Adversarially Learned One-Class Classifier for Novelty Detection

Mohammad Sabokrou[1], Mohammad Khalooei[2], Mahmood Fathy[1], Ehsan Adeli[3]
[1]Institute for Research in Fundamental Sciences
[2]Amirkabir University of Technology [3]Stanford University

Abstract

Novelty detection is the process of identifying the observation(s) that differ in some respect from the training observations (the target class). In reality, the novelty class is often absent during training, poorly sampled or not well defined. Therefore, one-class classifiers can efficiently model such problems. However, due to the unavailability of data from the novelty class, training an end-to-end deep network is a cumbersome task. In this paper, inspired by the success of generative adversarial networks for training deep models in unsupervised and semi-supervised settings, we propose an end-to-end architecture for one-class classification. Our architecture is composed of two deep networks, each of which trained by competing with each other while collaborating to understand the underlying concept in the target class, and then classify the testing samples. One network works as the novelty detector, while the other supports it by enhancing the inlier samples and distorting the outliers. The intuition is that the separability of the enhanced inliers and distorted outliers is much better than deciding on the original samples. The proposed framework applies to different related applications of anomaly and outlier detection in images and videos. The results on MNIST and Caltech-256 image datasets, along with the challenging UCSD Ped2 dataset for video anomaly detection illustrate that our proposed method learns the target class effectively and is superior to the baseline and state-of-the-art methods.

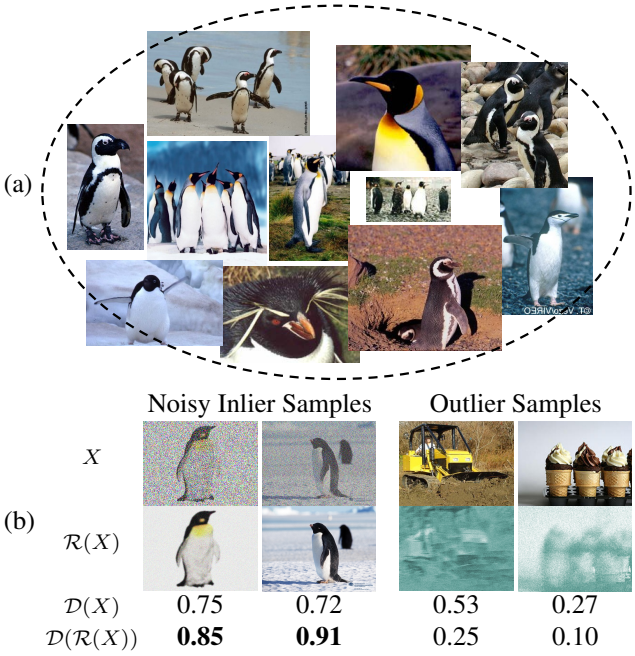

Figure 1. Example outputs of the proposed model, trained to detect Penguins (a), in response to inlier and outlier samples (b). The first row of (b) shows some example images, and the second row contains the output of the $\mathcal{R}$ network on them, *i.e.*, $\mathcal{R}(X)$. As can be seen, $\mathcal{R}$ enhanced the inlier samples (even in the presence of noise) but distorted the outliers. Two last rows show the score of $\mathcal{D}$ applied to X and $\mathcal{R}(X)$, respectively. $\mathcal{R}(X)$ is indeed more separable than only using the original input image, X.

1. Introduction

Novelty detection is the process of identifying the new or unexplained set of data to determine if they are within the norm (*i.e.*, inliers) or outside of it (*i.e.*, outliers). Novelty refers to the unusual, new observations that do not occur regularly or is simply different from the others. Such problems are especially of great interest in computer vision studies, as they are closely related to outlier detection [45, 50], image denoising [8], anomaly detection in images [10, 23] and videos [38]. Novelty detection can be portrayed in the context of one-class classification [30, 13, 18], which aims to build classification models when the negative class is absent, poorly sampled or not well defined. As such, the negative class can be considered as the novelty (*i.e.*, outlier or anomaly), while the positive (or target) class is well characterized by instances in the training data.

To accurately chart the intrinsic geometry of the positive class, the first step is to efficiently represent the data in a way that can entangle more or less the different explanatory factors of variation in the data. Recently, deep learning approaches have gained immense success in representing visual data for various vision-based applications [41, 42], especially in cases that they are trained in an end-to-end

fashion. However, for novelty detection or one-class classification applications, due to unavailability of data from the negative class, training an end-to-end deep network is not straightforward. Some efforts have been made, in recent years, to benefit from deep features in learning one-class classifiers [46, 38, 33, 20, **?**, **?**], few of which could train an end-to-end feature learning and classification model.

Inspired by the recent developments in generative adversarial networks (GANs) [14], we propose an end-to-end model for one-class classification and apply it to different applications including outlier detection, novelty detection in images and anomaly event detection in videos. The proposed architecture, similar to GANs, comprises two modules, which compete to learn while collaborating with each other for the detection task. The first module (denoted as $\mathcal{R}$) refines the input and gradually injects discriminative material into the learning process to make the positive and novelty samples (*i.e.*, inliers, and outliers) more separable for the detector, the second module (referred to as $\mathcal{D}$).

