Difficulty-Aware Simulator for Open Set Recognition

WonJun Moon, Junho Park, Hyun Seok Seong, Cheol-Ho Cho, and Jae-Pil Heo[⋆]

Sungkyunkwan University
{wjun0830,pjh4993,gustjrdl95,gersys,jaepilheo}@skku.edu

Abstract. Open set recognition (OSR) assumes unknown instances appear out of the blue at the inference time. The main challenge of OSR is that the response of models for unknowns is totally unpredictable. Furthermore, the diversity of open set makes it harder since instances have different difficulty levels. Therefore, we present a novel framework, DIfficulty-Aware Simulator (DIAS), that generates fakes with diverse difficulty levels to simulate the real world. We first investigate fakes from generative adversarial network (GAN) in the classifier's viewpoint and observe that these are not severely challenging. This leads us to define the criteria for difficulty by regarding samples generated with GANs having moderate-difficulty. To produce hard-difficulty examples, we introduce Copycat, imitating the behavior of the classifier. Furthermore, moderate- and easy-difficulty samples are also yielded by our modified GAN and Copycat, respectively. As a result, DIAS outperforms state-of-the-art methods with both metrics of AUROC and F-score. Our code is available at https://github.com/wjun0830/Difficulty-Aware-Simulator.

Keywords: Open Set Recognition, Unknown Detection

1 Introduction

Thanks to the advance of convolutional neural network (CNN), downstream tasks of computer vision have been through several breakthroughs [14, 24]. Although the performance of deep learning is now comparable to that of humans, distilling knowledge learned from known classes to detect unseen categories lags behind [10, 13]. Hence, unseen categories are often misclassified into one of the known classes. In this context, open set recognition was proposed to learn for the capability of detecting unknowns [2, 3, 36].

Generally, CNN are often highly biased to yield high confidence scores [32]. However, calibrating the confidence to distinguish open set data is infeasible due to the inaccessibility of those unseen data in the training phase [12]. Furthermore, it is known that the learned CNN classifiers tend to highly rely on the discriminative features to distinguish classes but ignore the other details [33]. Therefore, open set instances that share such discriminative features with closed set can be

[⋆] Corresponding author

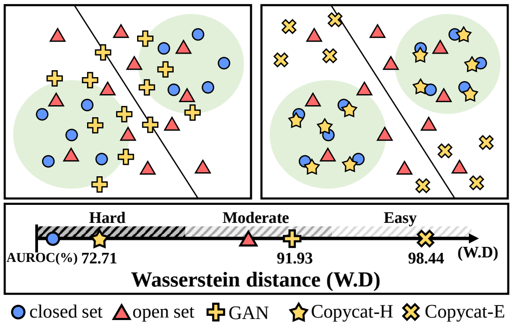


Fig. 1. Given a set of closed and open classes, we intuitively describe how the classifier recognizes fake examples from the image-generator and the Copycat. Green circles indicate the confident class boundaries. (Left) Both open set and generated instances by GAN usually project nearby the class boundaries or sometimes inside. (Right) Hard-fake instances produced by the Copycat embed deep inside the class boundaries, while easy ones usually embed far. (Bottom) Each set of fakes are represented on a line based on normalized Wasserstein distances (W.D) to closed set with the corresponding AUROC score. To measure W.D and AUROC scores, we conduct a primary experiment on CIFAR10 with the classifier trained only on closed set. In this paper, we define the difficulty levels according to two measures from the perspective of the classifier.

easily confused. In other words, open set may have a level of difficulty, which is determined according to the degrees of feature sharing with the closed set. In this regard, it is challenging to cope with all open set with various difficulty levels which can be encountered in the real world.

Due to the inaccessibility to open set during training, substantial efforts were put to simulate virtual open set [4,5,8,28,47]. RPL [5] applied 1-vs-rest training scheme and exploited features from other classes to form open set space, and PROSER [47] mixed features to simulate open set. Furthermore, GAN [11] is actively used to synthesize unknown samples [4,8,28]. However, diverse difficulty levels of open set are not taken into account. Specifically, feature simulation methods only utilize the features outside the class boundaries which are easy to be distinguished [5,47]. Besides, image generation-based methods mostly produce samples being predicted as unknown class by the classifier to represent open set [4,8,27,28]. These samples hardly have high difficulty, so the classifiers learned with them can be still vulnerable to difficult open set that contains semantically meaningful features of one of the known classes [9,30].

In this context, we propose a novel framework, DIfficulty-Aware Simulator (DIAS), that exposes diverse samples to the classifier with various difficulty levels from the classifier's perspective. As shown in Fig. 1, we found that a set of generated images with GAN is not very challenging for the classifier. Therefore, with the GAN as a criterion, we define the difficulty levels and introduce the Copycat, a feature generator producing hard fake instances. As the training iteration proceeds, the Copycat mimics the behavior of the classifier and generates real-like fake features that the classifier will likely yield a high probability. In other words, the classifier faces unknown features within its decision boundaries at every iteration. In this way, the classifier is repeatedly exposed to confusing instances and learns to calibrate even within the class boundaries. Moreover, we further ask the Copycat to create easy fake samples and also modify the image-level generator to take the classifier's perspective into account. These fake instances are additionally utilized to simulate the real world in which unseen examples with various difficulties may exist. Besides, DIAS is inherently equipped with a decent threshold to distinguish open set. It enables to avoid expensive process to search an appropriate confidence threshold of the classifier for OSR.

In summary, our contributions are: (i) We propose a novel framework, DIAS, for difficulty-aware open set simulation from the classifier's perspective. To the best of our knowledge, this is the first attempt to consider the difficulty level in OSR. (ii) We present Copycat, the difficult fake feature generator, by imitating the classifier's behavior with the distilled knowledge. (iii) We prove effectiveness with competitive results and demonstrate feasibility with an inherent threshold to identify open set samples.

