



An efficient deep neural model for detecting crowd anomalies in videos

Meng Yang¹ · Shucong Tian¹ · Aravinda S. Rao² · Sutharshan Rajasegarar³ · Marimuthu Palaniswami² · Zhengchun Zhou¹

Accepted: 1 October 2022 / Published online: 25 November 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Identifying unusual crowd events is highly challenging, laborious, and prone to errors in video surveillance applications. We propose a novel end-to-end deep learning architecture called Stacked Denoising Auto-Encoder (DeepSDAE) to address these challenges, comprising SDAE, VGG16 and Plane-based one-class Support Vector Machine (SVM), abbreviated as PSVM, to detect anomalies such as stationary people in an active scene or loitering activities in a crowded scene. The DeepSDAE framework is a hybrid deep learning architecture. It consists of a four-layered SDAE and an enhanced convolutional neural network (CNN) model. Our framework employs Reinforcement Learning to optimise the learning parameters to detect crowd anomalies. We use the Markov Decision Process (MDP) with Deep Q-learning to find the optimal Q value. We also present a *late fusion* procedure to combine individual decisions from the intermediate and final layers of the SDAE and VGG16 networks to detect different anomalies. Our experiments on four real-world datasets reveal the superior performance of our proposed framework in detecting (frame-level and pixel-level) anomalies.

Keywords Crowd behavior · Anomaly detection · Motion features · Late fusion · Video surveillance

1 Introduction

Video-based crowd motion analysis is a fundamental problem in surveillance applications. In particular, anomalous event detection is one of the most popular tasks in the domain of crowd motion analysis. Varadarajan and Odobez [1] defined a *crowded event* as an action that happens at temporal and spatial levels. Anomalous crowd behavior or events denotes any sudden incidents that capture human attention or exhibit different behavior patterns from the regular pattern of behavior. There is an increasing need to monitor public places (shopping malls, transport stations, airports, streets, and other gathering places) to identify anomalous crowd patterns, such as sudden changes in crowd movements or size. Furthermore, there is a need to detect anomalous events in an emergency, such as in public places, and raise the alarm accurately and timely.

Existing research has contributed significantly to monitoring human behavior using large surveillance cameras. The widely used closed-circuit television (CCTV) for crowd monitoring helps to improve security up to a certain extent. However, the CCTV alone cannot provide a complete perspective of all the places because of their “disjoint and fragmented” coverage [2]. For example, CCTV cameras are usually installed in fixed locations and have limited (fixed field of view) coverage. Moreover, employing and maintaining ubiquitous surveillance cameras is highly expensive.

Additionally, there is a high chance of not detecting any crowd events at large crowded venues because the security personnel who monitor the CCTV feeds are susceptible to fatigue and concentration loss. Human observers are not always sensitive to detect an event’s sudden occurrence: they cannot pay attention to all of the anomalous objects or behaviors in a scene. Hence, it is challenging to distinguish these unusual events from ordinary activities based solely on human observation.

In order to improve crowd event detection performance and tackle the challenges associated with real-life applications on a large scale, an automated monitoring system is needed to detect, analyse, and predict crowd behavior [3–5].

✉ Zhengchun Zhou
zzc@swjtu.edu.cn

Extended author information available on the last page of the article.

Thus, analysing and understanding a large group of people in crowds have attracted increasing attention. Although crowd analysis covers many tasks, this paper focuses on anomalous event detection.

This paper proposes a novel end-to-end hybrid deep learning framework for highly accurate anomalous object detection and abnormal movement pattern detection. In contrast to the preliminary analysis presented in [6], this work explores a new architecture called the DeepSDAE model, as shown in Fig. 1). The proposed framework employs a Stacked Denoising Auto-Encoder (SDAE) and an improved VGG16 model [7]. In particular, we extract continuous motion trajectories as pre-features and feed them to the SDAE. In contrast to the work presented in [6], we first extract two output channels from the SDAE's hidden and final layers. We then feed these channel outputs to (1) a Plane-based one-class Support Vector Machine (1SVM), abbreviated as PSVM and (2) an optimised VGG16 that feeds to another PSVM. Finally, we combine the decisions from these two PSVM outputs with reinforcement learning to detect crowd anomalies using a late fusion mechanism. Our approach detects anomalous crowd movement behaviors, such as those in the baseline datasets, namely the UCSD [8], Avenue [9, 10] and Subway Surveillance [11] datasets. The abnormal activities include non-pedestrian objects, such as people riding bicycles, skaters, carts, and anomalous motions in the UCSD dataset [8]; strange actions, a person walking in the wrong direction and non-pedestrian objects in the Avenue dataset [9, 10]; people entering and exiting the subway in the wrong direction, loitering and irregular interactions in the Subway Surveillance dataset [11]. In addition, it can detect abnormal activities, such as standing and loitering people appearing

in the densely crowded Melbourne Cricket Ground (MCG) dataset [5, 12]. Our main contributions in this work are:

