A Background-Agnostic Framework With Adversarial Training for Abnormal Event Detection in Video

Mariana Iuliana Georgescu , Radu Tudor Ionescu , *Member, IEEE,*
Fahad Shahbaz Khan , *Member, IEEE,* Marius Popescu , and Mubarak Shah , *Fellow, IEEE*

Abstract—Abnormal event detection in video is a complex computer vision problem that has attracted significant attention in recent years. The complexity of the task arises from the commonly-adopted definition of an abnormal event, that is, a rarely occurring event that typically depends on the surrounding context. Following the standard formulation of abnormal event detection as outlier detection, we propose a background-agnostic framework that learns from training videos containing only normal events. Our framework is composed of an object detector, a set of appearance and motion auto-encoders, and a set of classifiers. Since our framework only looks at object detections, it can be applied to different scenes, provided that normal events are defined identically across scenes and that the single main factor of variation is the background. This makes our method background agnostic, as we rely strictly on objects that can cause anomalies, and not on the background. To overcome the lack of abnormal data during training, we propose an adversarial learning strategy for the auto-encoders. We create a scene-agnostic set of out-of-domain pseudo-abnormal examples, which are correctly reconstructed by the auto-encoders before applying gradient ascent on the pseudo-abnormal examples. We further utilize the pseudo-abnormal examples to serve as abnormal examples when training appearance-based and motion-based binary classifiers to discriminate between normal and abnormal latent features and reconstructions. Furthermore, to ensure that the auto-encoders focus only on the main object inside each bounding box image, we introduce a branch that learns to segment the main object. We compare our framework with the state-of-the-art methods on four benchmark data sets, using various evaluation metrics. Compared to existing methods, the empirical results indicate that our approach achieves favorable performance on all data sets. In addition, we provide region-based and track-based annotations for two large-scale abnormal event detection data sets from the literature, namely ShanghaiTech and Subway.

Index Terms—Abnormal event detection, anomaly detection, auto-encoders, adversarial training, security and surveillance

✦

1 INTRODUCTION

ABNORMAL events are defined as rare occurrences that deviate from the normal patterns observed in familiar events [1]. Considering the prior work on video anomaly detection [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], we can devise a non-exhaustive taxonomy of abnormal events that includes: appearance anomalies (for example, a car or a truck in a pedestrian area), short-term motion anomalies (for example, a person running or a person throwing an object), long-term

motion anomalies (for instance, loitering) and group anomalies (for example, several people running inside a public building). These anomalies, which are commonly observed in publicly available data sets [12], [13], [14], [23], typically involve people and vehicles, which usually appear in surveillance videos captured in urban areas. This is because abnormal event detection is studied in the context of video surveillance. Additionally, we should emphasize that the classification of an event as normal or abnormal always depends on the context. For instance, driving a truck on the street is considered normal, but, if the truck enters a pedestrian area, the event becomes abnormal. Considering the commonly-adopted definition of abnormal events and the reliance on context, it is difficult to obtain a sufficiently representative set of anomalies for all possible contexts, making traditional supervised methods less applicable to abnormal event detection. Therefore, the majority of anomaly detection methods proposed so far are based on outlier detection [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [24], [25], [26], [27], learning normality models from training videos containing only normal events. During inference, an event is labeled as abnormal if it deviates from the normality model. In general, existing abnormal event detection methods build the normality model using local features [7], [8], [12], [13], [14], [19], [22], [27], [28], [29], [30], global (frame-level) features [4], [9], [11], [15], [16], [17], [18], [24], [25], [26], [31], or both [3], [5], [32]. Such methods work

- *Mariana Iuliana Georgescu and Marius Popescu are with the SecurifAI and the Department of Computer Science, University of Bucharest, Bucuresti 030018, Romania.*
  E-mail: georgescu_lily@yahoo.com, popescunmarius@gmail.com.
- *Radu Tudor Ionescu is with the SecurifAI and the Department of Computer Science, University of Bucharest, Bucuresti 030018, Romania, and also with the Romanian Young Academy, Bucharest 050663, Romania.*
  E-mail: raducu.ionescu@gmail.com.
- *Fahad Shahbaz Khan is with the Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi 44737, UAE. E-mail: fahad.khan@liu.se.*
- *Mubarak Shah is with the Center for Research in Computer Vision (CRCV), Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA. E-mail: shah@crcv.ucf.edu.*

well when training and testing are conducted on the same scene. If we switch to a different scene at inference time, however, the methods based on local or global features tend to fail because the features used for the normality model are specific to the training scene. There are a few approaches that do not require any training data [33], [34], [35], [36], instead employing different algorithms for change detection at test time. Such methods can be considered scene-agnostic, but they typically obtain much lower performance levels compared to methods that rely on training data to build normality models. Other works, such as [37], tackled frame anomaly detection as an action recognition task, considering only events that are always abnormal, irrespective of the context, e.g., arson attacks, burglaries or traffic accidents. By considering only generic abnormal events, the method developed by Sultani *et al.* [37] is implicitly scene-agnostic. However, the method falls outside the commonly-accepted definition of abnormal events and is rather considered by others [1] as an action recognition method.

A handful of methods [6], [38], [39] also have the potential to become scene-agnostic, while still taking into account the reliance on context of anomalies. This is achieved by applying an object detector before feature extraction, allowing the model to learn the normality only with respect to the objects, while ignoring the background or other elements in the scene. However, such methods are affected by significant viewpoint changes, frame resolution differences, frame rate variations and different types of normal events across scenes. Such methods work well across different scenes, especially in case of background variations. Similar to the above mentioned works [6], [38], [39], we employ an object detector, analyzing the abnormality at the object level by extracting appearance or motion features to represent each object. Different from preliminary works that rely on object detection [6], [38], [39], we are the first to present a cross-database (cross-domain) evaluation, demonstrating that our proposed method is not severely affected by scene variations. Remarkably, our cross-domain results on Avenue [12], ShanghaiTech [13], Subway [23] and UCSD Ped2 [14] surpass many of the recently reported in-domain results (see Section 4.9).

We conduct experiments on four challenging benchmarks, namely Avenue [12], ShanghaiTech [13], Subway [23] and UCSD Ped2 [14], reporting favorable performance levels compared to the state-of-the-art methods [4], [5], [6], [7], [8], [9], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [24], [25], [26], [30], [31], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48]. While we report results in terms of the standard frame-level area under the curve (AUC) metric, we also report our performance levels in terms of the Region-Based Detection Criterion (RBDC) and Track-Based Detection Criterion (TBDC). These criteria were recently introduced by Ramachandra *et al.* [24], who argued that the frame-level AUC is inadequate to fully evaluate abnormal event detection systems, essentially because it does not take into consideration spatial localization, counting a frame as a correct detection even when the pixels predicted as abnormal do not overlap with the ground-truth abnormal pixels. Since Ramachandra *et al.* [24] did not provide the region-based and track-based annotations required to compute RBDC and TBDC on ShanghaiTech and Subway, we labeled these data sets ourselves. We

release the annotations along with our open source code at: https://github.com/lilygeorgescu/AED.

Relation to Preliminary CVPR 2019 Version [6]. Since our method stems from the method proposed in [6], we briefly present our preliminary work and explain our new design changes. Ionescu *et al.* [6] was the first work to propose an object-centric framework, employing a single-stage detection framework (SSD [49] with Feature Pyramid Networks (FPN) [50]) on each frame in order to extract objects of interest. Representative deep unsupervised features for normal objects are learned using three convolutional auto-encoders (CAEs), one for appearance and two for motion. Upon training the auto-encoders, the concatenated latent features are clustered using the k-means algorithm to obtain clusters representing various types of normality. For each normality cluster, an SVM classifier is trained to discriminate the corresponding cluster from the rest, using the one-versus-rest scheme. During inference, the one-versus-rest SVM is applied to obtain a normality score for each detected object. The maximum among the scores assigned by the one-versus-rest SVM with respect to each normality cluster is the normality score for a given test sample. While our preliminary framework attained state-of-the-art results at the time of publication [6], we observed that the auto-encoders sometimes produce excessively good reconstructions for abnormal examples, resulting in a higher false negative rate. Furthermore, the previous framework contains multiple components that are not integrated into an end-to-end pipeline. In order to adopt an end-to-end processing pipeline in our current work, we remove the k-means clustering and the one-versus-rest classification steps, replacing them with three binary classifiers (two for motion and one for appearance) that are trained to discriminate between normal examples from the training set and pseudo-abnormal examples from a generic data set. Additionally, we make significant changes to the auto-encoders. First, we introduce skip connections and two decoder branches, one for adversarial training and one for object segmentation. To cope with the lack of anomalies during training, we introduce a generic set of out-of-domain data samples that play the role of abnormal samples, which we call "pseudo-abnormal". The pseudo-abnormal examples are database-agnostic, meaning that they can be used for any type of scene, as demonstrated throughout our experiments. Indeed, we utilize the same pseudo-abnormal examples for every data set that we experiment with. We emphasize that the auto-encoders are not supposed to reconstruct the pseudo-abnormal examples, i.e., we expect good reconstructions for normal data samples only. To address the issue regarding the excessively good reconstructions for abnormal examples observed in the preliminary framework [6], we perform adversarial training on the pseudo-abnormal examples to prevent the auto-encoders from generalizing to such data samples, inherently inducing the same behavior for abnormal examples. Indeed, adversarial training helps us to obtain poor reconstructions for abnormal objects, such as bicycles or cars in pedestrian areas. One last difference from our previous work [6] is to replace image gradients with optical flow. In summary, we propose the following changes with respect to our previous work [6]: (i) to replace the k-means clustering step with binary classifiers that distinguish between normal and pseudo-abnormal objects, (ii) to integrate skip connections, (iii) to add a segmentation decoder,

(iv) to perform adversarial training and (v) to use optical flow instead of image gradients. We emphasize that the proposed changes lead to a significantly different model, bringing consistently superior performance over the preliminary work [6], especially in terms of the recently-introduced RBDC and TBDC metrics [24].

In summary, our contribution is threefold:

- We propose a novel framework that leverages adversarial training in the context of abnormal event detection in video.
- We show that our framework is background agnostic by performing a series of cross-database experiments.
- We provide region-level and track-level annotations for ShanghaiTech and Subway, allowing future works to report results in terms of RBDC and TBDC.

The rest of this paper is organized as follows. We present related work on abnormal event detection in video in Section 2. Our method is described in detail in Section 3. We present the anomaly detection experiments and results in Section 4. Finally, our conclusions are drawn in Section 5.

2 RELATED WORK

Most of the recent works treat abnormal event detection as an outlier detection task [2], [3], [4], [5], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [24], [25], [26], [27], [28], [30], [32], [51], learning a model using only normal data. Then, at inference time, the events that diverge from the normality model are labeled as abnormal. Existing abnormal event detection methods can be categorized into distance-based approaches [6], [7], [17], [24], [25], [27], [29], [31], [52], [53], [54], reconstruction-based models [5], [11], [12], [13], [16], [18], [26], [32], [41], [42], probabilistic models [2], [3], [8], [14], [15], [23], [55], [56], [57] and change detection methods [33], [34], [35], [36]. A handful of preliminary works [3], [12], [28], [32], [51] proposed to build a dictionary of atoms representing normal events, labeling the events that are not represented in the dictionary as abnormal. For example, Dutta *et al.* [28] proposed an approach that builds a model of familiar events from training data using a sparse coding objective. Then, the model is incrementally updated in an unsupervised manner as new patterns are observed in the test data.

Other recent approaches [5], [11], [13], [17], [20], [25], [26], [27], [31], [38], [39], [56] employed deep learning in order to detect the anomalous frames in a video. For instance, Liu *et al.* [11] proposed to detect abnormal frames by predicting the next frame in the video, given the previous four frames. Their hypothesis is that an abnormal frame should be harder to predict than a normal one. Thus, the peak signal-to-noise ratio between the predicted frame and the original frame is expected to be lower for abnormal frames. More recently, Ramachandra *et al.* [25] used video-patches to detect the anomalies in videos. They stored a set of exemplars for each region in the video frames, comparing each video patch from the test video to each exemplar from the corresponding region. The minimum distance is interpreted as the anomaly score.

Similar to our work, which learns features in an unsupervised manner, there are a few works that employ unsupervised learning steps for abnormal event detection [5], [20], [28],

[42], [51]. There are also some works that are completely unsupervised [33], [34], [35], [36], requiring no training data to perform anomaly detection. Different from such works, our method employs an object detector trained with supervision. The object detector helps our method to become background-agnostic. Another way we introduce supervision into our framework is through pseudo-abnormal examples. Our auto-encoders learn to output poor reconstructions for pseudo-abnormal examples through adversarial training, while still being capable of reconstructing normal patterns. We also use the pseudo-abnormal examples to train binary classifiers on top of the auto-encoder features and reconstructions, in a supervised way.

