Actively learning in an open world

Zhiqiu Lin Bharath Hariharan
Cornell University

Abstract

Most recognition systems nowadays live in a "closed world". Such a naive assumption is however not what the reality is. In most practical scenarios, the recognition system must be able to confront an "open world" with concepts it has not yet encountered. In this paper, we formalize such a problem setup by proposing a system that "actively" combats the "open" nature of the world through interaction with an oracle, while leveraging and unifying techniques in the fields of open world, few-shot, and active learning.

1. Introduction

The past decade has seen dramatic progress in visual recognition systems. We now have recognition systems that can identify thousands, or even tens of thousands of classes. It has even been suggested that these systems are approaching human performance.

However, current systems are trained to recognize a fixed pre-determined vocabulary of concepts. The underlying assumption is that the inputs the system will encounter in deployment will consist *only* of the classes in this vocabulary. This assumption poses a problem in many practical applications because it is impossible to anticipate beforehand all the kinds of objects the perceptual system might encounter in the real world. For example, a home owner might buy some novelty furniture that is not in the vocabulary of a home robot. A self-driving car might run into an animal on the road that it has never seen before. A medical diagnosis system might encounter an extremely rare condition that it hasn't been trained on. In each of the cases the model must recognize that it has seen novel input: failure to do so can cause loss of life or property.

However, it is not merely enough to detect novel input but also learn from it so that future encounters can be handled well. For example, the next time the self-driving car sees a similar animal, it should recognize the animal and brake appropriately. Thus the recognition model must add this new concept into its vocabulary, but this introduces yet another challenge since modern systems require large amounts of training labels for this purpose. Unfortunately, collecting large amounts of training data for a concept requires time and ready access to many human annotators: neither of which a self-driving car driving on the road might have.

Motivated by these ideas we envisage a new kind of recognition system. Such a system will be initially trained with a *small core* vocabulary of concepts and will then be deployed. In deployment, it will encounter examples from a much larger set of classes, which it must autonomously discover, add to its vocabulary and learn to recognize. To do this, it will have to query an expert to acquire names for these new classes and to get training data, but it has limited access to such an expert: it will be expected to make predictions after only a few interactions with the expert. We thus have three objectives such a recognition system should satisfy: it should (a) quickly discover the full universe of classes starting from a small core, (b) learn effective recognition models not just for its initial set of classes but for all classes, and (c) achieve these goals with the least number of queries possible. We call such a system an *open world active recognition system*.

Such open world recognition models have not been built yet, but a few facets of the problem have been explored. Recognizing that a test image comes from a hitherto-unseen class is called open-set recognition and has started to see interest. Intelligently seeking out labels for images falls under the purview of active learning. To learn quickly from the limited labels acquired for the novel classes, the recognition system may have to look towards few-shot learning. However, in spite of the progress in these subfields, it is unclear how these advances should be put together to achieve the goal of open world, and what the challenges are when these disparate techniques operate in tandem.

This paper seeks to address this gap. We first formalize the problem of active open world recognition. We provide a benchmark consisting of an experimental protocol and evaluation metrics that capture the complex interactions between the three objectives described above. We then introduce a baseline framework for addressing this problem that can use as building blocks *any* state-of-the art active learning, few-shot learning and open-set recognition techniques. Finally, we analyze the performance of this framework as well as the intermediate performance of its building blocks, revealing intriguing properties of the problem and identifying key

research challenges.

2. Related work

Open set and open world recognition. In "open set" recognition, once the recognition system is trained and deployed, it should be able to detect and reject the samples that were not seen in the current vocabulary list. There has been a growing interest in this field in recent years, and many works aim to make state-of-the-art recognition systems, especially deep convolutional neural networks, capable of rejecting the "unknown" samples [19, 3, 16, 15, 4, 22, 11, 30, 17]. We refer curious readers to the review articles of recent development of open set recognition systems [9, 8] for more information.

Open world recognition [2] is one step up: it assumes that the vocabulary list of the recognition system can be grown by obtaining new labeled samples from an open world in an online or iterative fashion. This setup is conceivably more challenging than open set recognition, yet it resembles the nature of real world scenarios. The original open world recognition system however only passively receives labeled samples, retraining when enough are available. In this paper, we aimed to introduce an open world recognition system that is "active", which incorporates active learning techniques and exploits the availability of unlabeled pool of data by querying labels from the most informative samples for better recognition performance.

Active learning. Assuming accessibility to unlabeled samples, active learning techniques aim to improve the performance of recognition systems by actively querying new labeled samples from either a stream or a pool of unlabeled data with fewest budget possible. The implication is that not all samples are "equal" [25], and we refer readers to Settles' survey article [21] for a comprehensive discussion of the commonly deployed active learning algorithms. In this paper we assume we have access to a unlabeled pool of data since it is a more natural setup.

There have been recent works that tried to incorporate traditional active learning techniques into deep networks [7, 1, 29, 20, 12, 26]. Despite the variety of algorithms, most of the selection criteria can be categorized into three classes: an uncertainty-based approach (i.e. label the samples that current system is most uncertain of), a diversity-based approach (i.e. label the samples that best represent the distribution of the unlabeled pool), and expected model change (i.e. label the samples that if added to the training set will change the current system the most). As discussed in [29], the uncertainty-based approach, though intuitive, achieves the state-of-the-art performance in most cases. However most of these algorithms assume a "closed set" nature of the testing environment, while in this paper we look at an "open world" scenario. Nevertheless, we show simple ways of leveraging these techniques.

Imbalanced classification and few-shot learning. As we acquire labels for newly discovered classes, the resulting labeled set will be "imbalanced" and "few-shot", i.e. some classes in our vocabulary list might be small in terms of both the number of samples (e.g. smaller than 10 or 20) and the proportion in the training set. This could happen when the recognition system first discover a new class in the unlabeled pool that it has never seen before. There have been works aims to solve this [10, 27, 24, 6, 23, 28, 5]. In this paper, we adopted the prototypical network [23] as a baseline.

Other related tasks. There are many works that share similar flavors with the task and here we list them below to discuss about the similarity and difference.