1. The proposed novel end-to-end deep neural network architecture, called DeepSDAE, enables long-clue Spatio-temporal crowd motion anomaly detection, where the raw videos are used twice via two channels. We use this proposed approach only once while still achieving superior performance. Moreover, the proposed framework convergence is stable and quick because of our optimised deep learning framework.
2. We derive a set of optimal parameters for detecting the anomalies by modelling the crowd flow process as a Markov Decision Process (MDP) and solving them using a Deep Q-learning (DQN) method [13]. As a result, our approach produces a generalised set of learned parameters, improving the ability to detect similar events from different, new scenarios.
3. We evaluate our proposed framework on the MCG dataset [5, 12], the UCSD Ped1 and Ped2 [8], Avenue [9, 10], and Subway Surveillance [11] datasets. These datasets consist of various abnormal crowd activities, and we evaluate 13 other approaches comprehensively. The proposed DeepSDAE frame outperforms existing approaches in detecting anomalies (frame level or pixel level) in crowded scenes.
4. The DeepSDAE is a novel Reinforcement learning - deep learning model to detect crowd anomalies, where RL is firstly introduced to explore the parameter set. It produces superior results by learning the optimal crowd anomaly detection parameters (i.e., the time window of each tracklet, the neighbouring relationship of individuals and fusing decisions to arrive at anomaly

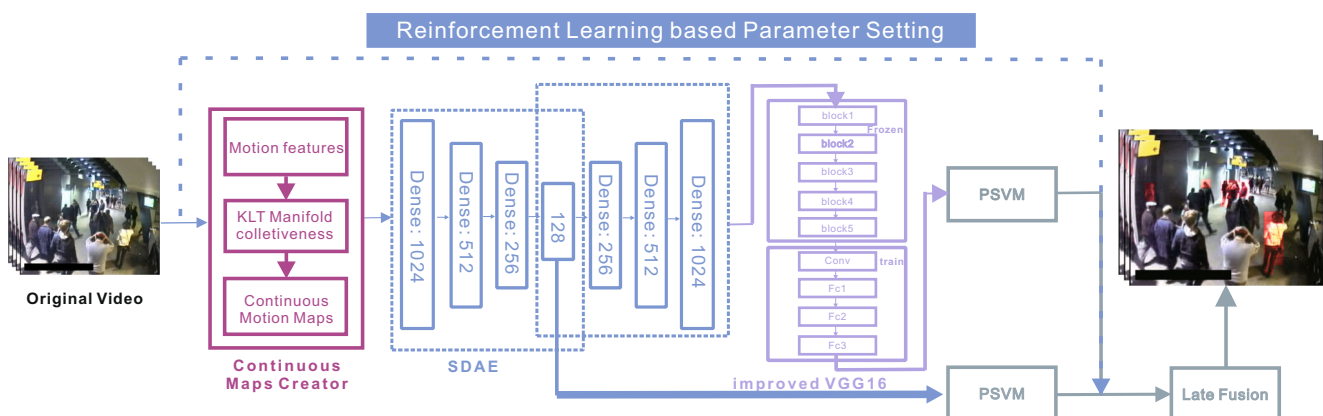


Fig. 1 Overview of the proposed end-to-end crowd anomaly detection architecture. We extract the motion features from the raw videos and represent the crowd trajectories as crowd collectiveness using Kanade-Lucas-Tomasi (KLT) tracklets' manifold, rendering continuous motion maps. The proposed DeepSDAE framework comprises SDAE with an optimised VGG16 model to discern the normal and abnormal

movement patterns. Classifying normal and abnormal patterns in highly crowded scenarios is challenging. Therefore, we introduce two information channels in the DeepSDAE framework with PSVMs to produce two anomaly scores and merge the decisions via the late fusion scheme, delivering outstanding crowd anomaly detection results

scores) via the Deep Q-learning by maximising the expected rewards.