These two networks are adversarially and unsupervisedly learned using the training data, which is composed of only the target class. Specifically, $\mathcal{R}$ learns to reconstruct the positive samples and tries to fool the detector (*i.e.*, $\mathcal{D}$). Whereas, $\mathcal{D}$ learns to distinguish original (positive) samples from the reconstructed ones. In this way, $\mathcal{D}$ learns merely the concept characterized by the space of all positive samples, and hence it can be used for distinguishing between positive and novelty classes. On the other hand, $\mathcal{R}$ learns to efficiently reconstruct the positive samples, while for negative (or novelty) samples it is unable to reconstruct the input accurately, and hence, for negative samples it acts as a decimator (or informally a distorter). In the testing phase, $\mathcal{D}$ operates as the actual novelty detector, while $\mathcal{R}$ improves the performance of the detector by adequately reconstructing the positive or target samples and decimating (or distorting) any given negative or novelty samples. Fig. 1 depicts example inputs and outputs of both $\mathcal{R}$ and $\mathcal{D}$ networks for a model trained to detect images of Penguins.

In summary, the main contributions of this paper are as follows: (1) We propose an end-to-end deep network for learning one-class classifier learning. To the best of our knowledge, this article is one of the firsts to introduce an end-to-end network for one-class classification. (2) Almost all approaches based on GANs in the literature [31] discard either the generator or the discriminator (analogous to $\mathcal{R}$ and $\mathcal{D}$, respectively, in our architecture) after training. Only one of the trained models is used, while our setting is more efficient and benefits from both trained modules to collaborate in the testing stage. (3) Our architecture learns the model in the complete absence of any training samples from the novelty class and achieves state-of-the-art performance in different applications, such as outlier detection in images and anomaly event detection in videos.

2. Related Works

One-class classification is closely related to rare event detection, outlier detection/removal, and anomaly detection. All these applications share the search procedure for a novel concept, which is scarcely seen in the data and hence can all be encompassed by the umbrella term *novelty detection*. Consequently, a wide range of real-world applications can be modeled by one-class classifiers. Conventional research often models the target class, and then rejects samples not following this model. Self-representation [45, 50, 10, 36] and statistical modeling [27] are the commonly used approaches for this task. For data representation, low level features [4], high level features (*e.g.*, trajectories [29]), deeply learned features [46, 37, 38] are used in the literature. A brief review of state-of-the-art novelty detection methods especially the ones based on adversarial learning in deep networks is provided in this section.

Self-Representation. Several previous works show that self-representation is a useful tool for outlier or novelty event detection. For instance, [10, 36] proposed self-representation techniques for video anomaly detection and outlier detection through learning a sparse representation model, as a measure for separating inlier and outlier samples. It is assumed that outlier or novel samples are not sparsely represented using the samples from the target class. In some other works (like [46, 10]), testing samples are reconstructed using the samples from the target class. The decision if it is an inlier or outlier (novel) is made based on the reconstruction error, *i.e.*, high reconstruction error for a sample indicates that it is more probably an outlier sample. In another work, Liu *et al.* [24] proposed to use a low-rank self-representation matrix in place of a sparse representation, penalized by the sum of unsquared self-representation errors. This penalization leads to more robustness against outliers (similar to [2]). Similarly, auto-encoders are also exploited to model and measure the reconstruction error for the related tasks of outlier removal and video anomaly detection, in [36, 46].

Statistical Modeling. More conventional methods tend to model the target class using a statistical approach. For instance, after extracting features from each sample, a distribution function is fit on the samples from the target class, and samples far from this distribution are considered as outliers or novelty (*e.g.*, [12, 49, 27]). In another work, Rahmani and Atia [32] proposed an algorithm termed Coherence Pursuit (CoP) for Robust Principal Component Analysis (RPCA). They assumed that the inlier samples have high correlations and can be spanned in low dimensional subspaces, and hence they have a strong mutual coherence with a large number of data points. As a result, the outliers either do not accord with the low dimensional subspace or form small clusters. Also, a method proposed in [48], OutlierPursuit, used convex optimization techniques to solve the PCA problem with robustness to corrupted entries, which led to the develop-

ment of many recent methods for PCA with robustness to outliers. Lerman *et al*. [22] described a convex optimization problem for detecting the outliers and called it REAPER, which can reliably fit a low-dimensional model to the target class samples.

Deep Adversarial Learning. In the recent years, GANs [14, 39] have shown outstanding success in generating data for learning models. They have also been extended to classification models even in the presence of not enough labeled training data (*e.g*., in [20, 40, 34]). They are based on a two-player game between two different networks, both concurrently trained in an unsupervised fashion. One network is the generator (G), which aims at generating realistic data (*e.g*., images), while the second network poses as the discriminator (D), and tries to discriminate real data from the data generated by G. One of the different types of GANs, closely related to our work, is the conditional GANs, in which G takes an image X as the input and generates a new image X'. Whereas, D tries to distinguish X from X', while G tries to *fool* D producing more and more realistic images. Very recently Isola *et al*. [17] proposed an "Image-to-image translation" framework based on conditional GANs, where both G and D are conditioned on the real data. They showed that a U-Net encoder-decoder [35] with skip connections could be used as the generator coupled with a patch-based discriminator to transform images with respect to different representations. In a concurrent work, [33] proposed to learn the generator as a reconstructor of normal events, and hence if it cannot properly reconstruct a chunk of the input frames, that chunk is considered an anomaly. However, in our work, the first module (*i.e*., $\mathcal{R}$) not only reconstructs the target class, but it also helps to improve the performance for the model on any given testing image, by refining samples belonging to the target class, and decimating/distorting the anomaly or outlier samples.