More closely related to our work, some methods use auto-encoders in order to learn useful features [5], [20], [27], [41], [42]. For example, Nguyen *et al.* [41] used an auto-encoder with two branches to detect anomalous frames. The input of the auto-encoder is the current frame of the video. The first branch predicts the frame intensity, while the second branch predicts the motion between the current frame and the next frame. Unlike the first branch, which is a standard auto-encoder, the second branch follows a U-Net architecture. Different from Nguyen *et al.* [41] and other methods based on auto-encoders, we train our auto-encoders on detected objects, which helps our method to better localize anomalies and to become background-agnostic. We also employ adversarial training, which enables us to obtain good reconstructions for normal objects only.

There are also some works [26], [40], [41] that use Generative Adversarial Networks (GANs) [58] in order to detect abnormal events in videos. For instance, Nguyen *et al.* [41] used a discriminator to distinguish between the generated optical flows and the real ones. Therefore, in order to reconstruct the motion, their network is also guided by the discriminator to produce realistic optical flows. Our method does not employ GANs, i.e., we do not train a generator and a discriminator in an adversarial fashion. In our case, the adversarial training consists of propagating the reversed gradients from an adversarial decoder through the encoder, forcing the auto-encoder to output bad reconstructions for pseudo-abnormal examples. While we integrate a binary classifier after each auto-encoder, the respective classifier is only trained to discriminate between normal and pseudo-abnormal reconstructions (both kinds of examples being generated, not real), without interfering with the learning process of the auto-encoder. We refrain from referring to our binary classifiers as discriminators to avoid any confusion with GANs. Unlike these previous works [26], [40], [41], we are the first to perform adversarial training using a generic set of pseudo-abnormal examples.

We note that auto-encoders [59], [60], [61] and adversarial models [62], [63] have been applied to anomaly detection in other domains, e.g., time series [59], images [60], [61], [62] or network traffic analysis [60], [63]. Abnormal event detection is commonly viewed as an independent domain, requiring the design of specific methods that take into account both motion and appearance as well as other particularities of the video domain. Hence, we consider methods such as [59], [60], [61], [62], [63] to be distantly related.

3 METHOD

3.1 Motivation

We distinguish two realistic requirements that are especially desired when designing a framework for abnormal event detection in real-word scenarios. Based on these requirements, we introduce a set of design choices in our proposed framework.

The first requirement is to learn from training videos containing only normal events, deeming supervised learning methods needing both positive (normal) and negative (abnormal) samples unusable for abnormal event detection. Nonetheless, we believe that including any form of supervision is an important step towards obtaining better performance in practice. Motivated by this, we incorporate two approaches for including supervision into our framework. The first approach is to employ a single-shot object detector [64], which is trained with class and bounding-box supervision, in order to obtain object detections that are subsequently used throughout the rest of the processing pipeline. The second approach consists of gathering a large and generic pool of pseudo-abnormal examples, substituting the need for abnormal examples during training.

The second requirement is to apply the same model on multiple scenes with different backgrounds, eliminating the need to retrain the model for each and every scene. For example, a model that is trained on road traffic is expected to work well on multiple road traffic scenes. This has motivated us to design a background-agnostic framework. Since we employ an object detector and analyze abnormality at the object level, our framework is close to being background-agnostic. We take it a step further and equip our auto-encoders with a segmentation branch to focus on reconstructing only the corresponding segment in each bounding box, thus completely ignoring the background. We conduct cross-database experiments to demonstrate that our framework is indeed background-agnostic.

3.2 Overview

Our abnormal event detection pipeline is illustrated in Fig. 1. An object detector is applied on the video, resulting in a set of object detections. For each object detection, optical flow maps are computed with respect to the previous and next frames. The object detections are given as input to an appearance auto-encoder, while the optical flow maps are given as input to two motion auto-encoders. The auto-encoders learn to reconstruct the detections and flow maps extracted from the input video. At the same time, the auto-encoders are prevented from learning to reconstruct examples from the pool of pseudo-abnormal training samples passing through the encoder and the adversarial decoder branch. The appearance auto-encoder has a third decoder branch that learns to reconstruct object segments. The segmentation branch helps the model to focus on the foreground object, ignoring the background inevitably caught inside each bounding box. The segmentation branch is also depicted in Fig. 1. The absolute differences between the inputs and reconstructions are subsequently used to learn three binary classifiers for discriminating between normal examples (labeled as positive) and pseudo-abnormal examples (labeled as negative). During inference, the average scores (class probabilities) provided by the three classifiers represent normality scores associated to the input object detections.

3.3 Object Detection and Preprocessing

In this work, we employ a fast single-stage object detection framework, YOLOv3 [64], that is pre-trained on MS COCO [65]. We do not fine-tune the object detector on abnormal event detection data sets due to the lack of bounding box annotations. Throughout our experiments, we did not observe any significant false negatives due to the employment of a pre-trained object detector. This is likely due to the fact that anomalous events are mostly associated with humans and their interaction with other humans or vehicles, e.g., humans riding bikes, people fighting, people stealing or throwing backpacks and so on. Such categories (human, bike, backpack, etc.) are present in the MS COCO data set. Nevertheless, the detector can be retrained in case of any missing categories.

We opted for YOLOv3 due to its combined advantage of superior detection performance (in terms of mean Average Precision) and high speed (72 frames per second on a single GPU). We are interested in achieving an optimal trade-off between accuracy and speed, YOLOv3 being an excellent choice in this regard. The object detector is applied on each frame t, resulting in a set of bounding boxes at each frame. To obtain the input for the appearance auto-encoder, we crop the objects according to the detected bounding boxes, then, we convert the resulting image crops to grayscale. To obtain segmentation maps to be used as ground-truth labels during training time, we employ Mask R-CNN [66]. Since Mask R-CNN is slower than YOLOv3, we only use it to obtain segments, necessary at training time, but not during inference. To obtain the motion representation corresponding to an object, we use optical flow. We compute the optical flow using the pre-trained version of SelFlow [67], applying it on each tuple of three consecutive frames. More precisely, to compute the flow at frame t, the SelFlow network receives the frames $t-1$, t and $t+1$. The resulting optical flow is composed of the forward and the backward optical flow maps. In order to extract the optical flow corresponding to each object in a frame t, we apply the bounding boxes detected by YOLOv3 to crop out the corresponding flow maps. As we obtain two flow maps from SelFlow, we employ two motion auto-encoders, one being used to encode the forward motion and the other to encode the backward motion. This is consistent with our preliminary framework [6], which also employed two motion auto-encoders, although these were trained on top of image gradients instead of optical flow maps. Our motion auto-encoders are trained on tensors with two channels, one for the orientation and the other for the magnitude of the motion vectors.

3.4 Pseudo-Abnormal Examples

We gathered a collection of pseudo-abnormal examples that are problem-agnostic, enabling us to use the same collection regardless of the data set. The purpose of using the pseudo-abnormal examples is twofold. First of all, pseudo-abnormal images force the auto-encoders to forget generic reconstruction patterns, projecting only normal objects to the learned
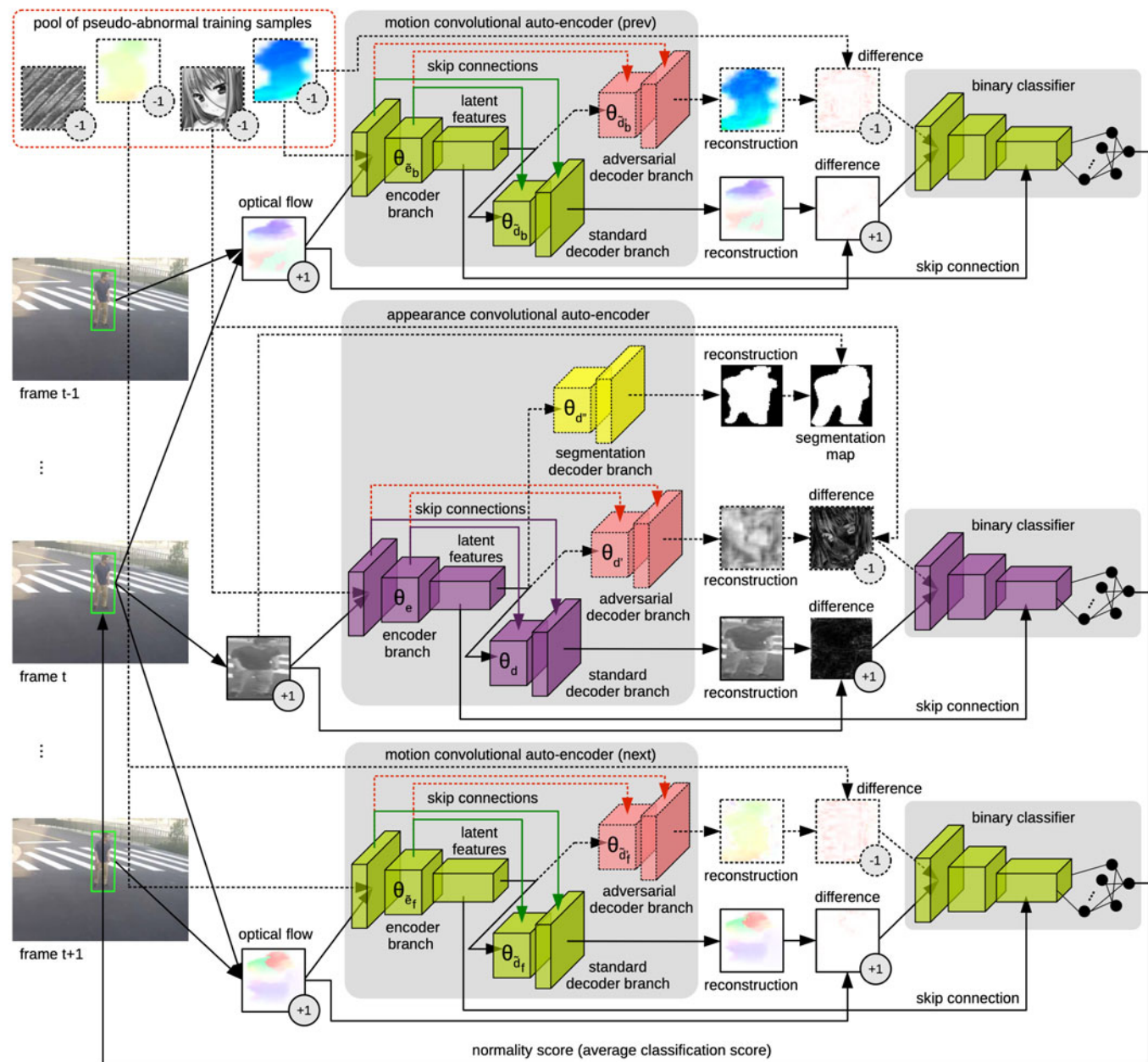
Fig. 1. Our anomaly detection framework based on training convolutional auto-encoders with skip connections on top of object detections. In the learning phase, pseudo-abnormal examples are used to train the adversarial decoder branch using gradient ascent. The absolute differences between the inputs and the reconstructions are provided as input to a binary classifier corresponding to each convolutional auto-encoder. In the inference phase, we can label a test sample as abnormal if the average classification score is negative, i.e., the sample is labeled as pseudo-abnormal. Components represented in dashed lines are removed during inference. Best viewed in color.

manifold. Second of all, they represent an excellent way to fill in for the lack of abnormal data that would be required to train the binary classifiers using classic supervision.

The collection of pseudo-abnormal examples does not contain objects that can appear in real abnormal event detection scenarios, such as people, cars, bicycles, firearms and so on. Instead, it contains texture images [68], flower images [69], anime images, butterfly images and some categories from Tiny ImageNet that are unrelated to abnormal event detection, such as *hourglass* and *acorn*. In total, the data set of pseudo-abnormal examples is composed of $66,918$ images. Since an object can be classified both as normal or abnormal, the distinction being based on the context, we completely refrain from adding any object class that is likely to appear in

a video surveillance scene. To obtain pseudo-abnormal motion patterns, we compute optical flow maps on frame triplets selected at time $t - k$, t and $t + k$ from the (normal) training video. We choose $k = \{3, 4, 5, 6\}$ for the experiments, artificially magnifying the motion stored in the pseudo-abnormal optical flow maps.

3.5 Architecture

Our neural architecture is composed of three independent processing streams (appearance, backward motion and forward motion), which are combined using a late fusion strategy. Each stream uses a convolutional auto-encoder (CAE) followed by a binary neural network classifier. The three auto-encoders share the same lightweight architecture, with

only their input and trainable weights being different. One CAE takes as input cropped grayscale images of objects and learns to encode the appearance of objects in its latent space. The other two CAEs receive as input the orientation and the magnitude of the motion vectors stored in the optical flow maps, learning to represent motion in their latent spaces. The input size for the appearance CAE is 64×64, while the input size for the motion CAEs is $64 \times 64 \times 2$.

Each encoder is composed of three convolutional (conv) layers, each followed by a max-pooling layer with a filter size of 2×2 applied at a stride of 2. The conv layers are formed of 3×3 filters. Each conv layer is followed by Rectified Linear Units (ReLU) [70] as the activation function. The first two conv layers consist of 32 filters, while the third layer consists of 16 filters. The latent representation is composed of 16 activation maps of size 8×8.

