Deep End-to-End One-Class Classifier

Mohammad Sabokrou, Mahmood Fathy, Guoying Zhao, and Ehsan Adeli

***Abstract*— One-class classification (OCC) poses as an essential component in many machine learning and computer vision applications, including novelty, anomaly, and outlier detection systems. With a known definition for a target or normal set of data, one-class classifiers can determine if any given new sample spans within the distribution of the target class. Solving for this task in a general setting is particularly very challenging, due to the high diversity of samples from the target class and the absence of any supervising signal over the novelty (nontarget) concept, which makes designing end-to-end models unattainable. In this article, we propose an adversarial training approach to detect out-of-distribution samples in an end-to-end trainable deep model. To this end, we jointly train two deep neural networks, $\mathcal{R}$ and $\mathcal{D}$. The latter plays as the discriminator while the former, during training, helps $\mathcal{D}$ characterize a probability distribution for the target class by creating adversarial examples and, during testing, collaborates with it to detect novelties. Using our OCC, we first test outlier detection on two image data sets, Modified National Institute of Standards and Technology (MNIST) and Caltech-256. Then, several experiments for video anomaly detection are performed on University of Minnesota (UMN) and University of California, San Diego (UCSD) data sets. Our proposed method can successfully learn the target class underlying distribution and outperforms other approaches.**

***Index Terms*— Generative adversarial network (GAN), one-class classification (OCC), video anomaly detection.**

I. Introduction

ONE-CLASS classification (OCC) is the task of detecting rare or outlier samples that do not follow the distribution of normal or inlier samples. Therefore, OCC is considerably related to different computer vision problems, such as novelty detection [1]–[4], outlier detection [5]–[8], image denoising [9], and anomaly detection [10]–[13]. These tasks can generally be defined under the umbrella of OCC [14]–[18], in which the target is to learn a classification model in the absence of an assumptive negative class. Under this setting, the negative class can be assumed as the outlier or anomaly class, and numerous training data points from the positive (or target) class pose the normal class, around which the OCC is built.

To precisely learn the intrinsic geometry of the target class, an efficient and discriminative representation of the data is needed that enables entangling the different explanatory variations within the training data. Recently, deep neural networks have achieved great success in visual data representation for a wide range of computer vision tasks [19], [20], mainly when they are learned as an end-to-end network. Designing an end-to-end deep network for novelty detection and OCC applications is not a straightforward task due to the unavailability of training samples from the negative class. In the past years, some efforts have been made to take advantage of learned features for training one-class classifiers [12], [21]–[25], most of which are not trainable in an end-to-end fashion. Recently, [1] proposed an end-to-end deep network applied to various applications such as outlier and anomaly detection in images and in videos.

Inspired by the recent developments in generative adversarial networks (GANs) [26], we leverage adversarial learning techniques to enable our one-class classifier to operate in an end-to-end manner. Our network is comprised of two modules, which compete during training but collaborate with each other in test time for the detection task. The first component (named $\mathcal{R}$) introduces discriminative factors for making the target and outlier samples more distinguishable for the $\mathcal{D}$ (i.e., detector).

$\mathcal{R}$ and $\mathcal{D}$ are trained on training samples only from the target class in a way that $\mathcal{R}$ can efficiently reconstruct the positive instances with the aim of fooling $\mathcal{D}$. On the other hand, $\mathcal{D}$ is trained to make a separation between the original (positive) and reconstructed samples. Thus, $\mathcal{D}$ learns the distribution of positive class to accurately detect the positive and novelty samples. Concurrently, $\mathcal{R}$ is trained to correctly reconstruct samples that are deemed to be positive. The negative samples are only given to the network during testing time. Therefore, the novelty concept(s) will be new for $\mathcal{R}$ and has a hard time correctly reconstructing them. This is a side effect of our training scheme that we fully take advantage of, for our OCC application. Hence, $\mathcal{R}$ acts as a decimator or a distorter network for samples from out of target class. In the testing time, $\mathcal{D}$ operates as an efficient detector and $\mathcal{R}$ supports it for improving the detection performance by perfectly reconstructing the target samples and decimating (or distorting) any other (i.e. novel) samples. The architecture of the proposed approach is shown in Fig. 1.

A preliminary version of this article was presented and published in Computer Vision and Pattern Recognition (CVPR) conference 2018 [1]. In this article, we extend the idea by forcing the autoencoder in $\mathcal{R}$ to learn a latent representation space that follows a normal distribution (to create a continuous latent space with simple interpolation capabilities for unseen

Manuscript received August 14, 2018; revised April 26, 2019, November 8, 2019, and January 8, 2020; accepted February 29, 2020. *(Corresponding author: Ehsan Adeli.)*

Mohammad Sabokrou and Mahmood Fathy are with the School of Computer Science, Institute for Research in Fundamental Science (IPM), Tehran 19395-5746, Iran (e-mail: sabokro@ipm.ir).

Guoying Zhao is with the Center for Machine Vision and Signal Analysis, University of Oulu, 90570 Oulu, Finland (e-mail: guoying.zhao@oulu.fi).

Ehsan Adeli is with the Department of Computer Science and the Department of Psychiatry and Behavioral Sciences, Stanford University, Stanford, CA 94305 USA (e-mail: eadeli@cs.stanford.edu).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2979049

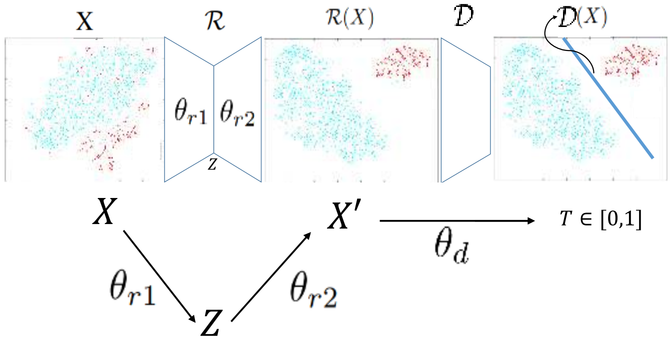


Fig. 1. Our method is composed of two modules, $\mathcal{R}$ and $\mathcal{D}$, which are trained adversarially. During training, $\mathcal{R}$ is optimized for reconstructing samples belonging to the target class to fool $\mathcal{D}$, whereas $\mathcal{D}$ detects if its input is real or generated by $\mathcal{R}$ (i.e., is fake). In testing, $\mathcal{R}$ can well reconstruct inlier samples while decimating its outlier inputs (has not seen any outliers during training). $\mathcal{D}$ classifies its input as positive (inlier or target) or negative (outlier or anomaly). As can be seen, $\mathcal{R}(X)$ results in better separability compared to the original feature space, X. The encoding, Z in $\mathcal{R}$, is learned such that it is able to reconstruct target class samples so that it successfully fools $\mathcal{D}$. $\mathcal{D}(\mathcal{R}(X))$, on the other hand, evaluates how likely it is that its input sample belongs to the target class.

target class data), including more comprehensive theoretical discussions and findings, and conduct more experiments. Specifically, the main differences between this article and its earlier version are: 1) we propose a new structure and loss function for $\mathcal{R}$, which improve its performance and 2) finding the optimum time to stop the training of $\mathcal{R} + \mathcal{D}$ was an important challenge in an earlier version of this article. In this article, we present a simple yet effective approach for stopping the training. 3) We explain the reject region (RR) of our method in detail and show how the method benefits from the additional component in the loss function. In Section III, these details are further explained.