3. Proposed Approach

The proposed one-class classification framework is composed of two main modules: (1) Network $\mathcal{R}$, and (2) Network $\mathcal{D}$. The former acts as a prepossessing and $\mathcal{R}$efinement (or $\mathcal{R}$econstruction) step, while the latter performs the $\mathcal{D}$iscrimination (or $\mathcal{D}$etection). These two networks are learned in an adversarial and unsupervised manner, within an end-to-end setting. In this section, we present a detailed overview of both. The overall schema of the proposed approach is shown in Fig. 2. It can be seen that $\mathcal{R}$ reconstructs its input, X, generates X', and tries to fool $\mathcal{D}$ so that it speculates that the reconstructed sample is the original data, not a reconstructed sample. On the other hand, $\mathcal{D}$ has access to the original set of data and is familiar with their concept. Hence it will reject the reconstructed samples. These two networks play a game, and after the training stage, in which samples from the target class are presented to the model, $\mathcal{R}$ will become an expert to reconstruct the samples from the target class with a minimum error to successfully fool $\mathcal{D}$. The training procedure leads to a pair of networks, $\mathcal{R}$ and $\mathcal{D}$, which both learn the distribution of the target class. These two modules are trained in a GAN-style adversarial learning framework, forming an end-to-end framework for one-class classification for novelty detection. To make the proposed method more robust against noise and corrupt input samples, a Gaussian noise (denoted by η in Fig. 2) is added to the input training samples and fed to $\mathcal{R}$. Detailed descriptions of each module and the overall training/testing procedures are described in the following subsections.

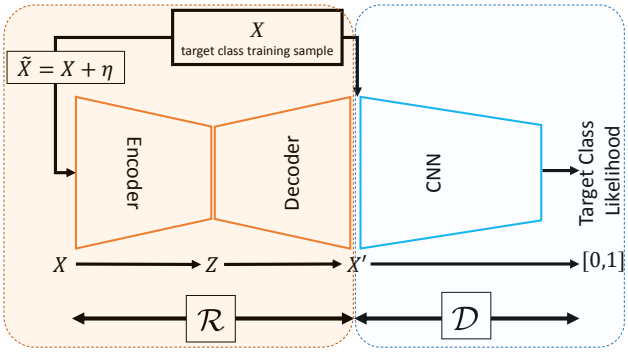

Figure 2. Overview of the proposed structure for one-class classification framework. $\mathcal{R}$ and $\mathcal{D}$ are two modules of the model, which are adversarially learned. $\mathcal{R}$ is optimized to reconstruct samples belonging to the target class, while it works as a decimator function for outlier inputs, whereas $\mathcal{D}$ classifies the input data positive (target) and negative (outlier or anomaly). $\mathcal{D}(\mathcal{R}(X))$ measures the likelihood of the given input sample belonging to the target class.

3.1. $\mathcal{R}$ Network Architecture

It is previously [45, 36] investigated that the reconstruction error of an auto-encoder, trained on samples from the target class, is a useful measure for novelty sample detection. Since the auto-encoder is trained to reconstruct target class samples, the reconstruction error for negative (novelty) samples would be high. We use a similar idea, but in contrast, we do not use it for the detection or the discrimination task. Our proposed model uses the reconstructed image to train another network for the discrimination task.

To implement the $\mathcal{R}$ network, we train a decoder-encoder Convolutional Neural Network (CNN) on samples from the target class to map any given input sample to the target concept. As a result, $\mathcal{R}$ will efficiently reconstruct the samples that share a similar concept as the trained target class, while for outlier or novelty inputs, it poorly reconstructs them. In other words, $\mathcal{R}$ enhances the inliers (samples from the target class), while it destructs or decimates the outliers, making it easier for the discriminator to separate the outliers from the vast pool of inliers. Fig. 3 shows the structure of $\mathcal{R}$

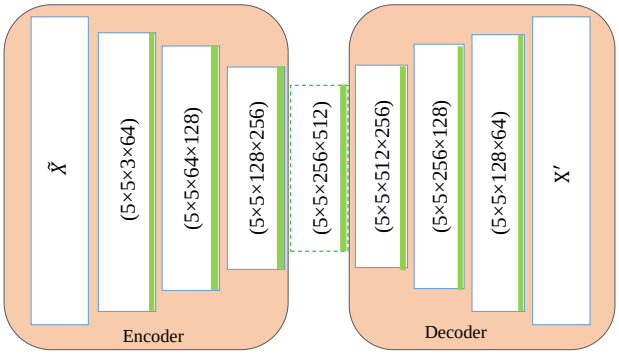

Figure 3. $\mathcal{R}$ network architecture, composed of encoding (first part) and decoding (second part) layers. The properties of each layer are indicated with four hyperparameters in this order: (first dimension of the kernel $\times$ the second dimension of the kernel $\times$ the number of input channels $\times$ the number of output channels).

architecture, which includes several convolution layers (as the encoder), followed by several deconvolution layers (for the decoding purpose). For improving the stability of the network similar to [31], we do not use any pooling layers in this network. Eventually, $\mathcal{R}$ learns the concept shared in the target class to reconstruct its inputs based on that concept. Also, after each convolutional layer, a batch normalization [16] operation is exploited, which adds stability to our structure.

3.2. $\mathcal{D}$ Network Architecture

The architecture for $\mathcal{D}$ is a sequence of convolution layers, which are trained to eventually distinguish the novel or outlier sample, without any supervision. Fig. 4 shows the details of this network's architecture. $\mathcal{D}$ outputs a scalar value, relative to the likelihood of its input following the distribution spanned by the target class, denoted by p_t. Therefore, its output can be considered as a target likelihood score for any given input.