1. **Class-incremental Learning**. Incremental learning assumes that the recognition system sees new samples from new classes in a streaming, sequential fashion. [**?**] While in our case too, new classes might appear over time, our setup differs in two ways: (a) the classes need not appear sequentially, and (b) we assume that we have the budget to store all examples the model encounters.

2. **Open set active learning**. A recent paper [14] made a first attempt to put open set recognition and active learning in the same setting. Arguably, our works are similar in that we aim to solve the same problem that is to query from an unlabeled pool which contains unseen classes. Our work is a natural and practical extension of their work, and there are some fundamental improvements in that: First, we examine not just the querying problem but also the retraining problem, which introduces interesting nuances related to class imbalance. Then, we propose a general framework for solving this problem which will be explained in later section. Furthermore, our setup is more realistic in that we have limited labelling budget at each time stamp, so we face the trade off between discovering more unseen classes or improving accuracy on the already discovered classes. None of those aspects are explored in their work, as they adopt a simpler setup by implicitly assuming the labelling budget is large enough to discover all classes at once.

3. **Open Long-tailed Recognition (OLTR)**. Our work can also be thought of as the next step up from the recent work on open long-tailed recognition [13]. In that work, they define the recognition problem to have a imbalanced distribution of head, tail, and open class samples. However, there is no active learning aspect of their work. Nonetheless, one could adopt their algorithm in our proposed framework to solve this more generalized problem, which can be a potential future work.

3. Open World Active Recognition Setup

We now formally define the problem of **Open World Active Recognition (OWAR)** and provide some evaluation metrics for this task.

3.1. Problem Setup

Let $\mathcal{X}$ be the space of images and C the set of classes. We assume that the recognition system has access to a large unlabeled pool of data D_u with classes $C_u \subset C$. There is also an oracle $O(\cdot)$ that could label any sample in D_u with a cost (e.g. a human annotator).

Note that however, we do not assume that the classes in unlabeled pool C_u is equal to C due to the open set nature of real world. D_u and C_u may also grow over time when there is access to more unlabeled data (e.g. a web scrapper that constantly downloads new images from Internet). No matter how large the unlabeled pool is, we must keep in mind that we only have limited budget for oracle and certainly could not label all samples in D_u. It might not even be a trivial task to get to know all the labels in C_u. Therefore, we denote all the labels that have been *discovered* by system to be C_d.

Practically, the recognition system should have some prior knowledge about the world, so the oracle $O(\cdot)$ will provide an initial set of labeled samples $D_0 \subset D_u$ with labels $C_0 \subset C_u$, that is the system starts with discovered classes $C_d \leftarrow C_0$. The recognition system will first train on D_0 and then be deployed. In the **deployed stage**, given an arbitrary test sample $x \in X$ with label $y \in C$, we expect the model to be able to perform:

1. **Open set detection**. If $y \notin C_d$, reject this sample as in open set.

2. **Multi-class classification**. If $y \in C_d$, do not reject and return the true label of x.

Note that there is a trade off between these two tasks: One can reject all test samples and obtain 100% success rate for open set recognition but fail miserably on multi-class classification, or one could treat all samples as if they belong to C_d. A good recognition system should be able to perform both tasks relatively well by balancing this trade off.

What makes this problem especially interesting is that we assume the recognition system is able to actively query for new labels of samples from D_u, as long as there is budget. Whenever the recognition system has labelling budget b to spend on the oracle, it enters the **query** stage where it selects b samples from D_u to be labeled.

After **query** stage, the system will enter the **retraining** stage to use the new labeled samples to improve its recognition performance.

Once the **retraining** stage is done, the system could be put back to the **deployed** stage. The only difference is that the system might observe more classes from the newly labeled sample: Assuming the classes of the queried b samples are C_b, then the discovered set of classes becomes $C_d \leftarrow C_d \cup C_b$. In general, the system is expected to iterate in this **query-retraining-deployed** fashion.

We therefore want to train a recognition system that could achieve the following goals with fewest query budget from the oracle:

1. Quickly discover all the unknown classes in unlabeled pool, i.e. we want $C_d = C_u \subseteq C$.

2. Achieve good performance on both open set recognition and multi-class classification tasks. Note that in some scenarios a system may be favorable to perform one task much better than another. Here we assume both tasks are equally important, but it should be easy to adapt it for a weighted performance of the two.

Given the goals of this active recognition system, we now formally define the evaluation metrics for this problem setup.

3.2. Evaluation Metrics

Assuming we have a labeled test set with n samples $T = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ with classes $\{y_1, \cdots, y_n\} \subseteq C$. Note that $C_d \subseteq C_u \subseteq C$. The recognition system $f(\cdot)$, given any test sample x_i, should be able to make a prediction $f(x_i) \in C_d \cup \{0\}$. We reserve 0 as the "open set" prediction, meaning the system believes that this sample belongs to the undiscovered open classes $C \setminus C_d$. The simplest evaluation metric one can think of is then the **overall accuracy** on all the test samples, which can be easily calculated as the ratio of test samples being classified correctly (i.e. $f(x_i)$ is the correct label). Note that this metric implies that if a test samples with true label $y_i \in C_u \setminus C_d$, it cannot make any contribution to the overall accuracy; thus this metric is also encouraging the system to discover as many classes in C_u as possible.

However, this metric doesn't reflect the trade off between multi-class classification and open set detection, and it could be biased towards the class distribution of the test set. For example, one could use a test set in which most classes are from the open set $C \setminus C_b$, and evaluate on a classifier that rejects any sample. This system would obtain a stellar overall accuracy, yet this result is meaningless. Therefore, we need more sophisticated methods that measures this trade off no matter how biased the test distribution is.