The structure of this article is as follows: Section 2 provides the literature review on abnormal crowd detection methods. In Section 3, we introduce our proposed methodology. Section 4 provides the evaluation results for the four datasets, and Section 5 provides the conclusion.

2 Related work

We find several methods proposed to detect anomalous crowd events in video surveillance applications. We will review the recent abnormal crowd detection approaches in this section. In the literature, we find crowd tracking algorithms [5, 14], object detection methods [15] and detecting people fighting methods [16]. However, these approaches work only in specific experimental scenarios; finding a general approach is still challenging. Motion representation methods (with texture and dynamic models) [17, 18] and sparse coding [19] have partially solved these challenges. These methods approached abnormal detection as the outlier detection problem. The generalised approach considers training regular image sequences and testing whether the new incoming frame is normal or abnormal compared with the trained regular pattern.

The Mixture of Dynamic Textures (MDT) [17] captures the crowd dynamics and the appearance in crowded scenarios to represent crowd behavior. The hierarchical mixture of dynamic texture (H-MDT) li2014anomaly modifies the original MDT to improve the temporal anomaly detection process. This modification involves partitioning each frame into multiple sub-blocks and extracting the small patches with spatio-temporal information as MDT. The resulting H-MDT creates multi-scale anomaly maps, both at temporal and spatial levels. An online conditional random fields (CRF) scheme detects anomalous events, fusing temporal and spatial anomaly maps. The results demonstrate that the H-MDT outperforms the MDT [17]. However, the training of CRF requires handcrafted annotated training samples, making it less attractive for real-world applications.

Deep learning has demonstrated exemplary performance in object detection and behavioral analysis. Likewise, deep learning is a promising approach for anomalous event detection. Xu et al. [20] have proposed a novel architecture, called Appearance and Motion DeepNet (AMDN), for anomalous event detection. Feng et al. [21] use the SDAE to extract the spatial features. They implement the time-dependent Long Short-Term Memory (LSTM), capturing the long-term clues. Hasan et al. [22] combine conventional handcrafted Spatio-temporal local features and convolutional fully-connected feed-forward auto-encoder

(Conv-AE) with learning local features and classifying activities. Chong et al. [23] proposed convolutional LSTM (ConvLSTM) to combine spatial features and temporal evolution of spatial features to detect video anomalies. The authors demonstrated that their framework works better than [22]. Dubey et al. [24] studied the joint learning approach to extract appearance and motion features using context-dependency called the Deep-network with Multiple Ranking Measures (DMRMs). Morias et al. [25] extract the dynamic skeleton features using a message-passing encoder-decoder recurrent network for anomaly detection.

The authors in [19] proposed a stacked Recurrent Neural Network (RNN) based deep model implementing coherent sparse coding. In addition, we find high-level feature extracting deep learning models proposed to detect anomalous event tasks [26, 27], such as the Fully Convolutional Neural Networks (FCNs) [26] and Plug-and-play Convolutional Neural Networks (PCCNs) [27]. Recently, algorithm called CO-attention Siamese Network (COSNetin) [28, 29] is utilized to tackling this problem, leading to a zero-shot solution, which is a unified and end-to-end trainable architecture, that can catch diverse joint feature. Dubey et al. [24] studied the joint learning approach to extract appearance and motion features using context-dependency called the Deep-network with Multiple Ranking Measures (DMRMs). By combining optical flow and HOG, Mishra et al. [30] presented a tensor-based model for motion description to identify any unusual behavior in the crowd scene. The authors [31, 32] proposed a two-fold CNN based framework to complete end-to-end solution, which can achieve robust classification with a specific and dedicated deep learning heuristic.

3 Methodology

Figure 1 illustrates our proposed end-to-end architecture, which uses continuous motion maps as input for the proposed DeepSDAE model for performing crowd anomaly detection. Continuous motion map is learned from raw video using a deep model that comprises crowd movement features [33]. The proposed deep network model includes a four-layer SDAE with a transfer learning-based VGG optimisation model.