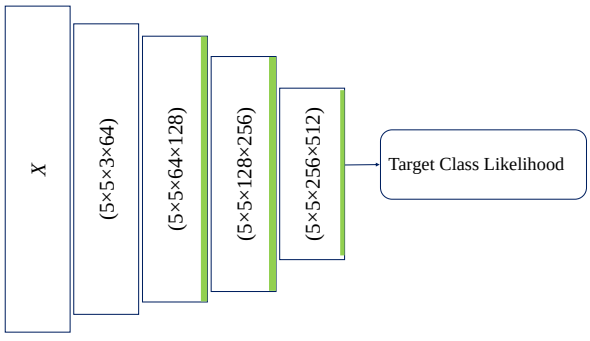

Figure 4. $\mathcal{D}$ network architecture, which determines if its input is from the target class or it is an outlier or novelty. Properties of the layers are denoted similarly to Fig. 3.

3.3. Adversarial Training of $\mathcal{R}$+$\mathcal{D}$

As mentioned in section 2, Goodfellow *et al.* [14] introduced an efficient way for adversarial learning of two networks (denoted by Generator (G) and Discriminator (D)), called Generative Adversarial Networks (GANs). GANs aim to generate samples that follow the same distribution as the real data, through adversarial learning of the two networks. G learns to map any random vector, Z from a latent space following a specific distribution, p_z, to a data sample that follows the real data distribution (p_t in our case), and D tries to discriminate between actual data and the fake data generated by G. Generator and Discriminator are learned in a two-player mini-max game, formulated as:

$$\min_G \max_D \Big(\mathbb{E}_{X\sim p_t}[\log(D(X))] + \mathbb{E}_{Z\sim p_z}[\log(1-D(G(Z)))]\Big). \tag{1}$$

In a similar way, we train the $\mathcal{R}$+$\mathcal{D}$ neural networks in an adversarial procedure. In contrast to the conventional GAN, instead of mapping the latent space Z to a data sample with the distribution p_t, $\mathcal{R}$ maps

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

in which η is the added noise sampled from the normal distribution with standard deviation σ, $\mathcal{N}(0, \sigma^2\mathbf{I})$. From now on, the noise model is denoted by $\mathcal{N}_\sigma$ for short. This statistical noise is added to input samples to make $\mathcal{R}$ robust to noise and distortions in the input images, in the training stage. As mentioned before, p_t is the supposed distribution of the target class. $\mathcal{D}$ is aware of p_t, as it is exposed to the samples from the target class. Therefore, $\mathcal{D}$ explicitly decides if $\mathcal{R}(\tilde{X})$ follows p_t or not. Accordingly, $\mathcal{R}$+$\mathcal{D}$ can be jointly learned by optimizing the following objective:

$$\min_{\mathcal{R}} \max_{\mathcal{D}} \Big(\mathbb{E}_{X\sim p_t}[\log(\mathcal{D}(X))] + \mathbb{E}_{\tilde{X}\sim p_t+\mathcal{N}_\sigma}[\log(1-\mathcal{D}(\mathcal{R}(\tilde{X})))]\Big), \tag{3}$$

Based on the above objective (similar to GAN), network $\mathcal{R}$ generates samples with the probability distribution of p_t, and as a result its own distribution is given by $p_r \sim \mathcal{R}(X \sim p_t; \theta_r)$, where θ_r is the parameter of the $\mathcal{R}$ network. Therefore, we want to maximize $p_t(\mathcal{R}(X \sim p_t; \theta_r))$.

To train the model, we calculate the loss $\mathcal{L}_{\mathcal{R}+\mathcal{D}}$ as the loss function of the joint network $\mathcal{R}$+$\mathcal{D}$. Besides, we need $\mathcal{R}$'s output to be close to the original input image. As a result, an extra loss is imposed on the output of $\mathcal{R}$:

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

Therefore, the model is optimized to minimize the loss function:

$$\mathcal{L} = \mathcal{L}_{\mathcal{R}+\mathcal{D}} + \lambda\mathcal{L}_{\mathcal{R}}, \tag{5}$$

where $\lambda > 0$ is a trade-off hyperparameter that controls the relative importance of the two terms. One of the challenging issues for training $\mathcal{R}$+$\mathcal{D}$ is defining an appropriate stopping criterion. Analyzing the loss functions of $\mathcal{R}$ and $\mathcal{D}$ modules to excerpt a stopping criterion based on is a burdensome task, and hence we use a subjective criterion. The training procedure is stopped when $\mathcal{R}$ successfully maps noisy images to clean images carrying the concept of the target class (*i.e.*, favorably fools the $\mathcal{D}$ module). Consequently, we have stopped the training of networks, when $\mathcal{R}$ can reconstruct its input with minimum error (*i.e.*, $\|X - X'\|^2 < \rho$, where ρ is a small positive number).

After a joint training of the $\mathcal{R}$+$\mathcal{D}$ network, the behavior of each single one of them can be interpreted as follows:

- $\mathcal{R}(X \sim p_t + \eta) \longrightarrow X' \sim p_t$, where $\|X - X'\|^2$ is minimized. This is because θ_r is optimized to reconstruct those inputs that follow the distribution p_t. Note that $\mathcal{R}$ is trained and operates similar to denoising auto-encoders [44] or, denoising CNNs [11], and its output will be a refined version of the input data. See Figures 1 and 5 for some samples of its reconstructed outputs.