Generalized Open Set Classification Rate (GOSCR). We propose GOSCR as a general metric for evaluating our system performance on open set recognition. It is adapted from the *Open Set Classification Rate* curve proposed in [4]. Unlike other common open set evaluation metrics such as Receiver Operating Characteristics, which is applied in many recent works in this field [8, 14, 16, 17], this OSCR curve

directly shows the trade off between multi-class classification accuracy and open set detection accuracy in a single plot. An OSCR plot can be generated by sliding a probability threshold. The x-axis is the ratio of open-set samples failed to be rejected, and the y-axis is the multi-class classification accuracy on closed-set example. The original OSCR curve is however only applicable to open set recognition systems that use a probability threshold for rejecting open-set samples. In reality, most of the open set recognition algorithms do not provide such a well-defined probability of a sample being in open-set. Our proposed GOSCR is hence a natural extension to OSCR to make it generalizable. We also hope this metric can be adopted in future for evaluation of any open set recognition system.

The only assumption for GOSCR is that an open set recognition algorithm, given any sample x (assuming it observes closed-set classes C_{closed}), should provide:

1. A closed-set prediction label, $f(x) \in C_{closed}$, if it is not rejected.

2. A open-set score $open(x)$, with higher score indicating less likelihood of this sample belonging to C_{closed}.

Assume we have a test set $T = T_{closed} \cap T_{open}$: A test sample with label (x, y) is in T_{closed} if $y \in C_{closed}$, and is in T_{open} if $y \notin C_{closed}$. Let θ be the threshold for open-set score. The GOSCR is a plot of False Positive Rate (FPR) against Correct Classification Rate (CCR):

$$FPR(\theta) = \frac{|\{x|x \in T_{open} \wedge open(x) \geq \theta\}|}{|T_{open}|} \tag{1}$$

$$CCR(\theta) = \frac{|\{x|x \in T_{closed} \wedge f(x) = y \wedge open(x) < \theta\}|}{|T_{closed}|} \tag{2}$$

In words, the x-axis (FPR) is the ratio of open-set samples with open-set score greater than or equal to threshold θ, and the y-axis (CCR) is the ratio of closed-set samples with correct closed-set prediction and open-set score smaller than threshold θ.

4. Framework of OWAR systems

We now propose a general framework for any recognition systems aiming to solve the **OWAR** problem. We will formalize how such a system should behave and highlight challenges it faces.

Deployed stage. At deployed stage, the recognition system should be able to perform open set recognition. We formalize this task by asking the system to produce two results: (a) An open-set rejection score, with higher score means more likely to be in open classes, (b) a closed-set label prediction, i.e. if the sample is not rejected, what is its class label in C_d. Furthermore, to perform open set recognition in real time, the system should also have an operating threshold for rejection, which may change over time. We followed most prior works, by picking this operating threshold through cross validation. Note that in some of the recent work such as [16], they picked the optimal threshold through meta-learning using extreme value theory on the open-set score distribution on the training set. However this approach is not yet applicable to most open set learning techniques, so we leave the optimal threshold selection as a future work. In short, at deployed stage, the system should provide two results for each test sample x:

1. $f(x) \in C_d$, that is the closed-set predicted label in discovered classes.

2. $open_{C_d}(x) \in \mathcal{R}$, that is the open-set score indicating how likely it belongs to open or undiscovered classes $C \setminus C_d$.

Query stage. At query stage, the system wanted to select the most useful samples from D_u to be labeled under a labelling budget. The challenges are two-fold: (a) It wants to quickly discover all classes in C_u that it has not seen yet. (b) It wants to pick the most informative samples from C_d to improve its performance on already discovered classes. The system should rank all samples based on the above two criteria. Given an arbitrary unlabeled sample x, the system can compute the $open_{C_d}(x)$ score by utilizing its open set recognition module. For the second criteria, the system could similarly have another score $info_{C_d}(x)$ that indicates the informativeness of the sample assuming it belongs to discovered classes C_d. To combine the two criteria, the system just need to perform a weighted sum of the two scores. Essentially, given the unlabeled pool $D_u = \{x_1, \cdot, x_n\}$, the system need to provide:

1. Open-set scores for each sample in D_u: $\{\text{open}_1, \cdots, \text{open}_n\}$ (Normalize to $[0, 1]$).

2. Informative scores for each sample in D_u: $\{\text{info}_1, \cdots, \text{info}_n\}$. (Normalize to $[0, 1]$).

3. Weighting factors for each sample in D_u given the computed open-set and informative scores: $\{w_1, \cdots, w_n\}$. ($w_i = \Theta(\text{open}_i, \text{info}_i) \in [0, 1]$). Θ is a function that takes in both open-set score and informative score of an unlabeled sample and returns a scalar as the weighted factor between the two.

4. Finally, we rank all unlabeled sample based on the weighted sum of open-set and informative scores: $\{w_1\text{open}_1 + (1 - w_1)\text{info}_1, \cdots, w_n\text{open}_n + (1 - w_n)\text{info}_n\}$, and it asks the oracle to label the top b samples.

The idea of introducing the weighting function $\Theta(\cdot,\cdot)$ is that for any unlabeled sample, if its open-set score is high, then it is preferable to give less weight to its informative score, which is calculated based on the assumption that this sample belongs to C_d.

The benefit of designing the framework in this fashion is that it allows us to take arbitrary pairs of existing open set recognition algorithm and active learning algorithm to be put in this framework. Also, this proposed framework includes the querying strategy proposed by the recent open set active learning paper [14]. They proposed a $ULDR$ querying score, which is the ratio of unlabeled density (i.e. informative score) to labeled density (i.e. open-set score). This $ULDR$ score can be splitted into a sum of informative score and open-set score if one takes a $log(\cdot)$. We attach the proof in supplemental material.

However, in this paper we set $\Theta(\cdot,\cdot) = 0.5$ and leave advanced design of this weighting function Θ for future works (one could potentially leverage some meta-learning techniques to improve this function).

