Conditional Gaussian Distribution Learning for Open Set Recognition

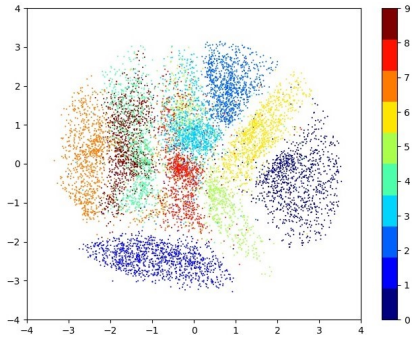
Xin Sun[1], Zhenning Yang[1], Chi Zhang[1], Keck-Voon Ling[2], Guohao Peng[1]
Nanyang Technological University, Singapore
[1]{xin001,zhenning002,chi007,peng0086}@e.ntu.edu.sg, [2]ekvling@ntu.edu.sg

Abstract

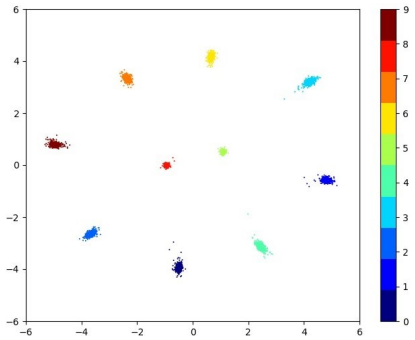
Deep neural networks have achieved state-of-the-art performance in a wide range of recognition/classification tasks. However, when applying deep learning to real-world applications, there are still multiple challenges. A typical challenge is that unknown samples may be fed into the system during the testing phase and traditional deep neural networks will wrongly recognize the unknown sample as one of the known classes. Open set recognition is a potential solution to overcome this problem, where the open set classifier should have the ability to reject unknown samples as well as maintain high classification accuracy on known classes. The variational auto-encoder (VAE) is a popular model to detect unknowns, but it cannot provide discriminative representations for known classification. In this paper, we propose a novel method, Conditional Gaussian Distribution Learning (CGDL), for open set recognition. In addition to detecting unknown samples, this method can also classify known samples by forcing different latent features to approximate different Gaussian models. Meanwhile, to avoid information hidden in the input vanishing in the middle layers, we also adopt the probabilistic ladder architecture to extract high-level abstract features. Experiments on several standard image datasets reveal that the proposed method significantly outperforms the baseline method and achieves new state-of-the-art results.

1. Introduction

In the past few years, deep learning has achieved state-of-the-art performance in many recognition/classification tasks [9, 10, 19, 26], but there are still multiple challenges when applying deep learning to real-world problems. One typical challenge is that incomplete knowledge exists during the training phase, and unknown samples may be fed into the system during the testing phase. While traditional recognition/classification tasks are under a common closed set assumption: all training and testing data come from the same label space. When meeting an unknown sample, traditional deep neural networks (DNNs) will wrongly recognize it as one of the known classes.



(a) VAE



(b) Ours: CGDL

Figure 1: Comparison of latent representations on MNIST dataset of the VAE (a) and the proposed method CGDL (b). The VAE is widely used in unknown detection, but it cannot provide discriminative features to undertake classification tasks as all features just follow one distribution. Comparatively, the proposed method can learn conditional Gaussian distributions by forcing different latent features to approximate different Gaussian models, which enables the proposed method to classify known samples as well as reject unknown samples.

The concept of open set recognition (OSR) [31] was proposed, assuming the testing samples can come from any classes, even unknown during the training phase. The open set classifier should have a dual character: unknown de-

tection and known classification[1]. Considering that during training it is not available to extract information from unknown samples, to realize unknown detection, many previous works analyze information from known samples by unsupervised learning [2, 28, 43, 44]. Among them, the variational auto-encoder (VAE) [13] is a popular method, in combination with clustering [2], GMM [44], or one-class [28] algorithm. The VAE is a probabilistic graphical model which is trained not only to reconstruct the input accurately, but also to force the posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ in the latent space to approximate one prior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, such as the multivariate Gaussian or Bernoulli distribution. The well-trained VAE is able to correctly describe known data, and deviated samples will be recognized as unknown. Fig. 1a is an example of the VAE latent representations on MNIST dataset when the prior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is the multivariate Gaussian. Although the VAE excels at unknown detection, it cannot provide discriminative representations to undertake classification tasks as all features only follow one distribution.

Here, to overcome this shortcoming, we propose a novel method, Conditional Gaussian Distribution Learning (CGDL), for open set recognition. Different from traditional VAEs, the proposed method is able to generate class conditional posterior distributions $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, k)$ in the latent space where k is the index of known classes. These conditional distributions are forced to approximate different multivariate Gaussian models $p_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_k, \mathbf{I})$ where $\boldsymbol{\mu}_k$ is the mean of the k-th multivariate Gaussian distribution, obtained by a fully-connected layer that maps the one-hot encoding of the input's label to the latent space. Fig. 1b is an example of latent representations of the proposed method on MNIST dataset. These learned features will be fed to an open set classifier, which consists of two parts: an unknown detector and a closed set classifier. As known samples tend to follow the prior distributions, the unknown detector will recognize those samples locating in lower probability regions as unknown. Meanwhile, for the known sample, the closed set classifier will calculate its prediction scores over all known classes and predict it as the class with the highest score.

