Contrastive Open Set Recognition

Baile Xu[1,2], **Furao Shen**[1,3*], **Jian Zhao**[4]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Department of Computer Science and Technology, Nanjing University
[3]School of Artificial Intelligence, Nanjing University
[4]School of Electronic Science and Engineering, Nanjing University
blxu@smail.nju.edu.cn, frshen@nju.edu.cn, jianzhao@nju.edu.cn

Abstract

In conventional recognition tasks, models are only trained to recognize learned targets, but it is usually difficult to collect training examples of all potential categories. In the testing phase, when models receive test samples from unknown classes, they mistakenly classify the samples into known classes. Open set recognition (OSR) is a more realistic recognition task, which requires the classifier to detect unknown test samples while keeping a high classification accuracy of known classes. In this paper, we study how to improve the OSR performance of deep neural networks from the perspective of representation learning. We employ supervised contrastive learning to improve the quality of feature representations, propose a new supervised contrastive learning method that enables the model to learn from soft training targets, and design an OSR framework on its basis. With the proposed method, we are able to make use of label smoothing and mixup when training deep neural networks contrastively, so as to improve both the robustness of outlier detection in OSR tasks and the accuracy in conventional classification tasks. We validate our method on multiple benchmark datasets and testing scenarios, achieving experimental results that verify the effectiveness of the proposed method.

Introduction

Traditional recognition algorithms work under a closed set assumption that the training data and test data share the same labels and feature space. However, it is usually difficult to collect training examples covering all potential classes of test samples in reality, and a traditional classifier would classify any test sample into one of the training classes, even if its true category has not been learned. A realistic recognition scenario for this challenge is *open set recognition* (OSR), where samples from unknown classes may appear during testing, and the recognition algorithm is required to detect unknown test samples while keeping a high classification accuracy of known classes (Scheirer et al. 2012).

In a traditional multi-class classification network, the output layer usually uses the softmax function to produce a probability distribution over the training classes. The softmax function does not estimate the probability of unknown classes due to its closed nature, so it is not suitable for OSR. A direct solution is thresholding the softmax scores (Hendrycks and Gimpel 2016) to reject estimations with low confidence, which provides a simple baseline for OSR research. However, the over-confidence phenomena have been witnessed when test samples from unknown categories are input to deep neural networks.

Although great progress has been made in the OSR research area, a recent study (Vaze et al. 2021) suggests that simply using state-of-the-art training mechanisms on closed-set classifiers could significantly boost their performances in OSR tasks. This discovery shows the potential of using better representation learning techniques to improve the OSR capability of the classifier. Inspired by this discovery, we intend to develop a more effective representation learning mechanism specifically for OSR tasks.

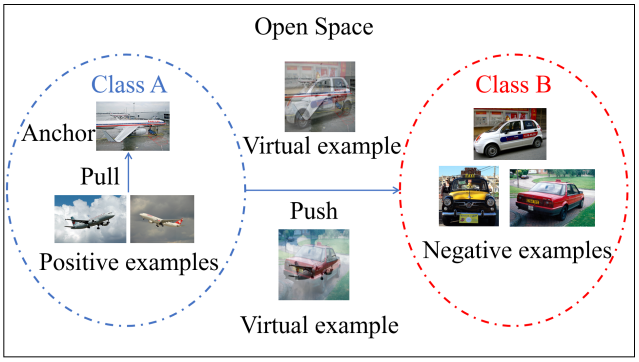


Figure 1: Overview of the proposed method. Supervised contrastive learning pulls the positive examples from the same class towards the anchor example, while pushing the negative examples away. Virtual examples generated by the mixup algorithm simulate unknown samples in the open space.

In our research, we use supervised contrast learning to learn high-quality representations by comparing the positive and negative pairs of training examples. We observed that the contrastively learned representations work better for detecting unknown targets. We also use the mixup algorithm to generate semantically vague virtual examples, so that the model could contrast real training examples from

*Corresponding author

known classes with unknown virtual examples in the training phase. In order to bring virtual examples with soft labels into the contrastive learning framework, we further design an enhanced supervised contrastive learning method that allows similarity-based relationships between pairs of training examples. This modification improves the performances in both OSR and closed-set classification tasks.

The contributions of this paper are summarized as follows:

1. We propose a contrastive learning based open set recognition method named ConOSR. We experimentally analyse the reason why contrastively learned features could boost the performance of a classifier in OSR tasks.

2. We enhance the Supervised Contrastive Learning algorithm with the ability to learn from soft targets. The enhanced method SupCon-ST outperforms the vanilla SupCon in closed-set classification, and also improves the performance of ConOSR to outperform state-of-the-art OSR methods.

Related Work

Open Set Recognition

Open Set Recognition was first defined in (Scheirer et al. 2012), together with important related definitions like open space and open space risk. A recent survey (Geng, Huang, and Chen 2020) categorizes OSR methods into discriminative methods and generative methods. The majority of recent discriminative methods are DNN-based methods, which enable deep networks with the ability of unknown detection by enhancing the output layer with various outlier detection mechanisms. Openmax (Bendale and Boult 2016) estimates the probability of a test sample belonging to an unknown class by measuring the distance between its activation vector and the mean activation vectors of known classes. Reciprocal Points Learning (Chen et al. 2020a) introduces a novel concept named reciprocal point, so as to model the latent open space for each known class in the feature space. PROSER (Zhou, Ye, and Zhan 2021) assigns placeholders to unknown classes in the feature space, trying to imitate open-set classes and predict the distribution of unknown data. PROSER also uses feature mixup to generate virtual examples as placeholders. CVAECapOSR (Guo et al. 2021) uses the capsule network as the feature encoding model, in order to learn compact feature representations for known classes.

Generative methods can be further categorized into instance generation methods and non-instance generation methods. The first group usually generates pseudo-examples using GANs (Goodfellow et al. 2014) to mimic unknown test samples in the open space. G-Openmax (Ge et al. 2017) extends Openmax by training DNNs with unknown samples generated by a conditional GAN. OSRCI (Neal et al. 2018) trains an encoder-decoder GAN to generate counterfactual instances close to training examples but not belong to any classes, and enhances the training data with counterfactual instances. Recently, OpenGAN (Kong and Ramanan 2021) proposes to use GAN-discriminator as open-set likelihood function and real-world data as outliers to improve the training of GANs, and it significantly outperforms prior OSR methods in image classification and pixel segmentation tasks. Non-instance generation methods train encoder-decoder networks to assist unknown sample detection. CROSR (Yoshihashi et al. 2019) utilizes both the prediction of the classification layer and the latent representation for reconstruction in the unknown detection step. GFROSR (Perera et al. 2020) uses the reconstruction model as data augmentation, forcing the network to learn features that capture object structure. Generative methods provide more background information for the recognition system by modeling data distribution, but training generative models significantly increase the total training cost of the recognition system.