Retraining stage. Once new labeled samples are obtained, the system will enter the retraining stage. To overcome the issue of "catastrophic forgetting", the canonical solution [18] is to keep an exemplar set of previously labeled samples and to retrain on both the new samples and exemplar set. We find this design to work well in our problem setup. Hence, we allow the system to keep an exemplar set P to store the most "useful" labeled samples to be used in future retraining stage. In practice, we store all the existing labeled samples seen so far because we want to focus on the other aspects of this problem in this paper. But maintaining a smaller exemplar set under a strict budget constraint could be another future direction of this work. To summarize, the system should be able to:

1. Maintain and update an exemplar set of previously labeled samples.

2. Use new labeled samples to improve the open set recognition performance with minimum cost.

5. Design of first OWAR system

In this section, we explore different few-shot, open-set, and active learning algorithms, and propose the first practical system to solve the **OWAR** problem.

5.1. Few-shot learning techniques

To combat the few-shot learning nature of **OWAR**, we adopted two techniques:

1. **Naive training**. When new labeled samples come in, we append new weight vectors to the final linear layer of network for newly discovered classes.

2. **Prototypical training**. Following the canonical work in incremental learning [18], we also train a prototypical network via updating the mean vector for each class when new labeled samples arrive. When testing on new samples, we use the distance of its output feature to different mean vectors as the activation score for each class.

5.2. Open set recognition techniques

In order to select the open set algorithm for our proposed system, we evaluate common open set techniques using the evaluation metrics mentioned in section 5.2 on CIFAR-100. The open set algorithms that we evaluate include:

1. **Softmax threshold**. Softmax score is obtained by transforming the class activation scores by a softmax layer for a well-defined probability score for each class. The softmax threshold method simply picks the class c that achieves highest probability score and use $(1 - P(y = c|x))$ as the open-set score. Despite the simplicity, we found it to have a consistent good performance on our CIFAR-100 benchmark.

2. **Entropy threshold**. This method shares the same virtue as softmax method since they both calculate how "uncertain" the model prediction is. One can use the entropy of the probability output $\sum_{c_i \in C_d} P(y = c_i|x)log(P(y = c_i|x)$ as the open-set score, as higher the negative entropy, the more uniform the probability output is, hence the more uncertainty.

3. **OpenMax threshold**. The OpenMax is proposed in [3] to replace the regular softmax layer in neural network. It uses extreme value theory to fit Weibull distributions of distances between the output of the penultimate layer to class mean vectors, thereby revising the softmax probability score of the network and producing a well-defined open set probability score. We did a grid search of hyperparameters (such as alpha and tail size) involved in this algorithm, yet we found that its performance on CIFAR-100 is not as satisfactory. Especially, the probability score revising for closed-set classes seems to weaken its performance on multi-class classification.

4. **Sigmoid threshold**. A sigmoid network is proposed in [22] for open set detection of text documents, though it can be easily adapted to image classification. It essentially replaces the softmax layer of the network by a sigmoid functions per closed-set class (note that sigmoid function is an output between $[0, 1]$. Thus it can be seen as training a binary classifier per closed-set class while sharing the same feature extractor, i.e. the convolutional network.). The open-set score is then 1 minus the maximum sigmoid output.

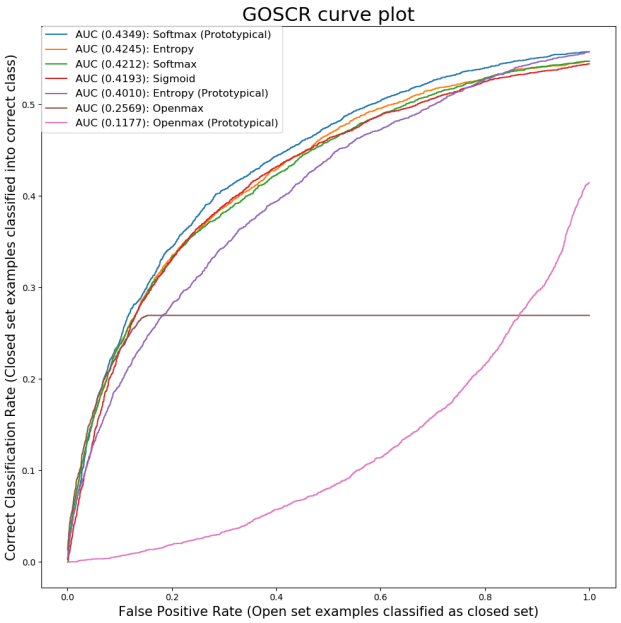


Figure 1. Evaluation of open set techniques on CIFAR-100 experiment.

For prototypical network, we use the distance between the output feature before last linear layer and the class mean vector as the activation score for each class. Note that this modification does not apply to the Sigmoid method.

We evaluate the above algorithms (note that Softmax, OpenMax, Entropy all have their prototypical couterparts) using both ROC and GOSCR metrics on CIFAR-100 with 50 closed-set classes and 50 open-set classes 1. We use random 5,000 samples from the 50 closed-set classes as the training set. We use ResNet-18 as the architecture and perform grid search of all hyper-parameters involved.

From the evaluation result we can see that:

1. Softmax, Entropy, and Sigmoid, despite their simplicity, achieve good results on our metric.

2. Openmax however is not as good when applied in this scenario. We hypothesize it is due to the revising of the probability score. Even worse its prototypical version suffers a huge decrease in performance, presumably because the activation scores computed by distance to class mean vectors are not suitable for fitting the Weibull distribution.

Hence, we choose Softmax as our baseline open set recognition algorithm for all following experiments. Adapting Openmax as well as other algorithms to work with prototypical network or other few-shot techniques can be a promising research direction in the future.

5.3. Active learning techniques

We adopted two recent canonical works in active learning domain. One is Learning Loss [29], and the other one is the Core-set Approach [20] (we applied the greedy version). The reason for adopting both is because they represent different querying heuristics:

1. Learning Loss is essentially an "uncertainty-based" approach in that it trains a loss prediction module along with the regular network in order to estimate the loss of the current sample. This criteria selects the unlabeled samples with highest predicted loss, as they are considered more "uncertain" by the network.

2. Core-set Approach on the other hand is a "distribution-based" approach. It tries to select a subset of samples from unlabeled pool D_u such that the set of selected samples can best approximate the distribution of D_u.