We first extract the continuous motion maps from the raw videos. The continuous motion maps help build the relation between appearance and motion features. The creation of the motion map involves extracting the motion features, represented via the KLT manifold collectiveness and finally rendering the continuous motion maps [6]. We then input these maps to the DeepSDAE. The denoising characteristic of DeepSDAE makes the framework more robust to changing crowd motions.

We use transfer learning to optimise the original VGG model to overcome the overfitting problem and improve the model generalisation. The optimised component comprises one convolution layer and three FC layers. The outputs from the VGG and the hidden layer of SDAE are fed to two PSVMs via two channels, respectively, to produce different anomaly scores. We chose PSVM in our architecture because of its demonstrated high anomaly detection performance in the recent work [7]. We employ the late fusion mechanism to combine the two anomaly decision scores. We derive the weight vectors for combining the decisions using the DQN framework.

3.1 DeepSDAE framework

In previous works, experiments performed on the MCG dataset [5, 12] using SDAE with one-class SVM [20] showed limitations in achieving good performance because the scenarios in the MCG dataset are highly crowded. This section introduces the unsupervised hybrid deep model called DeepSDAE to perform crowd anomaly detection. Figure 1 shows our framework, including an SDAE, an optimised VGG16 model, and a PSVM [7, 34] to efficiently separate the normal and abnormal patterns.

Detecting anomalies include first extracting the features and then applying SDAE to learn the embeddings. The SDAE produces two outputs: we feed (1) the first output from the hidden layer of the SDAE to a PSVM to produce an anomaly score, and (2) the second output from the decoder of the SDAE to the improved VGG model, followed by a PSVM, to produce another anomaly score. Finally, the outputs from the two PSVMs are combined using a deep reinforcement learning-based late fusion [35] mechanism to detect anomalous crowd events.

3.1.1 Learning representations

In this work, we exploit the SDAE's capability to learn useful representations from motion feature maps. At the pre-training stage, the target is to find out a suitable mapping function, and train one-layer auto-encoder, where are then being fed to the next layer, till forming stacked four-layer feedforward denoising neural network.

For fine-tuning, we use the the training data $\phi^c = \{d_i^c\}_{i=1}^{M^c}$, where c denotes the motion maps and M^c represents the number of training samples. The objective function in DeepSDAE model is given by,

$$J(\phi^c) = \sum_i^{M^c} \left\| d_i^c - \hat{d}_i^c \right\|_2^2 + \tau_F \sum_{i=1}^M \left(\left\| \omega_i^c \right\|_F^2 + \left\| \omega_i'^c \right\|_F^2 \right) \quad (1)$$

where d_i^c and \hat{d}_i^c represent the input motion features and the reconstructed sample of the SDAE; τ_F aims at setting

the balance between the two terms in the objective function by regularising the weights of the encoder and the decoder; ω_i^c and $\omega_i'^c$ represents the corresponding weights in encoder and decoder segments of the SDAE. The output of the hidden layer is used with sparsity constraints, aiming at making data representation perform better. We apply sparsity constraints on the outputs of the hidden units to find a valuable data representation. We use Stochastic Gradient Descent (SGD) for a stable and guaranteed global convergence.

3.1.2 Optimising the VGG16 model

The Computer Vision Group from Oxford University proposed a VGG16 architecture, in 2014, for performing deep learning-based classification tasks. The model has 16 layers comprising 13 convolutional layers and three FC layers. We can arrange the layers into five blocks (block 1 to block 5) with different convolutional and pooling layers. For example, blocks 1 and 2 in the VGG16 model have two convolutional layers followed by a max-pooling layer. In contrast, blocks 3-5 have three convolutional layers followed by the max-pooling layer. Among them, the size of all convolution kernels is 3×3 , and the size of pooling kernels is 2×2 . Therefore, the VGG16 network replaces the convolution kernels of size 5×5 and 7×7 by stacking the convolution kernels 3×3 . As a result, the optimised VGG16 model can obtain the same receptive field, significantly reducing the number of parameters and increase the depth of the network [36]. In addition, the stacked convolutional layers increase the non-linear transformation layers and feature extraction capability of the network. The non-linear transformation here uses the ReLU function, defined as:

$$\text{ReLU}(x) = \max(x, 0) \quad (2)$$

VGG16 model includes many parameters, and most parameters are concentrated in the FC layers. To decrease the overfitting of the model during training, we use transfer learning and fine-tuning [37] to optimise the VGG16 model. The optimised VGG16 model comprises a trained VGG16 model and a convolutional neural network, as shown in Fig. 2. We use binary cross-entropy metric as the loss function, which is defined as

$$\text{loss} = - \left(\sum_i^n y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right), \quad (3)$$

where n is the total number of samples, y_i is true label of the samples, and \hat{y}_i is the predicted label.

