Generative-discriminative Feature Representations for Open-set Recognition

Pramuditha Perera*
Johns Hopkins University
pperera3@jhu.edu

Vlad I. Morariu
Adobe Research
morariu@adobe.com

Rajiv Jain
Adobe Research
rajijain@adobe.com

Varun Manjunatha
Adobe Research
vmanjuna@adobe.com

Curtis Wigington
Adobe Research
wigingto@adobe.com

Vicente Ordonez
University of Virginia
vicente@virginia.edu

Vishal M. Patel
Johns Hopkins University
vpatel36@jhu.edu

Abstract

We address the problem of open-set recognition, where the goal is to determine if a given sample belongs to one of the classes used for training a model (known classes). The main challenge in open-set recognition is to disentangle open-set samples that produce high class activations from known-set samples. We propose two techniques to force class activations of open-set samples to be low. First, we train a generative model for all known classes and then augment the input with the representation obtained from the generative model to learn a classifier. This network learns to associate high classification probabilities both when the image content is from the correct class as well as when the input and the reconstructed image are consistent with each other. Second, we use self-supervision to force the network to learn more informative features when assigning class scores to improve separation of classes from each other and from open-set samples. We evaluate the performance of the proposed method with recent open-set recognition works across three datasets, where we obtain state-of-the-art results.

1. Introduction

Supervised classification systems are trained with the knowledge of a finite set of labeled training examples. When training data comes from k distinct known classes, a deep network classifier simultaneously learns a descriptive feature space and a decision rule that segments the feature space into k non-overlapping regions as shown in Figure 1(a). When an object outside the known class set (known as a novel object or an open-set object) is introduced to the network, the network will still associate it with one of the known k classes (Figure 1(b)). The goal of open-set detection[1] is to identify whether a given object instance belongs to the known class set or not. Once identified, open-set samples can be either discarded to prevent wrong association or used to improve the classification system [2], [23].

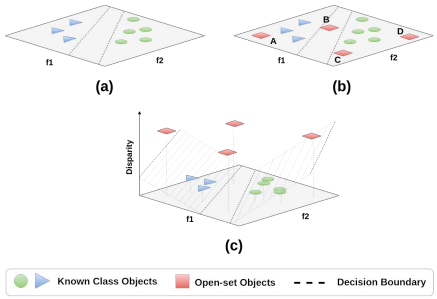

Figure 1. (a) Given a set of known classes, a classifier defines a positive half space for each class by *only* considering class separation. (b) An open-set object could project either near a decision boundary (samples B and C) or deep into the positive half space (samples A and D) of a given class. The latter is harder to detect when the class activation scores are considered. (c) We learn a classifier which takes into account more factors than just class separation. First, we use self-supervision to make the feature space more descriptive to force open-set samples to separate better from known classes. Second, we augment the feature space with a generative representation. The disparity between input images and their representations are taken into consideration when the decision boundaries are defined. When open-set samples produce high disparity, better open-set sample detection is obtained.

A straightforward solution to the stated problem is to threshold the probability of the most probable class produced by the network [12]. CNNs are trained with the objective of maximizing the probability of the correct class over the training data. Therefore, if the training process generalizes well enough, query samples from known classes

*This work was completed while the author was working as an intern at Adobe Research.

[1]The terms *open-set detection* and *multiple-class novelty detection* have been used interchangeably in the literature. We make no distinction between these two terms in this paper.

can be expected to produce high probabilities. However, the open-set recognition literature [26] points out the possibility of novel object samples producing equally high probabilities.

When a discriminative classifier is trained, it learns a set of features that are needed to discriminate between the known classes. In the ideal case, features that are not essential to separate the known classes are discarded during the learning process. We refer to these features as *optimal closed-set features*. However, *optimal closed-set features* are likely insufficient for capturing differences between open-set samples and known-classes [22]–additional features are likely required to separate the known classes and open-set samples. Open-set samples could end up producing high class-activations, depending on where in the feature space they are projected.

We investigate two techniques that reduce this effect. First, we extend *optimal closed-set features* so that features have the capacity to describe shapes, structure and semantics of known-class objects. During training, the classifier will consider the overall semantics of images (not just the discriminative aspects) when class decision boundaries are defined. As a result, open-set images will not be positioned in any of the positive half-spaces on the grounds of having different semantics. We obtain such diverse features by incorporating self-supervision in learning.

Second, we model the known-class objects using a generative model. Then, a classifier is learned by considering both the input image and its generative representation. The classifier will take into account the correspondence between the two inputs when the decision boundaries are obtained as shown in Figure 1(c). Since the generative model is trained using known-class images, it will not represent open-set samples well. As a result, open-set samples will demonstrate high disparity (Figure 1(c)) thereby getting projected out-side the positive half spaces of known classes.