- For any given outlier or novelty sample (denoted by $\hat{X}$) that does not follow p_t, $\mathcal{R}$ is confused and maps it into a sample, $\hat{X}'$, with an unknown probability distribution, $p_?$, (*i.e.*, $\mathcal{R}(\hat{X} \nsim p_t + \eta) \longrightarrow \hat{X}' \sim p_?$). In this case, $\|\hat{X} - \hat{X}'\|^2$ cannot become very small or close to zero. This is because $\mathcal{R}$ was not trained on the novelty concept and cannot reconstruct it accordingly (similar to [33]). Therefore, as a side effect, $\mathcal{R}$ decimates the outliers. As an example, Fig. 6 shows samples of a different concept being fed to $\mathcal{R}$ of a network trained to detect digit "1".

- We can expect that $\mathcal{D}(X' \sim p_t) > \mathcal{D}(\hat{X}' \nsim p_t)$, since $\mathcal{D}$ is trained to detect samples from the distribution p_t.

- It is interesting to note that in most cases $\mathcal{D}(\mathcal{R}(X \sim p_t)) - \mathcal{D}(\mathcal{R}(\hat{X} \nsim p_t)) > \mathcal{D}(X \sim p_t) - \mathcal{D}(\hat{X} \nsim p_t)$. This signifies that 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. 7 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 [5], and leave them for human intervention. These hard-to-decide cases are known to be in the reject region. As shown in Fig. 7 the reject region of $\mathcal{D}(X)$ is larger than that of $\mathcal{D}(\mathcal{R}(X))$.

3.4. $\mathcal{R}$+$\mathcal{D}$: One-Class Classification

In the previous subsection, characteristics of both $\mathcal{R}$ and $\mathcal{D}$ networks are explained in details. As discussed, $\mathcal{D}$ acts as the novelty detector, and benefits the support of $\mathcal{R}$. Hence, the One-Class Classifier (OCC) can be simply formulated by only using the $\mathcal{D}$ network (similar to [33]) as:

$$\text{OCC}_1(X) = \begin{cases} \text{Target Class} & \text{if } \mathcal{D}(X) > \tau, \\ \text{Novelty (Outlier)} & \text{otherwise,} \end{cases} \quad (6)$$

where τ is a predefined threshold. Although the above policy for novelty detection works as well as the state-of-the-art methods (explained in details in the Section 4), we further propose to incorporate $\mathcal{R}$ in the testing stage. To this end, $R(X, \theta_r)$ is used as a refinement step for X, in which θ_r is the trained model for the $\mathcal{R}$ module. θ_r is trained to reconstruct and enhance samples that follow p_t (*i.e.*, are from the target class). Consequently, instead of $\mathcal{D}(X)$ we use $\mathcal{D}(\mathcal{R}(X))$. Eq. (7) provides a summary of our proposed once-class classification scheme:

$$\text{OCC}_2(X) = \begin{cases} \text{Target Class} & \text{if } \mathcal{D}(\mathcal{R}(X)) > \tau, \\ \text{Novelty (Outlier)} & \text{otherwise.} \end{cases} \quad (7)$$

4. Experiment Results

In this section, the proposed method is evaluated on three different image and video datasets. The performance results are analyzed in details and are compared with state-of-the-art techniques. To show the generality and applicability of the proposed framework for a variety of tasks, we test it for detection of *(1) Outlier images*, and *(2) Video anomalies*.

4.1. Setup

All the reported results in this section are from our implementation using the TensorFlow framework [1], and Python ran on an NVIDIA TITAN X. The detailed structures of $\mathcal{D}$ and $\mathcal{R}$ are explained in details in Sections 3.2 and 3.1, respectively. These structures are kept fixed for different tasks, and λ in Eq. (5) is set equal to 0.4. The hyperparameters of batch normalization (as in [16]) are set as $\epsilon = 10^{-6}$ and decay$= 0.9$.

4.2. Outlier Detection

As discussed earlier, many computer vision applications face considerable amounts of outliers, since they are common in realistic vision-based training sets. On the other hand, machine learning methods often experience considerable performance degradation in the presence gross outliers, if they fail to deal with processing the data contaminated by noise and outliers. Our method can learn the shared concept among all inlier samples, and hence identify the outliers. Similar to [25, 50, 45], we evaluate the performance of our outlier detection method using MNIST[1] [21] and Caltech[2] [15] datasets.

[1] Available at http://yann.lecun.com/exdb/mnist/

[2] Available at http://www.vision.caltech.edu/Image_Datasets/Caltech256/

Figure 5. Examples of the output of $\mathcal{R}$ for several inlier and outlier samples from the UCSD Ped2 dataset: $\mathcal{R}$ is learned on normal patches. Left and right samples show the inlier (*i.e.*, target) and outlier (*i.e.*, novelty) samples, respectively. As can be seen, the network $\mathcal{R}$ enhances its input and shows to be robust to the noise present in its input. First row: Patch contaminated by some Gaussian noise; Second row: Original patches; Third row: The output of $\mathcal{R}$ on the noisy samples.

Figure 6. Outputs of $\mathcal{R}$ trained to detect digit "1" on MNIST dataset. Samples from classes "6" and "7" are given to the model as outliers. $\mathcal{R}$ failed to reconstruct them and fundamentally distorted them. In each pair of the images, the first one is the original image and the second one is the output of $\mathcal{R}$.

