed method was compared to state-of-the-art (SOTA) approaches to open-set recognition. Following Neal et al. [21], we evaluate both classification accuracy and open-set detection performance. To clarify the terminology, closed-set classes are the categories seen during training and open-set classes the novel categories only used for testing. All training is based on the training samples from closed-set classes. For testing, we use the testing samples from both the training and open-set classes. Classification accuracy is used to measure how well the model classifies closed-set samples, i.e., test samples from closed-set classes. The AUROC (Area Under ROC Curve) metric is used to measure how well the model detects open-set samples, i.e., test samples from open-set classes, within all test samples. To simplify the writing, we define the following acronyms: our basic represents prototypical networks with euclidean distance for open-set detection; GaussianE represents the Gaussian Embedding introduced in Sec. 4.2; OpLoss represents the proposed open-set loss.

5.1. Large-Scale Open-Set Recognition

Most prior works on open-set recognition are designed for the large-scale setting. Here we evaluate PEELER on CIFAR10 [13] and extended miniImageNet [6]. CIFAR10 consists of 60,000 images of 10 classes. Following [21], 6 classes are first randomly selected as closed-set classes, and the other 4 kept as open-set classes. Results are averaged over 5 random partitions of closed/open-set classes. Extended miniImageNet is designed for few-shot learning. We use the 64 training categories and 600 images per category as closed set training data, while the 300 extra images of each category are used for closed set testing. Images from 20 test categories are used for open set testing.

Training. On CIFAR10, we randomly sample 2 from the 6 closed-set classes per episode to apply the open-set loss, the remaining 4 classes are used to train classification. On miniImageNet, the closed/open-set partition is fixed. We use Adam [12] with an initial learning rate of 0.001, $\lambda = 0.5$ in (8) and 10,000 training episodes. The learning rate is decayed by a factor of 0.1 after 6,000 and 8,000 episodes.

Results. We compare the proposed method to several open-set recognition SOTA methods, including OpenMax [1], G-OpenMax [5], and Counterfactual [21], and an out-of-distribution SOTA method, Confidence [14]. All models use the same CNN backbone for fair comparison. We tested both a CNN, denoted as ConvNet, proposed by [21], and ResNet18 [8].

Table 2 shows that, for both backbones, PEELER outperforms all previous approaches by a significant margin. For a similar classification accuracy on seen classes, it achieves
The open-set problem follows the 5-way few-shot setting. We apply OpenMax on the activations of the pre-softmax layer, i.e., the negative of the distance of our Gaussian setting. All methods are implemented with the ResNet18 [8] for fair comparison.

As shown in Table 3, both OpenMax and Counterfactual have weaker performance than the proposed approach. Note that a AUROC of 50% corresponds to chance performance. The proposed open-set loss, Gaussian embedding, and their combination all provide gains.

### 5.3. Ablation Studies

#### Sampling Strategy in Training.
We study how the training sampling strategy affects open-set detection results. The number of unseen classes used to produce open-set samples by training episode is varied, for a fixed total number of open-set samples. The corresponding open-set detection results are shown in Figure 3 (a). The performance variation is minor, when compared to the average gain over the baseline method. We hypothesize that this robustness is due to task randomness. When the total number of training episodes is large, the model converges well no matter how many open-set classes are included in a single episode.

#### Factors in Open-set Testing.
For large-scale open-set recognition, the number of open-set samples follows from the number of open-set classes. The difficulty of the open-set problem is determined by the latter. We try to discover factors that determine the difficulty of the few-shot open-set problem. A first factor is the number of classes in the open-query set. We vary this number from 1 to 10, while keeping the training procedure and the total number of samples in the open-query set unchanged. Results are shown in Figure 3 (b). A second factor is the number of samples per class, which is changed while the number of classes remains 5. Results are shown in Figure 3 (c). The figures
It has 101 videos from 10 categories with different instances, and some frames in those videos do not have the object. During training, a 5-way 1-shot few-shot open-set model is trained as described in Sec. 5.2. Instead of category level classification, we perform instance level classification of each frame. During testing, only one video is considered, using the frames annotated as relevant as the support set. This implies that only one Gaussian class center is provided. The unlabeled frames are labeled as seen or unseen by their Mahalanobis distance to the center. Frames with distance higher than a threshold are detected as irrelevant. The number of annotated relevant frames varies from 1 to 3. Results are shown as the number of misclassified frames. The best VODC method in [35] is listed for comparison. Note that VODC requires the same number of annotated irrelevant frames, which PEELER does not. The proposed method largely outperforms VODC, halving the number of misclassified frames in all three settings. This shows that its feature embedding is a better solution for open-set detection.

6. Conclusion

In this work, we have revisited the open-set recognition problem in the context of few-shot learning. We proposed an extension of meta-learning that includes an open-set loss, and a better metric learning design. The resulting classifiers provide a new state-of-the-art for few-shot open-set recognition, on mini-ImageNet. It was shown that, with few modifications, the approach can also be applied to large-scale recognition, where it outperforms state of the art methods for open-set recognition. Finally, the XJTU-Stevens Dataset was used to demonstrate the effectiveness of the proposed model on a weakly supervised object discovery task.

Acknowledgement. Gang Hua was supported partly by National Key R&D Program of China Grant 2018AAA0101400 and NSFC Grant 61629301. Bo Liu and Nuno Vasconcelos were partially supported by NSF awards IIS-1637941, IIS-1924937, and NVIDIA GPU donations.
References


[5] Zongyuan Ge, Sergey Demyanov, and Rahil Garnavi. Generative OpenMax for Multi-Class Open Set Classification. 2019. 1, 2, 6, 7


[33] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. 2016. 1, 2, 3, 7