The parameters in the five blocks (1 to 5) in the pre-trained VGG-16 network are retained and frozen, reducing the number of trainable parameters, and a simple CNN

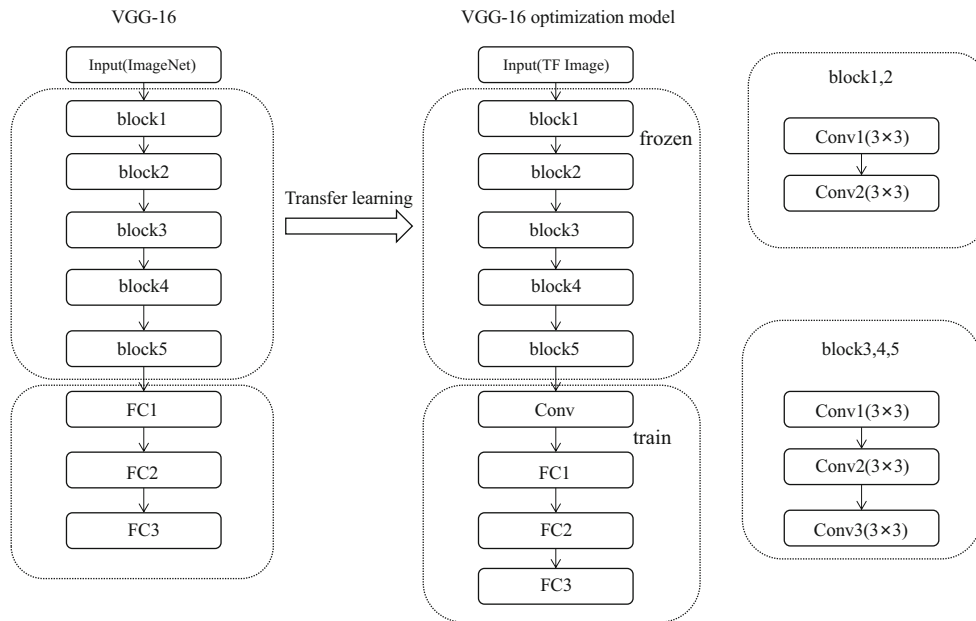


Fig. 2 The proposed VGG16 optimisation model

replaces the FC layer. Since the network contains only one convolutional layer, a relatively small dataset can meet the training requirements. Figure 2 shows the structure of our proposed VGG16 optimisation model.

When constructing a VGG16 model, we first initialise the model parameters based on ImageNet and then freeze the parameters of 5 blocks. Then, we connect the constructed simple CNN for training. Table 1 shows the changes in parameters after freezing. From Table 1 we clearly see that the number of trainable parameters that significantly reduced between VGG-16 (15,895,105) and optimized VGG-16 (1,180,417), which is a 92.6% reduction $\left(\frac{14,714,688}{15,895,105}\right)$ in trainable network parameters and the reduction is observed in the training time.

3.1.3 Classifying anomalies

We implement PSVM [7, 34] in this step for classifying anomalies. PSVM aims at finding a hyperplane from the high-dimensional feature space that can classify the normal and abnormal data. The reason we chose PSVM is that it is computationally simpler. Implementing PSVM involves implicitly projecting the data vectors $x_i \in R^d (i = 1, 2, \dots, n)$ to a high-dimensional feature space via kernel functions and requires solving the following quadratic optimisation problem

$$\min_{w, \xi, \rho} \frac{1}{2} \|W\|^2 + \frac{1}{\sigma \cdot n} \sum_{i=1}^N \xi_i, \quad (W * \varphi(x_i)) + \xi_i \geq \bar{\theta}, \quad (4)$$