2 Background and Related Works

Open set recognition To apply the classification models to real world with high robustness, OSR was first formalized as a constrained minimization problem [36]. Following them, earlier works used traditional approaches: support vector machines, Extreme Value Theory (EVT), nearest class mean classifier, and nearest neightbor [2,20,21,35,45]. Then, along with the development in CNN, deep learning algorithms have been widely adopted. In the beginning, softmax was tackled for its closed nature. To replace this, Openmax [3] tried to extend the classifier and K-sigmoid [37] conducted score analysis to search for threshold.

A recently popular stream for OSR is employing generative models to learn a representation that only preserves known samples. Conditional Variational AutoEncoder (VAE) was utilized in C2AE [31] to model the reconstruction error based on the EVT. CGDL [38] improved the VAE's weakness in closed set classification with conditional gaussian distribution learning. Moreover, flow-based model was employed for density estimation [46] and capsule network was adopted to support representation learning with conditional VAE [13]. Other approaches exploited the generative model's representation as an additional feature. GFROSR [32] employed reconstructed image from an autoencoder to augment the image while CROSR [42] adopted ladder network to utilize both the prediction and the latent features for unknown detection.

Other methods mostly fall into the category of simulating unknown examples, a more intuitive way for OSR. RPL [5] tried to conduct simulation with prototype learning. With prototypes, they designed an embedding space for open set at the center where the samples will yield low confidence scores. Then, based on manifold mixup [40], PROSER [47] set up the open space between class boundaries to keep each boundary far from others. GAN was also employed to simulate open set.

G-openmax [8] improved openmax [3] via generating extra images to represent the unknown class and OSRCI [28] developed encoder-decoder GAN architecture to generate counterfactual examples. Additionally, ARPL [4] enhanced prototype learning with generated fake samples and GAN was further extended to feature space [22]. DIAS shares similarity with these methods in that we simulate open set. However, the main difference comes from the consideration of difficulty gaps between open set instances from the classifier's perspective.

Multi-level knowledge distillation Knowledge Distillation (KD) was introduced in [17] where the student learns from the ground-truth labels and the soft-labels from the teacher. AT [44] exploited attention maps in every layer to transfer the knowledge and FSP [41] utilized the FSP matrix that contains the distilled knowledge from the teacher. Moreover, cross-stage connection paths were also introduced to overcome information mismatch arising from differences in model size [6]. Copycat share similar concept with multi-stage KD methods that it imitates encoding behavior of the classifier. Up to we know, developing a fake generator with KD is a novel strategy in the literature of OSR.

3 Methodology

3.1 Problem Formulation and Motivation

The configuration of OSR is different from classification since models can face unseen objects at the inference time. Suppose that a model trained with $\mathcal{D}_{tr} = (X, Y)$ over a set of classes K. X is a set of input data $\mathbf{x}$ and Y is a set of one-hot labels which each sample $\mathbf{y} \in \{0, 1\}^K$, where its value is 1 for the ground-truth class and 0 for the others. A typical classification evaluates the trained model on $\mathcal{D}_{te} = (T, Y)$ where X and T are sampled from the same set of classes.

On the contrary, $\hat{\mathcal{D}}_{te} = (\hat{T})$ of OSR contains instances over a novel category set $\hat{K}$. Since the conventional classifier assumes K categories, common approach is to distinguish open set based on confidence thresholding [16]. However, challenges in OSR come from the similarity between $\mathcal{D}_{te}$ and $\hat{\mathcal{D}}_{te}$ which often leads to high confidence scores for both. Furthermore, the diverse relationships between $\mathcal{D}_{te}$ and $\hat{\mathcal{D}}_{te}$ affect the threshold to be vulnerable to different datasets. To this end, the classifier is preferred to be calibrated to yield low score on unknown classes.

Intuitive approach to calibrate the classifier is to simulate the $\hat{\mathcal{D}}_{te}$ with fake set $\bar{X}$. Then, while the classifier is trained on $\mathcal{D}_{tr}$, it is also enforced to suppress the output logit of the fake sample $\bar{\mathbf{x}}$ to a uniform probability distribution:

$$\mathcal{L} = \mathcal{L}_{ce}(\mathbf{x}, \mathbf{y}) + \lambda \cdot \mathcal{L}_{ce}(\bar{\mathbf{x}}, 1/K \cdot \mathbf{u}), \tag{1}$$

where $\mathcal{L}_{ce}$, λ, and $\mathbf{u}$ each denotes cross-entropy loss, scale parameter for fake sample calibration, and the all-one vector.

Nevertheless, due to the nature of the real world where unknown classes are rampant, a single set of fakes cannot represent all unknowns [18]. This is because the difficulty levels of data in the perspective of the classifier can be significantly varying [34, 39]. This motivates us to develop a simulator to rehearse with fake

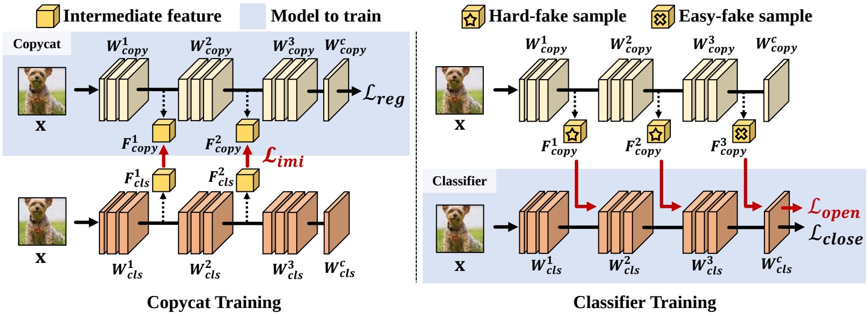


Fig. 2. Illustration of the joint training scheme between the Copycat and the classifier. (Left) Knowledge is distilled from the classifier to the Copycat while its parameters are also updated by the regularization loss. (Right) The classifier is exposed to fakes generated by the Copycat. Depending on the existence of the imitation loss $\mathcal{L}_{imi}$ in its group (W^1, W^2, W^3), each convolutional group of the Copycat outputs hard- or easy-fake instances. $\mathcal{L}_{open}$ is a general term for the loss computed from fake examples.