Each decoder starts with an upsampling layer, increasing the spatial support of the activation maps by a factor of $2 \times$. The upsampling operation is based on nearest neighbor interpolation. After upsampling, we apply a conv layer with 16 filters of 3×3. The first upsampling and conv block is followed by another two upsampling and conv blocks. The last conv layer of an appearance decoder is formed of a single conv filter, while the last conv layer of a motion decoder is formed of two filters. In both cases, the number of filters in the last conv layer is chosen such that the size of the output matches the size of the input.

We note that the appearance CAE incorporates one encoder e and three decoder branches. The first decoder d is used to reconstruct the normal objects, the second decoder d' is used to decode the pseudo-abnormal objects and the third decoder d'' is used to generate a mask that segments the object and ignores the background of the input image. Different from the appearance CAE, the motion CAEs incorporate one encoder $\tilde{e}_*$ and only two decoder branches, one denoted by $\tilde{d}_*$, which reconstructs the normal objects, and the other denoted by $\tilde{d}'_*$, which decodes the pseudo-abnormal examples, where $*$ can be replaced with values from the set $\{b, f\}$, where b represents the backward motion and f represents the forward motion.

In order to distinguish between normal and pseudo-abnormal examples, we train binary classifiers using classic supervision. We designate a binary classifier for each of the three streams. The actual input of a binary classifier is the absolute difference between the input and the output of the corresponding CAE. In a set of preliminary experiments, we tried to use directly the reconstructions instead of the absolute differences, but the results were considerably worse. For the appearance CAE, the absolute difference is a matrix of 64×64 components, while for the motion CAEs, the absolute difference is a tensor of $64 \times 64 \times 2$ components. Since the inputs of the binary classifiers are of the same shape as the CAEs, we employ the same sequence of three conv and max-pooling layers as in the encoders. The resulting activation maps are subsequently passed through a neural network that follows the LeNet architecture [71], except that the spatial support of the conv filters is always 3×3 and the pooling operation is max-pooling instead of average-pooling. In summary, a binary classifier is composed of five conv layers, a fully-connected layer and a Softmax classification layer.

We note that our architecture is also equipped with skip connections. Each conv layer in an encoder has skip connections to the corresponding conv layer in each standard or adversarial decoder that belongs to the same CAE as the encoder, following the U-Net architecture [72]. Instead of concatenating the features as in U-Net, we sum up the corresponding activation maps, as in ResNet [73]. Skip connections are also added between the last conv layer of an encoder and the third conv layer of the corresponding binary classifier. Since the first part of the binary classifier coincides with the encoder in terms of architecture, we can sum up the corresponding activation maps, both forming a tensor of $8 \times 8 \times 16$ components.

3.6 Training the Auto-Encoders

We train the auto-encoders using adversarial training, propagating the reversed gradients from the adversarial branch of each CAE through the corresponding encoder. Let θ_e, θ_d, $\theta_{d'}$ and $\theta_{d''}$ be the parameters of the appearance encoder e, the main appearance decoder d, the adversarial appearance decoder d' and the segmentation decoder d'', respectively. Analogously, let $\theta_{\tilde{e}_*}$, $\theta_{\tilde{d}_*}$ and $\theta_{\tilde{d}'_*}$ be the parameters of the motion encoder $\tilde{e}_*$, the main motion decoder $\tilde{d}_*$ and the adversarial motion decoder $\tilde{d}'_*$, where $* \in \{b, f\}$. For the motion auto-encoders, the loss function for reconstructing an input optical flow $\tilde{x}_*$ of $h \times w \times c$ components is the pixel-wise mean squared error:

$$\mathcal{L}_{*\text{mot-rec}}(\tilde{x}_*, \hat{x}_*) = \frac{1}{h \cdot w \cdot c} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{c} \left(\tilde{x}_{*ijk} - \hat{x}_{*ijk} \right)^2,$$

(1)

where $\hat{x}_* = \tilde{d}_*(\tilde{e}_*(\tilde{x}_*, \theta_{\tilde{e}_*}), \theta_{\tilde{d}_*})$ is the main output of a motion auto-encoder, $\forall * \in \{b, f\}$. We note that $h = w = 64$, while $c = 2$. Similarly, we define the loss for the adversarial branch of the motion CAEs as follows:

$$\mathcal{L}_{*\text{mot-adv}}(\tilde{x}_*, \bar{x}_*) = \frac{1}{h \cdot w \cdot c} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{c} \left(\tilde{x}_{*ijk} - \bar{x}_{*ijk} \right)^2,$$

(2)

where $\bar{x}_* = \tilde{d}'_*(\tilde{e}_*(\tilde{x}_*, \theta_{\tilde{e}_*}), \theta_{\tilde{d}'_*})$ is the output of an adversarial motion decoder, $\forall * \in \{b, f\}$. In the above equations, it is important to highlight that the notations $\tilde{e}_*$, $\tilde{d}_*$ and $\tilde{d}'_*$ are interchangeably used to denote components of both motion auto-encoders. In other words, $*$ joins the notations $\tilde{e}_b$, $\tilde{d}_b$ and $\tilde{d}'_b$ designated for the components of the backward motion CAE and the notations $\tilde{e}_f$, $\tilde{d}_f$ and $\tilde{d}'_f$ designated for the components of the forward motion CAE. We note that the independent notations are used in Fig. 1.

For the appearance CAE, we also have the segmentation decoder d'', which is trained jointly with the principal decoder d. Hence, the loss function for reconstructing an input x and a segmentation map s is the sum of the pixel-wise mean squared error computed with respect to main decoder and the logistic error computed with respect to the segmentation decoder:

$$\mathcal{L}_{\text{app-rec}}(x, \hat{x}, s, \hat{s}) = \frac{1}{h \cdot w} \sum_{i=1}^{h} \sum_{j=1}^{w} \left(x_{ij} - \hat{x}_{ij}\right)^2$$

$$+ \frac{1}{h \cdot w} \sum_{i=1}^{h} \sum_{j=1}^{w} -s_{ij} \cdot \log\left(\hat{s}_{ij}\right) - (1 - s_{ij}) \cdot \log\left(1 - \hat{s}_{ij}\right), \tag{3}$$

where x is a grayscale image, $\hat{x} = d(e(x, \theta_e), \theta_d)$ is the main output of the auto-encoder and $\hat{s} = d''(e(x, \theta_e), \theta_{d''})$ is the output of the segmentation decoder. We notice that the segmentation mask is estimated from the input x, the ground-truth segmentation mask s being used only for the comparison with the output $\hat{s}$. According to the described architecture, the size of both inputs is $h = w = 64$. As the segmentation map is a binary map separating the foreground object from the background, we opted for the logistic loss on the segmentation branch. The loss for the adversarial branch of the appearance CAE is defined analogously to Eq. (2):

$$\mathcal{L}_{\text{app-adv}}(x, \bar{x}) = \frac{1}{h \cdot w} \sum_{i=1}^{h} \sum_{j=1}^{w} \left(x_{ij} - \bar{x}_{ij}\right)^2, \tag{4}$$

where $\bar{x} = d'(e(x, \theta_e), \theta'_d)$ is the output of the adversarial decoder. Since our goal is to obtain poor reconstructions for the pseudo-abnormal examples, we train the decoders d' and $\tilde{d}'_*$ as adversaries.

For the motion auto-encoders, the parameters $\theta_{\tilde{d}_*}$ are updated to optimize the loss defined in Eq. (2), while the parameters $\theta_{\tilde{e}_*}$ of the encoder are updated to fool the decoder $\tilde{d}'_*$. This leads to the following update rules for the parameters:

$$\theta_{\tilde{d}_*} \leftarrow \theta_{\tilde{d}_*} - \eta \cdot \frac{\partial \mathcal{L}_{*\text{mot-rec}}}{\partial \theta_{\tilde{d}_*}}, \tag{5}$$

$$\theta_{\tilde{d}'_*} \leftarrow \theta_{\tilde{d}'_*} - \eta \cdot \frac{\partial \mathcal{L}_{*\text{mot-adv}}}{\partial \theta_{\tilde{d}'_*}}, \tag{6}$$

$$\theta_{\tilde{e}_*} \leftarrow \theta_{\tilde{e}_*} - \eta \frac{\partial \mathcal{L}_{*\text{mot-rec}}}{\partial \theta_{\tilde{e}_*}} + \eta \cdot \lambda \cdot \frac{\partial \mathcal{L}_{*\text{mot-adv}}}{\partial \theta_{\tilde{e}_*}}, \tag{7}$$

where η is the learning rate and λ is a weight for the reversed gradient. We note that the encoder is trained using gradient descent with respect to the main decoder and gradient ascent with respect to the adversarial decoder. To ensure convergence, λ must be less than 1 (otherwise, the gradient ascent step will be greater than the gradient descent step). As suggested in [74], we set $\lambda = 0.2$ for our experiments. For the appearance auto-encoder, the parameter update rules are equivalent, while only adding the standard gradient descent for the segmentation decoder:

$$\theta_d \leftarrow \theta_d - \eta \cdot \frac{\partial \mathcal{L}_{\text{app-rec}}}{\partial \theta_d}, \tag{8}$$

$$\theta_{d'} \leftarrow \theta_{d'} - \eta \cdot \frac{\partial \mathcal{L}_{\text{app-adv}}}{\partial \theta_{d'}}, \tag{9}$$

$$\theta_{d''} \leftarrow \theta_{d''} - \eta \cdot \frac{\partial \mathcal{L}_{\text{app-rec}}}{\partial \theta_{d''}}, \tag{10}$$

$$\theta_e \leftarrow \theta_e - \eta \frac{\partial \mathcal{L}_{\text{app-rec}}}{\partial \theta_e} + \eta \cdot \lambda \cdot \frac{\partial \mathcal{L}_{\text{app-adv}}}{\partial \theta_e}. \tag{11}$$

Once again, we set $\lambda = 0.2$ for the experiments. We train the auto-encoders using the Adam optimizer [75] with the learning rate $\eta = 10^{-4}$, keeping the default values for the other parameters of Adam.

3.7 Training the Binary Classifiers

After training the auto-encoders until convergence, we remove the additional decoder branches d', d'' and $\tilde{d}'_*$, while freezing the parameters θ_e, θ_d, $\theta_{\tilde{e}_*}$ and $\theta_{\tilde{d}_*}$ of the remaining components. We then pass all the training examples, including the adversarial ones, through the main appearance and motion decoders, obtaining the final reconstructions for the training data. The next step is to compute the absolute difference between each input example and its corresponding reconstruction. For a tensor x representing the absolute difference and the binary label y associated to x, we employ the binary cross-entropy to train the classifiers:

$$\mathcal{L}_{\text{cross-entropy}}(y, \hat{y}) = -y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}), \tag{12}$$

where $\hat{y}$ represents the prediction (class probability) for sample x. In order to use the cross-entropy loss, the normal examples, taken from the training video, are labeled with $y = 1$ and the pseudo-abnormal examples are labeled with $y = 0$. Our classifiers are optimized using Adam [75] with a learning rate of 10^{-3}.

3.8 Inference

During inference, we run the YOLOv3 detector to determine the bounding boxes of objects in the current frame. We then compute the optical flow maps for the entire frame. For each detected object, we apply the CAEs (without the additional decoder branches d', d'' and $\tilde{d}'_*$) to obtain the appearance and motion reconstructions. Then, we compute the absolute differences and pass them to the binary classifiers. Because we have three binary classifiers, we obtain three class probabilities for each object. A class probability is interpreted as a normality score normalized between 0 and 1. The final anomaly score for an object x is obtained by subtracting the average of the three normality scores from 1, resulting in an anomaly score between 0 and 1:

$$s(x) = 1 - \text{mean}\left(\hat{y}^{(i)}\right), \forall i \in \{1, 2, 3\}, \tag{13}$$

where $\hat{y}^{(i)}$ is a normality score provided by one of the three classifiers. We expect abnormal examples from the test video to be classified as pseudo-abnormal examples, having an abnormality score closer to 1.

By reassembling the anomaly scores of the detected objects into an anomaly map for each frame, we obtain pixel-level anomaly detections. Hence, our framework can also perform anomaly localization. When the bounding

boxes of two objects overlap, we keep the maximum anomaly score for the overlapping area. In order to make the pixel-level maps smoother, we apply a $3D$ mean filter. The frame-level anomaly score is obtained by taking the maximum inside the prediction map for the corresponding frame. We further apply a Gaussian filter to temporally smooth the frame-level anomaly scores.

4 EXPERIMENTS

4.1 Data Sets

We present results on four benchmark data sets, namely Avenue [12], ShanghaiTech [13], Subway [23] and UCSD Ped2 [14]. Although UMN [15] is among the most widely-used abnormal event detection benchmarks, we consider this data set as being small and saturated. Hence, we do not conduct experiments on UMN.

Avenue. The Avenue data set [12] consists of 16 training videos and 21 test videos. The training videos have a total of $15,328$ frames, while the test videos have 15,324 frames. The resolution of each video frame is 360×640 pixels. The original data set is annotated both at the pixel level and at the frame level. Region-level and track-level annotations are provided by Ramachandra *et al.* [24].

