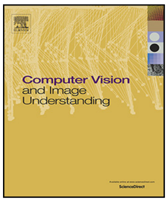
Computer Vision and Image Understanding 195 (2020) 102920

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Video anomaly detection and localization via Gaussian Mixture Fully Convolutional Variational Autoencoder

Yaxiang Fan [a,b,*], Gongjian Wen [b], Deren Li [c], Shaohua Qiu [a,b], Martin D. Levine [d], Fei Xiao [a]

[a] *National Key Laboratory of Science and Technology on Vessel Integrated Power System, Naval University of Engineering, Wuhan 430033, China*
[b] *Science and Technology on Automatic Target Recognition Laboratory (ATR), National University of Defense Technology, Changsha, 410073, China*
[c] *State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei, 430071, China*
[d] *Department of Electrical and Computer Engineering, Center for Intelligent Machines, McGill University, 3480 University Street, Montreal, H3A2A7, Canada*

ARTICLE INFO

Communicated by Nikos Paragios

Keywords:
Anomaly detection
Video surveillance
Variational autoencoder
Gaussian mixture model
Dynamic flow
Two-stream network

ABSTRACT

We present a novel end-to-end partially supervised deep learning approach for video anomaly detection and localization using only normal samples. The insight that motivates this study is that the normal samples can be associated with at least one Gaussian component of a Gaussian Mixture Model (GMM), while anomalies either do not belong to any Gaussian component. The method is based on Gaussian Mixture Variational Autoencoder, which can learn feature representations of the normal samples as a Gaussian Mixture Model trained using deep learning. A Fully Convolutional Network (FCN) that does not contain a fully-connected layer is employed for the encoder–decoder structure to preserve relative spatial coordinates between the input image and the output feature map. Based on the joint probabilities of each of the Gaussian mixture components, we introduce a sample energy based method to score the anomaly of image test patches. A two-stream network framework is employed to combine the appearance and motion anomalies, using RGB frames for the former and dynamic flow images, for the latter. We test our approach on two popular benchmarks (UCSD Dataset and Avenue Dataset). The experimental results verify the superiority of our method compared to the state of the art.

1. Introduction

Intelligent video surveillance using computer vision technology to analyze and understand long video streams, plays an irreplaceable role in public security. As an important component of intelligent video surveillance, anomalous event detection automatically discovers and identifies anomalies while monitoring an ever-changing scene and then takes timely measures to deal with emergencies. The major challenge to achieving this is that anomalous events are inherently difficult to define. After all, an anomaly refers to something that is different from the norm. But how different? Our approach to dealing with this issue is to invoke partially supervised learning, which requires only normal samples for training a Deep Neural Network. As a consequence, the samples that are not consistent with the normal samples are considered as anomalies.

In the past, researchers have conducted extensive research in developing so-called “hand-crafted” features to efficiently represent video events. Object trajectories of normal events (Anjum and Cavallaro, 2008; Bera et al., 2016) were extracted by employing conventional visual tracking methods to represent the movement of an object. Then, those objects corresponding to trajectories that deviated from the learnt trajectories were considered as anomalies. Since trajectory-based methods were generally found to be impractical for analyzing complex scenes, these methods were replaced by the use of local cuboids to model the trajectory path. These included low-level features such as spatio-temporal gradients (Kratz and Nishino, 2009; Lu et al., 2013), histograms of optical flow (HOF) (Cong et al., 2011), mixture of dynamic textures (MDTs) (Mahadevan et al., 2010) and acceleration features (Chen et al., 2015) are extracted from 2-D image patches or local 3-D video blocks.

A significant limitation of the methods based on handcrafted features is that they are difficult to adapt to the huge variations of anomalous events found in different scenes. Recently, following the impressive results of deep architectures on computer vision tasks such as object recognition (Simonyan and Zisserman, 2015; Krizhevsky et al., 2012), object detection (Girshick et al., 2014) and action recognition (Simonyan and Zisserman, 2014), attempts have been made to train deep networks for the task of anomalous event detection in video. Motivated by the success of deep learning technology, researchers (Xu et al., 2015; Sabokrou et al., 2016a; Hasan et al., 2016; Ravanbakhsh et al., 2017; Sabokrou et al., 2016b) began to apply it to anomalous event detection. Most of these methods utilize the deep network as the

* Corresponding author at: National Key Laboratory of Science and Technology on Vessel Integrated Power System, Naval University of Engineering, Wuhan 430033, China.
E-mail address: fanyaxiang@126.com (Y. Fan).

https://doi.org/10.1016/j.cviu.2020.102920
Received 10 September 2018; Received in revised form 15 August 2019; Accepted 29 January 2020
Available online 10 April 2020

features extraction and then train detection model, for example, a one-class SVM. However, these deep features are suboptimal because they are not designed or optimized for the whole problem (Chen and Huang, 2017).

Different from these methods mentioned above, we propose an end-to-end deep learning framework for training exclusively on the normal samples. The key idea behind our method is that the normal samples can be associated with at least one Gaussian component of the Gaussian Mixture Model (GMM). Then a test sample that cannot be associated with any Gaussian component is identified as anomaly. Our method is based on the Gaussian Mixture Variational Autoencoder[1] Tan et al. (2017), which is a model for probabilistic clustering within the framework of Variational Autoencoder (VAE) (Kingma and Welling, 2014). Similar to the Autoencoder (Bengio et al., 2013), it contains the encoder–decoder structure that permits learning a mapping from high dimensional data to a low-dimensional latent representation while ensuring a high reconstruction accuracy. Furthermore, the low-dimensional latent representation is constrained to be a Gaussian Mixture Model (GMM). The encoder–decoder structure and Gaussian Mixture constraint of the latent representations correspond to two main components of anomaly detection (Popoola and Wang, 2012): feature extraction and model construction. In fact, these two components are joint optimized in our method, which can maximize the performance of the joint collaboration. A fully Convolutional Network (FCN) that does not contain a fully-connected layer is employed for the encoder–decoder structure to preserve relative spatial coordinates between the input image and the output feature map. Over all, we called the deep network, a Gaussian Mixture Fully Convolutional Variational Autoencoder (GMFC-VAE).

Inspired by the human vision system and the two-stream hypothesis (Goodale and Milner, 1992), we employ a two-stream framework that has already yielded satisfactory results on video recognition tasks such as action recognition (Simonyan and Zisserman, 2014), action detection (Wang et al., 2016) and anomalous event detection (Xu et al., 2015). In detail, we use the GMFC-VAE to computationally simulate the two data pathways. The spatial stream operates on RGB frames and captures the appearance anomalies. For the temporal stream, dynamic flows[2] Wang et al. (2017), that is generated using a Ranking SVM formulation, instead of the conventional optical flow to capture the motion anomalies. The dynamic flow is an amalgamation of a number of *sequential optical flow* frames and can capture long-term temporal information, which optical flow cannot do.

In general, our proposed method includes three stages: training, testing and integrating. In the training stage, *image patches* of both RGB images and dynamic flows are densely sampled, and used as input for the two separate GMFC-VAE networks. This provides an opportunity to simultaneously learn both the latent representation and the Gaussian Mixture Model of the latent representation. Then during the testing stage, the latent representations of the RGB frame patches and the dynamic flow patches are obtained from the two GMFC-VAEs. This permits the computation of the conditional probability of the test patches that belong to each of the components of the Gaussian mixture model. A sample energy based method is used to detect both the appearance and motion anomalies by invoking the joint probabilities. Accordingly, all of the anomalous events are located based on both object motion and appearance. We conduct experiments on two widely available public datasets. The results of the experiments indicate that our method is very competitive compared to state-of-the-art algorithms.

In summary, the main contributions of our work are as follows:

- The detection of anomalies in a video is based on the hypothesis that the normal samples can be associated with at least one Gaussian component of the Gaussian Mixture model (GMM), while a test sample which is not associated with Gaussian components is declared to be anomaly. This is achieved by an end-to-end deep learning framework, which we refer to as a Gaussian Mixture Fully Convolutional Variational Autoencoder (GMFC-VAE). The latter is established based on the normal samples, learning feature representations of the normal samples as a Gaussian Mixture Model. To the extent of our knowledge, this is the first time that a Variational Autoencoder (VAE) framework has been considered for video anomaly detection.
- Instead of the usual optical flow, we adopted popular two-stream network to employ dynamic flows for detecting the motion anomalies.
- A sample energy based method is proposed to detect anomalies based on the joint probabilities of all of the components in the Gaussian Mixture Model.
- Experiments are used to evaluate our approach on two public datasets. These demonstrate the superiority of our method compared to the state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 is a detailed presentation of the proposed approach. Experimental evaluation is given in Section 4. Finally, Section 5 concludes the paper.

2. Introduction

Anomalous event detection and localization has been extensively studied in computer vision for the past 10 years and a wide variety of methods has been introduced. There exist two main categories of anomalous event detection and localization methods: methods based on handcrafted features and deep learning. These are reviewed in this section. Then we present a brief introduction of the Variational Autoencoder (VAE), the motivation of our work.

2.1. *Methods based on handcrafted features*

Handcrafted features have been conventionally used to represent an event. According to Yuan et al. (2016), these methods can be divided into two categories: those based on trajectories and those on cuboids.

