Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes

Article *in* Computer Vision and Image Understanding · February 2018

DOI: 10.1016/j.cviu.2018.02.006

5 authors, including:

Mohammad Sabokrou
Institute for Research in Fundamental Sciences (IPM)
29 PUBLICATIONS **210** CITATIONS
SEE PROFILE

Mohsen Fayyaz
University of Bonn
17 PUBLICATIONS **113** CITATIONS
SEE PROFILE

Mahmood Fathy
Iran University of Science and Technology
341 PUBLICATIONS **2,931** CITATIONS
SEE PROFILE

Zahra Moayed
Auckland University of Technology
5 PUBLICATIONS **30** CITATIONS
SEE PROFILE

Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes

M. Sabokrou[a,*], M. Fayyaz[b,*], M. Fathy[c], Z. Moayed[d], R. Klette[d]

[a] *School of Computer Science, Institute for Research in Fundamental Sciences (IPM) PO Box 19395-5746, Tehran, Iran*
[b] *University of Bonn, Bonn, Germany*
[c] *Iran University of Science and Technology, Tehran, Iran*
[d] *School of Engineering, Computer and Mathematical Sciences, EEE Department Auckland University of Technology, Auckland, New Zealand*

Abstract

The detection of abnormal behaviour in crowded scenes has to deal with many challenges. This paper presents an efficient method for detection and localization of anomalies in videos. Using *fully convolutional neural networks* (FCNs) and temporal data, a pre-trained supervised FCN is transferred into an unsupervised FCN ensuring the detection of (global) anomalies in scenes. High performance in terms of speed and accuracy is achieved by investigating the cascaded detection as a result of reducing computation complexities. This FCN-based architecture addresses two main tasks, feature representation and cascaded outlier detection. Experimental results on two benchmarks suggest that the proposed method outperforms existing methods in terms of accuracy regarding detection and localization.

Keywords: Video anomaly detection, CNN, transfer learning, real-time processing

1. Introduction

The use of surveillance cameras requires that computer vision technologies need to be involved in the analysis of very large volumes of video data. The detection of anomalies in captured scenes is one of the applications in this area.

*Equal Contribution

Email addresses: sabokro@ipm.ir (M. Sabokrou), fayyaz@iai.uni-bonn.de (M. Fayyaz), mahfathy@iust.ac.ir (M. Fathy), zmoayed@aut.ac.nz (Z. Moayed), rklette@aut.ac.nz (R. Klette)

Anomaly detection and localization is a challenging task in video analysis already due to the fact that the definition of "anomaly" is subjective, or context-dependent. In general, an event is considered to identify an "anomaly" when it occurs rarely, or unexpected; for example, see [1].

Compared to the previously published deep-cascade method in [1], this paper proposes and evaluates a different and new method for anomaly detection. Here we introduce and study a modified pre-trained *convolutional neural network* (CNN) for detecting and localizing anomalies. In difference to [1], the considered CNN is not trained from scratch but "just" fine-tuned. More in detail, for processing a video frame, [1] outlined a method where the frame was first divided into a set of patches, then the anomaly detection was organised based on levels of patches. In difference to that, the input of the proposed CNN algorithm is a full video frame in this paper. As a brief preview, the new method is methodically simpler but faster in both the training and testing phase where the accuracy of anomaly detection is comparable to the accuracy of the method presented in [1].

In the context of crowd scene videos, anomalies are formed by rare shapes or rare motions. Due to the fact that looking for unknown shapes or motions is a time-consuming task, state-of-the-art approaches learn regions or patches of normal frames as reference models. Indeed, these reference models include normal motion or shapes of every region of the training data. In the testing phase, those regions which differ from the *normal model* are considered to be abnormal. Classifying these regions into normal and abnormal requires extensive sets of training samples in order to describe the properties of each region efficiently.

There are numerous ways to describe region properties. Trajectory-based methods have been used to define behaviours of objects. Recently, for modeling spatio-temporal properties of video data, low-level features such as the *histogram of gradients* (HoG) or the *histogram of optic flows* (HoF) are used. These trajectory-based methods have two main disadvantages. They cannot handle occlusion problems, and they also suffer from high complexity, especially in crowded scenes.

CNNs proved recently to be useful for defining effective data analysis techniques for various applications. CNN-based approaches outperformed state-of-the-art meth-

ods in different areas including image classification [2], object detection [3], or activity recognition [4]. It is argued that handcrafted features cannot efficiently represent normal videos [5, 6, 7, 8]. In spite of these benefits, CNNs are computationally slow, especially when considering block-wise methods [3, 9]. Thus, dividing a video into a set of patches and representing them by using CNNs, should be followed by a further analysis with taking care about possible ways of speed-ups.

Major problems in anomaly detection using CNNs are as follows:

1. Too slow for patch-based methods; thus, CNN is considered as being a time-consuming procedure.
2. Training a CNN is totally supervised learning; thus, the detection of anomalies in real-world videos suffers from a basic impossibility of training large sets of samples from non-existing classes of anomalies.

Due to these difficulties, there is a recent trend to optimize CNN-based algorithms in order to be applicable in practice. Faster-RCNN [10] takes advantage of convolutional layers to have a feature map of every region in the input data, in order to detect the objects. For semantic segmentation, methods such as [11, 12] use *fully convolutional networks* (FCNs) for traditional CNNs to extract regional features. Making traditional classification CNNs to work as a fully convolutional network and using a regional feature extractor reduces computation costs. In general, as CNNs or FCNs are supervised methods, neither CNNs nor FCNs are capable for solving anomaly detection tasks,

To overcome aforementioned problems, we propose a new FCN-based structure to extract distinctive features of video regions. This new approach includes several initial convolutional layers of a pre-trained CNN using an AlexNet model [2] and an additional convolutional layer. AlexNet, similar to [13], is a pre-trained model proposed for image classification by using ImageNet [14, 15] and the MIT places dataset [16]. Extracted features, by following this approach, are sufficiently discriminative for anomaly detection in video data.

In general, entire frames are fed to the proposed FCN. As a result, features of all regions are extracted efficiently. By analysing the output, anomalies in the video are extracted and localized. The processes of convolution and pooling, in all of the CNN

layers, run concurrently. A standard NVIDIA TITAN GPU processes ≈ 370 *frames per second* (fps) when analyzing (low-resolution) frames of size 320×240. This is considered to be "very fast".

The main contributions of this paper are as follows:

- To the best of our knowledge, this paper is one of the first where FCN is used for anomaly detection.
- We adapt a pre-trained classification CNN to an FCN for generating video regions to describe motion and shape concurrently.
- We propose a new FCN architecture for time-efficient anomaly detection and localization.
- The proposed method performs as well as state-of-the-art methods, but our method outperforms those with respect to time; we ensure real-time for typical applications.
- We achieved a processing speed of 370 fps on a standard GPU; this is about three times faster than the fastest existing method reported so far.