This article makes the following contributions: 1) we present an efficient yet simple end-to-end learning method for OCC tasks. 2) Unlike the previous work (see [27]) that after training throw away the generator or the discriminator (i.e., $\mathcal{R}$ and $\mathcal{D}$ networks in our setup), our set up efficiently takes advantage of both trained networks to improve the performance in the testing step (inference time). 3) The proposed method is trained in the presence of only samples from the target class and outperforms state-of-the-art methods for various applications.

The rest of this article is organized as follows: A brief survey on related works is provided in Section II. Section III introduces the proposed method, architectures of $\mathcal{R}$ and $\mathcal{D}$, and their joint training strategy. The results of several experiments are reported in Section IV. Finally, Section V concludes this article.

II. Related Works

As discussed earlier, challenging and important tasks of detecting anomalies, outliers, or rare events can be formulated as an OCC task. All of these tasks search for a concept that is not (or is scarce) occurred within the training samples. They, hence, all can be solved by a *one-class classification* strategy. Traditional OCC methods learn a reference model for the target class. Then, samples with high divergence from such a reference model are detected as a novelty.

Statistical modeling [28] and self-representation learning [5], [6], [10], [29] are two commonly used solutions for solving the OCC problem. Efficient data representation is an essential part of the obtaining satisfactory final performance. The previous works used either low-level [30], high-level (e.g., trajectories [31]), or deep features [12], [21], [32]. Successful methods for OCC are briefly surveyed in the following.

A. Self-Representation

In several previous works, self-representation was used as a useful approach for novelty detection and OCC in general. As an example, [10] and [29] proposed exploiting self-representation for detecting the irregular events in videos by taking advantage of sparse representation learning. They used sparsity as criteria to distinguish between inlier and outlier samples. In other works (such as [10] and [21]), in the testing step, the samples are reconstructed by an encoder–decoder trained on only target (inlier) class data. Then, based on the error induced by the reconstruction, the input is decided to be inlier or outlier. Outliers are those that have larger errors of reconstruction. Liu *et al.* [33], instead of sparse representation, used a low-rank representation method, which led to more robustness against outliers [34]. In a similar way, encoder–decoder networks based on reconstruction error loss functions are used for removing outliers or detecting anomalies in videos [21], [29], [35].

B. Statistical Methods

Several works attempted to understand the manifold spanned by the positive samples by analyzing their statistical characteristics. In a simple way, they represented the samples from the target class to a feature space with reduced dimensionality, and a probability distribution with the maximum likelihood is fit on such represented samples. Samples that do not comply with the fit distribution can define the outliers (see [28], [36], [37]). Rahmani and Atia [38] presented a simple method, coherence pursuit, which assumed the correlation between inliers should be high. Therefore, the inlier samples should be part of a low-dimensional subspace. Outliers, on the other hand, do not follow the same subspace or may create small data clusters. Other works, such as Xu *et al.* [39] and Lerman *et al.* [40], proposed OutlierPursuit and REAPER, respectively, based on convex optimization methods.

C. Supervised Methods Based on Constrained Reconstruction

As discussed earlier, representation learning is vastly used outlier and anomaly classification. Other works have also used such learning techniques with the consideration of a constraint for similar tasks. In this case, given test data, it can be considered as an outlier if it does not follow the same constraint. Of such methods, spare representation learning [10] and minimum effort [41] are two widely examined methods. For instance, the method proposed in [41] considers the input sequence to be an anomaly if it is hard to reconstruct it using the previous observations in the same sequence. Similarly, [42]

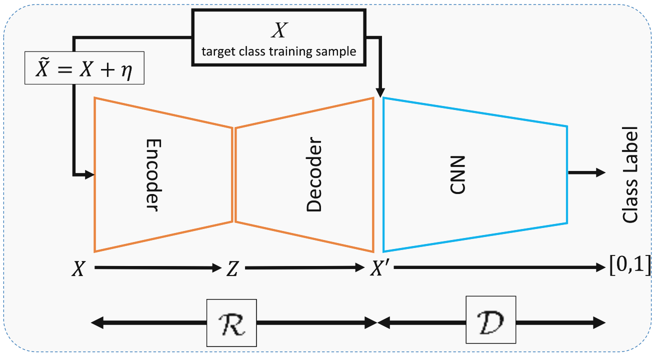


Fig. 2. Overall scheme of the proposed approach.

introduces a scene parsing method to detect foreground object anomalies. It assumes that all objects that cannot be properly explained through normal training can be anomalies. Several other works [1], [10], [29] train based on the same assumption that minimum reconstruction error is achieved if input samples are inliers.

III. Adversarial One-Class Classifier

In recent years, deep generative models, especially GANs [26], [43], have achieved promising results for data generation. Such methods have also been used for developing and improving other machine learning tasks, such as classification under limited supervision (e.g., in [23], [44], and [45]). GANs are trained by competition between two convolutional neural networks (CNNs) [i.e. generator (G) and discriminator (D) networks] in an unsupervised manner. G aims to generate realistic samples (e.g., images) and D acts as a detector network, and tries to distinguish between realistic and generated data by G. There are different versions of GANs, but the most related one to ours is conditional GAN (CGAN). With an input image X, the G of CGAN reconstructs (regenerates) an image X', and D learns to discriminate between X and X'. G attempts to convince D that the generated images are realistic. Recently, Isola *et al.* [46], inspired by CGANs, introduced a method for "image-to-image translation." Their method had both G and D conditioned on some real data. They investigated that encoder–decoder architectures similar to U-Net [47] can be used as the generator with a CNN discriminator network for transforming images from different representations. Similarly, [22] used a generator that reconstructs the normal samples. If such a network is not able to thoroughly reconstruct a testing input, it is deemed as an anomaly. In our method, we propose to use $\mathcal{R}$ not only to reconstruct its input samples (all from the target class during training) but also to improve the performance of the outlier detection. $\mathcal{R}$ does this by purifying the target class samples and distorting sample from the out-of-target-class-distribution.

In summary, our proposed framework for OCC is a combination of two modules: 1) $\mathcal{R}$efiner and 2) $\mathcal{D}$etector. The first network $\mathcal{R}$efins (or $\mathcal{R}$econstructs) the input samples, while the second one acts as a $\mathcal{D}$iscriminator (or $\mathcal{D}$etector). These networks are adversarially and unsupervisedly learned within an end-to-end learning procedure (see Fig. 2).

A. $\mathcal{R}$ and $\mathcal{D}$ Networks

Previously, [5], [29], and [48] have proven that the inability for prefect reconstruction by an auto-encoder, learned only on target class samples, can be used to define a discriminative and informative measure to separate inlier (i.e. target class) samples and outliers. This happens because such a network minimizes its reconstruction error loss function with respect to only its training samples containing only inliers. Hence, the error of reconstruction for the outlier samples would be large.

$\mathcal{R}$ itself has two components: 1) encoder ($\mathcal{R}_1$) and (2) decoder ($\mathcal{R}_2$). Conventional encoder–decoder neural networks (i.e., auto-encoders [49]) learn to generate compact representations and reconstruct their inputs; however, they have been mainly used for a few applications, such as denoising [50] or image in-painting [51]. As shown in previous works [12], [32], data from the target class can be statistically, using multivariate Gaussian distributions, which led to obtaining state-of-the-art results. Consequently, unlike the previous work [1], here, $\mathcal{R}_1$ and $\mathcal{R}_2$ are considered to be convolutional networks forced to learn a latent space (i.e., Z) that follows a Gaussian distribution as a prior-knowledge. Therefore, $\mathcal{R}_1$ is optimized to map the target class samples to a latent variable Z with the Gaussian distribution and, $\mathcal{R}_2$ reconstructs them back.