For the trajectory methods, object trajectories are first extracted using object detection and tracking. Each trajectory represents the movement of an object as a sequence of image coordinates. Thus the assumption is made that anomalous trajectories differ from normal ones. For example in Anjum and Cavallaro (2008), features are extracted from the object trajectories in traffic sequences and then clustered by a Mean-Shift Algorithm. Those trajectories that are far from cluster centers in feature-space are defined as being an anomaly. Similarly, Ouivirach et al. (2013) have proposed a method for tracking foreground blobs to obtain trajectories and then automatically train a bank of linear HMMs based on the trajectories. An anomalous event is then identified by an analysis of the patterns generated by these scene-specific statistical models. In Jiang et al. (2011), the anomalies are defined at multiple semantic levels, such as point anomaly of a video object, sequential anomaly of an object trajectory, and co-occurrence anomaly of multiple video objects. Then frequency-based analysis is exploited to automatically discover regular rules for normal events. Test samples that violate these rules are identified as anomalies. Recently, Bera et al. (2016) proposed an algorithm based on trajectory-level behavior learning. The anomaly is determined by measuring the Euclidean distance between the local and global pedestrian features of each pedestrian.

Trajectory-based methods have achieved satisfactory performance for anomalies based on speed and direction. However, most of these

[1] The approach is called Variational Deep Embedding (VaDE) in Tan et al. (2017). However, by consulting GMVAE (Dilokthanakul et al., 2017) and Rui Shu's blog (Shu, 0000), which involved work similar to Tan et al. (2017), we have chosen to call it the Gaussian Mixture Variational Autoencoder for easy understanding.

[2] To be distinguished from the terms optical flow and dynamic image.

trajectory methods have relied on object detection and tracking procedures, which are generally not very robust for crowded scenarios because these methods cannot handle occlusion problems (Feng et al., 2016; Sabokrou et al., 2018). Local cuboid-based features have also been proposed. Instead of object trajectories for representing events, these employ local features such as histograms of gradients (HOG), histograms of optical flow (HOF), spatio-temporal gradients extracted from local 2-D image patches or local 3-D video cuboids. In Jiang et al. (2011), the variations in local spatial–temporal gradients are used to represent the video events; anomalous events are detected using distribution-based hidden and coupled hidden Markov models. Based on the interaction forces of individuals in a group, Mehran et al. (2009) have suggested representing an event by a social force model (SFM) to capture the dynamics of crowd behavior; a bag of words approach is employed to distinguish anomalous frames from the normal ones. Mahadevan et al. (2010) proposed the use of a set of mixture of dynamic textures models to jointly model the dynamics and appearance in crowded scenes. In Cong et al. (2011) and Zhu et al. (2014), the Multi-scale Histogram of Optical Flow (MHOF) is extracted to represent an event and anomalies are detected based on a sparse reconstruction cost (SRC). Lu et al. (2013) updated the SRC detection model to contain a sparse combination of learning and 3D gradient features to represent an event. Similarly, Giorno et al. (2016) proposed detecting changes in video clips by finding frames that can be distinguished from previous frames. As an extension of the Bag of Video words (BOV) approach, Roshtkhari and Levine (2013) introduced a probability density function to encode spatio-temporal configurations of video volumes based on spatio-temporal gradient features. By combining statistical feature such as HOG and HOF together to represent the events, Yuan et al. (2016) detected anomalous events based on a statistical hypothesis test. By including velocity and entropy information to HOF, Colque et al. (2017) proposed a new spatiotemporal feature descriptor, called Histograms of Optical Flow Orientation and Magnitude and Entropy (HOFME). In general however, although local cuboid-based methods are robust when dealing with complex scenes, one disadvantage of these methods is that it may fail to detect the long term activities such as loitering. That is because loitering is related to global movement of a person in the long term rather than the very local cuboid movements.

2.2. Deep learning methods

Deep learning has recently been used for anomaly detection and has produced state of the art results. The first work that applied deep learning to anomaly detection was Sabokrou et al. (2016a), which was based on a cascade of auto-encoders. It employed the reconstruction error of the auto-encoder as well as a sparseness measurement of a sparse auto-encoder. Ravanbakhsh et al. (2017) used a Fully Convolutional Network as a pre-trained model and inserted an effective binary quantization layer as the final layer of the net to capture temporal CNN patterns. By combining these temporal CNN patterns with a hand-crafted feature (optical flow), they proposed a new measure for detecting local anomalies. Sabokrou et al. (2016b) employed fully convolutional neural networks (FCNs) to extract discriminative features of video regions. They modeled a normal event as a Gaussian distribution and labeled a test region that differed from the normal reference model as anomaly. Recently, Hasan et al. (2016) employed both a fully-connected auto-encoder and a fully-convolutional auto-encoder to learn temporal regularity. However, decisions were based on handcrafted features and short video clips, respectively. A regularity score was computed from the reconstruction errors to detect the anomalies.

The most similar to our paper is the work by Xu et al. (2015), which proposed a three-stream architecture (spatial, temporal and their joint representation) by employing the auto-encoder to learn the features. Following this, a one-class support vector machine was exploited to predict the anomaly scores for each stream. A late fusion strategy was then applied to integrate the three-stream scores to make a final decision. The primary difference from our approach is that we do not need to train one-class SVMs or any other event detection model in addition to the learned visual representations. In fact, our approach is an end-to-end deep learning framework, which learns the feature representations of the normal samples as a Gaussian Mixture Model (GMM) by using deep learning. Moreover, we employ so-called *dynamic flow images* instead of the usual optical flow images to represent the motion information.[3]

2.3. Variational Autoencoder

An Autoencoder (Bengio et al., 2013) learns a ***latent representation*** $\mathbf{z}$ for a set of data $\mathbf{x}$ by aligning the outputs $\tilde{\mathbf{x}}$ of the Autoencoder to be equal to the inputs $\mathbf{x}$. An Autoencoder consists of an encoder and a decoder.

In addition, by assuming that the latent representation $\mathbf{z}$accords with a Gaussian distribution, a Variational Autoencoder (VAE) (Kingma and Welling, 2014) produces a generative model that creates something very similar to the training data. By inheriting the architecture of a traditional Autoencoder, a Variational Autoencoder consists of two neural networks:

(1) Recognition network (encoder network)*: a probabilistic encoder* $g(\bullet;\phi)$, which map input $\mathbf{x}$ to the latent representation $\mathbf{z}$ to approximate the true (but intractable) posterior distribution $p(\mathbf{z}|\mathbf{x})$,

$$\mathbf{z} = g(\mathbf{x};\phi) \tag{1}$$

(2) Generative network (decoder network): *a generative decoder* $f(\bullet;\theta)$, which reconstructs the latent representation $\mathbf{z}$ to the input value $\tilde{\mathbf{x}}$ and does not rely on any particular input $\mathbf{x}$,

$$\tilde{\mathbf{x}} = f(\mathbf{z};\theta) \tag{2}$$

where ϕ and θ denote the parameters of these two networks.

The recognition network $g(\bullet;\phi)$ and generative network $f(\bullet;\theta)$could be represented as $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$, respectively. According to the variational inference theory (Blei et al., 2017), the loss function of the Variational Autoencoder is represented as:

$$\mathscr{L}(\theta,\phi,\mathbf{x}) = E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) \tag{3}$$

The first term of (3) is the expected log-likelihood of the input $\mathbf{x}$, which encourages the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ to reconstruct the input $\mathbf{x}$. It could be considered as the reconstruction loss and incurs a large value for good reconstructions. The second term of (3) is the Kullback–Leibler divergence (Kullback and Leibler, 1951) between the $q_\phi(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$, which are the distribution, we wish to learn with the encoder and the prior distribution of the latent representation $\mathbf{z}$, respectively. It measures the difference between two probability distributions and produces a small value when their similarity is very strong. Using the so-called "reparameterization trick" (Blei et al., 2017), the parameters ϕ and θ can be obtained by optimizing (3) via stochastic gradient variational bayes (Kingma and Welling, 2014).

Not only does that a VAE have the ability to generate a variety of complex data (Kingma and Welling, 2014; Gregor et al., 2015; Walker et al., 2016), it has also been shown to be effective for anomaly detection (Soelch et al., 2016; An and Cho, 2015). This is based on the assumption that the latent representation of normal samples is consistent with a Gaussian distribution. This implies that all training data samples are clustered in feature space and the anomalies are far from this cluster center. In fact, this hypothesis is not rigorous since the normal samples may indeed cluster around more than one centroid. In order to deal with this issue, we define a Gaussian Mixture Fully Convolutional Variational Autoencoder (GMFC-VAE), which is employed to detect anomalies. We assume that the latent representation of the training samples accords with a Mixture-of-Gaussians Model instead of a simple Gaussian distribution.

[3] See Section 3.1 for a discussion of *dynamic flow images*.

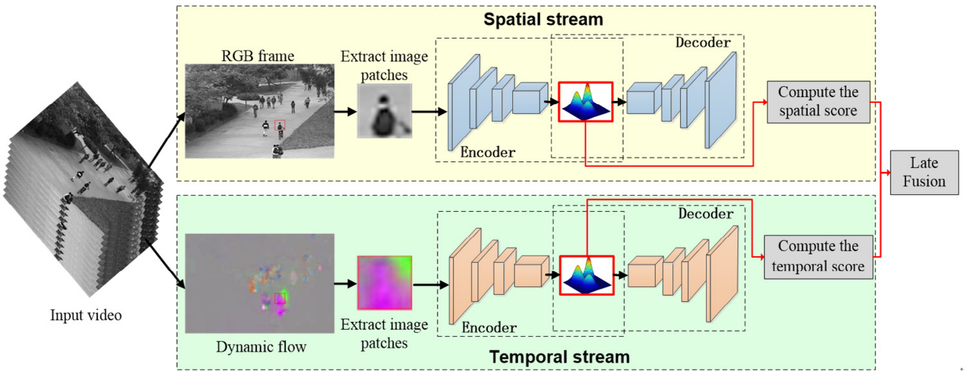


Fig. 1. Overview of the proposed method for anomalous event detection.