Section 2 provides a brief survey on existing work. We present the proposed method in Section 3 including the overall scheme of our method, and also details for anomaly detection and localization, and for the evaluation of different layers of the CNN for performance optimization. Qualitative and quantitative experiments are described in Section 4. Section 5 concludes.

2. Related Work

Object trajectory estimation is often of interest in cases of anomaly detection; see [19, 23, 24, 25, 26, 27, 28, 29, 30]. An object shows an anomaly if it does not follow learned normal trajectories. This approach usually suffers from many weaknesses, such as disability to efficiently handle occlusions, and being too complex for processing crowded scenes.

To avoid these two weaknesses, it is proposed to use spatio-temporal low level features such as optical flow or gradients. Zhang et al. [31] use a *Markov random field* (MRF) to model the normal patterns of a video with respect to a number of features, such as rarity, unexpectedness, and relevance. Boiman and Irani [32] consider an event as being abnormal if its reconstruction is impossible by using previous observations only. Adam et al. [33] use an exponential distribution for modeling the histograms of optical flow in local regions.

A *mixture of dynamic textures* (MDT) is proposed by Mahadevan et al. [34] for representing a video. In this method, the represented features fit into a Gaussian mixture model. In [35], the MDT is extended and explained in more details. Kim and Grauman [36] exploit a *mixture of probabilistic PCA* (MPPCA) model for representing local optical flow patterns. They also use an MRF for learning the normal patterns.

A method based on motion properties of pixels for behavior modeling is proposed by Benezeth et al. [37]. They described the video by learning a co-occurrence matrix for normal events across space-time. In [38], a Gaussian model is fitted into spatio-temporal gradient features, and a *hidden Markov model* (HMM) is used for detecting the abnormal events.

Mehran et al. [39] introduce *social force* (SF) as an efficient technique for abnormal motion modeling of crowds. Detection of abnormal behavior, using a method based on spatial-temporal oriented energy filtering, is proposed by In [40].

Cong et al. [41] construct an over-complete normal basis set from normal data. A patch is considered to be abnormal if it is impossible reconstructing it with this basis set.

In [42], a scene parsing approach is proposed by Antic et al. All object hypotheses for the foreground of a frame are explained by normal training. Those hypotheses, that cannot be explained by normal training, are considered as showing an anomaly. Saligrama et al. propose in [43] a method based on clustering of test data using optic-flow features. Ullah et al. [44] introduced an approach based on a cut/max-flow algorithm for segmenting crowd motion. If a flow does not follow the regular motion model, it is considered as being an anomaly. Lu et al. [45] propose a fast (140–150 fps) anomaly detection method based on sparse representation.

In [46], an extension of the *bag of video words* (BOV) approach is used by Roshtkhari et al. A context-aware anomaly detection algorithm is proposed in [47] where authors represent a video using motions and the context of the video. In [48], a method for modeling both motion and shape with respect to a descriptor (named "motion context") is proposed; authors consider anomaly detection as a matching problem. Roshkhari et al. [49] introduce a method for learning dominant events of a video by using the construction of a hierarchical codebook. Ullah et al. [50] learn an MLP neural network using trained particles to extract the video behavior. A *Gaussian mixture model* (GMM) is exploited for learning the behavior of particles using extracted features. In addition, in [51], an MLP neural network for extracting corner features from normal training samples is proposed; authors also label the test samples using that MLP.

Ullah et al. [52] extract corner features and analyze them based on their motion properties by an enthalpy model, using a random forest with corner features for detecting abnormal samples. Xu et al. [53] propose a unified anomaly energy function for detecting anomalies based on hierarchical activity-pattern discovery.

Work reported in [5, 7] models normal events based on a set of representative features which are learned on auto-encoders [54]. Authors use a one-class classifier for detecting anomalies as being outliers compared to the target (normal) class. See also the beginning of Section 1 where we briefly review work being reported in [1]; this paper proposes a cascaded classifier which takes advantage of two deep neural networks for anomaly detection. Here, challenging patches are identified at first by using a small deep network; then neighboring patches are passed into another deep network for further classification.

In [55], the *histogram of oriented tracklets* (HOT) is used for video representation and anomaly detection. A new strategy for improving HOT is also introduced in this paper. Yuan et al. [56] propose an informative *structural context descriptor* (SCD) to represent a crowd individually. In this work, a (spatial-temporal) SCD variation of a crowd is analyzed to localize an anomaly region.

An unsupervised deep learning approach is used in [20] for extracting anomalies in crowded scenes. In this approach, shapes and features are extracted using a PCANet [21] from 3D gradients. Then, a deep *Gaussian mixture model* (GMM) is used to

build a model that defines the event patterns. A PCANet is also used in [22]. In this study, authors exploit the *human visual system* (HVS) to define features in the spatial domain. On the other hand, a *multi-scale histogram of optical flow* (MHOF) is used to represent motion features of the video. PCANet is adopted to exploit these spatio-temporal features in order to distinguish abnormal events.

A hierarchical framework for local and global anomaly detection is proposed in [57]. Normal interactions are extracted by finding frequent geometric relationships between sparse interest points; authors model the normal interaction template by Gaussian process regression. Xiao et al. [58] exploit *sparse semi-nonnegative matrix factorization* (SSMF) for learning the local pattern of pixels. Their method learns a probability model by using local patterns of pixels for considering both the spatial and temporal context. Their method is totally unsupervised. Anomalies are detected by the learned model.

In [59], an efficient method is introduced for representing human activities in video data with respect to motion characteristics; the method is named as *motion influence map*. Blocks of a frame that have a low occurrence are labelled as being abnormal.

Li et al. [60] propose an unsupervised framework for detecting anomalies based on learning global activity patterns and local salient behavior patterns via clustering and sparse coding.

3. Proposed Method

This section explains at first the overall outline of the method. Then, a detailed description of the proposed method is given.

3.1. Overall Scheme

Abnormal events in video data are defined in terms of irregular shapes or motion, or possibly a combination of both. As a result of this definition, identifying the shapes and motion is an essential task for anomaly detection and localization. In order to identify the motion properties of events, we need a series of frames. In other words, a single frame does not include motion properties; it only provides shape information of that specific frame.

For analyzing both shape and motion, we consider the pixel-wise *average* of frame I_t and previous frame I_{t-1}, denoted by I'_t (not to be confused with a derivative),

$$I'_t(p) = \frac{I_t(p) + I_{t-1}(p)}{2} \tag{1}$$

where I_t is the t^{th} frame in the video. For detecting anomalies in I_t, we use the sequence $D_t = \langle I'_{t-4}, I'_{t-2}, I'_t \rangle$.