As also confirmed by [52], variational auto-encoders (VAEs) can learn a latent space with a Gaussian distribution for target class samples. Therefore, samples with a latent vector out of the Gaussian distribution can be considered as outliers. Following their idea, we found that encoding and decoding the target class sample with the restriction to force the latent space (i.e., Z) to follow Gaussian distribution leads to better performance. This is mainly because the latent variable is spanned in a continuous space defined by the Gaussian distribution, onto which unseen samples (during testing) can be reliably mapped (possibly through interpolation) [12], [32]. Consequently, both $\mathcal{P}(\mathcal{R}_1(X))$ and $||\mathcal{R}(X) - X||^2$ are accounted for training $\mathcal{R}$. Forcing the latent space of $\mathcal{R}$ to follow a specific distribution ensures the target class is spanned on a Gaussian and everything else not following that distribution can be counted as outliers. Hence, overall it improves the performance of separating the target and novelty samples. The detailed architectures of $\mathcal{R}$ and $\mathcal{D}$ are outlined in the Supplementary Material.

B. Training $\mathcal{R}$ and $\mathcal{D}$

Following explanation in II, Goodfellow *et al.* [26] proposed GANs, where two deep networks [denoted by generator (G) and discriminator (D)] were learned in an adverserial manner. Generating samples with the same distribution as the real samples data set is the target of such network. G trains to act as a transformer for mapping a random vector, Z following the probability distribution p_{latent}, to an instance with the real distribution (p_{target} in our case). D trains to make a discrimination between real and generated samples. Hence, $G + D$ are optimized in a two-player min–max game

$$\min_G \max_D \big(\mathbb{E}_{X\sim p_{\text{target}}}[\log(D(X))] + \mathbb{E}_{Z\sim p_{\text{latent}}}[\log(1 - D(G(Z)))]\big). \quad (1)$$

Here, we use a similar objective function as earlier with some modifications. Mapping a randomly generated latent space to the target class is not a well-defined task and there is no guarantee for the network to converge and recognize the whole space of samples from the target class, whereas similar to [53], we choose to learn a meaningful latent space (i.e., learning a representation of the target class using a VAE). This also avoids the problems of missing modes and mode collapse in our GAN setting, since the network directly learns from all target samples [54]–[56].

Similar to the conventional GANs, $\mathcal{R}+\mathcal{D}$ networks are adversarially and unsupervisedly trained. But unlike the classic GAN, instead of generating an instance with the p_{target} probability distribution from the latent space Z, $\mathcal{R}$ maps

$$\tilde{X} = (X \sim p_{\text{target}}) + (\eta \sim \mathcal{N}(0, \sigma^2\mathbf{I})) \longrightarrow X' \sim p_{\text{target}} \tag{2}$$

where η is sampled from $\mathcal{N}(0, \sigma^2\mathbf{I})$. For simplicity, in the rest of this article, $\mathcal{N}_\sigma$ will be used as the noise model. The role of η is to improve the robustness of $\mathcal{R}$ during training. As explained, p_{target} is the target class distribution. $\mathcal{D}$ knows p_{target} through the access to numerous samples belonging to the target class. Hence, $\mathcal{D}$ decides whether $\mathcal{R}(\tilde{X})$ complies with p_{target} or not. Therefore, $\mathcal{R}+\mathcal{D}$ are jointly trained with respect to the following objective:

$$\begin{aligned}\min_{\mathcal{R}} \max_{\mathcal{D}} \big(&\mathbb{E}_{X\sim p_{\text{target}}}[\log(\mathcal{D}(X))] \\ &+ \mathbb{E}_{\tilde{X}\sim p_{\text{target}}+\mathcal{N}_\sigma}[\log(1-\mathcal{D}(\mathcal{R}(\tilde{X})))]\big).\end{aligned} \tag{3}$$

With respect to the above-mentioned objective function (similar to GAN), $\mathcal{R}$ produces images following p_{target} distribution. Therefore, we are interested to maximize $p_{\text{target}}(\mathcal{R}(X \sim p_{\text{target}}; \theta_r))$, where θ_r is the parameter set of the $\mathcal{R}$.

To train the model, we calculate the loss $\mathcal{L}_{\mathcal{R}+\mathcal{D}}$ as the loss function of the joint network $\mathcal{R}+\mathcal{D}$ (i.e., the GAN loss). Besides, we need the output of $\mathcal{R}$ to resemble its original input, and therefore, we employ an extra loss component

$$\mathcal{L}_{\mathcal{R}} = \|X - X'\|^2. \tag{4}$$

In addition, to force Z [i.e., the output of $\mathcal{R}_1(X)$] to follow from a Gaussian distribution, Kullback–Leibler divergence (KL divergence) [57] $KL(Z||N_{(\mu_1,\sigma_1)})$ is also added to loss function. Therefore, the model minimizes

$$\mathcal{L} = \mathcal{L}_{\mathcal{R}+\mathcal{D}} + \lambda_1\mathcal{L}_{\mathcal{R}} + \lambda_2 KL(Z||N_{\mu_1,\sigma_1}) \tag{5}$$

where $\lambda_1, \lambda_2 > 0$ (trade-off hyperparameters) control the contribution of the terms. After training $\mathcal{R}+\mathcal{D}$ joint network, we can say the following.

1) $\mathcal{R}_1(X \sim p_{\text{target}} + \eta) \longrightarrow X' \sim \mathcal{N}(\mu_1, \sigma_1)$, θ_{r1} is learned to provide a representation following $\mathcal{N}(\mu_1, \sigma_1)$.
2) If $\mathcal{R}_2(Z \nsim \mathcal{N}(\mu_1, \sigma_1)) \longrightarrow X' \nsim \mathcal{N}(\mu_1, \sigma_1)$ or $\mathcal{P}(X \nsim p_{\text{target}} + \eta)$ is a very low value. As mentioned earlier, $\mathcal{R}_2$ is trained to map latent vector $Z \sim \mathcal{N}(\mu_1, \sigma_1)$ to a sample that follows from p_t. If Z does not follow $\mathcal{N}(\mu_1, \sigma_1)$), as it is expected to do so, the output of $\mathcal{R}_2$ will neither follow p_t. Similar concept was used in [52].
3) We can expect that $\forall X \sim p_{\text{target}} + \eta$ and $Y \nsim p_{\text{target}} + \eta$ we have $\mathcal{P}(R_1(X)|\mathcal{N}, \theta_{r1}) \geq \mathcal{P}(R_1(Y)|\mathcal{N}, \theta_{r1})$. This phenomenon is because of the KL divergence regularization in the loss function. Consequently, $\mathcal{R}_1$ can also be considered as a novelty detector.
4) With $\|X - X'\|^2$ minimized, $\mathcal{R}(X \sim p_{\text{target}} + \eta) \longrightarrow X' \sim p_{\text{target}}$, since θ_r is optimized to reconstruct inputs following p_{target}. Note, due to the way $\mathcal{R}$ is trained, it works like denoising auto-encoders [49] or denoising CNNs [50] (see Figs. 3 and 4 for examples).
5) Any input outlier instance $\hat{X}$, i.e., not following p_{target} will confuse $\mathcal{R}$ because it has not seen that concept during training. Therefore, $\mathcal{R}$ maps it to $\hat{X}'$ with an undefined distribution, $p_{\text{undefined}}$, [i.e., $\mathcal{R}(\hat{X} \nsim p_{\text{target}}+\eta) \longrightarrow \hat{X}' \sim p_{\text{undefined}}$]. Under this circumstance, $\|\hat{X} - \hat{X}'\|^2$ may not be minimized properly to become close to zero (similar to [22]). A *side effect* of this situatgion is that $\mathcal{R}$ decimates its input outliers (see Fig. 4 for some examples for a network $\mathcal{R}$ that was trained to detect digit "1"). In the examples is Fig. 4, digits "1" are properly reconstructed and others are somewhat decimated.
6) It is expected that $\mathcal{D}(X' \sim p_{\text{target}}) > \mathcal{D}(\hat{X}' \nsim p_{\text{target}})$. This is because $\mathcal{D}$ properly detects inputs following p_{target}.
7) In most cases $\mathcal{D}(\mathcal{R}(X \sim p_{\text{target}})) - \mathcal{D}(\mathcal{R}(\hat{X} \nsim p_{\text{target}})) > \mathcal{D}(X \sim p_{\text{target}}) - \mathcal{D}(\hat{X} \nsim p_{\text{target}})$. This signifies that