ShanghaiTech. The ShanghaiTech Campus data set [13] is one of the largest data sets for abnormal event detection. It contains 330 training videos and 107 test videos. The resolution of each video frame is 480×856 pixels and the data set has a total of $316,154$ frames. ShanghaiTech contains both frame-level and pixel-level annotations. We provide region-level and track-level annotations for ShanghaiTech.

Subway. The Subway surveillance data set [23] is formed of two videos, training and testing being conducted independently on the two videos. The length of one video (Entrance gate) is 96 minutes and the length of the other (Exit gate) is 43 minutes. The Entrance gate video has $144,251$ frames, while the Exit gate has $64,903$ frames. The resolution of each video frame is 384×512 pixels. For the Entrance video, we follow [3], [7] and split the video in 53 percent frames for training and 47 percent frames for testing. For the Exit video, similar to [7], [12], we use the first 15 minutes for training and the rest of the video for testing. This data set contains only frame-level labels. We provide pixel-level, region-level and track-level annotations for both videos in the Subway data set.

UCSD Ped2. The UCSD Ped2 data set [14] is formed of 16 training videos and 12 test videos. The training videos have a total of $2,550$ frames, while the test videos have $2,010$ frames. The resolution of each video frame is 240×360 pixels. As for Avenue, region-level and track-level annotations are available due to Ramachandra *et al.* [24].

4.2 Evaluation

As our first evaluation metric, we consider the area under the curve (AUC) computed with respect to the ground-truth frame-level annotations. At a given threshold, a frame is labeled as abnormal if at least one pixel inside the frame is abnormal. With some exceptions, we note that many previous works do not mention if the frame-level AUC is computed by (i) concatenating all frames then computing the score (this is the micro-averaged AUC), or by (ii) computing the frame-

level AUC for each video, then averaging the resulting scores (this is the macro-averaged AUC). We therefore report both frame-level AUC measures. Additionally, we report the Region-Based Detection Criterion (RBDC) and the Track-Based Detection Criterion (TBDC), two new metrics introduced by Ramachandra *et al.* [24]. In their work, Ramachandra *et al.* [24] demonstrated that the frame-level AUC and the pixel-level AUC are not representative metrics to evaluate abnormal event detection frameworks, thus proposing RBDC and TBDC as alternative metrics. RBDC takes into consideration every region that is detected as abnormal. If the intersection over union of a ground-truth region and a predicted region is at least β, the predicted region is considered a true positive, otherwise it is a false positive. TBDC takes into consideration the detection of tracks. A track is an abnormal event that occurs across several consecutive frames. Each track is formed of a set of regions. A track is considered detected if at least a fraction α of the ground-truth regions belonging to the track is detected. More details about these new evaluation metrics are presented in [1], [24]. Following Ramachandra *et al.* [24], we set $\alpha = 0.1$, $\beta = 0.1$ and compute the area under the ROC curve considering false positive rates that are less than or equal to 1.

As observed in [1], [24], a simple post-processing step can make the pixel-level AUC equal to the frame-level AUC. The post-processing step consists in labeling all pixels in a frame as abnormal if at least one pixel in the frame is abnormal, increasing the true positive rate without modifying the false positive rate. Hence, we refrain from independently reporting the pixel-level AUC.

4.3 Parameter and Implementation Details

In the object detection stage, we employ the YOLOv3 [64] detector, which is pre-trained on the MS COCO data set [65]. During training and inference, we keep the detections with a confidence level higher than 0.8 for Avenue and ShanghaiTech. Because Subway and UCSD Ped2 have lower frame resolutions, we set the confidence level at 0.5 for these two data sets. We employ the pre-trained Mask R-CNN [66] to obtain frame-level segmentation maps, and the pre-trained SelFlow [67] to obtain frame-level optical flow maps. The CAEs and the binary classifiers are trained from scratch in TensorFlow [76]. We train the CAEs for 20 epochs with the learning rate set to 10^{-4}, as in [6]. The binary classifiers are trained for 30 epochs with the learning rate set to 10^{-3}. In each experiment, we set the mini-batch size to 64 samples.

4.4 Preliminary Results

In Fig. 2, we present a set of qualitative results to analyze the behavior of the appearance and motion auto-encoders with and without adversarial training. For each input sample, we provide the corresponding output and the absolute difference between input and output, respectively. We notice that, without adversarial training, the auto-encoders provide excessively good reconstructions for the abnormal examples, indicating that the auto-encoders generalize well to abnormal data samples coming from a very close distribution to the training data distribution. Unfortunately, in our application domain, we do not want the auto-encoders
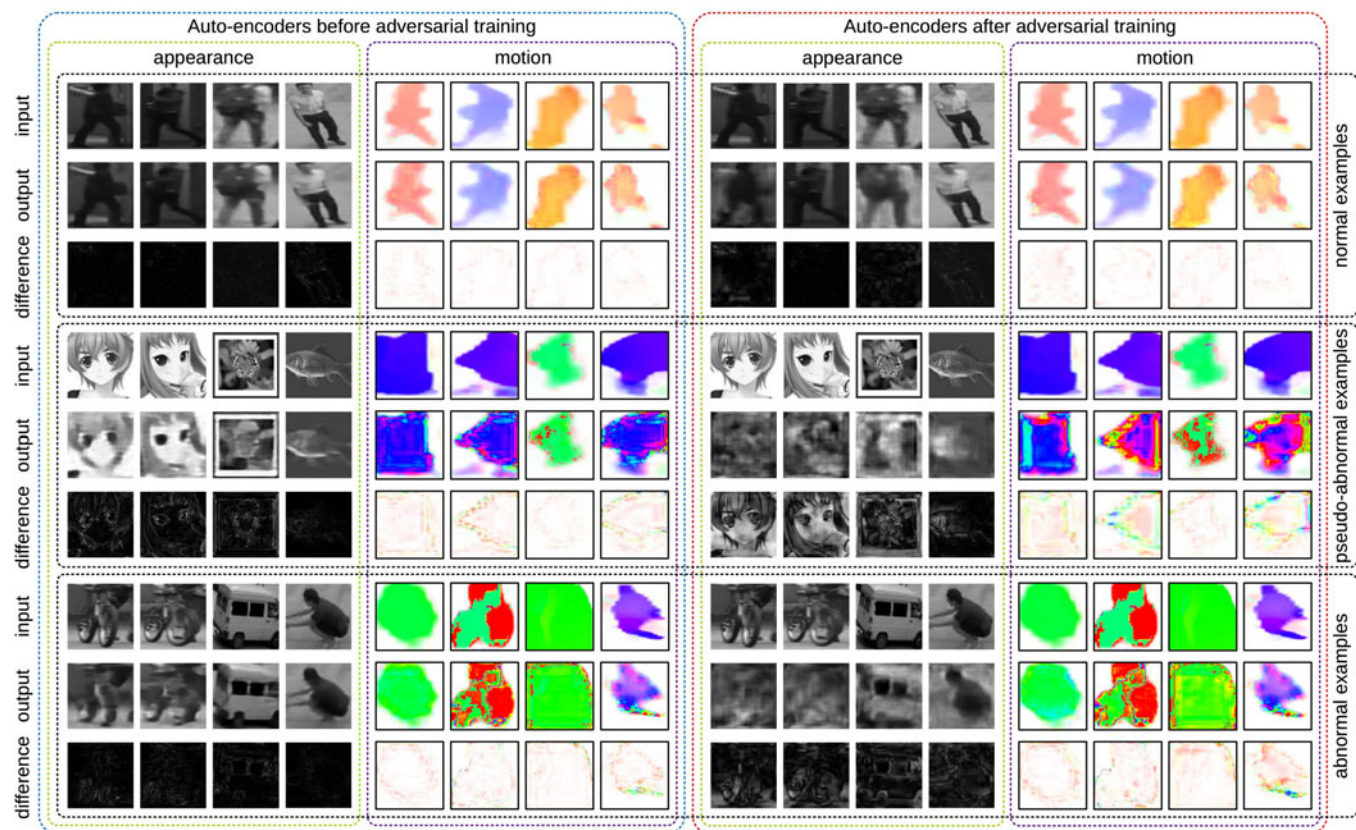
Fig. 2. Normal (top), pseudo-abnormal (middle) and abnormal (bottom) examples and optical flow maps with reconstructions provided by the appearance and the motion convolutional auto-encoders, which are trained either without adversarial training (left) or with adversarial training (right). The auto-encoders provide worse reconstructions for pseudo-abnormal and abnormal examples after adversarial training, which is the desired effect. The normal and abnormal samples are selected from the Avenue [12] and the ShanghaiTech [13] test sets, while the pseudo-abnormal examples are select from our pool of generic pseudo-abnormal examples. Best viewed in color.

to generalize to abnormal examples. Instead, we would prefer to obtain visibly worse reconstructions for the abnormal examples, thus enabling the detection of such examples. This is the main motivation behind our decision to introduce adversarial training. We observe that, in general, the auto-encoders based on adversarial training provide worse reconstructions. However, we are not interested in the quality of the reconstructions, but in the difference of quality between reconstructions for normal samples and reconstructions for abnormal samples. Indeed, we observe that the auto-encoders based on adversarial training exhibit visibly worse reconstructions for abnormal examples than for normal examples, this being the desired effect. We notice the same effect on the pseudo-abnormal examples, only at a greater level. This happens because the pseudo-abnormal examples are used in the adversarial training procedure, while the abnormal examples are selected from the test set. In summary, we conclude that introducing adversarial training is helpful in achieving the desired goal, that of obtaining visibly better reconstructions for normal examples than for abnormal examples.

4.5 Quantitative Results on Avenue

We first compare our approach with several state-of-the-art methods [4], [5], [6], [7], [9], [11], [12], [13], [16], [18], [19], [24], [25], [31], [33], [34], [35], [38], [39], [40], [41], [45], [46], [47] reporting results on the Avenue data set. The corresponding micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC and TBDC scores are presented in Table 1. The existing methods attain frame-level AUC scores between 70.2 and 89.6 percent. Notably, in terms of the frame-level AUC values, our method surpasses all exiting methods. With a micro-averaged AUC of 92.3 percent, our method is the only method surpassing the 90 percent threshold on the Avenue data set by a certain margin. There are at least two works [6], [16] that report the macro-averaged AUC. Our current approach obtains the same macro-averaged AUC as our preliminary method [6], but our micro-averaged AUC is almost 5 percent higher than the result reported in [6]. We conjecture that our approach attains superior results compared to previous methods because (i) it focuses specifically on detected objects, eliminating false positive events that are not caused by objects, and (ii) it uses adversarial training to increase reconstruction errors for abnormal objects, reducing the number of false negatives.

In terms of RBDC and TBDC, only Ramachandra et al. [24], [25] report the results on the Avenue data set. Additionally, we compute the RBDC and TBDC scores for our previous framework presented in [6] and for Liu et al. [11]. We should emphasize that the original implementation of Liu et al. [11] produces extremely low RBDC and TBDC scores (close to zero). We had to modify their framework by adding a post-processing step that removes abnormal regions smaller than a certain area, leading to the better RBDC and TBDC scores shown in Table 1. Compared to Ramachandra et al. [25], we

TABLE 1
Micro-Averaged AUC, Macro-Averaged AUC, RBDC and
TBDC Scores (in %) of Our Approach Compared to the
State-of-the-Art Methods [4], [5], [6], [7], [9], [11], [12],
[13], [16], [18], [19], [24], [25], [31], [33], [34], [35], [38],
[39], [40], [41], [45], [46], [47] on the Avenue Data Set

Method	AUC		RBDC	TBDC
	Micro	Macro		
Lu et al. [12]	80.9		-	-
Hasan et al. [5]	70.2		-	-
Del Giorno et al. [33]	78.3	-	-	-
Smeureanu et al. [31]	84.6	-	-	-
Ionescu et al. [34]	80.6	-	-	-
Luo et al. [13]	81.7	-	-	-
Liu et al. [11]	85.1	81.7*	19.59*	56.01*
Liu et al. [35]	84.4	-	-	-
Lee et al. [40]	87.2		-	-
Lee et al. [9]	90.0		-	-
Ionescu et al. [7]	88.9	-	-	-
Wu et al. [19]	86.6		-	-
Nguyen et al. [41]	86.9	-	-	-
Ionescu et al. [6]	87.4*	**90.4**	15.77*	27.01*
Tang et al. [18]	85.1		-	-
Dong et al. [4]	84.9		-	-
Park et al. [16]	-	88.5	-	-
Doshi et al. [38], [39]	86.4		-	-
Ramachandra et al. [24]	72.0		35.80	**80.90**
Ramachandra et al. [25]	87.2		41.20	78.60
Sun et al. [45]	89.6		-	-
Wang et al. [46]	87.0		-	-
Yu et al. [47]	89.6	-	-	-
Ours	**92.3**	**90.4**	**65.05**	66.85

When it is unclear if the reported frame-level AUC is micro-averaged or macro-averaged, we placed the score in the middle. All results are copied from the corresponding papers, except for those marked with asterisk (which are computed by ourselves using the official implementations). The best results are highlighted in bold.
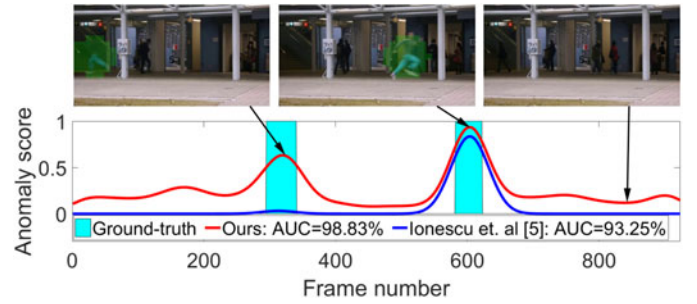

Fig. 3. Frame-level anomaly scores (on the vertical axis) provided by our current approach versus the earlier version proposed in [6], for test video 03 from Avenue [12]. Ground-truth abnormal events are represented in cyan, our scores are depicted in red and the scores of the earlier method are depicted in blue. Best viewed in color.

obtain a considerable improvement of 23.85 percent in terms of RBDC. We believe that this improvement is due to the fact that we detect objects in the scene, resulting in higher overlaps between our predicted regions and the ground-truth regions. In terms of TBDC, Ramachandra *et al.* [24] attains the state-of-the-art result of 80.90 percent. Since RBDC takes into consideration every region from the ground-truth and since we surpass Ramachandra *et al.* [24] by a large margin, we conjecture that our method is able to detect many more ground-truth regions with a lower rate of false positives. However, given the fact that our TBDC is lower than that of Ramachandra *et al.* [24], we conjecture that our method does not detect all the tracks in the ground-truth, given the maximum false positive rate of 1. Finally, comparing our method to the preliminary version proposed in [6], we report improvements of almost 50 percent in terms of RBDC and almost 40 percent in terms of TBDC, respectively. We therefore consider that the proposed framework is significantly better compared to its earlier version [6].

In Fig. 3, we present the frame-level anomaly scores (corresponding to a frame-level AUC of 98.83 percent) produced by our method versus the anomaly scores (corresponding to a frame-level AUC of 93.25 percent) produced by our previous approach [6] on test video 03 from Avenue. According to the ground-truth labels, which are also illustrated in Fig. 3, there are two abnormal events in the respective test video. Our approach is able to identify both events, without including any false positive detections, while our earlier approach [6] only identifies the second event. We would like to emphasize that adversarial training plays a key role in detecting the first abnormal event, which is missed by our preliminary framework [6].

4.6 Quantitative Results on ShanghaiTech

We further compare our method with the state-of-the-art approaches [4], [5], [6], [9], [11], [13], [16], [18], [37], [38], [39], [45], [46], [47] on the ShanghaiTech data set, presenting the corresponding results in Table 2. The state-of-the-art results in terms of the micro-averaged and the macro-averaged frame-level AUC are attained by our previous method [6]. The method proposed in the current work outperforms its previous version [6] by a margin of 4 percent in terms of the micro-averaged AUC and a margin of 4.4 percent in terms of the macro-averaged AUC, respectively. All other methods attain lower frame-level AUC scores, ranging from 60.9 to 79.3 percent. We note that there are no previous works reporting RBDC and TBDC scores on ShanghaiTech, since we are the first to provide the necessary region-level and track-level annotations for this data set. Nevertheless, we compute the RBDC and TBDC scores for our previous method proposed in [6] and for Liu *et al.* [11]. As on Avenue, we had to add a post-processing step to significantly improve the RBDC and TBDC scores of Liu *et al.* [11]. Our current method surpasses its earlier version by more than 20 percent in terms of RBDC. The improvement is even higher in terms of TBDC, the difference being 34.25 percent in favor of the current method. Compared to Liu *et al.* [11], our improvements are higher than 24 percent for both RBDC and TBDC metrics. This demonstrates that our method is able to better localize the anomalies, having a lower rate of false positives per frame. Since ShanghaiTech is one of the largest anomaly detection data sets, we consider our results reported in Table 2 as noteworthy.

In Fig. 4, we display the frame-level anomaly scores (corresponding to a frame-level AUC of 99.65 percent) of our method against the ground-truth labels on a ShanghaiTech test video with one abnormal event. On this video, we can clearly observe a strong correlation between our anomaly scores and the ground-truth labels. The preliminary framework proposed in [6] produces anomaly scores that are less

TABLE 2
Micro-Averaged AUC, Macro-Averaged AUC, RBDC and TBDC
Scores (in %) of Our Approach Compared to the State-of-the-
Art Methods [4], [5], [6], [9], [11], [13], [16], [18], [37], [38],
[39], [45], [46], [47] on the ShanghaiTech Data Set

Method	AUC		RBDC	TBDC
	Micro	Macro		
Hasan et al. [5]	60.9		-	-
Luo et al. [13]	68.0	-	-	-
Liu et al. [11]	72.8	80.6*	17.03*	54.23*
Sultani et al. [37]	-	76.5*	-	-
Lee et al. [9]	76.2		-	-
Ionescu et al. [6]	78.7*	84.9	20.65*	44.54*
Doshi et al. [38], [39]	71.6		-	-
Tang et al. [18]	73.0		-	-
Dong et al. [4]	73.7		-	-
Park et al. [16]	-	72.8	-	-
Sun et al. [45]	74.7		-	-
Wang et al. [46]	79.3		-	-
Yu et al. [47]	74.8	-	-	-
Ours	**82.7**	**89.3**	**41.34**	**78.79**

When it is unclear if the reported frame-level AUC is micro-averaged or macro-averaged, we placed the score in the middle. All results are copied from the corresponding papers, except for those marked with asterisk (which are computed by ourselves using the official implementations). The best results are highlighted in bold.
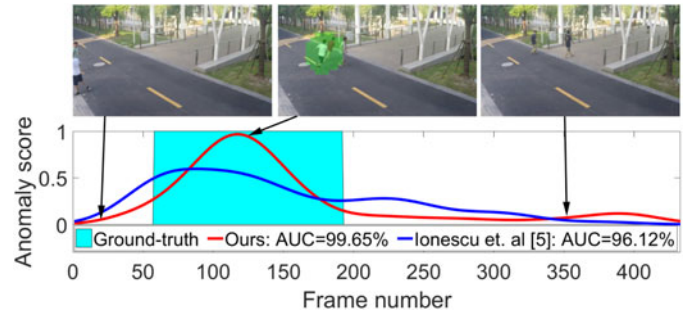

Fig. 4. Frame-level anomaly scores (on the vertical axis) provided by our current approach versus the earlier version proposed in [6], for test video 05_0024 from ShanghaiTech [13]. Ground-truth abnormal events are represented in cyan, our scores are depicted in red and the scores of the earlier method are depicted in blue. Best viewed in color.

correlated to the ground-truth, its frame-level AUC for the selected test video being 96.12 percent.

4.7 Quantitative Results on Subway

The Subway data set was originally annotated at the frame-level only. Therefore, in order to determine the ground-truth regions and tracks, we first had to annotate the data set at the pixel-level. During the manual annotation process, our annotators observed that there are some frames that were labeled as abnormal, but those frames did not contain any abnormal objects or events. In these circumstances, many false positive frames were counted as correct detections in previous works, which is wrong. In order to rectify this problem, we also relabeled the data set at the frame-level based on the identified abnormal regions, such that, if there is an abnormal region in a frame, the respective frame is considered anomalous. We note that our new labels were subject to the agreement of two independent annotators. Hence, we consider our new labels to be more accurate that the original ones.

In order to compare our work with all previous works [3], [5], [6], [7], [19], [29], [32], [33], [34], [36], we report the performance obtained using the original frame labels. Since we have access to the exact implementation of the top scoring method on the Subway data set, namely that of Ionescu et al. [7], we were able to compute the frame-level AUC, the RBDC and TBDC scores for the respective method. In addition, we compute the results of our preliminary framework [6] on Subway. The comparative results are reported in Table 3. As the Subway data set contains only one testing video per scene, the micro-averaged frame-level AUC is equivalent to the macro-averaged frame-level AUC.

Results on Subway Exit. Considering the old labels, on the Exit video, the state-of-the-art result of 95.1 percent in terms of the frame-level AUC is obtained by Ionescu et al. [7], our

method being able to achieve the fourth best score of 92.1 percent. Even though our results are lower than those of Ionescu et al. [7] on the original labels, when we switch to the new ones, we outperform their method by 0.9 percent. The same happens when we compare our current method with its previous version [6]. In terms of RBDC and TBDC, we surpass the state-of-the-art method of Ionescu et al. [7] by very large margins, namely 24.1 percent in terms of RBDC and 15.34 percent in terms of TBDC, respectively. Our RBDC and TBDC scores are also higher than those of the preliminary version [6], the differences being around 5 percent in favor of our method.

We compare our frame-level anomaly scores (corresponding to a frame-level AUC of 98.30 percent) against the ground-truth labels on a chunk of the Exit test video in Fig. 5. There are several abnormal events, which seem to be grouped into two temporal clusters. Our approach correctly identifies both groups of abnormal events, without false positives. As reference, the frame-level scores produced by the method of Ionescu et al. [7] are also included in Fig. 5. While the approach of Ionescu et al. [7] seems to identify both clusters of abnormal events, it also produces high anomaly scores for some normal frames, thus having a higher false positive rate compared to our approach.

Results on Subway Entrance. Considering the frame-level AUC on the old labels for the Entrance video, it seems that our method is surpassed by many other approaches. Hasan et al. [5] attained the state-of-the-art score of 94.3 percent, being closely followed by Ionescu et al. [7] with a score of 93.5 percent. Although our frame-level AUC score is lower on the old labels, when considering the new labels, we report an improvement of over 10 percent compared to Ionescu et al. [7]. We also obtain superior results in terms of RBDC and TBDC, respectively. Our approach outperforms the method presented in [7] by 41.27 percent in terms of RBDC, being able to detect many more regions with a lower rate of false positives. With a TBDC of 76.97 percent, we surpass the method of Ionescu et al. [7] by more than 27 percent. When compared to its previous version [6], the proposed framework attains superior results, regardless of the metric. Notably, we observe RBDC and TBDC improvements higher than 10 percent.

As for Subway Exit, we select a chunk of the Subway Entrance test video to compare our frame-level anomaly scores to those of Ionescu et al. [7] as well as to the ground-

TABLE 3
Frame-Level AUC, RBDC and TBDC Scores (in %) of Our Approach Compared to the State-of-the-Art
Methods [3], [5], [6], [7], [19], [29], [32], [33], [34], [36] on the Subway Data Set

Method	Exit				Entrance			
	AUC		RBDC	TBDC	AUC		RBDC	TBDC
	Old labels	New labels			Old labels	New labels		
Cong et al. [32]	83.0	-	-	-	80.0	-	-	-
Saligrama et al. [29]	-	-	-	-	89.1	-	-	-
Cheng et al. [3]	-	-	-	-	92.7	-	-	-
Hasan et al. [5]	80.7	-	-	-	**94.3**	-	-	-
Del Giorno et al. [33]	82.4	-	-	-	69.1	-	-	-
Ionescu et al. [34]	86.3	-	-	-	71.3	-	-	-
Wu et al. [19]	89.5	-	-	-	91.1	-	-	-
Ionescu et al. [7]	**95.1**	92.8*	23.85*	52.62*	93.5	82.4*	23.15*	49.90*
Ionescu et al. [6]	92.8*	91.5*	43.07*	62.34*	82.1*	84.3*	53.84*	66.41*
Pang et al. [36]	92.7	-	-	-	88.1	-	-	-
Ours	92.1	**93.7**	**47.95**	**67.96**	87.6	**92.2**	**64.42**	**76.97**

All results are copied from the corresponding papers, except for those marked with asterisk (which are computed by ourselves using the official implementations). The best results are highlighted in bold.

truth labels, illustrating the comparison in Fig. 6. There are three abnormal events in the selected chunk and our method provides peak anomaly scores for all three events, attaining a frame-level AUC of 94.56 percent on the selected chunk of video. However, our method also provides high anomaly scores for some frames before the first abnormal event. Nevertheless, the method of Ionescu et al. [7] exhibits high anomaly scores for many normal frames, thus having a much higher false positive rate. Consequently, the frame-level AUC obtained by Ionescu et al. [7] is only 91.76 percent.

4.8 Quantitative Results on UCSD Ped2

UCSD Ped2 is one of the most popular benchmarks in video anomaly detection, with a broad range of methods reporting results on it [4], [5], [6], [8], [9], [11], [13], [14], [15], [16], [17], [18], [20], [24], [25], [26], [30], [34], [35], [38], [39], [41], [42], [43], [44], [47], [48]. We compare the results of our approach with the results reported in literature in Table 4. Our preliminary approach [6] and five other recent methods [16], [38], [43], [47], [48] obtain frame-level scores higher than 97 percent. Nonetheless, our framework outperforms all these

methods, reaching a micro-averaged frame-level AUC of 98.7 percent and a macro-averaged frame-level AUC of 99.7 percent. In addition to [24], [25], we report the RBDC and TBDC scores for our previous method [6] and for Liu et al. [11]. Once again, we had to add a post-processing step to significantly improve the RBDC and TBDC scores of Liu et al. [11]. In terms of RBDC, our method outperforms Liu et al. [11], Ramachandra et al. [24], as well as our previous method [6]. The same can be said about the TBDC scores of these methods. In terms of RBDC and TBDC, our method is somewhat comparable to that of Ramachandra et al. [25], the former method attaining a superior TBDC score, while the latter one yielding a better RBDC score. An advantage of our method is its superiority over Ramachandra et al. [25] in terms of the frame-level AUC.

4.9 Cross-Database Quantitative Results

In order to demonstrate that our framework is indeed scene-agnostic, we conduct cross-database experiments considering seven pairs of data sets, namely ShanghaiTech→Avenue, Avenue→ShanghaiTech, Avenue→UCSD Ped2, ShanghaiTech→UCSD Ped2, Avenue→Subway Exit, UCSD Ped2→Subway
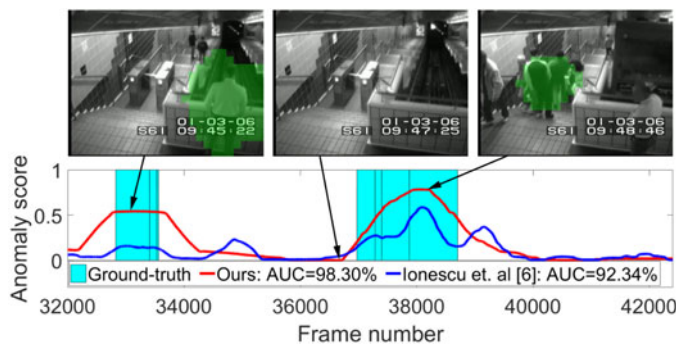

Fig. 5. Frame-level anomaly scores (on the vertical axis) provided by our approach versus the approach of Ionescu et al. [7], for a chunk of video trimmed out from Subway Exit. Ground-truth abnormal events are represented in cyan, our scores are depicted in red and the scores of Ionescu et al. [7] are depicted in blue. Best viewed in color.
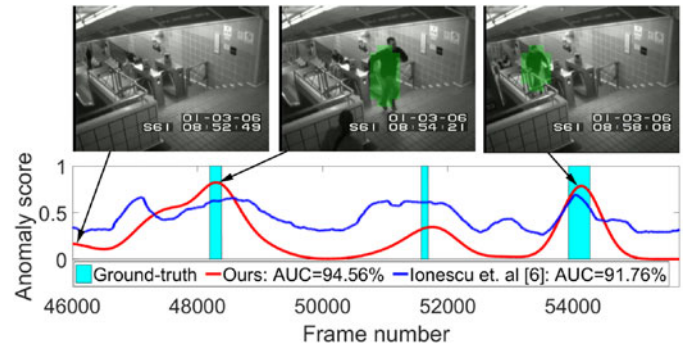

Fig. 6. Frame-level anomaly scores (on the vertical axis) provided by our approach versus the approach of Ionescu et al. [7], for a chunk of video trimmed out from Subway Entrance. Ground-truth abnormal events are represented in cyan, our scores are depicted in red and the scores of Ionescu et al. [7] are depicted in blue. Best viewed in color.

TABLE 4
Micro-Averaged AUC, Macro-Averaged AUC, RBDC and TBDC
scores (in %) of Our Approach Compared to the State-of-the-Art
Methods [4], [5], [6], [8], [9], [11], [13], [14], [15], [16], [17], [18],
[20], [24], [25], [26], [30], [34], [35], [38], [39], [41], [42], [43],
[44], [47], [48] on the UCSD Ped2 Data Set

Method	AUC		RBDC	TBDC
	Micro	Macro		
Kim et al. [8]	69.3		-	-
Mehran et al. [15]	55.6		-	-
Mahadevan et al. [14]	82.9		-	-
Hasan et al. [5]	90.0		-	-
Zhang et al. [30]	91.0		-	-
Ionescu et al. [34]	82.2	-	-	-
Luo et al. [13]	92.2	-	-	-
Ravanbakhsh et al. [26]	93.5		-	-
Xu et al. [20]	90.8		-	-
Liu et al. [11]	95.4	98.1*	38.34*	56.76*
Liu et al. [35]	87.5	-	-	-
Ravanbakhsh et al. [17]	88.4		-	-
Gong et al. [42]	94.1	-	-	-
Ionescu et al. [6]	94.3*	97.8	52.76*	72.88*
Lee et al. [9]	96.6		-	-
Nguyen et al. [41]	96.2	-	-	-
Dong et al. [4]	95.6		-	-
Doshi et al. [38], [39]	97.8		-	-
Ji et al. [43]	98.1		-	-
Lu et al. [44]	96.2		-	-
Park et al. [16]	-	97.0	-	-
Ramachandra et al. [24]	88.3		62.50	80.50
Ramachandra et al. [25]	94.0		**74.00**	89.30
Tang et al. [18]	96.3		-	-
Yu et al. [47]	97.3	-	-	-
Zaheer et al. [48]	98.1	-	-	-
Ours	**98.7**	**99.7**	69.23	**93.15**

When it is unclear if the reported frame-level AUC is micro-averaged or macro-averaged, we placed the score in the middle. All results are copied from the corresponding papers, except for those marked with asterisk (which are computed by ourselves using the official implementations). The best results are highlighted in bold.

TABLE 5
Micro-Averaged AUC, Macro-Averaged AUC, RBDC
and TBDC scores (in %) of Our Method for
a Series of Cross-Domain Experiments

Training→Test	AUC		RBDC	TBDC
	Micro	Macro		
Avenue→Avenue	92.3	90.4	65.05	66.85
ShanghaiTech→Avenue	83.6	81.0	46.56	62.76
ShanghaiTech→ShanghaiTech	82.7	89.3	41.34	78.79
Avenue→ShanghaiTech	76.3	86.3	32.55	63.89
UCSD Ped2→UCSD Ped2	98.7	99.7	69.23	93.15
Avenue→UCSD Ped2	87.0	97.2	47.43	68.58
Shanghai→UCSD Ped2	90.6	95.7	41.71	70.49
Training→Test	AUC		RBDC	TBDC
	Old	New		
Subway Exit→Subway Exit	92.1	93.7	47.95	67.96
Avenue→Subway Exit	92.1	92.4	47.54	63.93
UCSD Ped2→Subway Exit	92.8	92.8	46.60	65.73
Avenue+UCSD Ped2→Subway Exit	92.1	92.3	47.50	64.18

For each experiment, we specify the training and the test data sets.

set, evaluating it on the ShanghaiTech data set. Even though ShanghaiTech is much larger than Avenue, we are still able to obtain compelling results. In terms of the micro-averaged frame-level AUC, our cross-domain method surpasses most of the state-of-the-art methods [4], [5], [9], [13], [18], [38] (see Table 2), which were trained on the ShanghaiTech training set. In terms of RBDC, we observe a performance degradation of only 8.79 percent for our cross-domain method compared to its in-domain version. In terms of TBDC, our performance decreases from 78.79 to 63.89 percent, but we still outperform the state-of-the-art method of Ionescu *et al.* [6] by a significant margin of 19.35 percent.

Avenue→UCSD Ped2. When we train our method on Avenue and test it on UCSD Ped2, we observe significant performance drops in terms of RBDC and TBDC with respect to the in-domain baseline. The performance degradation is likely caused by camera viewpoint, frame rate and frame resolution discrepancies between Avenue and UCSD Ped2. Still, the RBDC and TBDC scores of our cross-domain framework are very close to those of the previous version of our in-domain method [6] (see Table 4). Furthermore, the macro-averaged AUC only drops by 2.5 percent with respect to the upper bound (99.7 percent) provided by our in-domain method. In terms of the frame-level AUC scores, our cross-domain model outperforms in-domain methods such as [8], [14], [15], [16], [34] (see Table 4).

ShanghaiTech→UCSD Ped2. When we train our framework on ShanghaiTech and test it on UCSD Ped2, we notice that our cross-domain method exhibits performance drops similar to those observed for the Avenue→UCSD Ped2 experiment, as the differences between ShanghaiTech and UCSD Ped2 are roughly the same. This time, our micro-averaged frame-level AUC is slightly higher (90.6 percent), leading to superior performance with respect to in-domain methods such as [5], [8], [14], [15], [17], [24], [34], [35] (see Table 4).

Avenue→Subway Exit. In terms of frame rate and frame resolution, Avenue and Subway Exit are very well aligned. Hence, it comes as no surprise that, when we train our background-agnostic framework on Avenue and apply it on Subway Exit, we observe generally low performance

Exit and Avenue + UCSD Ped2→Subway Exit. The corresponding results are presented in Table 5. We consider our in-domain results as upper bounds for the cross-database results.

ShanghaiTech→Avenue. As expected, the performance of our method on Avenue is lower when training is performed on ShanghaiTech instead of Avenue. For example, the micro-averaged AUC score degrades from 92.3 to 83.6 percent. Yet, this result is still better than many state-of-the-art results [5], [12], [13], [24] (see Table 1) that rely on models trained on the target data set (Avenue). Our cross-domain method also outperforms models that perform change detection at test time [33], [34], which do not require any training data. In terms of RBDC, our cross-domain approach outperforms the in-domain method of Ramachandra *et al.* [24] by 5.36 percent, being able to accurately detect the abnormal regions with a lower false positive rate. Remarkably, the TBDC score of our cross-domain method only drops by 4.09 percent compared to our in-domain framework. This confirms that the abnormal tracks are still being accurately detected, even though we train our framework on scenes from a different data set.

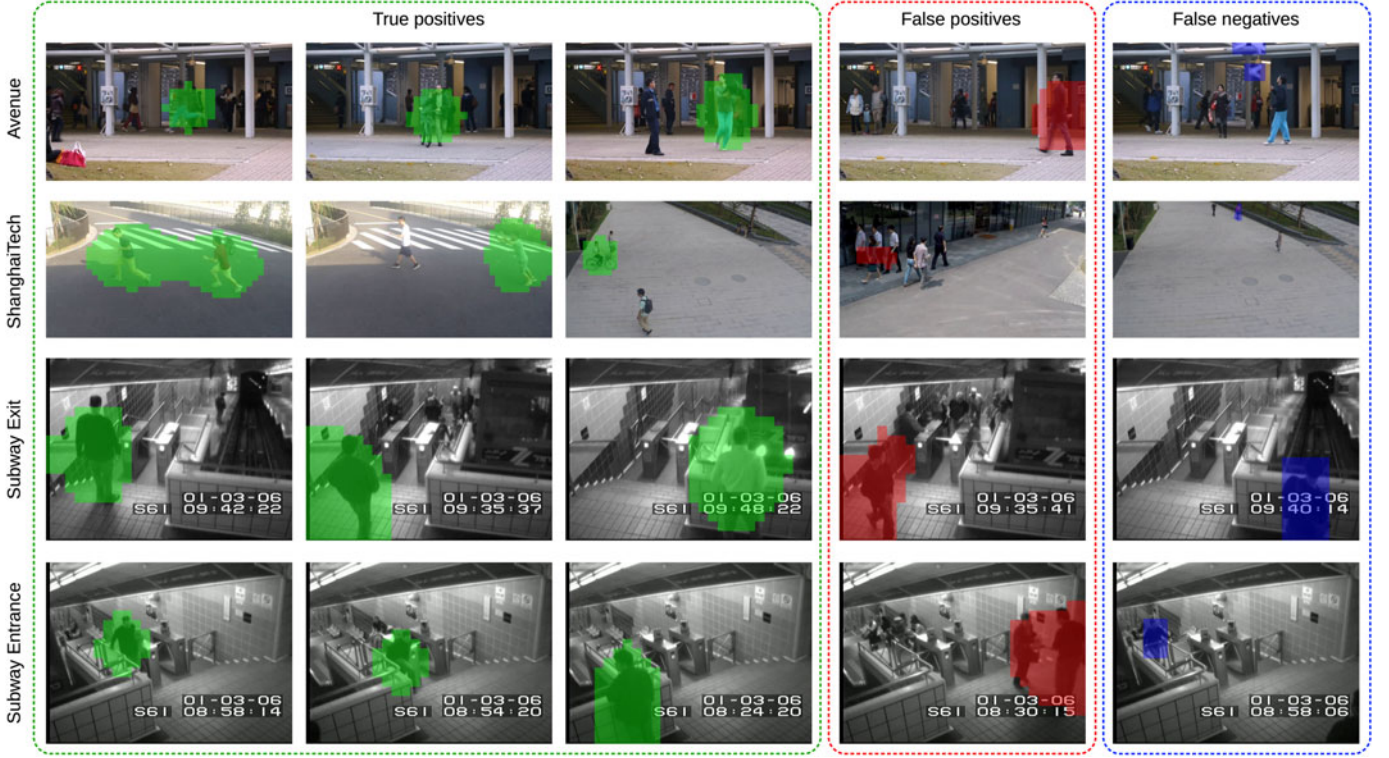*Avenue→ShanghaiTech.* In the second cross-domain experiment, we train our framework on the Avenue data

Fig. 7. True positive (left) versus false positive (middle) and false negative (right) detections of our framework. Examples are selected from the Avenue [12] (first row), the ShanghaiTech [13] (second row), the Subway Exit [23] (third row) and the Subway Entrance [23] (fourth row) data sets. Best viewed in color.

reductions with respect to the in-domain framework, the only exception being the larger TBDC drop from 67.96 to 63.93 percent. Still, the RBDC and TBDC scores reported for the Avenue→Subway Exit experiment are higher than the scores of the state-of-the-art in-domain methods [6], [7].