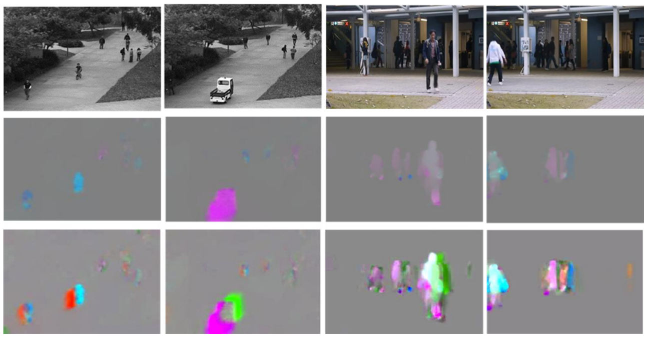


Fig. 2. Examples of dynamic flow image, optical flow and related RGB video frames. The first row shows a single frame from four different videos. The second row indicates the optical flow for the frames in the first row. Finally, the third row presents the dynamic flow for each of the complete videos. Thus the dynamic flow "summarizes" the overall behavior of all of the moving objects in the scene. Also note that the background has been removed.

3. Method

In this section, we present the proposed approach for anomaly detection and localization in detail. Firstly, based on the Ranking SVM formulation, dynamic flows are generated to represent the motion cue. Then, a two-stream Gaussian Mixture Fully Convolutional Variational Autoencoder (GMFC-VAE) is used to learn an anomaly detection model utilizing the normal samples of RGB images and dynamic flows, respectively. Given a test sample corresponding to an image patch and the two learned models, both the appearance anomaly and motion anomaly scores can be predicted by exploiting a sample energy-based method. Finally, these two complementary cues are fused to achieve the final detection results. The overview of the proposed method is illustrated in Fig. 1.

3.1. *Obtaining dynamic flow*

Dynamic flow is an *amalgamation* of a number of sequential frames, based on the optical flow computed for each frame in a video. Compared with the more familiar raw optical flow, which contains only the motion cues between two consecutive frames, dynamic flow is capable of capturing long-term temporal information.

Given a video of length n frames, each of which contain the conventional optical flows $\mathcal{F} = \left\{f_i\right\}_{i=1}^{n}$, where $f_i \in \mathbb{R}^{m_1 \times m_2 \times 2}$, and m_1, m_2 are the height and width of the image. According to Wang et al. (2017), the *horizontal* dynamic flow channel $F^u \in \mathbb{R}^{d_1 \times d_2}$ as well as the *vertical* flow channel $F^v \in \mathbb{R}^{d_1 \times d_2}$ can be approximated by minimizing the upper bound of $\sum \xi_{ij}$, by solving the following problem:

$$\text{Minimize: } L\left(F^u, F^v, \xi_{ij}\right) = \|F^u\|^2 + \|F^v\|^2 + C \sum_{i<j} \xi_{ij}$$

$$\text{Subject to: } \forall i < j$$

$$\left\langle F^u, \overline{f_i^u} \right\rangle + \left\langle F^v, \overline{f_i^v} \right\rangle \leq \left\langle F^u, \overline{f_j^u} \right\rangle + \left\langle F^v, \overline{f_j^v} \right\rangle + 1 - \xi_{ij} \qquad (4)$$

where ξ_{ij} is a slack variable and $\forall (i,j): \xi_{ij} \geq 0$. C is a soft margin parameter that controls the trade-off between margin size and training error. ${f_i}^u$ and ${f_i}^v$ represent the horizontal and vertical components of the optical flow image f_i, respectively. Given that $\bar{\bullet}$ represents an averaging operation, for example, $\overline{f_i^u} = \frac{1}{i}\sum_{t=1}^{i} f_t^u$. We also note that $\langle \bullet, \bullet \rangle$ signifies the inner product of the Time Varying Mean (TVM) flow image and the dynamic flow that are to be found.

Eq. (4) can be solved by training a linear ranking machine, such as RankSVM (Smola and Schölkopf, 2004). We observe that this will facilitates the conversion of a number of sequential optical flow frames to a two channel dynamic flow. For each optical flow frame f_i, we compute the dynamic flow F_i from the sequence of $\left\{f_{i'}\right\}_{i'=i}^{i+\Delta t}$, where Δt is the window size. Thus, a video of length n optical flow frames, $\mathcal{F} = \left\{f_i\right\}_{i=1}^{n}$, can be converted to a set of dynamic flows $\left\{F_i\right\}_{i=1}^{n-\Delta t}$ to represent the motion in the whole video. Thus the motion can represented by the dynamic flow F_i instead of commonly used optical flow f_i.

It can be observed in Fig. 2 that the dynamic flow images indicate an unbelievable ability to characterize events. The motion cues are represented by two regions of similar shape but different colors that indicate the beginning and evolution of the motion. In addition, the intensity of the color denotes the degree of movement. Compared with optical flow, which captures motion cues between two consecutive frames, the objects in the dynamic flow images are more salient and represent long-term temporal information.

3.2. *Learning appearance and motion anomaly detection models using Gaussian mixture fully convolutional variational autoencoders*

In this section, we present how to learn the appearance and motion anomalous detection models with Gaussian Mixture Fully Convolutional Variational Autoencoders (GMFC-VAE). Following the same strategy as Xu et al. (2015), we exploit the appearance cue (RGB frames) and motion cue (dynamic flows) to detect anomalies in both these domains.

We train two separate models for RGB and dynamic flow as inputs. A set of training patches $\mathbf{x} = \{x_i\}_{i=1}^{n}$ (RGB image patches or dynamic flow patches) are obtained by a sliding window from the training set of videos, where n is the number of the training patches. The size of each patch x_i is $d_1 \times d_2 \times c_l$, where d_1, d_2, and c_l represent the width, height and channel, respectively. All patches are linearly normalized into a range of $[0, 1]$ and employed as the input for training the GMFC-VAE.

Variational autoencoder typically assume that the priors of the latent representation $\mathbf{z}$ follow a simple Gaussian distribution. However,

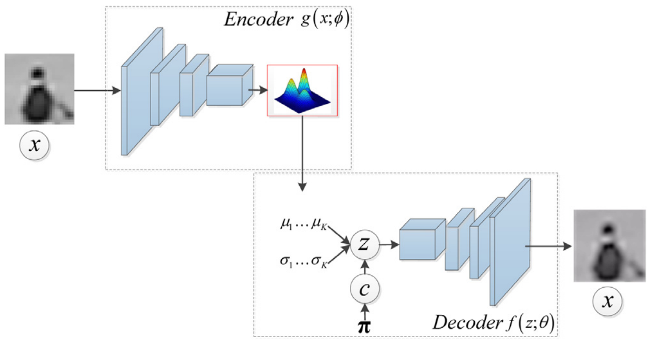


Fig. 3. The diagram of GMFC-VAE.

here we assume that a Mixture-of-Gaussians (MoG) is the prior of the latent representation (Tan et al., 2017). Consequently, the GM-VAE can be applied to describe the distribution of the normal samples. For normal samples $\mathbf{x}$ with latent representation $\mathbf{z}$, as shown in Fig. 3, the generative model can be reformulated as three steps: (1) choose a Gaussian mixture c; (2) obtain a latent vector $\mathbf{z}$; (3) according to the latent representation $\mathbf{z}$ obtain the reconstruction result $\mathbf{x}'$. It can be denoted as follow:

$$c \sim \text{Category}\,(\boldsymbol{\pi}) \tag{5}$$

$$\mathbf{z} \sim \mathcal{N}\left(\mu_c, \sigma_c^2 \mathbf{I}\right) \tag{6}$$

$$p(\mathbf{z}, c) = \pi_c \mathcal{N}\left(\mathbf{z}|\mu_c, \sigma_c^2\right) \tag{7}$$

$$[\mu_x; \log \sigma_x^2] = f\,(\mathbf{z}; \theta) \tag{8}$$

$$\mathbf{x} \sim \mathcal{N}\left(\mu_x, \sigma_x^2 I\right) \tag{9}$$

where K is a predefined number of components of the mixture and $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_K]$ is the prior probability of the Gaussian mixture components, such that $\pi_1 + \pi_2 + \cdots + \pi_K = 1$. The value of K is discussed in Section 4.5. The cth vector component is characterized by normal distributions with means μ_c and covariance σ_c^2. $\mathbf{I}$ is an identity matrix. $f\,(\bullet; \theta)$ is the *decoder* parametrized by θ, $\mathcal{N}\left(\mu_x, \sigma_x^2 I\right)$ is Gaussian distribution parametrized by μ_x and σ_x^2.