Both of these techniques are aimed at enhancing the *optimal closed-set features* to contain richer content-specific features. Additional knowledge provided by features helps the network to effectively identify open-set samples. Our contributions are the following:

1) We learn a richer deep feature space by forcing the network to learn features that capture object structure by performing self-supervision. This leads to a richer feature space for deciding whether a given sample belongs to a known class or not.
2) We train a generative model on known-class data and use the reconstructions from this model as input to the classification task. This allows the classifier to take into account the disparity between the input and a generative signal associated with the input. Open-set samples which yield a high disparity are easily detectable as shown in Figure 1(c).

2. Related Work

Open-set Recognition. Open-set recognition has received considerable attention in the computer vision community in recent years. The problem of open-set recognition was first formulated in [26], where authors pointed out the possibility of an open-set sample generating a very high activation score for one of the known class categories. Since then, several other works have analyzed this challenge in the context of deep networks [22],[11]. In [3], a $k+1$ classifier for a k class problem was used where the extra class was treated as the *open-set class*. A statistical method was used to apportion class probabilities to the open-set class. This alternative formulation, OpenMax, was proposed as an alternative to the SoftMax operator. In [7], a Generative Adversarial Network (GAN) based framework was used to estimate open-set class activations. A similar approach was taken in [16] where counterfactual images that lie between decision boundaries were used to simulate open-set class instances.

More recent works in open-set recognition have deviated from simulating open-set classes. The method proposed in [19] used a class conditioned generator to learn a representation that preserves only known-class samples. Then, open-set recognition was carried out based on the reconstruction error associated with the generator. In [29], the authors identified the importance of generative features in open-set recognition. They first learn a sophisticated generative model (an extension of a ladder network [25]) and append the learned feature with one of the classifier features. Then, an OpenMax classifier was learned using the augmented features. The feature augmentation proposed in our work is different from [29]. In [29], a generative model and a classifier are trained independently. We learn a classifier trained on the augmented input space and take into account the disparity between the two representations as we compute class activation scores.

It is also possible to use one-class classification algorithms[24],[21],[20] to solve open-set recognition by modelling known classes. However, since class labels of known-classes are not used in this approach, recognition results tend to be poor compared to standard open-set methods.

Self-Supervision. Self-supervision is an unsupervised machine learning technique where data itself provides supervision. It is usually carried out in addition to a primary objective (such as classification or detection) with the intention of producing a more generic and robust feature. Recent works in self-supervision introduced several techniques to improve the performance in classification and detection tasks. In all of these techniques, the network is forced to learn the shape structures of the underlying objects and their semantics thereby producing a richer feature.

For example, in [5], given an anchor image patch, self-supervision was carried out by asking the network to pre-

dict the relative position of a second image patch. To make such predictions, the network needs to learn object structure and relative order. In [6], a multi-task prediction framework extended this formulation, forcing the network to predict a combination of relative order and pixel color. In [8], the image was randomly rotated by a factor of 90 degrees and the network was forced to predict the angle of the transformed image. This method was simpler to implement and produced better results than previous self-supervision techniques. In our work, we follow [8] by using a series of different transformations (combination of rotating and flipping the image) in place of rotations. To the best of our knowledge this is the first attempt at using self-supervision for open-set recognition. The prediction of geometric transformations has been previously utilized in [10] in the one-class classification problem domain. However, [10] is different from our method as they used this network to generate classifier responses to characterize a signature for a given class.

3. Proposed Method

In this section, we motivate the need for a richer feature representation for effective open-set recognition. Then, we introduce conditioning on generative representation and self-supervision to overcome this challenge. Finally, we describe the proposed training and testing procedure.

3.1. Challenges in Open-set Recognition

An illustration of why open-set recognition is challenging is shown in Figure 1. When a classifier is trained, the positive half spaces of each class are identified (these half spaces are described by the vector defined using the final fully connected layer weights corresponding to the class). When a sample appears deeper in the identified positive half space, it will generate a larger class activation. On the other hand, a sample appearing near the half-space boundary will result in a lower class activation. When the network is trained, a feature embedding is learned such that each training sample is encouraged to be pushed deeper in to the positive space corresponding to its ground truth. Therefore, as long as the query samples follow the same distribution as the training samples, known-class samples are expected to produce large activation values.