examples in diverse difficulty levels. To overcome the new challenge in generating the hard-fake instances, we first introduce Copycat learning which generates hard fakes in Sec. 3.2 and describe the DIAS in Sec. 3.3. Finally, we explain the inherent threshold of DIAS which benefits inference procedure in Sec. 3.4

3.2 Copycat : Hard-fake feature generator

In Fig. 2, we illustrate how the Copycat interacts with the classifier to introduce fake examples. Note that, throughout the paper, the terms regarding the difficulty level (e.g. Hard, Moderate, and Easy) are the data difficulty from the perspective of the classifier, as in Fig. 1. Given an input image $\mathbf{x}$ and network W, we let $\mathbf{p} = W(\mathbf{x})$ stand for the output probability. W can be separted into different parts $(W^1, \cdots, W^n, W^c)$, where W^c is the fully-connected layer with the softmax activation and $W^1, \cdots, W^n$ are different groups of layers separated by predefined criteria. Then, predicting $\mathbf{p}$ with network W can be expressed as

$$\mathbf{p} = W^c \circ W^n \circ \cdots \circ W^1(\mathbf{x}). \tag{2}$$

We refer to "$\circ$" as nesting of functions where $g \circ f(x) = g(f(x))$. Note that the overall architecture for the Copycat and the classifier is equivalent and layers are grouped with the same criteria. For simplicity, we split the Copycat and the classifier into three groups of layers. This can be easily applied to other models [15,19] as the number of groups can be adjusted. We also let intermediate features to be denoted as $(F^1, \cdots, F^n)$. For instance, ith feature is calculated as:

$$F^i = W^i \circ W^{i-1} \circ \cdots \circ W^1(\mathbf{x}). \tag{3}$$

The key objective of the Copycat is to create virtual features of hard- and easy-difficulty levels. To introduce the training procedure, we first define I as an index set for convolutional groups which is subdivided into I_{hard} and I_{easy}. I_{hard} and I_{easy} imply the difficulty level of fake features that each convolutional group outputs. Then, we train the Copycat to mimic the classifier's knowledge at I_{hard} and to differ from the classifier at I_{easy} with the loss function formulated as:

$$\mathcal{L}_{copy} = \mathcal{L}_{reg} + \mathcal{L}_{imi}, \tag{4}$$

where $\mathcal{L}_{reg}$ denote a regularization loss and $\mathcal{L}_{imi}$ is an imitation loss. Note that while $\mathcal{L}_{reg}$ is for all layers in the Copycat, $\mathcal{L}_{imi}$ is only updated to layers in I_{hard}. Therefore, the Copycat gets to behave similar to the classifier at I_{hard}.

Without loss of generality, we state that the imitation losses are placed between the features of the Copycat and the classifier where they are of the same convolutional group index. We define the imitation loss as:

$$\mathcal{L}_{imi} = \sum_{j \in I_{hard}} \|F^j_{copy} - F^j_{cls}\|_1, \tag{5}$$

where F^j_{copy} and F^j_{cls} are the feature vectors from j-th convolutional group of the Copycat W^j_{copy} and the classifier W^j_{cls}. Then, with $\mathcal{L}_{imi}$, forcing the convolutional groups at I_{hard} of the Copycat to behave similarly to classifier's, hard layers in the Copycat become to produce difficult fake features from the classifier's perspective. However, if I_{easy} is defined in front of I_{hard}, corrupted features from I_{easy} can lead to unstable $\mathcal{L}_{imi}$ because I_{easy} is to yield abnormal features that are different from the ones of the classifier. Thus, as the quality of hard fakes is crucial in the Copycat, we define I_{easy} at the last.

Regularization loss has two purposes: hindering the replication procedure to prevent Copycat from being exactly the same as the classifier and diversifying easy fakes. Since the groups at I_{easy} of the Copycat do not have any connectivity to the classifier and are updated only with $\mathcal{L}_{reg}$, features from I_{easy} of the Copycat would be abnormal which are easy to be identified by the classifier. Moreover, as $\mathcal{L}_{reg}$ is iteratively applied, diverse easy fakes are produced. For $\mathcal{L}_{reg}$, we simply use $\mathcal{L}_{ce}$ with real labels since the classification loss plays two roles well.

3.3 Difficulty-Aware Simulator

The key idea of DIAS is to expose the classifier with open set of various difficulty levels. To consider the classifier's perspective in real-time, we apply joint scheme of the training phase between each generative model and the classifier.

Stage I : Generator Training. To come up with virtual open set for simulation, we employ a Copycat and an image-generator. As we discussed in Sec. 3.2, the Copycat is to prepare hard- and easy-fake features for robust training of the classifier. Moreover, inspired by Fig. 1, we employ GAN to generate moderate-level fake images but with a little modification to consider the classifier's viewpoint.

To synthesize fake images of the moderate difficulty, the generator is adversarially trained with the discriminator. Let D and G be the discriminator and the generator, respectively. The generator receives the noise vector $\mathbf{z}$ sampled from

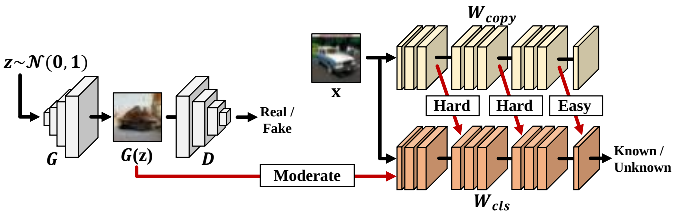


Fig. 3. Illustration of the proposed DIAS. Our GAN and Copycat each receives the noise vector z and input image x to produce fake instances with various difficulties. These instances are provided to the classifier to have a rehearsal for OSR.

normal distribution $\mathcal{N}$ and generates the output as $G(\mathbf{z})$. Then, the discriminator encodes both the image $\mathbf{x}$ and fake images $G(\mathbf{z})$ to the probability $[0,1]$ whether to predict the input is from a real distribution or not.