Similar to VAE, the *encoder* $g\,(\bullet; \phi)$ is used for approximate true posterior $p\,(\mathbf{z}, c|\mathbf{x})$,

$$[\tilde{\mu}, \widetilde{log\sigma}^2] = g\,(\mathbf{x}; \phi) \tag{10}$$

And the *encoder* $g\,(\bullet; \phi)$ and the decoder $f\,(\bullet; \theta)$ could be represented as $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{z})$, respectively. And from Eqs. (8)(9)(10), we obtain that

$$q\,(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\tilde{\mu}, \tilde{\sigma}^2 I\right) \tag{11}$$

$$p\,(\mathbf{x}|\mathbf{z}) = \mathcal{N}\left(\mathbf{x}|\mu_x, \sigma_x^2 I\right) \tag{12}$$

It follows that the loss function of the GM-VAE can be denoted as:

$$\mathscr{L} = -E_{\mathbf{z}\sim q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] + D_{KL}(q(\mathbf{z}, c|\mathbf{x}) \parallel p(\mathbf{z}, c)) \tag{13}$$

The first term in (13) can be regarded as a reconstruction cost. The second term is given by the Kullback–Leibler divergence between the Mixture-of-Gaussians (MoG) prior $p(\mathbf{z}, c)$ and the variational posterior $q(\mathbf{z}, c|\mathbf{x})$, which can be regarded as the regularization term.

By substituting the terms in Eq. (13) with Eqs. (5), (7), (11) and (12), and using the SGVB estimator, the loss function can be written as:

$$\begin{aligned}\mathscr{L}\left(\theta, \phi, \pi, \mu_c, \sigma_c\right) = & -\frac{1}{L}\sum_{i=1}^{L}\left\|x_i - \tilde{x}_i\right\|_2^2 + \frac{1}{2}\sum_{c=1}^{K}\gamma_c\left(log\sigma_c^2 + \frac{\tilde{\sigma}^2}{\sigma_c^2}\right. \\ & \left. + \frac{\left(\tilde{\mu} - \mu_c\right)^2}{\sigma_c^2}\right) \\ & -\sum_{c=1}^{K}\gamma_c \log\frac{\pi_c}{\gamma_c} - \frac{1}{2}\left(1 + \widetilde{log\sigma}^2\right)\end{aligned} \tag{14}$$

where ϕ and θ are the parameters of the encoder and decoder, L is the number of Monte Carlo samples in the SGVB estimator, x_i is the ith training patch, $\tilde{x}_i$ is the construct result of x_i, K is a predefined number of components of the mixture, π_c is the prior probability of the cth Gaussian mixture components, and γ_c denotes as $q\,(c|\mathbf{x})$ for simplicity.

From the probability model perspective, Eq. (13) can be rewritten as:

$$\begin{aligned}\mathscr{L} &= -E_{\mathbf{z}\sim q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] + D_{KL}(q(\mathbf{z}, c|\mathbf{x}) \parallel p(\mathbf{z}, c)) \\ &= -\int_{\mathbf{z}}\sum_{c} q\,(c|\mathbf{x})\;\; q\,(\mathbf{z}|\mathbf{x})\left[\log p(\mathbf{x}|\mathbf{z}) - \log q(\mathbf{z}, c|\mathbf{x}) + \log p(\mathbf{z}, c)\right] d\mathbf{z} \\ &= -\int_{\mathbf{z}}\sum_{c} q\,(c|\mathbf{x})\;\; q\,(\mathbf{z}|\mathbf{x})\left[\log\frac{p\,(\mathbf{x}|\mathbf{z})\,p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} + \log\frac{p\,(c|\mathbf{z})}{q(c|\mathbf{x})}\right] d\mathbf{z} \\ &= -\int_{\mathbf{z}} q\,(\mathbf{z}|\mathbf{x}) \log\frac{p\,(\mathbf{x}|\mathbf{z})\,p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} + \int_{\mathbf{z}} q\,(\mathbf{z}|\mathbf{x})\, D_{KL}\left(q\,(c|\mathbf{x}) \parallel p\,(c|\mathbf{z})\right) d\mathbf{z}\end{aligned} \tag{$*$}$$

It could be observed that in Eq. ($*$), the first term has no relationship with c. To minimize $\mathscr{L}$, $D_{KL}\left(q\,(c|\mathbf{x}) \parallel p\,(c|\mathbf{z})\right) \equiv 0$ should be satisfied. As a result, $q\,(c|\mathbf{x})$ try to approximate $p\,(c|\mathbf{z})$. And $q\,(c|\mathbf{x})$ could be computed as follow:

$$q\,(c|\mathbf{x}) = p(c|\mathbf{z}) = \frac{p(c)p(\mathbf{z}|c)}{\sum_{c'=1}^{K} p(c')p(\mathbf{z}|c')} \tag{15}$$

The details for optimizing of training stage can be found in Tan et al. (2017).

It has been observed (Springenberg et al., 2015) that the use of a convolution layer to replace both the max pooling and fully connected layers of standard CNNs outperforms the state of the art on several object recognition datasets. That is because when the network is large enough for the dataset it is being trained on and can learn all necessary invariances just with convolutional layers (Springenberg et al., 2015). Besides, compared with fully connected layer, convolution layer have fewer number of connections and weights. That lead convolution layer relatively cheap in terms of memory and compute power needed. Accordingly, we employ a fully convolutional model for the encoder–decoder network. We refer to this network as the as Gaussian Mixture Fully Convolutional Variational Autoencoder (GMFC-VAE). The details of the network architecture are found in Section 4.3.

3.3. Prediction

In last sub-section, we obtained all the parameters ($\left\{\theta, \phi, \pi, \mu_c, \sigma_c\right\}$, $c \in \{1, \ldots, K\}$) of GMFC-VAE. In this sub-section, we discuss how to detect an anomaly after having trained a GMFC-VAE of both the spatial and temporal streams. For the testing phase with test image patch $\mathbf{y}$ (RGB image patch or dynamic flow patch), the latent representation $\mathbf{z}$ can be achieved by the encoder $q_{\phi}(\mathbf{z}|\mathbf{y})$. The $p(\mathbf{z}|c)$ can be computed according to (6) as

$$p(\mathbf{z}|c) = \frac{1}{\sqrt{2\pi}\sigma_c} \cdot e^{-\frac{(\mathbf{z}-\mu_c)^2}{2\sigma_c{}^2}} \tag{16}$$

If query sample $\mathbf{y}$ is normal, its latent representation $\mathbf{z}$ must be associated with at least one Gaussian component i ($i = 1, 2, 3, \ldots, K$) of the training data, which would produce a relatively high conditional probability $p(\mathbf{z}|i)$. On the other hand, the conditional probability for the other Gaussian components j($j = 1, 2, 3, \ldots, K, j \neq i$) would be relatively low. In contrast, the latent representation of an anomalous query sample would most likely not be associated with any of Gaussian components. This would also engender a low conditional probability $p(\mathbf{z}|c')$ for all of the Gaussian components c'($c' = 1, 2, 3, \ldots, K$).

Motivated by Rougier et al. (2011) and Zong et al. (2018), the anomaly score of the query test sample is computed by an sample energy method in form of the log-likelihood:

$$\begin{aligned} E\ (\mathbf{z}) &= -\log\left(\sum_{c'=1}^{K} p(\mathbf{z}|c')p\left(c'\right)\right) \\ &= -\log\left(\sum_{c'=1}^{K} \pi_c \cdot \frac{1}{\sqrt{2\pi}\sigma_{c'}} \cdot e^{-\frac{(\mathbf{z}-\mu_{c'})^2}{2\sigma_{c'}{}^2}}\right) \end{aligned} \tag{17}$$

It is obvious that the anomalies have the higher scores. Suppose the appearance and motion anomaly scores are labeled $\mathcal{S}_{appearance}$ and $\mathcal{S}_{\text{motion}}$. Then the overall anomaly score, $\mathcal{S}_{overall}$, is their combination with the importance factors, α and β:

$$E_{overall} = \alpha E_{appearance} + \beta E_{\text{motion}} \tag{18}$$

Finally, we identify $\mathbf{y}$ as an anomaly if the following criterion is satisfied:

$$E_{overall} > \theta \tag{19}$$

where θ is a threshold that determines the sensitivity of the anomalous detection method. A discussion of these tests are found in Section 4.5.

4. Experiments

To evaluate both the qualitative and quantitative effectiveness of the proposed algorithm, we made comparisons with state-of-the-art algorithms and performed experiments with two public datasets, the UCSD and Avenue Datasets. In this section, we present the datasets, evaluation criteria, details of the experimental settings and experimental results.

4.1. Datasets

The UCSD dataset contains two subsets, Ped1 and Ped2, which were recorded at two different scenes by a fixed camera. In detail, the Ped1 Dataset contains 34 normal and 36 abnormal video clips of 238×158 pixels and each of the video clips contains 200 frames. As for the Ped2 Dataset, it consists of 16 normal and 14 abnormal video clips of size $320{\times}240$ pixels. The length of each video clip in the UCSD Ped2 Dataset is between 150 to 200 frames. For both the Ped1 and Ped2, the normal events contain pedestrians on the walkways, while the abnormal events include bikes, skaters, small cars, and people walking across a walkway or in the grass that surrounds the Walkway. For the two subsets, frame-level ground-truth is provided in the form of a binary flag per frame. In addition, 10 test clips from Ped1 and 12 from Ped2 are provided with pixel-level ground-truth.

The Avenue Dataset contains 15 normal and 21 abnormal videos clips of size 640×360 pixels, which were recorded in front of school corridors using a fixed camera. Each video clip is approximately 1 to 2 min long (25 frames/s). Object-level ground-truth (labeling anomalies with rectangular regions) is provided for this dataset. The normal events contain pedestrians walking in parallel to the camera plane, while the anomalous events contain people running, throwing objects and loitering.