Current networks tend to go deeper for higher accuracy in recognition/classification tasks [35]. However, traditional VAEs are restricted to shallow models as details of input could be lost in higher layers [25], which limits VAE's ability to extract high-level abstract features. To fully exploit information from known samples, we adopt the probabilistic ladder network [34] into the proposed method. This probabilistic ladder architecture allows information interactions between the upward path and the downward path, which enables the decoder to recover details discarded by the encoder. Although there are several successful applications of the probabilistic ladder network [7, 14, 25], this paper is the first to apply it to open set recognition.

In our experiments, we explore the importance of the probabilistic ladder architecture and the conditional distributions in the latent space for open set recognition. We empirically demonstrate that our method significantly outperforms baseline methods. In summary, this paper makes the following contributions:

- We propose a novel open set recognition method, called Conditional Gaussian Distribution Learning (CGDL). Compared with previous methods based on VAEs, the proposed method is able to learn conditional distributions for known classification and unknown detection.
- We develop a fully-connected layer to get the means of different multivariate Gaussian models, which enables posterior distributions in the latent space to approximate different Gaussian models.
- We adopt a probabilistic ladder architecture to learn high-level abstract latent representations to further improve open set classification scores.
- We conduct experiments on several standard image datasets, and the results show that our method outperforms existing methods and achieves new state-of-the-art performance.

2. Related Work

Open Set Recognition. The methods for open set recognition (OSR) can be broadly divided into two branches: traditional methods (e.g., SVM, sparse representation, Nearest Neighbor,etc.) and deep learning-based methods. In traditional methods, Scheirer *et al.* [31] proposed an SVM based method which adds an extra hyper-line to detect unknown samples. Jain *et al.* [11] proposed the P_I-SVM algorithm, which is able to reject unknown samples by adopting EVT to model the positive training samples at the decision boundary. Cevikalp *et al.* [5, 6] defined the acceptance regions for known samples with a family of quasi-linear 'polyhedral conic' functions. Zhang *et al.* [42] pointed out that discriminative information is mostly hidden in the reconstruction error distributions, and proposed the sparse representation-based OSR model, called SROSR. Bendale *et al.* [3] recognized unknown samples based on the distance between the testing samples and the centroids of the known classes. Júnior *et al.* [12] proposed the Nearest Neighbor Distance Ratio (NNDR) technique, which carries out OSR according to the similarity score between the two most similar classes. Considering deep learning achieves state-of-the-art performance in a wide range of

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

recognition/classification tasks, deep learning-based open set recognition methods are gaining more and more attention.

In deep learning-based methods, Bendale *et al.* [4] proposed the Openmax function to replace the Softmax function in CNNs. In this method, the probability distribution of Softmax is redistributed to get the class probability of unknown samples. Based on Openmax, Ge *et al.* [8] proposed the Generative Openmax method, using generative models to synthesize unknown samples to train the network. Shu *et al.* [33] proposed the Deep Open Classifier (DOC) model, which replaces the Softmax layer with a 1-vs-rest layer containing sigmoid functions. Counterfactual image generation, a dataset augmentation technique proposed by Neal *et al.* [22], aims to synthesize unknown-class images. Then the decision boundaries between unknown and known classes can be converged from these known-like but actually unknown sample sets. Yoshihashi *et al.* [37] proposed the CROSR model, which combines the supervised learned prediction and unsupervised reconstructive latent representation to redistribute the probability distribution. Oza and Patel [24] trained a class conditional auto-encoder (C2AE) to get the decision boundary from the reconstruction errors by extreme value theory (EVT). The training phase of C2AE is divided into two steps (closed-set training and open-set training), and a batch of samples need to be selected from training data to generate non-match reconstruction errors. This is difficult in practice and testing results are highly dependent on the selected samples. On the contrary, the proposed method is an end-to-end system and does not need extra data pre-processing.

Anomaly Detection. Anomaly detection (also called outlier detection) aims to distinguish anomalous samples from normal samples, which can be introduced into OSR for unknown detection. Some general anomaly detection methods are based on Support Vector Machine (SVM) [36, 21] or forests [27]. In recent years, deep neural networks have also been used in anomaly detection, mainly based on auto-encoders trained in an unsupervised manner [43, 2, 28, 44]. Auto-encoders commonly have a bottleneck architecture to induce the network to learn abstract latent representations. Meanwhile, these networks are typically trained by minimizing reconstruction errors. In anomaly detection, the training samples commonly come from the same distribution, thus the well-trained auto-encoders could extract the common latent representations from the normal samples and reconstruct them correctly, while anomalous samples do not contain these common latent representations and could not be reconstructed correctly. Although VAEs are widely applied in anomaly detection, it cannot provide discriminative features for classification tasks.

Apart from auto-encoders, some studies used Generative Adversarial Networks (GANs) to detect anomalies [32]. GANs are trained to generate similar samples according to the training samples. Given a testing sample, the GAN tries to find the point in the generator's latent space that can generate a sample closest to the input. Intuitively, the well-trained GAN could give good representations for normal samples and terrible representations for anomalies.

