C2AE: Class Conditioned Auto-Encoder for Open-set Recognition

Poojan Oza and Vishal M. Patel
Department of Electrical and Computer Engineering
Johns Hopkins University, 3400 N. Charles St, Baltimore, MD 21218, USA
poza2@jhu.edu, vpatel36@jhu.edu

Abstract

Models trained for classification often assume that all testing classes are known while training. As a result, when presented with an unknown class during testing, such closed-set assumption forces the model to classify it as one of the known classes. However, in a real world scenario, classification models are likely to encounter such examples. Hence, identifying those examples as unknown becomes critical to model performance. A potential solution to overcome this problem lies in a class of learning problems known as open-set recognition. It refers to the problem of identifying the unknown classes during testing, while maintaining performance on the known classes. In this paper, we propose an open-set recognition algorithm using class conditioned auto-encoders with novel training and testing methodologies. In this method, training procedure is divided in two sub-tasks, 1. closed-set classification and, 2. open-set identification (i.e. identifying a class as known or unknown). Encoder learns the first task following the closed-set classification training pipeline, whereas decoder learns the second task by reconstructing conditioned on class identity. Furthermore, we model reconstruction errors using the Extreme Value Theory of statistical modeling to find the threshold for identifying known/unknown class samples. Experiments performed on multiple image classification datasets show that the proposed method performs significantly better than the state of the art methods. The source code is available at: github.com/otkupjnoz/c2ae.

1. Introduction

Recent advancements in computer vision have resulted in significant improvements for tasks such as image classification [16], [24], [17], [48], Detection [41], [40], [4], [49], Clustering [19], [1], [8], [2], etc. Specifically, for classification task the rise of Deep Convolutional Neural Network has resulted in error rates surpassing the human-level performance [15]. These promising results, enable their potential use in many real world applications. However, when

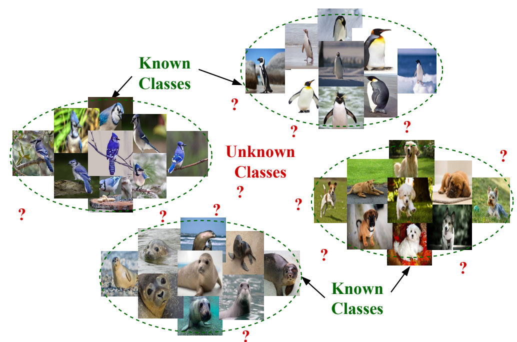


Figure 1: Open-set recognition problem: Data samples from Blue Jay, Seal, Dog and Penguin are from the known class set ($\mathcal{K}$). Also, many classes not known during training, will be present at testing, *i.e.*, samples from unknown class set ($\mathcal{U}$). The goal is to correctly classify any sample coming from set $\mathcal{K}$, as either Blue Jay, Seal, Dog or Penguin and identify samples coming from $\mathcal{U}$ as unknown.

deployed in a real world scenario, such systems are likely to observe samples from classes not seen during training (*i.e.* unknown classes also referred as "*unknown unknowns*" [44]). Since, the traditional training methods follow this closed-set assumption, the classification systems observing any unknown class samples are forced to recognize it as one of the known classes. As a result, it affects the performance of these systems, as evidenced by Jain *et al.* [18] with digit recognition example. Hence, it becomes critical to correctly identify test samples as either known or unknown for a classification model. This problem setting of identifying test samples as known/unknown and simultaneously correctly classifying all of known classes, is referred to as open-set recognition [44]. Fig. 1 illustrates a typical example of classification in the open-set problem setting.

In an open-set problem setting, it becomes challenging to identify unknown samples due to the incomplete knowledge of the world during training (*i.e.* only the known classes are accessible). To overcome this problem many open-set methods in the literature [7], [45], [50], [47] adopt recognition score based thresholding models. However, when using these models one needs to deal with two key questions, *1) what is a good score for open-set identification?* (*i.e.*, identifying a class as known or unknown), and given a score, *2) what is a good operating threshold for the model?*. There have been many methods that explore these questions in the context of traditional methods such as Support Vector Machines [44], [45], [18], Nearest Neighbors [21], [6] and Sparse Representation [50]. However, these questions are relatively unexplored in the context of deep neural networks.

Even-though deep neural networks are powerful in learning highly discriminative representations, they still suffer from performance degradation in the open-set setting [7]. In a naive approach, one could apply a thresholding model on SoftMax scores. However, as shown by experiments in [7], that model is sub-optimal for open-set identification. A few methods have been proposed to better adapt the SoftMax scores for open-set setting. Bendale *et al.* proposed a calibration strategy to update SoftMax scores using extreme value modeling [7]. Other strategies, Ge *et al.* [11] and Lawrence *et al.* [29] follow data augmentation technique using Generative Adversarial Networks (GANs) [13]. GANs are used to synthesize open-set samples and later used to fine-tuning to adapt SoftMax/OpenMax scores for open-set setting. Shu *et al.* [47] introduced a novel sigmoid-based loss function for training the neural network to get better scores for open-set identification.

All of these methods modify the SoftMax scores, so that it can perform both open-set identification and maintain its classification accuracy. However, it is extremely challenging to find a single score measure, that can perform both. In contrast to these methods, in the proposed approach the training procedure for open-set recognition using class conditional auto-encoders, is divided it into two sub-tasks, 1. closed-set classification, and 2. open-set identification. These sub-tasks are trained separately in a stage-wise manner. Experiments show that such approach provides good open-set identification scores and it is possible to find a good operating threshold using the proposed training and testing strategy.

In summary, this paper makes following contributions,

- A novel method for open-set recognition is proposed with novel training and testing algorithm based on class conditioned auto-encoders.
- We show that dividing open-set problem in sub-tasks can help learn better open-set identification scores.
- Extensive experiments are conducted on various image classification datasets and comparisons are performed against several recent state-of-the-art approaches. Furthermore, we analyze the effectiveness of the proposed method through ablation experiments.

2. Related Work

Open-set Recognition. The open-set recognition methods can be broadly classified in to two categories, traditional methods and neural network-based methods. Traditional methods are based on classification models such as Support Vector Machines (SVMs), Nearest Neighbors, Sparse Representation etc. Scheirer *et al.* [45] extended the SVM for open-set recognition by calibrating the decision scores using the extreme value distribution. Specifically, Scheirer *et al.* [45] utilized two SVM models, one for identifying a sample as unknown (referred as CAP models) and other for traditional closed-set classification. PRM Junior *et al.* [20] proposed a nearest neighbor-based open-set recognition model utilizing the neighbor similarity as a score for open-set identification. PRM Junior *et al.* later also presented specialized SVM by constraining the bias term to be negative. This strategy in the case of Radial Basis Function kernel, yields an open-set recognition model. Zhang and Patel [50] proposed an extension of the Sparse Representation-based Classification (SRC) algorithm for open-set recognition. Specifically, they model residuals from SRC using the Generalized-Pareto extreme value distribution to get score for open-set identification.

In neural network-based methods, one of the earliest works by Bendale *et al.* [7] introduced an open-set recognition model based on "activation vectors" (*i.e.* penultimate layer of the network). Bendale *et al.* utilized meta-recognition for multi-class classification by modeling the distance from "mean activation vector" using the extreme value distribution. SoftMax scores are calibrated using these models for each class. These updated scores, termed as OpenMax, are then used for open-set identification. Ge *et al.* [11] introduced a data augmentation approach called G-OpenMax. They generate unknown samples from the known class training data using GANs and use it to fine-tune the closed-set classification model. This helps in improving the performance for both SoftMax and OpenMax based deep network. Along the similar motivation, Neal *et al.* [29] proposed a data augmentation strategy called *counterfacutal image generation*. This strategy also utilizes GANs to generate images that resemble known class images but belong to unknown classes. In another approach, Shu *et al.* [47] proposed a k-sigmoid activation-based novel loss function to train the neural network. Additionally, they perform score analysis on the final layer activations to find an operating threshold, which is helpful for open-set identification. There are some related problems such as anomaly detection [31], [32], [36] and novelty detection [34], [39], [33], [43] etc., which are relaxed version of the open-set recognition formulation. But, for this paper we only focus on the open-set recognition problem.

Extreme Value Theory. Extreme value modeling is a branch of statistics that deals with modeling of statistical

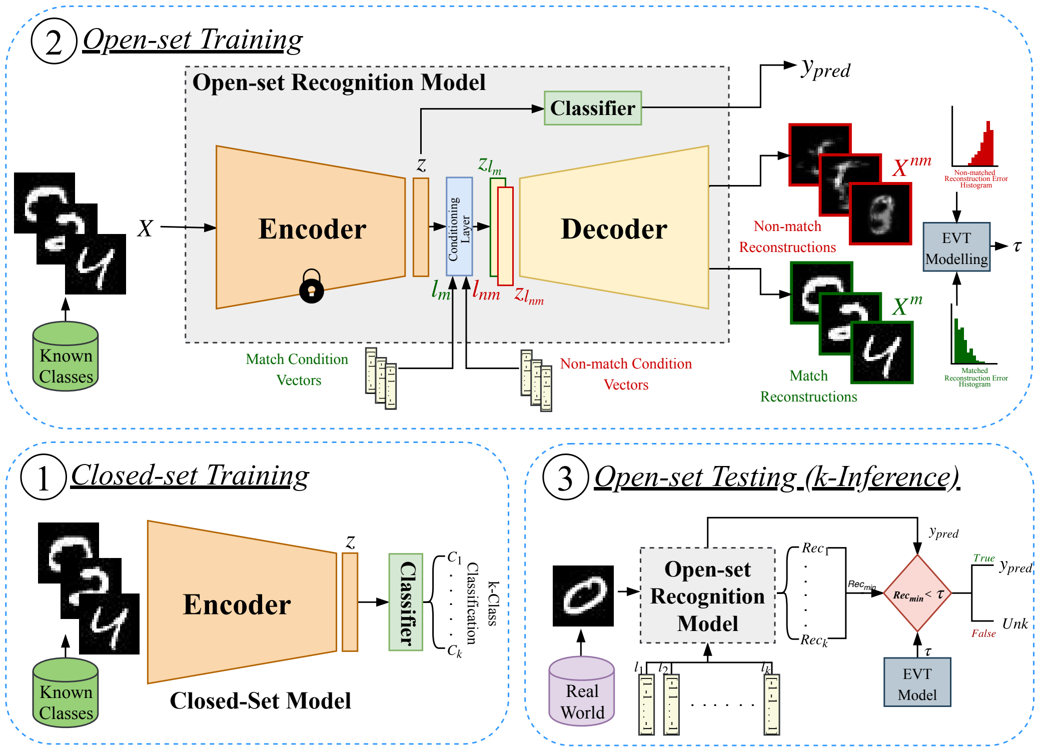


Figure 2: Block diagram of the proposed method: **1) Closed-set training**, Encoder ($\mathcal{F}$) and Classifier ($\mathcal{C}$) are trained with the traditional classification loss. **2) Open-set Training**, To train an open-set identification model, auto-encoder network Encoder ($\mathcal{F}$) with frozen weights, and Decoder ($\mathcal{G}$), are trained to perfectly or poorly reconstruct the images depending on the label condition vector. Reconstruction errors are then modeled using the extreme value distribution to find the operating threshold of the method. **3) Open-set Testing**, Open-set recognition model produces the classification prediction (y_{pred}) and k reconstruction errors, conditioned with each condition vector. If the minimum reconstruction error is below the threshold value obtained from the EVT model, the test sample is classified as one of the k classes, or else it is classified as unknown.

extremes. The use of extreme value theory in vision tasks largely deals with post recognition score analysis [35], [45]. Often for a given recognition model the threshold to reject/accept lies in the overlap region of extremes of match and non-match score distributions [46]. In such cases, it makes sense to model the tail of the match and non-match recognition scores as one of the extreme value distributions. Hence, many visual recognition methods including some described above, utilize extreme value models to improve the performance further [50], [45].

3. Proposed Method

The proposed approach divides the open-set recognition problem into two sub-tasks, namely, closed-set classification and open-set identification. The training procedure for these tasks are shown in Fig. 2 as stage-1 and stage-2. Stage-3 in Fig. 2 provides overview of the proposed approach at inference. In what follows, we present details of these stages.

3.1. Closed-set Training (Stage 1)

Given images in a batch $\{X_1, X_2, ..., X_N\} \in \mathcal{K}$, and their corresponding labels $\{y_1, y_2, ..., y_N\}$. Here, N is the batch size and $\forall y_i \in \{1, 2, .., k\}$. The encoder ($\mathcal{F}$) and the classifier ($\mathcal{C}$) with parameters Θ_f and Θ_c, respectively are trained using the following cross entropy loss,

$$\mathcal{L}_c(\{\Theta_f, \Theta_c\}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} \mathbb{I}_{y_i}(j) \log[p_{y_i}(j)], \quad (1)$$

where, $\mathbb{I}_{y_i}$ is an indicator function for label y_i (*i.e.*, one hot encoded vector) and $p_{y_i} = \mathcal{C}(\mathcal{F}(X_i))$ is a predicted probability score vector. $p_{y_i}(j)$ is probability of the i^{th} sample being from the j^{th} class.

3.2. Open-set Training (Stage 2)

There are two major parts in open-set training, conditional decoder training, followed by EVT modeling of the reconstruction errors. In this stage, the encoder and classifier weights are fixed and don't change during optimization.

3.2.1 Conditional Decoder Training

For any batch described in Sec. 3.1, $\mathcal{F}$ is used to extract the latent vectors as, $\{z_1, z_2, ..., z_N\}$. This latent vector batch is conditioned using the work by Perez *et al.* [37] called FiLM. FiLM influences the input feature map by applying a feature-wise linear modulations based on conditioning information. For an input feature z and vector l_j containing conditioning information we get following equations,

$$\gamma_j = H_\gamma(l_j), \quad \beta_j = H_\beta(l_j), \tag{2}$$

$$z_{l_j} = \gamma_j \odot z + \beta_j, \tag{3}$$

where,

$$l_j(x) = \begin{cases} +1, x = j, \\ -1, x \neq j, \end{cases} \quad x, j \ \in \{1, 2, ..., k\}.$$

Here, H_γ and H_β are neural networks with parameters Θ_γ and Θ_β. Tensors z_{l_j}, γ_j, β_j have the same shape and $\odot$ represents the Hadamard product. l_j is used for conditioning, and referred to as label condition vector in the paper. Also, the notation z_{l_j} is used to describe the latent vector z conditioned on the label condition vector l_j, i.e, $z|l_j$.

The decoder ($\mathcal{G}$ with parameters Θ_g) is expected to perfectly reconstruct the original input when conditioned on the label condition vector matching the class identity of the input, referred here as the match condition vector (l_m). However, here $\mathcal{G}$ is additionally trained to poorly reconstruct the original input when conditioned on the label condition vector, that does not match the class identity of the input, referred here as the non-match condition vector (l_{nm}). The importance of this additional constraint on the decoder is discussed in Sec. 3.2.3 while modeling the reconstruction errors using EVT. For the rest of this paper, we use superscript m and nm to indicate match and non-match, respectively.

For a given input X_i from the batch and $l_m = l_{y_i^m}$ and $l_{nm} = l_{y_j^{nm}}$, for any random $y_i^{nm} \neq y_i^m$ sampled from $\{1, 2, .., k\}$, be its corresponding match and non-match condition vectors, the feed forward path for stage-2 can be summarized through the following equations,

$$\begin{aligned} z_i &= \mathcal{F}(X_i), & & \\ \gamma_{y_i^m} &= H_\gamma(l_{y_i^m}), & \gamma_{y_i^{nm}} &= H_\gamma(l_{y_i^{nm}}), \\ \beta_{y_i^m} &= H_\beta(l_{y_i^m}), & \beta_{y_i^{nm}} &= H_\beta(l_{y_i^{nm}}), \\ z_{il_m} &= \gamma_{y_i^m} \odot z_i + \beta_{y_i^m}, & z_{il_{nm}} &= \gamma_{y_i^{nm}} \odot z_i + \beta_{y_i^{nm}}, \\ \tilde{X}_i^m &= \mathcal{G}(z_{l_m}). & \tilde{X}_i^{nm} &= \mathcal{G}(z_{l_{nm}}). \end{aligned}$$

Following the above feed-forward path, the loss functions in the second stage of training to train the decoder ($\mathcal{G}$ with parameters Θ_g) and conditioning layer (with parameters Θ_γ and Θ_β) are given as follows,

$$\mathcal{L}_r^m(\{\Theta_g, \Theta_\gamma, \Theta_\beta\}) = \frac{1}{N}\sum_{i=1}^{N} ||X_i - \tilde{X}_i^m||_1, \tag{4}$$

$$\mathcal{L}_r^{nm}(\{\Theta_g, \Theta_\gamma, \Theta_\beta\}) = \frac{1}{N}\sum_{i=1}^{N} ||X_i^{nm} - \tilde{X}_i^{nm}||_1, \tag{5}$$

$$\begin{aligned} \min_{\{\Theta_g, \Theta_\gamma, \Theta_\beta\}} \alpha\mathcal{L}_r^m&(\{\Theta_g, \Theta_\gamma, \Theta_\beta\}) \\ &+(1-\alpha)\mathcal{L}_r^{nm}(\{\Theta_g, \Theta_\gamma, \Theta_\beta\}). \end{aligned} \tag{6}$$

Here, the loss function $\mathcal{L}_r^m$ corresponds to the constraint that output generated using match condition vector $\tilde{X}_i^m$, should be perfect reconstruction of X_i. Whereas, the loss function $\mathcal{L}_r^{nm}$ corresponds to the constraint that output generated using non match condition vector $\tilde{X}_i^{nm}$, should have poor reconstruction. To enforce the later condition, another batch $\{X_1^{nm}, X_2^{nm}, ..., X_N^{nm}\}$, is sampled from the training data, such that new batch does not have class identity consistent with the match condition vector. This in effect achieves the goal of poor reconstruction when conditioned $l_{y_i^{nm}}$. This conditioning strategy in a way, emulates open-set behavior (as will be discussed further in Sec. 3.2.3). Here, the network is specifically trained to produce poor reconstructions when class identity of an input image does not match the condition vector. So, when encountered with an unknown class test sample, ideally none of the condition vector would match the input image class identity. This will result in poor reconstruction for all condition vectors. While, when encountered with the known test sample, as one of the condition vector will match the input image class identity, it will produce a perfect reconstruction for that particular condition vector. Hence, training with the non-match loss helps the network adapt better to open-set setting. Here, $\mathcal{L}_r^{nm}$ and $\mathcal{L}_r^m$ are weighted with $\alpha \in [0, 1]$ to get the the final training objective.

3.2.2 EVT Modeling

Extreme Value Theory. Extreme value theory is often used in many visual recognition systems and is an effective tool for modeling post-training scores [45], [46]. It has been used in many applications such as finance, railway track inspection etc. [28], [3], [12] as well as open-set recognition [7], [45], [50]. In this paper we follow the Picklands-Balkema-deHaan formulation [38], [5] of the extreme value theorem. It considers modeling probabilities conditioned on random variable exceeding a high threshold. For a given random variable W with a cumulative distribution function (CDF) $F_W(w)$ the conditional CDF for any w exceeding the threshold u is defined as,

$$F_U(w) = \mathcal{P}(w - u \leq w | w > u) = \frac{F_W(u+w) - F_W(u)}{1 - F_W(u)},$$

where, $\mathcal{P}(\cdot)$ denotes probability measure function. Now, given I.I.D. samples, $\{W_i, ..., W_n\}$, the extreme value the-

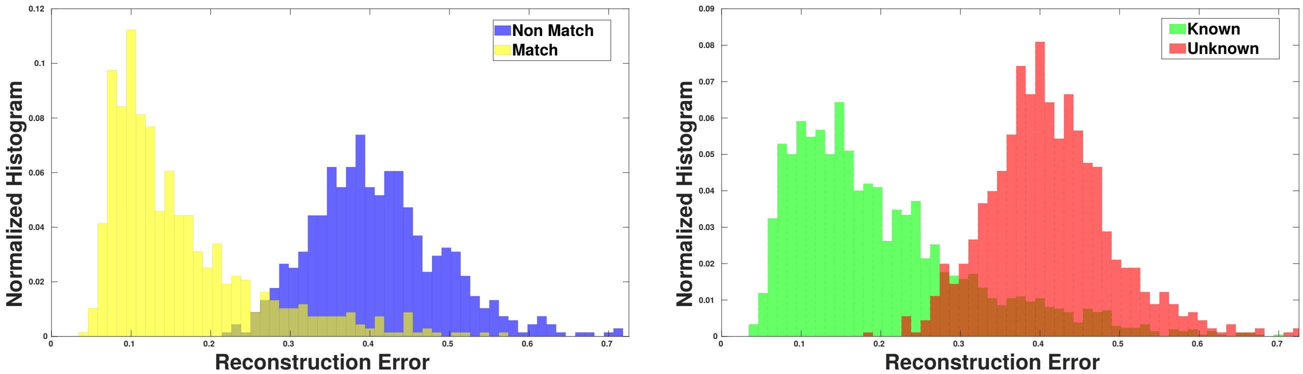


Figure 3: **(a)** Normalized histogram of match and non-match reconstruction errors. The match histogram in yellow shows distribution of elements in set S_m in Sec. 3.2.3. The non-match histogram showed in blue shows distribution of elements in set S_{nm}. **(b)** Normalized histogram of known and unknown reconstruction errors. The known histogram in green shows the distribution of known class and unknown histogram in red shows the distribution of unknown class reconstruction errors. Here, the histograms are computed using SVHN dataset.

orem [38] states that, for large class of underlying distributions and given a large enough u, F_U can be well approximated by the Generalized Pareto Distribution (GPD),

$$G(w;\zeta,\mu) = \begin{cases} 1-(1+\frac{\zeta \cdot w}{\mu})^{\frac{1}{\zeta}}, & \text{if } \zeta \neq 0, \\ 1-e^{\frac{w}{\mu}} & , \text{if } \zeta = 0, \end{cases} \quad (7)$$

such that $-\infty < \zeta < +\infty$, $0 < \mu < +\infty$, $w > 0$ and $\zeta w > -\mu$. $G(.)$ is CDF of GPD and for $\zeta = 0$, reduces to the exponential distribution with parameter μ and for $\zeta \neq 0$ takes the form of Pareto distribution [10].

Parameter Estimation. When modeling the tail of any distribution as GPD, the main challenge is in finding the tail parameter u to get the conditional CDF. However, it is possible to find an estimated value of u using mean excess function (MEF), *i.e.*, $E[W-u|W>u]$ [46]. It has been shown that for GPD, MEF holds a linear relationship with u. Many researchers use this property of GPD to estimate the value of u [46], [35]. Here, the algorithm for finding u, introduced in [35] for GPD is adopted with minor modifications. See [35], [46] for more details regarding MEF or tail parameter estimation. After getting an estimate for u, since from extreme value theorem [38], we know that set $\{w \in W \mid w > u\}$, follows GPD distribution, rest of the parameters for GPD, *i.e.* ζ and μ can be easily estimated using the maximum likelihood estimation techniques [14], except for some rarely observed cases [9].

3.2.3 Threshold Calculation

After the training procedure described in previous sections, Sec. 3.1 and Sec. 3.2, a set of match and non-match reconstruction errors are created from training set, $\{X_1, X_2, ..., X_{N_{train}}\} \in \mathcal{K}$, and their corresponding match and non match labels, $\{y_1^m, y_2^m, ..., y_{N_{train}}^m\}$ and $\{y_1^{nm}, y_2^{nm}, ..., y_{N_{train}}^{nm}\}$. Let, r_i^m be the match reconstruction error and r_i^{nm} be the non match reconstruction error for the input X_i, then the set of match and non match errors can be calculated as,

$$\begin{aligned} \tilde{X}_i^m &= \mathcal{G}(\mathcal{H}_\gamma(l_{y_i^m}) \odot \mathcal{F}(X_i) + \mathcal{H}_\beta(l_{y_i^m})), \\ \tilde{X}_i^{nm} &= \mathcal{G}(\mathcal{H}_\gamma(l_{y_i^{nm}}) \odot \mathcal{F}(X_i) + \mathcal{H}_\beta(l_{y_i^{nm}})), \\ S_m &= \{r_i^m \in \mathbb{R}^+ \cup \{0\} \mid r_i^m = \|X_i - \tilde{X}_i^m\|_1 \}, \\ S_{nm} &= \{r_i^{nm} \in \mathbb{R}^+ \cup \{0\} \mid r_i^{nm} = \|X_i - \tilde{X}_i^{nm}\|_1 \}, \\ &\forall i \in \{1, 2, ..., N_{train}\}. \end{aligned}$$

Typical histograms of S_m (set of match reconstruction errors) shown in Fig. 3a with color yellow, and S_{nm} (set of non-match reconstruction errors) shown in Fig. 3a with color blue. Note that the elements in these sets are calculated solely based on what is observed during training (i.e., without utilizing any unknown samples). Fig. 3b shows the histogram of reconstruction errors observed during inference from the test samples of known class set ($\mathcal{K}$) (shown in green), and unknown class set ($\mathcal{U}$) (shown in red). Comparing Fig. 3a and Fig. 3b, it can be observed that the distribution of S_m and S_{nm} computed during training, provides a good approximation for the error distributions observed during inference, for test samples from known set ($\mathcal{K}$) and unknown set ($\mathcal{U}$). This observation also validates that non match training emulates an open-set test scenario (also discussed in Sec. 3.2) where the input does not match any of the class labels. This motivates us to use S_m and S_{nm} to find an operating threshold for open-set recognition to make a decision about any test sample being known/unknown.

We can assume that the optimal operating threshold (τ^*) lies in the region $S_m \cap S_{nm}$. The underlying distributions of S_m and S_{nm} are not known. However, as explained in Sec. 3.2.2, it is possible to model the tails of S_m (right tail) and S_{nm} (left tail) with GPD as G_m and G_{nm} with $G(\cdot)$ being a CDF. Though, GPD is only defined for modeling maxima, before fitting G_{nm} left tail of S_{nm} we perform inverse transform as $S'_{nm} = -S_{nm}$. Assuming the prior probability of observing unknown samples is p_u, the probability of errors can be formulated as a function of the threshold τ,

$$\begin{aligned}
\tau^* &= \min_{\tau} \quad \mathcal{P}_{error}(\tau) \\
&= \min_{\tau} \quad [(1-p_u) * \mathcal{P}_m(r > \tau) + p_u * \mathcal{P}_{nm}(-r < -\tau)] \\
&= \min_{\tau} \quad [(1-p_u) * (1 - G_m(\tau)) + p_u * (1 - G_{nm}(\tau))].
\end{aligned}$$

Solving the above equation should give us an operating threshold that can minimize the probability of errors for a given model and can be solved by a simple line search algorithm by searching for τ^* in the range $\{S_m \cap S_{nm}\}$. Here, the accurate estimation of τ^* depends on how well S_m and S_{nm} represent the known and unknown error distributions. It also depends on the prior probability p_u, effect of this prior will be further discussed in Sec. 4.3.

3.3. Open-set Testing by k-inference (Stage 3)

Here, we introduce the open-set testing algorithm for proposed method. The testing procedure is described in Algo. 1 and an overview of this is also shown in Fig. 2. This testing strategy involves conditioning the decoder k-times with all possible condition vectors to get k reconstruction errors. Hence, it is referred as k-inference algorithm.

4. Experiments and Results

In this section we evaluate performance of the proposed approach and compare it with other open-set recognition methods. The experiments in Sec. 4.2, we measure the ability of algorithm to identify test samples as known or unknown without considering operating threshold. In second set of experiments, we measure overall performance of open-set recognition algorithm. Additionally through ablation experiments, we analyze contribution from each component of the proposed method.

4.1. Implementation Details

We use Adam optimizer [22] with learning rate 0.0003 and batch size, N=64. The parameter α, described in Sec. 3.2, is set equal to 0.9. For all the experiments, conditioning layer networks $\mathcal{H}_\gamma$ and $\mathcal{H}_\beta$ are a single layer fully connected neural networks. Another important factor affecting open-set performance is openness of the problem. Defined by Scheirer *et al.* [44], it quantifies how open the problem setting is,

Algorithm 1 k-Inference Algorithm

Require: Trained network models $\mathcal{F}, \mathcal{C}, \mathcal{G}, \mathcal{H}_\gamma, \mathcal{H}_\beta$
Require: Threshold τ from EVT model
Require: Test image X, k condition vectors $\{l_1, \ldots, l_k\}$
Latent space representation, $z = \mathcal{F}(X)$
Prediction probabilities, $p_y = \mathcal{C}(z)$
predict known label, $y_{pred} = argmax(p_y)$
for $i = 1, \ldots, k$ **do**
  $z_{l_i} = \mathcal{H}_\gamma(l_i) \odot z + \mathcal{H}_\beta(l_i)$
  $\tilde{X}_i = \mathcal{G}(z_{l_i})$
  $\text{Rec}(i) = ||X - \tilde{X}_i||_1$
end for
$\text{Rec}_{min} = sort(\text{Rec})$
if $\text{Rec}_{min} < \tau$ **do**
  predict X as Known, with label y_{pred}
else do
  predict X as Unknown
end if

$$\mathbf{O} = 1 - \sqrt{\frac{2 \times N_{train}}{N_{test} + N_{target}}}, \tag{8}$$

where, N_{train} is the number of training classes seen during training, N_{test} is the number of test classes that will be observed during testing, N_{target} is the number of target classes that needs to be correctly recognized during testing. We evaluate performance over multiple openness value depending on the experiment and dataset.

4.2. Experiment I : Open-set Identification

The evaluation protocol defined in [29] is considered and area under ROC (AUROC) is used as evaluation metric. AUROC provides a calibration free measure and characterizes the performance for a given score by varying threshold. The encoder, decoder and classifier architecture for this experiment is similar to the architecture used by [29] in their experiments. Following the protocol in [29], we report the AUROC averaged over five randomized trials.

4.2.1 Datasets

MNIST, SVHN, CIFAR10. For MNIST [26], SVHN [30] and CIFAR10 [23], openness of the problem is set to $\mathbf{O}$ = 13.39%, by randomly sampling 6 known classes and 4 unknown classes.
CIFAR+10, CIFAR+50. For CIFAR+M experiments, 4 known classes are sampled from CIFAR10. M non overlapping classes are used as the unknowns, sampled from the CIFAR100 [23]. Openness of the problem for CIFAR+10 and CIFAR+50 is $\mathbf{O}$ = 33.33% and 62.86%, respectively.
TinyImageNet. For experiments with the TinyImageNet

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50	TinyImageNet
SoftMax	0.978	0.886	0.677	0.816	0.805	0.577
OpenMax [7] (CVPR'16)	0.981	0.894	0.695	0.817	0.796	0.576
G-OpenMax [11] (BMVC'17)	0.984	0.896	0.675	0.827	0.819	0.580
OSRCI [29] (ECCV'18)	0.988	0.910	0.699	0.838	0.827	0.586
Proposed Method	**0.989**	**0.922**	**0.895**	**0.955**	**0.937**	**0.748**

Table 1: AUROC for open-set identification, values other than the proposed method are taken from [29].

[25], 20 known classes and 180 unknown classes with openness $\mathbf{O} = 57.35\%$ are randomly sampled for evaluation.

4.2.2 Comparison with state-of-the-art

For comparing the open-set identification performance, we consider the following methods:
I. SoftMax : SoftMax score of a predicted class is used for open-set identification.
II. OpenMax [7]: The score of k+1th class and score of the predicted class is used for open-set identification.
III. G-OpenMax [11]: It is a data augmentation technique, which utilizes the OpenMax scores after training the network with the generated data.
IV. OSRCI [29]: Another data augmentation technique called counterfactual image generation is used for training the network for k+1 class classification. We refer to this method as Open-set Recognition using Counterfactual Images (OSRCI). The score value $P(y_{k+1}) - \underset{i \leq k}{max}\ P(y_i)$ is used for open-set identification.

Results corresponding to this experiment are shown in Table 1. As seen from this table, the proposed method outperform the other methods, showing that open-set identification training in proposed approach learns better scores for identifying unknown classes. From the results, we see that our method on the digits dataset produces a minor improvement compared to the other recent methods. This is mainly do the reason that results on the digits dataset are almost saturated. On the other hand, our method performs significantly better than the other recent methods on the object datasets such as CIFAR and TinyImageNet.

4.3. Experiment II : Open-set Recognition

This experiment shows the overall open-set recognition performance evaluated with F-measure. For this experiment we consider LFW dataset [27]. We extend the protocol introduced in [44] where, 12 classes containing more than 50 images are considered as known classes and divided into 80/20 train-test split. Image size is kept to 64×64. We vary the openness from 0% to 93% by taking 18 to 5705 unknown classes during testing. Since, many classes contain only one image, instead of random sampling, we sort them according to the number of images per class and add it sequentially to increase the openness. It is obvious that with the increase in openness, the probability of observing unknown will also increase. Hence, it is reasonable to assume that prior probability p_u will be a function of openness. For this experiment we set $p_u = 0.5 * \mathbf{O}$.

4.3.1 Comparison with state-of-the-art

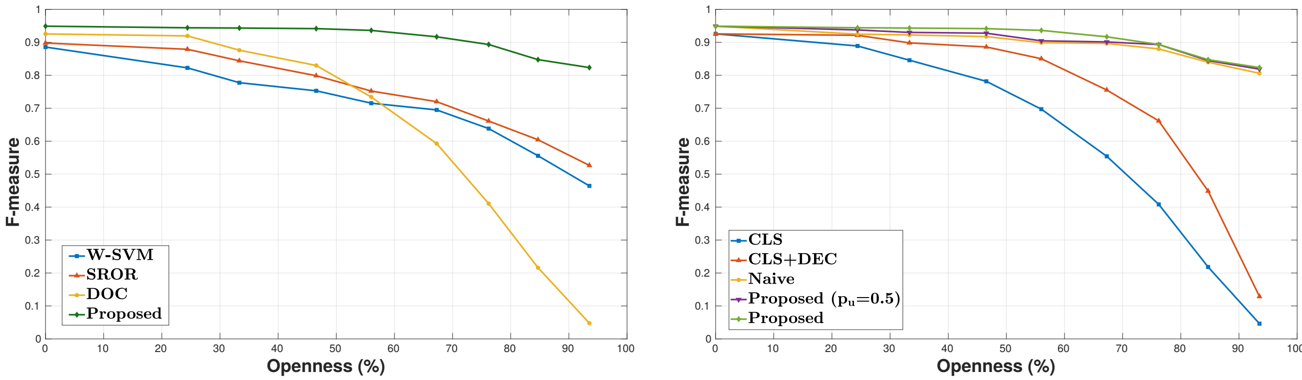
For comparing the open-set recognition performance, we consider the following methods:
I. W-SVM (PAMI'14) : W-SVM is used as formulated in [44], which trains Weibull calibrated SVM classifier for open set recognition.
II. SROR (PAMI'16) : SROR is used as formulated in [50]. It uses sparse representation-based framework for open-set recognition.
III. DOC (EMNLP'16) : It utilizes a novel sigmoid-based loss function for training a deep neural network [47].

To have a fair comparison with these methods, we use features extracted from the encoder ($\mathcal{F}$) to train W-SVM and SROR. For DOC, the encoder ($\mathcal{F}$) is trained with the loss function proposed in [47]. Experiments on LFW are performed using a U-Net [42] inspired encoder-decoder architecture. More details regarding network architecture is included in the supplementary material.

Results corresponding to this experiment is shown in Fig. 4a. From this figure, we can see that the proposed approach remains relatively stable with the increase in openness, outperforming all other methods. One interesting trend noticed here is, that DOC initially performs better than the statistical methods such as W-SVM and SROR. However with openness more than 50% the performance suffers significantly. While the statistical methods though initially perform poor compared to DOC, but remain relatively stable and performs better than DOC as the openness is increased (especially over $\mathbf{O}$ >50%).

4.3.2 Ablation Study

In this section, we present ablation analysis of the proposed approach on the LFW dataset. The contribution of individual components is reported by creating multiple baselines of the proposed approach. Starting with the most simple baseline, *i.e.*, thresholding SoftMax probabilities of a closed-set model, each component is added building up to the pro-



(a) F-measure comparisons for the open-set recognition experiment.
(b) F-measure comparisons for the ablation study.

Figure 4: Performance evaluation on the LFW dataset.

posed approach and are described as follows,

I. CLS : Encoder $(\mathcal{F})$ and the classifier $(\mathcal{C})$ are trained for k-class classification. Samples with probability score prediction less than 0.5 are classified as unknown.

II. CLS+DEC : In this baseline, only the networks $\mathcal{F}, \mathcal{C}$ and the decoder $(\mathcal{G})$ are trained as described in Sec. 3, except $\mathcal{G}$ is only trained with match loss function, $\mathcal{L}_r^m$. Samples with more than 95% of maximum train reconstruction error observed, are classified as unknown.

III. Naive : Here, the networks $\mathcal{F}$, $\mathcal{C}$ and $\mathcal{G}$ and the conditioning layer networks ($\mathcal{H}_\gamma$ and $\mathcal{H}_\beta$) are trained as described in Sec. 3, but instead of modeling the scores using EVT as described in Sec. 3.2.3, threshold is directly estimated from the raw reconstruction errors.

IV. Proposed method (p_u = 0.5) : $\mathcal{F}$, $\mathcal{C}$, $\mathcal{G}$ and condition layer networks ($\mathcal{H}_\gamma$ and $\mathcal{H}_\beta$) are trained as described in Sec. 3 and to find the threshold prior probability of observing unknown is set to $p_u = 0.5$.

V. Proposed method: Method proposed in this paper, with p_u set as described in Sec. 4.3.

Results corresponding to the ablation study are shown in Fig. 4b. Being a simple SoftMax thresholding baseline, **CLS** has weakest performance. However, when added with a match loss function ($\mathcal{L}_r^m$) as in **CLS+DEC**, the open-set identification is performed using reconstruction scores. Since, it follows a heuristic way of thresholding, the performance degrades rapidly as openness increases. However, addition of non match loss function ($\mathcal{L}_r^{nm}$), as in the **Naive** baseline, helps find a threshold value without relying on heuristics. As seen from the Fig. 4b performance of **Naive** baseline remains relatively stable with increase in openness, showing the importance of loss function $\mathcal{L}_r^{nm}$. Proposed method with p_u fixed to 0.5, introduces EVT modeling on reconstruction errors to calculate a better operating threshold. It can be seen from the Fig. 4b, such strategy improves over finding threshold based on raw score values. This shows importance applying EVT models on reconstruction errors. Now, if p_u is set to $0.5 * \mathbf{O}$, as in the proposed method, there is a marginal improvement over the fixed p_u baseline. This shows benefit of setting p_u as a function of openness. It is interesting to note that for large openness values (as $0.5 * \mathbf{O} \to 0.5$), both fixed p_u baseline and proposed method achieve similar performance.

5. Conclusion

We presented an open-set recognition algorithm based on class conditioned auto-encoders. We introduced training and testing strategy for these networks. It was also shown that dividing the open-set recognition into sub tasks helps learn a better score for open-set identification. During training, enforcing conditional reconstruction constraints are enforced, which helps learning approximate known and unknown score distributions of reconstruction errors. Later, this was used to calculate an operating threshold for the model. Since inference for a single sample needs k feed-forwards, it suffers from increased test time. However, the proposed approach performs well across multiple image classification datasets and providing significant improvements over many state of the art open-set algorithms. In our future research, generative models such as GAN/VAE/FLOW can be explored to modify this method. We will revise the manuscript with such details in the conclusion.

Acknowledgements

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government.

References

[1] Mahdi Abavisani and Vishal M Patel. Adversarial domain adaptive subspace clustering. In *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pages 1–8. IEEE, 2018.

[2] Mahdi Abavisani and Vishal M Patel. Deep multimodal subspace clustering networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1601–1614, 2018.

[3] Isabel Fraga Alves and Cláudia Neves. Extreme value distributions. In *International encyclopedia of statistical science*, pages 493–496. Springer, 2011.

[4] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Finding tiny faces in the wild with generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2018.

[5] August A Balkema and Laurens De Haan. Residual life time at great age. *The Annals of probability*, pages 792–804, 1974.

[6] 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.

[7] 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.

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[9] Vartan Choulakian and Michael A Stephens. Goodness-of-fit tests for the generalized pareto distribution. *Technometrics*, 43(4):478–484, 2001.

[10] Herbert Aron David and Haikady Navada Nagaraja. *Order statistics*. Wiley Online Library, 1970.

[11] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017.

[12] Xavier Gibert, Vishal M Patel, and Rama Chellappa. Deep multitask learning for railway track inspection. *IEEE Transactions on Intelligent Transportation Systems*, 18(1):153–164, 2017.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[14] Scott D Grimshaw. Computing maximum likelihood estimates for the generalized pareto distribution. *Technometrics*, 35(2):185–191, 1993.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[16] 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.

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[18] 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.

[19] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *Advances in Neural Information Processing Systems*, pages 24–33, 2017.

[20] Pedro Ribeiro Mendes Júnior, Terrance E Boult, Jacques Wainer, and Anderson Rocha. Specialized support vector machines for open-set recognition. *arXiv preprint arXiv:1606.03802*, 2016.

[21] 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.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

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

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[28] Gerald A Meehl, Thomas Karl, David R Easterling, Stanley Changnon, Roger Pielke Jr, David Changnon, Jenni Evans, Pavel Ya Groisman, Thomas R Knutson, Kenneth E Kunkel, et al. An introduction to trends in extreme weather and climate events: observations, socioeconomic impacts, terrestrial ecological impacts, and model projections. *Bulletin of the American Meteorological Society*, 81(3):413–416, 2000.

[29] 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 (ECCV)*, pages 613–628, 2018.

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[31] Poojan Oza and Vishal M Patel. Active authentication using an autoencoder regularized cnn-based one-class classifier. *arXiv preprint arXiv:1903.01031*, 2019.

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

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

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

[35] Pramuditha Perera and Vishal M Patel. Extreme value analysis for mobile active user authentication. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 346–353. IEEE, 2017.

[36] Pramuditha Perera and Vishal M Patel. Learning deep features for one-class classification. *arXiv preprint arXiv:1801.05365*, 2018.

[37] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*, 2017.

[38] James Pickands III et al. Statistical inference using extreme order statistics. *the Annals of Statistics*, 3(1):119–131, 1975.

[39] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in Neural Information Processing Systems*, pages 6822–6833, 2018.

[40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[41] 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.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[43] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4390–4399, 2018.

[44] 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, 2013.

[45] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014.

[46] Zhixin Shi, Frederick Kiefer, John Schneider, and Venu Govindaraju. Modeling biometric systems using the general pareto distribution (gpd). In *Biometric Technology for Human Identification V*, volume 6944, page 69440O. International Society for Optics and Photonics, 2008.

[47] Lei Shu, Hu Xu, and Bing Liu. Doc: Deep open classification of text documents. *arXiv preprint arXiv:1709.08716*, 2017.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] Vishwanath Sindagi and Vishal Patel. Dafe-fd: Density aware feature enrichment for face detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2185–2195. IEEE, 2019.

[50] He Zhang and Vishal M Patel. Sparse representation-based open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1690–1696, 2017.