Furthermore, our focus of generating fake images is to prepare the classifier for open set. Thus, optimization process for the generator should consider the classifier's prediction. However, arranging the process should be carefully designed to avoid two situations. On the one hand, when the perfect generator is trained, the classifier would not be capable of discriminating open set due to the equilibrium in the mini-max game [1,22]. On the other hand, when the generator only outputs images that the classifier predicts with the uniform distribution, generated samples would not be representative for simulating moderate-difficulty open set. Therefore, we encourage the generator to output closed set-like images, but very cautiously, with the classifier's predictions. Specifically, we use the negative cross-entropy loss to a uniform target vector. Accordingly, the generator is trained to produce fake images that are neither the known nor the outlier from the classifier's perspective. Formally, the overall optimization process of our GAN is conducted by:

$$\begin{aligned}\max_D \min_G \quad \mathcal{L}_{gan} &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{tr}}\big[\log D(\mathbf{x})\big] \\ &+ \mathbb{E}_{\mathbf{z}\sim\mathcal{N}}\big[\log\big(1 - D(G(\mathbf{z}))\big) - \beta/K\sum_{k=1}^{K}\log C_k\big(G(\mathbf{z})\big)\big]\end{aligned} \tag{6}$$

where β is the scale parameter for negative cross-entropy and $C_k(\cdot)$ outputs probability of class k for given input.

Stage II : Classifier Training. Facing all kinds of difficulty through virtual open set at training time, we pursue to drive the classifier to be prepared for handling unseen classes. Thus, we formulate the loss for the classifier as:

$$\mathcal{L}_{cls} = \mathcal{L}_{close} + \lambda \cdot \mathcal{L}_{open}, \tag{7}$$

where $\mathcal{L}_{close}$ and $\mathcal{L}_{open}$ are to discriminate within known and between known and unknown classes, respectively. To consider the difficulty levels in the classifier's behavior, we need to differentiate the objectives with their difficulties. Therefore, we employ the $\mathcal{L}_{ce}$ for both but with the smoothed label for calculating the $\mathcal{L}_{open}$. The smoothed label $\tilde{\mathbf{y}}$ is formed with smoothing ratio α as below:

$$\tilde{\mathbf{y}} = (1-\alpha)\cdot\mathbf{y} + \alpha/K\cdot\mathbf{u}. \tag{8}$$

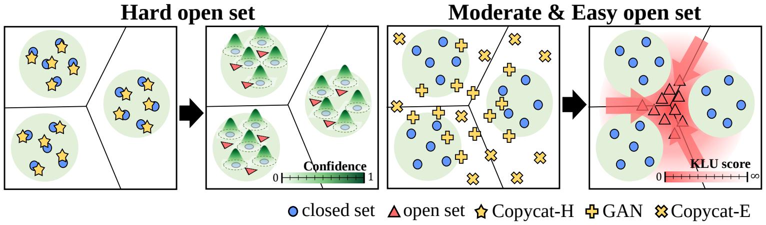


Fig. 4. Illustration about the changes in the classifier's perspective by facing fake open set. Confidence bar represents $\mathbf{p}$ and KLU score bar stands for the KL-divergence to the uniform distribution. (Left) Hard-difficulty open set samples are mostly located within the decision boundary of the classifier. They lead the classifier to calibrate its find-grained confidence distribution to distinguish such hard-difficulty samples. (Right) Moderate- and easy-difficulty samples are likely located near the boundary or free space. The classifier tries to gather them to the region with the uniform probability.

Generated fakes are encoded by the classifier to make a prediction as illustrated in Fig. 3. Specifically, intermediate features from the Copycat and fake images from GAN are passed on to the classifier to predict $\mathbf{p}^i_{copy}$ and $\mathbf{p}_{gan}$ as:

$$\mathbf{p}^i_{copy} = W^c_{cls} \circ W^n_{cls} \circ \cdots \circ W^{i+1}_{cls}(F^i_{copy}) \tag{9}$$

$$\mathbf{p}_{gan} = W^c_{cls} \circ W^n_{cls} \circ \cdots \circ W^1_{cls}(G(z)) \tag{10}$$

Then, open loss is calculated alternately depending on the training phase:

$$\mathcal{L}_{open} = \begin{cases} \sum_{i \in I} \mathcal{L}_{ce}(\tilde{\mathbf{y}}, \mathbf{p}^i_{copy}) \\ \sum_{z \sim \mathcal{N}} \mathcal{L}_{ce}(\frac{1}{K} \cdot \mathbf{u}, \mathbf{p}_{gan}) \end{cases} \tag{11}$$

To define $\tilde{\mathbf{y}}$ for each difficulty group, we grant a smaller value to α when the input has a high probability of belonging to one of the known categories. To be specific, we divide α into α_{hard} and α_{easy}. Note that motivated by ARPL [4], we set the target label for the fake image of the GAN to a uniform distribution since it works better to regard them as unknown. Likewise, we also set α_{easy} to 1 to further regard easy fakes as unknown. On the other hand, since identifying hard fake instances as open set might contradict the training procedure, α_{hard} is set to 0.5. Therefore, the easy and moderate fake examples are forced to reside outside the class boundaries, while a difficult set of fakes are implemented to be calibrated within the class boundaries.

Algorithm 1 Training DIAS

```
Require: Parameters of Classifier, Copycat,
    Generator and Discriminator $(\theta, \phi, \psi, \omega)$,
    Learning rate $\eta$
    for $i \in \{1, ..., epoch\}$ do
        ▷ Phase I: Training with Copycat
        $\mathcal{L}_{copy} = \mathcal{L}_{imi} + \mathcal{L}_{reg}$
        $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{copy}$
        $\mathcal{L}_{cls} = \mathcal{L}_{close} + \lambda \cdot \mathcal{L}_{open}$
        $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{cls}$
        ▷ Phase II: Training with GAN
        $\psi \leftarrow \psi - \eta \nabla_\psi \mathcal{L}_{gan}$
        $\omega \leftarrow \omega - \eta \nabla_\omega \mathcal{L}_{gan}$
        $\mathcal{L}_{cls} = \mathcal{L}_{close} + \lambda \cdot \mathcal{L}_{open}$
        $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{cls}$
    end for
```

In Fig. 4, we describe how diverse levels of difficulty assist the classifier to build robust decision boundaries. As can be seen, while easy and moderate fake sets play a role of gathering open set to the parts where they can retain uniform distribution, hard fakes force the classifier to calibrate its decision boundaries with the smoothed label. Our training procedure is outlined in Algorithm 1.

