Large-Scale Open-Set Classification Protocols for ImageNet

Andres Palechor Annesha Bhoumik Manuel Günther
Department of Informatics, University of Zurich, Andreasstrasse 15, CH-8050 Zurich
https://www.ifi.uzh.ch/en/aiml.html

Abstract

Open-Set Classification (OSC) intends to adapt closed-set classification models to real-world scenarios, where the classifier must correctly label samples of known classes while rejecting previously unseen unknown samples. Only recently, research started to investigate on algorithms that are able to handle these unknown samples correctly. Some of these approaches address OSC by including into the training set negative samples that a classifier learns to reject, expecting that these data increase the robustness of the classifier on unknown classes. Most of these approaches are evaluated on small-scale and low-resolution image datasets like MNIST, SVHN or CIFAR, which makes it difficult to assess their applicability to the real world, and to compare them among each other. We propose three open-set protocols that provide rich datasets of natural images with different levels of similarity between known and unknown classes. The protocols consist of subsets of ImageNet classes selected to provide training and testing data closer to real-world scenarios. Additionally, we propose a new validation metric that can be employed to assess whether the training of deep learning models addresses both the classification of known samples and the rejection of unknown samples. We use the protocols to compare the performance of two baseline open-set algorithms to the standard SoftMax baseline and find that the algorithms work well on negative samples that have been seen during training, and partially on out-of-distribution detection tasks, but drop performance in the presence of samples from previously unseen unknown classes.

1. Introduction

Automatic classification of objects in images has been an active direction of research for several decades now. The advent of Deep Learning has brought algorithms to a stage where they can handle large amounts of data and produce classification accuracies that were beyond imagination a decade before. Supervised image classification algorithms have achieved tremendous success when it comes to detecting classes from a finite number of *known* classes, what is commonly known as evaluation under the closed-set assumption. For example, the deep learning algorithms that attempt the classification of ten handwritten digits [16] achieve more than 99% accuracy when presented with a digit, but it ignores the fact that the classifier might be confronted with non-digit images during testing [6]. Even the well-known ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [26] contains 1000 classes during training, and the test set contains samples from exactly these 1000 classes, while the real world contains many more classes, e.g., the WordNet hierarchy [19] currently knows more than 100'000 classes.[1] Training a categorical classifier that can differentiate all these classes is currently not possible – only feature comparison approaches [22] exist – and, hence, we have to deal with samples that we do not know how to classify.

Only recently, research on methods to improve classification in presence of *unknown* samples has gained more attraction. These are samples from previously unseen classes that might occur during deployment of the algorithm in the real world and that the algorithm needs to handle correctly by not assigning them to any of the known classes. Bendale and Boult [2] provided the first algorithm that incorporates the possibility to reject a sample as unknown into a deep network that was trained on a finite set of known classes. Later, other algorithms were developed to improve the detection of unknown samples. Many of these algorithms require to train on samples from some of the unknown classes that do not belong to the known classes of interest – commonly, these classes are called *known unknown* [18], but since this formulation is more confusing than helpful, we will term these classes as the *negative* classes. For example, Dhamija *et al.* [6] employed samples from a different dataset, i.e., they trained their system on MNIST as known classes and selected EMNIST