Despite that these two approaches are drastically different, they can both be incorporated to our above-mentioned framework. The informative score (the information an unlabeled sample contains when assuming it belongs to C_d) for Learning Loss is simply the predicted loss; and for Core-set Approach (greedy version) the score is the distance between the unlabeled sample's intermediate feature to the nearest feature among all the labeled samples.

We first replicate these algorithms and evaluate their performance on CIFAR-10 following the exact protocol (start with 1,000 samples, query 1,000 new label samples for 10 rounds) and hyper-parameters in [29]. This also implies that for learning loss method, we query from a random 10K subset of the unlabeled pool each round as that paper argues it achieves better performance. We also compare with the naive Random Query strategy. Despite that the overall test accuracy for all algorithms are slightly lower than the reported result, Learning Loss indeed achieves the best performance by a good margin 2.

Similarly, we evaluate the performance of these algorithms on CIFAR-100 using the same protocol. However, Core-set Approach seems to work better in this case when the number of classes is larger. Both canonical algorithms still outperform Random Query by a good margin 3.

Since both Learning Loss and Core-set Approach are better than random query and each has its own virtue, we adopt both as baselines for the first **OWAR** recognition system.

6. Evaluation protocol and experiment results

In this section we propose the first evaluation protocol for **OWAR** based on the well-known CIFAR-100 dataset. CIFAR-100 has 100 classes, each with 500 training samples and 100 test samples. We select 40 out of 100 classes as initial seen classes C_0, and pick 25 examples per classes as

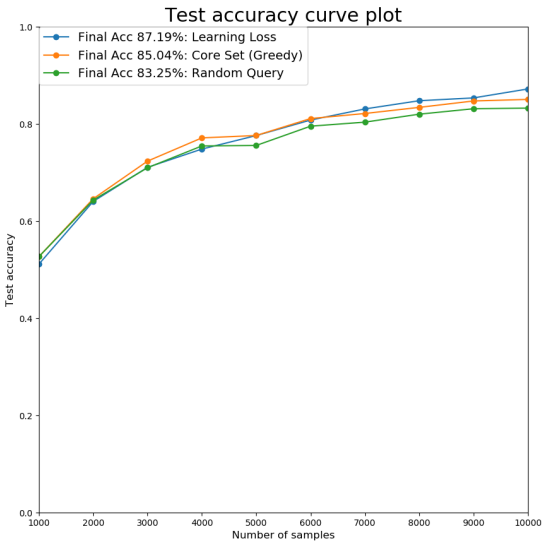


Figure 2. Evaluating active learning techniques on CIFAR-10.

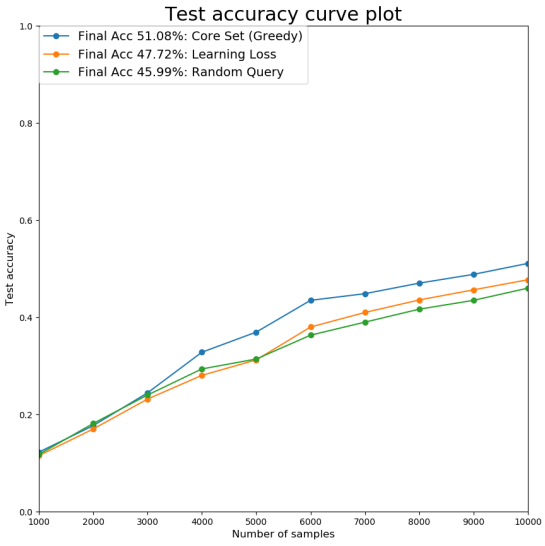


Figure 3. Evaluating active learning techniques on CIFAR-100.

the initial training set D_0. We select another 10 classes as the "hold-out" open classes and remove them from the training set. The remaining 50,000 - 1,000 - 5,000 = 44,000 training samples are the unlabeled pool D_u with $|C_U| = 90$. Then we proceed 400 iterations of **query-retraining-deployed** stages; at each new iteration the oracle has 20 query budgets.

We aim to evaluate the system performance using modern day deep network architecture. We adopt ResNet-18 as the base architecture for all the experiments, and use SGD optimizer with momentum 0.9 and L_2 weight decay of 0.0005. The learning rate is selected by grid search on a validation set for all the experiments.

We now evaluate our proposed framework on the full **query-retraining-deployed** pipeline, using Softmax as the baseline open set recognition algorithm. First of all, we discard the active query part of the system and stick to Random Query strategy in order to have a first glimpse into the performance of a recognition system in a combined setting of few-shot and open-set learning. Below we show the results of our two versions of recognition systems: Naive training and prototypical network. There is one important note here: Even though we grant our recognition system unlimited memory budget by storing all labeled samples of previous rounds in exemplar set, we do not want to retrain the feature extractor (which for ResNet is the convolutional network up to the last linear layer) on the exemplar set every round since it is uneconomical in terms of real-time deployment. Nonetheless, we observe that retraining feature extractor using the exemplar set is one of the best ways to boost the classification performance. Hence we devise a simple strategy to only retrain the feature extractor when the number of new labeled samples exceeds certain threshold. By not training the feature extractor, we can extract and store feature representations for all samples in the exemplar set: Therefore, for prototypical network training, we use these stored features to compute the class mean vectors; for naive training, we only use these features to train the last linear layer, thus improving the efficiency of the **retraining** stage by a large margin. For the below experiments, we retrain the feature extractor for every 800 new labeled samples, i.e. every 40 rounds.

Comparing Naive and Prototypical Training. We show the open set recognition performance of these two methods by calculating the area under our proposed GOSCR curve (AUC) 5.2 per round. GOSCR curve reflects the performance on both open set detection and multi-class classification and it doesn't require picking a threshold a priori, with higher AUC being more preferable.

There are several interesting results to be marked:

1. During the first few rounds, the the AUC decays due to introduction to a few-shot learning scenarios.

2. When the total number of samples is few in general, prototypical training demonstrates superior performance than naive training.

3. Whenever the feature extractor is updated, the performance increases by a great margin.

4. The two curves converge when samples become ample, suggesting that prototypical network is more useful when labeled data is scarce.