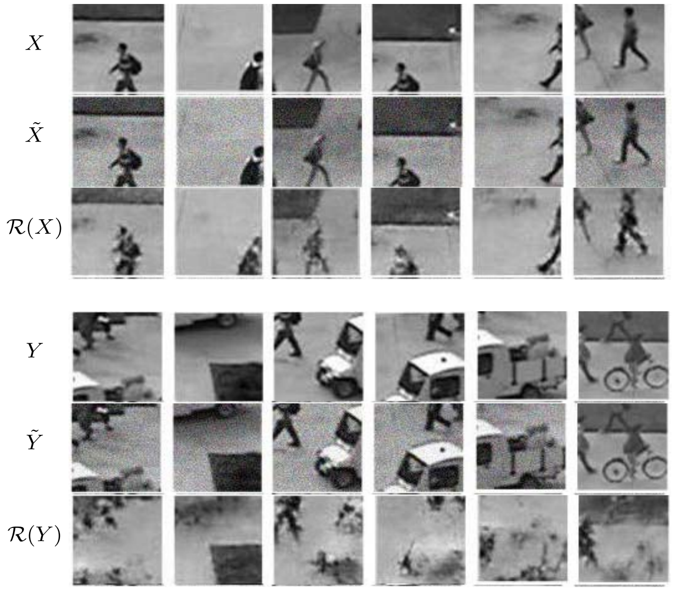


Fig. 3. Inlier (X) and outlier (Y) example outputs of $\mathcal{R}$. Samples are taken from UCSD Ped2. Note that $\mathcal{R}$ is trained on normal patches. $\mathcal{R}$ refines/enhances its input if it is an inlier/normal patch and distorts it if it is an outlier one. $\tilde{X}$ and $\tilde{Y}$ patches are the noisy versions of the inputs.

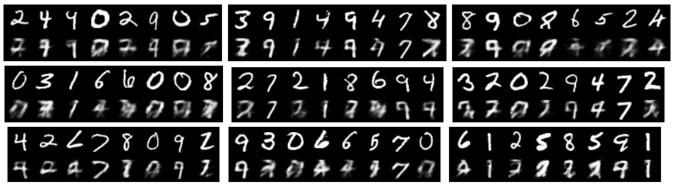


Fig. 4. Each pair of images shows input and output of $\mathcal{R}$ trained to recognize digit "1" from MNIST. Other digits are considered as outliers. $\mathcal{R}$ fails to reconstruct the outliers and distorts them.

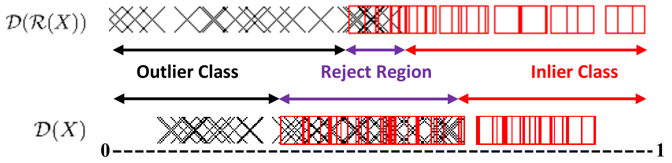


Fig. 5. $\mathcal{R}+\mathcal{D}$ jointly trained for detecting digit "1" as the inlier class. Top shows the $\mathcal{D}(\mathcal{R}(X))$ and bottom the $\mathcal{D}(X)$ scores, generated for 20 inliers (red square) and 20 outliers (× marks). The RR of $\mathcal{R}(X)$ is bigger compared with the RR of $\mathcal{D}(\mathcal{R}(X))$.

the output of $\mathcal{R}$ is more separable than original images. It is because of this fact that $\mathcal{R}$ supports $\mathcal{D}$ for better detection. To further explore this, Fig. 5 shows the score generated as the output of $\mathcal{D}$ for both cases. In some sensitive applications, it is more appropriate to avoid making decisions on difficult cases [58] and leave them for human intervention. These hard-to-decide cases are known to be in the RR. As shown in Fig. 5, the RR of $\mathcal{D}(X)$ is larger than that of $\mathcal{D}(\mathcal{R}(X))$.

C. OCC Using $\mathcal{R}+\mathcal{D}$

Previously, we discussed the details of $\mathcal{R}$ and $\mathcal{D}$ deep networks. As explained, $\mathcal{D}$ works as an OCC model with the support of $\mathcal{R}$. Therefore, the OCC is easy to formulate by using only the $\mathcal{D}$ network (similar to GAN [22]). If $\mathcal{D}(X) > \tau$ (τ is a predefined threshold), X is classified as the target class, and novelty or outlier, otherwise. However, such policy for OCC works satisfactorily (Section IV), we further involve $\mathcal{R}$ in the testing stage for a better performance. To this end, $\mathcal{R}(X,\theta_r)$ is exploited as a refinement procedure for X. θ_r optimized to enhance (by reconstruction) those samples that follow p_{target} while it decimates those that do not follow p_{target}. As a result, we propose to use $\mathcal{D}(\mathcal{R}(X))$ instead of $\mathcal{D}(X)$:

$$\mathrm{OCC}_2(X) = \begin{cases} \text{Target Class}, & \text{if } \mathcal{D}(\mathcal{R}(X)) > \tau \\ \text{Novelty (Outlier)}, & \text{otherwise}. \end{cases} \qquad (6)$$

IV. Experiments

We evaluate the performance and applicability of proposed method on four data sets from two different tasks of *outlier detection in images* and *anomaly detection in videos*. The best results are compared with the state-of-the-art methods.

A. Setup

We implemented all our experiments using Python and TensorFlow [60] on NVIDIA TITAN X. $\mathcal{D}$ and $\mathcal{R}$ and their architecture are explained in supplementary material. The hyperparameters of (5) are set equal to $\lambda_1 = 0.4$ and $\lambda_2 = 0.2$. The batch normalization hyperparameters are defined as $\epsilon = 10^{-6}$ and decay $= 0.9$ [61]. For fair comparisons, we use the same performance metrics and experimental setup as the previous work.

B. Outlier Detection

A wide range of computer vision tasks deals with OCC problems. The performance of machine learning methods drops in the presence of gross outliers, while our method has the capability of learning the shared concept among inliers and identifies the outliers.

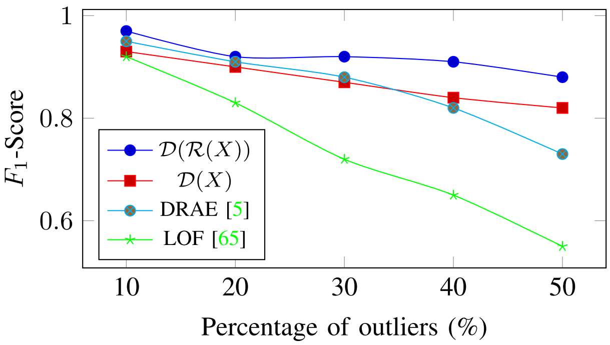


Fig. 6. F_1-score results for different methods on MNIST as a function of percentage of outliers.