[1] https://wordnet.princeton.edu

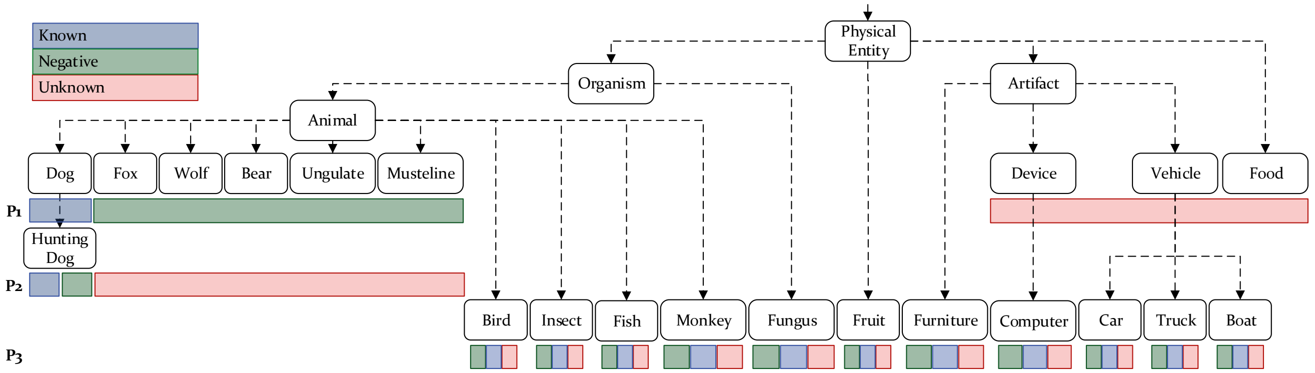

Figure 1: Class Sampling in our Open-Set Protocols. We make use of the WordNet hierarchy [19] to define three protocols of different difficulties. In this figure, we show the superclasses from which we sample the final classes, all of which are leaf nodes taken from the ILSVRC 2012 dataset. Dashed lines indicate that the lower nodes are descendants, but they might not be direct children of the upper nodes. Additionally, all nodes have more descendants than those shown in the figure. The colored bars below a class indicate that its subclasses are sampled for the purposes shown in the top-left of the figure. For example, all subclasses of “Dog” are used as known classes in protocol P_1, while the subclasses of “Hunting Dog” are partitioned into known and negatives in protocol P_2. For protocol P_3, several intermediate nodes are partitioned into known, negative and unknown classes.

letters as negatives. Other approaches try to create negative samples by utilizing known classes in different ways, e.g., Ge *et al.* [8] used a generative model to form negative samples, while Zhou *et al.* [30] try to utilize internal representations of mixed known samples.

One issue that is inherent in all of these approaches – with only a few exceptions [2, 25] – is that they evaluate only on small-scale datasets with a few known classes, such as 10 classes in MNIST [16], CIFAR-10 [14], SVHN[21] or mixtures of these. While many algorithms claim that they can handle unknown classes, the number of known classes is low, and it is unclear whether these algorithms can handle more known classes, or more diverse sets of unknown classes. Only lately, a large-scale open-set validation protocol is defined on ImageNet [28], but it only separates unknown samples based on *visual*[2] and not semantic similarity. Another issue of research on open-set classification is that most of the employed evaluation criteria, such as accuracy, macro-F1 or ROC metrics, do not evaluate open-set classification as it would be used in a real-world task. Particularly, the currently employed validation metrics that are used during training a network do not reflect the target task and, thus, it is unclear whether the selected model is actually the best model for the desired task.

In this paper we, therefore, propose large-scale open-set recognition protocols that can be used to train and test various open-set algorithms – and we will showcase the performance of three simple algorithms in this paper. We decided to build our protocols based on the well-known and well-investigated ILSVRC 2012 dataset [26], and we build three evaluation protocols P_1, P_2 and P_3 that provide various difficulties based on the WordNet hierarchy [19], as displayed in Fig. 1. The protocols are publicly available,[3] including source code for the baseline implementations and the evaluation, which enables the reproduction of the results presented in this paper. With these new protocols, we hope to foster more comparable and reproducible research into the direction of open-set object classification as well as related topics such as out-of-distribution detection. This allows researchers to test their algorithms on our protocols and directly compare with our results.

The contributions of this paper are as follows:

- We introduce three novel open-set evaluation protocols with different complexities for the ILSVRC 2012 dataset.
- We propose a novel evaluation metric that can be used for validation purposes when training open-set classifiers.
- We train deep networks with three different techniques and report their open-set performances.
- We provide all source code[3] for training and evaluation of our models to the research community.

[2] In fact, Vaze *et al.* [28] do not specify their criteria to select unknown classes and only mention *visual similarity* in their supplemental material.

[3] `https://github.com/AIML-IfI/openset-imagenet`

2. Related Work

In open-set classification, a classifier is expected to correctly classify known test samples into their respective classes, and correctly detect that unknown test samples do not belong to any known class. The study of unknown instances is not new in the literature. For example, novelty detection, which is also known as anomaly detection and has a high overlap with out-of-distribution detection, focuses on identifying test instances that do not belong to training classes. It can be seen as a binary classification problem that determines if an instance belongs to any of the training classes or not, but without exactly deciding which class [4], and includes approaches in supervised, semi-supervised and unsupervised learning [13, 23, 10].

However, all these approaches only consider the classification of samples into known and unknown, leaving the later classification of known samples into their respective classes as a second step. Ideally, these two steps should be incorporated into one method. An easy approach would be to threshold on the maximum class probability of the SoftMax classifier using a confidence threshold, assuming that for an unknown input, the probability would be distributed across all the classes and, hence, would be low [17]. Unfortunately, often inputs overlap significantly with known decision regions and tend to get misclassified as a known class with high confidence [6]. It is therefore essential to devise techniques that are more effective than simply thresholding SoftMax probabilities in detecting unknown inputs. Some initial approaches include extensions of 1-class and binary Support Vector Machines (SVMs) as implemented by Scheirer *et al.* [27] and devising recognition systems to continuously learn new classes [1, 25].