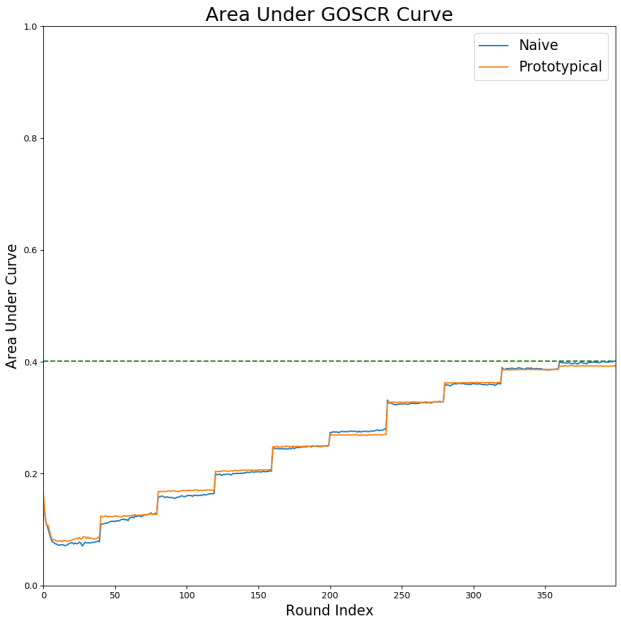


Figure 4. Area under GOSCR curve for both naive and prototypical training over 400 rounds. We retrain the feature extractor every 40 rounds.

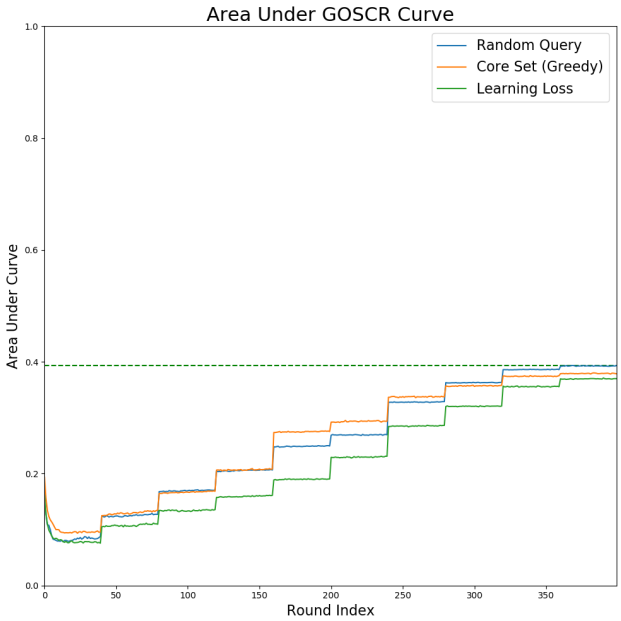


Figure 5. Comparison of different active learning strategies. Due to space limit, we only show the GOSCR metric here. The other plots such as overall accuracy, closed-set accuracy on discovered classes, and number of discovered classes can be found in supplemental material.

Exploring active querying strategy. We now explore adopting active querying into our systems. The below are results of a prototypical network training with different querying strategies. Note that as discussed in 3, the weighting function Θ for summing the open-set score and the informative score (provided by the active learning technique) is set to a constant 0.5.

There are several interesting findings to be marked:

1. Core-set Approach achieves superior results at least for the first few hundreds rounds by learning faster than other strategies.

2. In the meantime, Learning Loss behaves even worse than Random Query. We hypothesize it is because the loss prediction module is not updated in time, since we only retrain the network every 40 rounds, and the loss module could be out-dated when new samples as well as new classes are flowing in. Therefore, adapting Learning Loss with prototypical networks could be an interesting future task.

3. The three curves converge near the end, which is expected since we have a fixed size unlabeled pool, so all the active learning techniques will eventually converge.

7. Conclusion

We propose the **Open World Active Recognition (OWAR)** problem as a natural and more realistic extension to prior works in domains of open-world, few-shot, and active learning. We propose a general framework for any system solving **OWAR** problem, and through this framework we formalize and unify the different aspects of **OWAR**. We also propose new evaluation metrics for measuring the system performance in the open world scenario. We provide a benchmark for this task on CIFAR-100, and evaluate several baselines techniques on this task, in hope of pinpointing the key challenges and providing useful insights to inspire future works on this novel problem setup.

References

[1] W. H. Beluch, T. Genewein, A. N. rnbergere, and J. M. K. hlere. The power of ensembles for active learning in image classification. In *CVPR*, 2018. 2

[2] A. Bendale and T. Boult. Towards open world recognition. In *CVPR*, 2015. 2

[3] A. Bendale and T. Boult. Towards open set deep networks. In *CVPR*, 2016. 2, 5

[4] A. R. Dhamija, M. Gunther, and T. E. Boult. Reducing network agnostophobia. In *NeurIPS*, 2019. 2, 3

[5] M. Douze, A. Szlam, B. Hariharan, and H. Jégou. Low-shot learning with large-scale diffusion. In *CVPR*, 2018. 2

[6] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2

[7] Y. Gal, R. Islam, and Z. Ghahramani. Deep bayesian active learning with image data. In *ICML*, 2017. 2

[8] C. Geng, S.-J. Huang, and S. Chen. Recent advances in open set recognition: A survey v2. 2019. 2, 3

[9] C. Geng, S. jun Huang, and S. Chen. Recent advances in open set recognition: A survey. 2018. 2

[10] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. 2

[11] M. Hassen and P. K. Chan. Learning a neural-network-based representation for open set recognition. 2018. 2

[12] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang. Active self-paced learning for cost-effective and progressive face identification. In *PAMI*, 2017. 2

[13] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 2

[14] B. J. Meyer and T. Drummond. The importance of metric learning for robotic vision: Open set recognition and active learning. In *ICRA*, 2019. 2, 3, 5

[15] L. Neal, M. Olson, X. Weng, K. Wong, and F. Li. Open set learning with counterfactual images. In *ECCV*, 2018. 2

[16] P. Oza and V. M. Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *CVPR*, 2019. 2, 3, 4