1) Outlier Detection Data Sets: We follow [5], [6], and [62] to evaluate the results of our proposed method for the outlier detection task on Modified National Institute of Standards and Technology (MNIST) [63] and Caltech [64] data sets.

MNIST [63] consists of 60K handwritten digits in ten classes (digits "0"–"9"). One of these classes is considered as the inliers class and outliers are randomly sampled from other digits with a proportion of 10%–50%. This procedure is done for all ten classes, and the results are averaged.

Caltech-256 [64] has 256 categories of objects with each having at least 80 images (in total 30 607 images). The same as [6], we repeat this experiment three different times with images from $n \in \{1,3,5\}$ randomly chosen categories to define the inlier class. For categories with more images, the first 150 images are used. The "clutter" category was used to randomly sample outlier instances with a proportion of 50%.

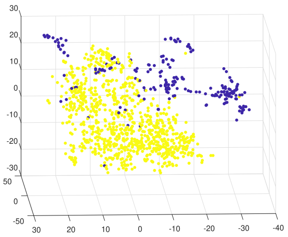
2) Outlier Detection Results: Results on MNIST. $\mathcal{R}$ and $\mathcal{D}$ are jointly learned only on the target class samples. Similar to [5], we use an F_1-score measure for evaluating and comparing our method with others. Fig. 6 compares the F_1-score of different methods by considering various proportions of outlier samples. This experiment confirms that the proposed method [$\mathcal{D}(\mathcal{R}(X))$] has a better performance than other two previous methods (i.e., local outlier factor (LOF) [65] and discriminative reconstructions of auto-encoder (DRAE) [5]). Note, in contrast to the baseline models, our method is not very sensitive to the number of outliers and is able to consistently and successfully detect the outliers. Furthermore, Fig. 6 shows that $\mathcal{D}(X)$ itself can be considered as a successful detector, even better than the previous works. Nevertheless, these results are even further improved when $\mathcal{R}$ module is incorporates, and it refines the samples from the inlier class and decimates the outliers. Therefore, $\mathcal{R}$ helps the distinguishability of the samples. In Fig. 7, 1000 samples of digit "5" are chosen as inliers and 200 samples from other classes as outliers. Fig. 7 shows the t-SNE [59] projection before and after reconstructing using $\mathcal{R}$. As can be seen, $\mathcal{R}(X)$ increases the distinguishability of the two classes, which helps $\mathcal{D}$ to accurately classify them.

Result on Caltech-256: Following [6] for the experimental setup on this data set, the performance of our method is compared with seven previous state-of-the-art methods, designed specifically for outlier detection (results from [6]). Table I shows this comparison with respect to F_1-score and area under the receiver operating characteristic (ROC)

TABLE I

COMPARISONS ON CALTECH-256. WE REPEAT THE EXPERIMENTS THREE TIMES BY CHANGING THE NUMBER OF INLIER CATEGORIES (#CAT), I.E., TAKING INLIERS FROM ONE, THREE, OR FIVE RANDOMLY CHOSEN CATEGORIES. IN ALL THREE SETTINGS, THE OUTLIERS ARE DRAWN RANDOMLY FROM THE CATEGORY 257-CLUTTER WITH THE NUMBER OF OUTLIERS BEING ALWAYS 50% OF THE NUMBER OF INLIERS. THE BEST RESULTS ARE TYPESET IN **BOLD**

	CoP [38]			REAPER [40]			OutlierPursuit [39]			LRR [33]			DPCP [66]			R-graph [6]			ALOCC [1]			Ours $\mathcal{D}(X)$			Ours $\mathcal{D}(\mathcal{R}(X))$		
#Cat	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
AUC	0.90	0.67	0.48	0.81	0.79	0.65	0.83	0.78	0.62	0.90	0.47	0.33	0.78	0.79	0.67	0.94	0.92	0.91	0.94	**0.93**	0.92	0.93	**0.93**	0.91	**0.95**	**0.93**	**0.93**
F_1	0.88	0.71	0.67	0.80	0.78	0.71	0.82	0.77	0.71	0.89	0.67	0.66	0.78	0.77	0.71	0.91	0.88	0.85	**0.92**	0.91	0.90	*0.91*	0.90	0.89	**0.92**	**0.93**	**0.91**



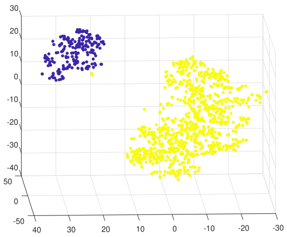


Fig. 7. Visualizing inlier and outlier samples X taken from the MNIST data set. X (left) and $\mathcal{R}(X)$ (right) are visualized in three dimensions based on t-SNE [59] projection. $\mathcal{R}$ is trained on digit "5." Here, Yellow and blue colors indicate inlier and outlier samples, respectively.

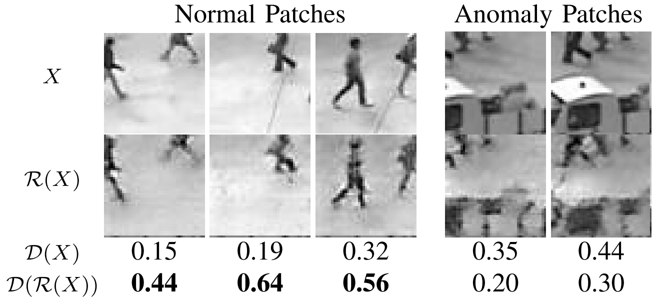

Normal Patches
Anomaly Patches
X
$\mathcal{R}(X)$
$\mathcal{D}(X)$ 0.15 0.19 0.32 0.35 0.44
$\mathcal{D}(\mathcal{R}(X))$ **0.44** **0.64** **0.56** 0.20 0.30

Fig. 8. Sample patches, X, reconstructed using trained $\mathcal{R}$, $\mathcal{R}(X)$ and their anomaly scores (last two rows), a scalar in range [0, 1].

curve (AUC). These results demonstrate that our method is comparable to others, and even in many cases, is superior to them. As it is clear, $\mathcal{D}(X)$ and $\mathcal{D}(\mathcal{R}(X))$ often have a better performance than all other previous methods. Furthermore, with the increase in inlier classes (from 1 to 5 in Table I), the performance of our method does not drop.

C. Video Anomaly Detection

Video anomaly detection is a challenging machine vision task. Due to the temporal characteristics of videos, anomaly detection in videos is even more burdensome than outlier detection in images. We evaluate the proposed method on the University of California, San Diego (UCSD) [67] Ped2 data set. Following previous work, the frame-level performance is used for reporting the performance of our method.

1) Data Sets: UCSD Ped2 data set [67] includes outdoor scenes with static 10-f/s camera and resolution of 240 × 360. The main moving objects in the scenes are pedestrians, and hence, all other objects, such as cars, skateboarders, wheelchairs, or bicycles, are defined as anomalies.

University of Minnesota (UMN) data set includes three scenarios, in which a group of individuals normally and with orderly pace move around. They suddenly start running away, defining the anomalies.