3.4 Inherent threshold of DIAS

DIAS has its benefits at the inference for threshold selection. In general, cross-validation is a widely adopted strategy to specify threshold when identifying the open set, although it is time-consuming work to find an appropriate threshold. On top of this, the value of threshold is very sensitive that it must be explored for every pair of known and unknown dataset [28]. In contrast, our proposed DIAS is inherently equipped with a criterion to detect unknowns. To be precise, we observe that $\max_k \tilde{\mathbf{y}}$ computed with α_{hard} works out to be the decent threshold. This is because we enforce the confidence scores for virtual open set to be equal or lower than the target confidence for difficult fake samples $\max_k \tilde{\mathbf{y}}$. Formally, given an image $\mathbf{x}$, we extend the closed set classifier by predicting label $\hat{y}$ as:

$$\hat{y} = \begin{cases} K+1 & \text{if } \max\big\{W(\mathbf{x})\big\} < \tau + \epsilon \\ \underset{k=1,\cdots,K}{\arg\max}\ \big\{W(\mathbf{x})\big\} & \text{otherwise} \end{cases} \tag{12}$$

where $K+1$ is a class for unknown and $\tau = (1-\alpha_{hard}) + \alpha_{hard}/K$. Note that τ is same as simply calculating the maximum value in Eq. 8 with α_{hard}. Therefore, we do not need any extra algorithm to compute the threshold.

4 Experiments

4.1 Evaluation protocols, metrics, and datasets

Evaluation protocols. With a c-class dataset, OSR scenario is generally designed by randomly selecting K classes as known where $(c \gg K)$. Then, the remaining $c-K$ classes are considered as open set classes. Then, five-randomized scenarios are simulated to measure the Area Under the Receiver Operating Characteristic (AUROC) curve or F-score. Note that the split information can be different over methods since they conduct experiments on randomized trials [5,28,31,43]. However, as different split often leads to unfair comparison, recent methods pre-define split information on each AUROC [13,28,42] and F1-score benchmark [43]. Following them, we conduct experiments on two standardized split information for a fair comparison. Split information is in the supplementary.

Metrics. We use two metrics: F1-score and AUROC score. F1 is a more practical measure, representing the classification accuracy. It is calculated as a weighted average of precision and recall. For OSR, F1-score is commonly obtained by adding an extra class for open set and searching for proper threshold. On the other hand, AUROC is a metric that does not require any calibration process. It

Table 1. AUROC score for detecting known and unknown samples. † indicates the reproduced result to unify the split information. The best results are indicated in bolds.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50	Tiny-IN
Softmax	0.978	0.886	0.677	0.816	0.805	0.577
OpenMax	0.981	0.894	0.695	0.817	0.796	0.576
G-OpenMax [8]	0.984	0.896	0.675	0.827	0.819	0.580
OSRCI [28]	$0.988_{\pm 0.004}$	$0.91_{\pm 0.01}$	$0.699_{\pm 0.038}$	0.838	0.827	0.586
CROSR [42]	$0.991_{\pm 0.004}$	$0.899_{\pm 0.018}$	-	-	-	0.589
C2AE [31]	-	$0.892_{\pm 0.013}$	$0.711_{\pm 0.008}$	$0.810_{\pm 0.005}$	$0.803_{\pm 0.000}$	$0.581_{\pm 0.019}$
GFROSR [32]	-	$0.955_{\pm 0.018}$	$0.831_{\pm 0.039}$	-	-	$0.657_{\pm 0.012}$
CGDL [38]	$0.977_{\pm 0.008}$	$0.896_{\pm 0.023}$	$0.681_{\pm 0.029}$	$0.794_{\pm 0.013}$	$0.794_{\pm 0.003}$	$0.653_{\pm 0.002}$
RPL [5]	$0.917_{\pm 0.006}$	$0.931_{\pm 0.014}$	$0.784_{\pm 0.025}$	$0.885_{\pm 0.019}$	$0.881_{\pm 0.014}$	$0.711_{\pm 0.026}$
PROSER [47]†	$0.964_{\pm 0.019}$	$0.930_{\pm 0.005}$	$0.801_{\pm 0.031}$	$0.898_{\pm 0.015}$	$0.881_{\pm 0.003}$	$0.684_{\pm 0.029}$
ARPL+cs [4]†	$0.991_{\pm 0.004}$	$0.946_{\pm 0.005}$	$0.819_{\pm 0.029}$	$0.904_{\pm 0.002}$	$0.901_{\pm 0.002}$	$0.710_{\pm 0.002}$
CVAECap [13]	$\mathbf{0.992}_{\pm 0.004}$	$\mathbf{0.956}_{\pm 0.012}$	$0.835_{\pm 0.023}$	$0.888_{\pm 0.019}$	$0.889_{\pm 0.017}$	$0.715_{\pm 0.018}$
DIAS **(Ours)**	$\mathbf{0.992}_{\pm 0.004}$	$0.943_{\pm 0.008}$	$\mathbf{0.850}_{\pm 0.022}$	$\mathbf{0.920}_{\pm 0.011}$	$\mathbf{0.916}_{\pm 0.007}$	$\mathbf{0.731}_{\pm 0.015}$

Table 2. Comparison of accuracy for closed set classes between baseline and DIAS.

Method	MNIST	SVHN	CIFAR10	CIFAR+	Tiny-IN
Softmax	**0.997**	0.966	0.934	0.960	0.653
DIAS **(Ours)**	**0.997**	**0.970**	**0.947**	**0.964**	**0.700**

considers the trade-off between true positive rate and false positive rate across different decision thresholds.