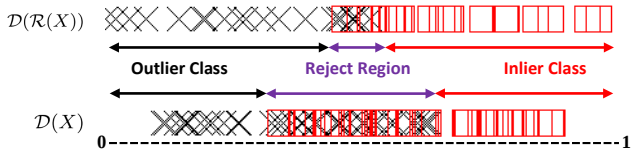

Figure 7. $\mathcal{R}$+$\mathcal{D}$ is trained on inlier samples (digit "1") from MNIST dataset. Top: $\mathcal{D}(\mathcal{R}(X))$ scores; Bottom: $\mathcal{D}(X)$ scores. The scores are generated on 20 inlier and 20 outlier samples. The red squares indicate inlier samples, while $\times$ marks are representatives of the outliers. Reject region for $\mathcal{R}(X)$ is larger than that of $\mathcal{D}(\mathcal{R}(X))$.

4.2.1 Outlier Detection Datasets

MNIST: This dataset [21] includes 60,000 handwritten digits from "0" to "9". Each of the ten categories of digits is taken as the target class (*i.e.*, inliers), and we simulate outliers by randomly sampling images from other categories with a proportion of 10% to 50%. This experiment is repeated for all of the ten digit categories.

Caltech-256: This dataset [15] contains 256 object categories with a total of 30,607 images. Each category has at least 80 images. Similar to previous works [50], we repeat the procedure three times and use images from $n \in \{1, 3, 5\}$ randomly chosen categories as inliers (*i.e.*, target). The first 150 images of each category are used, if that category has more than 150 images. A certain number of outliers are randomly selected from the "clutter" category, such that each experiment has exactly 50% outliers.

4.2.2 Outlier Detection Results

Result on MNIST: The joint network $\mathcal{R}$+$\mathcal{D}$ is trained on images of the target classes, in the absence of outlier samples. Following [45], we also report the F_1-score as a measure to evaluate the performance of our method and compare it with others. Fig. 8 shows the F_1-score of our method (and the state-of-the-art methods) as a function of the portion of outlier samples. As can be seen, our method (*i.e.*, $\mathcal{D}(\mathcal{R}(X))$) operates more efficient than the other two-state-of-the-art methods (LOF [7] and DRAE [45]). Also, it is important to note that with the increase in the number of outliers, our method operates consistently robust and successfully detects the outliers, while the baseline methods fail to detect the outliers as their portion increases. Furthermore, one interesting finding of these results is that, based in Fig. 8, $\mathcal{D}(X)$ itself can operate successfully well, and outperform the state-of-the-art methods. Nevertheless, it is even improved more when we incorporate $\mathcal{R}$ module, as it modifies the samples (*i.e.*, refines the samples from the target class, and decimates the ones coming from an outlier concept) and helps distinguishability of the samples.

Result on Caltech-256: In this experiment, similar setup as in [50] is used, and we compare our method with [50] and 6 other methods therein designed specifically for detecting outliers, including Coherence Pursuit (CoP) [32], OutlierPursuit [48], REAPER [22], Dual Principal Component Pursuit (DPCP) [43], Low-Rank Representation (LRR) [24], OutRank [28]. The results are listed in Table 1, which are comprised of F_1-score and area under the ROC curve (AUC). The results of other methods are borrowed from [50]. This table confirms that our proposed method performs at least as well as others, while in many cases it is superior to

Table 1. Results on the Caltech-256 dataset: Inliers are taken to be images of one, three, or five randomly chosen categories, and outliers are randomly chosen from category 257-clutter. **Two first rows:** Inliers are from one category of images, with 50% portion of outliers; **Two second rows:** Inliers are from three categories of images, with 50% portion of outliers; **Two last rows:** Inliers come from five categories of images, while outliers compose 50% of the samples. The last two columns have the results or our methods, $\mathcal{D}(X)$ and $\mathcal{D}(\mathcal{R}(X))$, respectively. Note that in each row the best result is typeset in **bold** and the second best in *italic* typeface.

	CoP [32]	REAPER [22]	OutlierPursuit [48]	LRR [24]	DPCP [43]	R-graph [50]	Ours $\mathcal{D}(X)$	Ours $\mathcal{D}(\mathcal{R}(X))$
AUC	0.905	0.816	0.837	0.907	0.783	**0.948**	0.932	*0.942*
F_1	0.880	0.808	0.823	0.893	0.785	0.914	*0.916*	**0.928**
AUC	0.676	0.796	0.788	0.479	0.798	0.929	*0.930*	**0.938**
F_1	0.718	0.784	0.779	0.671	0.777	0.880	*0.902*	**0.913**
AUC	0.487	0.657	0.629	0.337	0.676	*0.913*	*0.913*	**0.923**
F_1	0.672	0.716	0.711	0.667	0.715	0.858	*0.890*	**0.905**

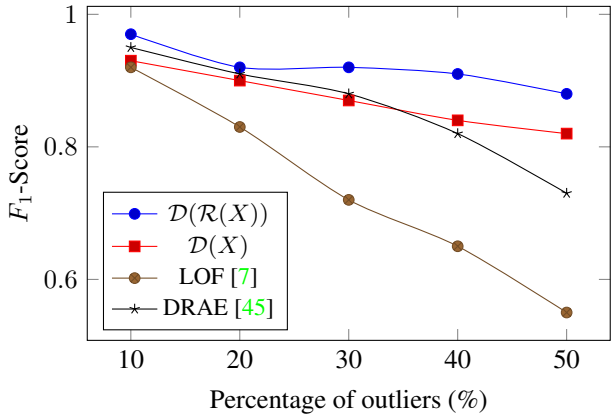

Figure 8. Comparisons of F_1-scores on MNIST dataset for different percentages of outlier samples involved in the experiment.