2) Results: Result on UCSD Ped2: We split the video frames to 2-D patches of 30 × 30 pixels (see Fig. 8 for examples). Training frames only consist of normal patches. During testing, test patches are fed to $\mathcal{R}$ and $\mathcal{D}$ networks, some examples of $\mathcal{R}$ outputs are shown in Fig. 8. As it is evident Fig. 8, normal patches are successfully reconstructed by $\mathcal{R}$ while the anomaly patches are distorted and not properly reconstructed. The anomaly likelihood scores, an output of our methods [i.e., $\mathcal{D}(X)$ and $\mathcal{D}(\mathcal{R}(X))$], are reported in the last rows. As the comparison shows, using both networks, i.e., $\mathcal{D}(\mathcal{R}(X))$, results in better distinguishability between normal and anomaly patches. This ultimately leads to a better one-class classifier for anomaly detection. As also reported by the previous work, high false positives define critical challenges for anomaly detection in videos. This means that algorithms often classify many "normal" cases as anomalies'. The left three columns in Fig. 8 show three tough "normal" cases, as the pedestrians are not completely visible (chosen deliberately to illustrate that using $\mathcal{D}(\mathcal{R}(X))$ results in the increase of discriminability compared with $\mathcal{D}(X)$).

Following [69], the frame-level equal error rate (EER) of our method and all others are reported. For the frame-level measure, a frame labeled as "anomaly," if a pixel of it is detected as an anomaly. The results of our methods and other state-of-the-art methods are listed in Table II. As can be seen, the proposed method is comparable with other considered methods, while we use a general-purpose approach that can be used for any type outlier and did not specifically tune for anomaly detection problem in videos (i.e., we did not use spatiotemporal cues for training the network). Note that, other methods, such as deep cascade [12] and deep anomaly [24], takes the advantages of both spatial and temporal complex features, whereas our method benefits only from spatial characteristics. In this experiment, we illustrated that our OCC operates at least as good as the previous methods under a general setting.

Results on UMN: UMN data set is a less challenging data set compared with the UCSD data set. We obtained the good results of ERR = 2.6% and AUC = 0.996, which are comparable with the state-of-the-art (e.g., deep-cascade [12]), even though we are not directly using spatiotemporal features. Table III shows our result in comparison with the state-of-the-art results.

D. Complexity

After training, our method can detect the outliers in nearly real-time during testing time. To show this, we calculate the

TABLE II

FRAME-LEVEL COMPARISONS ON PED2 IN TERMS OF ERRS

MPCCA [68]	MDT [69]	Multi-scale *et al.* [30]	Li *et al.* [11]	Dan Xu *et al.*	ALOCC [1] $\mathcal{D}(X)$	Ours $\mathcal{D}(X)$
30%	24%	30%	18.5%	20%	16%	16%
Ravanbakhsh *et al.* [45]	Ravanbakhsh *et al.* [22]	Dan Xu*et al.* [21]	Sabokrou *et al.* [32]	Deep-cascade [12]	ALOCC[1] $\mathcal{D}(\mathcal{R}(X))$	Ours $\mathcal{D}(\mathcal{R}(X))$
13%	14%	17%	9%	19%	13%	12.5%

TABLE III

COMPARISON OF EER AND AUC ON UMN DATA SET

	Chaotic invariant [70]	SF [69]	Cong *et al.* [10]	Saligrama *et al.* [71]	Li *et al.* [11]	Ours $\mathcal{D}(\mathcal{R}(X))$
EER	5.3	12.6	2.8	3.4	3.7	**2.6**
AUC	99.4	94.9	**99.6**	99.5	99.5	**99.6**

TABLE IV

RUN TIME OF PROPOSED METHOD FOR NOVELTY DETECTION. OUTLIER DETECTION AND ANOMALY DETECTION RESULTS ARE RELATED ON THE CALTECH-256 AND THE UCSD PED2 DATA SETS, RESPECTIVELY

Task	$\mathcal{D}(\cdot)$	$\mathcal{D}(\mathcal{R})(\cdot)$
Outlier Detection (Images/second)	271	122
Anomaly detection (Frames/second)	19.4	10.7

TABLE V

DIFFERENT STOPPING CONDITIONS FOR THE TRAINING PROCEDURE. STOPPING TRAINING BEFORE REACHING A NASH EQUILIBRIUM BETWEEN THE TWO NETWORKS LEADS TO HAVING ONE NETWORK AS THE WINNER (✓✓) AND THE OTHER ONE AS THE LOSER (✗). NASH EQUILIBRIUM WILL BE THE TRADEOFF BETWEEN THE TWO NETWORKS BEING TRAINED (✓)

$\mathcal{R}$	$\mathcal{D}$	Stopping Measure
✓✓	✗	$\|\|\mathcal{R}(X) - X\|\|^2 \leq \rho_1$ and $\forall X \in \mathcal{B} \quad \|\mathcal{D}(X) - 0.5\| \leq \rho_2$
✗	✓✓	$\|\|\mathcal{R}(X) - X\|\|^2 \geq \rho'_1$ and $\forall X \in \mathcal{B} \quad \|\mathcal{D}(X) - 0.5\| \geq \rho'_2$
✓	✓	$\|\|\mathcal{R}(X) - X\|\|^2 \leq \rho''_1$ and $\forall X \in \mathcal{B} \quad \\|\mathcal{D}(X) - 0.5\\| \geq \rho''_2; \quad \rho''_1 \geq \rho_1$ and $\rho''_2 \leq \rho'_2$

run time of $\mathcal{D}(\cdot)$ and $\mathcal{D}(\mathcal{R})(\cdot)$ when run on images or video frames for detecting anomalies. Results are shown in Table IV. As can be seen, anomaly detection in videos is a bit slower compared with when the model is applied to images. This is simply because our method requires dividing video frames to patches and feed them to the network, while for images, the network is given the resized version of the whole images.

E. Discussion

The obtained results demonstrate the capability of $\mathcal{R} + \mathcal{D}$ for detecting the out of target class distribution and confirm that it at least operates and the state-of-the-art methods. The challenging issue is to find the best architecture and carry out an efficient training procedure for such networks. The networks we exploited showed well-enough results, while they may still be improved. For example, by modifying the size/order of conv layers in any of the two networks $\mathcal{R} + \mathcal{D}$, we achieved slightly better results with margins of 0.02–0.04 compared with those reported in Table I.

1) Stopping Criterion: A very important factor in our algorithm is to decide when to stop the joint training of $\mathcal{R} + \mathcal{D}$. If we stop the training very early, the networks may be too immature. On the other hand, overtraining simply leads to confused $\mathcal{R}$ and hence unreliable outputs. We use a subjective criterion to define the stopping condition(s) for $\mathcal{R}$ and $\mathcal{D}$ modules. We stop the training when $\mathcal{R}$ can successfully reconstruct and denoise (noisy) normal images. Under such circumstances, $\mathcal{R}$ will have the ability to successfully fool $\mathcal{D}$. Formally, we stop the training when $\|X - X'\|^2 < \rho$, where ρ is a small positive scalar and $\mathcal{D}$ efficiently classifies its input, i.e., $\forall X \in \mathcal{B}$, $|\mathcal{D}(X) - 0.5| \leq \rho$, where $\mathcal{B}$ is a set of normal and generated samples.