Datasets. Prior to dataset explanation, we describe the term, openness, which indicates the ratio between the known and the unknown:

$$Openness = 1 - \sqrt{K/(K + \hat{K})} \tag{13}$$

where K and $\hat{K}$ stand for the number of classes for known and unknown, respectively. With openness, we discuss several benchmarking datasets: **MNIST**, **SVHN**, **CIFAR10** [23,26,29] contain 10 classes. Six classes are chosen to be known and four classes are to be unknown classes. Openness is 22.54%. **CIFAR+10**, **CIFAR+50** are artificially synthesized with four non-animal classes from CIFAR10 and N non-overlapping classes from CIFAR100 [38,43]. As more classes are considered as open set, openness is higher. Openness for each are 46.54% and 72.78%. **Tiny-ImageNet** [7] Tiny-IN is a subset of ImageNet which has 200 classes. We follow the common protocol [13,28] to resize it to 32 × 32. Afterwards, 20 classes are sampled to be used as closed set and the remaining classes as open set classes. Openness is 68.38%.

4.2 Experimental Results

OSR performances are in Tab. 1 and Tab. 3. Most of the baseline results in Tab. 1 are taken from the [13] where they reproduced papers' performances with the same configuration for a fair comparison. The scores of DIAS, PROSER [47], and ARPL+cs [4] are evaluated based on the same protocol with [13] on the split

Table 3. Average of macro-averaged F1-scores in five splits. We adopt the protocol from GCM-CF [43]. † indicates the reproduced performance with official code.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50
Softmax	0.767	0.762	0.704	0.778	0.660
Openmax	0.859	0.780	0.714	0.787	0.677
CGDL [38]	0.890	0.763	0.710	0.779	0.710
GCM-CF [43]	0.914	0.793	0.726	0.794	0.746
ARPL+cs [4] †	0.951$_{\pm 0.009}$	0.857$_{\pm 0.008}$	0.753$_{\pm 0.033}$	0.827$_{\pm 0.010}$	0.753$_{\pm 0.001}$
DIAS (**Ours**)	**0.953**$_{\pm 0.015}$	**0.880**$_{\pm 0.010}$	**0.809**$_{\pm 0.026}$	**0.859**$_{\pm 0.010}$	**0.829**$_{\pm 0.006}$

Table 4. Macro-averaged F1-scores on the MNIST with three other datasets as unknown.

Method	Omniglot	MNIST-noise	Noise
Softmax	0.595	0.801	0.829
Openmax [3]	0.780	0.816	0.826
CROSR [42]	0.793	0.827	0.826
CGDL [38]	0.850	0.887	0.859
PROSER [47]	0.862	0.874	0.882
CVAECapOSR [13]	0.971	**0.982**	0.982
DIAS (**Ours**)	**0.989**	**0.982**	**0.989**

publicized by [13,28]. We use their hyperparameters except the model architecture in PROSER to unify the backbone. Note that some papers cannot be compared under the same codebase due to non-reproducible results (split information nor the codes are not publicized) [46] and the contrasting assumption in the existence of open set data [22]. As shown, our method achieves significant improvements over the state-of-the-art techniques in CIFAR10, CIFAR+10, CIFAR+50, and tiny-ImageNet datasets, and shows comparable results in digit datasets where the performances are almost saturated. Also, Tab. 2 shows that DIAS improves the accuracy of the closed set along with its capability of detecting unknowns.

We discussed that DIAS establishes a standard to determine a proper threshold for unknowns in Sec. 3.4. In Tab. 3, we validate such claim that DIAS does not require expensive and complex tuning for the threshold search so thus it is much more practical than previous works. For a fair comparison, baseline results are from [43], and ARPL+cs and ours are tested on the same split. The results show that DIAS consistently outperforms the baselines with noticeable margins. For the threshold, we find ϵ in Eq. 12 works well when set to -0.05 for all experiments.

As previously did in [13,38,47], we conduct an additional open set detection experiment. Briefly, we train the classifier on MNIST and evaluate on Omniglot [25], MNIST-Noise, and Noise. Omniglot is an alphabet dataset of 1623 handwritten characters from 50 writing systems, and Noise is a synthesized dataset where each pixel is sampled from a gaussian distribution. MNIST-Noise is noise-embedded MNIST dataset. We sample 10000 examples from each dataset since MNIST contains 10000 instances. The macro F-score between ten digit classes and open set classes are measured to compare performances. The experimental results are reported in Tab. 4. Although the performances on all three datasets are almost saturated, DIAS provides competitive results with state-of-the-art methods.

Table 5. Ablation study on varying difficulties of fake examples. We report the AUROC scores on CIFAR100 dataset against varying openness.

Copycat Hard	GAN Moderate	Copycat Easy	Openness (%)			
			22.54	29.29	55.28	68.38
-	-	-	72.28	71.72	78.85	77.34
-	-	✓	72.45	72.51	78.67	78.36
-	✓	-	73.69	72.46	79.20	81.13
✓	-	-	76.59	74.82	81.05	80.91
✓	-	✓	76.79	74.92	81.06	81.56
✓	✓	✓	**76.92**	**75.55**	**81.59**	**83.95**

Table 6. Ablation study on each generator in DIAS with F1-score.