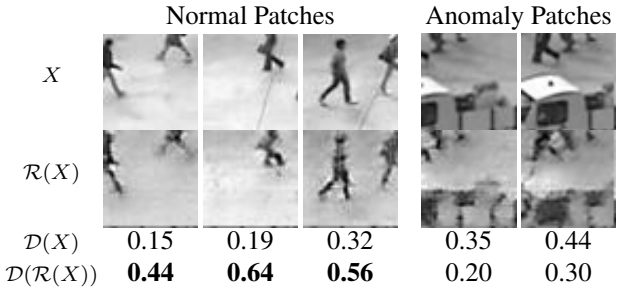

Figure 9. Examples of patches (denoted by X) and their reconstructed versions using $\mathcal{R}$ (*i.e.*, $\mathcal{R}(X)$): Three left Columns are normal patches, and two right ones are abnormal. The output of $\mathcal{D}$ is the likelihood of being a normal patch (a scalar in range [0,1]).

them. As can be seen, both proposed methods (*i.e.*, $\mathcal{D}(X)$ and $\mathcal{D}(\mathcal{R}(X))$) outperform all other methods in most cases. Interestingly, as we increase the number of inlier classes, from 1 to 3 and 5 (first, second and the last two rows, respectively), our method robustly learns the inlier concept(s).

4.3. Video Anomaly Detection

Anomaly event detection in videos or visual analysis of suspicious events is a topic of great importance in different computer vision applications. Due to the increased complexity of video processing, detecting abnormal events (*i.e.*, anomaly or novelty events) in videos is even a more burdensome task than image outlier detection. We run our method on a popular video dataset, UCSD [9] Ped2. The results are reported on a frame-level basis, as we aligned our experimental setup to previous works for comparison purposes.

4.3.1 Anomaly Detection Dataset

UCSD dataset: This dataset [9] includes two subsets, **Ped1** and **Ped2**, from two different outdoor scenes, recorded with a static camera at 10 fps and resolutions 158×234 and 240×360, respectively. The dominant mobile objects in these scenes are pedestrians. Therefore, all other objects (*e.g.*, cars, skateboarders, wheelchairs, or bicycles) are considered as anomalies. We evaluate our algorithm on Ped2.

4.3.2 Anomaly Detection Results

Result on UCSD Ped2: For this experiment, we divide the frames of the video into 2D patches of size 30×30. Training patches are only composed of frames with normal behaviors. In the testing phase, test patches are given to the joint network $\mathcal{R}$+$\mathcal{D}$, and the results are recorded. Fig. 9 shows examples of the output of $\mathcal{R}$ on the testing patches. As it is evident, normal patches (*i.e.*, left part of the figure) are successfully refined and reconstructed by the $\mathcal{R}$ network, while the abnormal ones (*i.e.*, the right part of the figure) are distorted and not adequately reconstructed. The last two rows in the figure show the likelihood score identified by our methods ($\mathcal{D}(X)$ and $\mathcal{D}(\mathcal{R}(X))$, respectively). $\mathcal{D}(\mathcal{R}(X))$ shows to yield more distinguishable results, leading to a better model for one-class classification and hence video anomaly detection. It is fascinating to note that one of the

Table 2. Frame-level comparisons on Ped2

Method	EER	Method	EER
IBC [6]	13%	RE [36]	15%
MPCCA [19]	30%	Ravanbakhsh *et al.* [34]	13%
MDT [26]	24%	Ravanbakhsh *et al.* [33]	14%
Bertini *et al.* [4]	30%	Dan Xu*et al.* [46]	17%
Dan Xu *et al.* [47]	20%	Sabokrou *et al.* [37]	19%
Li *et al.* [23]	18.5%	Deep-cascade [38]	**9%**
Ours - $\mathcal{D}(X)$	**16%**	Ours - $\mathcal{D}(\mathcal{R}(X))$	**13%**

most critical dilemmas for video anomaly detection methods is their high false positives. That is, algorithms often detect many of the 'normal' scenes as anomalies. In Fig. 9, three left columns are three tough 'normal' examples, as the human subject is not completely visible in the patch. We deliberately visualized these cases to illustrate how using $\mathcal{D}(\mathcal{R}(X))$ can effectively increase the discriminability of the system, compared to only $\mathcal{D}(X)$.

Similar to [26], we also report frame-level Equal Error Rate (EER) of our method and the compared methods. For this purpose, in any frame, if a pixel is detected as an anomaly, that frame is so labeled as 'anomaly.' Table 2 shows the result of our method in comparison to the state-of-the-art. The right column in Table 2 lists the results from methods based on variations of deep-learning. This table confirms that our method is comparable to all these approaches, while we are solving a more general problem that can be used for any outlier, anomaly or novelty detection problem. It is worth noting that other methods (especially Deep-cascade [38]) benefit from both spatial and temporal complex features, while our method operates on a patch-based basis, considering only spatial features of the frames. Our goal was to illustrate that the proposed method operates at least as well as the state-of-the-art, in a very general setting with no further tuning to the specific problem type. Simply, one can use spatiotemporal features and even further improve the results for anomaly event detection or related tasks.

4.4. Discussion

The experimental results confirmed that $\mathcal{R}$+$\mathcal{D}$ detects the novelty samples at least as well as the state-of-the-art or better than them in many cases, but finding the optimal structure and conducting the proper training procedure for these networks can be tedious and cumbersome tasks. The structure used in this paper proved well enough for our applications, while it can still be improved. We observed that achieving better performance is possible by modifying the structure, *e.g.*, by some modification in the size and the order of convolutional layers of $\mathcal{R}$+$\mathcal{D}$, we achieved better results by margins of 0.02 to 0.04 compared to the results reported in Table 1. Another important point is that it is very critical when to stop the training procedure of the joint network $\mathcal{R}$+$\mathcal{D}$. Stopping the training too early leads to immature learned network weights, while overtraining the networks confuses the $\mathcal{R}$ module and yields undesirable outputs. The stopping criterion outlined in Section 3.3 provides a right balance for the maturity of the joint network in understanding the underlying concept in the target class.

