AVID: Adversarial Visual Irregularity Detection

Mohammad Sabokrou[1,⋆], Masoud Pourreza[2,⋆], Mohsen Fayyaz[3,⋆], Rahim Entezari[4], Mahmood Fathy[1], Jürgen Gall[3], Ehsan Adeli[5]

[1]Institute for Research in Fundamental Sciences (IPM) [2]AI & ML Center of Part
[3]University of Bonn [4]Complexity Science Hub, Vienna [5]Stanford University

Abstract. Real-time detection of irregularities in visual data is very invaluable and useful in many prospective applications including surveillance, patient monitoring systems, *etc.* With the surge of deep learning methods in the recent years, researchers have tried a wide spectrum of methods for different applications. However, for the case of irregularity or anomaly detection in videos, training an end-to-end model is still an open challenge, since often irregularity is not well-defined and there are not enough irregular samples to use during training. In this paper, inspired by the success of generative adversarial networks (GANs) for training deep models in unsupervised or self-supervised settings, we propose an end-to-end deep network for *detection* and *fine localization* of irregularities in videos (and images). Our proposed architecture is composed of two networks, which are trained in competing with each other while collaborating to find the irregularity. One network works as a pixel-level irregularity $\mathcal{I}$npainter, and the other works as a patch-level $\mathcal{D}$etector. After an adversarial self-supervised training, in which $\mathcal{I}$ tries to fool $\mathcal{D}$ into accepting its inpainted output as regular (normal), the two networks collaborate to detect and fine-segment the irregularity in any given testing video. Our results on three different datasets show that our method can outperform the state-of-the-art and fine-segment the irregularity. [1]

1 Introduction

In the recent years, intelligent surveillance cameras are very much exploited for different applications related to the safety and protection of environments. These cameras are located in sensitive locations to encounter dangerous, forbidden or strange events. Every moment vast amounts of videos are captured by these cameras, almost all of which comprise normal every-day events, and only a tiny portion might be irregular events or behaviors. Accurate and fast detection of such irregular events is very critical in designing a reliable intelligent surveillance system. Almost in all applications, there is no clear definition of what the irregularity can be. The only known piece is whatever that deviates from the normal every-day activities and events in the area should be considered as irregularity [1]. This is a subjective definition that can include a wide-range of diverse events as irregularity and hence makes it hard for automated systems to decide if an event in the scene is really an irregularity. Therefore, systems are

[1] ⋆Mohammad Sabokrou, Masoud Pourreza, and Mohsen Fayyaz contributed equally.

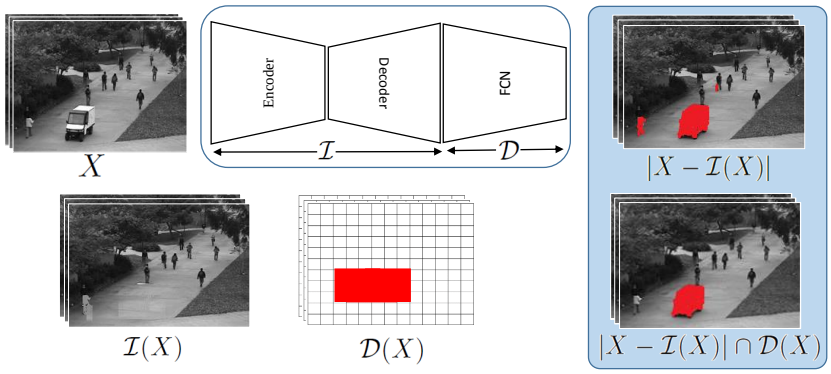

Fig. 1: The two networks $\mathcal{I}$ and $\mathcal{D}$ are trained jointly in an adversarial manner. $\mathcal{I}$ is an encoder-decoder convolutional network, which is trained to inpaint its input, X, *i.e.*, remove the irregularity. Therefore, $|X - \mathcal{I}(X)|$ indicates the pixel-level segmentation of the irregularity, from $\mathcal{I}$'s point-of-view. Whereas, $\mathcal{D}$ is a fully convolutional network (FCN), which identifies if different regions of its input are normal or irregular (patch-level). The intersection of the pixels denoted as irregularity in both $\mathcal{I}$ and $\mathcal{D}$ are labeled as the fine-segmentation of irregularity.

generally trained to learn the regularity, and rigorously tag everything else as irregularity [2].

Several different methods are used in the literature for learning the normal concept in visual scenes. Low-level visual features such as histogram of oriented gradients (HOG) [3] and histogram of optical flow (HOF) [4, 5] were the first feature subsets explored for representing regular scenes in videos. Besides, trajectory features [6] are also used for representing and modeling the videos, although they are not robust against problems like occlusion [3, 7]. Both low-level and trajectory features achieved good results while imposing a relatively high complexity to the system. Recently, with the surge of deep learning methods, several methods are proposed for detecting and localizing irregular events in videos [5, 7–10].

Although these deep learning-based methods effectively advanced the field, they fell short of learning end-to-end models for both detecting the irregularities and localizing them in spatio-temporal sequences, due to several challenges: (1) In applications like irregularity detection, there are little or no training data from the positive class (*i.e.*, irregularity), due to the nature of the application. Hence, training supervised models, such as convolutional neural networks (CNNs), is nearly impossible. Therefore, researchers (*e.g.*, in [10]) have usually utilized pre-trained networks to extract features from the scenes, and the decision is surrendered to another module. (2) To train end-to-end models for this task, just recently [11–14] used generative adversarial networks (GANs) and adopted unsupervised methods for learning the positive class (*i.e.*, irregular events). In these methods, two networks (*i.e.*, generator and discriminator) are trained. The generator generates data to compensate for the scarcity of the positive class,

while the discriminator learns to make the final decision making on whether its input is a normal scene or irregularity. Although they are trained with a very high computational load, the trained generator (or discriminator) is discarded at the testing time. Besides, most of these previous methods are patch-based approaches, and hence are often very slow. Note that these end-to-end models can only classify the scenes and do not precisely localize the irregularities. (3) Accurate pixel-level spatio-temporal localization of irregularities is still an ongoing challenge [9].

In addition to the above issues, as a general and ongoing challenging issue in video irregularity detection, detecting and localizing the irregularity in a pixel-level setting leads to models with many true positives while usually suffering from many false positive errors. On the contrary, some other methods operate on large patches (*e.g.*, [3]) and overcome the problem of high false positive error, with the price of sacrificing the detection rate. This motivated us to design a method that takes advantage from both pixel-level and patch-level settings, and come up with a model with high true positive rate while not sacrificing the detection rate. We do this by proposing an architecture, composed of two networks that are trained in an adversarial manner, the first of which is a pixel-level model and is trained to $\mathcal{I}$npaint its input by removing the irregularity it detects. The second network is a patch-level detector that $\mathcal{D}$etects irregularities in a patch level. The final irregularity detection and fine-segmentation is, then, defined as a simple intersection of the results from these two networks, having the benefits of both while discarding the pixels that result in high false positive errors (see Fig. 1).

According to the discussions above, in this paper, we propose an end-to-end method for joint detection and localization of irregularities in the videos, denoted as *AVID (Adversarial Visual Irregularity Detection)*. We use an adversarial training scheme, similar to those used in generative adversarial networks (GANs) [15]. But in contrast to previous GAN-based models (*e.g.*, [11, 13, 14, 16]), we show how the two networks ($\mathcal{I}$ and $\mathcal{D}$) can help each other to conduct the ultimate task of visual irregularity detection and localization. The two networks can be efficiently learned against each other, where $\mathcal{I}$ tries to inpaint the image such that $\mathcal{D}$ does not detect the whole generated image as irregularity. By regulating each other, these two networks are trained in a self-supervised manner [12, 17, 18]. Although, $\mathcal{I}$ and $\mathcal{D}$ compete with each other during training, they are trained to detect the video irregularity from different aspects. Hence, during testing, these two networks collaborate in detection and fine-segmentation of the irregularity.

In summary, the main contributions of this paper are three-fold: (1) We propose an end-to-end deep network for detection and localization of irregularities in visual data. To the best of our knowledge, this is the first work that operates on a video frame as a whole in an end-to-end manner (not on a patch level). (2) Our method can accurately localize the fine segments of the video that contain the irregularity. (3) Our proposed adversarial training of the two networks (one pixel-level and one patch-level) alleviates the high false positive rates of pixel-level methods while not suffering from high detection error rate of patch-level models.

2 Related Works

Detection of visual irregularities is closely related to different methods in the literature (including one-class classifiers, anomaly detection, outlier detection or removal methods). These approaches search for an irregularity, which is hardly and scarcely seen in the data. Traditional methods often learn a model for the normal class, and reject everything else (*i.e.*, identify as irregularity). Learning under a constraint (such as sparsity and compressed sensing) or statistical modeling are two common methods for modeling the normal class. For the case of visual data, feature representation (from videos and images) is an important part. Low-level features (such as HOG and HOF) and high-level ones (*e.g.*, trajectory) are widely used in the literature. In the recent years, similar to other computer vision tasks, deeply learned features are vastly utilized for irregularity detection. In this section, a brief review of the state-of-the-art methods for irregularity detection and related fields is provided.

Video Representation for Irregularity Detection. As one of the earliest representations for irregularity detection, trajectory were used [6, 19], such that an event not following a learned normal trajectory pattern is considered as anomaly. Optical-flows [4, 20–22], social forces (SF) [23], gradient features [3, 24], mixture of dynamic textures [2], and mixture of probabilistic PCAs (MPPCA) [25] are types of low-level motion representations used to model regular concepts. Deep learned features, using auto-encoders [26, 27], pre-trained networks [9], or PCAnet [28, 29] have recently shown great success for anomaly detection.

Constrained reconstruction as supervision. Representation learning for the normal (*i.e.*, regular) class under a constraint has shown effective to detect irregular events in visual data. If the new testing data does not conform to the constraint, it can potentially be considered as an irregularity. Learning to reconstruct normal concepts with sparse representation (*e.g.*, in [30]) and minimum effort (*e.g.*, in [1]) are widely exploited for this task. Boiman and Irani [1] consider an event as irregular if its reconstruction using the previous observations is nearly impossible. In [31], a scene parsing approach is proposed by Antic *et al.* in which all object hypotheses for the foreground of a frame are explained by normal training. Those hypotheses that cannot be explained by normal training are considered as anomaly. In [7, 12, 30] the normal class is learned through a model by reconstructing samples with minimum reconstruction errors. A high reconstruction error for a testing sample means this sample is irregular. Also, [7, 30] introduced a self-representation technique for video anomaly and outlier detection through a sparse representation, as a measure for separating inlier and outlier samples.

Deep Adversarial Learning. Recently, GANs [15] are widely being used for generating data to learn specific models. They are extended for prediction tasks, in which there are not enough data present for training (*e.g.*, in [11, 13, 14]). GANs are based on a game between two different networks, one generator (G) and one discriminator (D). G aims to generate sensible data, while D tries to discriminate real data from the fake data generated by G. A closely related type of GANs to our work is the conditional GANs [32]. In conditional GANs, G takes

an image X as the input and generates a new image X', whereas, D tries to distinguish X from X'. Isola *et al.* [33] proposed an 'Image-to-image translation' framework based on conditional GANs, where both G and D are conditioned on the real data. Using a U-Net encoder-decoder [34] as the generator and a patch-based discriminator, they transformed images with respect to different representations. In another work, [16] proposed to learn the generator as the reconstructor for normal events, and tag chunks of the input frame as anomaly if they cannot be properly reconstructed. In our work, $\mathcal{I}$ learns to inpaint its input and make it free from irregularity in pixel-level, and $\mathcal{D}$ regulates it by checking if its output is irregular or not. This self-supervised learning scheme leads to two networks that improve the detection and fine-segmentation performance for any given testing image. Liu *et al.* [35] proposed to learn an encoder-decoder GAN to generate the future video frame using optical-flow features, used for irregularity detection, *i.e.*, if the prediction is far from the real future frame, it is counted as irregularity. Similar to all other works, the work in [35] ignores the discriminator in the testing phase. Also they suffer from high false positive rates.

3 AVID: Adversarial Visual Irregularity Detection

The proposed method for irregularity detection and localization is composed of two main components: $\mathcal{I}$ and $\mathcal{D}$. $\mathcal{I}$ learns to remove the pixel-wise irregularity from its input frame (*i.e.*, $\mathcal{I}$npaint the video), while $\mathcal{D}$ predicts the likelihood of different regions of the video (patches) being an irregularity. These networks are learned in an adversarial and self-supervised way in an end-to-end setting. In the following, we outline the details of each network. An overall sketch of the proposed method is illustrated in Fig. 1. In summary, $\mathcal{I}$ learns to $\mathcal{I}$npaint its input X to fool $\mathcal{D}$ that the inpainted version does not have any irregularities. For $\mathcal{I}$ to learn to reconstruct skewed images, $\mathcal{I}$ is exposed to noisy versions of the videos in the data set and therefore it implicitly learns not only to remove the irregularity but also to remove the noise in the data. Besides, $\mathcal{D}$ knows the distribution of original data $\mathcal{P}_d$, as it has access to the data set containing all normal videos (or with a tiny portion of irregularities present in the data). Having access to $\mathcal{P}_d$, $\mathcal{D}$ simply rejects poorly inpainted or reconstructed data. These two networks self-supervise each other and are trained through this bilateral game. This structure is inspired by GAN models, although our model does not generate from scratch and only enhances its input tailored for detection of irregularities.

After the adversarial training, $\mathcal{I}$ will be an expert to inpaint its input (and make it devoid of noise), which successfully fools $\mathcal{D}$. Module $\mathcal{I}$ is a pixel-level irregularity detector and $\mathcal{D}$ a patch-level one, hence, $|X - \mathcal{I}(X)| \cap \mathcal{D}(X)$ can define the fine-segmentation and the precise location of the irregularity in any input testing video frame X. Note that each of the two networks $\mathcal{I}$ and $\mathcal{D}$ can be exploited for detecting and localizing the irregularity, but by aggregating them, we show a great improvement in the results. Detailed descriptions of each module along with the training and testing procedures are explained in the following.

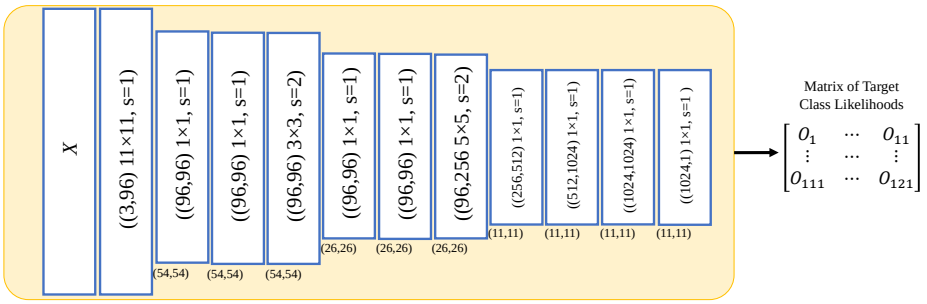

Fig. 2: Structure of $\mathcal{D}$, a FCN that scores video regions on how likely they are irregularities. All layers are convolutional layers and are described in this form $((C_1, C_2), K, s)$, with C_1 and C_2 as the number of channels of the input and output, K as the size of the applied kernel, and s as the stride for convoluted. Underneath each layer, the size of the feature maps are provided. Matrix $\mathcal{O}$, output of $\mathcal{D}$, defines regularity likelihood for each region.

3.1 $\mathcal{I}$: Inpainting Network

In some recent previous works [5, 7, 12], it is stated that when an auto-encoder is trained only on the inlier or normal class, the auto-encoder will be unable to reconstruct outlier or anomaly samples. Since parameters of auto-encoder are optimized to reconstruct samples from the normal (regular) class, as a side-effect, the reconstruction error of outliers (irregularities in our case) will be high. Also, in [12] in an unsupervised GAN-style training, a patch-based CNN is proposed that decimates outliers while enhancing the inlier samples. This makes the separation between the inliers and outliers much easier. In this paper, we use a similar idea, but in contrast: (1) $\mathcal{I}$ (analogous to the generator in GANs) is not directly used as an irregularity detector; (2) Instead of decimating outliers (irregularities in our case), our network inpaints its input by removing the irregularity from it. Implicitly, $\mathcal{I}$ operates similar to a de-noising network, which replaces the irregularity in the video with a dominant concept (*e.g.*, dominant textures).

Architecturally, $\mathcal{I}$ is an encoder-decoder convolutional network (implemented identical to U-Net [34]), which is trained only with data from the normal (regular) class. It learns to map any given input to the normal concept. Usually, irregularity occurs in some video frames, and $\mathcal{I}$ acts by reconstructing those deteriorated parts of the videos.

3.2 $\mathcal{D}$: Detection Network

Fully convolutional neural networks (FCNs) can effectively represent video frames and patches, and are previously used for different applications, such as semantic segmentation [36, 37]. In a recent work, [10] used a FCN for irregularity detection in videos, in which the authors used a pre-trained FCN just for describing the video patches. Thier method was not capable to detect (or score) the irregularity in the frames. Inspired by this idea, we use a FCN for the detection phase,

but train it in an adversarial manner (along with $\mathcal{I}$). Our model is, hence, an end-to-end trainable network. We train the FCN (*i.e.*, $\mathcal{D}$ network) to score (and hence detect) all irregular regions in the input frame all at once.

Unlike conventional discriminator networks in a GAN, where the discriminator just provides a judgment about its input as a whole, $\mathcal{D}$ is capable to judge about different regions of its input. Consequently, its output is a matrix of likelihoods, which imply if the regions of its input follow the distribution of the normal (regular) data or not (*i.e.*, $\mathcal{P}_d$). Fig. 2 shows the architecture of $\mathcal{D}$, which includes several convolutional layers. For this application, since local spatial characteristics are crucial, we do not use any pooling or fully connected layers, which ignore the spatial characteristics. On the other hand, to preserve the locality and enrich the features, several 1×1 convolutional layers are used.

3.3 Adversarial Training of $\mathcal{I}$+$\mathcal{D}$

Goodfellow *et al.* [15] proposed an efficient way for training two deep-neural networks (Generator (G) and Discriminator (D), in their terminology) through adversarial training, called GAN. GANs aim to learn the distribution of training data $\mathcal{P}_d$, and simultaneously generate new samples based on the same distribution $\mathcal{P}_d$. Therefore, G maps a vector of random variables (say Z) from a specific distribution $\mathcal{P}_Z$ to a sample from real data distribution and D seeks to discriminate between the actual data and the fake data generated by G. Generator and Discriminator are learned in a two-player mini-max game, formulated as:

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

Similarly, $\mathcal{I}$+$\mathcal{D}$ can be adversarially trained. Unlike conventional GANs, which map a latent space Z to a sample from $\mathcal{P}_d$, $\mathcal{I}$ maps a very noisy sample $X+\eta$ to a noise-less one that can fool $\mathcal{D}$ into identifying it as a normal scene *i.e.*,

$$\tilde{X} = (X \sim \mathcal{P}_d) + \gamma \left(\eta \sim \mathcal{N}(0, \sigma^2\mathbf{I})\right) \longrightarrow X' \sim \mathcal{P}_d, \tag{2}$$

where η is a Gaussian noise sampled from the normal distribution with standard deviation σ, *i.e.*, $\mathcal{N}(0,\sigma^2\mathbf{I})$. γ is a hyperparameter that defines how severely to contaminate X with noise. Note that the addition of η forces $\mathcal{I}$ to learn how to restore X from $\tilde{X}$, *i.e.*, in absent of irregularity.

On the other hand, $\mathcal{D}$ has access to the original pool of training data, hence, knows $\mathcal{P}_d$, and is trained to identify the normal class. In our case, $\mathcal{D}$ decides which region of $\mathcal{I}(\tilde{X})$ follows from $\mathcal{P}_d$. To fool $\mathcal{D}$, $\mathcal{I}$ is implicitly forced to inpaint its input. As mentioned above, $\mathcal{D}$ (*i.e.*, our discriminator network) judges on the regions of its input and not the whole image (which is the case for the GAN discriminators). Consequently, output of $\mathcal{D}(X)$ is a matrix, $\mathcal{O}\in\mathbb{R}^{n_1\times n_2}$, in which each cell $\mathcal{O}(i,j)$ corresponds to the i^{th} and j^{th} image region. Therefore, the joint training aims to maximize $\sum_{i=1,j=1}^{i=n_1,j=n_2}\mathcal{O}(i,j)$ (*i.e.*, maximize the likelihood of $\mathcal{I}$'s output to be normal). $n_1\times n_2 = n$ is the total number of regions judged by $\mathcal{D}$. Accordingly, $\mathcal{I}$+$\mathcal{D}$ can be jointly learned by optimizing the following objective:

$$\min_{\mathcal{I}} \max_{\mathcal{D}} \Big(\mathbb{E}_{X\sim\mathcal{P}_d}[\log(\|\mathcal{D}(X)\|^2)] + \mathbb{E}_{\tilde{X}\sim\mathcal{P}_d+\mathcal{N}_\sigma}[\log(\|\mathcal{Y}-\mathcal{D}(\mathcal{I}(\tilde{X}))\|^2)] \Big), \tag{3}$$

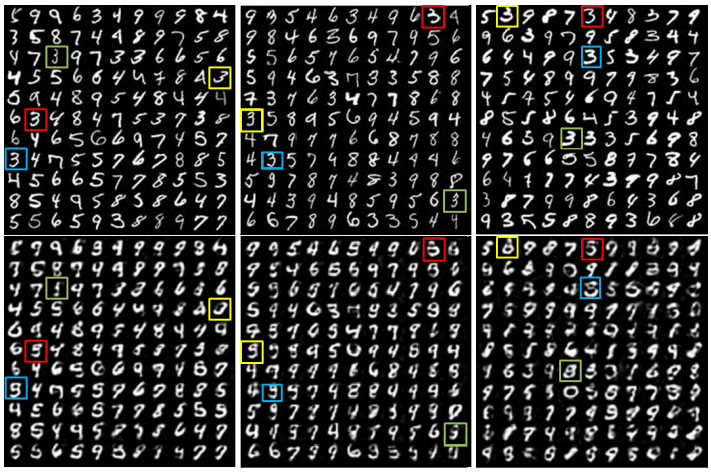

Fig. 3: Examples of images (X) and their inpainted versions using $\mathcal{I}$ (*i.e.*, $\mathcal{I}(X)$). The network is trained on images containing 0-9 digits, except digit '3'. These images are created from images in the MNIST dataset [38], to show how $\mathcal{I}$ and $\mathcal{D}$ operate. When digit '3' appears in a test image, it is considered as an irregularity. For clarity, several irregularity regions are marked in X and $\mathcal{I}(X)$.

where $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2} := \mathbf{1}^{n_1 \times n_2}$. Based on the above objective function, $\mathcal{I}$ learns to generate samples with the distribution of normal data (*i.e.*, $\mathcal{P}_d$). Hence, the parameters of this network, $\theta_{\mathcal{I}}$, are learned to restore a noisy visual sample. So, $\mathcal{I}(\tilde{X}, \theta_{\mathcal{I}})$ would be an irregularity-free version of $\tilde{X}$. For better understanding, suppose each frame of the video X is partitioned into $n = n_1 \times n_2$ non-overlapping regions (blocks), $B_{i \in 1:n}$. The proposed algorithm looks to find which of these regions are irregular. After the joint training of $\mathcal{I}$+$\mathcal{D}$, the modules can be interpreted as follows:

- $\forall i$; $B_i \sim \mathcal{P}_d + \mathcal{N}_\sigma$; $\mathcal{I}(\tilde{X} = B_{i \in 1:n}) \Rightarrow X' = B'_{i \in 1:n}$, where $\forall i$; $B'_i \sim \mathcal{P}_d$. This is the case if the input is free from irregularity and is already following $\mathcal{P}_d$. X' is the output of $\mathcal{I}$, and hence $\|X - X'\|$ is minimized (will be near zero). This is because of the fact that $\theta_{\mathcal{I}}$ is optimized to reconstruct its input (all B_i regions) while the output also follows $\mathcal{P}_d$. Note that $\mathcal{I}$ works similar but not exactly the same as the refinement network in [12], the de-noising auto-encoder in [7], or de-nosing convolutional neural network in [39]. Consequently, if the input frame is already free from irregularity, $\mathcal{I}$ acts only as an enhancement function.
- $\exists j$; $B_j \nsim \mathcal{P}_d$; $\mathcal{I}(\tilde{X} = B_{i \in 1:n}) \Rightarrow X' = B'_{i \in 1:n}$, where $\forall i$; $B'_i \sim \mathcal{P}_d$. This is the case if at least one of the regions in $\tilde{X}$ is irregular. Then, it is expected from $\mathcal{I}$ that $\mathcal{I}(B_j \nsim \mathcal{P}_d) \Rightarrow B'_j \sim \mathcal{P}_d$. The irregular region is forced to follow the normal data distribution, as $\mathcal{I}$ is trained to restore a normal region contaminated with strong noise ($\gamma(\eta \sim \mathcal{N}(0, \sigma^2 \mathbf{I}))$) to a clean noise-free

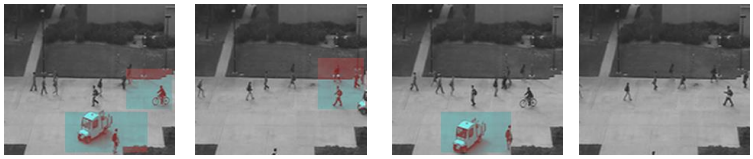

Fig. 4: Examples of the output of $\mathcal{D}$, *i.e.*, matrix $\mathcal{O}$, mapped on the original frames. The colored areas on the image are the low-scored regions in $\mathcal{O}$.

normal-looking region. In the testing phase, an irregular region, in $\mathcal{I}$'s point-of-view, is consider as strong noise. Note that in our experiments, $\gamma < 0.4$ is considered as weak and $\gamma \geq 0.4$ is considered as strong noise. Strong noise added to the training samples (inputs of $\mathcal{I}$) is considered as concepts that should be removed from the output of $\mathcal{I}$, for it to be able to fool $\mathcal{D}$. See Fig. 3, as a proof-of-concept example. Digit '3' is considered as an irregular concept in this example, and $\mathcal{I}$+$\mathcal{D}$ have never seen any '3' during training. So, $\mathcal{I}$ tries to replace it with a dominant concept from the normal concepts, which can be any digit between 0-9 except '3'. Consequently those B_is that follow $\mathcal{P}_d$ are not touched (or are enhanced), while those not following the normal data distribution are converted to a dominant concept (*i.e.*, inpainted).

– $\mathcal{D}(X = B_{i\in 1:n}) \Rightarrow \mathcal{O}_{i\in 1:n}$ where each element of matrix $\mathcal{O}$, output of $\mathcal{D}$, indicates the confidence for the corresponding region to be normal (regularity). Note that here $\mathcal{O}_i$ is analogous to $\mathcal{O}(a, b)$ with $i = b + (a - 1) \cdot n_1$. Let's consider $\exists j$; $B_j \nsim \mathcal{P}_d$; we expect that $\mathcal{O}_j \leq \mathcal{O}_{i\neq j}$. Parameters of $\mathcal{D}$, $\theta_{\mathcal{D}}$, are learned to map normal regions (*i.e.*, following $\mathcal{P}_d$) to 1, and 0 otherwise. Fig. 4 shows the results of $\mathcal{D}$, in which the locations with an irregularity have lower scores.

With a modification on the objective function and the structure of GANs, our two proposed deep networks are adversarially trained. They learn to identify irregularities from two different aspects (pixel-level and patch-level) in a self-supervised manner. The equilibrium point as the stopping criterion for the two networks is discussed in Section 4.3.

3.4 Irregularity Detection

In the previous subsections, detailed structures and behaviours of the two networks are explained. As mentioned, $\mathcal{I}$ acts as a pixel-level inpainting network, and $\mathcal{D}$ as a patch-level irregularity detector. The difference between the input and outputs of $\mathcal{I}$ for any testing frame X (*i.e.*, $|X - \mathcal{I}(X)|$) can be a guideline for where pixels of the input frames are irregular. On the other hand, $\mathcal{D}(X = B_{i\in 1:n})$ shows which regions of X are irregular (*i.e.*, those with $\mathcal{O}_j \leq \zeta$). As discussed earlier, the detection based on $\mathcal{I}$ leads to high false positive rate, and the detection solely based on $\mathcal{D}$ leads to high detection error rates. Therefore, outputs of these two networks are masked by each other and the intersection is considered as the irregularity localization.

To identify the regions of irregularities from the output of $\mathcal{D}$ (*i.e.*, matrix $\mathcal{O}$), we can consider all regions with $\{B_i|(\mathcal{O}_i \leq \zeta)\}$, where B_i is respective field of $\mathcal{O}_i$ on the input. As mentioned above, $\mathcal{I}$ will reconstruct its whole input image, except for the irregularities, which are inpainted. Consequently, pixels where $|X - \mathcal{I}(X)| \geq \alpha$ can be considered as potential pixels containing an irregularity. To alleviate the high false positive rate, we just mask these pixels with the above regions. Consequently, final irregularity fine-segmentation on X can be defined as

$$\mathcal{M} = \{|X - \mathcal{I}(X)| \geq \alpha\} \cap \{B_i|(\mathcal{O}_i \leq \zeta)\}. \tag{4}$$

3.5 Preprocessing of the Videos

Irregular events in visual data (especially in videos) are defined in terms of irregular shapes, motion, or possibly a combination of both. Therefore, to identify the motion properties of events, we require a series of frames. Two strategies can be adopted for this purpose: (1) Adding a LSTM sequence at the end of the proposed network [40]; (2) Using 3D kernels, instead of 2D ones in the same architectures we proposed (such as in [41]). However, these methods increase the number of parameters and the computational complexity of the model. [10] proposed a simple preprocessing step to feed videos instead of images to a CNN without any modification on the structures of a 2D CNN. To interpret both shape and motion, we consider the pixel-wise *average* of frame I_t and previous frame I_{t-1}, denoted by I'_t (not to be confused with a derivative): $I'_t(p) = \frac{1}{2}(I_t(p) + I_{t-1}(p))$, where I_t is the t^{th} frame in the video. For detecting irregularities on I_t, we use the sequence $X = \langle I'_{t-4}, I'_{t-2}, I'_t \rangle$, and input it to the three channels (similar to R, G, and B channels) of the networks.

4 Experimental Results

We evaluate the performance of the proposed method on two standard benchmarks for detecting and localizing irregular events in videos. Also, we create a dataset, called IR-MNIST, for evaluating and analyzing the proposed method to better showcase the abilities of the network modules, as a proof-of-concept.

The proposed method is implemented using PyTorch [42] framework. All reported results are from implementations on a GeForce GTX 1080 GPU. Learning rate of $\mathcal{I}$ and $\mathcal{D}$ are set to be the same and is equal to 0.002. Also, momentum of both is equal to 0.9, with a batch size of 16. The hyperparameter γ, which controls the scale of added Gaussian noise for the training samples (in $\mathcal{I}$) is equal to 0.4.

4.1 Datasets

UCSD: This dataset [2] includes two subsets, Ped1 and Ped2 . Videos are from an outdoor scene, recorded with a static camera at 10 fps. The dominant mobile objects in these scenes are pedestrians. Therefore, all other objects (*e.g.*, cars, skateboarders, wheelchairs, or bicycles) are considered as irregularities.

Fig. 5: Examples of the output of $\mathcal{I}$ on the UCSD dataset. Bottom row shows output of $\mathcal{I}$. Top row shows the original frames.

UMN: The UMN dataset is recorded for three different scenarios. In each scenario, a group of people walk in normal pace in an area, but suddenly all people run away (*i.e.*, they escape). The escape is considered to be the irregularity.
IR-MNIST (Available at `http://ai.stanford.edu/~eadeli/publications/data/IR-MNIST.zip`**):** To show the properties of the proposed method, we create a simple dataset using the images from MNIST [43]. To create each single image, randomly 121 samples are selected from the MNIST dataset and are put together as a 11×11 puzzle. Some samples are shown in Fig. 3. We create as much images to have 5000 training data and 1000 test samples. Training samples are created without using any images of the digit '3'. Hence, '3' is considered as an irregular concept. We expect that our method detects and localizes all patches containing '3' in the testing images, as irregularity.

4.2 Results

Results on UCSD. Fig. 5 visualizes the outputs of network $\mathcal{I}$ on several examples of UCSD frames. As can be seen, irregular objects such as bicycles and cars disappear in the output of $\mathcal{I}(X)$, and the regular regions are approximately reconstructed. Although $\mathcal{I}$ is trained to reconstruct regular regions with minimum loss, loss of quality is unavoidable, as a consequence of the strong noise applied to the inputs of $\mathcal{I}$ during training. This shortage, however, does not adversely affect our final decision, because maximum difference between X and $\mathcal{I}(X)$ happens in the pixels when an irregularity occurred. Fig. 4 also shows several output samples of the proposed detector $\mathcal{D}$ for detecting irregularity in videos. It confirms that irregular blocks can be appropriately detected. For a quantitative analysis, similar to [2], two standard measures on this dataset are considered. In frame level (FL) each of the frames is considered to be anomaly if at least one pixel is detected as irregularity. In pixel-level (PL) analysis, the region identified as anomaly should have an overlap of at least 40% with the ground-truth irregular pixels to be considered as irregularity. PL is a measure for evaluating the accuracy of the localization in a pixel-level. A comparison between the performance of the proposed and the state-of-the-art methods is provided in Table 1. The proposed method for detecting the irregular frames is comparable to state-of-the-

Table 1: Frame-level accuracy (FL) and pixel-level accuracy (PL) comparisons on the UCSD dataset. The last column shows if the methods are (1) based on $\mathbb{D}$eep learning or not, (2) $\mathbb{E}$nd-to-end deep networks or not, and finally (3) $\mathbb{P}$atched based methods or not.

Method	Ped1 (FL/PL)	Ped2 (FL/PL)	($\mathbb{D}$/$\mathbb{E}$/$\mathbb{P}$)
IBC[1]	(14/26)	(13/26)	(✗/✗/✓)
MDT[2]	(25/58)	(24/54)	(✗/✗/✗)
Bertini *et al.* [3]	(31/70)	(30/–)	(✗/✗/✓)
Xu *et al.* [44]	(22/–)	(20/42)	(✗/✗/✗)
Li *et al.* [45]	(16/–)	(18/29)	(✗/✗/✗)
RE [7]	(–/–)	(15/–)	(✓/✗/✓)
Xu *et al.* [26]	(16/40)	(17/42)	(✓/✗/✓)
Sabokrou *et al.* [27]	(–/–)	(19/24)	(✓/✗/✓)
Deep-Cascade[9]	(9.1/15.8)	(8.2/19)	(✓/✗/✓)
Deep-Anomaly[10]	(–/–)	(**11**/**15**)	(✓/✗/✗)
Ravanbakhsh *et al.* [14]	(**7**/34)	(**11**/–)	(✓/✗/✓)
ALOCC [12]	(–/–)	(13/–)	(✓/✓/✓)
$\mathcal{D}(X)$	(–/16.7)	(–/17.2)	(✓/✓/✗)
$\mathcal{I}(X)$	(17.3/–)	(17.8/–)	(✓/✓/✗)
AVID	(12.3/**14.4**)	(14/**15**)	(✓/✓/✗)

Table 2: EER and AUC perforamnce metrics on UMN dataset.

	Chaotic invariant [46]	SF [2]	Cong [30]	Saligrama [47]	Li [45]	Ours (AVID)
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**

art methods, but the localization performance outperforms all other methods by a large margin. As can be seen, [9, 12, 14] achieve a better performance by a narrow margin in a frame-level aspect compared to us, but unlike ours, these methods are not able to process images as a whole in an end-to-end fashion. They require to split a frame into a set of patches and feed them to the network one-by-one. The last column of Table 1 shows which methods are not patch-based and end-to-end. Furthermore, the performances of $\mathcal{I}$ and $\mathcal{D}$ as independent baselines are also reported in this table, which show that each single one of them can preform as well as previous state-of-the-arts, while our final end-to-end model, AVID, outperforms all of these methods.

Results on UMN. Table 2 shows the irregularity detection performance in terms of equal error rate (EER) and area under the ROC curve (AUC). As discussed earlier, this dataset has some limitations, as there are only three types of abnormal scenes in the dataset with very high temporal-spatial abrupt changes between normal and abnormal frames. Also, there are no pixel-level ground truth for this dataset. Based on these limitations, to evaluate the method, EER and AUC are reported in frame-level settings. Since this dataset is simple, and

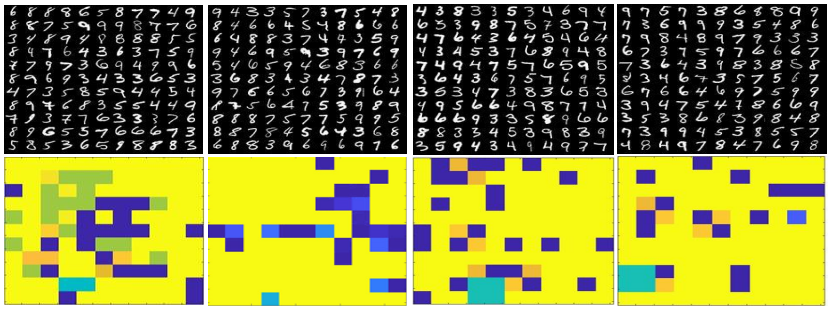

Fig. 7: Examples outputs of $\mathcal{D}$ on IR-MNIST. Bottom row shows (resized version of) the heat-map, for input testing images (in the top row). Here, '3' is intended as an irregular concept.

irregularity localization is not important, only the global detector is used to evaluate the results. Because of the simplicity of this dataset, previous methods already performed reasonably good on this dataset. AUC of our method is comparable with the other best results and EER of our approach is better (by 0.2 percent) than the second best.

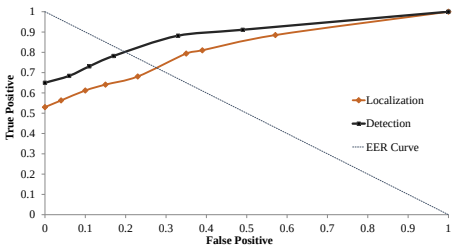

Fig. 6: ROC curves for detection and localization performance on IR-MNIST.

Results on IR-MNIST as a toy data-set. Fig. 3 confirms that the network $\mathcal{I}$ can properly substitute irregular regions with a (closest) normal concept. Since '3' is considered as an irregular concept, $\mathcal{I}$ converted it to another digit, which is most similar to '3'. Several samples of irregular regions in Fig. 3 are marked (on both the original and inpainted version of the same samples). Similarly, we evaluate $\mathcal{D}$ in detecting irregular regions on an image. Fig. 7 shows the heat-map of $\mathcal{D}$'s output for several samples, where blue is 1 and yellow indicates 0, and other colors are in between (in a parula colormap). Note that the output is resized to have the same size as the original images. Fig. 6 shows the localization and detection performance on the IR-MNIST dataset using the receiver operating characteristic (ROC) curve. This curve is drawn by repeatedly changing the two thresholds in Eq. (4) and recording the results. Detection is just based on if a frame contains an irregular concept ('3' digits) or not (checked over 1000 different testing samples). For localization all 11×11 regions of an images are considered, and if the region is correctly detected, it is counted as a true localization. So, $11 \times 11 \times 1000$ regions are evaluated. The EERs of the detection and localization are equal to 21% and 29%, receptively.

4.3 Discussion

Added noise to the input of $\mathcal{I}$ in training phase. In some cases similar to de-nosing auto-encoder [48], de-noisng CNN [39] or one-class classification tasks [12], researchers added noise to the training data to make their network robust against noise. We also contaminated our training data with a statistical noise, with γ indicating its intensity. This hyperparameter actually plays a very interesting role for training the model. Using this hyperparameter, we can control the learning pace between $\mathcal{I}$ and $\mathcal{D}$. Since, $\mathcal{I}$ sees only normal samples during training, in the noise-free case, it can easily reconstruct them so that $\mathcal{D}$ is fooled. The added noise actually makes $\mathcal{I}$ to learn how to inpaint and remove the irregularity to a pixel-level. Therefore, a very small value for γ leads to a task, which is very easy for $\mathcal{I}$ and a very large value will mislead $\mathcal{I}$ from seeing the actual normal data distribution (*i.e.*, $\mathcal{P}_d$). Based on our experiments, $\gamma = 0.4$ leads to good results. From another point-of-view, γ is a very good means to create a proper scheduling between $\mathcal{I}$ and $\mathcal{D}$, which is a very interesting and recent topic on for GANs [49].

Stopping criterion. In conventional GANs, the stopping criterion is defined as when the model reaches a Nash equilibrium between G and D. However, for our case, the optimum point for $\mathcal{I}$ and $\mathcal{D}$ is not often obtained at the same time. During learning of these two networks, when they are competing with each other, different conditions may occur. At a time that $\mathcal{D}$ is capable to efficiently classify between fake and real data (*i.e.*, work as an accurate classifier on the validation data), we save its parameters, $\theta_{\mathcal{D}}$. Also, when $\mathcal{I}$ generates samples as well as the normal class (*i.e.*, $\|X - \mathcal{I}(X)\|^2$ is in the minimum point), the parameters of $\mathcal{I}$, $\theta_{\mathcal{I}}$, are also saved. So, at different time spans $\theta_{\mathcal{I}}$ and $\theta_{\mathcal{D}}$ are saved , during the training procedure. Similar to other GAN-style models, finding the optimum point for stopping adversarial training of $\mathcal{I}$+$\mathcal{D}$ is a hard task.

Mode collapse. One of the major concerns in GANs is the mode collapse issue, which often occurs when the generator only learns a portion of the real-data distribution and outputs samples from a single mode (*i.e.*, ignores other modes). For our case, it is a different story as $\mathcal{I}$ directly sees all possible samples of the normal class and implicitly learns the manifold spanned by them. Reconstructing the training samples, instead of starting from a random latent space, is an acceptable way to avoid the mode collapse issue [50].

5 Conclusions

In this paper, we proposed an efficient method for irregularity detection and localization in visual data (*i.e.*, images and videos). Two proposed deep networks, $\mathcal{I}$ and $\mathcal{D}$ are adversarially trained in a self-supervised setting. $\mathcal{I}$ learns to efficiently reconstruct normal (regular) regions and implicitly inpaints irregular ones. $\mathcal{D}$ learns to score different regions of its input on how likely they are irregularities. Integrating the outputs of the pixel-level results from $\mathcal{I}$, and the patch-level results from $\mathcal{D}$ provides a promising irregularity detection metric, as well as fine-segmentation of the irregularity in the visual scene. The results on several synthetic and real datasets confirm that the proposed adversarially learned network is capable of detecting irregularity, even when there are no irregular

samples to use during training. Our method benefits from the advantages of both pixel-level and patch-level methods, while not having their shortcomings.

Acknowledgements: This research was in part supported by a grant from IPM (No. CS1396-5-01). Mohsen Fayyaz and Juergen Gall have been financially supported by the DFG project GA 1927/4-1 (Research Unit FOR 2535) and the ERC Starting Grant ARCA (677650).

References

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

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

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

[4] Colque, R.V.H.M., Caetano, C., de Andrade, M.T.L., Schwartz, W.R.: Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. IEEE Transactions on Circuits and Systems for Video Technology **27** (2017) 673–682

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

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

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

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

[9] Sabokrou, M., Fayyaz, M., Fathy, M., Klette, R.: Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. IEEE Transactions on Image Processing **26** (2017) 1992–2004

[10] Sabokrou, M., Fayyaz, M., Fathy, M., Moayedd, Z., et al.: Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. Computer Vision and Image Understanding (2018)

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

[12] Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E.: Adversarially learned one-class classifier for novelty detection. CVPR (2018)

[13] Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging, Springer (2017) 146–157
[14] Ravanbakhsh, M., Sangineto, E., Nabi, M., Sebe, N.: Training adversarial discriminators for cross-channel abnormal event detection in crowds. arXiv preprint arXiv:1706.07680 (2017)
[15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
[16] Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., Sebe, N.: Abnormal event detection in videos using generative adversarial nets. arXiv preprint arXiv:1708.09644 (2017)
[17] Odena, A.: Semi-supervised learning with generative adversarial networks. In: Data Efficient Machine Learning workshop, ICML. (2016)
[18] Do-Omri, A., Wu, D., Liu, X.: A self-training method for semi-supervised gans. In: ICLR. (2018)
[19] Piciarelli, C., Foresti, G.L.: On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters **27** (2006) 1835–1842
[20] Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. IEEE transactions on pattern analysis and machine intelligence **30** (2008) 555–560
[21] Cong, Y., Yuan, J., Tang, Y.: Video anomaly search in crowded scenes via spatio-temporal motion context. IEEE transactions on information forensics and security **8** (2013) 1590–1599
[22] Benezeth, Y., Jodoin, P.M., Saligrama, V., Rosenberger, C.: Abnormal events detection based on spatio-temporal co-occurences. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 2458–2465
[23] Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 935–942
[24] Kratz, L., Nishino, K.: Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 1446–1453
[25] Kim, J., Grauman, K.: 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, IEEE (2009) 2921–2928
[26] Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N.: Learning deep representations of appearance and motion for anomalous event detection. BMVC (2015)
[27] Sabokrou, M., Fathy, M., Hoseini, M., Klette, R.: Real-time anomaly detection and localization in crowded scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2015) 56–62

[28] Feng, Y., Yuan, Y., Lu, X.: Learning deep event models for crowd anomaly detection. Neurocomputing **219** (2017) 548–556
[29] Fang, Z., Fei, F., Fang, Y., Lee, C., Xiong, N., Shu, L., Chen, S.: Abnormal event detection in crowded scenes based on deep learning. Multimedia Tools and Applications **75** (2016) 14617–14639
[30] Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3449–3456
[31] Antić, B., Ommer, B.: Video parsing for abnormality detection. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2415–2422
[32] Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
[33] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)
[34] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
[35] Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection–a new baseline. arXiv preprint arXiv:1712.09867 (2017)
[36] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
[37] Nie, D., Wang, L., Adeli, E., Lao, C., Lin, W., Shen, D.: 3-d fully convolutional networks for multimodal isointense infant brain image segmentation. IEEE Transactions on Cybernetics (2018)
[38] LeCun, Y., Cortes, C., Burges, C.J.: Mnist handwritten digit database. AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist **2** (2010)
[39] Divakar, N., Babu, R.V.: Image denoising via cnns: An adversarial approach. In: New Trends in Image Restoration and Enhancement, CVPR workshop. (2017)
[40] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. (2014) 3104–3112
[41] Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence **35** (2013) 221–231
[42] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)
[43] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86** (1998) 2278–2324

[44] Xu, D., Song, R., Wu, X., Li, N., Feng, W., Qian, H.: Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. Neurocomputing **143** (2014) 144–152
[45] Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. IEEE transactions on pattern analysis and machine intelligence **36** (2014) 18–32
[46] Wu, S., Oreifej, O., Shah, M.: Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1419–1426
[47] Saligrama, V., Chen, Z.: Video anomaly detection based on local statistical aggregates. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2112–2119
[48] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning, ACM (2008) 1096–1103
[49] Liu, S., Bousquet, O., Chaudhuri, K.: Approximation and convergence properties of generative adversarial learning. In: Advances in Neural Information Processing Systems. (2017) 5551–5559
[50] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. International Conference on Learning Representations (2016)