Table 1 Parameters of the optimised VGG16 model

Layer(type)	Parameter	Layer(type)	Parameter	Output shape
Original VGG16	15,895,105	Optimized VGG16	1,180,417	(None,4,4,512)
Block 1-5	14,714,688	Frozen Layer	0	(None,4,4,512)
Conv2d_1	131,328	Conv2d_1	131,328	(None,4,4,256)
Flatten_1	0	Flatten_1	0	(None,4096)
Dense_1 (Dense)	1,048,832	Dense_1 (Dense)	1,048,832	(None,256)
Dense_2 (Dense)	257	Dense_2 (Dense)	257	(None,1)
Total params: 15,895,105	Total params: 15,895,105			
Trainable params: 15,895,105	Trainable params: 1,180,417			
Non-trainable params: 0	Non-trainable params: 14,714,688			

The frozen blocks in the modified VGG-16 model increases the compactness of the model because of the frozen parameters

where W denotes the weight vector, σ denotes the regularisation parameter indicating the proportion of anomalous data from the entire data; ξ_i are the slack variables that allow us to control the vectors on either side of the classification hyperplane and $\bar{\theta}$ is a pre-defined offset that allows that aids in classification. We make use of the radial basis function (RBF) kernel to map input data to the high-dimensional feature space. The RBF is defined as

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\tau^2}}, \quad (5)$$

where τ denotes the extent of spread of the RBF kernel. The anomaly score in the testing data are denoted as x_t , where it can be explored by using $\delta = W * \varphi(x_t) - \bar{\theta}$. The σ denotes the lower boundary for the support vectors and the corresponding upper boundary for the proportion of abnormal data. An unsupervised PSVM model training method is implemented in this paper [34].

3.1.4 Merging decisions via late fusion scheme

The anomalies score output from two branches is combined by implementing *late fusion*. Late fusion can learn semantic representations directly from the unimodal features [35]. In addition, we use a weight vector to merge these two PSVM values as $A = [\omega, 1 - \omega]$, where ω is the weight of the upper branch (SDAE-VGG-PVSM), and we choose this parameter by utilising a reinforcement learning. Anomaly score from each branch ($\delta = W * \varphi(x_t) - \bar{\theta}$) is calculated and the final anomaly decision score is given by

$$\delta_c = \omega\delta_1 + (1 - \omega)\delta_2. \quad (6)$$

We use a flexible threshold η to depict the receiver operating characteristic curve (ROC) in our work. If $\delta_c < \eta$, then the frame will be recognised as normal data at the frame level and the associated pixels will also be detected as normal data at the pixel level. On the other hand, if $\delta_c \geq \eta$, then the frames and the related pixels will be treated as anomalies. The ROC curve and the corresponding Area Under Curve (AUC) are obtained by changing the threshold η . The details are discussed in Section 4 (experimental evaluation).

3.2 OP-RL method

Learning optimal parameters is an essential but difficult task, and we usually choose them empirically. We need to learn a set of three parameters, N , K and ω , for detecting crowd anomalies. The parameters N and K are from the motion maps. The parameter N denotes the value of the time window of each tracklet and K denotes the parameter about the number of nearest neighbour in k -NN graph, where it is used to represent the interactions among the individuals

obtained from the KLT tracklets. The third parameter, ω , is the weight vector of the first branch in the late fusion scheme, producing the combined anomaly decision score δ_c . We develop a reinforcement learning approach to select the optimal parameters. We model the parameter learning process as the MDP and propose a optimal parameter learning method (OP-RL) to find the optimal parameters for anomalies detection.

In reinforcement learning, we let the *agent* learn an interactive environment by trial and error while using a reward to measure its interactions. Figure 3 demonstrates the fundamental elements and the process involved in reinforcement learning. An *environment* is a world in which the agent operates. State S_t is the current situation sequence, and S_{t+1} is the subsequent sequence the agent perceives. The reward is the feedback from the environment, and it also measures the influence of action (A_t), where t represents the current time. The agent's target is to interact with the environment using different actions and get the maximal reward in the future. The reward R_t at timestep t could be defined as $R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$, where γ is a discounted factor. The optimal action-value function, $Q^*(s, a)$, is calculated using $Q^*(s, a) = \max_{\pi} \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$, where π is a policy sequence of actions.