In addition, it is important note is that training a model in absence of the novelty/outlier class can be considered as weak supervision. For many problems this is acceptable, as all the samples we have are often inliers. When dealing with outlier detection problems, we can assume that number samples from the target class is much larger than the outlier samples. However, if we train the model at the presence of small number of outlier samples, the model still works. In a followup experiment, we mixed data from target (90%) and outlier (10%) classes in the training phase of the Ped2 experiment, and observed that the EER only dropped by 1.3%, which is still comparable to the state-of-the-art methods.

One of the major concerns in GANs is the mode collapse issue [3], which often occurs when the generator only learns a portion of real-data distribution and outputs samples from a single mode (*i.e.*, it ignores other modes). For our case, it is a different story as $\mathcal{R}$ directly sees all possible samples of the target class data and implicitly learns the manifold spanned by the target data distribution.

5. Conclusion

In this paper, we have proposed a general framework for one-class classification and novelty detection in images and videos, trained in an adversarial manner. Specifically, our architecture consists of two modules, $\mathcal{R}$econstructor and $\mathcal{D}$iscriminator. The former learns the concept of a target class to reconstruct images such that the latter is fooled to consider those reconstructed images as real target class images. After training the model, $\mathcal{R}$ can reconstruct target class samples correctly, while it distorts and decimates samples that do not have the concept shared among the target class samples. This eventually helps $\mathcal{D}$ discriminate the testing samples even better. We have used our models for a variety of related applications including outlier and anomaly detection in images and videos. The results on several datasets demonstrate that the proposed adversarially learned one-class classifier is capable of detecting samples not belonging to the target class (*i.e.*, they are novelty, outliers or anomalies), even though there were no samples from the novelty class during training.

Acknowledgement

This research was in part supported by a grant from IPM (No. CS1396-5-01).

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] E. Adeli, K.-H. Thung, L. An, F. Shi, and D. Shen. Robust feature-sample linear discriminant analysis for brain disorders diagnosis. In *Advances in Neural Information Processing Systems*, pages 658–666, 2015.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[4] M. Bertini, A. Del Bimbo, and L. Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3):320–329, 2012.

[5] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[6] O. Boiman and M. Irani. Detecting irregularities in images and in video. *International journal of computer vision*, 74(1):17–31, 2007.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[8] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[9] A. Chan and N. Vasconcelos. Ucsd pedestrian dataset. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(5):909–926, 2008.

[10] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456. IEEE, 2011.

[11] N. Divakar and R. V. Babu. Image denoising via cnns: An adversarial approach. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1076–1083. IEEE, 2017.

[12] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer, 2000.

[13] A. B. Gardner, A. M. Krieger, G. Vachtsevanos, and B. Litt. One-class novelty detection for seizure analysis from intracranial eeg. *JMLR*, 7(Jun):1025–1044, 2006.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[15] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[18] S. S. Khan and M. G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

[19] J. Kim and K. Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2921–2928. IEEE, 2009.

[20] W. Lawson, E. Bekele, and K. Sullivan. Finding anomalies with generative adversarial networks for a patrolbot. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 12–13, 2017.

[21] Y. LeCun, C. Cortes, and C. J. Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[22] G. Lerman, M. B. McCoy, J. A. Tropp, and T. Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015.

[23] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, 2014.

[24] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010.

[25] J. Liu, Z. Lian, Y. Wang, and J. Xiao. Incremental kernel null space discriminant analysis for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 792–800, 2017.

[26] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1975–1981. IEEE, 2010.

[27] M. Markou and S. Singh. Novelty detection: a reviewpart 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[28] H. Moonesignhe and P.-N. Tan. Outlier detection using random walks. In *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, pages 532–539. IEEE, 2006.

[29] B. T. Morris and M. M. Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2287–2301, 2011.

[30] M. M. Moya and D. R. Hush. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, 9(3):463–474, 1996.

[31] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[32] M. Rahmani and G. Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. *arXiv preprint arXiv:1609.04789*, 2016.

[33] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe. Abnormal event detection in videos using generative adversarial nets. *IEEE International Conference on Image Processing (ICIP)*, 2017.

[34] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe. Training adversarial discriminators for cross-channel abnormal event detection in crowds. *arXiv preprint arXiv:1706.07680*, 2017.

[35] O. Ronneberger, P. Fischer, and T. 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.

[36] M. Sabokrou, M. Fathy, and M. Hoseini. Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder. *Electronics Letters*, 52(13):1122–1124, 2016.

[37] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette. Real-time anomaly detection and localization in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 56–62, 2015.

[38] 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 Transactions on Image Processing*, 26(4):1992–2004, 2017.

[39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

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

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[43] M. C. Tsakiris and R. Vidal. Dual principal component pursuit. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18, 2015.

[44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[45] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1519, 2015.

[46] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015.

[47] D. Xu, R. Song, X. Wu, N. Li, W. Feng, and H. Qian. Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. *Neurocomputing*, 143:144–152, 2014.

[48] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[49] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne. Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

[50] C. You, D. P. Robinson, and R. Vidal. Provable self-representation based outlier detection in a union of subspaces. *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.