There are also some related tasks focusing on novel classes. For example, few-shot learning [39, 40, 41] aims to undertake vision tasks on new classes with scarce training data. Incremental learning [20] aims to make predictions on both old classes and new classes without accessing data in old classes.

3. Preliminaries

Before introducing the proposed method, we briefly introduce the terminology and notation of VAE [13].

The VAE commonly consists of an encoder, a decoder and a loss function $\mathcal{L}(\boldsymbol{\theta};\boldsymbol{\phi};\boldsymbol{x})$. The encoder is a neural network that has parameters $\boldsymbol{\phi}$. Its input is a sample $\boldsymbol{x}$ and its output is a hidden representation $\boldsymbol{z}$. The decoder is another neural network with parameters θ. Its input is the representation $\boldsymbol{z}$ and it outputs the probability distribution of the sample. The loss function in the VAE is defined as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta};\boldsymbol{\phi};\boldsymbol{x}) = &-D_{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})\big) \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\big]\end{aligned} \tag{1}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is the approximate posterior, $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is the prior distribution of the latent representation $\boldsymbol{z}$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the likelihood of the input $\boldsymbol{x}$ given latent representation $\boldsymbol{z}$. On the right-hand side of Eqn. 1, the first term is the KL-divergence between the approximate posterior and the prior. It can be viewed as a regularizer to encounter the approximate posterior to be close to the prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. The second term can be viewed as the reconstruction errors.

Commonly, the prior over the latent representation $\boldsymbol{z}$ is the centered isotropic multivariate Gaussian $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z};\mathbf{0},\mathbf{I})$. In this case, the variational approximate posterior could be a multivariate Gaussian with a diagonal covariance structure:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z};\boldsymbol{\mu},\boldsymbol{\sigma}^2\mathbf{I}) \tag{2}$$

where the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ of the approximate posterior are outputs of the encoding multi-layered perceptrons (MLPs). The latent representation $\boldsymbol{z}$ is defined as $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ and $\odot$ is the element-wise product. Let J be the dimensionality of $\boldsymbol{z}$, then the KL-divergence can be calculated [13]:

$$\begin{aligned}&-D_{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z})\big) \\ &= \frac{1}{2}\sum_{j=1}^{J}\big(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\big)\end{aligned} \tag{3}$$

With loss function $\mathcal{L}(\boldsymbol{\theta};\boldsymbol{\phi};\boldsymbol{x})$, the VAE is trained not only to reconstruct the input accurately, but also to force the posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ in the latent space to approximate the prior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. If a sample locates in the low probability region of the learned distribution, this sample will be recognized as unknown.

4. Proposed Method

In this section, we describe the proposed method in detail. Firstly, we describe the architecture of the proposed model. Then, we introduce the training phase and the testing phase to describe the functions of each module.

4.1. Architecture

The architecture of the proposed method is composed of four modules (as shown in Fig. 2):
1. Encoder $\mathcal{F}$
2. Decoder $\mathcal{G}$
3. Known Classifier $\mathcal{C}$
4. Unknown Detector $\mathcal{D}$

Encoder $\mathcal{F}$. To extract high-level abstract latent features, the probabilistic ladder architecture is adopted in each layer. In detail, the l-th layer in the encoder $\mathcal{F}$ is expressed as follows:

$$\begin{aligned}\boldsymbol{x}_l &= \text{Conv}(\boldsymbol{x}_{l-1})\\ \boldsymbol{h}_l &= \text{Flatten}(\boldsymbol{x}_l)\\ \boldsymbol{\mu}_l &= \text{Linear}(\boldsymbol{h}_l)\\ \boldsymbol{\sigma}_l^2 &= \text{Softplus}(\text{Linear}(\boldsymbol{h}_l)\end{aligned}$$

where `Conv` is a convolutional layer followed by a batch-norm layer and a PReLU layer, `Flatten` is a linear layer to flatten 2-dimensional data into 1-dimension, `Linear` is a single linear layer and `Softplus` applies $\log(1+\exp(\cdot))$ non-linearity to each component of its argument vector (Fig. 3 illustrates these operations). The latent representation $\boldsymbol{z}$ is defined as $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\odot$ is the element-wise product, and $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ are the outputs of the top layer L.