Contrastive Learning

Contrastive learning is an area of representation learning that has attracted much research attention in recent years. Most contrastive learning methods are self-supervised (Van den Oord, Li, and Vinyals 2018; He et al. 2020; Chen et al. 2020b; Chen and He 2021), which do not rely on task-specific supervision. A major problem in self-supervised contrastive learning is how to get positive and negative pairs without supervision. MOCO (He et al. 2020) and SimCLR (Chen et al. 2020b) use multiple views of a single training example as positive pairs and different training examples as negative pairs, but they require a large number of negative pairs to achieve good performances. BYOL (Grill et al. 2020) and SimSiam (Chen and He 2021) use siamese network structure and stop-gradient to avoid using negative pairs, so they could work with smaller batches of training data. In the ImageNet classification task, recent self-supervised contrastive learning methods have achieved comparable results with supervised learning.

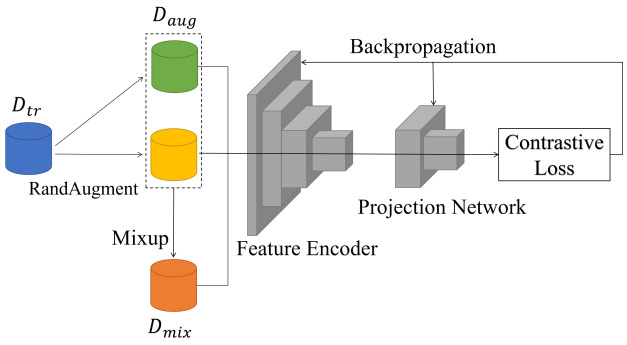
Supervised contrastive learning (SupCon) (Khosla et al. 2020) sets its basis on learning representations that maximize the similarities between positive pairs from the same class and the differences between negative pairs from different classes. SupCon outperforms self-supervised methods in terms of classification accuracy by a large margin. SupCon also outperforms plain CNN networks in multiple closed-set classification tasks. However, SupCon does not attract as much research attention as self-supervised methods because it can not work with unlabelled data.

Contrastive Open Set Recognition

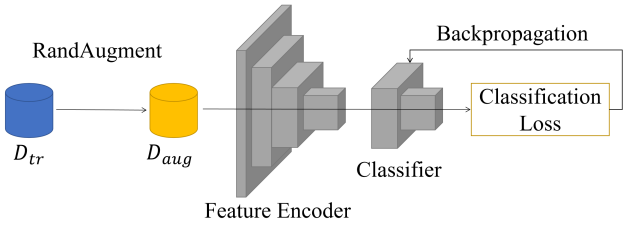
In this section, we describe the proposed Contrastive Open Set Recognition (ConOSR) in detail. An overview of the proposed ConOSR training pipeline is shown in Figure 2. Our method consists of a contrastive learning step and a classifier training step.

In the contrastive learning step, the data preprocessing module generates augmented views of the training data $\mathcal{D}_{tr}$ using RandAugment, and then mixes them up to get a batch of virtual examples. After that, the encoder network and the projection network are optimized to minimize the contrastive loss computed on both the augmented data $\mathcal{D}_{aug}$ and the mixed data $\mathcal{D}_{mix}$.

In the classifier training phase, $\mathcal{D}_{tr}$ is preprocessed by the RandAugment algorithm, and then forward propagated



(a) Step 1: Contrastive representation learning.



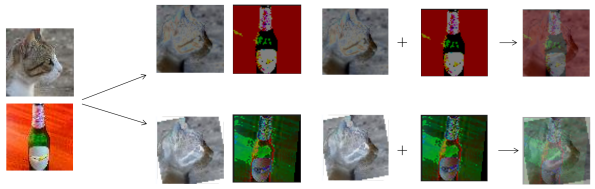
(b) Step 2: Training the classifier.

Figure 2: Overview of the ConOSR training pipeline.

through the fixed encoder network to obtain the feature representations. Then the classification network is optimized to minimize the cross entropy loss. After convergence, the thresholds for rejecting unknown test samples are estimated using $\mathcal{D}_{tr}$. Details of the components in the framework will be described introduced in the following subsections.

Data Augmentation and Contrastive Learning

As shown in Figure 2, we adopt two different data augmentation techniques. RandAugment (Cubuk et al. 2020) and Mixup (Zhang et al. 2017) are state-of-the-art data augmentation methods widely used in various fields. An illustration of the augmentation methods is shown in Figure 3.



(a) RandAugment (b) Mixup

Figure 3: The data augmentation techniques used in the proposed framework. (a) RandAugment conducts random visual transformations on the input image, while keeping its semantic content; (b) Mixup generates a virtual example by linearly mixing the contents and the labels of two examples.

RandAugment Given a training image, RandAugment randomly selects N transformations from 14 available transformations, and then applies the selected transformations on the image sequentially. The magnitude of the transformation is controlled by a global hyper-parameter M. RandAugment enriches the visual information of training examples, while keeping their semantic contents unchanged, so that the model can learn transform invariant feature representations.

For each training example $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ in a batch $(\boldsymbol{x}_i, y_i)_{i=1}^n$, we use RandAugment to generate two augmented views $\widetilde{\boldsymbol{x}}_{2k}$ and $\widetilde{\boldsymbol{x}}_{2k+1}$. The randomly selected augmentation functions ensure the visual difference between $\widetilde{\boldsymbol{x}}_{2k}$ and $\widetilde{\boldsymbol{x}}_{2k+1}$ during multiple epochs of training.

While augmenting the images with RandAugment, we also enhance the labels of training examples with label smoothing. If the total number of classes is m and the training example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ belongs to the k-th class, then the smoothed label $\widetilde{\boldsymbol{y}}_i = (\widetilde{y}_{i1}, \widetilde{y}_{i2}, \ldots, \widetilde{y}_{im})$ is formulated as:

$$\widetilde{y}_{ij} = \begin{cases} \sigma & j = k \\ \dfrac{1-\sigma}{m-1} & otherwise \end{cases} \tag{1}$$

.

Mixup Mixup constructs virtual examples by linearly mixing pairs of training examples. Given two training examples $(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i)$ and $(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)$ randomly sampled from $\mathcal{D}_{aug}$, a virtual example $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is constructed as:

$$\hat{\boldsymbol{x}} = \gamma \widetilde{\boldsymbol{x}}_i + (1-\gamma)\widetilde{\boldsymbol{x}}_j, \tag{2}$$

$$\hat{\boldsymbol{y}} = \gamma \widetilde{\boldsymbol{y}}_i + (1-\gamma)\widetilde{\boldsymbol{y}}_j, \tag{3}$$

where $\gamma \in [0, 1]$ is randomly selected from the uniform distribution.