UCSD Ped2→Subway Exit. The frame resolution discrepancy between Subway Exit and UCSD Ped2 is only slightly larger than the discrepancy between Subway Exit and Avenue. It is therefore natural to expect minor variations between Avenue→Subway Exit and UCSD Ped2→Subway Exit. While the frame-level AUC is moderately higher when UCSD Ped2 is used as source data set, we attain a slightly better RBDC score when Avenue is used as source data set. We conclude that both Avenue and UCSD Ped2 data sets provide useful training data for Subway Exit.

Avenue + UCSD Ped2→Subway Exit. To study if we can gain additional performance by combining multiple source data sets, we considered an experiment where Avenue and UCSD Ped2 are jointly used as source data sets and Subway Exit is used as target data set. We observe insignificant performance changes when we put the source data sets together as opposed to considering them as independent source data sets. This result hints that trying to combine normal training data for multiple sources is not always helpful.

Overall. Since our cross-domain method is able to outperform many state-of-the-art methods [4], [5], [6], [7], [8], [9], [13], [14], [15], [18], [19], [24], [32], [33], [34], [38] that are trained on the target data sets, we conclude that our method is background-agnostic, providing good results even when testing is performed on scenes never seen during training.

4.10 Qualitative Results

Fig. 7 illustrates a set of true positive, false positive and false negative abnormal event localizations from Avenue, ShanghaiTech, Subway Exit and Subway Entrance. We note that the anomaly maps overlapped over the presented frames were subject to a $3D$ mean filter, hence, the shapes of the detections do not coincide with object bounding boxes.

Avenue. The examples selected from the Avenue data set are illustrated on the first row. From left to right, the true positive detections are *a person running*, *a person walking besides a bike* and *a person throwing an object*. The false positive example is *a person entering the scene* from an unusual location (closer to the camera than usual). The false negative example is represented by *papers thrown in the air*, which are not detected because *paper* is not among the classes known by the pre-trained object detector.

ShanghaiTech. The examples from ShanghaiTech are illustrated on the second row of Fig. 7. The true positive abnormal events detected by our framework are (from left to right) *two people running*, *a person jumping* and *a person riding a bike*. The false positive example consists of *two people that are detected in the same bounding box* by the object detector, generating a very unusual motion. The false negative example is *a person riding a skateboard*, which is not detected by YOLOv3 because the person is too small (just entering the scene from the far end).

Subway Exit. The examples from Subway Exit are presented on the third row. From left to right, the first two true positive detections represent *a person walking in the wrong direction*, while the third true positive example is *a person loitering*. The false positive example is *a person crossing the scene*

TABLE 6
Micro-Averaged AUC, Macro-Averaged AUC, RBDC and TBDC Scores (in %) Obtained by Making Gradual Design Changes to Our Original Method Presented in [6], Until the Framework Converges to Our Current Proposal

Object Detector	CAE Input Type	Motion Type	CAE Adversarial Branch	CAE Segmentation Branch	CAE Skip Connections	Anomaly Detection Method	Avenue				ShanghaiTech			
							AUC		RBDC	TBDC	AUC		RBDC	TBDC
							Micro	Macro			Micro	Macro		
SSD-FPN	fusion	gradients				k-means+OVR SVM	87.4	90.4	15.77	27.01	78.70	84.90	20.65	44.54
SSD-FPN	motion	gradients				MAE	77.8	78.8	16.12	30.17	72.20	79.30	24.24	49.69
SSD-FPN	appearance	-				MAE	77.6	77.4	16.77	29.90	75.60	81.90	26.30	50.44
SSD-FPN	fusion	gradients				MAE	78.3	79.2	16.42	30.01	72.40	79.70	24.35	49.74
YOLOv3	motion	gradients				MAE	72.4	74.6	47.88	48.34	65.60	76.80	19.11	44.72
YOLOv3	appearance	-				MAE	78.3	76.5	49.94	44.22	67.50	78.40	21.69	48.13
YOLOv3	fusion	gradients				MAE	74.8	75.4	51.49	48.80	65.80	77.70	19.54	45.16
YOLOv3	motion	SelFlow				MAE	80.3	81.7	50.60	52.65	69.30	79.80	22.84	52.66
YOLOv3	fusion	SelFlow				MAE	81.6	82.3	52.89	52.70	69.80	80.60	22.90	52.70
YOLOv3	motion	SelFlow	✓			MAE	77.6	81.3	44.96	49.66	69.00	80.50	23.23	52.62
YOLOv3	appearance	-	✓			MAE	76.7	79.7	48.84	45.28	67.30	79.00	21.48	48.67
YOLOv3	fusion	SelFlow	✓			MAE	80.0	83.4	49.98	51.69	69.20	81.20	23.23	52.68
YOLOv3	appearance	-	✓	✓		MAE	77.3	79.5	49.86	43.45	67.90	77.10	20.67	48.29
YOLOv3	fusion	SelFlow	✓	✓		MAE	79.9	83.7	50.72	51.38	69.20	81.10	23.11	52.71
YOLOv3	motion	SelFlow	✓			classifier on AD	84.8	85.7	46.30	62.54	78.80	87.90	34.79	72.97
YOLOv3	appearance	-	✓	✓		classifier on AD	79.7	81.5	36.98	49.66	62.30	75.10	24.21	54.28
YOLOv3	fusion	SelFlow	✓	✓		classifier on AD	86.2	85.9	50.68	63.74	77.50	87.70	33.78	69.79
YOLOv3	motion	SelFlow	✓			classifier on LF	86.5	86.2	49.96	63.10	81.60	88.60	36.43	74.31
YOLOv3	appearance	-	✓	✓		classifier on LF	89.2	83.4	48.13	53.38	63.30	74.10	30.44	66.80
YOLOv3	fusion	SelFlow	✓	✓		classifier on LF	86.6	88.2	51.79	63.26	81.90	89.10	37.31	75.33
YOLOv3	motion	SelFlow	✓			classifier on AD+LF	86.1	82.5	45.29	65.02	80.00	87.30	37.49	75.71
YOLOv3	appearance	-	✓	✓		classifier on AD+LF	87.4	83.3	49.15	53.23	64.80	77.20	31.00	68.35
YOLOv3	fusion	SelFlow	✓	✓		classifier on AD+LF	91.6	90.6	62.11	**67.55**	82.40	88.90	41.26	77.29
SSD-FPN	fusion	SelFlow	✓	✓		classifier on AD+LF	82.6	80.7	21.29	34.12	78.40	81.70	28.73	59.10
YOLOv3	motion	SelFlow	✓		concatenate	classifier on AD+LF	87.4	87.2	52.87	64.17	81.00	87.40	37.24	75.54
YOLOv3	appearance	-	✓	✓	concatenate	classifier on AD+LF	81.7	83.7	46.08	49.94	68.50	75.80	33.05	72.37
YOLOv3	fusion	SelFlow	✓	✓	concatenate	classifier on AD+LF	90.9	90.5	64.59	65.12	82.50	88.30	38.79	76.68
YOLOv3	motion	SelFlow	✓		sum	classifier on AD+LF	84.8	84.0	46.58	63.57	81.40	88.10	37.78	76.05
YOLOv3	appearance	-	✓	✓	sum	classifier on AD+LF	83.0	82.4	39.57	52.83	63.90	73.50	33.72	69.31
YOLOv3	fusion	SelFlow	✓	✓	sum	classifier on AD+LF	**92.3**	90.4	**65.05**	66.85	**82.70**	**89.30**	**41.34**	**78.79**

Best results are highlighted in bold. Notations: MAE represents the MAE between the CAE input and output; AD is short for absolute differences between the CAE input and output; LF is short for the CAE latent features.

from left to right. The false negative example is *a person loitering*.

Subway Entrance. The examples from Subway Entrance are presented on the last row of Fig. 7. From left to right, the first two true positive examples represent *a person jumping over the gate*, while the third true positive example is *a person loitering*. The false positive example is represented by *two people interacting*, one of them facing the wrong direction (according to the definition of abnormal events for Subway Entrance [23]). The false negative example is *a person walking in the wrong direction*.

4.11 Ablation Study

We perform an ablation study on the Avenue and the ShanghaiTech data sets to emphasize the effect of each component over the overall performance of our framework. The ablation results are presented in Table 6. The first row corresponds the method presented in [6], which attains good AUC scores, yet seems to underperform in terms of RBDC and TBDC. Our first design change is to eliminate the k-means clustering and the one-versus-rest (OVR) SVM. The model is left with three auto-encoders, the anomaly score being computed as the mean absolute error (MAE) between the input and the reconstruction of each CAE. Although the AUC scores record significant drops with respect to the original method [6], we observe some performance improvements in terms of RBDC and TBDC. Our next design change is the replacement of the SSD-FPN [50] with YOLOv3 [64]. This change causes further performance drops on Avenue, in terms of AUC, and on ShanghaiTech, in terms of all metrics. However, YOLOv3 brings significant improvements on Avenue, in terms of RBDC and TBDC. Even though the results obtained with YOLOv3 [64] are not very encouraging, we have decided to continue the experiments with YOLOv3 [64], as it detects four times more objects than SSD-FPN, while having a lower false positive rate. The models presented so far use the image gradients as input to the motion auto-encoders, as proposed in [6]. Our next change is to replace image gradients with optical flow maps given by SelFlow [67]. This seems to be a very important design change, leading to significant performance gains with respect to all metrics. We continue our ablation study by considering the integration of an adversarial branch in each auto-encoder, while also switching from conventional training to adversarial training. Through adversarial training, we obtained improvements of around 1 percent in terms of the macro-averaged frame-level AUC on both data sets.

In a similar manner, we updated the architecture and the loss of the appearance CAE such that the model can also output segmentation maps. This change seems to bring some slight improvements on Avenue, in terms of RBDC. Upon integrating the binary classifiers into our framework, we observe major improvements with respect to all performance metrics. We note that training the binary classifiers would not be possible without the adversarial component. Hence, the adversarial component plays an indirect yet important role in our framework, giving us a good reason to keep it. As input for the binary classifiers, we considered three options. Our first option is to use the absolute differences (AD) between inputs and reconstructions of the CAEs, obtaining improvements of more than 12 percent in terms of TBDC and more than 6 percent in terms of the micro-averaged frame-level AUC, on both data sets. Our second option, which is based on providing the latent features (LF) as input to the binary classifiers, leads to further performance gains. Our third option is to combine the absolute differences and the latent features. In the combination, the latent features are passed to the binary classifiers through some skip connections, as illustrated in Fig. 1. By combining the absolute differences and the latent features, we reach the state-of-the-art macro-averaged frame-level AUC of 90.6 percent on the Avenue data set, surpassing the method presented in [6]. Interestingly, we note that replacing YOLOv3 back with SSD-FPN, while keeping all the other design changes presented so far, does not seem to be effective, confirming that YOLOv3 is a better choice in the end. Our next design change is to integrate skip connections into the CAEs, drawing our inspiration from U-Net [72]. This design change seems to degrade performance. Our last design change is to integrate skip connections by summing up the corresponding features instead of concatenating them as in U-Net. The resulting micro-averaged frame-level AUC scores of 92.3 percent on the Avenue data set and 82.7 percent on the ShanghaiTech data set are state-of-the-art.

4.12 Running Time

Compared to our previous method described in [6], one of the most important changes in terms of time is the replacement of SSD-FPN with YOLOv3. For optimal performance, we process the video in mini-batches of 64 frames. Even though YOLOv3 [64] is much faster than SSD [49] with Feature Pyramid Networks [50], it still requires about 0.84 seconds to process 64 frames, thus running at 72 frames per second (FPS). The slowest component of our framework is SelFlow [67], which runs at 20 FPS on mini-batches of 32 frames. Due to the fact that our CAEs are very light, they require only 1.5 milliseconds to extract the latent features and to output the reconstruction for one input image. Our binary classifiers are even faster, requiring less than 1 millisecond to obtain the normality score for one object. Reassembling the anomaly scores of the detected objects into an anomaly map for each frame takes less than 1 millisecond. Putting all the components together, our framework runs at 18 FPS with a reasonable average of 5 objects per frame. We note that the reported speed of 18 FPS is for a

sequential processing pipeline on a single thread. This is to fairly compare with other methods from the literature. However, we note that running the pipeline on two threads in parallel increases the speed to 24 FPS, while still using a single GPU. Hence, our framework can process the video in real time using parallel processing. Nevertheless, we would like to note that 76 percent of the processing time is spent computing the optical flow. Therefore, one way to further speed up the running time is to replace SelFlow with a faster optical flow predictor. The reported running times were measured on a GeForce GTX 3090 GPU with 24 GB of VRAM.