Decoder $\mathcal{G}$. The l-th layer in the decoder $\mathcal{G}$ is expressed as follows:

$$\begin{aligned}\tilde{\boldsymbol{c}}_{l+1} &= \text{Unflatten}(\tilde{\boldsymbol{z}}_{l+1})\\ \tilde{\boldsymbol{x}}_{l+1} &= \text{ConvT}(\tilde{\boldsymbol{c}}_{l+1})\\ \tilde{\boldsymbol{h}}_{l+1} &= \text{Flatten}(\tilde{\boldsymbol{x}}_{l+1})\\ \tilde{\boldsymbol{\mu}}_l &= \text{Linear}(\tilde{\boldsymbol{h}}_{l+1})\\ \tilde{\boldsymbol{\sigma}}_l^2 &= \text{Softplus}(\text{Linear}(\tilde{\boldsymbol{h}}_{l+1})\\ \boldsymbol{z}_l &= \tilde{\boldsymbol{\mu}}_l + \tilde{\boldsymbol{\sigma}}_l^2 \odot \boldsymbol{\epsilon}\end{aligned}$$

where `ConvT` is a transposed convolutional layer and `Unflatten` is a linear layer to convert 1-dimensional data into 2-dimension (Fig. 4 illustrates these operations). In the

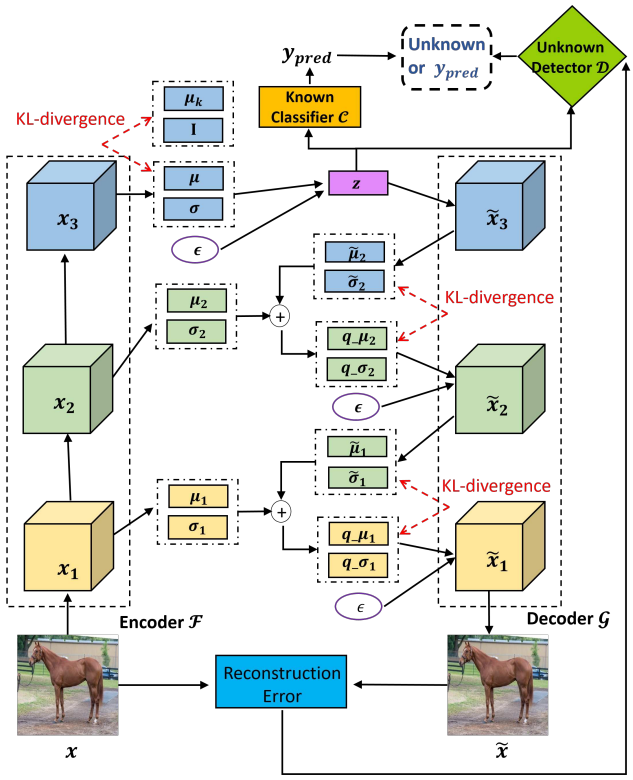


Figure 2: Block diagram of the proposed method: The **encoder $\mathcal{F}$** and **decoder $\mathcal{G}$** are applied with the probabilistic ladder architecture to extract high-level abstract latent features. The **known classifier $\mathcal{C}$** takes latent representations as input and produces the probability distribution over the known classes. The **unknown detector $\mathcal{D}$** is modeled by the conditional Gaussian distributions and reconstruction errors from training samples, which is used for unknown detection. During training, the proposed model is trained to minimize the sum of the reconstruction loss $\mathcal{L}_r$, KL-divergence $\mathcal{L}_{KL}$ (both in the latent space and middle layers) and classification loss $\mathcal{L}_c$. During testing, the **unknown detector $\mathcal{D}$** will judge whether this sampler is unknown by its latent features and reconstruction errors. If this sample is known, the **known classifier $\mathcal{C}$** will give out its predicted label.

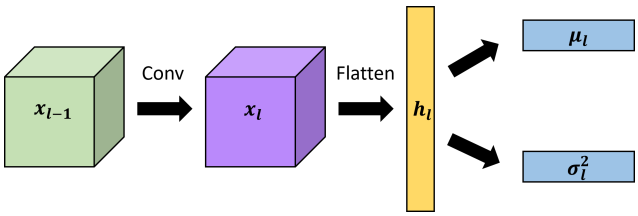


Figure 3: Operations in the upward pathway.

l-th layer, the bottom-up information ($\boldsymbol{\mu}_l$ and $\boldsymbol{\sigma}_l$) and top-down information ($\tilde{\boldsymbol{\mu}}_l$ and $\tilde{\boldsymbol{\sigma}}_l$) are interacted by the following equations defined in [34]:

$$\boldsymbol{q}\_\boldsymbol{\mu}_l = \frac{\tilde{\boldsymbol{\mu}}_l \tilde{\boldsymbol{\sigma}}_l^{-2} + \boldsymbol{\mu}_l \boldsymbol{\sigma}_l^{-2}}{\tilde{\boldsymbol{\sigma}}_l^{-2} + \boldsymbol{\sigma}_l^{-2}} \tag{4}$$

$$\boldsymbol{q}\_\boldsymbol{\sigma}_l^{\boldsymbol{2}} = \frac{1}{\tilde{\boldsymbol{\sigma}}_l^{-2} + \boldsymbol{\sigma}_l^{-2}} \tag{5}$$

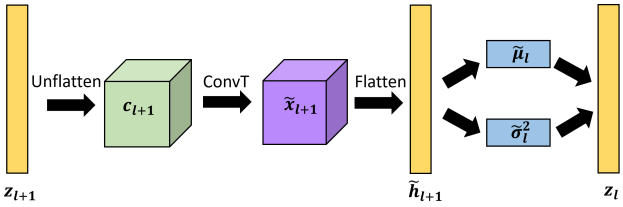


Figure 4: Operations in the downward pathway.