Table 1 gives detailed information regarding these two datasets.

4.2. Evaluation criteria

To compare with existing methods for anomaly detection, we used two evaluation criteria, frame-level and pixel-level, which currently are widely used in anomaly detection research. The details of the two evaluation criteria are as follow:

(1) *Frame-level criterion*: A detected anomalous frame is true positive if it contains at least one anomalous pixel.

(2) *Pixel-level criterion:* A detected anomalous frame is true positive if there is more than 40% overlap with a ground truth region is detected as an anomaly region. This criterion can be used to evaluate the anomaly localization capability.

The *Receiver Operating Characteristic (ROC)* curve of *True positive rate (TPR)* versus *False positive rate (FPR)* is used to measure the accuracy (Mahadevan et al., 2010), where *TPR* represents the rate of correctly labeled and *FPR* represents the rate of incorrectly labeled frames.

Two evaluation criteria are select as quantitative indexes based on the ROC curves:

(1) *Area Under Curve (AUC)*: Area under the ROC curve.

(2) *Equal Error Rate* (*EER*): The ratio of misclassified frames when the *FPR* equals the miss rate, i.e., the *FPR* at which $FPR = 1 - TPR$.

4.3. Implementation details

(1) *Experimental Setup*

First, all of the frames are resized to 420×280. To construct the dynamic flow, we first compute optical flow for each consecutive pair of frames, according to Brox et al. (2004). Following Soelch et al. (2016), the values of f_i^u and f_i^v are transformed into the discrete range [0,255] by employing $f_i^u = f_i^u \times a + b$ and $f_i^v = f_i^v \times a + b$, with $a = 16$ and $b = 128$. The window size Δt that generates the dynamic flows is set to $\Delta t = 20$. Then f_i^u, f_i^v are stacked to form a two-channel image and input to (4) to generate a single dynamic flow. Then the flow magnitude F^m is computed as the third channel of the dynamic flow using $F^m = \sqrt{(F^u)^2 + (F^v)^2}$. The third magnitude channel is set to zero when feeding to the network so that it gives no effect on learning and inference.

To detect the appearance and motion anomalies, two distinct GMFC-VAEs are placed at each input to GMFC-VAEs. Two sets of frames, one containing the RGB and the other, the dynamic flow image data of normal samples are supplied as input. These image frame samples are divided into small patches of size 28×28 with a stride $d_1 = 7$. We eliminate the massive number of patches that do not contain any moving pixels based on a frame difference. This set is then randomly sampled to provide 960K training patches. In the testing stage, the test patches are generated by use sliding windows with a size of 28 × 28 and a stride of $d_2 = 28$. That implies that a test frame outputs a score map of resolution 15 × 10, thereby splitting each frame into a grid of 150 square samples. We arbitrarily select 0.5 for both α and β in (18).

The Encoder and Decoder are pre-trained utilizing a stacked Auto-Encoder which has the same network architecture as the Encoder and Decoder. The network weights were initialized by "Xavier" initialization (Jia et al., 2014) and optimized using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001, a weight decay of 0:9 for each 20 epochs a hyperparameter β_1 of 0.9, a hyperparameter β_2 of 0.999 and mini-batches of size 100. In addition, the epoch is set as 200. The proposed method is implemented in Python and Keras (Chollet, 2015), which used a Theano backend. The number of mixture components K was initialized as $K = 20$ and then updated according to the discussion in Section 4.5.

(2) *Network* architecture

We note that the architecture of the encoder resembles the convolutional stage of Model C in Springenberg et al. (2015). In detail, the encoder has four convolution layers and the size of the first three convolution kernels is 3 × 3. The first convolutional layer has 32 filters with a stride of 2 and generates 32 feature maps with a resolution of 14 × 14. The second and third convolutional layers contain 64 and 128 filters, respectively, with a stride of 2. The resolution of the output feature maps of the second and the third layers are 7 × 7 and 4 × 4, respectively. This is followed up with a fourth convolution layer, which comprises 256 filters of size 4 × 4. It generates 64 feature maps with a resolution of 1 × 1 and can be converted to a 64-D feature. Each convolutional layer is followed by a ReLU nonlinearity.

The Decoder comprises the reverse architecture of the encoder. Two fully-connected layers are placed in parallel at the end of the encoders and result in the means μ_c and covariance σ_c of each of the cth components, respectively. A new sampling layer follows the

Table 1
Main characteristics of the two public datasets.

Datasets	Number of normal video clip	Number of abnormal video clip	Clip length	Normal events	Abnormal events	Link
UCSD Ped1 dataset	34	36	200 frames	Pedestrians on the walkways	Bikes, skaters, small cars, and people walking across a walkway or in the grass that surrounds the Walkway	http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm
UCSD Ped2 dataset	12	16	150—200 frames			
Avenue dataset	17	55	Approximately 1 to 2 min long	Pedestrians walking in parallel to the camera plane	People running, throwing objects and loitering	http://www.cuhk.edu.hk/leojia/projects/detecrab

Table 2
Specifications of the GMFC-VAE model.

Layer	Input	Kernel size	Kernel number	Stride/pad	Output	Last/next layer
I0	3 × 28 × 28	N/A	N/A	N/A	3 × 28 × 28	N/A
C1	3 × 28 × 28	3 × 3	64	2/0	64 × 14 × 14	I0/C2
C2	64 × 14 × 14	3 × 3	128	2/0	128 × 7 × 7	C1/C3
C3	128 × 7 × 7	3 × 3	256	2/1	256 × 4 × 4	C2/C4
C4	256 × 4 × 4	4 × 4	64	1/0	64 × 1	C3/F5
F5	64 × 1	1 × 1	N/A	1/0	64 × 1	C4/S7
F6	64 × 1	1 × 1	N/A	1/0	64 × 1	C4/S7
S7	64 × 1	N/A	N/A	N/A	64 × 1	F5&F6/D8
D8	64 × 1	4 × 4	256	1/0	256 × 4 × 4	D7/D9
D9	256 × 4 × 4	3 × 3	128	2/1	128 × 7 × 7	D8/D10
D10	128 × 7 × 7	3 × 3	64	2/0	64 × 14 × 14	D9/D11
D11	64 × 14 × 14	3 × 3	3	2/0	3 × 28 × 28	D10/O12
O12	3 × 28 × 28	N/A	N/A	N/A	N/A	O12

I = input layer, C = convolutional layer, F = fully connected layer, S = sampling layer, D = deconvolutional layer, O = output layer.
The Encoder and Decoder consist of I0, C1, C2, C3, C4 and D8, D9, D10, D11, O12, respectively.

two fully-connected layers to compute the latent representation $\mathbf{z}'$, as described in (5) and (6). Finally, the latent representation $\mathbf{z}'$ is input to the decoder to obtain the appropriate reconstructed image patch. The detailed configurations of the whole network architecture are shown in Table 2.

4.4. Experimental results

(1) UCSD dataset

Consider the qualitative behavior of the detection performance of the UCSD Ped1 and Ped2 datasets in Figs. 4 and 5. The corresponding ROC curves for pixel- and frame-level behaviors are displayed by varying the threshold parameter θ. The ROC curve for several methods are provided for comparison, including seven methods that use handcrafted features (Yuan et al., 2016; Mehran et al., 2009; Cong et al., 2011; Lu et al., 2013; Roshtkhari and Levine, 2013; Colque et al., 2017) and four that use deep learning (Xu et al., 2015; Sabokrou et al., 2016a; Hasan et al., 2016; Sabokrou et al., 2016b). The results of these contrast methods are obtained in their respective paper. A quantitative comparison in terms of the Area Under the Curve (AUC) and the Equal Error Rate (EER) are shown in Table 3. From an examination of Table 3, it is quite obvious that the use of deep learning features outperforms employing handcrafted features.

The ROCs of Fig. 4 show that our method is comparable to other methods on the UCSD ped1 dataset. Based on frame-level evaluation, our method achieved 94.9% AUC and 11.3% EER on this dataset. This outperforms all of the methods used for comparison. For the pixel level evaluation, our method achieved 91.4% AUC and 36.3% EER, which is better than the other methods except for Statistical Hypothesis Detector (Yuan et al., 2016). In detail, the Statistical Hypothesis Detector (Yuan et al., 2016) ahead by 1.7% and 5.8% of AUC and EER to our method. Compared with the Statistical Hypothesis Detector (Yuan et al., 2016), as shown in Fig. 4(b), our method achieves a relatively higher True Positive Rate (TPR) at a low False Positive Rate (FPR). This is crucial for a practical detection system. More quantitative results for frame-level evaluation and pixel-level evaluation are shown in the 1st, 2nd and 3rd and 4th columns, respectively, of Table 3.

The ROCs of the UCSD ped2 dataset are presented in Fig. 5 and indicate that the proposed method nearly reaches the best of the state-of-the-art. The right side of Table 3 shows the frame- and pixel-level results for the tested methods. Our frame-level EER is 12.6% whereas the best result of 11% is achieved by Deep-Anomaly (Sabokrou et al., 2016b). As well, the pixel-level EER is 19.2%, which is 4.2% less than the Deep-Anomaly (Sabokrou et al., 2016b) algorithm. However, it should be noted that Deep-Anomaly (Sabokrou et al., 2016b) is a combination of a pre-trained CNN (i.e., AlexNet) and a new convolutional layer. Consequently, it is not trained end-to-end and the parameters of the features extraction and anomaly detection cannot be trained jointly. Results of AUCs show that our method outperforms all the methods (Deep-Anomaly Sabokrou et al. (2016b) algorithm does not provide the AUCs) with respect to both the frame-level and pixel-level measure.