Copycat	GAN	MNIST	SVHN	C10	C+10	C+50
-	-	0.767	0.762	0.704	0.778	0.660
-	✓	0.926	0.840	0.777	0.850	0.775
✓	-	0.948	0.860	0.788	0.846	0.816
✓	✓	**0.953**	**0.880**	**0.809**	**0.859**	**0.829**

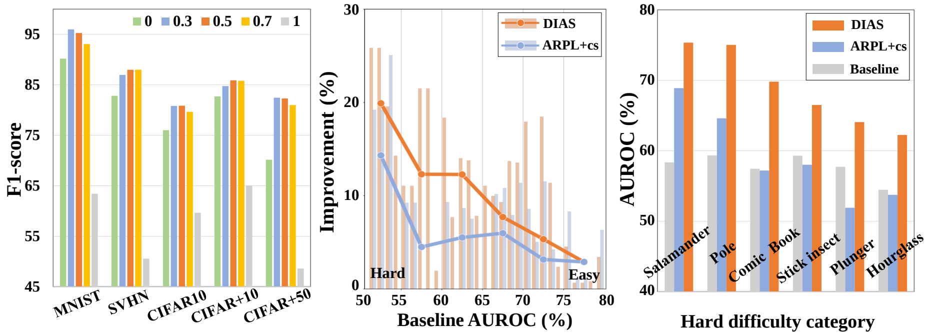
4.3 Ablation study and Further analysis

Effect of varying difficulty levels. In Tab. 5, we summarize the ablation analysis on varying difficulties of fake samples. As reported, our hard-difficulty fakes have highest contribution to improve the OSR performance. It validates that the detailed calibration of decision boundaries by facing with hard-difficulty fake examples is significantly helpful to enhance the model robustness toward unseen samples. For the relatively small improvement brought by the easy-difficulty examples, we think that the supervised models are already quite robust against such easy cases. More importantly, by utilizing all the difficulty-level samples for the simulation, DIAS boosts the AUROC at various openness configurations. On top of this, we also evaluate each generator in Tab. 6 with F1-score to show their relative importance. Note that, we search for the best threshold by Eq. 12 to distinguish open set with F1-score for approaches without the Copycat, while it occurs by itself for DIAS when processing the hard fakes of Copycat. Results demonstrate that both our generators are suitable for simulating open set instances and also validate the unique advantages of Copycat; significantly improving the performance with its inherent threshold for identifying unknowns.

Effect of smoothing ratio. Since α_{hard} is an important parameter, we study its impact in Fig. 5 (a). Intuitively, the target label for the fake samples become uniform distribution when α_{hard} is 1, while it becomes one-hot label with α_{hard} of 0. The tendency observed from the gray bar with $\alpha_{hard} = 1$, forcing the hard-difficulty samples to have uniformly distributed class probability drastically degrades the performance. On the other hand, the performance is also dropped if we treat the fake samples as known classes with $\alpha_{hard} = 0$, since in such case the hard-difficulty samples are no longer utilized to calibrate within the class decision boundaries. Excepting those extreme cases, we observe that DIAS has low sensitivity to the choice of the hyperparameter α_{hard}. Note that, all the results reported in Tab. 3 is produced with α_{hard} of 0.5.

Table 7. Validity of the inherent threshold in DIAS. For DIAS (*), we searched the best threshold for identifying unknowns in DIAS.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50
DIAS	$0.953_{\pm 0.015}$	$0.880_{\pm 0.010}$	$0.809_{\pm 0.026}$	$0.859_{\pm 0.010}$	$0.829_{\pm 0.006}$
DIAS (*)	$0.970_{\pm 0.004}$	$0.883_{\pm 0.009}$	$0.809_{\pm 0.026}$	$0.861_{\pm 0.009}$	$0.833_{\pm 0.005}$



(a) Effect of α_{hard}

(b) Performance improvements w.r.t. class-wise difficulty

Fig. 5. (a) Effect of α_{hard}. Results demonstrate that hard fake samples significantly contribute to calibrate the decision boundary. Regarding these as either known (green) or unknown (gray) decreases the performances. **(b)** Performance improvements over the baseline Softmax classifier w.r.t. the class-wise difficulty. **(b-Left)** The difficulty of open classes are determined by the AUROC scores of the Softmax classifier (x-axis). Thus, the lower AUROC scores, the harder the classes are, and placed on the left side. The bars in the graph show the class-wise improvements from the baseline Softmax classifier in order of the difficulty level, while the curves represent the average improvements over classes within 5% intervals on the AUROC scores of the Softmax classifier. As shown, DIAS is more effective in distinguishing harder open classes. **(b-Right)** Comparison of class-specific AUROC scores on harder open classes. Such results validate the effectiveness of DIAS for identifying open set classes which our baseline classifiers find difficult.

Validity of the inherent threshold is confirmed in Tab. 7. Although the optimal threshold may differ between datasets, only small gaps between the best and inherent thresholds verify that smoothed probability for hard fake examples is suitable to be an adequate threshold because there is no searching cost.

Further analysis We conducted an in-depth analysis on Tiny-ImageNet to explore why the proposed method is effective in detecting unknowns and how effective it is on each class and each difficulty group. For this study, we assume that difficulty levels are only varying across classes. In other words, we do not consider the instance-wise difficulty in this experiment. Specifically, we utilize class-wise AUROC score of the Softmax classifier to determine the difficulty. The class-wise improvements of the OSR is reported in Fig. 5 (b) with comparison against the most recent simulation method [4]. Those results validate the merits of proposed DIAS especially in distinguishing confusing known and unknown instances, while it provides improvements across all levels of difficulties.

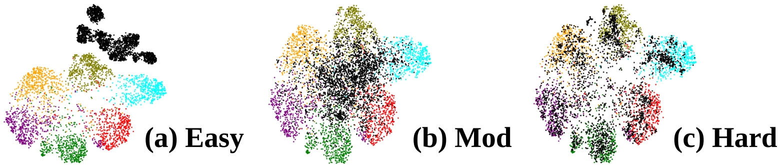


Fig. 6. t-SNE of fake distributions (Black). As the difficulty level gets higher, black dots are harder to be separated.

To understand how DIAS enables better separation in all difficulty levels, we examine our generators. Specifically, we visualized the feature space of the Softmax classifier with fake samples from our generators on CIFAR10. In Fig. 6, six colored clusters other than black correspond to each closed set class. We find that our generators are actually generating diverse fakes as we intentionally designed. Easy fakes are embedded out of class clusters (Fig. 6 (a)), moderate ones are partially mixed with known samples (Fig. 6 (b)), and finally, hard-difficulty examples significantly overlie on top of the knowns (Fig. 6 (c)). Hence, as our virtual open set covers broader range of difficulty levels in open set, DIAS produced better results across all difficulty levels.

