IJCNN 2019. International Joint Conference on Neural Networks. Budapest, Hungary. 14-19 July 2019

Deep Learning and One-class SVM based Anomalous Crowd Detection

Meng Yang
School of Computing and Information Systems
The University of Melbourne
Melbourne, Australia
myang3@student.unimelb.edu.au

Sutharshan Rajasegarar
School of Information Technology
Deakin University
Melbourne, Australia
srajas@deakin.edu.au

Sarah M. Erfani
School of Computing and Information Systems
The University of Melbourne
Melbourne, Australia
sarah.erfani@unimelb.edu.au

Christopher Leckie
School of Computing and Information Systems
The University of Melbourne
Melbourne, Australia
caleckie@unimelb.edu.au

***Abstract*—Anomalous event detection in videos is an important and challenging task. This paper proposes a deep representation approach to the problem, which extracts and represents features in an unsupervised way. This algorithm can detect anomalous activity like standing statically and loitering among a crowd of people. Our proposed framework is a two-channel scheme by using feature channels extracted from the appearance and foreground of the original video. Two hybrid deep learning architectures SDAE-DBN-PSVM (a four-layer Stacked Denoising Auto-encoder with three-layer Deep Belief Nets and Plane-based one class SVM) are implemented for these two channels to learn the high-level feature representation automatically and produce two anomaly scores. Finally, a fusion scheme is proposed for combining anomaly scores and detecting anomalous events. Experimental results on a large real-world dataset (MCG) and two benchmark datasets (UCSD and Subway) demonstrate the effectiveness of this approach. Furthermore, quantitative analyses of the effects of the amount of training data and the illumination conditions of the video on the accuracy of anomaly detection are presented.**

***Index Terms*—anomalous event detection, deep representation, stacked denoising auto-encoder, deep belief nets, video surveillance**

I. Introduction

Abnormal event/behavior monitoring in crowded scenarios is an important and challenging topic for pattern recognition. Anomalous event detection usually follows learning the ordinary crowd movement at first, followed by differentiating the small amount of anomalies from these normal patterns [1].

There are several methods created to solve this task. A very popular category is trajectory based approaches [2]–[4]. Authors encoded the pedestrians' tracks using a hyperspherical clustering based approach, in order to find the abnormal loitering behavior [3]. In [4], the authors extracted features from trajectories based on Multiple-scale Histogram Optical Flow for joint model building. However, the joint sparsity model can only handle linearly connected objects. Another category is motion representation based methods [5]–[8]. In [6], authors used Social Force Model to describe the pedestrian movement pattern. In [5], a model called mixture dynamic textures (MDT) is built based on motion representation for abnormal event detection. Researchers proposed a spatio-temporal graph model called Markov Random Field (MRF), which can distinguish abnormal and normal activities by capturing the combination of optical flow and the Probabilistic Principal Component Analyzers Mixture [8]. However, it is still difficult to handle practical issues like complex shape changes, occlusion and overlapping by using these existing approaches.

Deep learning based methods are also quite popular. In [9], Xu et al. used a Stacked Denoising Auto-encoder (SDAE) based deep learning architecture to capture and learn features in an automatic way. However, it only uses short-term appearance and motion information, which cannot fully characterize activities in videos involving complex contexts. Recently, the authors of [10], [11] used high level deep learning models to solve this task. In [10], fully convolutional neural networks (FCNs) with temporal data, and in [11], Convolutional Neural Networks (CNN) are used for learning features. However, using these high level deep learning models has high computational complexity. Although there are some existing deep learning based approaches for anomaly detection, most of them are aimed at ordinary anomalous object detection. Further, they cannot guarantee high speed, low computational complexity or both.

Therefore, in this paper, we propose a novel multi-task hybrid deep learning framework, in order to not only perform short-term anomalous object detection, but also achieve long-term anomalous motion/behavior detection, as well as guaranteeing high accuracy and efficiency. Our architecture is comprised of a stacked denoising auto-encoder (SDAE), deep belief network (DBN) and plane-based one class SVM (PSVM) (refer to Fig. 1). The DBN-PSVM combination helps achieve higher detection accuracy by means of dimensionality

978-1-7281-2009-6/$31.00 ©2019 IEEE

reduction to obtain a few high level representative features via the DBN before applying the PSVM.

Our main contributions are listed as follows:

- A novel SDAE-DBN-PSVM framework is proposed that can achieve long-term spatio-temporal anomalous motion/behavior detection, which can guarantee high accuracy and efficiency at the same time.
- To evaluate this deep learning architecture, we conduct experiment and evaluation on a couple of real-world video sequences from the Melbourne Cricket Ground (MCG), which is a large sports stadium in Melbourne. We used the video data of six cameras named C1 to C6. Sample frames from the MCG video are shown in Fig. 2. C1 is installed at the top of venue, and C2-C6 are installed at different place of corridors that it can give diverse perspectives in video data. We detect anomalous behaviors such as standing or loitering, which are important categories of movement anomalies in the MCG dataset.
- In order to compare with other existing methods, we also test on two benchmark datasets: UCSD and Subway, then compare with existing methods.
- Furthermore, we conduct quantitative analyses by varying the amount of training data and the illumination conditions of video in the MCG dataset, which assess the robustness of detection to changing numbers of frames and the brightness conditions.

The rest of the paper is organized as follows. The proposed deep learning architecture is demonstrate in Section II, followed by introducing the detailed components and schemes in Sections III and IV. In Section V evaluation results are discussed, and finally we conclude in Section VI.

II. Methodology

Our proposed architecture for anomalous event detection is illustrated in Fig. 1. This method considers the context and scenario properties of a video, which can depict the activity of moving objects, and use an unsupervised approach. Instead of using hand-crafted features, we first learn a deep model of crowd features [12]. For this, we use two separate scenario level motion feature channels that are extracted from the appearance and foreground of the original video sequences to identify the activity tracks of objects. The obtained motion features are then used with our proposed deep learning architecture to detect anomalous behavior in an unsupervised manner.

Our deep learning architecture consists of three stacked components: SADE, DBN and PSVM. The SDAE is first used to automatically extract and learn the high level features from each channel. The denoising capability of the SADE helps to obtain robust features that are less sensitive to corruption in the input. Next, these features are fed to the DBN-PSVM component. The DBNs help reduce the data dimensionality for decreasing the computational complexity. The output of the DBN is then fed to the PSVM, which identifies the anomalous events in the data and provides an anomaly score. It has been shown that the DBN-PSVM architecture greatly improves the computational complexity as well as the accuracy of detecting anomalies in the data [13]. Here we use it with the SDAE to effectively detect the abnormal activities in video sequences. Finally, the anomalous scores from both the channels are combined using late fusion to detect the anomalies. Next, we describe each of the components of our architecture in detail.

III. Spatio-Temporal Crowd Features

We use two branches to learn a feature representation from (1) the original image sequence/video, as well as from (2) the foreground image sequence, which is extracted using a Gaussian Mixture Model. For these two branches, we use two separate scene-level motion feature channels to extract the movement tracks of objects, which are called continuous motion maps. These are then used with our deep learning architecture for extracting high-level features.

A. Motion Channels

The usual inputs of a Stacked Denoising Auto-encoder are the image patches that have been selected randomly from different frames [9]. In our work, we use a scene-independent motion channel to represent (1) the original image sequence and (2) the foreground sequence (i.e., the foreground that is extracted from the image sequence). The reason why we use this motion channel is that our application environment is quite crowded, and widely-used features like optical flow cannot reliably represent motion patterns. However, some scene-independent properties can depict crowd behavior for groups at the scene-level. We use collectiveness [14] as a property to indicate the degree to which individuals act in unison in a collective motion. This approach is suitable for detecting people loitering in the normal-speed flow of people, which is the most common form of anomalous behavior observed in the MCG dataset.

B. KLT Manifold Collectiveness

The collectiveness descriptor is demonstrated by the Kanade-Lucas-Tomasi (KLT) tracklets [14]. In our experiment, we set the time window of each tracklet to be 30 frames, and the nearest neighbour number K in the KNN graph of the tracklet set is set to 10. The descriptor and computation algorithm proposed in [14] are used to extract crowd collectiveness for every tracklet from the whole frame. This algorithm combines similar paths from the crowd on a collective manifold. The process is briefly described below.

Consider the manifold collectiveness motions of a crowd and example images of real human movement (refer to Fig. 3). We can see that the collectiveness degree of some regions is low in Fig. 3.b, which can indicate people who are loitering.

First, regularity (1) is defined to measure the crowdedness relations between individuals and their neighbors:

$$\omega_t\left(i,j\right)=\max\left(C_t\left(i,j\right),0\right) \tag{1}$$

where i is the current individual, j is the neighbor around i. $C_t(i,j)$ is the tracklet relevance between i and j at time

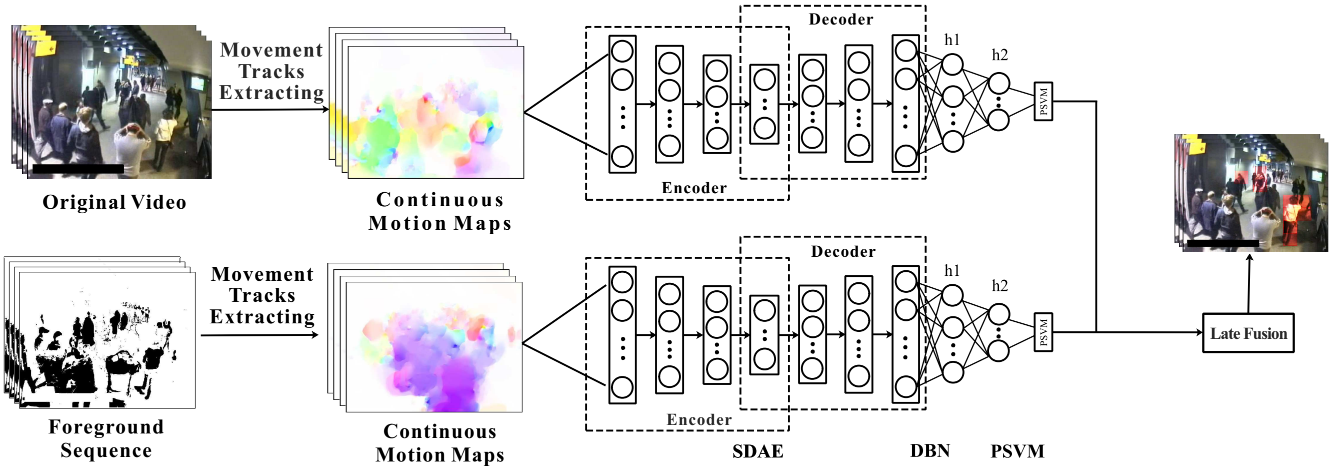


Fig. 1. Overview of the proposed deep learning framework. Original video and foreground video sequence are taken as the input of two branches of channel, then movement tracks are extracted to produce continuous motion maps. Training and testing on a hybrid deep learning model SDAE-DBN-PSVM, then follows to achieve anomalous event detection.

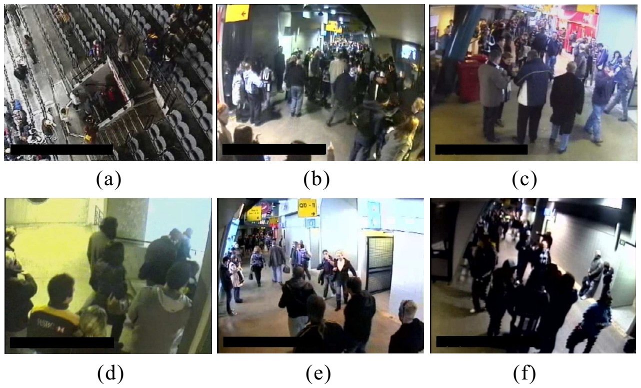


Fig. 2. Sample frames of different cameras from MCG dataset. (a) view of C1, (b) view of C2, (c) view of C3, (d) view of C4 (e) view of C5, (f) view of C6.

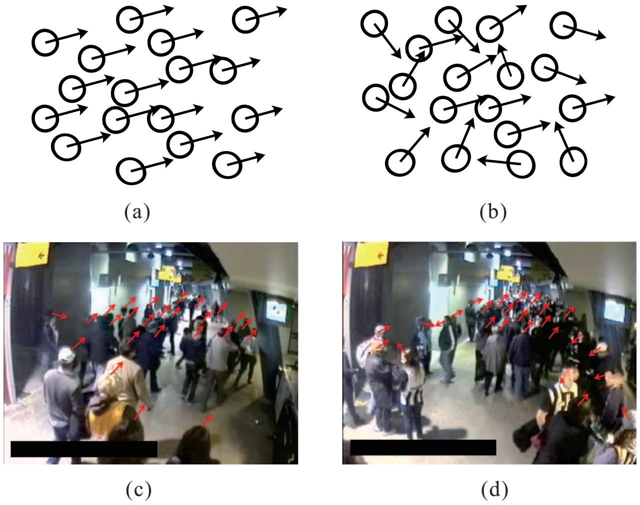


Fig. 3. Sample motion structure and frames showing the different degrees of crowd collectiveness. (a) Collectiveness with high coherence. (b) Collectiveness with low coherence. (c) Individuals in the crowd moving coherently indicate high collectiveness, (d) randomly moving individuals demonstrate low collectiveness.

t, which is computed using KNN. N is defined empirically to represent the interaction relationship among individuals. Each individual in this union has a fixed number of neighbors. Therefore, $\omega_t(i,j) \in [0,1]$ is used for measuring each object's behavior similarity among its neighborhood.

Next, we measure the behavior consistency among pairwise individuals on the collective manifold. Because the consistency of two objects with a long distance between them cannot be predicted accurately, a collective manifold based approach is proposed using pair similarity [15]. By depicting the relevance of the network by a graph, crowd consistency can be measured in a more accurate way.

Let $\gamma_n = \{p_0 \to p_1 \to ... \to p_n\}(p_0 = i, p_n = j)$ denote a path from node p_0 to node p_n, on the weighted adjacency matrix between object i and j. The similarity on this path γ_n is described in (2). Let the set P_n denote all paths with length n between i and j, then the n-path regularity $\upsilon_n(i,j)$ can be defined as shown in (3),

$$\upsilon_{\gamma_n} = \prod_{k=0}^{n} \omega_t \left(p_k, p_{k+1}\right) \tag{2}$$

$$\upsilon_n \left(i,j\right) = \sum_{\gamma_n \in \rho_n} \upsilon_{\gamma_n} \left(i,j\right) \tag{3}$$

At this stage, we can now define individual collectiveness and crowd collectiveness. Since individual collectiveness cannot be directly and accurately summarized on different scales, a regularization function should be generated to summarize all of the path regularities.

After generating a suitable regularization function, individual collectiveness can be calculated as

$$\phi\left(i\right) = \sum_{k=1}^{\infty} z^k \phi_k\left(i\right) = \left[Ze\right]_i \tag{4}$$

where z is the regularization factor, and z^k is the weight for k-path regularity. ϕ_k is the collection of all paths' integrated individual collectiveness, where $\{k = 1 \ldots \infty\}$. The crowd collectiveness is calculated by the average value of all of individual collectiveness,

$$\Phi = \frac{1}{|C|}\sum_{i=1}^{|C|}\phi(i) = \frac{1}{|C|}e^{T}\left(\left(I - zW\right)^{-1} - I\right)e \qquad (5)$$

where e is a vector with all values as 1, and W is associated with crowd set C and is the graph's weighted adjacency matrix.

C. Continuous Motion Maps

After extracting the collectiveness at a scene-level, we produce two types of continuous motion maps as the input to our deep model. For the first type of motion map, in order to address pixel-level anomalous event detection, we divide each collectiveness map into several blocks. Then, for each block, we average the descriptor map of that block. Therefore, each frame produces one vector in the continuous motion map matrix, and each row corresponds to a block. This continuous motion map is the input of the SDAE for loitering detection at the pixel-level. For the other type of motion map, we average the descriptor map per frame across the temporal domain to output a continuous motion map at the frame-level for anomalous event detection. Furthermore, one frame can produce hundreds of tracklets, but the coverage of the tracklets over time is sparse. Therefore, interpolation is used to output the complete and continuous feature map.

IV. Deep Learning Architecture

Given the complexity of the continuous motion matrix from the previous section, we propose a deep learning architecture, which is an unsupervised hybrid architecture called SDAE-DBN-PSVM (as referred to Fig.4) that includes a Stacked Denoising Autoencoder (SDAE), Deep belief net (DBN) and Plane-based one-class SVM (PSVM) [13], [16], [17], that can effectively learn and separate the normal and anomalous data. We have experimented using the SDAE with one-class SVM as in [9], but it failed to achieve high accuracy. Hence, we add the DBN with PSVM in this paper, to achieve a better performance. The two separate motion channels form the input for the deep learning framework. At first, SDAEs are used to learn the feature representation from the continuous motion maps. Then, these motion channels are used to train two-layer DBNs with PSVM for producing anomaly scores. The anomaly scores from the two branches are then combined by a late fusion scheme, which is used for anomalous event detection. Our proposed approach is described in detail below.

A. Stacked Denoising Autoencoder (SDAE)

A SDAE can be used to learn useful representations in a deep network [17]. It can also help increase the performance of PSVM when it is used for learning higher level representations in an unsupervised way.

In the pre-training stage of SADE, a single denoising auto-encoder is learned. After training the mapping function, the output is utilized as the input of the next layer, followed by stacking the denoising autoencoder for the multi-layer feedforward neural network implementation.

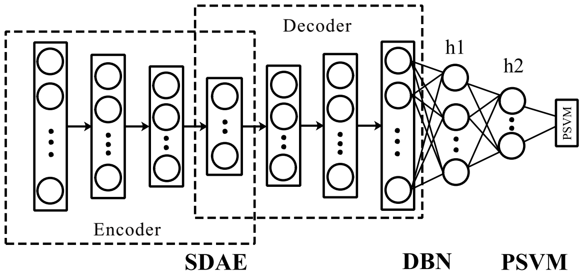


Fig. 4. Our SDAE-DBN-PSVM deep learning architecture.

The next stage is fine-tuning which starts with the training data $\psi^l = \{x_i^l\}_{i=1}^{N^l}$, where N^l denotes the number of training data examples, and l denotes the continuous motion maps from the original image sequence or the subtracted foreground of the image sequence. The objective function used for fine-tuning is based on $2N+1$ layers as follows:

$$J\left(\psi^l\right) = \sum_{i}^{N^l}\left\|x_i^l - \hat{x}_i^l\right\|_2^2 + \lambda_F\sum_{i=1}^{N}(\left\|W_i^l\right\|_F^2 + \left\|{W'}_i^l\right\|_F^2) \qquad (6)$$

where W denotes the weights, and λ_F sets the balance of these two terms. Sparsity constraints are applied on hidden units' outputs to find a suitable data representation.

Stochastic gradient descent (SGD) is used to make convergence in a quicker manner. Further, the input data is split into small patches during the fine-tuning. After the whole process is completed, the output is used as the input of the next stage DBN with a PSVM for anomaly scoring.

B. Deep Belief Nets (DBN)

DBN is a generative model with multiple layers, which is trained to address output data from the previous SDAE layer, so the following PSVM can separate the normal and anomalous data. The aim of using DBNs is to reduce the data's dimensionality, so that the low-dimensional set of features can help speed up our whole propsed architecture [18].

In this paper, DBNs are trained from Restricted Boltzmann Machine (RBM) in a layer-wise manner [13]. An RBM has visible units v that represent observations, and hidden units g for feature representation. n-dimensional input vectors are mapped to the f-dimensional feature space using the DBN, such that f corresponds to $|g|$, where $f < n$. Training an RBM aims to minimize the energy function $E = (v, g)$ that finds the value of the parameter θ. A possible method is to maximize the log-likelihood of v in $E = (v, g)$ using the gradient:

$$\frac{\partial \log \rho(v)}{\partial \theta} = E_{\rho(g|v)}\left[\frac{\partial E(v,g)}{\partial \theta}\right] - E_{\rho(v|g)}\left[\frac{\partial E(g,v)}{\partial \theta}\right] \qquad (7)$$

where $p = (v, g)$ is the combination of visible and hidden vectors, and it is calculated by $p(v,g) = \frac{e^{-E(v,g)}}{z}$.

To train a stacked RBM, we first train a single RBM, followed by training other RBNs and stacking each one on the previous ones. In this paper, we use a three-layer model. After we obtain the stacked RBMs, we initialize the weights of the resulting DBNs in a bottom-up way.

C. Plane-based One-class SVM (PSVM)

Followed by DBN, a Plane-based one-class SVM (PSVM) is used in our architecture for anomaly scoring. The PSVM finds a hyperplane in a higher dimensional feature space that separates the normal data from the anomalies. Although there are other one-class SVMs available, such as Spherical [19] and Ellipsoidal [20] based schemes, PSVM is computational simpler. Furthermore, DBNs are used in front of the PSVM as a feature reduction stage, which can help overcome the limitation of directly using SVM, so that the hybrid model can be used for complex and high-dimensional datasets.

In this scheme, the data vectors $x_i \in R^d (i = 1, 2, ..., l)$ are implicitly projected to a higher dimensional feature space. Next, by solving a quadratic optimization formulation, it finds a hyper-plane that can separate the normal points from the anomalies [16]:

$$\min_{\varpi,\xi,\rho} \frac{1}{2}\|\varpi\|^2 + \frac{1}{\tau l}\sum_{i=1}^{l}\xi_i, \quad (\varpi.\varphi(x_i)) + \xi_i \geq \rho \tag{8}$$

where ϖ denotes the weight vector, ξ_i are the slack variables, which allow some of the data vectors to fall on the other side of the plane, $\tau \in [0, 1]$ is the regularization parameter that controls the fraction of outliers and ρ is the pre-defined offset.

By introducing a kernel function φ, the data are mapped to a higher dimensional feature space. We use a RBF (radial basis function) kernel in this work:

$$k(x, y) = e\frac{-\|x - y\|^2}{2\sigma^2} \tag{9}$$

where σ is the spread of the kernel. After training, the anomaly score for an unseen test point x_t can be found as $\delta = \omega.\varphi(x_t) - \rho$.

The parameter τ provides an upper bound and a lower bound on the fraction of outliers and the support vectors, respectively. In our work, we use a heuristic based unsupervised approach for selecting this parameter efficiently as detailed in [16].

D. Anomalous Event Detection

In order to combine the anomaly scores from the two branches, we use a simple fusion scheme as the last stage. We set the weight vector to $A = [a^0, a^F]$, where a^0 is the weight for the original data branch, a^F is the weight for the foreground branch, and $a^0 + a^F = 1$. We determined the weights heuristically to give the highest accuracy in the training data in this work. Further, we have observed that the weight selection has a small impact on the final outcome. However, one can use a more sphisticated optimisation based approach to find these weights based on the data as detailed in [9].

After obtaining the anomaly score $\delta_i = (\omega.\varphi(x)) - \rho$ from each channel $i; i = 1, 2$, the combined anomaly score δ_c is obtained as follows:

$$\delta_c = a^0\delta_1 + a^F\delta_2 \tag{10}$$

TABLE I
DETAILS OF THE MCG DATASET.

Date	Camera	Time	Resolution
16-Sep-2011	C2, C5, C6	00:18:01	640 × 480
23-Sep-2011	C2, C5, C6	00:22:01	640 × 480
24-Sep-2011	C5, C6	00:14:01	640 × 480
01-Oct-2011	C6	05:15:01	640 × 480

We conduct experiments and evaluation analysis datasets at the frame-level as well as the pixel-level. So we assign $\Psi = 1$ if a frame or a pixel is detected as normal ($\Psi = 1, \delta_c < \eta$), $\Psi = 0$ if it is detected as abnormal. We have experimentally analysed its impact on the detection accuracy in our evaluation. Through changing the threshold η, we could derive the ROC curve in the next section.

V. Results and Discussion

A. Experimental Setup and Datasets

Experimental Setup. The proposed approach is implemented on Visual Studio 2013 and Matlab 2013b. The code for calculating the continuous motion maps is written in Matlab. The experimental laptop is on Windows 10 Intel i7 with Geforce GTX 1060 NVIDIA graphics card.

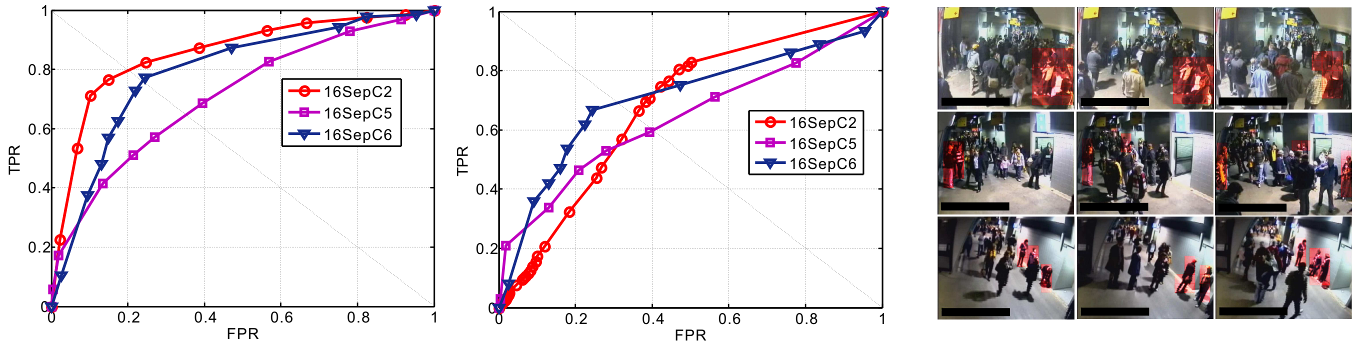
Datasets. Table I gives details of the MCG datasets.

In the following section, we evaluate our framework on the MCG dataset at first, and then in the second experiment, we test on two benchmark datasets UCSD and Subway, and then compare with other state-of-the-art (SOTA) methods. Finally, we conduct experiments on different cameras and different episodes of the MCG datasets for episode evaluation, to see whether the brightness conditions and pre-training time affect the detection performance.

B. Quantitative Evaluation on MCG Dataset

Parameter Setting. For the first layer of SDAEs, the number of neurons is set to be 1000, followed by reducing by half each time in the rest of layers. We use four layers in the encoder part, so the numbers of neurons are defined as: 1000 → 500 → 250 → 125. For the decoder part, it is symmetric to the encoder. In the training phase, the parameter of adding Gaussian noise is set to 0.0001 and the learning rate is set to $\lambda_F = 0.0001$. In the parameters of the DBN, the pre-training learning rate is set to $0.001 - 0.01$ and fine tuning rate is 0.11. The number of epochs is 10 and 20 for pre-training and fine tuning, respectively. The parameters of the PSVMs are tuned based on the Quick Model Selection approach as in [16]. $\gamma(2^{-15}, 2^{-13}, ..., 2^3)$ and $C(2^{-5}, 2^{-3}, ..., 2^{15})$ are used for the RBF kernel and PSVM, and selected using cross validation. In terms of fusion scheme weights, we have used a systematic search for the parameter value $A = [a^0, a^F]$, and our experimental results showed little change, so we choose $[0.6, 0.4]$ as the fusion scheme weights with the highest accuracy.

Frame-level based Anomalous Event Detection. A frame is labeled as abnormal if there is at least one anomalous



(a)

(b)

(c)

Fig. 5. ROC Curve for MCG datasets 16-Sep-Camera 2, 16-Sep-Camera 5, 16-Sep-Camera 6 and Example Detection Results, (a) Detection at Frame-level, (b) Detection at Pixel-level, (c) Example Detection results of C2, C5 and C6.

TABLE II
AUC AND EER RESULT AT FRAME-LEVEL AND PIXEL-LEVEL ON MCG 16-SEP-CAMERA2, 16-SEP-CAMERA5 AND 16-SEP-CAMERA6 DATASETS.

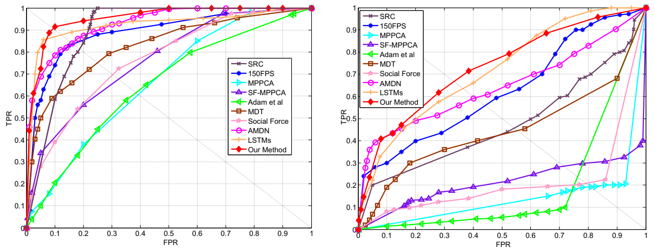
Date	Frame-level		Pixel-level	
	AUC	*EER*	*AUC*	*EER*
16-Sep-C2	85.4%	20.0%	67.6%	35.0%
16-Sep-C5	70.1%	35.0%	64.1%	40.2%
16-Sep-C6	79.1%	23.0%	70.4%	30.7%

object [21]. So abnormal event detection at the frame-level is to detect these frames. In this subsection, we analyzed MCG datasets 16-Sep-2011-C2, 16-Sep-2011-C5 and 16-Sep-2011-C6. The detection result is demonstrated using a ROC curve through changing the threshold η, where the x-axis denotes the false positive rate (FPR) and the y-axis is the true positive rate (TPR). The FPR is those frames that are normal in the ground truth but detected as abnormal, TPR is those frames that are anomalous for both ground truth and detection result. The ROC curves for the MCG datasets are shown in Fig. 5.a.

Then, we utilize Area Under the ROC Curve (AUC) and Equal Error Rate (EER) for quantitative evaluation. EER denotes the value when the false positive rate is equal to the false negative rate. In Table II, we report the AUC and EER of testing on MCG dataset 16-Sep-2011-Camera 2, Camera 5 and Camera 6 at the frame-level.

We can see the results of 16Sep-C2 and 16Sep-C6 are quite competitive. However, the result of 16Sep-C5 is not as good as the other two. The reason is the scene in Camera 5 is quite complex (quite dark, blurred and crowded).

Pixel-level based Anomalous Event Detection. In terms of pixel-level detection, if the detected part is more than 40% of the real abnormal pixels, this detection will be identified as a true positive. On the other hand, if at least one pixel in a normal frame is detected as anomalous, it is identified as a false positive. The results of these three cameras are shown as ROC curves in Fig. 5.b, and the AUC and EER is reported in Table II. We can see that the behavior is similar to the frame-level detection. Note that this feature extraction approach is mainly based on the lighting conditions. If the lighting condition is good enough, our approach can achieve very good performance, but the results can deteriorate in some dark and blurry scenes. Therefore, an important direction for future research is to find methods that can reduce the impact of changes in brightness.



(a)

(b)

Fig. 6. ROC Curve for UCSD Ped1 dataset, (a) Detection at Frame-level, (b) Detection at Pixel-level.

C. Quantitative Analysis on UCSD and Subway Datasets

In this subsection, we apply our architecture on two benchmark datasets: UCSD Pedestrian dataset 1 (UCSD Ped1) [5] and Subway [22] datasets. The UCSD Ped1 dataset [5] is obtained from a surveillance camera on a pedestrian walkway that contains 34 video sequences in training dataset (the scenarios in the category are all normal) and 36 videos in test dataset (which contains abnormal events). Subway dataset [22] is a subway station scene, where anomalous behavior corresponds to moving in the wrong direction. This has two sequences containing entrances (144249 frames) and exit (64900 frames).

In terms of parameter settings, we change the number of neurons of the encoder part to be $1024 \rightarrow 512 \rightarrow 256 \rightarrow 128$, the parameter of adding Gaussian noise is changed to be 0.0003, and the learning rate is fixed to be $\lambda_F = 0.001$. The parameter set of DBN and PSVM are the same as the MCG experiment. $A = [a^0, a^F]$ is also set to $[0.6, 0.4]$.

For the UCSD Ped1 dataset, we evaluate the performance at both frame-level and pixel-level. The results are demonstrated

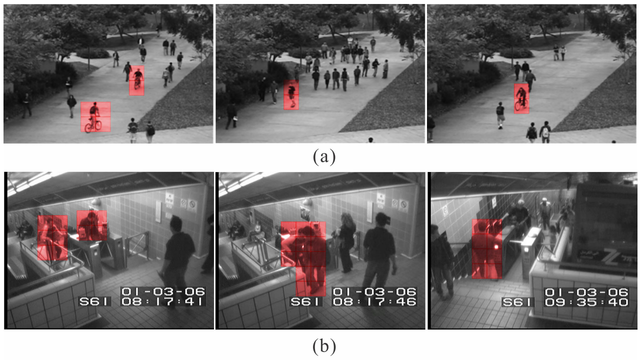
TABLE III
SOTA APPROACHES COMPARISON IN TERMS OF AUC AND EER. FRAME-LEVEL AND PIXEL-LEVEL DETECTION RESULTS ON UCSD DATASET.

Method	**Frame-level**		**Pixel-level**	
	AUC	***EER***	***AUC***	***EER***
SRC [23]	86.0%	19.0%	45.3%	54.0%
150 FPS [24]	91.8%	15.0%	63.8%	43.0%
MPPCA [8]	67.0%	40.0%	19.0%	44.1%
SF- MPPCA [5]	76.9%	32.0%	21.3%	71.0%
Adam [22]	64.9%	38.0%	19.7%	76.0%
MDT [5]	81.8%	25.0%	44.1%	58.0%
SF [25]	76.8%	31.0%	21.3%	71.0%
H-MDT [21]	-	17.8%	66.2%	35.2%
AMDN [9]	92.1%	16.0%	67.2%	40.1%
TCP [11]	95.7%	8.9%	63.4%	41.4%
LSTMs [26]	92.8%	11.5%	71.7%	36.3%
Our Method	**94.3%**	**10.0%**	**70.3%**	**34.0%**

TABLE IV
SOTA APPROACHES COMPARISON IN TERMS OF AUC AND EER. FRAME-LEVEL DETECTION ON SUBWAY DATASET.

Method	**Entrance**		**Exit**	
	AUC	***EER***	***AUC***	***EER***
SRC [23]	83.3%	24.4%	80.2%	26.4%
Saligrama [27]	-	-	88.4%	17.9%
MDT [5]	90.8%	16.7%	90.2%	16.4%
FCNs [10]	90.1%	17.4%	89.7%	16.2%
Our Method	**90.5%**	**16.9%**	**90.8%**	**15.4%**

as ROC curves in Fig. 6. For the Subway dataset, it is tested on a frame-level. Table III and IV show a comparison of our proposed method with other state-of-the-art (SOTA) methods on the UCSD and Subway dataset, respectively. In terms of the UCSD Ped1 dataset, our approach has better performance than most previous methods, and it is comparable with the best baseline [11]. For the Subway dataset, our method outperforms all the competing approaches. Some visualized results are demonstrated in Fig. 7.a (UCSD Ped1 dataset) and Fig. 7.b (Subway dataset).



(a)

(b)

Fig. 7. Examples of anomalous detection on the (a) UCSD dataset, (b) Subway dataset.

TABLE V
MCG DATASET: ACCURACY OF TRAINING 1000 FRAMES FROM 16-SEP-C2, 1000 FRAMES FROM A MIXTURE WITH 16-SEP-C2 AND 23-SEP-C2, AND 2000 FRAMES FROM A MIXTURE WITH 16-SEP-C2 AND 23-SEP-C2.

Datasets	**Frames**	**Accuracy**
16-Sep-C2	1000	85.4%
16-Sep-C2 and 23-Sep-C2	500 (each)	84.7%
16-Sep-C2 and 23-Sep-C2	1000 (each)	85.1%

D. Episode Evaluation on MCG Dataset

We also conducted two experiments on the MCG dataset using different cameras and different episodes in order to test: 1) How does the accuracy change as a result of increasing the amount of training data? 2) How does the accuracy change as a result of increasing the number of episodes in the training data? 3) What is the effect of different episodes? 4) What can we do to decrease the impact of changing illumination conditions and crowdedness? In the following experiments, we use a fixed threshold, and report the accuracy of frame-level based detection.

Different Episodes from the Same Camera. We aim to analyze how the accuracy changes by increasing the amount of training data and number of episodes. We train the model using 1000 frames from 16-Sep-C2 at first, then, using a mixture dataset with 500 frames from 16-Sep-C2 and 500 frames from 23-Sep-C2 (16-Sep-C2 and 23-Sep-C2 are different episodes from Camera 2). After that, we use 1000 frames from 16-Sep-C2 and 1000 frames from 23-Sep-C2 as the training data. We only change the training and keep the same testing data. The results are given in Table V.

As shown in Table V, the obtained accuracy of training using 1000 frames from 16-Sep-C2 is slightly higher than using 500 frames from 16-Sep-C2 and 500 frames from 23-Sep-C2. It shows that although accuracy is influenced by using different episodes with different influence weights, our architecture can handle this change. By comparing the accuracy of the latter two cases, each episode that provides 500 frames has similar performance compared with the case where each one provides 1000 frames. Therefore, accuracy is insensitive to different numbers of pre-training frames. In general, our architecture can guarantee detection performance using a small amount pre-training frames, so it decreases training speed and computational complexity.

Different Episodes from Different Cameras. We conduct a quantitative analysis of how the individual episodes actually influence accuracy, and how to decrease the impact of some episodes with darker and blurred scenes. Therefore, we train:

1) Sequence1: 16-Sep-C2 and 23-Sep-C2 (Mixture episode), each episode provides 1000 frames. We observe the accuracy and the relative change compared with training 1000 frames from 16-Sep-C2 (Single episode). The scenario condition of these two episodes are similar.

2) Sequence2: 16-Sep-C5, 23-Sep-C5 and 24-Sep-C5, each episode provides 1000 frames. We observe the accuracy and

TABLE VI
MCG DATASET: QUANTITATIVE ANALYSIS OF USING DIFFERENT EPISODES AS TRAINING DATA.

Datasets	Mixture Episode	Single Episode	Relative Change
Sequence1	85.1%	85.4%	- 0.4%
Sequence2	74.2%	70.7%	4.7%
Sequence3	74.6%	70.7%	5.2%

the relative change compared with training 1000 frames from 16-Sep-C5.

3) Sequence3: 16-Sep-C5, 23-Sep-C5, 24-Sep-C5 and 01-Oct-C5, each episode provides 1000 frames. We observe the accuracy and the relative change compared with training 1000 frames from 16-Sep-C5. The conditions of episode 16-Sep-C5 are different from 23-Sep-C5, 24-Sep-C5 and 01-Oct-C5, as 16-Sep-C5 is dark and blurred.

For sequence1, we use the rest of 16-Sep-C2 video as test data. For sequence2 and sequence3, the rest of the 16-Sep-C5 video is used as test data. The results are reported in Table VI. Table VI shows that the impact of using more training data is less than the impact of using episodes with different illumination conditions and video quality. Further, the episode with better conditions can balance out the influence of an episode with bad conditions.

VI. CONCLUSIONS

This paper presents a novel framework for anomalous activity detection. This method is based on the context and scenario properties of video, which can depict the activity of moving objects. An approach is proposed to estimate crowd features using deep learning in order to build continuous motion maps, which are then used as the input to a SDAE in order to learn the feature representation automatically. We then combine this with a hybrid model DBN-PSVM and a fusion scheme to perform the final abnormal event detection. Experimental results on a real-world dataset from a major sports stadium demonstrate that this framework can perform frame-level and pixel-level anomalous behavior detection like standing or loitering among a crowd of people. We also test on two benchmark datasets and compare with other state-of-the-art methods. It shows that our approach outperforms these baseline methods. Furthermore, we have provided a quantitative analysis of the impact of the number, duration and lighting conditions of different training episodes on the accuracy of abnormal event detection.

REFERENCES

[1] M. Yang, L. Rashidi, S. Rajasegarar, and C. Leckie, "Graph stream mining based anomalous event analysis," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2018, pp. 891–903.

[2] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 747–757, 2000.

[3] M. Yang, S. Rajasegarar, A. S. Rao, C. Leckie, and M. Palaniswami, "Anomalous behavior detection in crowded scenes using clustering and spatio-temporal features," in *International Conference on Intelligent Information Processing*. Springer, 2016, pp. 132–141.

[4] X. Mo, V. Monga, R. Bala, and Z. Fan, "Adaptive sparse representations for video anomaly detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 631–645, 2014.

[5] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *IEEE Conference on CVPR*, 2010, pp. 1975–1981.

[6] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *IEEE Conference on CVPR*, 2009, pp. 935–942.

[7] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Textures of optical flow for real-time anomaly detection in crowds," in *2011 8th IEEE international conference on AVSS*, 2011, pp. 230–235.

[8] J. Kim and K. Grauman, "Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates," in *IEEE Conference on CVPR*, 2009, pp. 2921–2928.

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

[10] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, 2018.

[11] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, "Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1689–1698.

[12] J. Shao, C. C. Loy, and X. Wang, "Learning scene-independent group descriptors for crowd understanding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, pp. 1290–1303, 2017.

[13] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[14] B. Zhou, X. Tang, and X. Wang, "Measuring crowd collectiveness," in *IEEE Conference on CVPR*, 2013, pp. 3049–3056.

[15] N. Biggs, N. L. Biggs, and B. Norman, *Algebraic graph theory*. Cambridge university press, 1993, vol. 67.

[16] Z. Ghafoori, S. Rajasegarar, S. M. Erfani, S. Karunasekera, and C. A. Leckie, "Unsupervised parameter estimation for one-class support vector machines," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2016, pp. 183–195.

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, 2006.

[19] D. M. Tax and R. P. Duin, "Support vector data description," *Machine Learning*, vol. 54, pp. 45–66, 2004.

[20] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, "Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 5, pp. 518–533, 2010.

[21] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 18–32, 2014.

[22] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 555–560, 2008.

[23] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *IEEE Conference on CVPR*, 2011, pp. 3449–3456.

[24] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2720–2727.

[25] V. Reddy, C. Sanderson, and B. C. Lovell, "Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture," in *IEEE Conference on CVPRW*, 2011, pp. 55–61.

[26] Y. Feng, Y. Yuan, and X. Lu, "Deep representation for abnormal event detection in crowded scenes," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 591–595.

[27] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *IEEE Conference on CVPR*, 2012, pp. 2112–2119.