Fig. 6 shows some examples of the detection result on the UCSD dataset, in which detected anomalous events are labeled with red masks. The first row and the second row of Fig. 6 are the results of USCD Ped1 and Ped2, respectively. It is obvious that the proposed method is able to detect different kinds of anomalous events, such as bicycling (Fig. 6(a) (b) (e) (f) (h)), skateboarding (Fig. 6(e) and (h)), cars (Fig. 6(d) and (g)) and wheelchair (Fig. 6(c)).

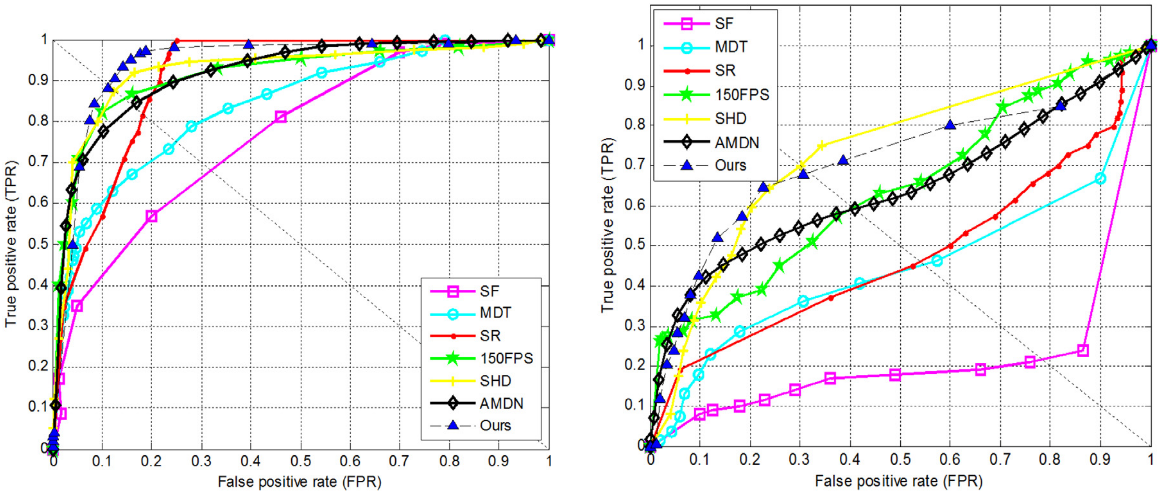
(2) Avenue dataset

Only a few methods have been tested on the Avenue dataset, which is a new dataset that has been made public recently in Lu et al. (2013). Since the anomalies are labeled by rectangle regions but are not actually rectangular, the ground truth contains background as well as foreground pixels. Because of this, we ignore the Pixel-level measure and use only the frame-level measure for testing. Three approaches are presented for comparison: they are the Detection at 150FPS (Lu et al., 2013), Discriminative Framework (Giorno et al., 2016) and Learning Temporal Regularity (Hasan et al., 2016). The results of these methods are obtained from their respective papers. The frame-level evaluation, in the form of AUC and EER, are presented in Table 4 and the ROC curves in Fig. 7.

Compared to the best result of the state-of-art approaches, our method shows an improvement of 2.5% in terms of frame-level AUC. Note that actually only the Learning Temporal Regularity (Hasan et al., 2016) algorithm provides the frame-level EER. However, from Fig. 7,

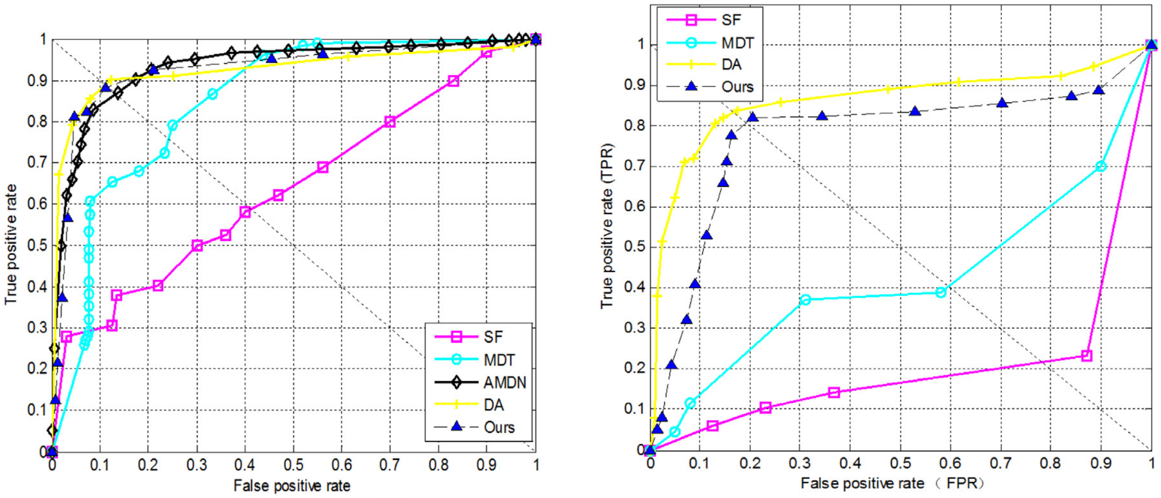
Table 3
Comparison with the state of the art methods in terms of AUC% (Area Under ROC) and EER% (Equal Error Rate) on USCD dataset.

Method	Ped1 (frame level)		Ped1 (pixel level)		Ped2 (frame level)		Ped2 (pixel level)	
	EER	AUC	EER	AUC	EER	AUC	EER	AUC
Social Force (SF) (Mehran et al., 2009)	31	67.5	67.5	19.7	42	55.6	80	–
MDT (Mahadevan et al., 2010)	25	81.8	58	44.1	25	82.9	54	–
Sparse Reconstruction (Cong et al., 2011)	19	–	54	45.3	–	–	–	–
Detection at 150FPS (Lu et al., 2013)	15	91.8	43	63.8	–	–	–	–
Dense STV (Roshtkhari and Levine, 2013)	16.0	89.9	57.7	41.7	–	–	–	–
Statistical Hypothesis Detector (Yuan et al., 2016)	12.1	93.7	**30.5**	**73.1**	–	–	–	–
HOFME (Colque et al., 2017)	33.1	72.7	–	–	20	87.5	–	–
Cascade Auto-encoders (Sabokrou et al., 2016a)	–	–	–	–	15	–	–	–
Deep-Anomaly (Sabokrou et al., 2016b)	–	–	–	–	**11**	–	**15**	–
Learning Temporal Regularity (Hasan et al., 2016)	27.9	81.0	–	–	21.7	90.0	–	–
AMDN (double fusion) (Xu et al., 2015)	16	92.1	40.1	67.2	17	90.8	–	–
Our method	**11.3**	**94.9**	36.3	71.4	12.6	**92.2**	19.2	**78.2**



(a) Frame-level ROC (b) Pixel-level ROC.

Fig. 4. ROC curves for the UCSD Ped1 dataset. Abbreviation: Social Force (SF) (Mehran et al., 2009), MDT (Mahadevan et al., 2010), Sparse Reconstruction (SR) (Cong et al., 2011), Detection at 150FPS (150FPS) (Lu et al., 2013), Statistical Hypothesis Detector (SHD) (Yuan et al., 2016), AMDN (Xu et al., 2015).



(a) Frame-level ROC (b) Pixel-level ROC.

Fig. 5. ROC curves for the UCSD Ped2 dataset. Abbreviation: Social Force (SF) (Mehran et al., 2009), MDT (Mahadevan et al., 2010), AMDN (Xu et al., 2015), Deep-Anomaly (DA) (Sabokrou et al., 2016b).

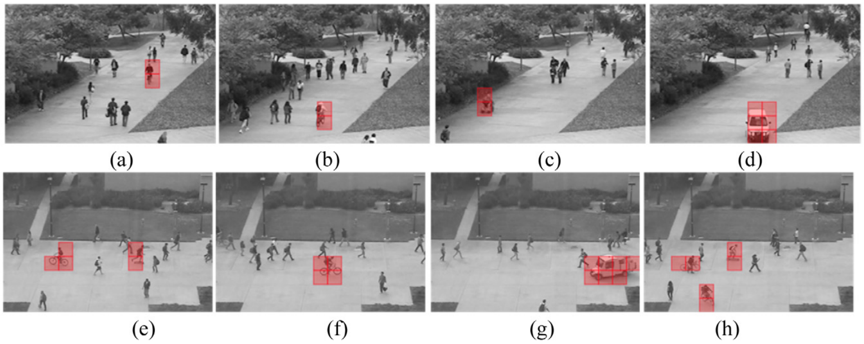


Fig. 6. Examples of abnormality detection results on the UCSD dataset.