Mixup is important in the ConOSR framework because it generates virtual examples with ambiguous semantics, so that the virtual examples could simulate unknown examples in the training phase. In order to contrast real examples with virtual examples, we use $\mathcal{D}_{aug}$ and $\mathcal{D}_{mix}$ at the same time in the contrastive learning step.

Supervised Contrastive Learning with Soft Targets

The network structures in contrastive learning consists of a feature encoder $\boldsymbol{\phi}(\cdot)$ and a projection network $\boldsymbol{\psi}(\cdot)$. The encoder network maps a training example $\boldsymbol{x}_i$ to a representation vector $\boldsymbol{h}_i \in \mathbb{R}^{de}$. Then the projection network further maps $\boldsymbol{h}_i$ to a projection vector $\boldsymbol{z}_i \in \mathbb{R}^{dp}$, which is used for calculating the contrastive loss. The target of contrastive learning is maximizing the difference of similarity between positive pairs and negative pairs in the projection space. In the vanilla SupCon algorithm, the contrastive loss is defined as:

$$\mathcal{L}^{sup} = \sum_i -\frac{1}{|P_i|} \sum_{j \in P_i} \log \frac{\exp(\boldsymbol{z_i} \cdot \boldsymbol{z_j}/\tau)}{\sum_{k \neq i} \exp(\boldsymbol{z_i} \cdot \boldsymbol{z_k}/\tau)}, \tag{4}$$

where P_i is is the set of positive examples belonging to the same class as i, and τ is the temperature hyper-parameter.

SupCon distinguishes positive examples and negative examples according to their labels. However, the hard partition of positive and negative examples conflicts with the soft labels in our augmented training set. Therefore, we propose

an enhanced version of SupCon, which could take training examples with soft labels as inputs.

Our contrastive learning framework allows a similarity-based relationship between samples, rather than dividing them into positive and negative pairs. Given a pair of labeled samples $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and $(\boldsymbol{x}_i, \boldsymbol{y}_j)$, a pairwise similarity metric $s(\boldsymbol{y}_i, \boldsymbol{y}_j)$ is defined using the label vectors. We also want $s(\boldsymbol{y}_i, \boldsymbol{y}_j)$ to incorporate into equation (4) without changing its result. So when $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are limited to one-hot vectors, $s(\boldsymbol{y}_i, \boldsymbol{y}_j) = 1$ if $\boldsymbol{y}_i = \boldsymbol{y}_j$, otherwise $s(\boldsymbol{y}_i, \boldsymbol{y}_j) = 0$. Considering this condition, we define $s(\boldsymbol{y}_i, \boldsymbol{y}_j)$ as the cosine similarity by default:

$$s(\boldsymbol{y}_i, \boldsymbol{y}_j) = \frac{\boldsymbol{y}_i \cdot \boldsymbol{y}_j}{||\boldsymbol{y}_i|| ||\boldsymbol{y}_j||}, \tag{5}$$

then we define the SupCon-ST loss function as:

$$\mathcal{L}^{scst} = -\sum_i \sum_{j \neq i} \frac{s(\boldsymbol{y}_i, \boldsymbol{y}_j)}{\sum\limits_{k \neq i} s(\boldsymbol{y}_i, \boldsymbol{y}_k)} \log \frac{\exp(\boldsymbol{z_i} \cdot \boldsymbol{z_j}/\tau)}{\sum\limits_{k \neq i} \exp(\boldsymbol{z_i} \cdot \boldsymbol{z_k}/\tau)}. \tag{6}$$

The form of equation (6) is similar to the cross entropy loss. When all the labels vectors are one-hot vectors, equation (6) is equivalent to equation (4). Comparing to the vanilla SupCon loss, the major advantage of the SupCon-ST loss is that it allows the labels to be arbitrary real vectors, so that we can employ label smoothing and mixup in the contrastive learning framework. SupCon-ST also makes it possible to enhance supervised contrastive learning with other training schemes that use soft targets like knowledge distillation.

Classifier Training and Unknown Detection

The second phase of the framework is training a light-weight classifion network $f(\cdot)$ on top of the feature encoder $\phi(\cdot)$. In this phase, we still uses RandAugment and label smoothing to preprocess training data.

Given a training example $(\boldsymbol{x}, \boldsymbol{y})$, the probability of $\boldsymbol{x}$ belonging to class i is estimated by the softmax function:

$$y_i' = \mathbb{P}(y_i = 1 | \boldsymbol{x}) = \frac{e^{f_i(\boldsymbol{\phi}(\boldsymbol{x}))}}{\sum_{j=1}^{k} e^{f_j(\boldsymbol{\phi}(\boldsymbol{x}))}}, \tag{7}$$

then the cross entropy loss is derived as:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = -\sum_i y_i \log y_i'. \tag{8}$$

The parameters of $f(\cdot)$ are optimized by minimizing the loss, while the parameters of $\phi(\cdot)$ are fixed.

At the end of the training phase, the rejection thresholds for detecting unknown instances are estimated. For each training example $(\boldsymbol{x}, \boldsymbol{y})$ in class i, if $i = \arg\max_j f_j(\phi(\boldsymbol{x}))$, i.e. $(\boldsymbol{x}, \boldsymbol{y})$ is correctly classified, then the output logit $f_i(\phi(\boldsymbol{x}))$ is added to the class-wise logit set T_i. After all training examples are processed, the λ percentile of each set T_i is recorded as the class-wise rejection theshold ϵ_i.

During the test phase, a test sample $\boldsymbol{x}$ is labeled as an unknown sample if $\max_i f_i(\phi(\boldsymbol{x})) < \epsilon_i$. The rejection thresholds can be manually adjusted by tuning the hyper-parameter λ. We set the $\lambda = 5$ by default, which represents the desired false negative rate on the training set. However, the actual false negative rate on the test data is usually higher than λ.

