Conditional Variational Capsule Network for Open Set Recognition

Yunrui Guo[*1,2], Guglielmo Camporese[*2], Wenjing Yang[1], Alessandro Sperduti[2], and Lamberto Ballan[2]

[1]National University of Defense Technology, China
[2]Department of Mathematics "Tullio Levi-Civita", University of Padova, Italy

Abstract

In open set recognition, a classifier has to detect unknown classes that are not known at training time. In order to recognize new categories, the classifier has to project the input samples of known classes in very compact and separated regions of the features space for discriminating samples of unknown classes. Recently proposed Capsule Networks have shown to outperform alternatives in many fields, particularly in image recognition, however they have not been fully applied yet to open-set recognition. In capsule networks, scalar neurons are replaced by capsule vectors or matrices, whose entries represent different properties of objects. In our proposal, during training, capsules features of the same known class are encouraged to match a pre-defined gaussian, one for each class. To this end, we use the variational autoencoder framework, with a set of gaussian priors as the approximation for the posterior distribution. In this way, we are able to control the compactness of the features of the same class around the center of the gaussians, thus controlling the ability of the classifier in detecting samples from unknown classes. We conducted several experiments and ablation of our model, obtaining state of the art results on different datasets in the open set recognition and unknown detection tasks.

1. Introduction

Over the past decade, deep learning has become the dominant approach in many computer vision problems, achieving spectacular results on many visual recognition tasks [13, 27, 7, 29]. However, most of these results have been obtained in a closed set scenario, where a critical assumption is that all samples should belong to at least one labeled category. When observing a sample from an unknown category, closed-set approaches are forced to choose a class label from one of the known classes, thus limiting their applicability in dynamic and ever-changing scenarios.
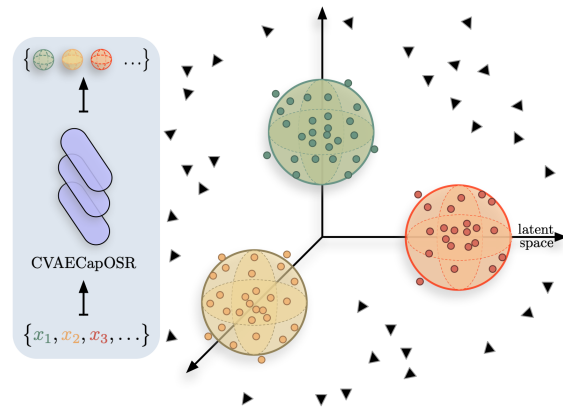
---

*Indicates equal contributions.

Figure 1: CVAECapOSR Model. Input samples are fed into the Capsule Network that produces distributions over the latent space. Each class has its own prior gaussian distribution in the feature space, that in the figure are represented as spheres. After training, the known samples (represented as small points) are clustered around the class target gaussians. The samples belonging to unknown classes are represented as black triangles, far from the target distributions.

To overcome such a limitation, open set recognition has been introduced to enable a classification system to identify all of known categories, while simultaneously detecting unknown test samples [25, 1]. In the open set scenario, samples included/excluded in label space are referred to as knowns/unknowns. Therefore, open set classifiers need to use incomplete knowledge learned from a finite set of accessible categories to devise effective representations able to separate knowns from unknowns. Early works have identified this issue, thus proposing methods employing different thresholding strategies for rejection of unknowns [25, 1].

Deep neural networks, despite demonstrating strong capabilities in learning discriminative representations in closed scenarios, show accuracy degradation within open set settings [2]. As a naive strategy, modeling a thresh-

old for Softmax outputs has been demonstrated to be a sub-optimal solution for deep neural networks to identify unknowns. Thus the Extreme Value Theory was introduced to better adapt these discriminative models, fully based on supervised learning, for open-set settings. The underpinning idea is to calibrate Softmax scores so to estimate the probability of unknowns [32, 2]. In addition to deep discriminative models, deep generative models focusing on learning efficient latent feature representations by unsupervised leaning, have been widely utilized in open-set recognition tasks, and have gained successes one after the other [16, 19, 21, 28]. In particular, The Variational Auto-Encoder (VAE) is a typical probabilistic generative model ideal for detecting unknowns, due to its ability in learning low-dimensional representations in latent space not only supporting input reconstruction but also approximating a specified prior distribution. On the other hand, the VAE-based models may be not sufficiently effective for identifying known categories as all feature representations only follow one distribution. To this end, we employ a Conditional VAE (CVAE) that uses multiple prior distributions for modeling the known classes, and indirectly the unknown counterpart. Furthermore, we propose to represent the input samples with probabilistic capsules, given their already proved representation power capability [22, 24].

Capsule Networks (CapsNet) [24] were proposed as an alternative to Convolutional Neural Networks (CNNs). Unlike CNNs' scalar neurons, capsules ensemble a group of neurons to accept and output vectors. The vector of an activated capsule represents the various properties of a particular object, such as position, size, orientation, texture, etc. In essence, CapsNet can be viewed as an encoder encoding objects by distributed representations, which is exponentially more efficient than encoding them by activating a single neuron in a high-dimensional space. Besides, CapsNet has been successfully used to detect fake images and videos in a task setting similar to open set recognition [18]. This motivated us to design a novel capsule network architecture in combination with CVAE for the open set recognition problem, dubbed CVAECapOSR, that is depicted in Figure 1.

The contributions of this paper are three-fold: *i)* We present a novel open set recognition framework based on CapsNet and show its advantages for learning an efficient representation for known classes. *ii)* We integrate CapsNet and conditional VAEs. In contrast to general VAEs that encourage the latent representation to approximate a single prior distribution, our model exploits multiple priors (i.e. one for each class), and it forces the latent representation to follow the gaussian prior selected by the class of the input sample. *iii)* We conduct extensive experiments on all the standard datasets used for open set recognition, obtaining very competitive results, that in several cases outperform previous state of the art methods by a large margin.

2. Related Work

The open set recognition problem was introduced by [25] and was initially formalized as a constrained minimization problem based on Support Vector Machines (SVMs), whereas subsequent works focused on other more traditional approaches, such as Extreme Value Theory [9, 26], sparse representation [34], and Nearest Neighbors [10].

Following the success achieved by deep learning in many computer vision tasks, deep networks were first introduced for open set recognition in [2], in which it is proposed an Openmax function by calibrating the Softmax probability of each class with a Weibull distribution model. Subsequently, [5] extended Openmax to G-Openmax by introducing a generative adversarial network in which the generator produces synthetic samples of novel categories and the discriminator learns the explicit representation for unknown classes. A similar strategy has been adopted in [16], that presented a data augmentation technique based on generative adversarial networks, referred as counterfactual images generation. More recently, Yoshihashi *et al*. [32] analyzed and demonstrated the usefulness of training deep networks jointly for classification and reconstruction in the open set scenario. Specifically, the authors proposed to separate the knowns from the unknowns using the representations produced by the unsupervised training, while maintaining the discrimination capability of the model using the representations computed via the supervised learning process.

C2AE [19] introduced an architecture for open set recognition and unknown detection based on class conditioned VAEs by modelling the reconstruction error of the model based on the Extreme Value Theory. Sun *et al*. [28] have recently argued that one disadvantage of VAE-based architectures for open set recognition is the inadequate discriminative ability on instances of known classes. Therefore, the authors employed a conditional Gaussian distribution VAE model for learning conditional distributions of known classes and rejecting unknowns. A different approach is presented in [33], where normalizing flows are employed for density estimation of known samples. Specifically, the authors proposed an architecture that uses a CNN encoder and an invertible neural network that jointly learns the density of the input. However, a potential issue not discussed in the paper is that the CNN encoder has no bijective property, that is crucial to employ the change-of-variables formula for density evaluation. Additionally, [3] introduced the concept of reciprocal points in prototype leaning to manage the open space. Although this work shows excellent performances in rejecting unknowns coming from a different dataset with respect to the known samples, the unknown detection capability degrades when the source of unknown samples is the same of the known counterpart.

3. Preliminaries

3.1. The Open Set Recognition Problem

In the open set recognition problem the model has to classify test samples that can belong to classes not seen during training. Given a classification dataset $D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ such that $\boldsymbol{x}_i \in \mathcal{X}$ is an input sample, $y_i \in \{1, \ldots, K\}$ is the corresponding category label, the open set problem consists in the classification of test samples among $K + U$ classes, where the U are the number of unknown classes. In the literature, the dataset used for training is called *closed* dataset, meanwhile the one used during evaluation, that contains samples from unseen classes, is called *open* dataset. In order to quantify the openness of a dataset during the evaluation, following [25, 16] we consider the *openness* measure as $O = 1 - \sqrt{\frac{K}{M}}$ where K and $M = K + U$ are the number of classes observed during training and test, respectively.

3.2. Conditional VAE Formulation

The Conditional Variational Eutoencoder (CVAE) directly derives from the VAE model [11] and its objective is based on the estimation of the conditioned density $p(x|y)$ of the data x given the label y. It is one of the most powerful probabilistic generative models for its theory elegancy, strong framework compatibility and efficient manifold representations. The CVAEs commonly consist on an encoder that maps the input $\boldsymbol{x}$ and class y to a pre-fixed distribution over the latent variable $\boldsymbol{z}$, and on a decoder that, given a latent variable $\boldsymbol{z}$ and the class y tries to reconstruct the input $\boldsymbol{x}$. During training the model is trained by minimizing the negative variational lower bound of the conditional density of the data, defined as follows:

$$\mathcal{L}\big[\boldsymbol{x}, y; \theta, \phi\big] = D_{\text{KL}}\Big[q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}|y)\Big]$$
$$- \mathbb{E}_{q_\phi(\boldsymbol{z}|x)}\Big[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}, y)\Big]$$

where $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ denotes the posterior of the encoder, and $p_\theta(\boldsymbol{z}|y)$ indicates the prior distribution over the latent variable $\boldsymbol{z}$ conditioned on the class y. The first term in the loss function is a regularizer that enforces the approximate posterior distribution $q_\phi(\boldsymbol{x}|\boldsymbol{z})$ to be close to the prior distribution $p_\theta(\boldsymbol{z}|y)$, while the second term is the average reconstruction error of the chained encoding-decoding process. The original VAE [11], that uses an unconditioned prior distribution, presumes that $p_\theta(\boldsymbol{z})$ is an isotropic multivariate Gaussian $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is a general multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma^2})$. With these assumptions, the KL-divergence term given a K-dimensional $\boldsymbol{z}$, can be computed in closed form and expressed as:

$$D_{\text{KL}}\Big[q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z})\Big] = -\frac{1}{2} \sum_i^K (1 + \log(\boldsymbol{\sigma}_i^2) - \boldsymbol{\mu}_i^2 - \boldsymbol{\sigma}_i^2).$$

For the CVAE the KL-divergence term can be computed or estimated using only tractable latent prior distributions $p(\boldsymbol{z}|y)$ [20].

3.3. CapsNet Formulation

The capsule network, proposed by Hinton et al. [8], is a shallow architecture composed by two convolutional layers and two capsule layers. The first convolutional operation converts the pixel intensities of the input image $\boldsymbol{x}$ to primary local feature maps, while the second convolutional layer produces the primary capsules $\mathbf{u}_i$. Each capsule corresponds to a set of matrices that rotate the primary capsules for predicting the pose transformation $\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i$. Afterwards, the digit capsules $\mathbf{v}_j$, used for classification, are produced as the weighted sum of the primary capsules $\mathbf{v}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$, where the coefficients c_{ij} are determined by the dynamic routing algorithm (DR), in which the primary capsules are compared to the digit capsules. For t-th iteration of DR, the coefficients are updated by,

$$\mathbf{c}_i^{(t+1)} = \mathbf{Softmax}(\mathbf{b}_i^{(t+1)}), \ \ b_{ij}^{(t+1)} = b_{ij}^{(t)} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j^{(t)}.$$

For all layers of capsules, a squash function is used to introduce non-linearity and shrunk the length of capsule vectors into $[0, 1]$,

$$\mathbf{Squash}(\mathbf{v}) = \frac{\|\mathbf{v}\|^2}{1 + \|\mathbf{v}\|^2} \frac{\mathbf{v}}{\|\mathbf{v}\|}.$$

In this way the norm of the capsule stands for the probability of a particular feature being present in the input image $\boldsymbol{x}$.

4. Proposed Method

4.1. Model Architecture

Our model, depicted in Figure 2, is based on a CVAE with K different gaussian prior distributions, one for each known class. Given the input image $\boldsymbol{x}$ and its corresponding label y the encoder processes $\boldsymbol{x}$ producing the feature representation $\boldsymbol{x}_f$. Afterward the capsule network computes the distribution $q(\boldsymbol{z}|\boldsymbol{x})$ that is pushed toward the conditioned prior $p(\boldsymbol{z}|y) = \boldsymbol{T}_y$ during the learning process. Using the distance information between $q(\boldsymbol{z}|\boldsymbol{x})$ and all the targets we estimate the class $\hat{y}$, and using the reparametrization trick we sample $\boldsymbol{z}$ from $q(\boldsymbol{z}|\boldsymbol{x})$. Given $\hat{y}$ and $\boldsymbol{z}$ we compute the reconstruction $\hat{\boldsymbol{x}}$ through the decoder that is a convolutional neural network that uses transposed convolutions. After this general description of the computation of our model, we now present in deep the architecture step by step.

Encoding Stage. The blocks involved in the encoding stage are an encoder and a capsule network. The encoder is a convolutional neural network that processes the input image $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$ producing the feature $\boldsymbol{x}_f \in \mathbb{R}^{d_c \times d_h \times d_w}$. Then, similar to [24], the capsule network processes the
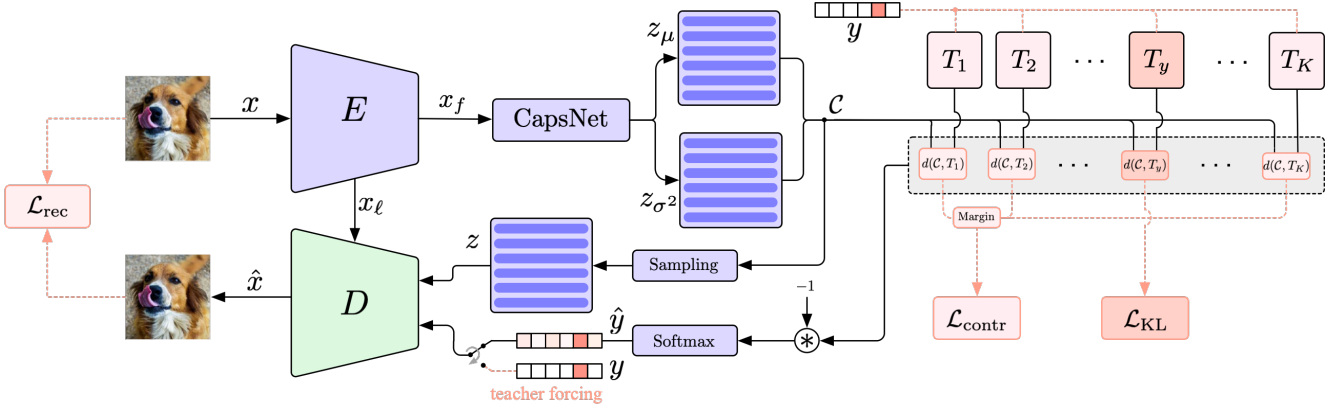
Figure 2: Outline of our CVAECapOSR model. The dashed orange lines stand for the computation of the model during training, whereas the solid black lines indicate the computation done by the model both during training and testing time.

feature representation $\boldsymbol{x}_f$ by computing the primary capsules $\boldsymbol{x}_{\text{pc}} \in \mathbb{R}^{f_1 \times (\frac{d_c}{f_1} d_h d_w)}$ and then the digit capsules $\boldsymbol{x}_{\text{dc}} \in \mathbb{R}^{K \times f_2}$ using the dynamic routing algorithm. We indicate with f_1 the dimension of the primary capsules and with f_2 the dimension of the digit capsules. Given the capsules $\boldsymbol{x}_{\text{dc}}$ we compute the mean $\boldsymbol{z}_\mu \in \mathbb{R}^{K \times d}$ and variance $\boldsymbol{z}_{\sigma^2} \in \mathbb{R}^{K \times d}$ of the capsules distribution $\mathcal{C} = q(\boldsymbol{z}|\boldsymbol{x})$ by applying a capsule-wise fully connected layer with d output units. In this way the probabilistic capsule network produces K mean capsules $\{\boldsymbol{z}_\mu^{(k)}\}_{k=1}^K$ and K variance capsules $\{\boldsymbol{z}_{\sigma^2}^{(k)}\}_{k=1}^K$, each of size d.

Contrastive Variational Stage. We design each CVAE prior target $p(\boldsymbol{z}|y = k)$ to be a gaussian distribution $\boldsymbol{T}_k = \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k\right)$ with learnable mean vector $\tilde{\boldsymbol{\mu}}_k \in \mathbb{R}^{Kd}$ and learnable diagonal covariance matrix $\tilde{\boldsymbol{\Sigma}}_k \in \mathbb{R}^{Kd \times Kd}$ with $1 \leq k \leq K$. In order to simplify the notation of our model framework we consider the targets $\boldsymbol{T}_k$ as gaussian distributions defined by $\boldsymbol{\mu}_k \in \mathbb{R}^{K \times d}$ being the reshaped version of the $\tilde{\boldsymbol{\mu}}_k$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{K \times d}$ being the reshaped diagonal of $\tilde{\boldsymbol{\Sigma}}_k$. In order to map input instances from the same class into compacted and separated regions of the latent space, during the learning process, we let the probabilistic capsule $\mathcal{C}$ to be attracted by the y-th target distribution $\boldsymbol{T}_y$ and at the same time we let all the other targets $\boldsymbol{T}_{\neq y}$ to be repulsed by $\mathcal{C}$. Using this contrastive strategy, we encourage the encoded representation to belong to the correct region of the latent space while maintaining all the targets sufficiently far apart to each other. We then estimate the class $\hat{y}$ of the input sample $\boldsymbol{x}$ as:

$$\hat{y}_k = p(y = k|\boldsymbol{x}) = \frac{e^{-\gamma d(\mathcal{C}, \boldsymbol{T}_k)}}{\sum_{j=1}^K e^{-\gamma d(\mathcal{C}, \boldsymbol{T}_j)}}, \quad 1 \leq k \leq K,$$

where

$$d(\mathcal{C}, \boldsymbol{T}_i) = \frac{1}{K} \sum_{k=1}^K D_{\text{KL}}\Big[\mathcal{C}^{(k)} || \boldsymbol{T}_i^{(k)}\Big],$$

is the distance between the probabilistic capsules $\mathcal{C}$ and the target $\boldsymbol{T}_i$, and γ is a coefficient parameter that controls the hardness of the probability assignment. In this way we estimate the probability of $\boldsymbol{x}$ being of class k considering the whole configuration of capsules $\{\mathcal{C}^{(k)}\}_{k=1}^K$ and not only the single most activated capsule as done in [24].

Decoding Stage. Given the class estimate $\hat{y} \in \mathbb{R}^K$ we compute its learnable embedding $\hat{y}_e = \text{Embedding}(\arg\max_k(\hat{y}_k)) \in \mathbb{R}^d$, and given the sampled latent capsules $\boldsymbol{z} \in \mathbb{R}^{K \times d}$ we compute the reconstruction $\hat{\boldsymbol{x}}$ through the decoder starting from $\boldsymbol{z}_y \in \mathbb{R}^{K \times d}$ with $\boldsymbol{z}_y^{(k)} = \hat{y}_e + \boldsymbol{z}^{(k)}$. The decoder is a convolutional neural network with transposed convolutions that follows a symmetrical structure of the encoder. Similar to [23], we implement lateral connections $\boldsymbol{x}_\ell$ with $1 \leq \ell \leq 4$ from the internal features of the encoder to the decoder that during training are randomly dropout for making the decoder less dependent from the internal representations of the encoder.

4.2. Training

We train the model on the closed dataset, and during the learning process for a single input sample $(\boldsymbol{x}, y)$ we minimize the following loss function:

$$\mathcal{L}(\boldsymbol{x}, y) = \mathcal{L}_{\text{KL}}(\boldsymbol{x}, y) + \alpha \mathcal{L}_{\text{contr}}(\boldsymbol{x}, y) + \beta \mathcal{L}_{\text{rec}}(\boldsymbol{x}), \quad (1)$$

where

$$\mathcal{L}_{\text{KL}}(\boldsymbol{x}, y) = d(\mathcal{C}, \text{sg}[\boldsymbol{T}_y]), \quad (2)$$

$$\mathcal{L}_{\text{contr}}(\boldsymbol{x}, y) = \frac{1}{K-1} \sum_{k \neq y}^K \big[m_k - d(\text{sg}[\mathcal{C}], \boldsymbol{T}_k)\big]^+, \quad (3)$$

$$\mathcal{L}_{\text{rec}}(\boldsymbol{x}) = \left\| \hat{\boldsymbol{x}} - \boldsymbol{x} \right\|_2^2. \tag{4}$$

As already defined in [30], the function sg[·] in Eq. (2) and Eq. (3) stands for the stop-gradient operator that is defined as the identity at forward computation time and has zero partial derivatives, constraining its argument to be a non-updated constant. The loss term in Eq. (2) is responsible for pushing the probabilistic capsules $\mathcal{C}$ toward the target $\boldsymbol{T}_y$, leading to the concentration of all the density of known samples in the targets region. On the other hand, the contrastive loss term in Eq. (3) pushes all the targets not related to y far away from the distribution $\mathcal{C}$ using a margin loss with margin m_k where $[\cdot]^+$ is the function that returns the positive part of its argument. By considering $\boldsymbol{T}_{\neq y}$ to be the otherness of $\boldsymbol{T}_y$, therefore of $\mathcal{C}$, the contrastive term not only avoid the collapse of the prior targets, but also encourages the separation between one class and *all the other* classes, potentially the unknown counterpart. Finally, the loss term in Eq. (4) is the mean squared error reconstruction between the input and the output of the model. We aggregate all loss terms in Eq. (1) and we control the strength of $\mathcal{L}_{\text{contr}}$ with a parameter α and the strength of $\mathcal{L}_{\text{rec}}$ with a parameter β. During training we found beneficial to use teacher forcing in the decoder i.e. we decided to feed y instead of the estimate $\hat{y}$ to the decoder, while during validation and testing we feed only the estimated quantity, enabling in this way, the independence of our model respect to the label y during inference.

4.3. Inference

We use the model's natural rejection rule based on the probabilistic distance between samples and targets to detect unknowns, and directly classify known samples with the minimum probabilistic distance over than a given threshold.

Given a new sample $\boldsymbol{x}$ we decide if it is an outlier as follows:

$$\hat{y} = \begin{cases} K+1, & \text{if } \max_k \left\{ d(\mathcal{C}, \boldsymbol{T}_k) \right\} = d^* < \tau \\ \arg\max_k \left\{ d(\mathcal{C}, \boldsymbol{T}_k) \right\}, & \text{otherwise.} \end{cases}$$

where τ is found using cross validation, and $K+1$ is the new, unknown class not seen during training.

5. Experiments

Recent works in this area followed the protocol presented in [16]. In that work, an open set recognition scenario is obtained by randomly selecting K classes from a specific dataset as known (see below for more details), while the remaining classes are considered to be open set classes. This procedure is applied to five random splits. However, as recently shown by [21], performance across different splits varies significantly (e.g. AUROC on CIFAR10 varied between 77% to 87% across different splits),

and there are serious reproducibility issues. Moreover, not only the splits have a large influence on the results, but also the strategy used to select the samples belonging to the unknown classes. Therefore, starting from the splits used in [16] and following [21], we publicly release our code and data[1], as well as the implementation of other state-of-the-art methods, to foster a fair comparison on this task.

5.1. Datasets

We evaluate open set recognition performance on the standard datasets used in previous works, i.e. MNIST [15], SVHN [17], CIFAR10 [12], CIFAR+10, CIFAR+50 and TinyImageNet [14].

MNIST, SVHN, CIFAR10. All three datasets contain ten categories. MNIST consists of hand-written digit images, and it has $60,000$ 28×28 grayscale images for training and $10,000$ for testing. SVHN contains street view house numbers, consisting of ten digit classes each with between $9,981$ and $11,379$ 32×32 color images. Then we consider the CIFAR10 dataset, which has $50,000$ 32×32 color images for training and $10,000$ for testing. Following [16], in the unknown detection task each dataset is randomly partitioned into 6 known classes and 4 unknown classes. In this setting, the openness score is fixed to 22.54%.

CIFAR+10, CIFAR+50. To test our model in a setting of higher openness values, we perform CIFAR+Q experiments using CIFAR10 and CIFAR100 [12]. To this end, 4 known classes are sampled from CIFAR10 and Q unknown classes are drawn randomly from the more diverse and larger CIFAR100 dataset. Openness scores of CIFAR+10 and CIFAR+50 are 46.54% and 72.78%, respectively.

TinyImageNet. For TinyImagenet dataset, which is a subset of ImageNet that contains 200 classes, we randomly sampled 20 classes as known and the remaining classes as unknown. In this setting, the openness score is 68.37%.

5.2. Metrics

Open set classification performance is usually measured using F-score and AUROC (Area Under ROC Curve) [6]. F-score is used to measure the in-distribution classification performance, while AUROC is commonly reported by both open set recognition and out-of-distribution detection literature. AUROC provides a calibration free measure and characterizes the performance for a given score by varying the discrimination threshold [4]. In our experiments, we use macro averaged F1-score on the open set recognition task, and the AUROC for the unknown detection task. For both metrics, higher values are better.

---

[1]Code and data publicly available on https://github.com/guglielmocamporese/cvaecaposr.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50	TinyImageNet
Softmax † [28]	0.978	0.886	0.677	0.816	0.805	0.577
Openmax † [2]	0.981	0.894	0.695	0.817	0.796	0.576
G-Openmax † [5]	0.984	0.896	0.675	0.827	0.819	0.580
OSRCI † [16]	0.988 ± 0.004	0.91 ± 0.01	0.699 ± 0.038	0.838	0.827	0.586
CROSR [32]	0.991 ± 0.004	0.899 ± 0.018	-	-	-	0.589
C2AE ‡ [19]	-	0.892 ± 0.013	0.711 ± 0.008	0.810 ± 0.005	0.803 ± 0.000	0.581 ± 0.019
GFROR ‡ [21]	-	0.955 ± 0.018	0.831 ± 0.039	-	-	0.657 ± 0.012
CGDL § [28]	0.977 ± 0.008	0.896 ± 0.023	0.681 ± 0.029	0.794 ± 0.013	0.794 ± 0.003	0.653 ± 0.002
RPL § [3]	0.917 ± 0.006	0.931 ± 0.014	0.784 ± 0.025	0.885 ± 0.019	0.881 ± 0.014	0.711 ± 0.026
CVAECapOSR (ours)	**0.992** ± 0.004	**0.956** ± 0.012	**0.835** ± 0.023	**0.888** ± 0.019	**0.889** ± 0.017	**0.715** ± 0.018

Table 1: AUROC scores on the detection of known and unknown samples. Results are averaged over 5 different splits of known and unknown classes partitions. As discussed in Section 5.3, we report the results on the same data splits and, for the sake of clarity, we highlight the source of the results used to populate the table: † are provided by [16], ‡ is from [21] and § are the results that we obtained by running the code of the original paper.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50	TinyImageNet
CVAECapOSR fixed Targets	**0.997** ± 0.006	0.953 ± 0.022	0.823 $\pm \pm 0.012$	0.868 ± 0.018	0.829 ± 0.009	0.706 ± 0.014
CVAECapOSR learn Targets	0.992 ± 0.004	**0.956** ± 0.012	**0.835** $\pm \pm 0.023$	**0.888** ± 0.019	**0.889** ± 0.017	**0.715** ± 0.018

Table 2: AUROC scores on the detection of known and unknown samples comparing our model that uses fixed targets (first row) versus our model that learns the targets (second row) during the learning process. Results are averaged over 5 splits.

5.3. Experimental Results

Following [28], we conducted two major experiments in which our model has to solve the *unknown detection* task and the *open set recognition* task. For all the experiments we use ResNet34 [7] as the encoder backbone of our model.

Unknown Detection. In the unknown detection problem the model is trained on a subset of the dataset using K classes, and the evaluation is done by measuring the model capability on detecting unknown classes, not seen during training. The evaluation is performed by considering the binary recognition task of the known *vs* unknown classes, and performances are reported in terms of AUROC scores. The results, shown in Table 1, are averaged over five random splits of known and unknown classes, provided by [16].

As already discussed (and recently shown in [6, 21]), performance across different splits varies significantly. For this reason we use the exact data splits provided by [16] that have been used also in other recent works [32, 21]. Nevertheless, not all the results reported in these works are directly comparable; although the splits are the same, [21] followed a particular strategy in selecting the open set classes in CIFAR+10 and CIFAR+50 experiments (i.e. they selected 10 and 50 samples from vehicle classes instead of purely random classes, which has gained a large impact on these results). Therefore, following [21], we have run the code of [28] and [3] (whereas the results of [19] are provided by [21], since the code is no more available), and we compare all the results with the state of the art papers that

have the same splits and use the same evaluation setting, and that can be reproduced. As shown in Table 1, we obtain state of the art results, outperforming all previous methods, on all the datasets. Moreover, as also previously reported, we will release all data and code to guarantee reproducibility.

One important fact we observed during the training process is the boost we obtained by letting the targets distributions $\boldsymbol{T}_k$ to be learned and not to be used as fixed priors, as shown in Table 2. We initialized the learnable targets with $\boldsymbol{\mu}_k^{(i)} = \mathbb{1}_d \cdot \delta_{k=i}$ and $\boldsymbol{\Sigma}_k^{(i)} = \mathbb{1}_d$. We noticed that learning the targets without considering the contrastive term in the loss function ($\alpha = 0$) caused the collapse of the targets into one single distribution, leading to poor results. We thus consider the contrastive term, and we set $\alpha = 1.0$, $\beta = 0.05$ and $m_k = 10$.

Open Set Recognition. In the open set recognition problem, the model is trained on the closed dataset that contains K classes, and it is evaluated on the open dataset considering $K + 1$ classes. In this experimental setting, we evaluate the model using the macro F1-score on the $K + 1$ classes. In the first experiment for open set recognition, we train on all the classes of the MNIST dataset and we then evaluate the performances by including new datasets in the open set. Similarly to [32], we used Omniglot, MNIST-Noise, and Noise that are datasets of grayscale images. Each of this dataset contains $10,000$ test images, the same as MNIST. The Omniglot dataset contains

Method	Omniglot	MNIST-noise	Noise
Softmax [28]	0.595	0.801	0.829
Openmax [5]	0.780	0.816	0.826
CROSR [32]	0.793	0.827	0.826
CGDL [28]	0.850	0.887	0.859
CVAECapOSR (ours)	**0.971**	**0.982**	**0.982**

Table 3: Results for the open set recognition on the MNIST dataset. We report the macro-averaged F1-score for 11 classes (10 from the test partition of the MNIST, and 1 from the test of another dataset).

AUROC scores w.r.t. different Openness Values (O)				
Openness Variation:	$O = 0\%$	$O = 15.98\%$	$O = 30.72\%$	$O = 39.69\%$
Influence of the Feature Extractor (before CapsNet)				
CapsNet	0.971	0.753	0.767	0.781
ResNet20 + CapsNet	**0.981**	**0.948**	**0.949**	**0.950**
Improvement	+0.020	+0.195	+0.182	+0.169
Influence of CapsNet				
ResNet20 + FC	0.975	0.581	0.595	0.606
ResNet20 + CapsNet	**0.981**	**0.948**	**0.949**	**0.950**
Improvement	+0.006	+0.367	+0.354	+0.344
Influence of Dynamic Routing				
ResNet20 + CapsNet	0.981	0.948	0.949	0.950
ResNet20 + CapsNet + DR	**0.982**	**0.952**	**0.954**	**0.955**
Improvement	+0.001	+0.004	+0.005	+0.005

Table 4: Ablation study on the model architecture. We report results for the unknown detection task on SVHN dataset with outliers from CIFAR100. The performance is evaluated by AUROC with different openness values.

hand-written characters from the alphabets of many languages, while the Noise dataset has images synthesized by randomly sampling each pixel value independently from a uniform distribution on $[0, 1]$. MNIST-Noise is also a synthesized set, constructed by superimposing MNIST's test images on Noise. The results of the open set recognition on these datasets are shown in Table 3. On each dataset, we outperform state of the art results by a large margin. On Omniglot, we improve the F1-score by $+0.121$, in the MNIST-Noise by $+0.095$, and in the Noise by $+0.123$.

In the second experiment of open set recognition, following the same protocol used in [16], all samples from the 10 classes in CIFAR10 dataset are considered as known data, and samples from ImageNet and LSUN are selected as unknown samples. In order to have the same image size as known samples, we resized or croped the unknown samples, obtaining the following datasets: ImageNet-crop, ImageNet-resize, LSUN-crop, and LSUN-resize. For each dataset, we consider all their $10,000$ test samples as the unknown samples in the open set. The performance of the method is evaluated using the macro-averaged F1-scores in the 11 classes (10 known classes and 1 unknown), and the results are shown in Table 5. We can see that our method outperforms all previous methods under the F1-score on

ImageNet-resize, ImageNet-crop, LSUN-crop, and LSUN-resize.

Ablation Study on the Model Architecture. In order to verify the contribution of each part of our model, we perform ablation on the relevance of the main model's components: the capsule network CapsNet, and the feature extractor. ResNet20 is selected as the feature extractor for getting shorter training times. We investigate also the impact of different components of CapsNet, in order to understand their importance. We consider four different variations of our model architecture: the model with CapsNet and dynamic routing that does not use the ResNet20 feature extractor, but just a single convolutional layer; ResNet20+CapsNet that includes the residual feature extractor before CapsNet and doesn't use dynamic routing; ResNet20+FC where a fully connected layer replaces CapsNet; and ResNet20+CapsNet+DR that implements dynamic routing in CapsNet. For all CapsNets that do not implement the dynamic routing, we process each capsule by a fully connected layer. For the ablation, we consider the entire SVHN dataset as the closed dataset, and we consider unknown samples from the CIFAR100 for the open dataset. We then consider different numbers of unknown classes of CIFAR100, leading to different openness values O. The results of the ablation analysis are reported in Table 4. We can see that the residual network used as a feature extractor in the encoder helps, especially when the openness O of the open set increases. This fact highlights the importance of having already pre-processed features for the CapsNet on the unknown detection problem when openness increases. Furthermore, another emerging fact is the higher representation capability of the CapsNet with respect to a FC layer: as the openness increases, the AUROC improvement increases up to $+0.344$. This result suggests that capsules are more capable in detecting unknown samples with respect to standard artificial neurons. This fact emerges also from the t-SNE [31] visualization of the latent space, reported in Figure 3, where the separation between known and unknown produced by the probabilistic capsules is more evident with respect to the one produced by a standard FC. Finally, from the experiments we see that dynamic routing achieves better performances with respect to not using it, and that the largest boost on using the capsule network is given by the pose transformation.

Implementation details. We also investigated the importance of the parameters α, β, m_k and γ. To this end, we conducted the unknown detection experiment with the ResNet20+CapsNet+DR architecture on SVHN, using CIFAR100 as the open dataset with openness $\mathcal{O} = 30.72\%$. As suggested by the results reported in Table 6, we set $\alpha = 1.0$ and $m_k = 10.0$ and, finally, we set $\beta = 0.05$ and $\gamma = 1$ empirically.

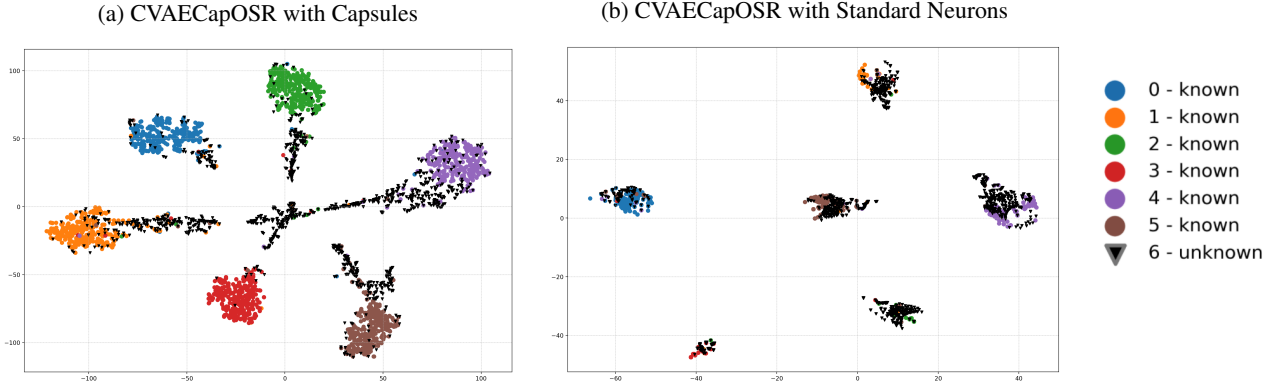(a) CVAECapOSR with Capsules   (b) CVAECapOSR with Standard Neurons

Figure 3: t-SNE latent space visualizations obtained with different components (i.e., (a) Capsules, (b) Standard Neurons) of the SVHN test set including the CIFAR+100 test set from which unknown samples are sampled. In particular, we use an openness value of $O = 35.67\%$. In both pictures, the unknown samples are represented by black triangles.

Method	ImageNet-crop	ImageNet-resize	LSUN-crop	LSUN-resize
Softmax † [28]	0.639	0.653	0.642	0.647
Openmax † [2]	0.660	0.684	0.657	0.668
CROSR [32]	0.721	0.735	0.720	0.749
C2AE ‡ [19]	0.837	0.826	0.783	0.801
CGDL § [28]	0.840	0.832	0.806	0.812
RPL § [3]	0.811	0.810	0.846	0.820
CVAECapOSR (ours)	**0.857**	**0.834**	**0.868**	**0.882**

Table 5: Open set recognition results on CIFAR-10 with various outliers added to the test set as unknowns. We evaluate the model using macro-averaged F1-scores on 11 classes (10 from the the test of the CIFAR10, and 1 from various test datasets). For the sake of clarity, we highlight the source of the results used to populate the table: † are provided by [16], ‡ is from [21] and § are the results that we obtained by running the code of the original paper

Contr. Params	$m_k = 5.0$	$m_k = 10.0$	$m_k = 20.0$
$\alpha = 0.5$	0.527	0.564	0.947
$\alpha = 1.0$	0.937	0.954	0.949
$\alpha = 2.0$	0.944	0.951	0.945

Table 6: AUROC scores for different values of the parameters α, m_k in the loss function. Red cells indicate that targets during the learning process overlap at some point, leading to poor results. Green cells indicate no collapse of prior targets, suggesting good values for α, and m_k.

6. Conclusion

In this paper, we introduced CVAECapOSR, a model for open set recognition based on CVAE that produces probabilistic capsules as latent representations through the capsule network. We extended the standard framework of CVAEs using multiple gaussian prior distributions rather than just one for all known classes in the closed dataset. Furthermore, targets are set to be learnable in order to cluster knowns inside their target regions. The contrastive term is used to model the otherness for known classes and to keep the target regions to be mutually separated. Experimental results, obtained on several datasets, show the effectiveness and the high performances on unknown detection and open set recognition tasks.

References

[1] A. Bendale and T. E. Boult. Towards open world recognition. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902, 2015. 1

[2] A. Bendale and T. E. Boult. Towards open set deep networks. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 6, 8

[3] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian. Learning open set network with discriminative reciprocal points. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 507–522, 2020. 2, 6, 8

[4] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proc. of International Conference on Machine Learning (ICML)*, 2006. 5

[5] Z. Ge, S. Demyanov, and R. Garnavi. Generative openmax for multi-class open set classification. In *Proc. of British Machine Vision Conference (BMVC)*, 2017. 2, 6, 7

[6] C. Geng, S.-J. Huang, and S. Chen. Recent advances in open set recognition: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, in press, 2020. 5, 6

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 6

[8] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *Proc. of Int'l Conference on Artificial Neural Networks (ICANN)*, 2011. 3

[9] L. P. Jain, W. J. Scheirer, and T. E. Boult. Multi-class open set recognition using probability of inclusion. In *Proc. of European Conference on Computer Vision (ECCV)*, 2014. 2

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

[11] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proc. of International Conference on Learning Representations (ICLR)*, 2014. 3

[12] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 1

[14] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. 2015. 5

[15] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. 5

[16] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li. Open set learning with counterfactual images. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 5, 6, 7, 8

[17] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[18] H. H. Nguyen, J. Yamagishi, and I. Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *Proc. of IEEE Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. 2

[19] P. Oza and V. M. Patel. C2AE: Class conditioned autoencoder for open-set recognition. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2302–2311, 2019. 2, 6, 8

[20] A. Pagnoni, K. Liu, and S. Li. Conditional variational autoencoder for neural machine translation. *ArXiv*, abs/1812.04405, 2018. 3

[21] P. Perera, V. I. Morariu, R. Jain, V. Manjunatha, C. Wigington, V. Ordonez, and V. M. Patel. Generative-discriminative feature representations for open-set recognition. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11811–11820, 2020. 2, 5, 6, 8

[22] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo. Deepcaps: Going deeper with capsule networks. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[23] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. of International Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. 4

[24] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 3, 4

[25] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7):1757–1772, 2013. 1, 2, 3

[26] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(11):2317–2324, 2014. 2

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of International Conference on Learning Representations (ICLR)*, pages 1–14, 2015. 1

[28] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng. Conditional gaussian distribution learning for open set recognition. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6, 7, 8

[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014. 1

[30] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5

[31] L. Van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 7

[32] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura. Classification-reconstruction learning for open-set recognition. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6, 7, 8

[33] H. Zhang, A. Li, J. Guo, and Y. Guo. Hybrid models for open set recognition. In *Proc. of European Conference on Computer Vision (ECCV)*, 2020. 2

[34] H. Zhang and V. M. Patel. Sparse representation-based open set recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(8):1690–1696, 2017. 2