Consider an open-set image that is projected onto one of the following regions:

1) Intersection of all class boundaries. This will arise when the open-set image does not have any components/regions common with any of the known classes (See points B and C in Figure 1)(b). In this case, the class activation scores of all classes will be low. These types of open-set samples may be filtered by thresholding the maximum class activation score.

2) Deep into the positive half space of a class. This situation (such as points A and D in Figure 1(b)) arises when the open-set image has a semantically similar component/region to that of a known class (or the network perceives to be so). As a result, the activation of the aforementioned class becomes high. These instances cannot be easily rejected by considering class activations. We specifically focus on the latter case and investigate techniques that can reduce class activation scores of open-set samples.

3.2. Self-Supervised Learning

When a closed-set classifier is trained, the classifier learns only features that are necessary to differentiate between known classes. However, these features are not always descriptive enough to separate out open-set samples from known classes. By introducing a more descriptive feature, we reduce the activation magnitude of open-set samples. For this purpose, we extend the conventional classification network into a multi-task network where an auxiliary classifier performs self-supervision.

We adopt the self-supervision framework proposed in [9]. In [9], a geometric transformation is applied to an input at random from a finite set of transformations, and the self-supervision branch of the network is used to predict which transformation was applied. In order to determine the transformation that was applied, the network needs to learn structural properties of image content such as shape and orientation. As a result, when a self-supervision branch is added on top of the classification task, the intermediate features becomes more descriptive.

Figure 2(a) and (b) illustrate network architectures of a conventional classification network and a classification network extended to perform self-supervision respectively. In the former case, each training instance is passed through the classification network (C) to produce a classification loss l_c. In the latter case, the classification network (C) has two output branches. Each forward pass consists of two steps. In the first step, a classification loss l_c is produced by passing the input through the open-set classification branch. During the second step, the input image is subjected to a random transformation. The transformed image is passed through the transformation classification branch to arrive at self-supervision loss l_{ss}. When evaluating the self-supervision loss, the transformation applied to the input is considered to be ground truth. The network is trained by considering a composite loss of the form $\alpha_1 l_c + \alpha_2 l_{ss}$. In our experiments, we chose $\alpha_1 = 0.8, \alpha_2 = 0.2$ with the aim of giving more importance to the primary classification task[2]. For our experiments we used 14 transformations where each transformation was formed by randomly flipping the image (horizontally and vertically) and by rotating image by multiples of 90 degrees.

[2]Please refer to the supplementary material for a sensitivity analysis of these parameters.

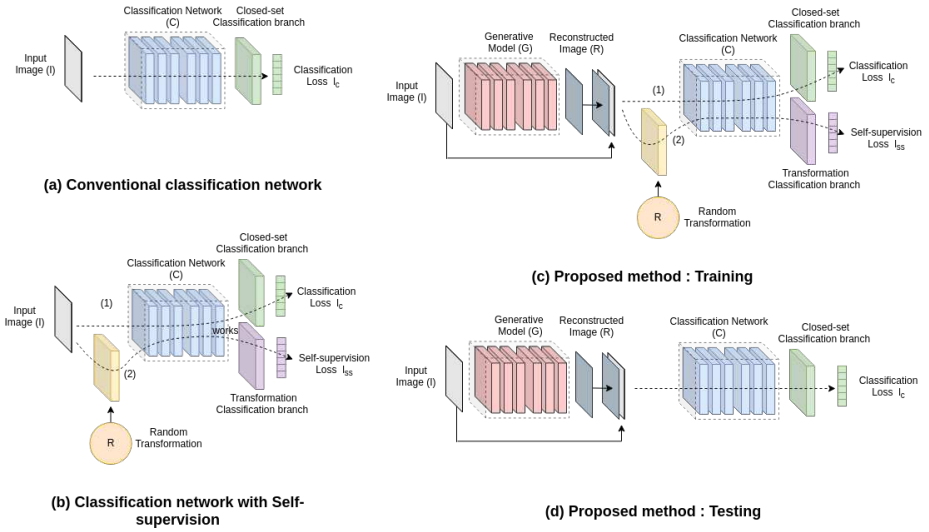

Figure 2. Comparison of the network architectures. (a) A conventional classification network. (b) A classification network with self-supervision. (c) The proposed network consists of a generative model and a self-supervision component. The input is concatenated with the reconstruction obtained from the generative model prior to feeding into the classification network C.

In section 4.3, we demonstrate the effectiveness of introducing self-supervision through an ablation study.