[17] P. Oza and V. M. Patel. Deep cnn-based multi-task learning for open-set recognition. 2019. 2, 3

[18] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 5

[19] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. In *PAMI*, 2013. 2

[20] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 2, 6

[21] B. Settles. Active learning literature survey. 2010. 2

[22] L. Shu, H. Xu, and B. Liu. Doc: Deep open classification of text documents. 2017. 2, 5

[23] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2

[24] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016. 2

[25] K. Vodrahalli, K. Li, and J. Malik. Are all training examples created equal? an empirical study. 2018. 2

[26] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. Cost-effective active learning for deep image classification. In *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 2017. 2

[27] Y. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 2

[28] Y.-X. Wang, M. H. Ross Girshick, and B. Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018. 2

[29] D. Yoo and I. S. Kweon. Learning loss for active learning. In *CVPR*, 2019. 2, 6

[30] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura. Classification-reconstruction learning for open-set recognition. In *CVPR*, 2019. 2

Actively learning in an open world
Supplementary Material

Zhiqiu Lin Bharath Hariharan
Cornell University

1. Proof of ULDR score being included in our framework

The ULDR score proposed by [1] is the first method that ranks all the unlabeled samples in the unlabeled pool for query selection under the presence of unseen open classes. In their paper, they use a metric learning approach such that samples being closer in the metric space are deemed to have similar semantic meaning. Formally, this is saying if samples x_1 and x_2 have features extracted by an extractor being u_1 and u_2, then the (euclidean) distance between u_1 and u_2 measures the semantic similarity between x_1 and x_2. They also assume that the metric learning model they trained can be generalized well to samples in open classes. Therefore, their ULDR score is proposed as the "unlabeled" to "labeled" density as follows. Given a sample x_i with feature u_i, the labeled samples C, the unlabeled pool U, the ULDR score of x_i can be computed by:

$$D_{unlabeled}(u_i, U) = \sum_{u_j \in U, i \neq j} exp(\frac{-|u_i - u_j|^2}{2\sigma^2}) \tag{1}$$

$$D_{labeled}(u_i, C) = \sum_{c_k \in C} exp(\frac{-|u_i - c_k|^2}{2\sigma^2}) \tag{2}$$

$$ULDR(u_i, C, U) = \frac{D_{unlabeled}(u_i, U)}{D_{labeled}(u_i, C)} \tag{3}$$

Note that the σ above is a hyperparameter. The unlabeled density $D_{unlabeled}(u_i, U)$ is the Gaussian kernel sum of all the distances between u_i to other features in unlabeled pool, which measures how close is u_i to the unlabeled pool. This can be thought of as an *informative* score in our framework with the distribution-based active learning approach. On the other hand, $D_{labeled}(u_i, C)$ is the Gaussian kernel sum of all the distances between u_i to other features in labeled samples, which measures how close u_i is from the already labeled samples. This can be thought of as an *open-set* score in our framework (taking the negative version). For our framework to adapt this approach, we only need to compute the $log(\cdot)$ of this ULDR score, since it doesn't change the ranking of the samples.

$$ULDR_{log}(u_i, C, U) = log(\frac{D_{unlabeled}(u_i, U)}{D_{labeled}(u_i, C)}) = log(D_{unlabeled}(u_i, U)) + (-log(D_{labeled}(u_i, C))) \tag{4}$$

This therefore can be thought as a method in our proposed framework with a constant weighting function being 0.5. It can be a future work to evaluate the effectiveness of this ULDR score (as well as the metric learning approach adopted in their paper) in our benchmark.

2. Comparing different active learning methods on CIFAR-100

In this section we present results on evaluation metrics other than GOSCR 1 for different active learning strategies.

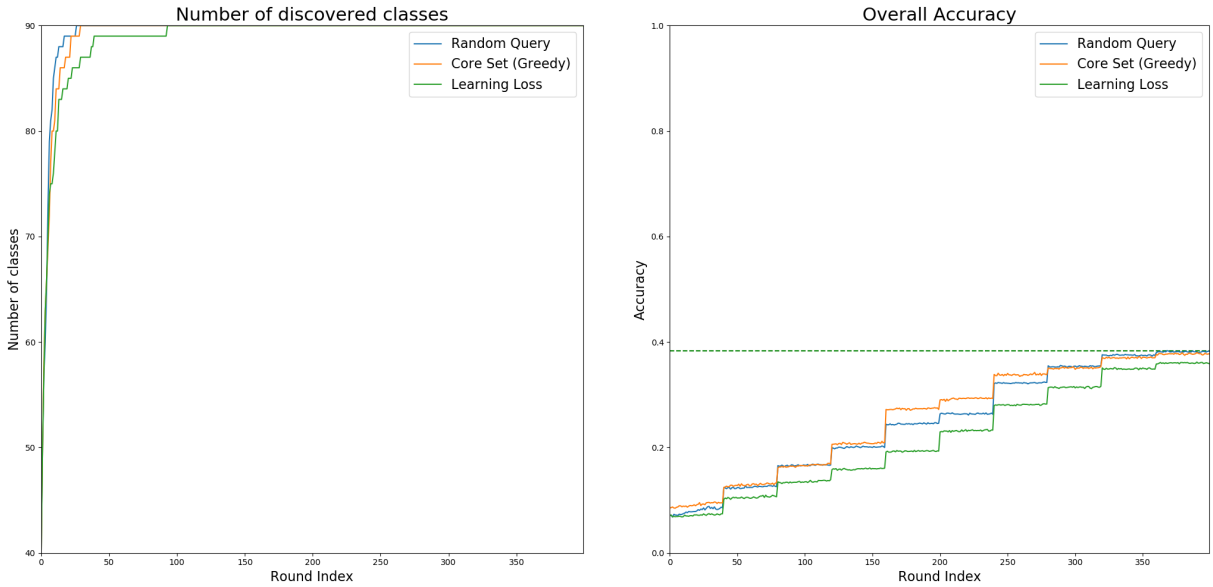


Figure 1. Comparison of different active learning strategies using prototypical training approach. Here we show both overall accuracy and number of discovered classes. Note that in order to meaningfully compare the overall accuracy, we pick the open-set threshold each round a posteriori such that the open set detection accuracy is always 0.5.