Known Classifier $\mathcal{C}$. The known classifier $\mathcal{C}$ is a Softmax layer, which takes the latent representation $\boldsymbol{z}$ as input. It produces the probability distribution over the known classes.

Unknown Detector $\mathcal{D}$. When training is completed, the unknown detector $\mathcal{D}$ is modeled by information hidden in the latent representations and reconstruction errors. During the testing phase, the unknown detector $\mathcal{D}$ is used as a binary classifier to judge whether the input is known or unknown (details are discussed in Sec. 4.3).

4.2. Training

During the training phase, the proposed model forces the conditional posterior distributions $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, k)$ to approximate different multivariate Gaussian models $p_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_k, \mathbf{I})$ where k is the index of known classes, and the mean of k-th Gaussian distribution $\boldsymbol{\mu}_k$ is obtained by a fully-connected layer which maps the one-hot encoding of the input's label to the latent space. The KL-divergence in latent space (Eqn. 3) is modified as follows:

$$\begin{aligned}
&-D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, k) \,||\, p_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{z})) \\
&= \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, k)\big(\log p_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, k)\big) d\boldsymbol{z} \\
&= \int \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)\big(\log \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_k, \mathbf{I}) - \log \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)\big) d\boldsymbol{z} \\
&= \frac{1}{2}\sum_{j=1}^{J}\big(1 + \log(\sigma_j^2) - (\mu_j - \mu_j^{(k)})^2 - \sigma_j^2\big)
\end{aligned} \tag{6}$$

During the training phase, the model is trained to minimize the sum of the reconstruction loss $\mathcal{L}_r$, KL-divergence $\mathcal{L}_{KL}$ and classification loss $\mathcal{L}_c$. To measure classification loss $\mathcal{L}_c$, we use softmax cross-entropy of prediction and ground-truth labels. To measure reconstruction loss $\mathcal{L}_r$, we use the L_1 distance between input images $\boldsymbol{x}$ and reconstructed image $\tilde{\boldsymbol{x}}$. As the probabilistic ladder architecture is adopted, the KL-divergence is considered not only in the latent space but also in the middle layers:

$$\begin{aligned}
\mathcal{L}_{KL} = -\frac{1}{L}\big[&D_{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, k) \,||\, p_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{z})\big) \\
&+ \sum_{l=1}^{L-1} D_{KL}\big(q_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l+1}, \boldsymbol{x}) \,||\, q_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l+1})\big)\big]
\end{aligned} \tag{7}$$

where

$$q_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l+1}, \boldsymbol{x}) = \mathcal{N}(\tilde{\boldsymbol{x}}_l; \boldsymbol{q}\_\boldsymbol{\mu}_l, \boldsymbol{q}\_\boldsymbol{\sigma}_l^{\boldsymbol{2}}) \tag{8}$$

$$q_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l+1}) = \mathcal{N}(\tilde{\boldsymbol{x}}_l; \tilde{\boldsymbol{\mu}}_l, \tilde{\boldsymbol{\sigma}}_l^{\boldsymbol{2}}) \tag{9}$$

The loss function used in our model is summarized as follows:

$$\mathcal{L} = -(\mathcal{L}_r + \beta\mathcal{L}_{KL} + \lambda\mathcal{L}_c) \tag{10}$$

where β is increased linearly from 0 to 1 during the training phase as described in [34] and λ is a constant.

4.3. Testing

When training is completed, we model the per class multivariate Gaussian model $f_k(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}_k, \boldsymbol{\sigma}_k^2)$ where $\boldsymbol{m}_k$ and $\boldsymbol{\sigma}_k^2$ are the mean and variance of the latent representations of all correctly classified training samples in k-th class. If the dimension of the latent space is n: $\boldsymbol{z} = (z_1, ..., z_n)$, the probability of a sample locating in the distribution $f_k(\boldsymbol{z})$ is defined as follows:

$$P_k(\boldsymbol{z}) = 1 - \int_{m_0-|z_0-m_0|}^{m_0+|z_0-m_0|} \cdots \int_{m_n-|z_n-m_n|}^{m_n+|z_n-m_n|} f_k(\boldsymbol{t})\, d\boldsymbol{t} \tag{11}$$

We also analyze information hidden in the reconstruction errors. The reconstruction errors of input from known classes are commonly smaller than that of unknown classes [24]. Here we obtain the reconstruction error threshold by ensuring 95% training data to be recognized as known. Details of the testing procedure are described in Algo. 1.