3.3. Augmenting with Generative Representation

As the second contribution of our work, we augment the input with its representation obtained through a generative framework. Let us first consider a generative model trained on the images of known classes. For example, the generative model can be a deep auto-encoder network. Ideally, the generative model will be able to represent and reproduce samples of known classes. On the other hand, since the generative model has not seen samples from open-set classes, it will not be able to represent (re-produce) such samples equally well. If this is the case, there will be high correspondence between input images and reconstructed images generated by the generative model for known class samples. Correspondence will be low for open-set samples.

In Figure 1(c), we illustrate the implication of augmenting a generative reconstruction to the open-set problem. In this idealistic case, we have denoted the disparity between the original image and the reconstructed image as an additional axis. Here, the disparities for known samples are smaller compared to open-set samples. In this scenario, the classifier will learn two new positive half planes defined by hyper-planes similar to that of shown in Figure 1(c). If disparity is considerably high, it will force opens-set samples to be outside the positive half space of all the classes.

Based on this intuition, we carry out the training process in two steps. First, we train a generative network (G) using training samples. Then, given an input x, we train the classification network (C) by considering the augmented input $[x, G(x)]$ as shown in Figure 2(c).

3.4. Training and Testing Procedure

Architecture. We use the network architecture proposed in [16]. The encoder network used for the autoencoder consists of 10 convolutional 3×3 layers, where each layer is followed by a batch-normalization and leaky ReLu(0.2) operation. The decoder network has a similar structure to that of the encoder and is constructed with transpose-convolution layers instead. The classifier network consists of 9 layers of 3×3 convolution filters followed by batch-normalization and leaky ReLu(0.2) operations. It is terminated using a fully-connected layer. The only difference in our classifier from [16] is that our network accepts a 6-channel image as the input.

In order to investigate the impact that different architectures have on open-set rejection performance, we vary the classifier and generative model and study the impact they have on open-set recognition. In Table 1, we tabulate open-set recognition performance in terms of AUC-ROC under different architectures across five different known-openset splits for the CIFAR10 dataset. Here, vanilla AE and vanilla CNN refers to the network architectures used in [16]. Conditioned AE [28] is a modified version of Vanilla AE, where a fully connected layer classifier is connected to the latent space. This version of the AE produces better known-openset separation in reconstructed image space due to this additional constraint. WRN28-10 and WGAN refers to standard wide-ResNet(depth 28 and width 10) [31] and Wasserstein GAN [1] respectively. According to Table 1, we observe that using a more sophisticated network, both as a generative model and a classification model have contributed towards improving average open-set recognition performance.

Generative Model	Classifier Model	Open-set Performance					
		1	2	3	4	5	Avg
Vanilla AE	Vanilla CNN	78.0	76.7	84.9	84.9	79.4	80.8
Conditioned-AE	Vanilla CNN	79.1	77.3	85.7	87.4	80.3	82.0
Conditioned-AE	WRN28-10	77.5	81.7	86.2	87.5	82.6	83.1
WGAN	WRN28-10	81.7	79.2	85.5	87.2	84.3	83.6

Table 1. Impact of using different architectures on open-set recognition on the CIFAR10 dataset. We observe that using more sophisticated generative models and classifiers both improve open-set performance.

Input : Training sample x,label y, Transformation Set T, Models: G, C , Weights α_1,α_2
Output: Models: C

Classification Step.
$\hat{x} \longleftarrow G(x)$
$z = [x, \hat{x}]$;
l_c = CrossEntropy$(C(z), y)$

Self-supervision Step.
Pick transformation randomly.
r = rand$(\Omega(T))$
$t = T[r]$;
$z = [t(x), t(\hat{x})]$;
l_{ss} = CrossEntropy$(C(z), r)$
$l_t = \alpha_1 l_c + \alpha_2 l_{ss}$
Backpropagate to change and C.

Algorithm 1: Training Algorithm

Training. We trained all networks for 1000 iterations using the Adam optimizer with a batch size 64, learning rate of 0.001 and parameters $(0.5, 0.999)$. The training process is outlined in Figure 2(c) and Algorithm 1. First, generative model G is trained using training data. Then, as descried in Algorithm 1, each training sample x is first augmented and $z = [x, \hat{x}]$ is passed through the classification branch of the network to obtain the classification loss l_c. Then, a transformation is randomly selected from the set of available transformations. If the chosen transformation index is r, the transformed image is augmented and $z = [t(x), t(\hat{x})]$ (where t is the transformation selected) is passed through the self-supervision branch to produce a self-supervision loss which is calculated using cross-entropy by considering r as the ground-truth label. The composite loss t_l is backpropagated to find gradients associated with each network weight. Finally, network C is updated according to the network updating algorithm.

Testing. During inference, the self-supervision branch of the network is disregarded as shown in Figure 2(d). Given a query image x, first the augmented representation $[x, G(x)]$ is obtained. Then, the augmented input is passed through the classifier network to obtain class activations $a = C([x, G(x)])$. If the maximum activation $max(a)$ is below a predetermined threshold γ, it is declared that the input is an open-set instance. In practice, threshold γ is determined such that a minimum true positive rate is guaranteed on a validation set. In our experiments we picked γ such that true positive ratio is at least 0.9.

4. Experimental Results

We evaluate the performance of the proposed method on standard datasets used for open-set recognition and compare with state-of-the art methods. First, we report performance on open-set recognition and out-of-distribution recognition tasks respectively. Then we consider a case study on the CIFAR10 dataset to analyze performance of the proposed method qualitatively. We conclude the latter section with an ablation study.

4.1. Open-set Recognition

Recent deep learning based open-set recognition methods followed the protocol in [16] and used the numbers reported in [16] as a baseline for comparison. In [16], an open-set recognition scenario is simulated on a multi-class classification dataset by randomly selecting n classes as known. The remaining classes are considered to be open-set classes. This protocol is used to simulate five trials of open-set recognition and performance is measured using the average area under the curve of ROC (AUC-ROC) curve.

Performance across different splits varies significantly (in our experiments AUC for CIFAR10 varied between 77% to 87% across different splits). There are many possible known-openset combinations one could consider when the above protocol is followed($\binom{10}{6}$ for CIFAR10, SVHN and $\binom{200}{20}$ for TinyImageNet). Open-set performance is highly correlated with the classifier performance. A better classifier is able to reject open-set samples more effectively (for example in [29], open-set performance improves when a DenseNet backbone is used as compared to a vanilla CNN). Therefore, for a fair comparison, we argue that all methods should use identical splits and the same network backbone.

In this spirit, we use the same autoencoder and classifier architectures as [16]. Further, we test on the same known-openset splits as [16][3]. Note that [19] used different known-openset splits in their evaluation. We used the code released by the authors of [19][4] to evaluate open-set performance on the same splits and we report these results in our paper.

We carried out tests on the following datasets using the protocol described in [16]:

CIFAR10 and SVHN. Both CIFAR10 [13] and SVHN [17] are 10-class classification datasets. CIFAR10 contains data from four vehicle classes and six animal classes. SVHN is a dataset of photographed numbers. In our tests we considered splits from [16] where six classes are chosen to be known. Remaining classes are considered to be open-set.

[3]Exact splits used by [16] can be found at github.com/lwneal/counterfactual-open-set.

[4]Code is found at github.com/otkupjnoz/c2ae. We validated results obtained for considered class splits with authors of [19].

CIFAR+10. CIFAR+10 training set consists vehicle classes of CIFAR10 dataset as known-classes. Vehicle classes from CIFAR10 and 10 vehicle classes samples from CIFAR100 [14] is considered to be open-set classes.
CIFAR+50. Same training setting as CIFAR10+. The vehicle classes from CIFAR10 and 50 vehicle classes samples from CIFAR100 are considered to be open-set classes.
TinyImageNet. TinyImageNet is a sub-set of 200 classes taken from the ImageNet dataset [4]. 20 classes are considered to be known and remaining 180 classes are considered to be open-set. Known-openset splits are chosen to be the same as in [16].

In Table 2, we tabulate open-set detection performance of known-classes for the proposed method with baseline methods. For each experiment, we indicated the *openness*[26], defined by $1 - \sqrt{\frac{K}{M}}$, where K and M denote the number of known classes and total number of classes, respectively. The performance of the baseline methods is obtained from [29] and [16]. According to Table 2, the proposed method has a significant improvement for the CIFAR10 dataset with an increase in performance of over 10%. A similar improvement is seen for the CIFAR+10 and CIFAR+50 test cases. Since CIFAR+50 dataset has more openness due to more open-set classes, it has produced slightly lower performance compared to CIFAR+10. For the SVHN dataset, the performance improvement is about 2%. For TinyImageNet, our performance is on par with other open-set methods where the proposed method performs marginally better. Table 3 lists the closed set classification accuracy for each dataset. In both Tables 2 and 3, we reported the performance of our method when WideResNet28-10 [31] classifier is used. It can be observed that using WideResNet, which is a better classifier, open-set recognition performance increases in majority of time. This result suggests that better performance can be obtained by using more sophisticated classifiers.

4.2. Out-of-distributional Detection

We evaluate the performance of the proposed method in Out-of-distributional detection (OOD) [12] on CIFAR10 dataset. Out-of-distributional detection is a special case of open-set detection. Here, it is assumed that the open-set samples follow a different distribution than the known-set distribution. Following the protocol outlined in [29], we considered all classes in CIFAR10 as known-classes and trained a 13-layer VGG model as specified in [29]. The output channels of each 3×3 convolutional block number were 64, 128, and 256, and they consist of two, two, and four convolutional layers with the same configuration. Then, we consider test images from ImageNet and LSUN dataset [30] as out-of-distributional images when each are cropped and resized respectively [15].

Table 4 shows the out-of-distributional performance in terms of macro-averaged F1 score. For the proposed method, following other OOD works [15], every sample producing a score lower than a 10%th percentile of matched scores were identified as open-set. It should be noted that it is customary to detect OOD samples based on SoftMax scores [12]. Therefore in Table 4 we reported F1 scores for the proposed method both when SoftMax scores and class activations are considered for decision making. All other numbers except ours are taken from [29]. According to Table 4, the proposed method out-performs baseline methods in all test cases. It should be noted that SoftMax scores yielded better OOD detection compared to class activation scores whenever images are cropped instead of resized. This is not surprising as an image crop contains little structure. As a result, image crops are more likely to produce balanced probabilities thereby making open-set detection based on SoftMax probabilities more effective.

4.3. Case Study and Ablation Study

We conducted a case-study on CIFAR10 dataset where all animal classes (bird, cat, deer, dog, frog and horse) were considered to be known. Vehicle classes (airplane, car, ship and truck) were considered to be open-set. We compare the performance of a conventional CNN network (Figure 2(a)) with the proposed method (Figure 2(c)). The conventional CNN produced a AUC of 84.35% where as the proposed method produced an AUC of 91.24%.

Figure 3 visualizes the score histograms generated for open-set samples and known-class samples for both methods. As evident from Figure 3, the proposed method has better score separation between open-set and known-set samples. This is why a larger AUC value has been obtained from the ROC curve for the proposed method.

To understand why a better score separation was obtained, we visualized the final feature space for both baseline CNN and the proposed method using tSNE [27] in Figure 4. In both cases, six clusters can be observed in the tSNE visualization plane in Figure 4; these clusters correspond to each class. However, there is a considerable over-lap between known-set samples and open-set samples in the baseline CNN (Figure 4(a)). On the other hand, under the proposed scheme (Figure 4(b)) overlap between known and open-set samples are less. Further we note that known clusters appearing under the proposed method is more compact compared to the baseline case. This is because proposed method models the whole data distribution (as a result of self-supervision and generative feature augmentation) as opposed to modeling just the boundary as usually done in conventional CNNs. The proposed method has a lower overlap between known and open-set samples in the feature space. Therefore, it produced better separation between known and open-set distributions as shown in Figure 3.

	CIFAR10 13.39%	CIFAR+10 33.33%	CIFAR+50 62.86%	SVHN 13.39%	TinyImageNet 57.35%
SoftMax	67.7± 3.8	81.6± *N.R.*	80.5± *N.R.*	88.6± 1.4	57.7± *N.R.*
OpenMax (CVPR16) [3]	69.5±4.4	81.7± *N.R.*	79.6± *N.R.*	89.4±1.3	57.6± *N.R.*
G-OpenMax (BMVC17) [7]	67.5±4.4	82.7±N.R.	81.9± *N.R.*	89.6±1.7	58.0± *N.R.*
OSRCI (ECCV18) [16]	*69.9±3.8*	*83.8*± *N.R.*	*82.7*± *N.R.*	*91.0±1.0*	58.6± *N.R.*
C2AE (CVPR19) [19]	71.1 ± 0.8	81.0 ± 0.5	80.3 ± 0.0	89.2 ± 1.3	58.1 ± 1.9
CROSR(CVPR19) [29]	*N.R.*	*N.R.*	*N.R.*	89.9±1.8	*58.9±N.R.*
Ours (Plain CNN)	**80.7±3.9**	**92.8±0.2**	**92.6±0.0**	**93.5±1.8**	**60.8±1.7**
Ours (WRN-28-10)	83.1±3.9	91.5±0.2	91.3±0.2	95.5±1.8	64.7±1.2

Table 2. Open-set detection performance in terms of AUC-ROC curve. N.R. is used when the original work did not report a particular figure.

	CIFAR10	CIFAR+10	CIFAR+50	SVHN	TinyImageNet
Ours (Plain CNN)	92.8±1.7	94.4±0.0	94.4±0.0	96.6±0.4	49.2±2.9
Ours (WRN-28-10)	95.09±1.3	97.4±0.2	97.4±0.2	97.29±1.3	55.9±2.8

Table 3. Closed-set accuracy for the proposed method.

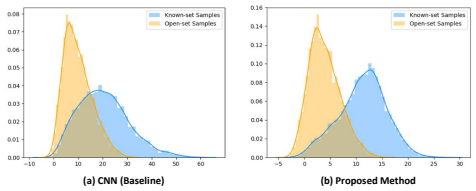

Figure 3. Score histograms for open-set and known-set samples. There is a better separation between the two distributions under the proposed method.

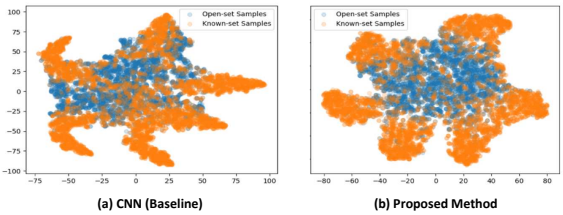

Figure 4. tSNE visualization of the feature space for (a)Conventional CNN and for the (b) proposed method. In both cases, known-class samples are clustered into six clusters - each representing each known class. However, variance of each cluster is large in the baseline CNN. There is considerable amount of over-lap between known and open-set samples. On the other, known-class clusters seems more compact under the proposed method. Overlap between known and open-set samples are lower compared to (a).

In Figure 5, we show eight open-set images that had produced the largest activations in the baseline CNN. It should be noted that although these images have generated high score activations, none of them have a close resemblance to any of the known-set of classes. In the same figure, we illustrate class activation scores obtained by the baseline CNN (middle column) and the proposed method (right column). Since the range of activation scores is different under the

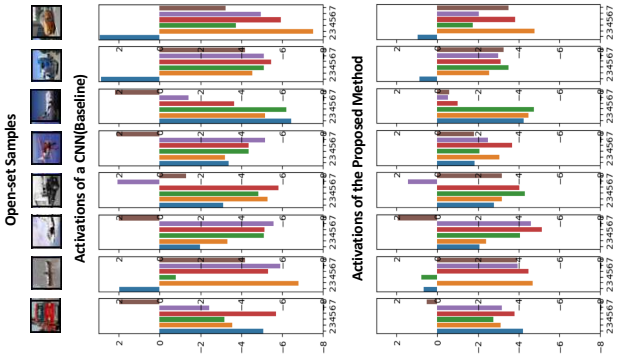

Figure 5. Top Row: Visualization of open-set samples that produced highest activations for the baseline CNN. Middle Row: z-score normalized activations produced by the baseline CNN for each image. Bottom Row: z-score normalized activations generated by proposed method. In six instances (out of eight), energy of produced activation has been reduced by the proposed method.

two methods, as Figure 3 shows, for a fair comparison we have normalized these scores using z-score normalization by considering all open-set scores under each scheme.

According to Figure 3 (Middle), the baseline CNN has produced a score around 2 for all samples. On the other hand, under the proposed scheme, the same images have generated lower scores. Except for the third and fourth images, activations produced by all other images have been reduced by at least by a factor of half. This example illustrates that the proposed method has even lowered activations for hard open-set samples.

Finally, it is worth noting the contribution each component of our proposal has towards the final outcome of the he algorithm. In order to assess this, we carried out an ablation study on CIFAR10 by considering animal classes as known-set classes. We considered following cases.

Baseline. The classifier network operating on only the input images as shown in Figure 2(a).

Training Method	Detector	ImageNet-Crop	ImageNet-Resize	LSUN-Crop	LSUN-Resize
Cross-entropy	SoftMax [12]	63.9	65.3	64.2	64.7
	OpenMax [3]	66.0	68.4	65.7	66.8
Counterfactual	SoftMax [16]	63.6	63.5	65.0	64.8
LadderNet	SoftMax [12]	64.0	64.6	64.4	64.7
	OpenMax [3]	65.3	67.0	65.2	65.9
	CROSR [29]	62.1	63.1	62.9	63.0
DHRNet	SoftMax [12]	64.5	64.9	65.0	64.9
	OpenMax [3]	65.5	67.5	65.6	66.4
	CROSR [29]	72.1	73.5	72.0	74.9
Ours	Activations	*75.7*	**79.2**	*75.1*	**80.5**
	SoftMax	**82.1**	*77.7*	**84.3**	*78.4*

Table 4. Performance of out-of-distributional object detection for CIFAR10 dataset with VGG13 network. Performance is measured using macro-F1 measure.

	Classification Accuracy	**Open-set Rejection(AUC)**
Baseline	89.7	84.4
Self-supervision	92.4	88.8
Augmented Classifier	91.5	88.4
Proposed Method	92.6	91.2

Table 5. Tabulation of classification performance (accuracy) and open-set rejection performance(AUC) for the ablation study.

Self-supervision. Classification network extended to perform self-supervision as shown in Figure 2(b).

Augmented Classifier. Generative feature is used to augment the input image space. A classifier is trained on the augmented input. No self-supervision is used.

Proposed method. Classifier is learned on augmented image space with self-supervision (Figure 2(c)).

In Table 5 we report closed-set classification accuracy along with open-set rejection performance in AUC-ROC. According to Table 5, the baseline produced a AUC-ROC value of 84.0%. The introduction of self-supervision and augmented features both independently improved open-set performance by 4%, where improvement induced by augmented features is marginally better than self-supervision. Finally, when both techniques are combined (the proposed method), performance further improves by 2.7% to arrive at 91.2%. This study demonstrates that each component of the proposal is contributing towards the final performance boost that is observed.

In Figure 6 we visualize reconstructions (of randomly chosen samples) obtained through the generative model. According to Figure 6, all reconstructed images take the form of a blurry version of the input images. However, we note that known-set samples carry more details compared to open-set classes. For an example, it is hard to predict the class label of open-set classes by merely looking at the reconstructed image. However, the amount of information preserved in the reconstructed image is not a very good indicator to detect open-set images (AUC is merely 66.7% when it is used as an indicator). Nevertheless, it provides information that can be leveraged to make a better informed decision.

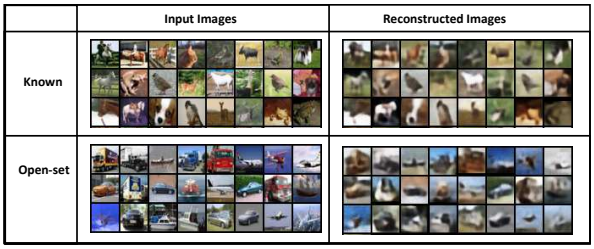

Figure 6. Reconstructed images produced by the auto encoder trained on known-set images. All reconstructed images are blurred versions of the input images. However, details are preserved better in known-set images. Note that predicting the class without the ground truth reference is hard for open-set images.

5. Conclusion

We explore the detection of open-set samples more effectively by learning richer feature representations than are usually needed for closed-set classification. We used self-supervision and augmented the input image with a representation obtained from a generative model to enhance network's ability to reject open-set samples. These improvements forced the classifier to look beyond what is required to perform closed-set classification when producing decision regions. We evaluated the proposed method in open-set detection and out-of-distributional image detection experiments where we produced state-of-the-art results.

We carried out a study investigating the importance of each component of the proposed method. Further, we demonstrated qualitatively how proposed method results in better separation in feature space thereby producing lower activations for open-set samples. Finally, we experimented with different choices of generative models and classifiers, where we concluded that using more sophisticated models in both cases would benefit open-set detection performance. In the future, we hope to investigate how this algorithm can be extended to other computer vision tasks such as object detection and semantic segmentation.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 4

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

[3] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 7, 8

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6

[5] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1422–1430, 2015. 2

[6] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[7] Zongyuan Ge, Sergey Demyanov, and Rahil Garnavi. Generative openmax for multi-class open set classification. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*, 2017. 2, 7

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018. 3

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. 3

[10] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9758–9769. Curran Associates, Inc., 2018. 3

[11] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[12] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. 1, 6, 8

[13] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 5

[14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). 6

[15] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *International Conference on Learning Representations (ICLR)*, 2018. 6

[16] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 4, 5, 6, 7, 8

[17] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 2011. 5

[18] P. Oza and V. M. Patel. Active authentication using an autoencoder regularized cnn-based one-class classifier. In *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pages 1–8, 2019.

[19] Poojan Oza and Vishal M. Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5, 7

[20] Poojan Oza and Vishal M Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2019. 2

[21] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[22] Pramuditha Perera and Vishal M. Patel. Deep transfer learning for multiple class novelty detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[23] Pramuditha Perera and Vishal M. Patel. Face-based multiple user active authentication on mobile devices. *IEEE Transactions on Information Forensics and Security*, 14(5):1240–1250, 2019. 1

[24] Pramuditha Perera and Vishal M. Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019. 2

[25] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3546–3554. Curran Associates, Inc., 2015. 2

[26] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 35, July 2013. 2, 6

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 6

[28] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 4

[29] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5, 6, 7, 8

[30] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[31] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 4, 6

[32] H. Zhang and V. M. Patel. Sparse representation-based open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1690–1696, 2017.