In addition, one may ask how the Copycat is able to generate more confusing fakes than GAN. This is because the generator in GAN learn to generate fakes that share features with knowns in the viewpoint of the discriminator, while the difficulty levels depend on the viewpoint of the classifier. As the Copycat learns the classifier's perspective iteratively, the Copycat is equipped with the strength to generate confusing fake instances from the classifier's viewpoint. AUROC and W.D from Fig. 1 and visualized feature maps in Fig. 6 further support the choice of setting the Copycat as the most difficult-fake instance generator.

5 Conclusion

OSR assumes numerous objects are present that do not belong to learned classes. When classifiers are facing with these, they misclassify them into one of the known categories, often with high confidence. To prepare the classifier for handling unknowns, there have been works to simulate virtual open instances. However, these works only considered unknowns as one set. We claim that considering various levels of difficulty in OSR is an untapped question to be studied. To this end, we proposed the Difficulty-Aware Simulator to simulate open set with fakes at various difficulty levels. Also, we introduced the Copycat and the GAN-based generator in the classifier's perspective for preparing adequate fake samples for classifier tuning. Extensive experiments demonstrate that our proposed DIAS significantly improves the robustness of the classifier toward open set instances.

Acknowledgements. This work was supported in part by MSIT/IITP (No. 2022-0-00680, 2020-0-00973, 2020-0-01821, and 2019-0-00421), MCST/KOCCA (No. R2020070002), and MSIT&KNPA/KIPoT (Police Lab 2.0, No. 210121M06).

References

1. Arora, S., Ge, R., Liang, Y., Ma, T., Zhang, Y.: Generalization and equilibrium in generative adversarial nets (gans). In: International Conference on Machine Learning. PMLR (2017)
2. Bendale, A., Boult, T.: Towards open world recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2015)
3. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
4. Chen, G., Peng, P., Wang, X., Tian, Y.: Adversarial reciprocal points learning for open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2021). https://doi.org/10.1109/TPAMI.2021.3106743
5. Chen, G., Qiao, L., Shi, Y., Peng, P., Li, J., Huang, T., Pu, S., Tian, Y.: Learning open set network with discriminative reciprocal points. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. Springer (2020)
6. Chen, P., Liu, S., Zhao, H., Jia, J.: Distilling knowledge via knowledge review. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee (2009)
8. Ge, Z., Demyanov, S., Chen, Z., Garnavi, R.: Generative openmax for multi-class open set classification. In: Proceedings of the British Machine Vision Conference (BMVC) (2017)
9. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence (2020)
10. Girish, S., Suri, S., Rambhatla, S.S., Shrivastava, A.: Towards discovery and attribution of open-world gan generated images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14094–14103 (2021)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems (2014)
12. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning. PMLR (2017)
13. Guo, Y., Camporese, G., Yang, W., Sperduti, A., Ballan, L.: Conditional variational capsule network for open set recognition. Proceedings of the IEEE international conference on computer vision (ICCV) (2021)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
16. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. International Conference on Learning Representations (ICLR) (2017)
17. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Workshop on Deep Learning and Representation Learning Workshop (2015)

18. Hsu, Y.C., Shen, Y., Jin, H., Kira, Z.: Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
19. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2017)
20. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: European Conference on Computer Vision. Springer (2014)
21. Júnior, P.R.M., De Souza, R.M., Werneck, R.d.O., Stein, B.V., Pazinato, D.V., de Almeida, W.R., Penatti, O.A., Torres, R.d.S., Rocha, A.: Nearest neighbors distance ratio open-set classifier. Machine Learning (2017)
22. Kong, S., Ramanan, D.: Opengan: Open-set recognition via open data generation. In: ICCV (2021)
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems (2012)
25. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science (2015)
26. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE (1998)
27. Lee, K., Lee, H., Lee, K., Shin, J.: Training confidence-calibrated classifiers for detecting out-of-distribution samples. International Conference on Learning Representations (ICLR) (2018)
28. Neal, L., Olson, M., Fern, X., Wong, W.K., Li, F.: Open set learning with counterfactual images. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
29. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
30. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2015)
31. Oza, P., Patel, V.M.: C2ae: Class conditioned auto-encoder for open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
32. Perera, P., Morariu, V.I., Jain, R., Manjunatha, V., Wigington, C., Ordonez, V., Patel, V.M.: Generative-discriminative feature representations for open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
33. Roady, R., Hayes, T.L., Kemker, R., Gonzales, A., Kanan, C.: Are open set classification methods effective on large-scale datasets? Plos one (2020)
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision (2015)
35. Scheirer, W.J., Jain, L.P., Boult, T.E.: Probability models for open set recognition. IEEE transactions on pattern analysis and machine intelligence (2014)
36. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE transactions on pattern analysis and machine intelligence (2012)

37. Shu, L., Xu, H., Liu, B.: Doc: Deep open classification of text documents. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, (EMNLP) (2017)
38. Sun, X., Yang, Z., Zhang, C., Ling, K.V., Peng, G.: Conditional gaussian distribution learning for open set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
39. Tudor Ionescu, R., Alexe, B., Leordeanu, M., Popescu, M., Papadopoulos, D.P., Ferrari, V.: How hard can it be? estimating the difficulty of visual search in an image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
40. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: International Conference on Machine Learning. PMLR (2019)
41. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
42. Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., Naemura, T.: Classification-reconstruction learning for open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
43. Yue, Z., Wang, T., Sun, Q., Hua, X.S., Zhang, H.: Counterfactual zero-shot and open-set visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
44. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. International Conference on Learning Representations (ICLR) (2017)
45. Zhang, H., Patel, V.M.: Sparse representation-based open set recognition. IEEE transactions on pattern analysis and machine intelligence (2016)
46. Zhang, H., Li, A., Guo, J., Guo, Y.: Hybrid models for open set recognition. In: European Conference on Computer Vision. Springer (2020)
47. Zhou, D.W., Ye, H.J., Zhan, D.C.: Learning placeholders for open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)