5. Experiments and Results

5.1. Implementation details

In the proposed method, we use the SGD optimizer with a learning rate of 0.001, and fix the batch size to 64. The backbone is the re-designed VGGNet defined in [37]. The dimensionality of the latent representation $\boldsymbol{z}$ is fixed to 32. For loss function described in Sec. 4.2, the parameter β is increased linearly from 0 to 1 during the training phase as

Algorithm 1 Testing procedure

Require: Testing sample $\boldsymbol{X}$
Require: Trained modules $\mathcal{F}, \mathcal{G}, \mathcal{C}$
Require: Threshold τ_l of Gaussian distributions
Require: Threshold τ_r of reconstruction errors
Require: For each class k, let $\boldsymbol{z}_{i,k}$ is the latent representation of each correctly classified training sample $\boldsymbol{x}_{i,k}$
for $k = 1, \ldots, K$ **do**
  compute the mean and variance of each class: $\boldsymbol{m}_k = mean_i(\boldsymbol{z}_{i,k})$, $\boldsymbol{\sigma}_k^2 = var_i(\boldsymbol{z}_{i,k})$
  model the per class multivariate Gaussian: $f_k(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}_k, \boldsymbol{\sigma}_k^2)$
end for
latent representation $\boldsymbol{Z} = \mathcal{F}(\boldsymbol{X})$
predicted known label $y_{pred} = argmax(\mathcal{C}(\boldsymbol{Z}))$
reconstructed image $\tilde{\boldsymbol{X}} = \mathcal{G}(\boldsymbol{Z})$
reconstruction error $R = ||\boldsymbol{X} - \tilde{\boldsymbol{X}}||_1$
if $\forall k \in \{1, ..., K\}, P_k(\boldsymbol{Z}) < \tau_l$ **or** $R > \tau_r$ **then**
  predict $\boldsymbol{X}$ as unknown
else
  predict $\boldsymbol{X}$ as known with label y_{pred}
end if

described in [34], while the parameter λ is set equal to 100. The networks were trained without any large degradation in closed set accuracy from the original ones. The closed set accuracy of the networks for each dataset are listed in Table. 1. The threshold τ_l of conditional Gaussian distributions is set to 0.5, and the threshold τ_r of reconstruction errors is obtained by ensuring 95% training data be recognized as known.

5.2. Ablation Analysis

In this section, we analyze our contributions from each component of the proposed method on CIFAR-100 dataset [16]. The CIFAR-100 dataset consists of 100 classes, containing 500 training images and 100 testing images in each class. For ablation analysis, the performance is measured by F-measure (or F1-scores) [30] against varying Openness [31]. Openness is defined as follows:

$$Openness = 1 - \sqrt{\frac{2 \times N_{train}}{N_{test} + N_{target}}} \tag{12}$$

where N_{train} is the number of known classes seen during training, N_{test} is the number of classes that will be observed during testing, and N_{target} is the number of classes to be recognized during testing. We randomly sample 15 classes out of 100 classes as known classes and varying the number of unknown classes from 15 to 85, which means Openness is varied from 18% to 49%. The performance is evaluated by the macro-average F1-scores in 16 classes (15 known classes and *unknown*).

Table 1: Comparison of closed set test accuracies between the plain CNN and the proposed method CGDL. Although the training objective of CGDL is classifying known samples as well as learning conditional Gaussian distributions, there is no significant degradation in closed set accuracy.

Architecture	MNIST	SVHN	CIFAR-10
Plain CNN	0.997	0.944	0.912
CGDL	0.996	0.942	0.912

We compare the following baselines for ablation analysis:

I. CNN: In this baseline, only the encoder $\mathcal{F}$ (without ladder architecture) and the known classifier $\mathcal{C}$ are trained for closed set classification. This model can be viewed as a traditional convolutional neural network (CNN). During testing, learned features will be fed to $\mathcal{C}$ to get the probability scores of known classes. A sample will be recognized as unknown if its probability score of predicted label is less than 0.5.

II. CVAE: The encoder $\mathcal{F}$, decoder $\mathcal{G}$ and classifier $\mathcal{C}$ are trained without the ladder architecture, and the testing procedure is the same as baseline I. This model can be viewed as a class conditional variational auto-encoder (CVAE).

III. LCVAE: The probabilistic ladder architecture is adopted in the CVAE, which contributes to the KL-divergences during training (Eqn. 7). We call this model as LCVAE. The testing procedure is the same as baseline I and II.

IV. CVAE+CGD: The model architecture and training procedure are the same as baseline II, while the conditional Gaussian distributions (CGD) are used to detect unknowns during testing.

V. LCVAE+CGD: In this baseline, LCVAE is introduced along with CGD-based unknown detector. The training and testing procedure are respectively the same as baseline III and IV.

VI. LCVAE+RE: Different from baseline V, reconstruction errors (RE), instead of CGD, are used in LCVAE to detect unknown samples.

VII. Proposed Method: The training procedure is the same as baseline V and VI, while during testing conditional Gaussian distributions and reconstruction errors are together used for unknown detection.

The experimental results are shown in Fig. 5. Among baseline I, II and III, unknown detection simply relies on the known classifier $\mathcal{C}$. Although the performance is a little improved when the probabilistic ladder architecture is adopted