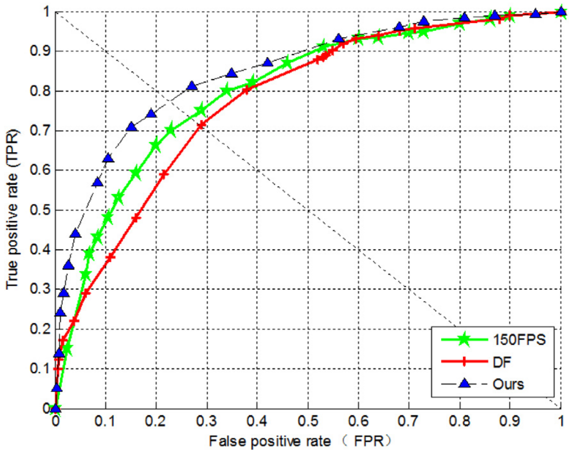


Fig. 7. Frame-level ROC curves for the Avenue dataset. Abbreviation: Detection at 150FPS (150FPS) (Sabokrou et al., 2016a), Discriminative Framework (DF) (Giorno et al., 2016).

Table 4
Performance on the Avenue dataset (AUC% and EER% at the frame-level).

Method	EER	AUC
Detection at 150FPS (Lu et al., 2013)	–	80.9
Discriminative framework (Giorno et al., 2016)	–	78.3
Learning temporal regularity (Hasan et al., 2016)	25.1	70.2
Our method	**22.7**	**83.4**

it can be observed that our method achieves a lower EER than the Detection at 150FPS (Cong et al., 2011) algorithm and the Discriminative Framework (Giorno et al., 2016) algorithm. Overall, the results prove that our method is very effective on the Avenue dataset. Fig. 8 shows some examples of the detected anomalous events, such as waving hands (Fig. 8(a)), throwing papers (Fig. 8(b)), blocking the camera (Fig. 8(c)) and running (Fig. 8(d)).

4.5. Analysis

(1) Evaluation of the number of mixture components

In this sub-section, we analyze the impact of the number of mixture components K[4] on the detection results. Recall that K is pre-defined and decides the number of normal patch clusters. In the experiments, various K values were applied and the Area Under the ROC Curve (AUC) at the frame-level was calculated.

Table 5 presents the detection performance on the UCSD datasets. We observe that performance on both Ped1 and Ped2 increases with K when it is less than 20. This is due to the fact that small values of K produce less clusters that causes inadvertent clusters of normal patches. Some samples that belong to different categories are grouped into one cluster thereby losing some local information. The performance holds steady with increasing number of mixture components when K is beyond 20. According to Eqs. (15) and (17), using larger K requires more computational. Therefore $K = 20$ seems to be a reasonable choice by trading-off algorithm performance against computational cost.

Table 5
Performance (AUC% of frame-level) V.S. K.

K	1	2	5	10	15	20	30	40
UCSD Ped1	68.5	72.3	77.9	84.1	90.8	94.9	94.8	94.9
UCSD Ped2	63.7	65.1	70.4	79.3	86.4	91.7	92.2	92.2

Table 6
Performance under different setting of the UCSD Ped1.

	Frame-level		Pixel-level AUC	
	EER	AUC	EER	AUC
Spatial stream	15.2	90.6	43.3	62.1
Temporal stream	17.1	87.7	46.7	60.8
Late fusion	11.3	94.9	36.3	71.4

(2) Evaluation of spatial and temporal streams

To further demonstrate the validity of our method, we evaluate the performance of GMFC-VAE under two different settings in (18): (1) Spatial Stream: Only the appearance cue is used for detection ($\alpha = 1, \beta = 0$); (2) Temporal Stream: ($\alpha = 0, \beta = 1$). We compare the results with our late fusion results ($\alpha = 0.5, \beta = 0.5$) on UCSD ped1 in Table 6. Clearly, the performance of both the Spatial Stream and Temporal Stream are worse than the late fusion result. That is because either the appearance cue or the motion cue is employed in the two cases.

Some examples of the anomalous detection results of the three settings on the UCSD Ped1 dataset are shown in Fig. 9. The anomalous events include: (a) small car, (b) skater, (c) cyclist and people walking across a walkway or on the grass surrounding it, (d) cyclist. The four columns display the ground-truth, Spatial Stream detection result, Temporal Stream detection result and late fusion result, respectively.

The Temporal Stream is able to detect the appearance anomalies such as the cars, the skater and the cyclist (Fig. 9II-a, II-b, II-d), while missing the skater standing on his skates (Fig. 9II-b) and the cyclist on the bike (Fig. 9II-d). The Temporal Stream also missed the persons walking across a walkway and in the grass surrounding it (Fig. 9II-c).

For the temporal stream, all the missed patches of the spatial stream are identified, such as the missing skater on his skates (Compare Fig. 9II-b and III-b) and the cyclist on the bike (Compare Fig. 9II-d and III-d). However, a big disadvantage of using just the temporal stream, is that compared to the ground truth it produces some false detections (Fig. 9III) as a result of complex motion and occlusion. As shown in

[4] See Eq. (5).

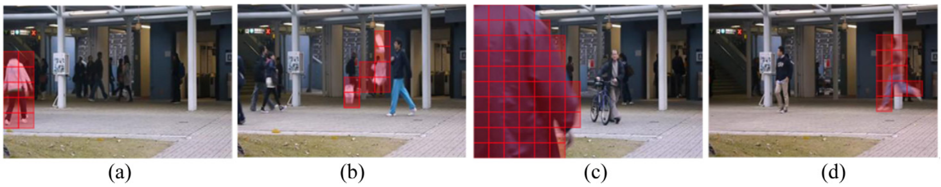


Fig. 8. Examples of abnormality detection results on the Avenue dataset.

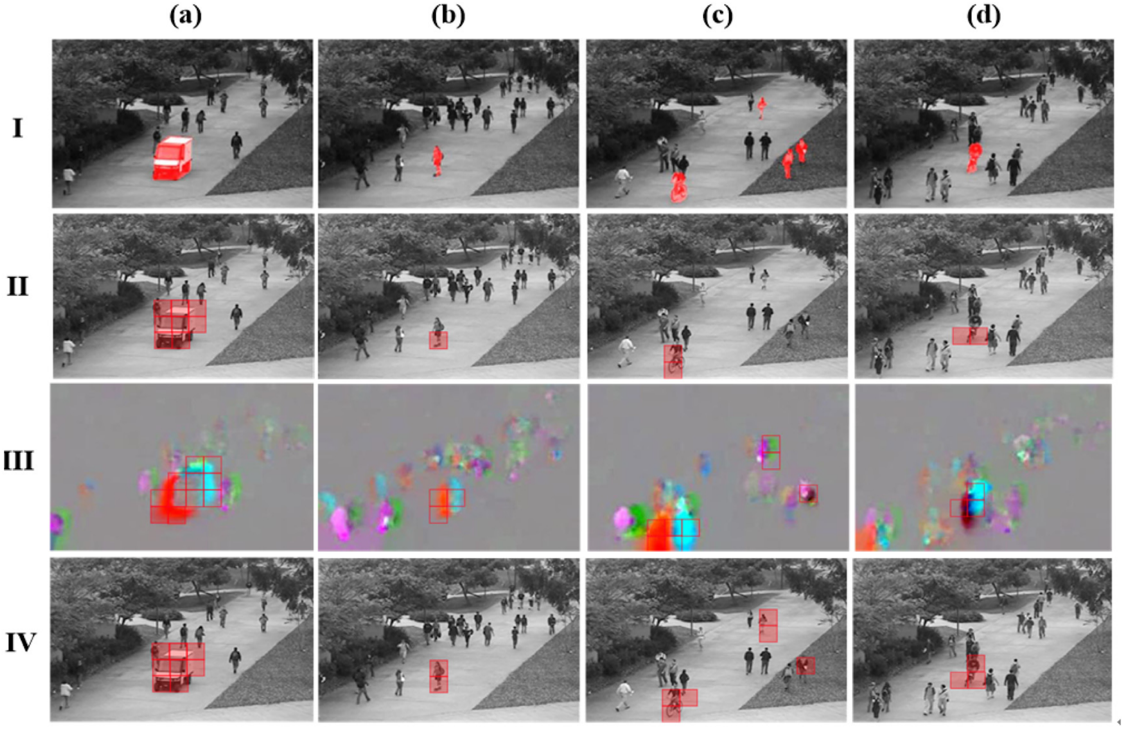


Fig. 9. Examples of the spatial stream and temporal stream detection results on the UCSD Ped1 dataset, in which detected abnormal events are labeled with red masks (I, II, V) or red rectangle (III). (I) Ground-truth. (II) Ours (Appearance) (III) Ours (Motion). (IV) Ours (Late fusion).

Fig. 9IV, combining the spatial and temporal streams (late fusion, as given in (10)) can compromise the misdetection of the spatial stream and false detection of the temporal stream. By combining motion and appearance cues, the detection accuracy can be greatly improved.

(3) Evaluation of main components.