To analyze the stopping criteria in more depth, we need to note that a min–max game between $\mathcal{R}$ and $\mathcal{D}$ is going on, which needs to be stopped when they are in their best possible performance point. In other words, Nash equilibrium [26] needs to happen. By achieving a Nash equilibrium, $\mathcal{R}$ and $\mathcal{D}$ will work decently while not being in their maximum performance point (that can be obtained independently). During training, three interest conditions may occur for $\mathcal{D}$ and $\mathcal{R}$. (1) If $\mathcal{R}$ meets its best performance, i.e., $||\mathcal{R}(X) - X||^2 \leq \rho_1$, $\mathcal{D}$ will be confused to distinguish between generated and real data and possibly generate likelihood probability for both type of data will near 0.5, i.e., $|\mathcal{D}(X) - 0.5| \leq \rho_2$. (2) $\mathcal{D}$ obtains a high accuracy, when also $\mathcal{R}$ reconstructs data well but not and the previous condition, which means $||\mathcal{R}(X) - X||^2 \leq \rho'_1$ where ρ'_1 is close but a bit greater than ρ_1. (3) $\mathcal{R}$ and $\mathcal{D}$ obtain a balanced performance. To reach this condition, i.e., the Nash Equilibrium, both previous conditions need to occur. Finding the stopping point to reach this balance is not an easy task. We used a simple trick by running the joint training when $\mathcal{R}$ was on its maximum performance with respect to condition (1), θ_r and θ_d were save as best parameter settings of $\mathcal{R}$, and networks were continued to be trained until $\mathcal{D}$ satisfy condition (2). Parameters of every network are saved when they were in their best performance. These parameters are used in the testing phase. Table V summarizes these three conditions.

2) Weak Supervision: Knowing that the model is trained in the presence of only target class samples, we can consider this a weak signal for supervision. For dealing with the outlier detection problem, it is reasonable if we assume that the number of outlier samples is very small compared with the inliers in the training set.

In such situations, training the model with a small number of outliers will not significantly harm the model. We examined this setting and created a data set of inlier (90%) and outlier (10%) classes using the Ped2 data set and trained our model. The resulting EER only dropped by a margin

TABLE VI

EVALUATION OF $\mathcal{R}$ ARCHITECTURE WITH DIFFERENT SIZES FOR THE LATENT VARIABLE (DENOTED BY $|Z|$), WITH A BASE ARCHITECTURE SIMILAR TO FIG. 1 OF THE SUPPLEMENTARY MATERIAL. THE LAYER TYPES OF $\mathcal{R}_1$ AND $\mathcal{R}_2$ ARE "CONVENTIONAL AUTO-ENCODER" (I.E., FULLY CONNECTED) OR CONVOLUTIONAL. LAST COLUMN INDICTES THE SIZE OF THE RR INTERVAL FOR $\mathcal{D}(\mathcal{R}(X))$. IN THIS EXPERIMENT, THE NETWORKS ARE TRAINED TO DETECT DIGIT "1" AS THE INLIER (FIG. 5)

Base architecture for $\mathcal{R}$	$\|Z\|$	RR
Conventional Auto-encoder	256	0.400
Conventional Auto-encoder	512	0.270
Conventional Auto-encoder	1024	0.350
CNN Encoder-Decoder	256	0.098
CNN Encoder-Decoder	512	0.110
CNN Encoder-Decoder	1024	0.240
Restricted CNN Encoder-Decoder	256	0.085
Restricted CNN Encoder-Decoder	512	0.087
Restricted CNN Encoder-Decoder	1024	0.140

of 1.3%. However, a simple trick can be used to even avoid this decrease in performance. First, we can learn the network on all available samples, which probably leads to a high false-positive rate (i.e., for the target class). Then, using this network, we can reject all nontarget samples and fine-tune the previously trained network on the remaining samples. Using this trick, we achieved a performance equal to the outcome of the proposed method in the absence of novelty class.

3) Mode Collapse: Mode collapse is a well-known issue [72] in adversarial deep learning frameworks. Such a problem often arises when the generator, instead of learning the whole real-data distribution, only learns a portion of it and subsequently can only generate from a single mode (i.e., it ignores other modes). This issue is solved in our case as $\mathcal{R}$ directly knows about the whole distribution of real data (i.e. it access to all target class samples). Consequently, it implicitly learns the manifold spanned by the target data distribution. As it is also investigated in [54] and [55], auto-encoder GANs are less prone to mode collapse as the generator has access to the pool of real data available in the data set.

4) Architecture of $\mathcal{R}$: As explained in the previous sections, $\mathcal{R}$ helps $\mathcal{D}$ to better detect novelties. Conventional fully connected auto-encoders, AEs with convolution encoder–decoder, and constrained convolutional encoder–decoder are three options, which can be used as the base architecture for $\mathcal{R}$. Another important factor is the size of the latent variable Z. With a sensitivity analysis on Z (i.e., changing its size and restricting it to follow Gaussian distribution), we found that it leads to the largely different length of the rejected region interval. Table VI shows a summary of the outcomes for the rejected region for different architectures. The Gaussian restricted convolutional CNN with the size of the latent variable Z equal to 256 leads to the narrowest RR on the validation set of the MNIST experiment. We defined the RR as the intersection between the boundaries of the target and the novelty classes. Samples lying in this region are very likely to confuse the classifier and be misclassified. If we can push the samples from the two classes away from each other, the RR becomes smaller, and the classifier will induce less error.

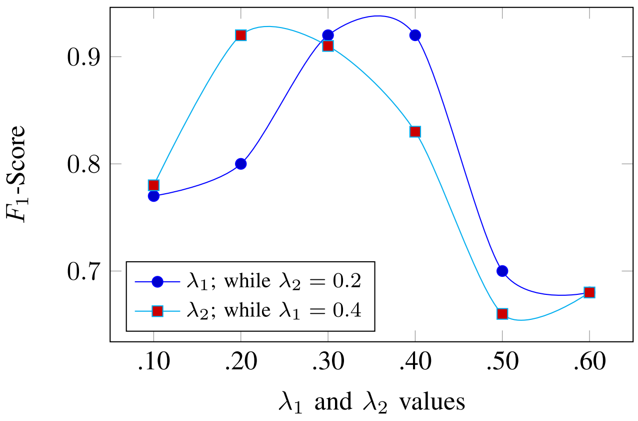


Fig. 9. Comparisons of F_1-scores with respect to different values of λ_1 or λ_2 on the MNIST data set for 30% of outlier samples involved in the experiment.

5) Selecting λ_1 and λ_2: As shown in (5), these two hyper-parameters create a tradeoff between the reconstruction error and the Kullback–Leibler divergence of the latent space of $\mathcal{R}$ and the Gaussian distribution. For $\mathcal{R}$ to efficiently reconstruct the inlier samples, λ_1 should be fairly large. Furthermore, if λ_1 is set a very large value, $\mathcal{R}$ soon becomes an expert to reconstruct inlier samples while $\mathcal{D}$ is yet not properly trained and, hence, will be confused to distinguish between X and $\mathcal{R}(X)$. In addition, we observed that a high value for λ_2 decreases the performance of our method for detecting the novelties. We explored the effect of different values of λ_1 and λ_2 and visualized the results in Fig. 9.

V. CONCLUSION

We introduced an efficient and general method for OCC, learned in an adversarial manner. The proposed method is tailored depending on two networks: $\mathcal{R}$econstructor and $\mathcal{D}$iscriminator. $\mathcal{R}$ learns to simultaneously generate (i.e. reconstruct) samples with the same concept of target class in such a way that $\mathcal{D}$ cannot recognize such samples are reconstructed. After training, $\mathcal{R}$ is able to enhance the target class samples by reconstructing them. At the same time, $\mathcal{R}$ distorts and decimates data if it does not follow the target class distribution. In the testing time, refining samples using $\mathcal{R}$ helps $\mathcal{D}$ in making better discrimination. We have adopted the proposed model for a wide range of applications, such as outlier and anomaly detection in images and videos. Results of different benchmarks illustrate that the proposed OCC is able to detect samples with high divergence from the shared concept among normal samples.