While the above methods make use only of known samples in order to disassociate unknown samples, other approaches require samples of some negative classes, hoping that these samples generalize to all unseen classes. For example, Dhamija *et al.* [6] utilize negative samples to train the network to provide low confidence values for all known classes when presented with a sample from an unknown class. Many researchers [8, 29, 20] utilize generative adversarial networks to produce negative samples from the known samples. Zhou *et al.* [30] combined pairs of known samples to define negatives, both in input space and deeper in the network. Other approaches to open-set recognition are discussed by Geng *et al.* [9].

One problem that all the above methods possess is that they are evaluated on small-scale datasets with low-resolution images and low numbers of classes. Such datasets include MNIST [16], SVHN [21] and CIFAR-10 [14] where oftentimes a few random classes are used as known and the remaining classes as unknown [9]. Sometimes, other datasets serve the roles of unknowns, e.g., when MNIST build the known classes, EMNIST letters [11] are used as negatives and/or unknowns. Similarly, the known classes are composed of CIFAR-10 and other classes from CIFAR-100 or SVHN are negatives or unknowns [15, 6]. Only few papers make use of large-scale datasets such as ImageNet, where they either use the classes of ILSVRC 2012 as known and other classes from ImageNet as unknown [2, 28], or random partitions of ImageNet [25, 24].

Oftentimes, evaluation protocols are home-grown and, thus, the comparison across algorithms is very difficult. Additionally, there is no clear distinction on the similarities between known, negative and unknown classes, which makes it impossible to judge in which scenarios a method will work, and in which not. Finally, the employed evaluation metrics are most often not designed for open-set classification and, hence, fail to address typical use-cases of open-set recognition.

3. Approach

3.1. ImageNet Protocols

Based on [3], we design three different protocols to create three different artificial open spaces, with increasing level of similarity in appearance between inputs – and increasing complexity and overlap between features – of known and unknown classes. To allow for the comparison of algorithms that require negative samples for training, we carefully design and include negative classes into our protocols. This also allows us to compare how well these algorithms work on previously seen negative classes and how on previously unseen unknown classes.

In order to define our three protocols, we make use of the WordNet hierarchy that provides us with a tree structure for the 1000 classes of ILSVRC 2012. Particularly, we exploit the `robustness` Python library [7] to parse the ILSVRC tree. All the classes in ILSVRC are represented as leaf nodes of that graph, and we use descendants of several intermediate nodes to form our known and unknown classes. The definition of the protocols and their open-set partitions are presented in Fig. 1, a more detailed listing of classes can be found in the supplemental material. We design the protocols such that the difficulty levels of closed- and open-set evaluation varies. While protocol P_1 is easy for open-set, it is hard for closed-set classification. On the contrary, P_3 is more easy for closed-set classification and more difficult in open-set. Finally, P_2 is somewhere in the middle, but small enough to run hyperparameter optimization that can be transferred to P_1 and P_3.

Table 1: ImageNet Classes used in the Protocols. This table shows the ImageNet parent classes that were used to create the three protocols. Known and negative classes are used for training the open-set algorithms, while known, negative and unknown classes are used in testing. Given are the numbers of *classes: training / validation / test* samples.

	Known	Negative	Unknown
P_1	All dog classes 116: 116218 / 29055 / 5800	Other 4-legged animal classes 67: 69680 / 17420 / 3350	Non-animal classes 166 : — / — / 8300
P_2	Half of hunting dog classes 30: 28895 / 7224 / 1500	Half of hunting dog classes 31: 31794 / 7949 / 1550	Other 4-legged animal classes 55: — / — / 2750
P_3	Mix of common classes including animals, plants and objects 151: 154522 / 38633 / 7550	Mix of common classes including animals, plants and objects 97: 98202 / 24549 / 4850	Mix of common classes including animals, plants and objects 164: — / — / 8200

In the first protocol P_1, known and unknown classes are semantically quite distant, and also do not share too many visual features. We include all 116 dog classes as known classes – since dogs represent the largest fine-grained intermediate category in ImageNet which makes closed-set classification difficult – and select 166 non-animal classes as unknowns. P_1 can, therefore, be used to test out-of-distribution detection algorithms since knowns and unknowns are not very similar. In the second protocol P_2, we only look into the animal classes. Particularly, we use several hunting dog classes as known and other classes of 4-legged animals as unknown. This means that known and unknown classes are still semantically relatively distant, but image features such as fur is shared between known and unknown. This will make it harder for out-of-distribution detection algorithms to perform well. Finally, the third protocol P_3 includes ancestors of various different classes, both as known and unknown classes, by making use of the `mixed_13` classes defined in the `robustness` library. Since known and unknown classes come from the same ancestors, it is very unlikely that out-of-distribution detection algorithms will be able to discriminate between them, and real open-set classification methods need to be applied.