The optimal action-value function follows an identity called the Bellman equation. This equation is based on the intuition that if the optimal value $Q^*(s', a')$ of the state sequence s' at the next timestep was known for all possible actions a' , then the optimal strategy is to select the action a' that maximises the expected value $r + \gamma Q^*(s', a')$, where,

$$Q^*(s, a) = \mathbb{E}_{s' \sim \varepsilon} \left[r + \gamma \max_{a'} Q^*(s', a') | s, a \right]. \quad (7)$$

As discussed above, the reinforcement learning algorithm interactively updates the action-value function using the Bellman equation. The action-value function can be

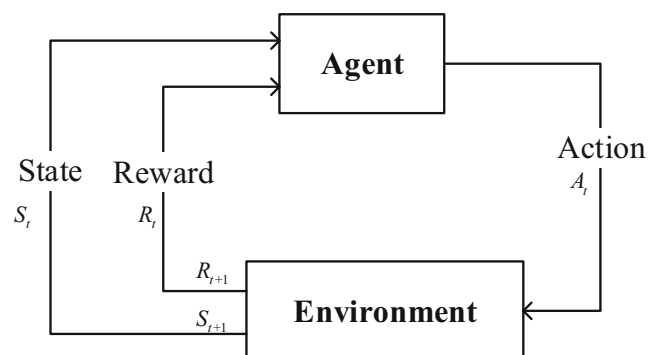


Fig. 3 The basic elements and the processes involved in reinforcement learning. The *environment* is a world in which the agent operates. State S_t is the current situation sequence, and S_{t+1} is the subsequent sequence the agent perceives. The reward is the feedback from the environment, and it also measures the influence of action (A_t), where t represents the current time

denoted as $Q(s, a; \theta)$, where the weight θ can be learned with a neural network such as the Q-network. Training the Q-network involves minimising the loss function $L_i(\theta_i)$ for each iteration i , given by

$$L_i(\theta_i) = \mathbb{E}_{s,a \sim \rho(\cdot)} \left[(y_i - Q(s, a; \theta_i))^2 \right], \quad (8)$$

where $y_i = \mathbb{E}_{s' \sim \varepsilon} [r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a]$ is the target for iteration i and $\rho(s, a)$ is a probability distribution over the State sequence s and the action sequence a . When optimising the loss function $L_i(\theta_i)$, the parameter θ_{i-1} is fixed. The gradient is calculated using $\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_s \left[(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)) \nabla_{\theta_i} Q(s, a; \theta_i) \right]$. Note that when optimising the loss function using SGD, computing the gradient is usually computationally expensive.

Markov decision process(MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, P is the state transition probability from state s to the next state s' under action a , R is a reward function represents the expected reward for executing action a in state s , and $\gamma \in (0, 1)$ is the discount factor. The agent interacts with the environment with its policy π , a mapping from state to action.

In terms of the detailed algorithm, we adopt the Deep Q-learning (DQN) to solve the parameter optimization problem (refer to Fig. 4). DQN is the first deep reinforcement learning method proposed by DeepMind [13] to learn control policies directly from the high-dimensional sensory inputs. The process of Q-value iteration in the DQN framework is the same as what we discussed before and Fig. 5 illustrates our entire approach. In reinforcement learning, the main component for one episode¹ is (state, next state, action, reward). State is expressed as S_t , next state as S_{t+1} , action as set A_t , and the reward as R_t at the current time t . We use the generated motion maps as the state S_t . The parameters N , K and ω are chosen from the action set $A_t = (N, K, \omega)$. In the current episode t , reward R_t is calculated as $R_t = \delta_c$ to measure how the actions are chosen. The reward, R_t , in this work is a value between 0 and 1, and the higher reward indicates the action chosen being better. For the next episode $t + 1$, we use the next state of the updated motion maps S_{t+1} .

The optimal action-value function $Q^*(S_t, A_t)$ is defined as the maximum expected return achievable by following any strategy, after seeing the State sequence S and taking the action a . The objective is to find an optimal policy which can maximize the expected discounted long-term reward $\tilde{R} = \mathbb{E}_{\pi} [\sum_{t=0}^{\infty} \gamma^t R_t]$, where where π is a policy mapping sequences to actions. Through the iterative process, RL model converges, when the value of \tilde{R} is maximal. Finally,

¹Sequence of states, actions and rewards that reach a terminal state

the decided action set $A_t = (N, K, \omega)$ by using the above converged model corresponds to the optimal parameter set.

As the value of N , K and ω are all discrete, which is also the reason why we take DQN as the reinforcement learning algorithm. The action space chosen in DQN is discrete, where action space is continues in some other algorithms like DDPG. In details, for N , the value in action space is changing from 3 to 10, round to one decimal place. So for this value, the size of action space for this value is 70. In terms of parameter K , the value in action space is changing from 1 to 10. The size of action space for this value is 10. For parameter ω , the value in action space is changing from 0 to 1, round to one decimal place. The size of action space for this value is 10.