We start with this sequence D_t when representing video frames on grids of decreasing size $w \times h$. D_t is defined on a grid Ω_0 of size $w_0 \times h_0$. The sequence D_t is subsequently passed on to an FCN, defined by the k^{th} intermediate convolutional layer, for $k = 0, 1, \ldots, L$, each defined on a grid Ω_k of size $w_k \times h_k$, where $w_k > w_{k+1}$, and $h_k > h_{k+1}$. We use $L = 3$ for the number of convolutional layers.

The output of the k^{th} intermediate convolutional layer of the FCN are feature vectors $f_k \in \mathbb{R}^{m_k}$ (i.e., each containing m_k real feature values), starting with $m_0 = 1$. For the input sequence D_t, the output of the k^{th} convolutional layer is a matrix of vector values:

$$\left\{ f_k^t(i,j,1:m_k) \right\}_{(i,j)=(1,1)}^{(w_k,h_k)} = \left\{ \left[f_k^t(i,j,1), \ldots, f_k^t(i,j,m_k) \right]^\top \right\}_{(i,j)=(1,1)}^{(w_k,h_k)} \tag{2}$$

Each feature vector $f_k^t(i,j,1:m_k)$ is derived from a specific *receptive field* (i.e., a sub-region of input D_t).

In other words, first, a high-level description of D_t is provided for the t^{th} frame of the video. Second, D_t is represented subsequently by the k^{th} intermediate convolutional layer of the FCN, for $k = 1, \ldots, L$. This representation is used for identifying a set of partially pairwise overlapping regions in Ω_k, called the *receptive fields*. Hence, we represent frame I_t at first by a sequence D_t on Ω_0, and then by m_k maps

$$f_{k,l} = \left\{ f_k^t(i,j,l) \right\}_{(i,j)=(1,1)}^{(w_k,h_k)}, \text{ for } l = 1, 2, \ldots, m_k \tag{3}$$

on Ω_k, for $k = 1, \ldots, L$. Recall that the size $w_k \times h_k$ decreases with increases of k values.

Suppose that we have q training frames from a video which are considered to be normal. To represent these normal frames with respect to the k^{th} convolutional layer of the FCN, we have $w_k \times h_k \times q$ vectors of length m_k, defining our 2D normal region

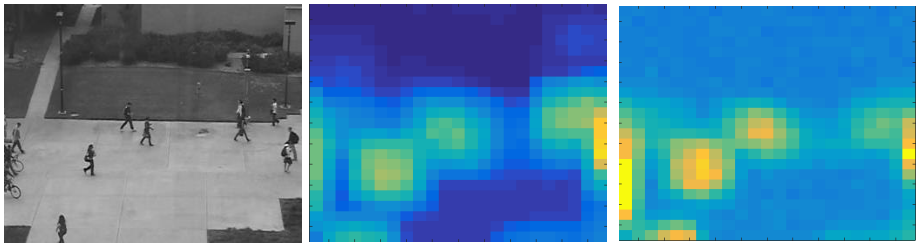

Figure 1: Effect of representing receptive fields with an added convolutional layer. *Left:* Input frame. *Middle:* Heat-map visualisation of the 2^{nd} layer of a pre-trained FCN. *Right:* Heat-map visualisation of the 3^{nd} layer of a pre-trained FCN with added convolutional layer.

descriptions; they are generated automatically by a pre-trained FCN. For modeling the normal behavior, a Gaussian distribution is fitted as a one-class classifier to the descriptions of normal regions so that it defines our *normal reference model*. In the testing phase, a test frame I_t is described in a similar way by a set of regional features. Those regions which differ from the normal reference model are labeled as being *abnormal*. In particular, the features generated by a pre-trained CNN (2^{nd} layer of AlexNet) are sufficiently discriminative. These features are learned based on a set of independent images which are not necessarily related to video surveillance applications only.

Consequently, suspicious regions are represented by a "more discriminant" feature set. This new representation leads to a better performance for distinguishing abnormal regions from normal ones. In other words, we transform the generated features by AlexNet into an anomaly detection problem. This work is done by an auto-encoder which is trained on all normal regions. As a result, those suspicious $f_k^t(i,j,1:m_k)$ regions are passed to an auto-encoder to have a better representation. This is done by the $(k+1)^{st}$ convolutional layer whose kernels are learned by a sparse auto-encoder.

Let $T_k^t(i,j,1:m_k)$ be the transformed representation of $f_k^t(i,j,1:m_k)$ by a sparse auto-encoder; see Figure 1. The abnormal region is visually more distinguishable in the heat-map when the regional descriptors are represented again by the auto-encoder (i.e., the final convolutional layer).

Then, for the new feature space, those regions which differ from the normal ref-

erence model are labeled as being abnormal. This proposed approach ensures both accuracy and speed.

Suppose that $f(i, j, 1 : m_k) \in \mathbb{R}^{m_k}$ is the description of an abnormal region. By moving backward from the k^{th} to the 1^{st} layer of the FCN, we can identify regions in input frames with descriptions $f_k^t(i, j, 1 : m_k)$. See Subsection 3.3 for more details.

Figure 2 shows the work-flow of the proposed detection method. First, input frames are passed on to a pre-trained FCN. Then, $h_k \times w_k$ regional feature vectors are generated in the output of the k^{th} layer. These feature vectors are verified using Gaussian classifier G_1. Those patches, which differ significantly from G_1 as a normal reference model, are labeled as being abnormal. More specifically, G_1 is a Gaussian distribution which is fitted to all of the normal extracted regional feature vectors; regions which completely differ from G_1 are considered to be an anomaly.

Those suspicious regions, which are fitted with low confidence, are given to a sparse auto-encoder. At this stage, we also label these regions based on a Gaussian classifier G_2 which works similar to G_1. G_2 is also a Gaussian classifier, trained on all extracted

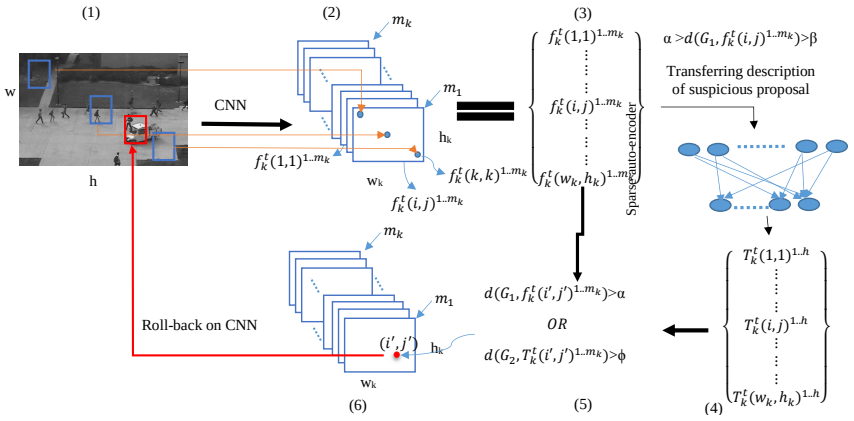

Figure 2: Schematic sketch of the proposed method. (1) Input video frame of size $w_0 \times h_0$. (2) k^{th} layer of the FCN generates m_k grids where every point of these grids $f_k^t(i, j)^{1..m_k}$ describes a region of the input (of the t^{th} frame) of FCN. (3) Descriptions are rewritten as a set of feature vectors; d denotes the Mahalanobis distance. (4) Those feature vectors (region descriptions) that are "sufficiently away" from the reference model are confidently detected as describing an anomaly; the remaining ones are transformed using a sparse auto-encoder that is optimized by training on normal videos. (5) Joint anomaly detector. (6) Location of descriptions that identify anomalies.

regional feature vectors from training video data which are represented by an auto-encoder. Finally, the location of those abnormal regions can be annotated by a roll-back on the FCN.

3.2. Anomaly Detection

In this paper, a video is represented using a set of regional features. These features are extracted densely, and their description is given by feature vectors in the output of the k^{th} convolutional layer.

A Gaussian classifier $G_1(.)$ is fitted to all normal regional features generated by the FCN. Those regional features, for which their distance to $G_1(.)$ is bigger than threshold α, are considered to be abnormal. Those ones that are compatible to G_1 (i.e., their distance is less than threshold β) are labeled as being normal. A region is *suspicious* if it has a distance to G_1 being between α and β.

All suspicious regions are given to the next convolutional layer which is trained on all normal regions generated by the pre-trained FCN. The new representation of these suspicious regions is more discriminative, and denoted by

$$T_{k,n} = \left\{T_k^t(i,j,n)\right\}_{(i,j)=(1,1)}^{(w'_k,h'_k)}, \quad \text{for } n = 1,2,\ldots,h \tag{4}$$

where h is the size of the feature vectors generated by the auto-encoder, which equals the size of the hidden layers.

In this step, only the suspicious regions are processed. Thus, some points (i,j) in grid (w_k,h_k) are ignored and not analysed in the grid (w'_k,h'_k). Similar to G_1, we create a Gaussian classifier G_2 on all of the normal training regional features which are represented by our auto-encoder. Those regions which are not sufficiently fitted to G_2 are considered to be abnormal.

Equations (5) and (6) summarize anomaly detection by using two fitted Gaussian classifiers. First, we have that

$$G_1(f_k^t(i,j,1:m_k)) = \begin{cases} \text{Normal} & \text{if } d(G_1, f_k^t(i,j,1:m_k)) \leq \beta \\ \text{Suspicious} & \text{if } \beta < d(G_1, f_k^t(i,j,1:m_k)) < \alpha \\ \text{Abnormal} & \text{if } d(G_1, f_k^t(i,j,1:m_k)) \geq \alpha \end{cases} \tag{5}$$

Then, for a suspicious region represented by $T_k^t(i, j, 1 : h)$, we have that:

$$G_2(T_k^t(i, j, 1 : h_k)) = \begin{cases} \text{Abnormal} & \text{if } d(G_2, T_k^t(i, j, 1 : h)) \geq \phi \\ \text{Normal} & \text{otherwise} \end{cases} \quad (6)$$

Here, $d(G, \mathbf{x})$ is the Mahalanobis distance of a regional feature vector $\mathbf{x}$ from the G-model.

3.3. Localization

The first convolutional layer has m_1 kernels of size $x_1 \times y_1$. They are convolved on sequence D_t for considering the t^{th} frame. As a result of this convolution, a feature is extracted.

Recall that each region for the input of the FCN is described by a feature vector of length m_1. In this continuous process, we have m_k maps as output for the kth layer. Consequently, a point in the output of the k^{th} layer is a description for a subset of overlapping $(x_1 \times y_1)^{th}$ receptive fields in the input of the FCN. See Figure 3.

The order of layers in the modified version of AlexNet is denoted by

$$\text{AlexNet Order} \rightarrow [C_1, S_1, C_2, S_2, C_3, fc_1, fc_2] \quad (7)$$

where C and S are a convolutional layer and a sub-sampling layer, respectively. The two final layers are fully connected.

Assume that n regional feature vectors $(i_1, j_1) \cdots (i_n, j_n)$, generated in layer C_k on grid Ω_k, are identified as showing an anomaly. The location (i, j) in Ω_k corresponds

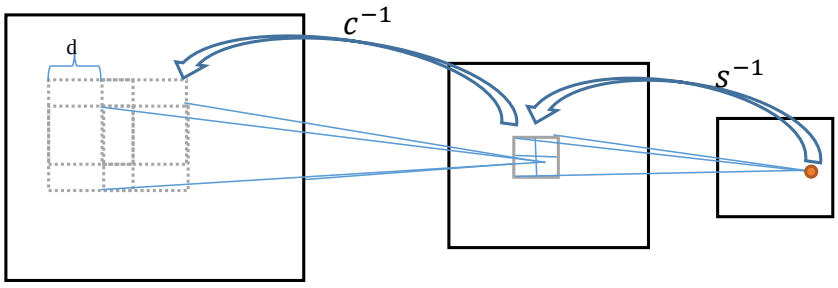

Figure 3: An example of back-ward on CNN to find the related respective filed on input frame.

to

$$C_1^{-1}(\cdots S_{k-1}^{-1}(C_k^{-1}(i,j))) \tag{8}$$

as the rectangular region in the original frame.

Suppose we have m_k kernels of size $x_k \times y_k$ which are then convolved with stride d on the output of the previous layer of C_k. $C_k^{-1}(i,j)$ is the (rectangular) set of all locations in Ω_{k-1} which are mapped in the FCN on (i,j) in Ω_k. Function S_k^{-1} is defined in an analogous way.

The sub-sampling (mean pooling) layer can also be considered as a convolutional layer which has only one kernel. Any region, detected as being an abnormal region in the original frame (i.e., in Ω_0), is then a combination of some overlapping and large patches. This leads to a poor localization performance.

As a case in point, a detection in the 2^{nd} layer causes 51×51 overlapping receptive fields. To achieve more accuracy in anomaly detection, those pixels in Ω_0 are identified to show an anomaly which are covered by more than ζ related receptive fields (we decided for ζ=3 experimentally).

3.4. Investigating Optimum FCN Structure for Anomaly Detection

This section analyses the quality of different layers of a pre-trained CNN for generating regional feature vectors. We adapt (in this paper in general) a classification by CNN into an FCN by solely using convolutional layers. Selecting the best layer for representing the video is crucial considering the following two aspects:

(1) Although deeper features are usually more discriminative, using these deeper features is time-consuming. In addition, since the CNN is trained for image classification, going deeper may create over-fitted features for image classification.

(2) Going deeper leads to larger receptive fields in the input data; as a result, the likelihood of inaccurate localization increases which then has inverse effects on performance.

For the first two convolutional layers of our FCN model, we use a modified version of

AlexNet named *Caffe reference model*.[1]

This model is trained on 1,183 categories, each with 205 scene categories from the MIT places database [16], and 978 object categories from the train data of ILSVRC2012 (ImageNet) [14, 15] having 3.6 million images.

The implemented FCN has three convolutional layers. For finding the best convolutional layer k, we set initially k to 1, and then increase it to 3. When the best k is decided, deeper layers are ignored.

The general findings are described at an abstract level. First we use the output of layer C_1. For distinguishing abnormal from normal regions, corresponding receptive fields are small in size, and generated features are not capable of achieving the suitable results. Therefore, here we have lots of false positives. Later, the output of C_2 is used as a deeper layer. At this stage, we achieve better performance compared to C_1 due to the following reasons: A corresponding receptive field in the input frames of C_1 is now sufficiently large, and the deeper features are more discriminative. At $k = 3$, we have the results in layer C_3 as output. Although the capacity of the network increases, results are not as good as for the 2^{nd} convolutional layer. It seems that by adding one more layer, we achieved deeper features; however, these features are also likely to over-fit the image classification tasks since the network is trained for ImageNet.

Consequently, we decided for the C_2 output for extracting regional features. Similar to [18], we transformed the description of each generated regional feature using a convolutional layer; the kernels of the layer are learned using a sparse auto-encoder. This new layer is called C_T; it is on top of the C_2 layer of the CNN. The combination of three (initial) layers of a pre-trained CNN (i.e., C_1, S_1, and C_2) with an additional (new) convolutional layer is our new architecture for detecting anomalies. Figure 4 shows the proposed FCN structure. To emphasise further the effects of using this structure, see Tables 1 to 3.

Table 1 shows the performance of different layers of the pre-trained CNN.

Table 2 reports the performance of using the proposed architecture with different numbers s of kernels in the $(k + 1)^{th}$ convolutional layer.

[1] Caffe is a framework maintained by UC Berkeley [62].

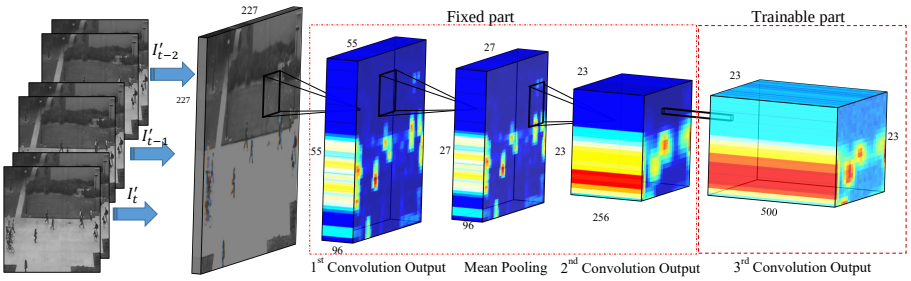

Figure 4: Proposed FCN structure for detecting anomalies. This FCN is only used for regional feature extraction. At later stages, two Gaussian classifiers are embedded for labeling abnormal regions.

Table 1: Evaluating CNN convolutional layers for anomaly detection

Layer	Output in C_1	Output in C_2	Output in C_3
Proposed size	11×11	51×51	67×67
Frame-level EER	40%	13%	20%
Pixel-level EER	47%	19%	25%

We represent video frames with our FCN. A Gaussian classifier is exploited at the final stage of the FCN (see the performance for 100, 256, and 500 kernels in Table 2). We also evaluated the performance when two Gaussian classifiers are used in a similar approach to a cascade. Frame-level and pixel-level EER measures are introduced in the next section. Recall that the smaller the value for EER, the better the performance.

Table 3 reports the performance of processing network outputs in C_2 output and C_T output with cascaded classifiers.

When evaluating different CNNs, our results confirm that the proposed CNN archi-

Table 2: Effect of the number of kernels in the $(k+1)^{th}$ convolutional layer, used for representing regional features when using C_2 as outputs

Number s of kernels	100	256	500	500 & two classifiers
Frame-level EER	19%	17%	15%	**11%**

Table 3: Effect of adding the $(k+1)th$ convolutional layer, used for representing regional features when using C_2 for outputs

Layer	C_2	C_T and two classifiers
Frame-level EER	13%	**11%**

tecture appears to be the best choice for the studied data.

Training the proposed FCN: The optimum FCN includes two parts, a fixed and a trainable part. The fixed part is copied from the Alexnet (as we mentioned before). All training video frames are represented by the fixed part; every frame is represented by 23×23 feature vectors of size 256 each. A sparse auto-encoder is learned using all the feature vectors which are extracted from normal videos. The $1 \times 1 \times 256$ feature vectors are transferred into a sparse vector of size $1 \times 1 \times 500$ (see Fig. 4). Suppose that we have m feature vectors of $(1, 256)$ dimensions while $\mathbf{x_i} \in \mathbb{R}^{D=1\times 256}$ is the raw data for learning an auto-encoder. The auto-encoder attempts to minimize Equ. (9) by reconstructing the raw data

$$\begin{aligned} L = & \frac{1}{m}\sum_{i=1}^{m}\|\mathbf{x_i} - \mathbf{W}_2 \cdot \delta(\mathbf{W}_1\mathbf{x}_i + \mathbf{b}_1) + \mathbf{b}_2\|^2 \\ & + \sum_{i=1}^{D}\sum_{j=1}^{s} W_{ij}^2 \quad + \quad \beta\sum_{j=1}^{s} R(\rho|\rho'_j) \end{aligned} \tag{9}$$

where s is the number of nodes in the hidden layer of the auto-encoder, $\mathbf{W}_1 \in \mathbb{R}^{s\times D}$, and $\mathbf{W}_2 \in \mathbb{R}^{D\times s}$ are a weight matrix and a weight vector, respectively, which map the input layer nodes to hidden layer nodes, and hidden layer nodes to the output layer nodes, respectively.

W_{ij} is the weight between the j^{th} hidden layer node and the i^{th} output layer node, and δ is equal to a sigmoid function applied to all components of an s-dimensional vector.

Furthermore, $\mathbf{b}_1 \in \mathbb{R}^s$ and $\mathbf{b}_2 \in \mathbb{R}^D$ are the bias of the output layer and the hidden layer, respectively; both have a constant number in all of their elements.

$R(\rho|\rho'_j)$ is a regularization function and is set in order to enforce the activation of the hidden layer to be sparse. R is based on similarities between a Bernoulli distribution

with ρ as its parameter and the active node distribution. The parameter β is the weight of the penalty term in the sparse auto-encoder objective.

We can efficiently optimize the above objective with respect to $\mathbf{W}_1$ via the stochastic gradient descent approach. As mentioned previously, size s has an effect on the detection of anomaly regions; see Table 2.

4. Experimental Results

We evaluate the performance of the proposed method on UCSD [61] and Subway benchmarks [33]. We show that our proposed method detects anomalies at high speed, similar to a real-time method in video surveillance, with equal or even better performance than other state-of-the-art methods.

All tests are done based on the optimum FCN which is defined in the previous section (Figure 4). The parameters of β (weight of penalty term), ρ (sparsity), and s (the size of hidden layer) in the objective function of the auto-encoder are set to be 0.1, 0.05, and 500, respectively. This means that feature learning is done with an auto-encoder with 0.05 sparsity.

For implementing our deep-anomaly architecture we use the Caffe library [62]. All experiments are done using a standard *NVIDIA TITAN GPU* with *Caffe* framework.

4.1. UCSD and Subway Datasets

To evaluate and compare our experimental results, we use two datasets.

***UCSD Ped2** [61].* Dominant dynamic objects in this dataset are walkers where crowd density varies from low to high. An appearing object such as a car, skateboarder, wheelchair, or bicycle is considered to create an anomaly. All training frames in this dataset are normal and contain pedestrians only.

This dataset has 12 sequences for testing, and 16 video sequences for training, with 320×240 resolution. For evaluating the localization, the ground truth of all test frames is available. The total numbers of abnormal and normal frames are ≈2,384 and ≈2,566, respectively.

***Subway** [33].* This dataset contains two sequences recorded at the entrance (1 h and 36 min, 144,249 frames) and exit (43 min, 64,900 frames) of a subway station. People entering and exiting the station usually behave normally. Abnormal events are defined by people moving in the wrong direction (i.e., exiting the entrance or entering the exit), or avoiding payment. This dataset has two limitations: The number of anomalies is low, and there are predictable spatial localizations (at entrance or exit regions).

4.2. Evaluation Methodology

We compare our results with state-of-the-art methods using a *receiver operating characteristic* (ROC) curve, the *equal error rate* (EER), and the *area under curve* (AUC). Two measures at frame level and pixel level are used, which are introduced in [34] and often exploited in later work. According to these measures, frames are considered to be abnormal (positive) or normal (negative). These measures are defined as follows:

(1) *Frame-level*: In this measure, if one pixel detects an anomaly then it is considered to be abnormal.

(2) *Pixel-level*: If at least 40 percent of anomaly ground truth pixels are covered by pixels that are detected by the algorithm, then the frame is considered to show an anomaly.

4.3. Qualitative and Quantitative Results

Figure 5 illustrates the output of the proposed system on samples of the UCSD and Subway dataset. The proposed method detects and localizes anomalies correctly in these samples. The main problem of an anomaly detection system is a high rate of false-positives.

Figure 6 shows regions which are wrongly detected as being an anomaly using our method. Actually, false-positives occur in two situations: too crowded scenes, and when people walk in different directions. Since walking in opposite direction of other pedestrians is not observed in the training video, this action is also considered as being abnormal using our algorithm.

(A) (B)

Figure 5: Output of the proposed method on Ped2 UCSD and Subway dataset. *A-left and B-left:* Original frames. *A-Right and B-Right:* Anomaly regions are indicated by red.

Figure 6: Some examples of false-positives in our system. *Left:* A pedestrian walking in opposite direction to other people. *Middle:* A crowded region is wrongly detected as being an anomaly. *Right:* People walking in different directions.

Frame-level and pixel-level ROCs of the proposed method in comparison to state-of-the-art methods are provided in Figure 7; left and middle for frame-level and pixel-level EER on UCSD Ped2 dataset, respectively. The ROCs show that the proposed method outperforms the other considered methods in the UCSD dataset.

Table 4 compares the frame-level and pixel-level EER of our method and other state-of-the-art methods. Our frame-level EER is 11%, where the best result in general is 10%, achieved by Tan Xiao et al. [58]. We outperform all other considered methods except [58]. On the other hand, the pixel-level EER of the proposed approach is 15%, where the next best result is 17%. As a result, our method achieved a better performance than any other state-of-the-art method in the pixel-level EER metric by 2%.

The frame-level ROC of the Subway dataset is shown in Figure 7 (right). In this dataset, we evaluate our method in both the entrance and exit scenes. The ROC confirms that our method has a better performance than MDT [35] and SRC [41] methods.

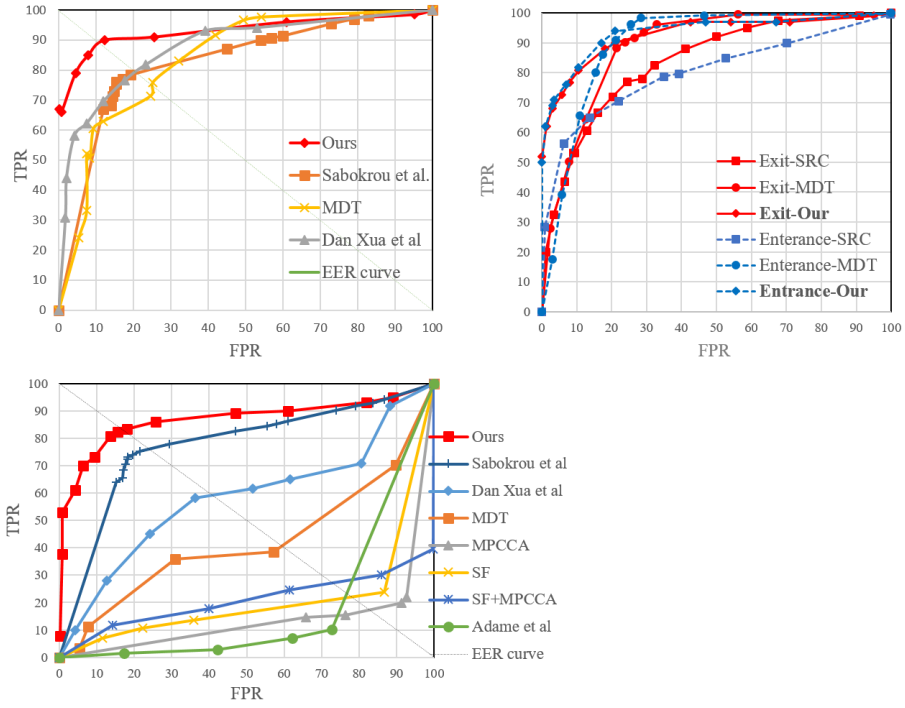

Figure 7: ROC comparison with state-of-the-art methods. *Upper left:* Frame-level of UCSD Ped2. *Bottom left:* Pixel-level of UCSD Ped2. *Upper right:* Subway dataset.

We also discuss the comparison of AUC and EER in this dataset in Table 5.

For the exit scene, we outperform the other considered methods in respect to both AUC and EER measures; we outperform by 0.5% and 0.4% in AUC and EER, respectively. For the entrance scenes, the AUC of the proposed method achieves better results compared to all other methods by 0.4%. The proposed method gains better outcomes in terms of EER for all methods except Saligrama et al. [43]; they achieve better results by 0.3%.

4.4. Run-time Analysis

For processing a frame, three steps need to be performed: Some pre-processing such as resizing the frames and constructing the input of the FCN, and representing the

Table 4: EER for frame and pixel level comparisons on Ped2; we only list first author in this table for reasons of available space

Method	Frame-level	Pixel-level	Method	Frame-level	Pixel-level
IBC [32]	13%	26%	Reddy [63]	20%	—
Adam [33]	42%	76%	Bertini [64]	30%	—
SF [39]	42%	80%	Saligrama [43]	18%	
MPCCA [36]	30%	71%	Dan Xu [7]	20%	42%
MPCCA+SF [34]	36%	72%	Li [35]	18.5%	29.9%
Zaharescu [40]	17%	30%	Tan Xiao [58]	**10%**	17%
MDT [34]	24%	54%	Sabokrou [5]	19%	24%
Ours	11%	**15%**			

Table 5: AUC-EER comparison on Subway dataset

Method	SRC [41]	MDT [34]	Saligrama et al. [43]	Ours
Exit	80.2/26.4	89.7/16.4	88.4/17.9	**90.2/16**
Entrance	83.3/24.4	90.8/16.7	–/–	**90.4**/17

input by the FCN are considered as the first and second step, respectively. In the final step, the regional descriptors must be checked by a Gaussian classifier.

With respect to these three steps, run-time details of our proposed method for processing a single frame are provided in Table 6. The total time for detecting an anomaly in a frame is $\approx$0.0027 sec. Thus, we achieve 370 fps, and this is much faster than any of the other considered state-of-the-art methods.

Table 7 shows the speed of our method in comparison to other approaches (note that these results are obtained on different hardware). There are some key points which

Table 6: Details of run-time (second/frame)

	Pre-processing	Representation	Classifying	Total
Time (in sec)	0.0010	0.0016	0.0001	0.0027

make our system fast. The proposed method benefits from fully convolutional neural networks. These types of networks perform feature extraction and localization concurrently. This property leads to less computations.

Table 7: Run-time comparison on Ped2 (in sec)

Method	IBC [32]	MDT [34]	Roshtkhari et al. [46]	Li et al. [35]	Xiao et al. [58]	Ours
Run-time	66	23	0.18	0.80	0.29	≈**0.0027**

Furthermore, by combining six frames into a three-channel input, we process a cubic patch of video frames at just one forward-pass. As mentioned before, for detecting abnormal regions, we only process two convolutional layers, and for some regions we classify them using a sparse auto-encoder. Processing these shallow layers results in reduced computations. Considering these tricks, besides processing fully convolutional networks in parallel, results in faster processing for our system compared to other methods.

5. Conclusions

This paper presents a new FCN architecture for generating and describing abnormal regions for videos. By using the strength of FCN architecture for patch-wise operations on input data, the generated regional features are context-free. Furthermore, the proposed FCN is a combination of a pre-trained CNN (an AlexNet version) and a new convolutional layer where kernels are trained with respect to the chosen training video. This final convolutional layer of the proposed FCN needs to be trained. The proposed approach outperforms existing methods with respect to processing speed. Besides, it is also a solution for overcoming limitations in training samples used for learning a complete CNN. This method enables us to run a deep learning-based method at a speed of about 370 fps. Altogether, the proposed method is both fast and accurate for anomaly detection in video data.

Acknowledgement

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

References

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

[2] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, Advances Neural Information Processing Systems (2012) 1097–1105.

[3] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Computer Vision Pattern Recognition, 2014, pp. 580–587.

[4] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, CoRR abs/1406.2199.

[5] M. Sabokrou, M. Fathy, M. Hoseini, R. Klette, Real-time anomaly detection and localization in crowded scenes, in: Computer Vision Pattern Recognition Workshops, 2015, pp. 56–62.

[6] M. Sabokrou, M. Fathy, Z. Moayed and R. Klette, Fast and accurate detection and localization of abnormal behavior in crowded scenes in: Machine Vision and Applications,2017(28), pp. 965–985

[7] D. Xu, E. Ricci, Y. Yan, J. Song, N. Sebe, Learning deep representations of appearance and motion for anomalous event detection, CoRR abs/1510.01553.

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

[9] A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, J. Schmidhuber, Fast image scanning with deep max-pooling convolutional neural networks, in: IEEE Int. Conf. Image Processing, 2013, pp. 4034–4038.

[10] S. Ren, K. He, R. B. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Analysis Machine Intelligence (2015), abs/1506.01497.

[11] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Analysis Machine Intelligence (2016), abs/1605.06211.

[12] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Computer Vision Pattern Recognition, 2015.

[13] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, Advances Neural Information Processing Systems (2014) 487–495.

[14] ImageNet (2017). `image-net.org`

[15] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Computer Vision Pattern Recognition, 2009, pp. 248–255.

[16] MIT places database (2017). `places.csail.mit.edu`

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks. in: ICLR, 2014.

[18] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Computer Vision Pattern Recognition, 2014, pp. 1717–1724.

[19] F. Jiang, J. Yuan, S. A. Tsaftaris, A. K. Katsaggelos, Anomalous video event detection using spatiotemporal context, Computer Vision Image Understanding 115 (3) (2011) 323 – 333.

[20] F. Yachuang, Y. Yuan, L. Xiaoqiang, Learning deep event models for crowd anomaly detection, Neurocomputing (2017) 548–556.

[21] C. Tsung-Han, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, PcaNet: A simple deep learning baseline for image classification?, IEEE Trans. Image Processing (2015) 5017–5032.

[22] F. Zhijun, F. Fei, Y. Fang, C. Lee, N. Xiong, L. Shu, S. Chen, Abnormal event detection in crowded scenes based on deep learning, Multimedia Tools Applications (2016) 14617–14639.

[23] S. Wu, B. E. Moore, M. Shah, Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes, in: Computer Vision Pattern Recognition, 2010, pp. 2054–2060.

[24] C. Piciarelli, G. L. Foresti, On-line trajectory clustering for anomalous events detection, Pattern Recognition Letters volume 27 (2006) 1835–1842.

[25] C. Piciarelli, C. Micheloni, G. L. Foresti, Trajectory-based anomalous event detection, IEEE Trans. Circuits Systems Video Technology 18 (11) (2008) 1544–1554.

[26] P. Antonakaki, D. Kosmopoulos, S. J. Perantonis, Detecting abnormal human behaviour using multiple cameras, Signal Processing 89 (9) (2009) 1723 – 1738.

[27] S. Calderara, U. Heinemann, A. Prati, R. Cucchiara, N. Tishby, Detecting anomalies in people's trajectories using spectral graph analysis, Computer Vision Image Understanding 115 (8) (2011) 1099 – 1111.

[28] B. T. Morris, M. M. Trivedi, Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach, IEEE Trans. Pattern Analysis Machine Intelligence 33 (11) (2011) 2287–2301.

[29] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, S. Maybank, A system for learning statistical motion patterns, IEEE Trans. Pattern Analysis Machine Intelligence 28 (9) (2006) 1450–1464.

[30] F. Tung, J. S. Zelek, D. A. Clausi, Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance, Image Vision Computing 29 (4) (2011) 230 – 240.

[31] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, Semi-supervised adapted HMMS for unusual event detection, in: Computer Vision Pattern Recognition, Vol. 1, 2005, pp. 611–618.

[32] O. Boiman, M. Irani, Detecting irregularities in images and in video, Int. J. Computer Vision 74 (1) (2007) 17–31.

[33] A. Adam, E. Rivlin, I. Shimshoni, D. Reinitz, Robust real-time unusual event detection using multiple fixed-location monitors, IEEE Trans. Pattern Analysis Machine Intelligence 30 (3) (2008) 555–560.

[34] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: Computer Vision Pattern Recognition, 2010, pp. 1975–1981.

[35] W. Li, V. Mahadevan, N. Vasconcelos, Anomaly detection and localization in crowded scenes, in: IEEE Trans. Pattern Analysis Machine Intelligence (2014) 18–32.

[36] J. Kim, K. Grauman, Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates, in: Computer Vision Pattern Recognition, 2009, pp. 2921–2928.

[37] Y. Benezeth, P. M. Jodoin, V. Saligrama, C. Rosenberger, Abnormal events detection based on spatio-temporal co-occurrences, in: Computer Vision Pattern Recognition, 2009, pp. 2458–2465.

[38] L. Kratz, K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, in: Computer Vision Pattern Recognition, 2009, pp. 1446–1453.

[39] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: Computer Vision Pattern Recognition, 2009, pp. 935–942.

[40] A. Zaharescu, R. Wildes, Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing, in: European Conf. Computer Vision, vol. 1, 2010, pp. 563–576.

[41] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: Computer Vision Pattern Recognition, 2011, pp. 3449–3456.

[42] B. Antic, B. Ommer, Video parsing for abnormality detection, in: Int. Conf. Computer Vision, 2011, pp. 2415–2422.

[43] V. Saligrama, Z. Chen, Video anomaly detection based on local statistical aggregates, in: Computer Vision Pattern Recognition, 2012, pp. 2112–2119.

[44] H. Ullah, N. Conci, Crowd motion segmentation and anomaly detection via multi-label optimization, in: ICPR Workshop Pattern Recognition Crowd Analysis, 2012.

[45] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 fps in Matlab, in: Int. Conf. Computer Vision, 2013, pp. 2720–2727.

[46] M. Javan Roshtkhari, M. D. Levine An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions, in: Computer Vision Image Understanding 117 (10) (2013) 1436 – 1452.

[47] Y. Zhu, N. M. Nayak, A. K. Roy-Chowdhury, Context-aware modeling and recognition of activities in video, in: Computer Vision Pattern Recognition, 2013, pp. 2491–2498.

[48] Y. Cong, J. Yuan, Y. Tang, Video anomaly search in crowded scenes via spatio-temporal motion context, IEEE Trans. Information Forensics Security 8 (10) (2013) 1590–1599.

[49] M. J. Roshtkhari, M. D. Levine, Online dominant and anomalous behavior detection in videos, in: Computer Vision Pattern Recognition, 2013, pp. 2611–2618.

[50] H. Ullah, L. Tenuti, N. Conci, Gaussian mixtures for anomaly detection, in crowded scenes, in: SPIE Electronic Imaging, 2013, pp. 866303–866303.

[51] H. Ullah, M. Ullah, N. Conci, Real-time anomaly detection in dense crowded scenes, in: in: SPIE Electronic Imaging, 2014, pp. 902608–902608.

[52] H. Ullah, M. Ullah, N. Conci, Dominant motion analysis in regular and irregular crowd scenes, in: International Workshop on Human Behavior Understanding 2014, pp. 62–72.

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

[54] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Int. Conf. Machine Learning, 2008, pp. 1096–1103.

[55] H. Mousavi, M. Nabi, H. K. Galoogahi, A. Perina, V. Murino, Abnormality detection with improved histogram of oriented tracklets in: Int. Conf. Image Analysis Processing, 2015, pp. 722–732.

[56] Y. Yuan, J. Fang, Q. Wang, Online anomaly detection in crowd scenes via structure analysis, IEEE Trans. Cybernetics (2015) 548–561.

[57] K. W. Cheng, Y. T. Chen, W. H. Fang, Video anomaly detection and localization using hierarchical feature representation and gaussian process regression, in: Computer Vision Pattern Recognition, 2015, pp. 2909–2917.

[58] T. Xiao, C. Zhang, H. Zha, Learning to detect anomalies in surveillance video, IEEE Signal Processing Letters 22 (9) (2015) 1477–1481.

[59] D. G. Lee, H. I. Suk, S. K. Park, S. W. Lee, Motion influence map for unusual human activity detection and localization in crowded scenes, IEEE Trans. Circuits Systems Video Technology 25 (10) (2015) 1612–1623.

[60] N. Li, X. Wu, D. Xu, H. Guo, W. Feng, Spatio-temporal context analysis within video volumes for anomalous-event detection and localization, Neurocomputing 155 (2015) 309 – 319.

[61] Ucsd anomaly detection dataset (2017). http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm

[62] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM Int. Conf. Multimedia, 2014, pp. 675–678.

[63] V. Reddy, C. Sanderson, B. C. Lovell, Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture, in: Computer Vision Pattern Recognition Workshops, 2011, pp. 55–61.

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