The main limitations of this article involve not being able to efficiently localize the novelty or abnormality in an image or video frame. That is why we had to run our method on a patch-based manner. But this makes the model very slow. Proposing an end-to-end method that localizes the novelty by processing the whole frame at one step can be very useful. Furthermore, similar to almost all GANs, our architecture is difficult to train as finding a good tradeoff for the stopping criterion is not trivial. These two limitations can define directions for future studies.

References

[1] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3379–3388.

[2] K. Lee, K. Lee, K. Min, Y. Zhang, J. Shin, and H. Lee, "Hierarchical novelty detection for visual object recognition," 2018, *arXiv:1804.00722*. [Online]. Available: http://arxiv.org/abs/1804.00722

[3] Q. Chen, R. Luley, Q. Wu, M. Bishop, R. W. Linderman, and Q. Qiu, "AnRAD: A neuromorphic anomaly detection framework for massive concurrent data streams," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1622–1636, May 2018.

[4] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty detection using level set methods," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 576–588, Mar. 2015.

[5] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1511–1519.

[6] C. You, D. P. Robinson, and R. Vidal, "Provable self-representation based outlier detection in a union of subspaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3395–3404.

[7] D. Ienco, R. G. Pensa, and R. Meo, "A semisupervised approach to the detection and characterization of outliers in categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1017–1029, May 2017.

[8] F. Dufrenois, "A one-class kernel Fisher criterion for outlier detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 982–994, May 2015.

[9] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 60–65.

[10] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. CVPR*, Jun. 2011, pp. 3449–3456.

[11] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[12] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1992–2004, Apr. 2017.

[13] E. Epaillard and N. Bouguila, "Variational Bayesian learning of generalized Dirichlet-based hidden Markov models applied to unusual events detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1034–1047, Apr. 2019.

[14] M. M. Moya and D. R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Netw.*, vol. 9, no. 3, pp. 463–474, Apr. 1996.

[15] A. Gardner, A. Krieger, G. Vachtsevanos, and B. Litt, "One-class novelty detection for seizure analysis from intracranial EEG," *J. Mach. Learn. Res.*, vol. 7, pp. 1025–1044, Jun. 2006.

[16] S. S. Khan and M. G. Madden, "One-class classification: Taxonomy of study and review of techniques," *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 345–374, Jun. 2014.

[17] Z. Ghafoori, S. M. Erfani, S. Rajasegarar, J. C. Bezdek, S. Karunasekera, and C. Leckie, "Efficient unsupervised parameter estimation for one-class support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5057–5070, Oct. 2018.

[18] L. Livi, A. Sadeghian, and W. Pedrycz, "Entropic one-class classifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3187–3200, Dec. 2015.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[20] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[21] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," 2015, *arXiv:1510.01553*. [Online]. Available: http://arxiv.org/abs/1510.01553

[22] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 1577–1581.

[23] W. Lawson, E. Bekele, and K. Sullivan, "Finding anomalies with generative adversarial networks for a patrolbot," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 12–13.

[24] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Comput. Vis. Image Understand.*, vol. 172, pp. 88–97, Jul. 2018.

[25] M. Sabokrou, M. Fathy, Z. Moayed, and R. Klette, "Fast and accurate detection and localization of abnormal behavior in crowded scenes," *Mach. Vis. Appl.*, vol. 28, no. 8, pp. 965–985, Nov. 2017.

[26] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[28] M. Markou and S. Singh, "Novelty detection: A review—Part 1: Statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, 2003.

[29] M. Sabokrou, M. Fathy, and M. Hoseini, "Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder," *Electron. Lett.*, vol. 52, no. 13, pp. 1122–1124, Jun. 2016.

[30] M. Bertini, A. Del Bimbo, and L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Comput. Vis. Image Understand.*, vol. 116, no. 3, pp. 320–329, Mar. 2012.

[31] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.

[32] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 56–62.

[33] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 663–670.

[34] E. Adeli, K.-H. Thung, L. An, F. Shi, and D. Shen, "Robust feature-sample linear discriminant analysis for brain disorders diagnosis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 658–666.

[35] M. Sabokrou, M. Khalooei, and E. Adeli, "Self-supervised representation learning via neighborhood-relational encoding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8010–8019.

[36] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 1–8.

[37] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining Knowl. Discovery*, vol. 8, no. 3, pp. 275–300, May 2004.

[38] M. Rahmani and G. Atia, "Coherence pursuit: Fast, simple, and robust principal component analysis," 2016, *arXiv:1609.04789*. [Online]. Available: http://arxiv.org/abs/1609.04789

[39] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2496–2504.

[40] G. Lerman, M. B. McCoy, J. A. Tropp, and T. Zhang, "Robust computation of linear models by convex relaxation," *Found. Comput. Math.*, vol. 15, no. 2, pp. 363–410, Apr. 2015.

[41] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 17–31, Apr. 2007.

[42] B. Antic and B. Ommer, "Video parsing for abnormality detection," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2415–2422.

[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[44] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.* Cham, Switzerland: Springer, 2017, pp. 146–157.

[45] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," 2017, *arXiv:1706.07680*. [Online]. Available: http://arxiv.org/abs/1706.07680

[46] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016, *arXiv:1611.07004*. [Online]. Available: http://arxiv.org/abs/1611.07004

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Berlin, Germany: Springer, 2015, pp. 234–241.

[48] M. Sabokrou *et al.*, "Avid: Adversarial visual irregularity detection," in *Proc. Asian Comput. Vis. Conf.*, 2018, pp. 488–505.

[49] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

[50] N. Divakar and R. V. Babu, "Image denoising via CNNs: An adversarial approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1076–1083.

[51] P. Hand and V. Voroninski, "Global guarantees for enforcing deep generative priors by empirical risk," 2017, *arXiv:1705.07576*. [Online]. Available: http://arxiv.org/abs/1705.07576

[52] H. Xu *et al.*, "Unsupervised anomaly detection via variational autoencoder for seasonal KPIs in Web applications," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 187–196.

[53] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1558–1566.

[54] D. Bang and H. Shim, "MGGAN: Solving mode collapse using manifold guided training," 2018, *arXiv:1804.04391*. [Online]. Available: http://arxiv.org/abs/1804.04391

[55] A. Srivastava, L. Valkoz, C. Russell, M. U. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3308–3318.

[56] T. Che, Y. Li, A. Paul Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," 2016, *arXiv:1612.02136*. [Online]. Available: http://arxiv.org/abs/1612.02136

[57] J. M. Joyce, "Kullback–Leibler divergence," in *Proc. Int. Encyclopedia Stat. Sci.* Cham, Switzerland: Springer, 2011, pp. 720–722.

[58] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006.

[59] L. van der Maaten and G. Hinton, "Visualizing non-metric similarities in multiple maps," *Mach. Learn.*, vol. 87, no. 1, pp. 33–55, Apr. 2012.

[60] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: http://arxiv.org/abs/1603.04467

[61] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[62] J. Liu, Z. Lian, Y. Wang, and J. Xiao, "Incremental kernel null space discriminant analysis for novelty detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 792–800.

[63] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradientbased learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998

[64] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

[65] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Rec.*, vol. 29, 2000, pp. 93–104.

[66] M. C. Tsakiris and R. Vidal, "Dual principal component pursuit," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 10–18.

[67] A. Chan and N. Vasconcelos, "UCSD pedestrian dataset," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 909–926, May 2008.

[68] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2921–2928.

[69] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1975–1981.

[70] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1419–1426.

[71] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2112–2119.

[72] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.