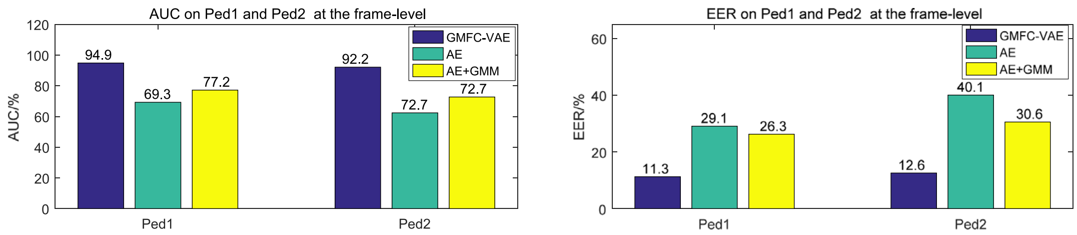
In this sub-section, we compare our method with two baseline models to analyze the importance of main components of our GMFC-VAE. For the first baseline model, to demonstrate the importance of the regularization term in (13) (the second term in (13)), we replace the GMFC-VAE structure with a Stacked Autoencoder (SAE) with the similar architecture that ignore the reparameterization trick. Similar as in Ribeiro et al. (2018), the reconstruction error is used to identify whether the test image patch is anomaly or not. For both the spatial stream and temporal stream, the reconstruction error is first computed by summing up all the pixel-wise errors between the input patch and the output patch and then the normalization processing is carried out. The anomaly score fusion procedure and the thresholding scheme are same as in (18) and (19). We refer to this model as AE. We also show the performance of using GMM directly on the latent representations from the learned AE of the first model (we call it AE+GMM as the second baseline model). And the anomaly prediction procedure of AE+GMM is the same with the Section 3.3. Fig. 10 shows the results of our comparison with the two baseline models on the UCSD datasets.

From Fig. 10, it can be observed that our method outperform the other two baseline models, with the highest frame-level AUC and the lowest frame-level EER both on the Ped1 and Ped 2. The experiment result significantly confirms the importance of the regularization term and the advantage of jointly optimizing the parameters of AE and GMM by end-to-end learning. Besides, the result of AE+GMM is better than the result of AE, which indicate that the GMM contributed to the growth of detection performance. This in line with what observed in previous works on anomaly detection (Sabokrou et al., 2016a, 2018).

5. Conclusion

In this paper, we presented an effective partially supervised deep learning methodology for detecting and locating anomalous events in surveillance videos. Our approach builds upon a two-stream network framework, which employs RGB frames and dynamic flows, respectively. In the training stage, image patches of normal samples for each stream are extracted as input to train a Gaussian Mixture Fully Convolutional Variational Autoencoder (GMFC-VAE) that learns a Gaussian Mixture Model (GMM). In the testing stage, the conditional probabilities of each component of a Gaussian Mixture of test patches are obtained by employing the GMFC-VAE for each stream. We introduce a sample energy based method for predicting an appearance and motion anomaly score. These two cues are then fused to achieve the final detection results. Both the qualitative and quantitative results on two challenging datasets show that our method outperforms the state-of-the-art methods.

However, our work is validated only in the public video sequences record by a fixed camera. In more complex conditions like the changing



(a) AUC on Ped1 and Ped2 at the frame-level (b) EER on Ped1 and Ped2 at the frame-level

Fig. 10. Performance comparison of GMFC-VAE, AE and AE+GMM.

scenes, it takes difficulties to learn the proper estimation of the distribution parameters because of lack of massive similar normal events. One possible solution is to utilize the new observed the normal events to constantly optimize the detection model to avoid variances to collapse for some space regions with poor representation of normal samples. Another direction is to train the detection model consistently through Reinforcement Learning (RL) by getting rewards from detection result.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

An, J., Cho, S., 2015. Variational autoencoder based anomaly detection using reconstruction probability.

Anjum, N., Cavallaro, A., 2008. Multifeature object trajectory clustering for video analysis. IEEE Trans. Circuits Syst. Video Technol. 18 (11), 1555–1564.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8).

Bera, A., Kim, S., Manocha, D., 2016. Real-time anomaly detection using trajectory-level crowd behavior learning. In Proceedings - 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2016, pp. 1289–1296.

Blei, D., Kucukelbir, A., McAuliffe, J., 2017. Variational inference: A review for statisticians. J. Amer. Statist. Assoc. 112 (518), 859–877.

Brox, T., Bruhn, A., Papenberg, N., Weickert, J., 2004. High accuracy optical flow estimation based on a theory for warping. In: ECCV.

Chen, Z., Huang, X., 2017. End-to-end learning for lane keeping of self-driving cars. In: IEEE Intelligent Vehicles Symposium.

Chen, C., Shao, Y., Bi, X., 2015. Detection of anomalous crowd behavior based on the acceleration feature. IEEE Sens. J. 15 (12), 7252–7261.

Chollet, F., 2015. Keras. https://github.com/fchollet/keras.

Colque, R.M., Caetano, C., Toledo, M., Schwartz, W.R., 2017. Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. IEEE Trans. Circuits Syst. Video Technol. 27 (3), 673–682.

Cong, Y., Yuan, J., Liu, J., 2011. Sparse reconstruction cost for anomalous event detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3449–3456.

Dilokthanakul, N., Mediano, P.A., Garnelo, M., Lee, M.C., Salimbeni, H., Arulkumaran, K., Shanahan, M., 2017. Deep unsupervised clustering with gaussian mixture variational autoencoders. In: ICLR.

Feng, Y., Yuan, Y., Lu, X., 2016. Deep representation for abnormal event detection in crowded scenes. In: Proceedings of the 24th ACM International Conference on Multimedia.

Giorno, A.D., Bagnell, J.A., Hebert, M., 2016. A discriminative framework for anomaly detection in large videos. In: ECCV.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR.

Goodale, M.A., Milner, A.D., 1992. Separate visual pathways for perception and action. Trends Neurosci. 15, 20–25.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, Rezende, Danilo, Wierstra, Daan, 2015. Draw: A recurrent neural network for image generation. In: ICCV.

Hasan, M., Choi, J., Neumanny, J., Roy-Chowdhury, A.K., Davis, L.S., 2016. Learning temporal regularity in video sequences. In: CVPR.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093.

Jiang, F., Yuan, J., Tsaftaris, S.A., Katsaggelos, A.K., 2011. Anomalous video event detection using spatiotemporal context. Comput. Vis. Image Underst. 115 (3), 323–333.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR).

Kingma, Diederik P., Welling, Max, 2014. Auto-encoding variational Bayes. In: ICLR.

Kratz, L., Nishino, K., 2009. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1446–1453.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: IEEE Conference on Neural Information Processing Systems (NIPS).

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Ann. Math. Stat. 22 (1), 79–86.

Lu, C., Shi, J., Jia, J., 2013. Anomalous event detection at 150 FPS in MATLAB. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 2720–2727.

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

Mehran, R., Oyama, A., Shah, M., 2009. Anomalous crowd behavior detection using social force model. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 935–942.

Ouivirach, K., Gharti, S., Dailey, M.N., 2013. Incremental behavior modeling and suspicious activity detection. Pattern Recognit. 46 (3), 671–680.

Popoola, O.P., Wang, K., 2012. Video-based anomalous human behavior recognition—A review. IEEE Trans. Syst. Man Cybern. 42 (6), 865–877.

Ravanbakhsh, M., Nabi, M., Mousavi, H., Sangineto, E., Sebe, N., 2017. Plug-and-play CNN for crowd motion analysis: An application in anomalous event detection. In: WACV.

Ribeiro, M., Lazzaretti, A.E., Lopes, H.S., 2018. A study of deep convolutional auto-encoders for anomaly detection in videos. 105, 13–22.

Roshtkhari, M.J., Levine, M.D., 2013. Online dominant and anomalous behavior detection in videos. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 2611–2618.

Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J., 2011. Robust video surveillance for fall detection based on human shape deformation. IEEE Trans. Circuits Syst. Video Technol. 21 (5), 611–622.

Sabokrou, M., Fathy, M., Hoseini, M., 2016a. Video anomaly detection and localization based on the sparsity and reconstruction error of auto-encoder. Electron. Lett. 52 (13), 1122–1124.

Sabokrou, Mohammad, Fayyaz, Mohsen, Fathy, Mahmood, Klette, Reinhard, 2016b. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. arXiv preprint arXiv:1609.00866.

Sabokrou, M., Fayyaz, M., Fathya, M., Moayedc, Z., Klettec, R., 2018. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. Comput. Vis. Image Underst. (172), 88–97.

Shu, R., Gaussian Mixture VAE: Lessons in variational inference: generative models, and deep nets. http://ruishu.io/2016/12/25/gmvae/.

Simonyan, K., Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. In: IEEE Conference on Neural Information Processing Systems (NIPS).

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR).

Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14, 199–222.

Soelch, M., Bayer, J., Ludersdorfer, M., Smagt, P., 2016. Variational inference for on-line anomaly detection in high-dimensional time series. In: Proceedings of the 33rd International Conference on Machine Learning Workshop.

Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M., 2015. Striving for simplicity: The all convolutional net. In: ICML.

Tan, H., Tang, B., Zhou, H., 2017. Variational deep embedding: a generative approach to clustering Y Zheng. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 1965–1972.

Walker, Jacob, Doersch, Carl, Gupta, Abhinav, Hebert, Martial, 2016. An uncertain future: Forecasting from static images using variational autoencoders. In: ECCV.

Wang, J., Cherian, A., Porikli, F., 2017. Ordered pooling of optical flow sequences for action recognition. In: WACV.

Wang, X., Farhadi, A., Gupta, A., 2016. Actions ~ transformations. In: CVPR.

Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N., 2015. Learning deep representations of appearance and motion for anomalous event detection. In: BMVC. pp. 1–12.

Yuan, Y., Feng, Y., Lu, X., 2016. Statistical hypothesis detector for anomalous event detection in crowded scenes. IEEE Trans. Cybern. 99, 1–12.

Zhu, X., Liu, J., Wang, J., Li, C., Lu, H., 2014. Sparse representation for robust anomalousity detection in crowded scenes. Pattern Recognit. 47, 1791–1799.

Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H., 2018. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations.