Classification-Reconstruction Learning for Open-Set Recognition

Ryota Yoshihashi[1] Wen Shao[1] Rei Kawakami[1]
Shaodi You[2] Makoto Iida[1] Takeshi Naemura[1]
[1]The University of Tokyo [2]Data61-CSIRO

{yoshi,shao,rei,naemura}@nae-lab.org, Shaodi.You@data61.csiro.au, iida@ilab.eco.rcast.u-tokyo.ac.jp

Abstract

Open-set classification is a problem of handling 'unknown' classes that are not contained in the training dataset, whereas traditional classifiers assume that only known classes appear in the test environment. Existing open-set classifiers rely on deep networks trained in a supervised manner on known classes in the training set; this causes specialization of learned representations to known classes and makes it hard to distinguish unknowns from knowns. In contrast, we train networks for joint classification and reconstruction of input data. This enhances the learned representation so as to preserve information useful for separating unknowns from knowns, as well as to discriminate classes of knowns. Our novel Classification-Reconstruction learning for Open-Set Recognition (CROSR) utilizes latent representations for reconstruction and enables robust unknown detection without harming the known-class classification accuracy. Extensive experiments reveal that the proposed method outperforms existing deep open-set classifiers in multiple standard datasets and is robust to diverse outliers. The code is available in https://nae-lab.org/~rei/research/crosr/.

1. Introduction

To be deployable to real applications, recognition systems need to be tolerant of unknown things and events that were not anticipated during the training phase. However, most of the existing learning methods are based on the closed-world assumption, that is, the training datasets are assumed to include all classes that appear in the environments where the system will be deployed. This assumption can be easily violated in real-world problems, where covering all possible classes is almost impossible [26]. Closed-set classifiers are error-prone to samples of unknown classes, and this limits their usability [47, 44].

In contrast, open-set classifiers [37] can detect samples that belong to none of the training classes. Typically, they fit a probability distribution to the training samples in some feature space, and detect outliers as unknowns. For the features to represent the samples, almost all existing deep open-set classifiers rely on those acquired via fully supervised learning [3, 10, 41], as shown in Fig. 1 (a). However, they are for emphasizing the discriminative features of known classes; they are not necessarily useful for representing unknowns or separating unknowns from knowns.

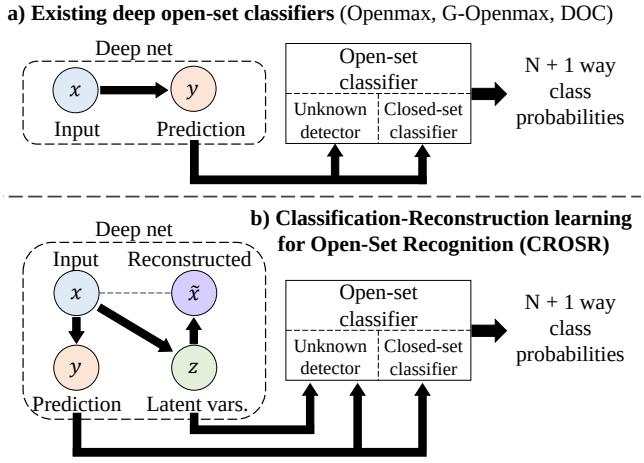


Figure 1. Overview of existing and our deep open-set classification models. Existing models (a) utilize only their network's final prediction $\boldsymbol{y}$ for classification and unknown detection. In contrast, in CROSR (b), a deep net is trained to provide a prediction $\boldsymbol{y}$ and a latent representation for reconstruction $\boldsymbol{z}$ within known classes. An open-set classifier (right), which consists of an unknown detector and a closed-set classifier, exploits $\boldsymbol{y}$ for closed-set classification, and $\boldsymbol{y}$ and $\boldsymbol{z}$ for unknown detection.

In this study, our goal is to learn efficient feature representations that are able to classify known classes as well as to detect unknowns as outliers. Regarding the representations of outliers that we cannot assume beforehand, it is natural to add unsupervised learning as a regularizer so that the learned representations acquire information that are important in general but may not be useful for classifying given classes. Thus, we utilize unsupervised learning of reconstructions in addition to supervised learning of classifications. Reconstruction of input samples from

low-dimensional latent representations inside the networks is a general way of unsupervised learning [16]. The representation learned via reconstruction are useful in several tasks [51]. Although there are previous successful examples of classification-reconstruction learning, such as semi-supervised learning [32] and domain adaptation [11], this study is the first to apply deep classification-reconstruction learning to open-set classification.

Here, we present a novel open-set classification framework, called Classification-Reconstruction learning for Open-Set Recognition (CROSR). As shown in Fig. 1 (b), the open-set classifier consists of two parts: a closed-set classifier and an unknown detector, both of which exploit a deep classification-reconstruction network.[1] While the known-class classifier exploits supervisedly learned prediction $\boldsymbol{y}$, the unknown detector uses a reconstructive latent representation $\boldsymbol{z}$ together with $\boldsymbol{y}$. This allows unknown detectors to exploit a wider pool of features that may not be discriminative for known classes. Additionally, in higher-level layers of supervised deep nets, details of input tend to be lost [51, 6] , which may not be preferable in unknown detection. CROSR can exploit reconstructive representation $\boldsymbol{z}$ to complement the lost information in the prediction $\boldsymbol{y}$.

To provide effective $\boldsymbol{y}$ and $\boldsymbol{z}$ simultaneously, we further design *deep hierarchical reconstruction nets (DHRNets)*. The key idea in DHRNets is the bottlenecked lateral connections, which is useful to learn rich representations for classification and compact representations for detection of unknowns jointly. DHRNets learn reconstruction of each intermediate layer in classification networks using latent representations, i.e., mapping to low-dimensional spaces, and as a result it acquires hierarchical latent representation. With the hierarchical bottlenecked representation in DHRNets, the unknown detector in CROSR can exploit multi-level anomaly factors easily thanks to the representations compactness. This bottlenecking is crucial, because outliers are harder to detect in higher dimensional feature spaces due to *concentration on the sphere* [53]. Existing autoencoder variants, which are useful for outlier detection by learning compact representations [52, 1], cannot afford large-scale classification because the bottlenecks in their mainstreams limit the expressive power for classification. CROSR with a DHRNet becomes more robust to a wide variety of unknown samples, some of which are very similar to the known-class samples. Our experiments in five standard datasets show that representations learned via reconstruction serve to complement those obtained via classification.

Our contribution is three-fold: First, we discuss the usefulness of deep reconstruction-based representation learning in open-set recognition for the first time; all of the other deep open-set classifiers are based on discriminative representation learning in known classes. Second, we develop a novel open-set recognition framework, CROSR, which is based on DHRNets and jointly performs known classification and unknown detection using them. Third, we conducted experiments on open-set classification in five standard image and text datasets, and the results show that our method outperforms existing deep open-set classifiers for most combinations of known data and outliers. The code related to this paper is available in https://nae-lab.org/~rei/research/crosr/.

[1] We refer to detection of unknowns as *unknown detection*, and known-class classification as *known classification*.

2. Related work

Open-set classification Compared with closed-set classification, which has been investigated for decades [8, 5, 9], open-set classification has been surprisingly overlooked. The few studies on this topic mostly utilized either linear, kernel, or nearest-neighbor models. For example, Weibull-calibrated SVM [38] considers a distribution of decision scores for unknown detection. Center-based similarity space models [7] represent data by their similarity to class centroids in order to tighten the distributions of positive data. Extreme value machines [35] model class-inclusion probabilities using an extreme-value-theory-based density function. Open-set nearest neighbor methods [18] utilizes the distance ratio to the nearest and second nearest classes. Among them, sparse-representation-based open-set recognition [49] shares the idea of reconstruction-based representation learning with ours. The difference is in that we consider deep representation learning, while [49] uses a single-layer linear representation. These models cannot be applied to large-scale raw data without feature engineering.

The origin of deep open-set classifiers was in 2016 [3], and few deep open-set classifiers have been reported since then. G-Openmax [10], a direct extension of Openmax, trains networks with synthesized *unknown* data by using generative models. However, it cannot be applied to natural images other than hand-written characters due to the difficulty of generative modeling. DOC (deep open classifier) [41, 42], which is designed for document classification, enables end-to-end training by eliminating outlier detectors outside networks and using sigmoid activations in the networks for performing joint classification and outlier detection. Its drawback is that the sigmoids do not have the *compact abating property* [38]; namely, they may be activated by an infinitely distant input from all of the training data, and thus its open space risk is not bounded.

Outlier detection Outlier (also called anomaly or novelty) detection can be incorporated in the concept of open-set-classification as an unknown detector. However, outlier detectors are not open-set classifiers by themselves because they have no discriminative power within known classes. Some of the generic methods for anomaly detection are one-class extension of discriminative models such as SVM [25]

or forests [21], generative models such as Gaussian mixture models [34], and subspace methods [33]. However, most of the recent anomaly-detection literature focuses on incorporating domain knowledge specific to the task at hand, such as cues from videos [48, 15], and they cannot be used to build a generic-purpose open-set classifiers.

Deep nets have also been examined for outlier detection. The deep approaches mainly use autoencoders trained in an unsupervised manner [52], in combination with GMM [54], clustering [1], or one-class learning [30]. Generative adversarial nets [12] can be used for outlier detection [40] by using their reconstruction errors and discriminators' decisions. This usage is different from ours that utilizes latent representations. However, in outlier detection, deep nets are not always the absolute winners unlike in supervised learning, because nets need to be trained in an unsupervised manner and are less effective because of that.

Some studies use networks trained in a supervised manner to detect anomalies that are not from the distributions of training data [14, 20]. However, their methods cannot be simply extended to open-set classifiers because they use input preprocessing, for example, adversarial perturbation [13], and this operation may degrade known-class classification.

Semi-supervised learning In semi-supervised learning settings including domain adaptation, reconstruction is useful as a data-dependent regularizer [32, 23]. Among them, ladder nets [32] are partly similar to ours in terms of using lateral connections, except that ladder nets do not have the bottleneck structure. Our work aims at demonstrating that the reconstructive regularizers are also useful in open-set classification. However, the usage of the regularizers is largely different; CROSR uses them to prevent the representations from overly specializing to known classes, while semi-supervised learners use them to incorporate unlabeled data in their training objectives. Furthermore, in semi-supervised learning settings reconstruction errors are computed on *unlabeled* data as well as labeled training data. In open-set settings, it is impossible to compute reconstruction errors on any *unknown* data; we only use labeled (known) training data.

3. Preliminaries

Before introducing CROSR, we briefly review Openmax [3], the existing deep open-set classifier. We also introduce the terminology and notation.

Openmax is an extension of Softmax. Given a set of known classes $\mathcal{K} = \{C_1, C_2, ..., C_N\}$ and an input data point $\boldsymbol{x}$, Softmax is defined as following:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{f}(\boldsymbol{x}), \qquad (1) \\ p(C_i|\boldsymbol{x}, \boldsymbol{x} \in \mathcal{K}) &= \text{Softmax}_i(\boldsymbol{y}) = \frac{\exp(x_i)}{\sum_j^N \exp(x_j)}, \end{aligned}$$

where $\boldsymbol{f}$ denotes the network as a function and $\boldsymbol{y}$ denotes the representation of its final hidden layer, whose dimensionality is equal to the number of the known classes. To be consistent with [3], we refer to it as the activation vector (AV). Softmax is designed for closed-set settings where $\boldsymbol{x} \in \mathcal{K}$, and in open-set settings, we need to consider $\boldsymbol{x} \notin \mathcal{K}$. This is achieved by calibrating the AV by the inclusion probabilities of each class:

$$\begin{aligned} \text{Openmax}_i(\boldsymbol{x}) &= \text{Softmax}_i(\hat{\boldsymbol{y}}), \qquad (2) \\ \hat{\boldsymbol{y}}_i &= \begin{cases} \boldsymbol{y}_i \boldsymbol{w}_i & (i \leq N) \\ \sum_{i=1}^N \boldsymbol{y}_i (1 - \boldsymbol{w}_i) & (i = N+1), \end{cases} \end{aligned}$$

where w_i represents the belief that $\boldsymbol{x}$ belongs to the known class C_i. Here, $\hat{\boldsymbol{y}}$, the *calibrated activation vector* prevents Openmax from giving high confidences to outliers that give small $\boldsymbol{w}$, i.e., the unknown samples that do not belong to C_i. Formally, the class C_{N+1} represents the *unknown* class. Usage of $p(\boldsymbol{x} \in C_i)$ can be understood as a proxy for $p(\boldsymbol{x} \in \mathcal{K})$, which is harder to model due to inter-class variances.

For modeling class-belongingness $p(\boldsymbol{x} \in \mathcal{K})$, we need a distance function $d(\cdot, \cdot)$ and its distribution. The distance measures the affinity of a data point to each class. Statistical extreme-value theory suggests that the Weibull family of distributions is suitable [35] for this purpose. Assuming that d of the inliers follows a Weibull distribution, class-belongingness can be expressed using the cumulative density function,

$$\begin{aligned} p(\boldsymbol{x} \in C_i) &= 1 - R_\alpha(i) \cdot \text{WeibullCDF}(d(\boldsymbol{x}, C_i); \rho_i) \\ &= 1 - R_\alpha(i) \exp\left(- \left(\frac{d(\boldsymbol{x}, C_i)}{\eta_i} \right)^{m_i} \right). \qquad (3) \end{aligned}$$

Here, $\rho_i = (m_i, \eta_i)$ are parameters of the distribution that are derived from the training data of the class C_i. $R_\alpha(i) = \max\left(0, \frac{\alpha - \text{rank}(i)}{\alpha}\right)$ is a heuristic calibrator that makes a larger discount in more confident classes, and is defined by a hyperparameter α. $\text{rank}(i)$ is the index in the AV sorted in descending order.

As a class-belongingness measure, we used the ℓ^2 distance of AVs from the class means, similarly to nearest non-outlier classification [2]:

$$d(\boldsymbol{x}, C_i) = |\boldsymbol{y} - \boldsymbol{\mu}_i|_2 \,. \qquad (4)$$

This gives a strong simplification assuming that $p(\mathbf{x} \in C_i)$ depends only on the $\boldsymbol{y}$.

4. CROSR: Classification-reconstruction learning for open-set recognition

Our design of *CROSR* is based on observations about Openmax's formulation: AVs are not necessarily the best representations for modeling the class-belongingness

$p(\boldsymbol{x} \in C_i)$. Although AVs in supervised networks are optimized to give correct $p(C_i|\boldsymbol{x})$, they are not encouraged to encode information about $\boldsymbol{x}$, and it is not sufficient to test whether $\boldsymbol{x}$ itself is probable in C_i. We alleviate this problem by exploiting reconstructive latent representations, which encode more about $\boldsymbol{x}$.

4.1. Open-set classification with latent representations

To enable the use of latent representations for reconstruction in the unknown detector, we extend the Openmax classifier (Eqns. 1 – 4) as follows. We replace Eqn. 1 for applying the main-body network $\boldsymbol{f}$ to both known classification and reconstruction:

$$\begin{aligned} (\boldsymbol{y}, \boldsymbol{z}) &= \boldsymbol{f}(\boldsymbol{x}), \\ p(C_i|\boldsymbol{x}, \boldsymbol{x} \in \mathcal{K}) &= \mathrm{Softmax}_i(\boldsymbol{y}), \\ \tilde{\boldsymbol{x}} &= \boldsymbol{g}(\boldsymbol{z}). \end{aligned} \tag{5}$$

Here we have introduced $\boldsymbol{g}$, a decoder network only used in training to make the latent representation $\boldsymbol{z}$ meaningful via reconstruction. $\tilde{\boldsymbol{x}}$ is the reconstruction of $\boldsymbol{x}$ using $\boldsymbol{z}$. These equations correspond to the left part of Fig. 1 (b).

The network's prediction $\boldsymbol{y}$ and latent representation $\boldsymbol{z}$ are jointly used in the class-belongingness modeling. Instead of Eqn. 4, CROSR considers the joint distributions of $\boldsymbol{y}$ and $\boldsymbol{z}$ to be a hypersphere per class:

$$d(\boldsymbol{x}, C_i) = |[\boldsymbol{y}, \boldsymbol{z}] - \boldsymbol{\mu}_i|_2 \,. \tag{6}$$

Here, $[\boldsymbol{y}, \boldsymbol{z}]$ denotes concatenation of the vectors of $\boldsymbol{y}$ and $\boldsymbol{z}$, and $\boldsymbol{\mu}_i$ denotes their mean within class C_i.

4.2. Deep Hierarchical Reconstruction Nets

After designing the open-set classification framework, we must specify the function form, i.e., the network architecture for $\boldsymbol{f}$. The network used in CROSR needs to effectively provide a prediction $\boldsymbol{y}$ and latent representation $\boldsymbol{z}$. Our design of deep hierarchical reconstruction nets (DHRNets) simultaneously maintains the accuracy of $\boldsymbol{y}$ in known classification and provides a compact $\boldsymbol{z}$.

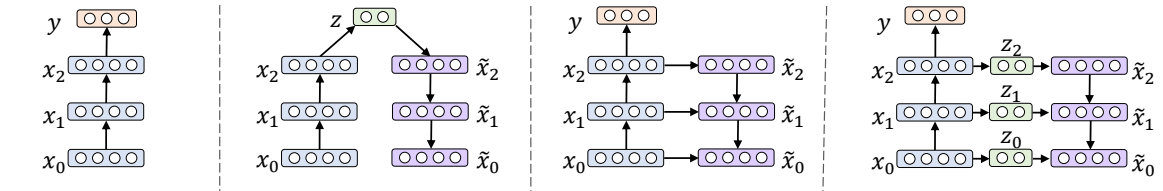
For a conceptual explanation, DHRNet extracts the latent representations from each stage of middle-level layers in the classification network. Specifically, it extracts a series of latent representations $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, ..., \boldsymbol{z}_L$ from multi-stage features $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, ..., \boldsymbol{x}_L$. We refer to these latent representations as *bottlenecks*. The advantage of this architecture is that it can detect outlying factors that are hidden in the input data but vanish in the middle of the inference chains. Since we cannot presume a stage where the outlying factors are most obvious, we construct the input vector for the unknown detector $\boldsymbol{z}$ by simply concatenating $\boldsymbol{z}_l$ from the layers. Here, $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, ..., \boldsymbol{z}_L$ can be interpreted as decomposed factors to generate $\boldsymbol{x}$. To draw an analogy, unknown detection using decomposed latent representations is similar to overhauling [27] mechanical products, where one disassembles $\boldsymbol{x}$ into parts $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, ..., \boldsymbol{z}_L$, investigates the parts for anomalies, and reassembles them into $\tilde{\boldsymbol{x}}$.

Figure 2 compares the existing architectures and DHRNet. Most of the closed-set classifiers and Openmax rely on supervised classification-only models (a) that do not have useful factors for outlier detection other than $\boldsymbol{y}$, because $\boldsymbol{x}_l$ usually has high dimensionality for known-class classification. Employing autoencoders (b) is a straightforward way to introduce latent representations for reconstruction, but there is a problem in using them for open-set classification. Deep autoencoders gradually reduce the dimensionality of the intermediate layers $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, ...$, for effective information compression. This is not good for large-scale closed-set classification, which needs a fairly large number of neurons in all layers to learn a rich feature hierarchy. LadderNet (c) can be regarded as a variant of an autoencoder, because it performs reconstruction. However, the difference lies in the *lateral connections*, through which part of $\boldsymbol{x}_l$ flows to the reconstruction stream without further compression. Their role is in a detail-abstract decomposition [46]; that is, LadderNet encodes abstract information in the main stream and details in the lateral paths. While this is preferable for open-set classification because the outlying factors of unknowns may be in the details as well as in the abstracts, LadderNet itself does not provide compact latent variables.DHRNet (d) further enhances the decomposed information's effectiveness for unknown detection by compressing the lateral streams in compact representations $\boldsymbol{z_1}, \boldsymbol{z_2}, ..., \boldsymbol{z_L}$.

In detail, the l-th layer of DHRNet is expressed as

$$\begin{aligned} \boldsymbol{x}_{l+1} &= \boldsymbol{f}_l(\boldsymbol{x}_l), \\ \boldsymbol{z}_l &= \boldsymbol{h}_l(\boldsymbol{x}_l), \\ \tilde{\boldsymbol{x}}_l &= \boldsymbol{g}_l(\tilde{\boldsymbol{x}}_{l+1} + \tilde{\boldsymbol{h}}_l(\boldsymbol{z}_l)). \end{aligned} \tag{7}$$

Here, $\boldsymbol{f}_l$ denotes a block of a feature transformation in the network, i.e., a series of convolutional layers between downsampling layers in a plain CNN or a densely-connected block in DenseNet [17]. $\boldsymbol{h}_l$ denotes an operation of non-linear dimensionality reduction, which consists of a ReLU and a convolution layer, while $\tilde{\boldsymbol{h}}_l$ means a reprojection to the original dimensionality of $\boldsymbol{x}_l$. The pair of $\boldsymbol{h}_l$ and $\tilde{\boldsymbol{h}}_l$ is similar to an autoencoder. $\boldsymbol{g}_l$ is a combinator of the top-down information $\tilde{\boldsymbol{x}}_{l+1}$ and lateral information $\tilde{\boldsymbol{h}}_l(\boldsymbol{z}_l)$. While the function forms for $\boldsymbol{g}_l$ are investigated by [31], we choose to use an element-wise sum and subsequent convolutional and ReLU layers as the simplest form among the possible variants. When inputting $\boldsymbol{z}_l$ to the unknown detectors, the spatial axes are reduced by global max pooling to form a one-dimensional vector. This performs slightly better than vectorization by using average pooling or flattening. Figure 3 illustrates these operations, and the stack of operations gives the overall network shown in Fig. 2 (d).



(a) Supervised net (b) Autoencoder (c) LadderNet (d) Ours: Deep hierarchical reconstruction net

Figure 2. Conceptual illustrations of (a–c) existing models and (d) our model.

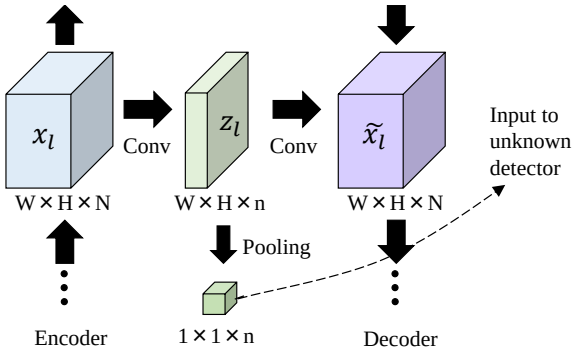


Figure 3. Implementation of the deep hierarchical reconstruction net with convolutional layers.

Training We minimize the sum of classification errors and reconstruction errors in training data from known classes. To measure the classification error, we use softmax cross entropy of $\boldsymbol{y}$ and the ground-truth labels. To measure the reconstruction error of $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$, we use the ℓ^2 distance in the images and the cross entropy of one-hot word representations in the texts. Note that we cannot use the data of the *unknown* classes in training and the reconstruction loss is computed only with known samples. The whole network is differentiable and trainable using gradient-based methods. After the network is trained and its weights fixed, we compute Weibull distributions for unknown detection.

Implementation There are some more minor differences between our implementation and the ladder nets in [32]. First, we use dropout in intermediate layers instead of noise addition, because it results in slightly better closed-set accuracy. Second, we do not penalize reconstruction errors of intermediate layers. This enables us to avoid the separate computation of 'noisy' and 'clean' layers that was originally needed for intermediate-layer reconstruction. We simply refer to our network without bottlenecks; in other words where $\boldsymbol{h}_l$ and $\boldsymbol{h}'_l$ are identity transformations, as LadderNet. For the experiments, we implement LadderNet and DHRNet with various backbone architectures.

5. Experiments

We experimented with CROSR and other methods on five standard datasets: MNIST, CIFAR-10, SVHN, Tiny-ImageNet, and DBpedia. These datasets are for closed-set classification, and we extended them in two ways: 1) class separation and 2) outlier addition. In class-separation setting, we selected some classes randomly in order to use them as knowns. We used the remainder as unknowns. In this setting, which has been used in the open-set literature [41, 28], unknown samples come from the same domain as that of knowns. Outlier addition is a protocol introduced for out-of-distribution detection [14]; the networks are trained on the full training data, but in the test phase, outliers from another dataset are added to the test set as *unknowns*. The merit of doing so is that we can test the robustness of the classifiers against a larger diversity of data than in the original datasets. The class labels of the *unknowns* were not used in any case and they all were treated as a single *unknown* class.

Table 1. Closed-set test accuracy of used networks. Despite adding reconstruction terms to the training objectives for LadderNet and DHRNet, there was no significant degradation in accuracy in known classification.

		MNIST	C-10	SVHN
Plain CNN	Supervised only	0.991	0.934	0.943
	LadderNet	0.993	0.928	–
	DHRNet (ours)	0.992	0.930	0.945
DenseNet	Supervised only	–	0.944	–
	DHRNet (ours)	–	0.940	–

MNIST MNIST is the most popular hand-written digit benchmark. It has 60,000 images for training and 10,000 for testing from ten classes. Although near-100% accuracy has been achieved in closed-set classification [4], the open-set extension of MNIST remains a challenge due to the variety of possible outliers.

As outliers, we used datasets of small gray-scale images, namely Omniglot, Noise, and MNIST-Noise. Omniglot is a dataset of hand-written characters from the alphabets of various languages. We only used the test set because the outliers are only needed in the test phase. 'Noise' is a set of images we synthesized by sampling each pixel value independently from a uniform distribution on [0, 1]. MNIST-Noise is also a synthesized set, made by superimposing MNIST's test images on Noise, and thus its images are more similar to the inliers. Figure 4 shows their samples. Each dataset has 10,000 test images, the same as MNIST, and this makes the known-to-unknown ratio 1:1.

We used a seven-layer plain CNN for MNIST. It consists

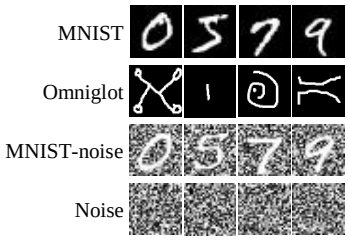


Figure 4. Sample images from MNIST and outlier sets.

Table 2. Open-set classification results in MNIST with various outliers added to the test set as *unknowns*. We report macro-averaged F1-scores in eleven classes (*0–9* and *unknown*). A larger score is better.

Backbone network	Training method	UNK detector	Omniglot	MNIST-noise	Noise
Plain CNN	Supervised only	Softmax	0.592	0.641	0.826
		Openmax	0.680	0.720	**0.890**
	LadderNet	Softmax	0.588	0.772	0.828
		Openmax	0.764	0.821	0.826
	DHRNet (ours)	Softmax	0.595	0.801	0.829
		Openmax	0.780	0.816	0.826
		CROSR (ours)	**0.793**	**0.827**	0.826

Table 3. Open-set classification results in CIFAR-10. A larger score is better.

Backbone network	Training method	UNK detector	ImageNet-crop	ImageNet-resize	LSUN-crop	LSUN-resize
Plain CNN	Counterfactual [28]		0.636	0.635	0.650	0.648
Plain CNN	Supervised only	Softmax	0.639	0.653	0.642	0.647
		Openmax	0.660	0.684	0.657	0.668
	LadderNet	Softmax	0.640	0.646	0.644	0.647
		Openmax	0.653	0.670	0.652	0.659
		CROSR	0.621	0.631	0.629	0.630
	DHRNet (ours)	Softmax	0.645	0.649	0.650	0.649
		Openmax	0.655	0.675	0.656	0.664
		CROSR (ours)	**0.721**	**0.735**	**0.720**	**0.749**
DenseNet	Supervised only	Softmax	0.693	0.685	0.697	0.722
		Openmax	0.696	0.688	0.700	0.726
	DHRNet (ours)	Softmax	0.691	0.726	0.688	0.700
		Openmax	0.729	0.760	0.712	0.728
		CROSR (ours)	**0.733**	**0.763**	**0.714**	**0.731**

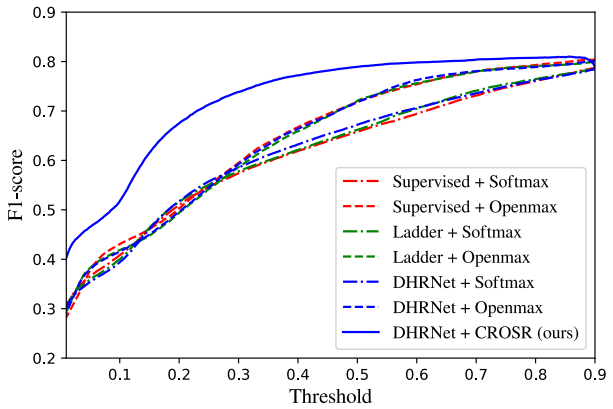


Figure 5. Relationship between the rejection threshold and F1-score. These plots are from test results for CIFAR-10 and ImageNet-crop using VGGNets.

Table 4. Open-set text classification results for DBpedia. F1-scores are shown for various train/test class ratios.

Method	4/14	4/12	4/8	4/4
DOC	0.507	0.568	0.733	0.985
Softmax	0.460	0.503	0.662	**0.988**
Openmax	0.532	0.574	0.729	0.986
CROSR (ours)	**0.582**	**0.627**	**0.765**	0.987

of five convolutional layers with 3×3 kernels and 100 output channels, followed by ReLU non-linearities. Max pooling layers with a stride of 2 are inserted after every two convolutional layers. At the end of the convolutional layers, we put two fully connected layers with 500 and 10 units, and the last one was directly exposed to the Softmax classifier. In DHRNet, lateral connections are put after every pooling layer. The dimensionalities of the latent representations $\mathbf{z}_l$ were all fixed to 32.

CIFAR-10 CIFAR-10 has 50,000 natural images for training and 10,000 for testing. It consists of ten classes, containing 5,000 training images for each class. In CIFAR-10, each class has large intra-class diversities by color, style, or pose difference, and state-of-the-art deep nets make a fair number of classification errors within known classes.

We examined two types of network, a plain CNN and DenseNet [17], a state-of-the-art network for closed-set image classification. The plain CNN is a VGGNet [43]-style network re-designed for CIFAR, and it has 13 layers. The layers are grouped into three convolutional and one fully connected block. The output channels of each convolutional block number 64, 128, and 256, and they consist of two, two, and four convolutional layers with the same configuration. All convolutional kernels are 3×3. We set the depth of DenseNet to 92 and the growth rate to 24. The dimensionalities of the latent representations $\mathbf{z}_l$ were all fixed to 32, the same as in MNIST.

We used the outliers collected by [20] from other datasets, i.e., ImageNet and LSUN, and we resized or cropped them so that they would have the same sizes [2] Among the outlier sets used in [14], we did not use synthesized sets of Gaussian and Uniform because they can be

[2]URL: https://github.com/facebookresearch/odin.

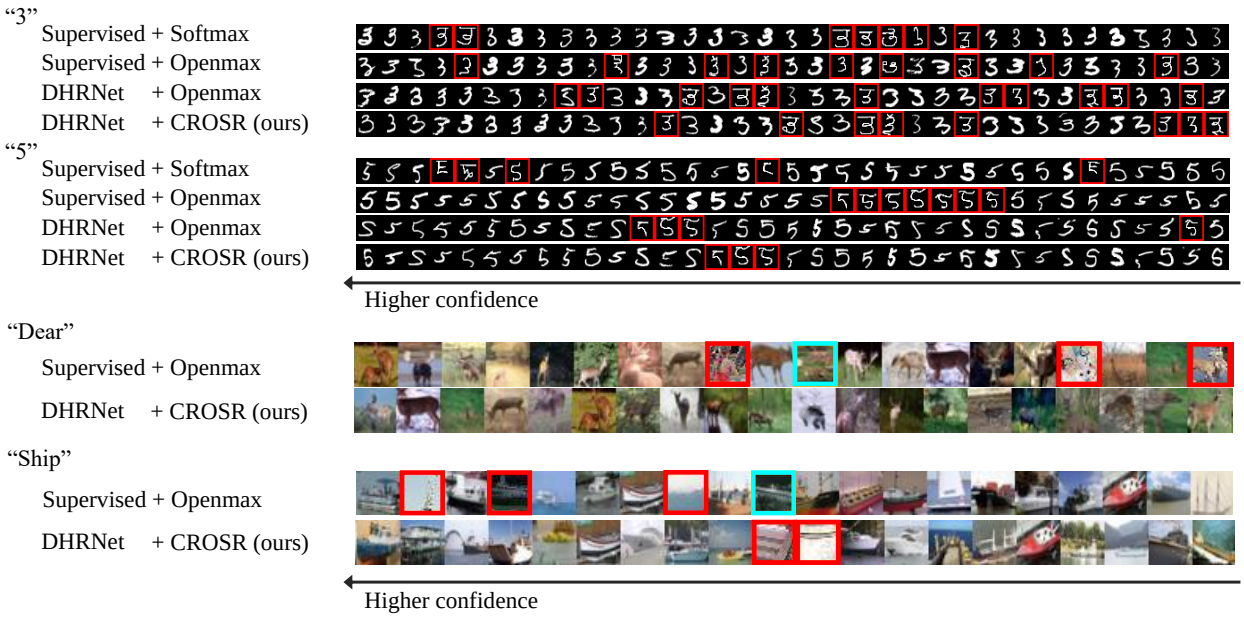


Figure 6. Visualized samples. Sampled data points are sorted by each methods' confidence score, and the top samples are listed. The red boxes show unknown samples, and the cyan ones show misclassification in known classes. Fewer unknowns to the left indicate higher robustness.

easily detected by baseline outlier-removal techniques. The datasets each have 10,000 test images, which is the same as in MNIST and this makes the known-to-unknown ratio 1:1.

SVHN and TinyImageNet SVHN is a dataset of 10-class digit photographs, and TinyImageNet is a 200-class subset of ImageNet. In these datasets, we compare CROSR with recent GAN-based methods [10, 28] that utilize *unknown* training data synthesized by GANs. A concern in the comparisons was the instability of the training and resulting variance in the quality of the training data generated by the GAN-based mechanisms, which may make comparisons hard [22]. Thus, we exactly followed the evaluation protocols used in [28] (class separation within each single dataset, averaging over five trials, area-under-the-curve criteria), and directly compared our results against the reported numbers. Our backbone network was the same as the one used in [28] that consists of nine convolutional layers and one fully connected layers, except that ours had decoding parts as shown in Eqn. 7.

DBpedia The DBpedia ontology classification dataset contains 14 classes of Wikipedia articles, 40,000 instances for training and 5,000 for testing. We selected this dataset because it has the largest number of classes among the often-used datasets in the literature of the convnet-based large-scale text classification [50] and for ease in making various class splits. We conducted the open-set evaluation with class separation using 4 random classes as knowns and 4, 8, and 10 as unknowns.

In DBpedia, we implemented DHRNet on the basis of a shallow-and-wide convnet [19], which had three convolutional layers with kernels whose sizes were 3, 4, and 5, and whose output dimension was 100. Text-classification convnets are extendable to DHRNet by setting $W =$ (maximum text length) and $H = 1$ in Fig. 3. The dimensionality of its bottleneck was 25. We also implemented DOC [41] using the same architecture as ours for a fair comparison.

Training DHRNet We confirmed that DHRNet can be trained by using the joint classification-reconstruction loss. We used the SGD solver with learning-rate scheduling tuned in each dataset. We set the weights of the reconstruction loss and the classification loss to the same value 1.0. In principle, the weight of reconstruction error should be as large as possible while keeping the close-set validation accuracy, which would give the most regularized and well-fitted model. However, we obtained satisfactory results with the default value and did not tune them further. The closed-set test errors of the networks for each dataset are listed in Table 1. All of the networks were trained without any large degradation in closed-set accuracy from the original ones. This and the subsequent experiments were conducted using Chainer [45].

Weibull distribution fitting We used `libmr` library [39] to compute the parameters in Weibull distribution. It has the hyperparameters α from Eqn. 3 and `tail_size`, the number of extrema used to define the tails of the distributions. We used the values suggested in [3], namely $\alpha = 10$ and `tail_size` $= 20$. For MNIST and CIFAR-10, we did not use the rank calibration with α in Eqn. 3, since it does not improve the performance due to the small num-

ber of classes. For DenseNet in CIFAR-10, we noticed that Openmax performed worse with the default parameters, so we changed `tail_size` to 50. Since heavily tuning these hyperparameters for specific types of outlier runs counter to the motivation of open-set recognition for handling *unknowns*, we did not tune them for each of the test sets.

Results We show the results for MNIST in Table 2, for CIFAR-10 in Table 3, and for DBpedia in Table 4. The reported values are F1-scores [36] of known classes and *unknown* as a class with a threshold 0.5. CROSR outperformed all of the other methods consistently except in two settings. Specifically, in MNIST, CROSR outperformed Supervised + Openmax by more than 10% in F1-score when using Omniglot or MNIST-noise as outliers, whereas it slightly underperformed with Noise, the easiest outliers. CROSR also performed better than or as well as the stronger baselines LadderNet + Openmax and DHRNet + Openmax. In CIFAR-10, the results for varying thresholds are also shown in Fig. 5, in which it is clear that CROSR outperformed the other methods regardless of the threshold.

Interestingly, LadderNet with Openmax outperformed the supervised-only networks. For instance, LadderNet-Openmax achieved an 8.4% gain in F1-score in the MNIST-vs-Omniglot setting and a 10.1% gain in the MNIST-vs-MNIST-Noise setting. This means regularization using the reconstruction loss is beneficial for unknown detection; in other words, using supervised losses in known classes is not the best for training open-set deep networks. However, no gains were had by adding only the reconstruction-error term to training objectives in the natural image datasets. This means we need to use the reconstructive factors in the networks in a more explicit form by adopting DHRNet.

For DBpedia, CROSR outperformed the other methods, except when the number of train/test classes was 4/4, which is equivalent to the closed-set settings. While DOC and Openmax performed almost on a par with each other, the improvement of CROSR over Openmax was also significant in this dataset.

Comparison with GAN-based methods Table 5 summarizes the results of ours and the GAN-based methods. Ours outperformed all of the other methods in MNIST and TinyImageNet, and all except Counterfactual in SVHN. While the relative improvements are within the ranges of the error bars, these results still means that our method, which does not use any synthesized training data, can perform on par or slightly better than the state-of-the-art GAN-based methods.

In combination with anomaly detectors To investigate how latent representations can be exploited more effectively, we replaced the ℓ^2 distance in Eqn. 6 by one-class learners. We used the most popular one-class SVM (OCSVM) and Isolation Forest (IsoForest).

Table 5. Comparisons of CROSR with recent GAN-based methods [10].

Method / dataset	MNIST	SVHN	TinyImageNet
Openmax	0.981 ± 0.005	0.894 ± 0.013	0.576
G-Openmax	0.984 ± 0.005	0.896 ± 0.017	0.580
Counterfactual	0.988 ± 0.004	$\mathbf{0.910} \pm 0.010$	0.586
CROSR (ours)	$\mathbf{0.991 \pm 0.004}$	0.899 ± 0.018	**0.589**

Table 6. Open-set classification results for MNIST with different unknown detectors. Larger values are better.

UNK detector	Omniglot	Noise	MNIST-noise
Supervised +			
$-\ell^2$	0.680	0.890	0.720
–OCSVM	0.647	0.899	0.919
Our DHRNet +			
$-\ell^2$	**0.793**	0.826	0.827
–OCSVM	0.702	**0.979**	**0.976**
–IsoForest	0.649	0.908	0.839

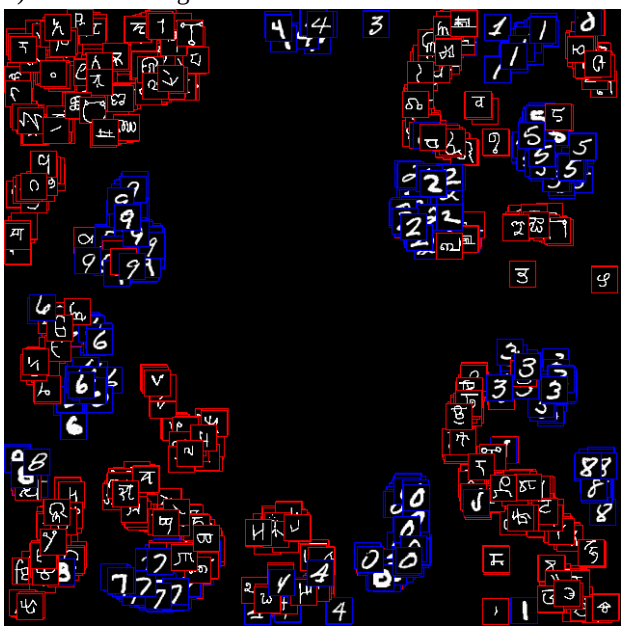
For simplicity, we used the default hyperparameters in `scikit-learn` [29]. The results are shown in Table 6. It reveals that OCSVM had a more than 15% gain in F1-score in synthesized outliers, while it caused a 9% degradation in Omniglot. Although we did not find an anomaly detector that consistently gave performance improvements on all the datasets, the results are still encouraging. The results suggest that DHRNet encodes more useful information that is not fully exploited by the per-class centroid based outlier modeling.

Visualization Figure 6 shows the test data from the known and unknown classes, sorted by the models' final confidences computed by Eqn. 3. In this figure, unknown data at higher order mean that the model is deceived by that data. It is clear that our methods gave lower confidences to the unknown samples, and they were deceived only by samples that had high similarity to the inlier.

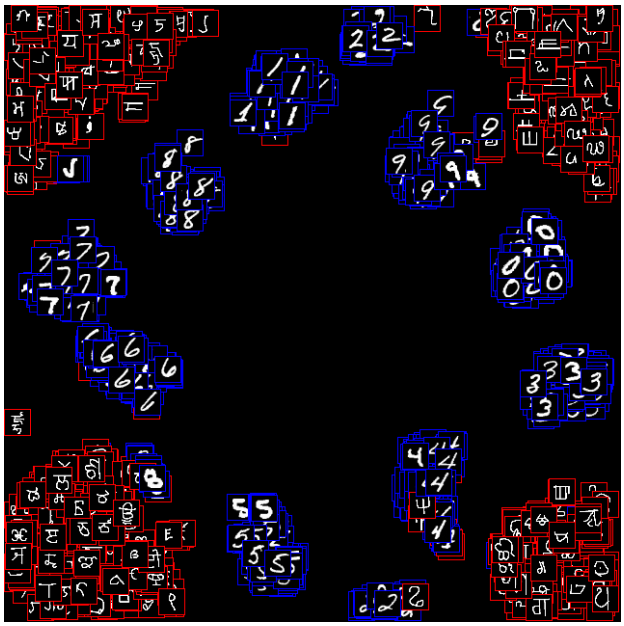
We additionally visualize the learned representations by using t-distributed stochastic neighbor embedding (t-SNE) [24]. Figure 7 shows distributions of the representations extracted from known- and unknown-class images in the test sets, embedded into two-dimensional planes. Here we compare the distributions of the prediction $\boldsymbol{y}$ from the supervised net and that of the concatenation of the prediction and the latent variable $[\boldsymbol{y}, \boldsymbol{z}]$ from our DHRNet. Their usages are shown in Eqns. (4) and (6) of the main text. While the existing deep open-set classifiers exploit only $\boldsymbol{y}$, our CROSR exploits $[\boldsymbol{y}, \boldsymbol{z}]$. With the latent representation, the clusters of knowns and unknowns are more clearly separated, and this suggests that the representations learned by our DHRNet are preferable for open-set classification.

Run time Despite of the extensions we made to the network, CROSR's computational cost in the test was not much larger than Openmax's. Figure 7 shows the run times, which were computed on a single GTX Titan X graphic processor. The overhead of computing the latent representations was as small as 3–5 ms/image, negligible in relation to the orig-

A) MNIST-Omniglot

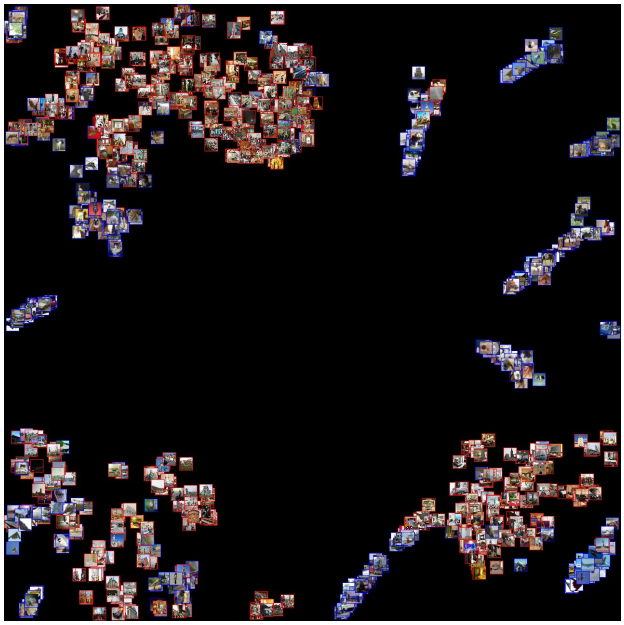


a) Supervised net

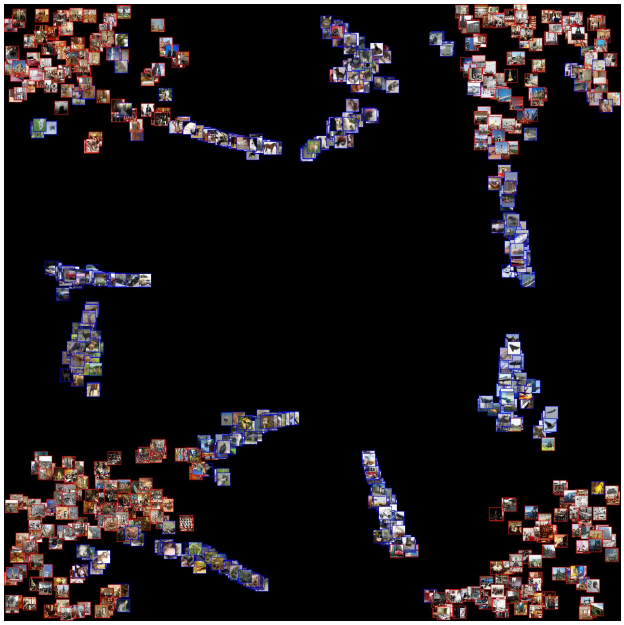


b) DHRNet (ours)

B) CIFAR10-LSUN



a) Supervised net



b) DHRNet (ours)

Figure 7. Distributions of the known- and unknown-class images from the test sets over the representation spaces. Images with blue frames are known samples, and ones with red are unknowns. With the representations from our DHRNet, which contain both the prediction $\boldsymbol{y}$ and reconstruction latent variables $\boldsymbol{z}$, the clusters of knowns and unknowns are more clearly separated.

inal cost when the backbone network is large.

6. Conclusion

We described CROSR, a deep open-set classifier augmented by latent representation learning for reconstruction. To enhance the usability of latent representations for un-

Table 7. Run times of the models (milli seconds/image). The times were measured in CIFAR-10 with a batch size = 1.

Method / Architecture	Plain CNN	DenseNet
Softmax	9.3	63.2
Openmax	11.7	69.4
CROSR (ours)	16.5	72.4

known detection, we also developed a novel deep hierarchical reconstruction net architecture. Comprehensive experiments conducted on multiple standard datasets demonstrated that CROSR outperforms previous state-of-the-art open-set classifiers in most cases.

Acknowledgement

This work is in part supported by JSPS KAKENHI Grant Number JP18K11348, and Grant-in-Aid for JSPS Fellows JP16J04552. The authors would like to thank Dr. Ari Hautasaari for his helpful advice to improve the manuscript.

References

[1] C. Aytekin, X. Ni, F. Cricri, and E. Aksu. Clustering and unsupervised anomaly detection with L2 normalized deep auto-encoder representations. In *IJCNN*, 2018. 2, 3

[2] A. Bendale and T. Boult. Towards open world recognition. In *CVPR*, pages 1893–1902, 2015. 3

[3] A. Bendale and T. E. Boult. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016. 1, 2, 3, 7

[4] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010. 5

[5] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 2

[6] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *CVPR*, pages 4829–4837, 2016. 2

[7] G. Fei and B. Liu. Breaking the closed world assumption in text classification. In *NAACL-HLT*, 2016. 2

[8] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936. 2

[9] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 2

[10] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi. Generative OpenMax for multi-class open set classification. *BMVC*, 2017. 1, 2, 7, 8

[11] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, pages 597–613. Springer, 2016. 2

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 3

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 3

[14] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 3, 5, 6

[15] R. Hinami, T. Mei, and S. Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *ICCV*, pages 3639–3647, 2017. 3

[16] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 2

[17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017. 4, 6

[18] P. R. M. Júnior, R. M. de Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017. 2

[19] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014. 7

[20] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018. 3, 6

[21] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *International Conference on Data Mining (ICDM)*, pages 413–422. IEEE, 2008. 3

[22] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are GANs created equal? A large-scale study. In *NIPS*, 2018. 7

[23] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. In *ICML*, 2016. 3

[24] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9(Nov):2579–2605, 2008. 8

[25] L. M. Manevitz and M. Yousef. One-class SVMs for document classification. *JMLR*, 2(Dec):139–154, 2001. 2

[26] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969. reprinted in McC90. 1

[27] R. K. Mobley, L. R. Higgins, and D. J. Wikoff. *Maintenance engineering handbook*. Mcgraw-hill New York, NY, 2008. 4

[28] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li. Open set learning with counterfactual images. *ECCV*, 2018. 5, 6, 7

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011. 8

[30] P. Perera and V. M. Patel. Learning deep features for one-class classification. *arXiv preprint arXiv:1801.05365*, 2018. 3

[31] M. Pezeshki, L. Fan, P. Brakel, A. Courville, and Y. Bengio. Deconstructing the ladder network architecture. In *ICML*, pages 2368–2376, 2016. 4

[32] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *NIPS*, pages 3546–3554, 2015. 2, 3, 5

[33] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for traffic anomaly detection. *ACM SIGMETRICS Performance Evaluation Review*, 35(1):109–120, 2007. 3

[34] S. Roberts and L. Tarassenko. A probabilistic resource allocating network for novelty detection. *Neural Computation*, 6(2):270–284, 1994. 3

[35] E. Rudd, L. P. Jain, W. J. Scheirer, and T. Boult. The extreme value machine. *PAMI*, 40(3), March 2017. 2, 3

[36] Y. Sasaki et al. The truth of the F-measure. *Teach Tutor mater*, 1(5):1–5, 2007. 8

[37] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *PAMI*, 35(7):1757–1772, 2013. 1

[38] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *PAMI*, 36(11):2317–2324, 2014. 2

[39] W. J. Scheirer, A. Rocha, R. Michaels, and T. E. Boult. Meta-recognition: The theory and practice of recognition score analysis. *PAMI*, 33:1689–1695, 2011. 7

[40] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017. 3

[41] L. Shu, H. Xu, and B. Liu. DOC: Deep open classification of text documents. In *EMNLP*, 2017. 1, 2, 5, 7

[42] L. Shu, H. Xu, and B. Liu. Unseen class discovery in open-world classification. *arXiv preprint arXiv:1801.05609*, 2018. 2

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 6

[44] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, et al. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018. 1

[45] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *NIPSW*, volume 5, pages 1–6, 2015. 7

[46] H. Valpola. From neural PCA to deep unsupervised learning. In *Advances in Independent Component Analysis and Learning Machines*, pages 143–171. Elsevier, 2015. 4

[47] M. J. Wilber, W. J. Scheirer, P. Leitner, B. Heflin, J. Zott, D. Reinke, D. K. Delaney, and T. E. Boult. Animal recognition in the mojave desert: Vision tools for field biologists. In *WACV*, pages 206–213. IEEE, 2013. 1

[48] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. *BMVC*, 2015. 3

[49] H. Zhang and V. M. Patel. Sparse representation-based open set recognition. *PAMI*, 39(8):1690–1696, 2017. 2

[50] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *NIPS*, pages 649–657, 2015. 7

[51] Y. Zhang, K. Lee, and H. Lee. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *ICML*, pages 612–621, 2016. 2

[52] C. Zhou and R. C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 665–674. ACM, 2017. 2, 3

[53] A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012. 2

[54] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018. 3