We can see that Core-set Approach indeed achieves the best performance on overall accuracy especially when labeled samples are few.

Note that Random Query actually discovers all the unseen classes in unlabeled pool first. This may be due to that we only have 50 unseen classes. In future, it is worthwhile evaluate our system on datasets with an unlabeled pool containing much more classes, such as ImageNet.

3. Investigating the effect of retraining feature extractor for prototypical network

Retraining feature extractor seems to be the key to success, as we can see that whenever we update the CNN on all exemplar samples, the performance under our metrics will be boosted. In contrast, when we do not retrain the feature extractor, the performance seems to be stuck. However, it is not the case that the system doesn't change at all during the period of no retraining. In fact, when we look at the test accuracy per class, we can see that even though the overall test accuracy does not change much, the accuracy of individual classes fluctuate drastically.

Below we show the change of class accuracy from round 124 to round 125 (using prototypical network and random query strategy). Note that even though the overall test accuracy doesn't change, the individual class accuracy fluctuates a lot as shown in the upper chart of figure 2. It can be seen from this chart that the accuracy of some classes are increasing, while some others are decreasing, which balance out such that the over accuracy only change a little.

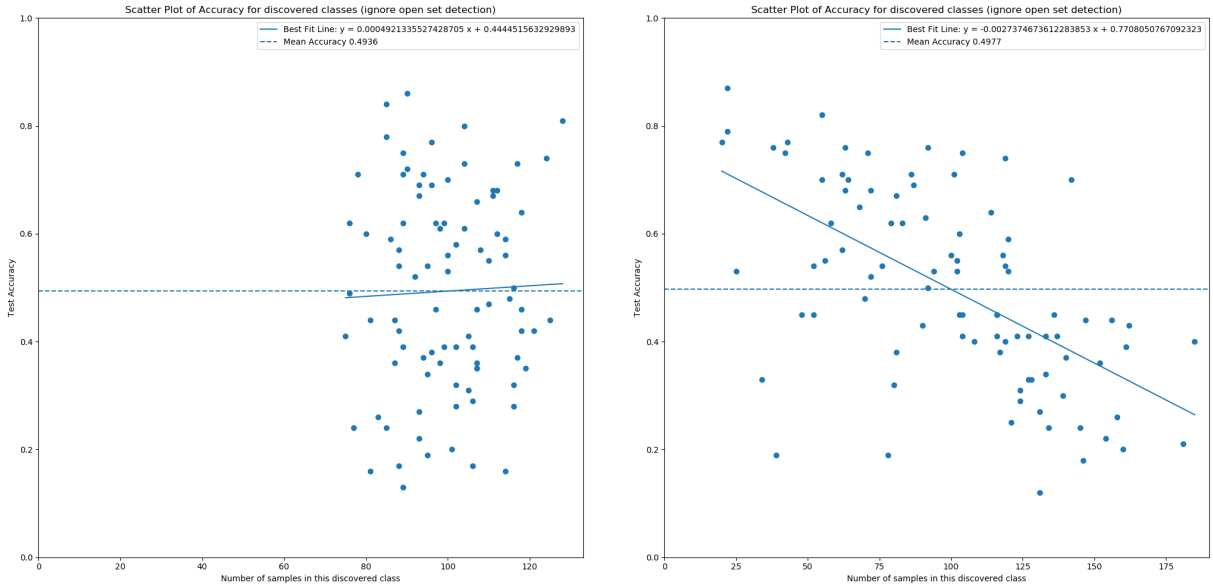
Figure 2. The change of individual class accuracy (ignoring open-set detection) from round 124 to round 125 (upper chart) and from round 120 to 121 (bottom chart). X-axis is the class index, and y-axis is the change of accuracy (centered at 0). The black dotted line shows the change of overall test accuracy from previous round. The green bars show classes that enjoy an increase in test accuracy, and red bars show classes that suffer from a decrease in accuracy. Note that we retrain the feature extractor every 40 rounds, so no retraining happens on the upper chart at round 125. Since we observe the same trend for all rounds with no retraining - here we are only picking a random round. The bottom chart is round 120 and we retrain feature extractor at this round, thus most classes enjoy a boost in accuracy.

On the other hand, when we retrain the feature extractor, all class accuracy boost with only a few decrease as shown in the bottom chart of figure 2. In future, we would like to see more powerful algorithms that can boost the class accuracy without retraining the feature extractor; or retrain the feature extractor with relatively little cost.

4. Investigating the difference between active learning and random query

It is mysterious that Core-set Approach and Random Query achieve comparable performance once the number of labeled samples becomes large, especially at the later rounds (in fact, Random Query even perform slightly better than Core-Set Approach starting from round 280). Supposedly, the performance of active learning and random query strategy should converge eventually as we see more labeled samples in the unlabeled pool. Yet out of curiosity, we look closely into the behaviors of Core-set Approach and Random Query when they are converging and make some interesting observations.

It turns out when adopting active learning techniques such Core-set Approach, the system is more likely to pick samples from the harder classes with lower accuracy. Here we show scatter plots of number of queried samples per individual class versus the class accuracy at round 399 in figure 3:



(A) Random Query (B) Active Learning (Core-set Approach)

Figure 3. Scatter plots of number of labeled samples per class (x-axis) versus class test accuracy. The dotted line in the plot is the mean accuracy for all classes. The solid line is the best fit line for the scatter plot.

From these two plots, we can see that when using random query strategy, the class accuracy is fairly random and most classes get similar number of samples. However, when we adopt active learning strategy, more budget have been dedicated to classes with lower accuracy (i.e. the harder classes). This implies that in future, one may want to design active learning strategy that assigns different weights to each class according to whether a class is easy or difficult to learn.

References

[1] B. J. Meyer and T. Drummond. The importance of metric learning for robotic vision: Open set recognition and active learning. In *ICRA*, 2019. 1