5 CONCLUSION

In this paper, we have presented a set of significant design changes to our abnormal event detection approach presented at CVPR 2019 [6]. More specifically, we replaced the k-means clustering and the one-versus-rest SVM with a set of binary classifiers that learn from normal and pseudo-abnormal examples. Additionally, we modified the convolutional auto-encoders by adding adversarial and segmentation branches, as well as skip connections. Our design changes resulted in significant performance improvements in terms of both RBDC and TBDC. As a secondary contribution of our work, we release region-level and track-level annotations for the ShanghaiTech [13] and the Subway [23] data sets. Our experiments conducted on Avenue, ShanghaiTech, Subway and UCSD Ped2 indicate that our approach generally attains state-of-the-art results. We also demonstrated that our approach is background-agnostic, outperforming many approaches from the recent literature [4], [5], [6], [9], [13], [13], [18], [24], [33], [38], when our method is tested on scenes not seen during training (unlike the related works that are trained and tested on the same scenes).

In future work, we aim to study new ways to improve the computational time of our abnormal event detection framework. More precisely, we will pursue faster object detection and optical flow estimation methods, which currently account for 95 percent of our total processing time.

REFERENCES

[1] B. Ramachandra, M. J. Jones, and R. R. Vatsavai, "A survey of single-scene video anomaly detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 25, 2020, doi: 10.1109/TPAMI.2020.3040591.

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

[3] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2909–2917.

[4] F. Dong, Y. Zhang, and X. Nie, "Dual Discriminator Generative Adversarial Network for Video Anomaly Detection," *IEEE Access*, vol. 8, pp. 88170–88176, 2020.

[5] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 733–742.

[6] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, "Object-centric auto-encoders and dummy anomalies for abnormal event detection in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7842–7851.

[7] R. T. Ionescu, S. Smeureanu, M. Popescu, and B. Alexe, "Detecting abnormal events in video using Narrowed Normality Clusters," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1951–1960.

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

[9] S. Lee, H. G. Kim, and Y. M. Ro, "BMAN: Bidirectional multi-scale aggregation networks for abnormal event detection," *IEEE Trans. Image Process.*, vol. 29, pp. 2395–2408, 2019.

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

[11] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection – A new baseline," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6536–6545.

[12] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2720–2727.

[13] W. Luo, W. Liu, and S. Gao, "A revisit of sparse coding based anomaly detection in stacked RNN framework," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 341–349.

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

[15] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 935–942.

[16] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14 372–14 381.

[17] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, "Plug-and-Play CNN for crowd motion analysis: An application in abnormal event detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1689–1698.

[18] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, "Integrating prediction and reconstruction for anomaly detection," *Pattern Recognit. Lett.*, vol. 129, pp. 123–130, 2020.

[19] P. Wu, J. Liu, and F. Shen, "A deep one-class neural network for anomalous event detection in complex scenes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2609–2622, Jul. 2020.

[20] D. Xu, Y. Yan, E. Ricci, and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Comput. Vis. Image Understanding*, vol. 156, pp. 117–127, 2017.

[21] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3313–3320.

[22] X. Zhang, S. Yang, J. Zhang, and W. Zhang, "Video Anomaly detection and localization using motion-field shape description and homogeneity testing," *Pattern Recognit.*, vol. 105, 2020, Art. no. 107394.

[23] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, Mar. 2008.

[24] B. Ramachandra and M. Jones, "Street scene: A new dataset and evaluation protocol for video anomaly detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2569–2578.

[25] B. Ramachandra, M. Jones, and R. Vatsavai, "Learning a distance function with a Siamese network to localize anomalies in videos," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2598–2607.

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

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

[28] J. K. Dutta and B. Banerjee, "Online detection of abnormal events using incremental coding length," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 3755–3761.

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

[30] Y. Zhang, H. Lu, L. Zhang, X. Ruan, and S. Sakai, "Video anomaly detection based on locality sensitive hashing filters," *Pattern Recognit.*, vol. 59, pp. 302–311, 2016.

[31] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe, "Deep appearance features for abnormal behavior detection in video," in *Proceedings Int. Conf. Image Anal. Process.*, 2017, pp. 779–789.

[32] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3449–3456.

[33] A. DelGiorno, J. Bagnell, and M. Hebert, "A discriminative framework for anomaly detection in large videos," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 334–349.

[34] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2895–2903.

[35] Y. Liu, C.-L. Li, and B. Póczos, "Classifier two-sample test for video anomaly detections," in *Proc. Brit. Mach. Vis. Conf.*, 2018, Art. no. 71.

[36] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai, "Self-trained deep ordinal regression for end-to-end video anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 173–12 182.

[37] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6479–6488.

[38] K. Doshi and Y. Yilmaz, "Any-shot sequential anomaly detection in surveillance videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 934–935.

[39] K. Doshi and Y. Yilmaz, "Continual learning for anomaly detection in surveillance videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1025–1034.

[40] S. Lee, H. G. Kim, and Y. M. Ro, "STAN: Spatio-temporal adversarial networks for abnormal event detection," in *Proceedings IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 1323–1327.

[41] T.-N. Nguyen and J. Meunier, "Anomaly detection in video sequence with appearance-motion correspondence," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[42] D. Gong *et al.*, "Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1705–1714.

[43] X. Ji, B. Li, and Y. Zhu, "TAM-Net: Temporal enhanced appearance-to-motion generative network for video anomaly detection," in *Proceedings Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[44] Y. Lu, F. Yu, M. Kumar, K. Reddy, and Y. Wang, "Few-shot scene-adaptive anomaly detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 125–141.

[45] C. Sun, Y. Jia, Y. Hu, and Y. Wu, "Scene-aware context reasoning for unsupervised abnormal event detection in videos," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 184–192.

[46] Z. Wang, Y. Zou, and Z. Zhang, "Cluster attention contrast for video anomaly detection," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 2463–2471.

[47] G. Yu *et al.*, "Cloze test helps: Effective Video anomaly detection via learning to complete video events," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 583–591.

[48] M. Z. Zaheer, J.-h. Lee, M. Astrid, and S.-I. Lee, "Old is gold: Redefining the adversarially learned one-class classifier training paradigm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14 183–14 193.

[49] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[50] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[51] H. Ren, W. Liu, S. I. Olsen, S. Escalera, and T. B. Moeslund, "Unsupervised behavior-specific dictionary learning for abnormal event detection," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 28.1–28.13.

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

[53] Q. Sun, H. Liu, and T. Harada, "Online growing neural gas for anomaly detection in changing surveillance scenes," *Pattern Recognit.*, vol. 64, pp. 187–201, Apr. 2017.

[54] H. T. Tran and D. Hogg, "Anomaly detection using a convolutional winner-take-all autoencoder," in *Proc. Brit. Mach. Vis. Conf.*, 2017, Art. no. 139.

[55] Y. Feng, Y. Yuan, and X. Lu, "Learning deep event models for crowd anomaly detection," *Neurocomputing*, vol. 219, pp. 548–556, 2017.

[56] R. Hinami, T. Mei, and S. Satoh, "Joint detection and recounting of abnormal events by learning deep generic knowledge," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3639–3647.

[57] S. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2054–2060.

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

[59] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 2725–2732.

[60] Y.-g. Kim, Y. Kwon, H. Chang, and M. C. Paik, "Lipschitz continuous autoencoders in application to anomaly detection," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 2507–2517.

[61] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 665–674.

[62] L. Beggel, M. Pfeiffer, and B. Bischl, "Robust anomaly detection in images using adversarial autoencoders," in *Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2020, pp. 206–222.

[63] Z. Chen, B. Liu, M. Wang, P. Dai, J. Lv, and L. Bo, "Generative adversarial attributed network anomaly detection," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1989–1992.

[64] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv: 1804.02767*.

[65] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[66] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[67] P. Liu, M. R. Lyu, I. King, and J. Xu, "SelFlow: Self-supervised learning of optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4571–4580.

[68] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.

[69] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1447–1454.

[70] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

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

[72] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[74] R. McHardy, H. Adel, and R. Klinger, "Adversarial training for satire detection: Controlling for confounding variables," in *Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 660–665.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Representations*, 2015.

[76] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Conf. Operating Syst. Des. Implementation*, 2016, pp. 265–283.

Iuliana Georgescu received the BSc degree from the Faculty of Mathematics and Computer Science and the MSc degree in artificial intelligence from University of Bucharest, in 2017 and 2019, respectively. She is currently working toward the PhD degree at the Faculty of Mathematics and Computer Science, University of Bucharest. Although she is early in her research career, she is the first author of six papers published at conferences and journals. Her research interests include artificial intelligence, computer vision, machine learning, deep learning, and medical image processing.

Radu Ionescu (Member, IEEE) received the PhD degree from the University of Bucharest in 2013. He is currently a professor at the University of Bucharest, Romania. He has authored or coauthored more than 90 articles at international venues including the CVPR, NeurIPS, ICCV, ACL, EMNLP, and NAACL, and a research monograph with Springer. His research interests include machine learning, computer vision, image processing, computational linguistics, and medical imaging. He was the recipient of the 2014 Award for Outstanding Doctoral Research from the Romanian Ad Astra Association, the Caianiello Best Young Paper Award at ICIAP 2013, the 2017 Young Researchers in Science and Engineering Prize for young Romanian researchers, and the Danubius Young Scientist Award 2018 for Romania. He participated in several international competitions receiving top ranks, fourth place in the Facial Expression Recognition Challenge of WREPL 2013, third place in the Native Language Identification Shared Task of BEA-8 2013, second place in the Arabic Dialect Identification Shared Task of VarDial 2016, first place in the Arabic Dialect Identification Shared Tasks of VarDial 2017 and 2018, and first place in the Native Language Identification Shared Task of BEA-12 2017. He is also the editor of the journal *Mathematics*.

Fahad Khan (Member, IEEE) received the MSc degree in intelligent systems design from the Chalmers University of Technology, Sweden, and the PhD degree in computer vision from the Autonomous University of Barcelona, Spain. He is currently a faculty member at the MBZ University of AI, UAE, and Linköping University, Sweden. Before joining MBZUAI, he was a lead scientist with the Inception Institute of Artificial Intelligence, UAE. His research interests include a wide range of topics within computer vision, such as object recognition, object detection, action recognition and visual tracking. He has published articles in high-impact computer vision journals and conferences in these areas. He has achieved top ranks on various international challenges including, the Visual Object Tracking VOT, first in 2014, 2016, and 2018, second in 2015, the VOT-TIR, first in 2015 and 2016, the OpenCV Tracking, first in 2015, and the PASCAL VOC, first in 2010. He was the recipient of the Best Paper Award at ICPR 2016.

Marius Popescu received the PhD degree in 2004 with the thesis Machine Learning Applied in Natural Language Processing. He is currently an associate professor at the University of Bucharest, Department of Computer Science. His research interests include artificial intelligence, machine learning, computational linguistics, information retrieval, authorship identification, and computer vision. His achievements in these fields include, a method for word sense disambiguation that was awarded third prize at Senseval 3 in 2004, the ENCOPLOT method for plagiarism detection that won the first international competition in plagiarism detection in 2009, followed by ranking fourth at PAN@CLEF 2010 and second at PAN@CLEF 2011, a method for authorship analysis obtaining the best results in author identification at PAN@CLEF 2012, methods that ranked on third place in the Native Language Identification Shared Task of BEA-8, on fourth place in the Facial Expression Recognition Challenge of WREPL 2013, second place in the Arabic Dialect Identification Shared Task of VarDial 2016, and first place in the Native Language Identification Shared Task of BEA-12.

Mubarak Shah (Fellow, IEEE) is currently the trustee chair professor and the founding director of the Center for Research in Computer Vision at the University of Central Florida (UCF). He is the editor of an international book series on video computing, was the editor-in-chief of *Machine Vision and Applications,* and an associate editor for the *ACM Computing Surveys* and *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He was the program cochair of CVPR 2008, an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence,* and a guest editor of the special issue of the *International Journal of Computer Vision on Video Computing*. His research interests include video surveillance, visual tracking, human activity recognition, visual analysis of crowded scenes, video registration, and UAV video analysis. He was an ACM distinguished speaker and IEEE distinguished visitor speaker. He was the recipient of the ACM SIGMM Technical Achievement Award, IEEE Outstanding Engineering Educator Award, Harris Corporation Engineering Achievement Award, an honorable mention for the ICCV 2005 Where Am I? Challenge Problem, the 2013 NGA Best Research Poster Presentation, second place in Grand Challenge at the ACM Multimedia 2013 conference, and runner up for the Best Paper Award in ACM Multimedia Conference, in 2005 and 2010, respectively. He was also the recipient of the Pegasus Professor Award, University Distinguished Research Award, Faculty Excellence in Mentoring Doctoral Students, Scholarship of Teaching and Learning Award, Teaching Incentive Program Award, and Research Incentive Award from UCF. He is also a fellow of the NAI, AAAS, IAPR, and SPIE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.