Analysis

In this section, we put forward some analysis for the proposed SupCon-ST method. For the conciseness of equations, we denote the components in equation (6) as:

$$S_{ij} = \frac{s(\boldsymbol{y}_i, \boldsymbol{y}_j)}{\sum\limits_{k \neq i} s(\boldsymbol{y}_i, \boldsymbol{y}_k)}, P_{ij} = \frac{\exp(\boldsymbol{z_i} \cdot \boldsymbol{z_j}/\tau)}{\sum\limits_{k \neq i} \exp(\boldsymbol{z_i} \cdot \boldsymbol{z_k}/\tau)}, \tag{9}$$

so that the SupCon-ST loss function can be rewritten as:

$$\mathcal{L}^{scst} = \sum_i \mathcal{L}_i = \sum_i \sum_{j \neq i} \mathcal{L}_{ij} = -\sum_i \sum_{j \neq i} S_{ij} log P_{ij}. \tag{10}$$

Gradient Derivation of SupCon-ST Loss

In this subsection, we study the property of the SupCon-ST by analyzing its gradient derivation. We start by analyzing the gradients for the pairwise losses with respect to a specific sample $(\boldsymbol{x_i}, \boldsymbol{y_i})$ when the sample plays three different roles.

When $(\boldsymbol{x_i}, \boldsymbol{y_i})$ is the anchor, the gradient for the pairwise loss $\mathcal{L}_{ij}$ with respect to the projection vector $\boldsymbol{z}_i$ is:

$$\begin{aligned} \frac{\partial \mathcal{L}_{ij}}{\partial \boldsymbol{z_i}} &= -\frac{S_{ij}}{\tau} \{\boldsymbol{z_j} - \frac{\sum\limits_{k \neq i} \boldsymbol{z_k} \exp(\boldsymbol{z_i} \cdot \boldsymbol{z_k}/\tau)}{\sum\limits_{k \neq i} \exp(\boldsymbol{z_i} \cdot \boldsymbol{z_k}/\tau)}\} \\ &= -\frac{S_{ij}}{\tau} (\boldsymbol{z_j} - \sum_{k \neq i} P_{ik} \boldsymbol{z_k}). \end{aligned} \tag{11}$$

Similarly, when $(\boldsymbol{x_i}, \boldsymbol{y_i})$ is the positive sample:

$$\begin{aligned} \frac{\partial \mathcal{L}_{ji}}{\partial \boldsymbol{z_i}} &= -\frac{S_{ji}}{\tau} \{\boldsymbol{z_j} - \frac{\boldsymbol{z_j} \exp(\boldsymbol{z_j} \cdot \boldsymbol{z_i}/\tau)}{\sum\limits_{k \neq j} \exp(\boldsymbol{z_j} \cdot \boldsymbol{z_k}/\tau)}\} \\ &= -\frac{S_{ji}}{\tau} (1 - P_{ji}) \boldsymbol{z_j}. \end{aligned} \tag{12}$$

When $(\boldsymbol{x_i}, \boldsymbol{y_i})$ is a negative sample in the contrast loss of the anchor $(\boldsymbol{x_j}, \boldsymbol{y_j})$ and the positive example $(\boldsymbol{x_n}, \boldsymbol{y_n})$, the gradient for $\mathcal{L}_{jn}$ with respect to $\boldsymbol{z}_i$ is:

$$\frac{\partial \mathcal{L}_{jn}}{\partial \boldsymbol{z_i}} = -\frac{S_{jn}}{\tau} \{\frac{\boldsymbol{z_j} \exp(\boldsymbol{z_j} \cdot \boldsymbol{z_i}/\tau)}{\sum\limits_{k \neq j} \exp(\boldsymbol{z_j} \cdot \boldsymbol{z_k}/\tau)}\} = \frac{S_{jn} P_{ji}}{\tau} \boldsymbol{z_j}. \tag{13}$$

Then we can derive the gradients for sample-wise contrastive losses $\mathcal{L}_i$ and $\mathcal{L}_j$:

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{z_i}} &= \sum_{j \neq i} \frac{\partial \mathcal{L}_{ij}}{\partial \boldsymbol{z_i}} = -\frac{1}{\tau} (\sum_{j \neq i} S_{ij} \boldsymbol{z_j} - \sum_{j \neq i} S_{ij} \sum_{k \neq i} P_{ik} \boldsymbol{z_k}) \\ &= \frac{1}{\tau} \sum_{j \neq i} (P_{ij} - S_{ij}) \boldsymbol{z_j}. \end{aligned} \tag{14}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_j}{\partial \boldsymbol{z_i}} &= \frac{\partial \mathcal{L}_{ji}}{\partial \boldsymbol{z_i}} + \sum_{n \notin \{i,j\}} \frac{\partial \mathcal{L}_{jn}}{\partial \boldsymbol{z_i}} \\ &= \frac{1}{\tau}(S_{ji}P_{ji}\boldsymbol{z_j} - S_{ij}\boldsymbol{z_j} + \sum_{n \notin \{i,j\}} S_{jn}P_{ji}\boldsymbol{z_j}) \\ &= \frac{1}{\tau}(\sum_{n \neq j} S_{jn}P_{ji}\boldsymbol{z_j} - S_{ij}\boldsymbol{z_j}) = \frac{1}{\tau}(P_{ji} - S_{ji})\boldsymbol{z_j}.\end{aligned} \tag{15}$$

Finally, we get the gradients for $\mathcal{L}^{scst}$:

$$\begin{aligned}\frac{\partial \mathcal{L}^{scst}}{\partial \boldsymbol{z_i}} &= \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{z_i}} + \sum_{j \neq i} \frac{\partial \mathcal{L}_j}{\partial \boldsymbol{z_i}} \\ &= \frac{1}{\tau}\sum_{j \neq i}(P_{ij} + P_{ji} - S_{ij} - S_{ji})\boldsymbol{z_j}.\end{aligned} \tag{16}$$

The final form of gradients is simple and easy to explain. Given the anchor $\boldsymbol{z}_i$, P_{ij} can be seen as the estimated probability distribution of the positive sample $\boldsymbol{z}_j$, while S_{ij} is the expected output calculated from labels of samples. Minimizing $\mathcal{L}^{scst}$ optimizes the network parameters to make P_{ij} align with S_{ij}.

Rethinking the Properties of Contrastive Learning

As has been discussed in (Khosla et al. 2020) the self-supervised InfoNCE loss (Chen et al. 2020b) is a special case of SupCon. The differences between these loss functions and SupCon-ST are caused by different definitions of $s(\cdot, \cdot)$. We discuss the properties of different contrastive losses in this subsection.