To enable algorithms that require negative samples, the negative classes are selected semantically similar to the known or at least in-between the known and the unknown. It has been shown that selecting negative samples too far from the known classes does not help in creating better-suited open-set algorithms [6]. Naturally, we can only define semantic similarity based on the WordNet hierarchy, but it is unclear whether these negative classes are also structurally similar to the known classes. Tab. 1 displays a summary of the parent classes used in the protocols, and a detailed list of all classes is presented in the supplemental material.

Finally, we split our data into three partitions, one for training, one for validation and one for testing. The training and validation partitions are taken from the original ILSVRC 2012 training images by randomly splitting off 80% for training and 20% for validation. Since training and validation partitions are composed of known and negative data only, no unknown data is provided here. The test partition is composed of the original ILSVRC validation set containing 50 images per class, and is available for all three groups of data: known, negative and unknown. This assures that during testing no single image is used that the network has seen in any stage of the training.

3.2. Open-Set Classification Algorithms

We select three different techniques to train deep networks. While other algorithms shall be tested in future work, we rely on three simple, very similar and well-known methods. In particular, all three loss functions solely utilize the plain categorical cross-entropy loss $\mathcal{J}_{\text{CCE}}$ on top of SoftMax activations (often termed as the SoftMax loss) in different settings. Generally, the weighted categorical cross-entropy loss is:

$$\mathcal{J}_{\text{CCE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} w_c t_{n,c} \log y_{n,c} \tag{1}$$

where N is the number of samples in our dataset (note that we utilize batch processing), $t_{n,c}$ is the target label of the nth sample for class c, w_c is a class-weight for class c and $y_{n,c}$ is the output probability of class c for sample n using SoftMax activation:

$$y_{c,n} = \frac{e^{z_{c,n}}}{\sum\limits_{c'=1}^{C} e^{z_{c',n}}} \tag{2}$$

of the logits $z_{n,c}$, which are the network outputs.

The three different training approaches differ with respect to the targets $t_{n,c}$ and the weights w_c, and how negative samples are handled. The first approach is the plain SoftMax loss (S) that is trained on only samples from the K known classes. In this case, the number of classes $C = K$ is equal to the number of known classes, and the targets are computed as one-hot encodings:

$$\forall n, c \in \{1, \dots, C\} : t_{n,c} = \begin{cases} 1 & c = \tau_n \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $1 \leq \tau_n \leq K$ is the label of the sample n. For simplicity, we select the weights for each class to be identical: $\forall c : w_c = 1$, which is the default behavior when training deep learning models on ImageNet. By thresholding the maximum probability $\max\limits_c y_{c,n}$, cf. Sec. 3.3, this method can be turned into a simple out-of-distribution detection algorithm.

The second approach is often found in object detection models [5] which collect a lot of negative samples from the background of the training images. Similarly, this approach is used in other methods for open-set learning, such as G-OpenMax [8] or PROSER [30].[4] In this Background (BG) approach, the negative data is added as an additional class, so that we have a total of $C = K + 1$ classes. Since the number of negative samples is usually higher than the number for known classes, we use class weights to balance them:

$$\forall c \in \{1, ..., C\} : w_c = \frac{N}{CN_c} \tag{4}$$

where N_c is the number of training samples for class c. Finally, we use one-hot encoded targets $t_{n,c}$ according to (3), including label $\tau_n = K + 1$ for negative samples.

As the third method, we employ the Entropic Open-Set (EOS) loss [6], which is a simple extension of the SoftMax loss. Similar to our first approach, we have one output for each of the known classes: $C = K$. For known samples, we employ one-hot-encoded target values according to (3), whereas for negative samples we use identical target values:

$$\forall n, c \in \{1, \dots, C\} : t_{n,c} = \frac{1}{C} \tag{5}$$

Sticking to the implementation of Dhamija *et al.* [6], we select the class weights to be $\forall c$: $w_c = 1$ for all classes including the negative class, and leave the optimization of these values for future research.

3.3. Evaluation Metrics

Evaluation of open-set classification methods is a more tricky business. First, we must differentiate between validation metrics to monitor the training process and testing methods for the final reporting. Second, we need to incorporate both types of algorithms, the ones that provide a separate probability for the unknown class and those that do not.