Table 2: The Area Under the ROC curve (AUROC) on detecting known and unknown samples. Results are averaged among five randomized trials.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50	Ting-ImageNet
Softmax	0.978 ± 0.002	0.886 ± 0.006	0.677 ± 0.032	0.816 ± -	0.805 ± -	0.577 ± -
Openmax [4]	0.981 ± 0.002	0.894 ± 0.008	0.695 ± 0.032	0.817 ± -	0.796 ± -	0.576 ± -
G-Openmax [8]	0.984 ± 0.001	0.896 ± 0.006	0.675 ± 0.035	0.827 ± -	0.819 ± -	0.580 ± -
OSRCI [22]	0.988 ± 0.001	0.910 ± 0.006	0.699 ± 0.029	0.838 ± -	0.827 ± -	0.586 ± -
C2AE [24]	0.989 ± 0.002	0.922 ± 0.009	0.895 ± 0.008	0.955 ± 0.006	0.937 ± 0.004	0.748 ± 0.005
ours: CGDL	**0.994 ± 0.002**	**0.935 ± 0.003**	**0.903 ± 0.009**	**0.959 ± 0.006**	**0.950 ± 0.006**	**0.762 ± 0.005**

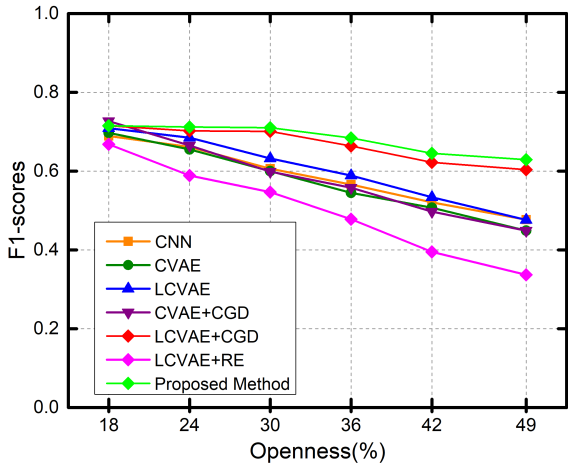


Figure 5: F1-scores against varying Openness with different baselines for ablation analysis.

(baseline III), the overall performance in these three baselines is weak as the F1-scores degrade rapidly as the Openness increases. Conditional Gaussian distributions (CGD) is added for unknown detection in CVAE model (baseline IV), but it has seen no visible change in performance. In baseline V, this trend is alleviated by introducing CGD-based unknown detector into LCVAE. This shows the importance of the probabilistic ladder architecture for open set recognition. It is also the reason why the CGD-based unknown detection achieves better performance in baseline V than in baseline IV. If we only use reconstruction errors to detect unknowns (baseline VI), the results are worst. However, if reconstruction errors are added to the CGD-based unknown detector (baseline VII), there is a little improvement in performance. As a result, applying conditional Gaussian distributions and reconstruction errors to detect unknowns with the probabilistic ladder architecture achieves the best performance.

Table 3: Open set classification results on MNIST dataset with various outliers added to the test set as unknowns. The performance is evaluated by macro-averaged F1-scores in 11 classes (10 known classes and *unknown*).

Method	Omniglot	MNIST-noise	Noise
Softmax	0.595	0.801	0.829
Openmax [4]	0.780	0.816	0.826
CROSR [37]	0.793	0.827	0.826
ours: CGDL	**0.850**	**0.887**	**0.859**

5.3. Comparison with State-of-the-art Results

In this section, we compare the proposed method with state-of-the-art methods. We report our results under two different experimental set-ups, where the difference is that in the first set-up, the performance is measured by the model's ability on detecting unknown samples, and in the second set-up, the performance is measured by F1-scores in all known classes and *unknown*.

Unknown Detection. Following the protocol defined in [22], we use four standard image datasets: MNIST [18], SVHN [23], CIFAR-10 [15] and Tiny-ImageNet [17], to measure the model's ability to identify unknown samples. For MNIST, SVHN and CIFAR-10 datasets, each dataset is randomly partitioned into 6 known classes and 4 unknown classes. Meanwhile, the model is also trained on CIFAR-10 as described previously with 4 known classes, but the test set is replaced with 10 unknown classes randomly chosen from CIFAR-100 [16] dataset. This task is reported as CIFAR+10. Similarly, 50 unknown classes are randomly chosen from CIFAR-100 [16] dataset to set up the CIFAR+50 task. For the Tiny-ImageNet dataset, we randomly choose 20 classes as known classes. The remaining 180 classes are defined as unknown. The performance is measured by the Area Under the ROC curve (AUROC) on detecting known and unknown samples, and the results shown in Table. 2 are averaged among 5 separate samples of known and unknown. From this table, we can see that our method sig-

Table 4: Open set classification results on CIFAR-10 dataset with various outliers added to the test set as unknowns. The performance is evaluated by macro-averaged F1-scores in 11 classes (10 known classes and *unknown*).
*We report the experimental results reproduced in [37].

Method	ImageNet-crop	ImageNet-resize	LSUN-crop	LSUN-resize
Softmax [37]*	0.639	0.653	0.642	0.647
Openmax [4]	0.660	0.684	0.657	0.668
LadderNet+Softmax [37]	0.640	0.646	0.644	0.647
LadderNet+Openmax [37]	0.653	0.670	0.652	0.659
DHRNet+Softmax [37]	0.645	0.649	0.650	0.649
DHRNet+Openmax [37]	0.655	0.675	0.656	0.664
CROSR [37]	0.721	0.735	0.720	0.749
C2AE [24]	0.837	0.826	0.783	0.801
ours: CGDL	**0.840**	**0.832**	**0.806**	**0.812**

nificantly outperforms previous works and achieves a new state-of-the-art performance.