Two key properties of the InfoNCE contrastive loss are pinpointed in (Wang and Isola 2020). **Alignment**: the feature encoding of the anchor should be close to the encodings of positive examples. **Uniformity**: normalized feature vectors should be uniformly distributed on the unit hypersphere. According to this analysis, $\mathcal{L}^{scst}$ can be decomposed into $\mathcal{L}_{align}$ and $\mathcal{L}_{uniform}$ as following:

$$\begin{aligned}\mathcal{L}^{scst} &= -\sum_i \sum_{j \neq i} \frac{S_{ij}}{\tau} \boldsymbol{z_i} \cdot \boldsymbol{z_j} + \sum_i \log \sum_{k \neq i} \exp(\frac{\boldsymbol{z_i} \cdot \boldsymbol{z_k}}{\tau}) \\ &= \mathcal{L}_{align} + \mathcal{L}_{uniform}.\end{aligned} \tag{17}$$

From this decomposition, we can see that the definition of $s(\cdot, \cdot)$ only affects $\mathcal{L}_{align}$. S_{ij} represents the magnitude of alignment between the pair of samples i and j. Self-supervised InfoNCE only aligns the anchor with its alternative view, and SupCon extends the range of alignment to the samples from the same class. SupCon-ST further extends the range of alignment to all the samples. There is a perfectly aligned encoder that maps all the inputs to a single feature vector, but the uniformity property prevents the existence of this feature collapse.

On the other hand, $\mathcal{L}_{uniform}$ is irrelevant with $s(\cdot, \cdot)$, therefore the uniformity property is identical in all variations of the InfoNCE loss. $\mathcal{L}_{uniform}$ is minimized when the distribution of feature encodings follows the uniform distribution on the unit hypersphere.

The uniformity property also induces the intrinsic hard negative mining property. Specifically, when $\tau \to 0^+$, we have the following approximation of $\mathcal{L}_{uniform}$ with respect to the anchor i:

$$\begin{aligned}\lim_{\tau \to 0^+} \mathcal{L}^i_{uniform} &= \lim_{\tau \to 0^+} \log \sum_{j \neq i} \exp(\boldsymbol{z_i} \cdot \boldsymbol{z_j}/\tau) \\ &= \lim_{\tau \to 0^+} \frac{1}{\tau} \max_{j \neq i} \boldsymbol{z_i} \cdot \boldsymbol{z_j}.\end{aligned} \tag{18}$$

When τ is small, the uniformity loss concentrates on pushing away the nearest samples. However, the uniformity loss does not consider the semantic similarity between samples. As a result, the ability of mining hard negative samples is weakened in supervised contrastive learning, where the nearest samples are more likely to belong to the same class as the anchor.

This phenomenon is described as the *negative-positive-coupling effect* in (Yeh et al. 2022), which also proposes a decoupled contrastive loss to remove this effect. The decoupled contrastive loss removes the positive example from the sum-up term in the denominator of the InfoNCE loss. From the above analysis, we can see that this modification removes the positive examples from $\mathcal{L}_{uniform}$, so that $\mathcal{L}_{uniform}$ can focus on negative examples. We consider implementing the decoupling modification in SupCon-ST as a potential improvement in future research.

Experiment

In this section, we experimentally compare the proposed method with state-of-the-art OSR methods on benchmark datasets. The performances of the proposed method in conventional closed-set classification and open-set recognition tasks are tested. In all the experiments, the feature encoder backbone of ConOSR is the same as that used in (Neal et al. 2018). The projection network in the contrastive learning step is an MLP with two fully connected layers, both consisting of 128 nodes. The classification network is also an MLP with a 128-node fully connected layer. An implementations of our method can be found at https://github.com/NJU-RINC/ConOSR.

Unknown Detection

Recent research works on OSR usually follow the protocol defined in (Neal et al. 2018). A multi-class classification dataset is divided into two subsets by randomly selecting k classes as known data, leaving the remaining classes to simulate the open space in OSR scenarios. The split of the dataset significantly affects the results of OSR experiments. The performance of a deep OSR network is also positively related to the learning ability of its backbone network. Therefore, for a fair comparison, we use the same backbone network and dataset splits with ARPL (Chen et al. 2021).

The benchmark datasets are listed as following:

MNIST \ SVHN \ CIFAR-10: These datasets are classification datasets with 10 classes, of which 6 classes are

Dataset	MNIST	SVHN	CIFAR-10	CIFAR+10	CIFAR+50	TinyImageNet
Openness	22.54%	22.54%	22.54%	46.55%	72.78%	68.38%
Softmax	97.8	88.6	67.7	81.6	80.5	57.7
OpenMax	98.1	89.4	69.5	81.7	79.6	57.6
G-OpenMax	98.4	89.6	67.5	82.7	81.9	58.0
OSRCI	98.9	91.0	69.9	83.8	82.7	58.6
C2AE	98.9	89.2	71.1	81.0	80.3	58.1
RPL++	99.3	95.1	86.1	85.6	85.0	70.2
GFROSR	N.R	93.5	80.7	92.8	92.6	60.8
PROSER	N.R	94.3	89.1	96.0	95.3	69.3
ARPL	**99.7**	96.7	91.0	97.1	95.1	78.2
ConOSR (vanilla SupCon)	**99.7**	98.8	93.7	97.9	97.0	79.6
ConOSR (SupCon-ST)	**99.7**	**99.1**	**94.2**	**98.1**	**97.3**	**80.9**

Table 1: Open Set recognition results in terms of the AUC-ROC curve. Results are averaged among five trials. N.R means the original paper does not report the corresponding result.

selected as known classes and the other 4 classes are used as unknown.

CIFAR+10 \ CIFAR+50: 4 classes from CIFAR-10 are selected as known classes, and $10\backslash 50$ classes selected from CIFAR-100 are used as unknown.

TinyImageNet: TinyImageNet consists of 200 classes. We select 20 classes as known classes and use the remaining 180 classes as unknown.

On each benchmark dataset, we conduct the experiment over five trials using the same data split as (Chen et al. 2021), and report the mean results. Area Under the ROC curve (AUROC) is used as evaluation metric. AUROC is a threshold-independent metric which can be interpreted as the probability that a positive example is assigned a higher detection score than a negative example (Geng, Huang, and Chen 2020).

The complexity of each OSR experiment is measured by *Openness*, defined as $Openness = 1 - \sqrt{K/M}$ in (Neal et al. 2018), where K and M denote the number of training and test classes respectively.

In these experiments, we compare our method with the baselines including Softmax Thresholding (Hendrycks and Gimpel 2016), OpenMax (Bendale and Boult 2016), G-OpenMax (Ge et al. 2017), OSRCI (Neal et al. 2018), C2AE (Oza and Patel 2019), RPL++ (Chen et al. 2020a), GFROSR (Perera et al. 2020), PROSER (Zhou, Ye, and Zhan 2021), and ARPL (Chen et al. 2021)

Table 1 shows the results of this experiment. The baseline performances are cited from (Zhou, Ye, and Zhan 2021; Chen et al. 2021). N.R means that the original paper has not reported the corresponding result. We report the results of two variations of the ConOSR framework. The first variation uses the vanilla SupCon algorithm in the contrastive learning step, while the second uses the proposed SupCon-ST.

From Table 1, we can see that almost all the methods have achieved good results on digital number datasets MNIST and SVHN. In particular, the results on MNIST are almost saturated. However, our method still raises the AUROC on SVHN to 99.1. On natural image datasets, our method also achieves better results than SOTA methods PROSER and ARPL. Compared with the second best method ARPL, ConOSR with SupCon-ST improves the results on TinyImageNet by a margin of 2.7.

The results in Table 1 also show that better unknown detection results can be achieved by replacing the vanilla SupCon with SupCon-ST. SupCon-ST improves the AUROC by $0.2 - 0.5$ on simple datasets such as SVHN and CIFAR, and its advantage increases to 1.3 on the most challenging dataset TinyImageNet.

Closed Set Classification

We validate the effectiveness of the proposed SupCon-ST on conventional classification tasks by comparing it with vanilla SupCon and plain CNN. When training Plain CNNs, we use the same data augmentation methods as SupCon-ST. In this group of experiments, we train the network on the full sets of CIFAR-10/100, and the first 100 classes of TinyImageNet. The averaged results over 5 random trials are reported in Table 2.

Dataset	CIFAR-10	CIFAR-100	TinyImageNet
Plain CNN	94.0	71.6	63.7
ARPL	94.1	72.1	N.R.
SupCon	94.1	72.4	63.7
SupCon-ST	**94.6**	**73.0**	**66.1**

Table 2: Comparison of average closed set classification accuracy.

From the results, we can see that the accuracy of vanilla SupCon is similar to that of the plain CNN, while SupConST outperforms them by a relatively large margin. These results indicates that supervised contrastive learning boosts the classification accuracy on traditional closed-set recognition tasks. However, due to its incompatibility with soft targets, vanilla SupCon uses less training tricks than others, which leads to its interior performance.

We also cited the results of ARPL (Chen et al. 2021) for comparison. ARPL uses ResNet-34 as its backbone, which

is a stronger network than the backbone in our implementations. On the other hand, the authors of (Chen et al. 2021) do not apply as many data augmentations as we do in their experiments. To the extent of our knowledge, the majority of existing OSR methods have report degraded results in closed-set classification tasks. ARPL is one of the methods that outperform the plain CNN baseline. ARPL also sets its basis on improving the representation learning part of the OSR system, and hence boots its classification accuracy in conventional closed-set tasks. The strong positive relationship between closed-set accuracy and OSR performance has been studied in (Vaze et al. 2021).

Open Set Recognition

We use another group of experiments to verify the performance of the proposed method in OSR tasks. In these experiments, we follow the protocol used in (Zhou, Ye, and Zhan 2021). At training time, the whole dataset is used for training OSR models. During testing, samples from another dataset are added to the test set, and combined as a new class. The evaluation metric in these experiments is macro-averaged F1-scores over all the classes in the training set and the novel class of unknown test samples, so that the performances on both known and unknown data are evaluated.

We conduct the first experiment using MNIST as the training set, and test samples from three other datasets: Omniglot (Lake, Salakhutdinov, and Tenenbaum 2015), MNIST-Noise, and Noise. Following (Zhou, Ye, and Zhan 2021), we set the number of unknown samples as 10,000 so that their number is equal to the number of test samples of the known classes. The test set of Omniglot contains 13,180 samples, so we select the first 10,000 images sorted by ascending index of file names. We synthesize the Noise dataset by sampling each pixel of generated images between [0, 1] from a uniform distribution. MNIST-Noise is synthesized by adding the generated noise images atop the MNIST testing samples.

Dataset	Omniglot	Noise-MNIST	Noise
Softmax	59.5	64.1	82.9
OpenMax	68.0	72.0	82.6
CROSR	79.3	82.7	82.6
PROSER	86.2	87.4	88.2
ConOSR	**95.4**	**98.7**	**98.8**

Table 3: Open set recognition on MNIST with samples from various datasets added to the test set. We report macro F1 in 11 classes.

The second experiment uses CIFAR-10 as the training set, and introduces the test sets of two other data sets as unknown samples: TinyImageNet and LSUN (Yu et al. 2015). CIFAR-10, TinyImageNet and LSUN all have a test set consisting of 10,000 images. To remove the difference of image size between CIFAR-10 and TinyImageNet & LSUN, we use two different ways to process the unknown images: (1) resizing the images to 32×32; (2) cropping a 32×32 patch from each image.

Dataset	TIN (Crop)	TIN (Resize)	LSUN (Crop)	LSUN (Resize)
Softmax	63.9	65.3	64.2	64.7
OpenMax	66.0	68.4	65.7	66.8
OSRCI	63.6	63.5	65.0	64.8
CROSR	72.1	73.5	72.0	74.9
GFROSR	75.7	79.2	75.1	80.5
PROSER	84.9	82.4	86.7	85.6
ConOSR	**89.1**	**84.3**	**91.2**	**88.1**

Table 4: Open set recognition on CIFAR-10 with samples from TinyImageNet (TIN) and LSUN datasets added to the test set. We report macro F1 in 11 classes.

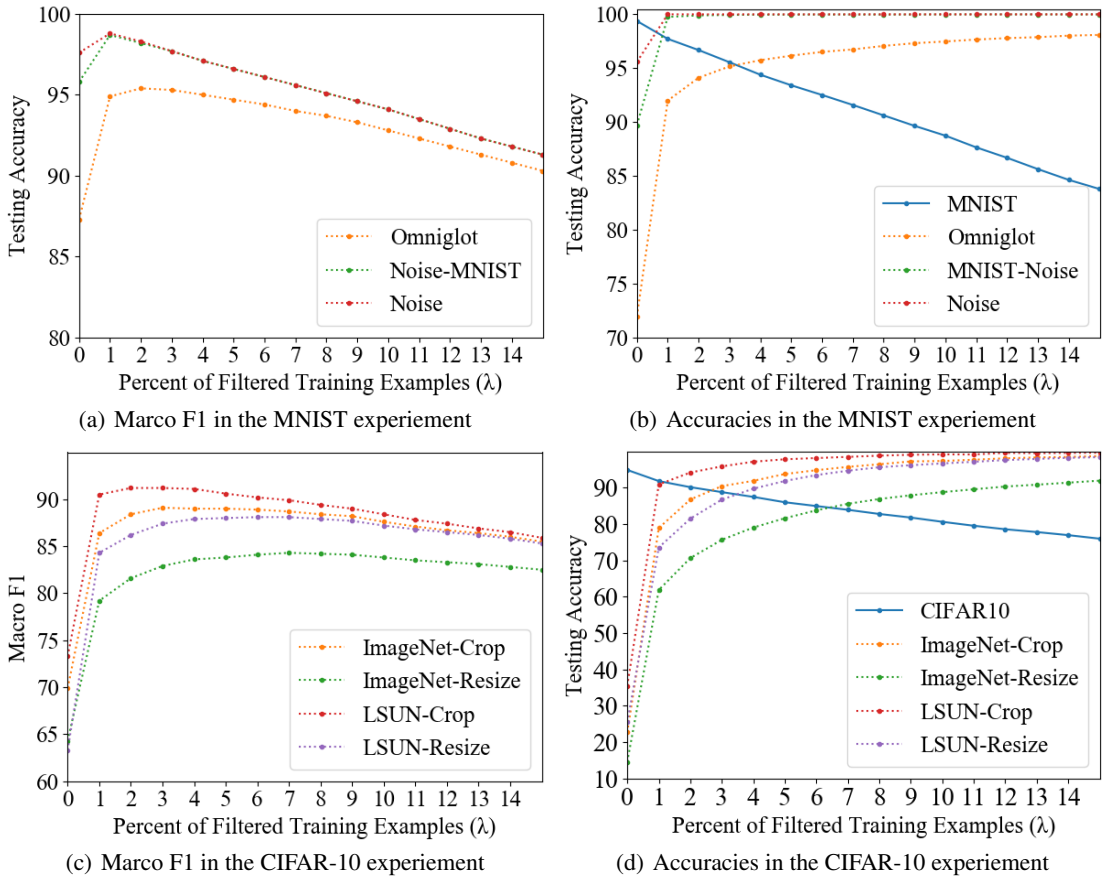
The results of these experiments are shown in Table 3 and Table 4, where the results of other methods are cited from (Zhou, Ye, and Zhan 2021). The balance of classification accuracy between unkown and unknown classes is largely affected by the hyperparameter λ, thus affecting the F1 score. Therefore, we optimize λ through grid search in $[1, 15]$ and report the best macro F1.

In Table 3, we can see that when the background of training images is clean, detecting noisy images is a simple task. Detecting samples from Omniglot is most challenging, mainly because the unknown samples are as clean as the training examples. In this group of experiments, our proposed method significantly outperforms the other methods. The accuracy gap between ConOSR and the second best method is larger than 10% when unknown samples come from Noise-MNIST and Noise.

ConOSR also outperforms other methods on all the datasets in the second experiment. We can see from Table 4 that the advantage of OSR is more obvious when unknown samples are obtained by cropping. This phenomenon indicates that ConOSR works better at detecting semantically meaningless images, because the contrastive learning algorithm focuses on learning the most distinctive features between classes, while patches randomly cropped from large images often contain fewer such features. On the resized datasets, ConOSR has an advantage of about 2% compared with the second best method PROSER.

The macro-F1 metric is easily affected by the value of the hyper-parameter λ, so we use this group of experiments to analyse how λ affects the results. We set the value of λ by grid search in range $[0, 15]$, and show how the macro F1 scores and accuracies of known classes and the class of unknown examples change with it. The results with varying λ are illustrated in Fig.4.

Naturally, increasing λ increases the classification accuracy of unknown instances while reducing classification accuracy of known instances. In simple tasks, such as detecting noise outliers from MNIST images, the accuracy of unknown instances easily reaches 100% when $\lambda = 1$, so further increasing λ only results in degraded results. In the CIFAR-10 experiment, the accuracy of unknown instances could not reach its limit without setting a large λ. The best macro F1 scores are usually obtained near the points where



(a) Marco F1 in the MNIST experiement

(b) Accuracies in the MNIST experiement

(c) Marco F1 in the CIFAR-10 experiement

(d) Accuracies in the CIFAR-10 experiement

Figure 4: Classification accuracy and macro F1 against varying λ.

the accuracies of known instances approximate the accuracies of unknown instances, and remain stable before the testing accuracies for unknown instances are approximately saturated. The recommended default value $\lambda = 5$ results in good F1 scores in the CIFAR-10 experiments.

Analytical Experiment

We conduct another experiment to analyse the reason why contrastive learning could boost the ability of open set recognition. Here, we first put forward a brief analysis.

Similar to many DNN-based discriminative OSR methods, ConOSR detects unknown samples by thresholding the network outputs, and rejects the samples with low outputs. We can infer from the principle of DNNs that this detection mechanism rejects test images which do not contain enough features to sufficiently activate the nodes in the penultimate layer. In other words, these methods work by detecting the absence of necessary features for identifying a test sample as any known class, rather than detecting novel features occurring in the image. A recent study (Dietterich and Guyer 2022) names this property "the familiarity hypothesis", and presents strong evidence to support this hypothesis.

Compared with plain CNN networks, supervised contrastive learning focuses on learning the most distinctive features between known classes. As a result, these features are less likely to exist in the unknown samples, hence reducing the difficulty of unknown detection. On the other hand, they could not generalize well on another domain.

In order to validate the analysis above, we conduct an experiment to compare contrastively learned features with the features learned by plain CNN networks on the CIFAR-100 and TinyImageNet datasets. Each dataset is divided into two halves according to label index. The first half is used as the training data in OSR tasks, and the second half is used to simulate the unknown data in the testing phase. We first train and test the models under the common OSR protocol, recording the AUROC scores and closed-set classification accuracies. Then, the parameters of feature encoders are fixed, and new classifiers are trained atop them with the training data of unknown classes. Finally, we record the accurracies of the new classifiers on the unknown classes to see how the features generalize on the unknown data.

The results of this experiment are shown in Table 5. The closed set classification results on known classes are similar to the results in Table 2. SupCon and plain CNN achieve comparable accuracies, while SupCon-ST significantly outperforms both of them. When we transform the feature encoders to another domain, plain CNN in turn outperforms contrastive learning methods. Comparing the AUROC scores, we can see that both variations of ConOSR are

Dataset	CIFAR-100			TinyImageNet		
Metric	Accuracy (Known)	Accuracy (Unknown)	AUROC	Accuracy (Known)	Accuracy (Unknown)	AUROC
Plain CNN	77.2	**62.6**	76.7	63.7	**49.1**	68.1
ConOSR (SupCon)	77.8	59.3	77.9	63.8	41.3	71.6
ConOSR (SupCon-ST)	**79.5**	60.5	**79.1**	**66.1**	45.4	**72.1**

Table 5: Comparison of OSR performance and transferrability of feature representations.

better at unknown detection than the plain CNN. Specifically, the vanilla SupCon gets similar results with plain CNN in terms of closed-set classification accuracy, but still achieves much better AUROC scores. SupCon-ST outperforms the vanilla SupCon in terms of all evaluation metrics, suggesting the superiority of using mixup and label smoothing.

The results in this experiment provide support for our analysis above, which suggests that the supervised contrastive learning is more "focused" on distinguishing the classes they learned. As a result, the learned features could not be transferred to a totally novel domain as well as the commonly learned features. However, this property makes it easier for classifiers to detect the absence of features, which is beneficial for their performances in OSR tasks.

In order to better present this property of the proposed method, we use class activation maps to illustrate the difference between the features learned by contrastive learning and plain CNN networks. We randomly choose 4 pairs of known/unknown images from CIFAR-100, computing the class acvtivation maps of both images using the classifier weights of the known classes. In each pair, the heatmaps are computed according to the minimum/maximum activation value in the two images. The class activation maps are shown in Fig.5. From the comparison of class activation

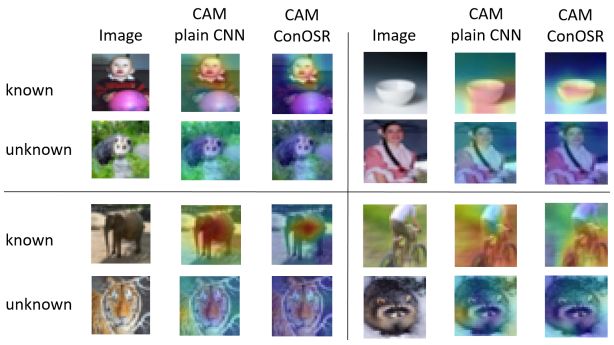


Figure 5: Class activation maps (CAMs) of plain CNN networks and the proposed ConOSR. We randomly choose 4 pairs of known/unknown images from CIFAR-100, computing the class activation maps of both images using the weights of the known classes.

maps, we can see that the hot zones in ConOSR CAMs are much smaller than the hot zones in plain CNN CAMs, focusing on the important part of the objective. By comparing the CAMs of unknown images, we can see that the colors of most areas in ConOSR CAMs are much deeper than the CAMs of plain CNNs, indicating that ConOSR is not interested in any regions of the unknown images. These results also support our analysis that contrastive learning learns the most discriminative features of each class, making it easier to detect the absence of important features.

Conclusion

In real world recognition scenarios, collecting training examples to cover the categories of all potential test instances is difficult. Open set recognition (OSR) is a realistic type of recognition task targeting this difficulty, which requires the classifiers to distinguish test samples from unseen classes while maintaining a high classification accuracy of seen classes. From a representation learning perspective, we propose a contrastive learning method for OSR (ConOSR) based on Supervised Contrastive Learning with Soft Targets (SupCon-ST). With the SupCon-ST, we are able to utilize label smoothing and mixup in the contrastive training phase, resulting in deep networks with better robustness in OSR tasks and better accuracy in closed-set classification.

However, the proposed method is not computationally efficient compared to common deep learning methods. First, contrastive learning requires more training epochs to converge than conventional training pipelines. Second, SupCon-ST requires more GPU memory to work properly. In our experiments, we have to vary the batch-size of training data regarding the number of classes, so that each mini-batch contains a few positive pairs of examples from each class. What makes it worse is that, for a mini-batch of n training examples in the contrastive learning phase, $2n$ views are generated via RandAugment, and another $2n$ virtual examples are generated via mixup. Therefore, the cost of memory space increases drastically as the number of classes grows.

In future, we will study how to combine the proposed method with clustering methods, so that our method could work with less space cost. Extending open set recognition to life-long learning scenarios is also an interesting direction for future research.

Acknowledgements

This work was supported in part by the STI 2030-Major Projects of China under Grant 2021ZD0201300, and by the National Science Foundation of China under Grant 62276127.

References

Bendale, A.; and Boult, T. E. 2016. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1563–1572.

Chen, G.; Peng, P.; Wang, X.; and Tian, Y. 2021. Adversarial Reciprocal Points Learning for Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Chen, G.; Qiao, L.; Shi, Y.; Peng, P.; Li, J.; Huang, T.; Pu, S.; and Tian, Y. 2020a. Learning Open Set Network with Discriminative Reciprocal Points. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 507–522. Cham: Springer International Publishing. ISBN 978-3-030-58580-8.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750–15758.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 702–703.

Dietterich, T. G.; and Guyer, A. 2022. The Familiarity Hypothesis: Explaining the Behavior of Deep Open Set Methods. *arXiv preprint arXiv:2203.02486*.

Ge, Z.; Demyanov, S.; Chen, Z.; and Garnavi, R. 2017. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*.

Geng, C.; Huang, S.-j.; and Chen, S. 2020. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Bing, X.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In *International Conference on Neural Information Processing Systems*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33: 21271–21284.

Guo, Y.; Camporese, G.; Yang, W.; Sperduti, A.; and Ballan, L. 2021. Conditional Variational Capsule Network for Open Set Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 103–111.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673.

Kong, S.; and Ramanan, D. 2021. OpenGAN: Open-Set Recognition via Open Data Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 813–822.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266): 1332–1338.

Neal, L.; Olson, M.; Fern, X.; Wong, W.-K.; and Li, F. 2018. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 613–628.

Oza, P.; and Patel, V. M. 2019. C2AE: Class Conditioned Auto-Encoder for Open-Set Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Perera, P.; Morariu, V. I.; Jain, R.; Manjunatha, V.; Wigington, C.; Ordonez, V.; and Patel, V. M. 2020. Generative-discriminative feature representations for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11814–11823.

Scheirer, W. J.; de Rezende Rocha, A.; Sapkota, A.; and Boult, T. E. 2012. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7): 1757–1772.

Van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints*, arXiv–1807.

Vaze, S.; Han, K.; Vedaldi, A.; and Zisserman, A. 2021. Open-Set Recognition: A Good Closed-Set Classifier is All You Need. In *International Conference on Learning Representations*.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 9929–9939. PMLR.

Yeh, C.-H.; Hong, C.-Y.; Hsu, Y.-C.; Liu, T.-L.; Chen, Y.; and LeCun, Y. 2022. Decoupled contrastive learning. In *European Conference on Computer Vision*, 668–684. Springer.

Yoshihashi, R.; Shao, W.; Kawakami, R.; You, S.; Iida, M.; and Naemura, T. 2019. Classification-Reconstruction Learning for Open-Set Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu, F.; Zhang, Y.; Song, S.; Seff, A.; and Xiao, J. 2015. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv preprint arXiv:1506.03365*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2021. Learning Placeholders for Open-Set Recognition. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4401–4410.