The final evaluation on the test set should differentiate between the behavior of known and unknown classes, and at the same time include the accuracy of the known classes. Many evaluation techniques proposed in the literature do not follow these requirements. For example, computing the area under the ROC curve (AUROC) will only consider the binary classification task: known or unknown, but does not tell us how well the classifier performs on the known classes. Another metric that is often applied is the macro-F1 metric [2] that balances precision and recall for a K+1-fold binary classification task. This metric has many properties that are counter-intuitive in the open-set classification task. First, a different threshold is computed for each of the classes, so it is possible that the same sample can be classified both as one or more known classes and as unknown. These thresholds are even optimized on the test set, and often only the maximum F1-value is reported. Second, the method requires to define a particular probability of being unknown, which is not provided by two of our three networks. Finally, the metric does not distinguish between known and unknown classes, but just treats all classes identically, but consequences of classifying an unknown sample as known are different from misclassifying a known sample.

The evaluation metric that follows our intuition best is the Open-Set Classification Rate (OSCR), which handles known and unknown samples separately [6]. Based on a single probability threshold θ, we compute both the Correct Classification Rate (CCR) and the False Positive Rate (FPR):

$$\begin{aligned} \text{CCR}(\theta) &= \frac{\Big|\{x_n \mid \tau_n \leq K \wedge \operatorname*{arg\,max}\limits_{1 \leq c \leq K} y_{n,c} = \tau_n \wedge y_{n,c} > \theta\}\Big|}{|N_K|} \\ \text{FPR}(\theta) &= \frac{\Big|\{x_n \mid \tau_n > K \wedge \max\limits_{1 \leq c \leq K} y_{n,c} > \theta\}\Big|}{|N_U|} \end{aligned} \tag{6}$$

where N_K and N_U are the total numbers of known and unknown test samples, while $\tau_n \leq K$ indicates a known sample and $\tau_n > K$ refers to an unknown test sample. By varying the threshold θ between 0 and 1, we can plot the OSCR curve [6]. A closer look to (6) reveals that the maximum is only taken over the known classes, purposefully leaving out the probability of the unknown class in the BG approach.[5] Finally, this definition differs from [6] in that we use a $>$ sign for both FPR and CCR when comparing to θ, which is critical

[4] While these methods try to sample better negatives for training, they rely on this additional class for unknown samples.

[5] A low probability for the unknown class does not indicate a high probability for any of the known classes. Therefore, the unknown class probability does not add any useful information.

when SoftMax probabilities of unknown samples reach 1 to the numerical limits.

Note that the computation of the Correct Classification Rate – which is highly related to the classification accuracy on the known classes – has the issue that it might be biased when known classes have different amount of samples. Since the number of test samples in our known classes is always balanced, in our evaluation we are not affected by this bias, so we leave the adaptation of that metric to unbalanced datasets as future work. Furthermore, the metric just averages over all samples, telling us nothing about different behavior of different classes – it might be possible that one known class is always classified correctly while another class never is. For a better inspection of these cases, open-set adaptations to confidence matrices need to be developed in the future.

3.4. Validation Metrics

For validation on SoftMax-based systems, often classification accuracy is used as the metric. In open-set classification, this is not sufficient since we need to balance between accuracy on the known classes and on the negative class. While using (weighted) accuracy might work well for the BG approach, networks trained with standard SoftMax and EOS do not provide a probability for the unknown class and, hence, accuracy cannot be applied for validation here. Instead, we want to make use of the SoftMax scores to evaluate our system.

Since the final goal is to find a threshold θ such that the known samples are differentiated from unknown samples, we propose to compute the validation metric using our confidence metric:

$$\begin{gathered}\gamma^- = \frac{1}{N_N}\sum_{n=1}^{N_N}\left(1 - \max_{1\leq c\leq K} y_{n,c} + \delta_{C,K}\frac{1}{K}\right)\\ \gamma^+ = \frac{1}{N_K}\sum_{n=1}^{N_K} y_{\tau_n} \qquad \gamma = \frac{\gamma^+ + \gamma^-}{2}\end{gathered} \tag{7}$$

For known samples, γ^+ simply averages the SoftMax score for the correct class, while for negative samples, γ^- computes the average deviation from the minimum possible SoftMax score, which is 0 in case of the BG class (where $C = K + 1$), and $\frac{1}{K}$ if no additional background class is available ($C = K$). When summing over all known and negative samples, we can see how well our classifier has learned to separate known from negative samples. The maximum γ score is 1 when all known samples are classified as the correct class with probability 1, and all negative samples are classified as any known class with probability 0 or $\frac{1}{K}$. When looking at γ^+ and γ^- individually, we can also detect if the training focuses on one part more than on the other.