Open Set Recognition. An ideal open set classifier can not only reject unknown samples but also classify known classes. In the following experiments, the models are trained by all training samples of one dataset, but in the testing phase, samples from another dataset are added to the test set as unknown samples. We measure the open set recognition performance by the macro-averaged F1-scores in known classes and *unknown* on MNIST and CIFAR-10 datasets.

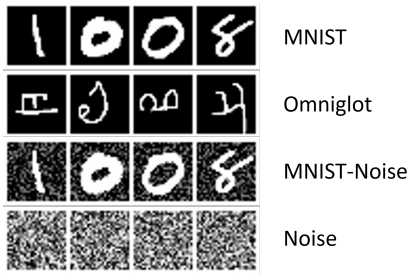


Figure 6: Examples from MNIST, Omniglot, MNIST-Noise, and Noise datasets.

Firstly, we choose MNIST, the most popular handwritten digit dataset, as the training set. As outliers, we follow the set up in [37], using Omniglot [1], MNIST-Noise, and Noise these three datasets. Omniglot is a dataset containing various alphabet characters. Noise is a synthesized dataset by setting each pixel value independently from a uniform distribution on $[0, 1]$. MNIST-Noise is also a synthesized dataset by adding noise on MNIST testing samples. Each dataset contains 10, 000 testing samples, the same as MNIST, and this makes the known-to-unknown ratio 1:1. Fig. 6 shows examples of these datasets. The open set recognition scores are shown in Table. 3 and the proposed method achieves the best results on all given datasets.

Secondly, following the protocol defined in [37], all samples in CIFAR-10 dataset are collected as known data, and samples from other datasets, i.e., ImageNet [29] and LSUN [38], are selected as unknown samples. We resize or crop the unknown samples to make them have the same size with known samples. ImageNet-crop, ImageNet-resize, LSUN-crop, and LSUN-resize these four datasets are generated, and each dataset contains 10,000 testing images, which is the same as CIFAR-10. This makes during testing the known-to-unknown ratio 1:1. The performance of the method is evaluated by macro-averaged F1-scores in 11 classes (10 known classes and *unknown*), and our results are shown in Table. 4. We can see from the results that on all given datasets, the proposed method is more effective than previous methods and achieves a new state-of-the-art performance.

6. Conclusion

In this paper, We have presented a novel method for open set recognition. Compared with previous methods solely based on VAEs, the proposed method can classify known samples as well as detect unknown samples by forcing posterior distributions in the latent space to approximate different Gaussian models. The probabilistic ladder architecture is adopted to preserve the information that may vanish in the middle layers. This ladder architecture obviously improves the open set performance. Moreover, reconstruction information is added to the unknown detector to further improve the performance. Experiments on several standard image datasets under two set-ups show that the proposed method significantly outperforms the baseline methods and achieves new state-of-the-art results.

References

[1] Simon Ager. Omniglot-writing systems and languages of the world. *Retrieved January*, 27:2008, 2008. 8

[2] Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu. Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks*, pages 1–6. IEEE, 2018. 2, 3

[3] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1893–1902, 2015. 2

[4] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 3, 7, 8

[5] Hakan Cevikalp and Hasan Serhan Yavuz. Fast and accurate face recognition with image sets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1564–1572, 2017. 2

[6] Hakan Cevikalp and Bill Triggs. Polyhedral conic classifiers for visual object detection and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 261–269, 2017. 2

[7] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pages 2199–2207, 2016. 2

[8] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. In *British Machine Vision Conference*, 2017. 3, 7

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[11] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer, 2014. 2

[12] Pedro R Mendes Júnior, Roberto M de Souza, Rafael de O Werneck, Bernardo V Stein, Daniel V Pazinato, Waldir R de Almeida, Otávio AB Penatti, Ricardo da S Torres, and Anderson Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017. 2

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. 2, 3

[14] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 2

[15] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010. 7

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 6, 7

[17] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 7

[18] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010. 7

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[20] Yaoyao Liu, An-An Liu, Yuting Su, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. *arXiv preprint arXiv:2002.10211*, 2020. 3

[21] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001. 3

[22] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision*, pages 613–628, 2018. 3, 7

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 7

[24] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3, 5, 7, 8

[25] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015. 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

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

[28] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4393–4402, 2018. 2, 3

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 8

[30] Yutaka Sasaki et al. The truth of the f-measure. *Teach Tutor mater*, 1(5):1–5, 2007. 6

[31] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012. 1, 2, 6

[32] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg 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

[33] Lei Shu, Hu Xu, and Bing Liu. Doc: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. 3

[34] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. How to train deep variational autoencoders and probabilistic ladder networks. In *33rd International Conference on Machine Learning*, 2016. 2, 5, 6

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[36] David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199, 1999. 3

[37] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019. 3, 5, 7, 8

[38] 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. 8

[39] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers, 2020. 3

[40] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9587–9595, 2019. 3

[41] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019. 3

[42] He Zhang and Vishal M Patel. Sparse representation-based open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1690–1696, 2016. 2

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

[44] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018. 2, 3