4. Experiments

Considering that our goal is not to achieve the highest closed-set accuracy but to analyze the performance of open-set algorithms in our protocols, we use a ResNet-50 model [12] as it achieves low classification errors on ImageNet, trains rapidly, and is commonly used in the image classification task. We add one fully-connected layer with C nodes. For each protocol, we train models using the three loss functions SoftMax (S), SoftMax with Background class (BG) and Entropic Open-Set (EOS) loss. Each network is trained for 120 epochs using Adam optimizer with a learning rate of 10^{-3} and default beta values of 0.9 and 0.999. Additionally, we use standard data preprocessing, i.e., first, the smaller dimension of the training images is resized to 256 pixels, and then a random crop of 224 pixels is selected. Finally, we augment the data using a random horizontal flip with a probability of 0.5.

Fig. 2 shows OSCR curves for the three methods on our three protocols using logarithmic FPR axes – for linear FPR axes, please refer to the supplemental material. We plot the test set performance for both the negative and the unknown test samples. Hence, we can see how the methods work with unknown samples of classes[6] that have or have not been seen during training. We can observe that all three classifiers in every protocol reach similar CCR values in the closed-set case (FPR=1), in some cases, EOS or BG even outperform the baseline. This is good news since, oftentimes, open-set classifiers trade their open-set capabilities for reduced closed-set accuracy. In the supplemental material, we also provide a table with detailed CCR values for particular selected FPRs, and γ^+ and γ^- values computed on the test set.

Regarding the performance on negative samples of the test set, we can see that BG and EOS outperform the SoftMax (S) baseline, indicating that the classifiers learn to discard negatives. Generally, EOS seems to be better than BG in this task. Particularly, in P_1 EOS reaches a high CCR at FPR=10^{-2}, showing the classifier can easily reject the negative samples, which is to be expected since negative samples are semantically and structurally far from the known classes.

When evaluating the unknown samples of the test set that belong to classes that have not been seen during training, BG and EOS classifiers drop performance, and compared to the gains on the validation set the behavior is almost similar to the SoftMax baseline. Especially when looking into P_2 and P_3, training with the negative samples does not clearly improve the open-set

[6] Remember that the known and negative sets are split into training and test samples so that we never evaluate with samples seen during training.

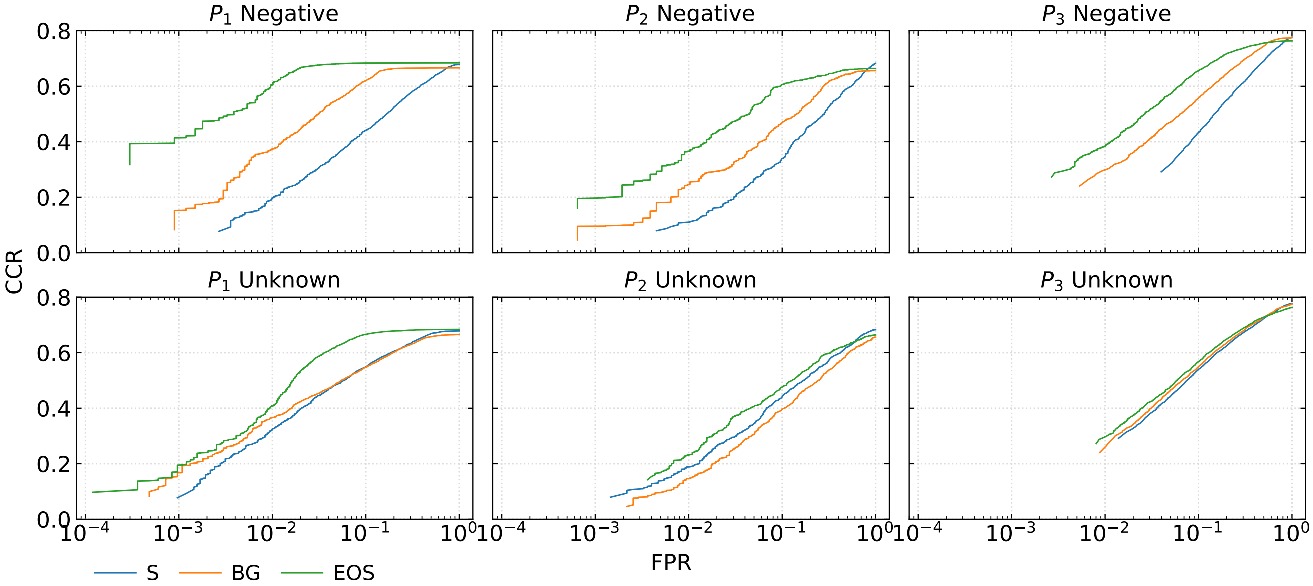

Figure 2: Open-Set Classification Rate Curves. OSCR curves are shown for test data of each protocol. The top row is calculated using negative test samples, while the bottom row uses unknown test samples. Curves that do not extend to low FPR values indicate that the threshold in (6) is maximized at $\theta = 1$.

classifiers. However, in P_1 EOS still outperforms S and BG for higher FPR, indicating the classifier learned to discard unknown samples up to some degree. This shows that the easy task of rejecting samples very far from the known classes can benefit from EOS training with negative samples, i.e., the denoted open-set method is good for out-of-distribution detection, but not for the more general task of open-set classification.

5. Discussion

After we have seen that the methods perform well on negative and not so well on unknown data, let us analyze the results. First, we show how our novel validation metrics can be used to identify gaps and inconsistencies during training of the open-set classifiers BG and EOS. Fig. 3 shows the confidence progress across the training epochs. During the first epochs, the confidence of the known samples (γ^+, left in Fig. 3) is low since the SoftMax activations produce low values for all classes. As the training progresses, the models learn to classify known samples, increasing the correct SoftMax activation of the target class. Similarly, because of low activation values, the confidence of negative samples (γ^-, right) is close to 1 at the beginning of the training. Note that EOS keeps the low activations during training, learning to respond only to known classes, particularly in P_1, where values are close to 1 during all epochs. On the other hand, BG provides lower confidences for negative samples (γ^-). This indicates that the class balancing technique in (4) might have been too drastic and that higher weights for negative samples might improve results of the BG method. Similarly, employing lower weights for the EOS classifier might improve the confidence scores for known samples at the cost of lower confidence for negatives. Finally, from an open-set perspective, our confidence metric provides insightful information about the model training; so far, we have used it to explain the model performance, but together with more parameter tuning, the joint γ metric can be used as criterion for early stopping as shown in the supplemental material.

We also analyze the SoftMax scores according to (2) of known and unknown classes in every protocol. For samples from known classes we use the SoftMax score of the correct class, while for unknown samples we take the maximum SoftMax score of any known class. This enables us to make a direct comparison between different approaches in Fig. 4. When looking at the score distributions of the known samples, we can see that many samples are correctly classified with a high probability, while numerous samples provide almost 0 probability for the correct class. This indicates that a more detailed analysis, possibly via a confusion matrix, is required to further look into the details of these errors, but this goes beyond the aim of this paper.

More interestingly, the distribution of scores for unknown classes differ dramatically between approaches

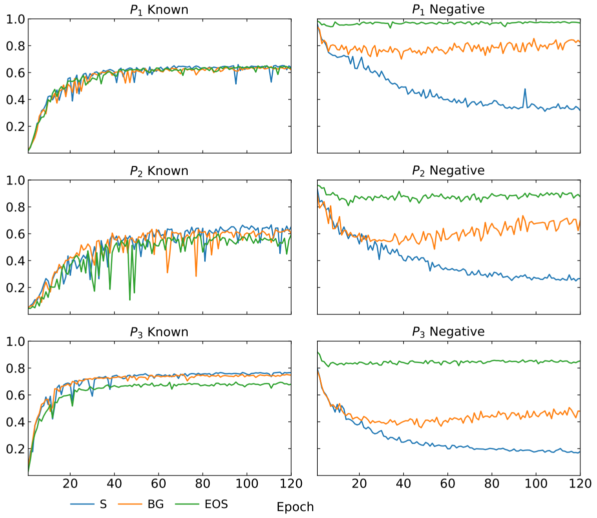

Figure 3: Confidence Propagation. Confidence values according to (7) are shown across training epochs of S, BG and EOS classifiers. On the left, we show the confidence of the known samples (γ^+), while on the right the confidence of negative samples (γ^-) is displayed.

and protocols. For P_1, EOS is able to suppress high scores almost completely, whereas both S and BG still have the majority of the cases providing high probabilities of belonging to a known class. For P_2 and, particularly, P_3 a lot of unknown samples get classified as a known class with very high probability, throughout the evaluated methods. Interestingly, the plain SoftMax (S) method has relatively high probability scores for unknown samples, especially in P_2 and P_3 where known and unknown classes are semantically similar.

6. Conclusion

In this work, we propose three novel evaluation protocols for open-set image classification that rely on the ILSVRC 2012 dataset and allow an extensive evaluation of open-set algorithms. The data is entirely composed of natural images and designed to have various levels of similarities between its partitions. Additionally, we carefully select the WordNet parent classes that allow us to include a larger number of known, negative and unknown classes. In contrast to previous work, the class partitions are carefully designed, and we move away from implementing mixes of several datasets (where rejecting unknown samples could be relatively easy) and the random selection of known and unknown classes inside a dataset. This allows us to differentiate between methods that work well in out-of-distribution detection, and those that really perform open-set classification. A more detailed comparison of the protocols is provided in the supplemental material.

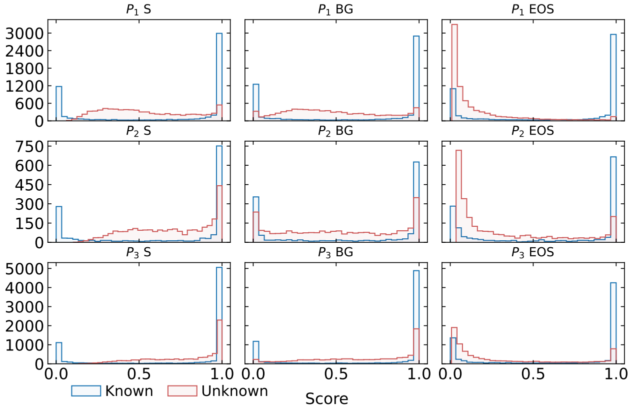

Figure 4: Histograms of SoftMax Scores. We evaluate SoftMax probability scores for all three methods and all three protocols. For known samples, we present histograms of SoftMax score of the target class. For unknown samples, we plot the maximum SoftMax score of any known class. For S and EOS, the minimum possible value of the latter is $\frac{1}{K}$, which explains the gaps on the left-hand side.

We evaluate the performance of three classifiers in every protocol using OSCR curves and our proposed confidence validation metric. Our experiments show that the two open-set algorithms can reject negative samples, where samples of the same classes have been seen during training, but face a performance degradation in the presence of unknown data from previously unseen classes. For a more straightforward scenario such as P_1, it is advantageous to use negative samples during EOS training. While this result agrees with [6], the performance of BG and EOS in P_2 and P_3 shows that these methods are not ready to be employed in the real world, and more parameter tuning is required to improve performances. Furthermore, making better use of or augmenting the negative classes also poses a challenge in further research in open-set methods.

Providing different conclusions for the three different protocols reflects the need for evaluation methods in scenarios designed with different difficulty levels, which we provide within this paper. Looking ahead, with the novel open-set classification protocols on ImageNet we aim to establish comparable and standard evaluation methods for open-set algorithms in scenarios closer to the real world. Instead of using low-resolution images and randomly selected samples from CIFAR, MNIST or SVHN, we expect that our open-set protocols will establish benchmarks and promote reproducible research in open-set classification. In future work, we will investigate and optimize more different open-set algorithms and report their performances on our protocols.

References

[1] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[2] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[3] Annesha Bhoumik. Open-set classification on ImageNet. Master's thesis, University of Zurich, 2021.

[4] Paul Bodesheim, Alexander Freytag, Erik Rodner, and Joachim Denzler. Local novelty detection in multi-class recognition problems. In *Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[5] Akshay Dhamija, Manuel Günther, Jonathan Ventura, and Terrance E. Boult. The overlooked elephant of object detection: Open set. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[6] Akshay Raj Dhamija, Manuel Günther, and Terrance E. Boult. Reducing network agnostophobia. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[7] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019.

[8] Zongyuan Ge, Sergey Demyanov, and Rahil Garnavi. Generative OpenMax for multi-class open set classification. In *British Machine Vision Conference (BMVC)*, 2017.

[9] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(10):3614–3631, 2021.

[10] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[11] Patrick Grother and Kayee Hanaoka. NIST special database 19 handprinted forms and characters 2nd edition. Technical report, National Institute of Standards and Technology (NIST), 2016.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[16] Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. The MNIST database of handwritten digits, 1998.

[17] Ofer Matan, R.K. Kiang, C.E. Stenard, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, and Yann Le Cun. Handwritten character recognition using neural network architectures. In *USPS Advanced Technology Conference*, 1990.

[18] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[19] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[20] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *European Conference on Computer Vision (ECCV)*, 2018.

[21] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems (NIPS) Workshop*, 2011.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[23] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[24] Ryne Roady, Tyler L. Hayes, Ronald Kemker, Ayesha Gonzales, and Christopher Kanan. Are open set classification methods effective on large-scale datasets? *PLOS ONE*, 15(9), 2020.

[25] Ethan M. Rudd, Lalit P. Jain, Walter J. Scheirer, and Terrance E. Boult. The extreme value machine. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.

[27] Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Towards open set recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7), 2013.

[28] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zissermann. Open-set recognition: A good closed-set classifier is all you need? In *International Conference on Learning Representations (ICLR)*, 2022.

[29] Yang Yu, Wei-Yang Qu, Nan Li, and Zimin Guo. Open-category classification by adversarial sample generation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[30] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.