4 Results and discussion

4.1 Experiment setup

We implemented our proposed framework in C++ and Python using Anaconda 3.5 package manager in Visual Studio 2015. We used Windows 10 Operating System (OS) and NVIDIA[®] Geforce[®] GTX 2080Ti graphics card for our experiments.

In the four-layer SDAE architecture we used, the number of neurons in the first layer is set to be 1024, every time reduced by half for the rest of the layers. Precisely, the SDAE encoder consists $1024 \rightarrow 512 \rightarrow 256 \rightarrow 128$ neurons in the four layers, and the corresponding decoder neurons in the four layers as $128 \rightarrow 256 \rightarrow 512 \rightarrow 1024$. The learning rate λ_F is set to 0.0001. At the pre-training and fine-tuning stage, the number of epochs is set to 10 and 20, respectively. We used Quick Model Selection method [34] for tuning the parameters of the PSVMs, where $C = (2^{-5}, 2^{-3}, \dots, 2^{15})$. The best value of γ for the RBF kernel is selected from $\gamma = (2^{-15}, 2^{-13}, \dots, 2^3)$, using cross validation. We used the tuning parameters provided in [34].

The optimal parameters learned by using Reinforcement Learning are given in Table 2. The parameters N and K are from the motion maps, where the value of N denotes the value of the time window of each tracklet and K denotes the parameter about the number of nearest neighbour in k -NN graph, where it is used to represent the interactions among the individuals obtained from the KLT tracklets. In terms of ω , it is the weight vector of the first branch in the late fusion scheme, producing the combined anomaly decision score δ_c .

It seems that the parameters are adaptive to different type of videos, based on different type of anomalies, but the change of these parameters are very slight.

We utilise SDAE to learn valuable representations in an unsupervised manner. In addition, PSVM is highly effective in classifying the outliers [34]. Therefore, the proposed

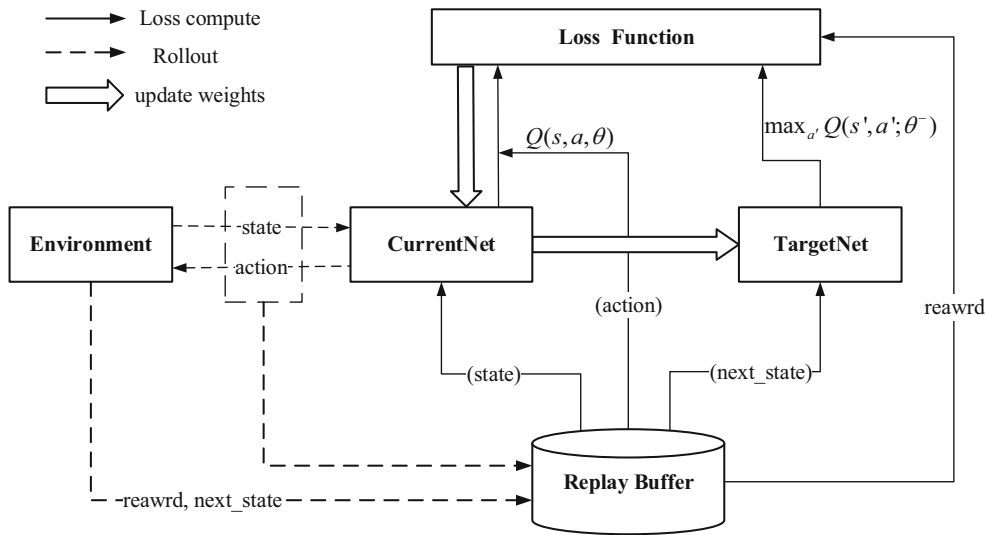


Fig. 4 Deep Q-learning framework for learning optimised set of parameters

method guarantees better performance by overcoming the lousy performance caused by the noise and unlabelled training data in traditional one-class SVM. Furthermore, we implement DQN to learn optimal parameters instead of manually choosing parameters.

We evaluate our proposed framework on four datasets. These include UCSD [8], Avenue [9, 10], Subway Surveillance [11] and densely crowded MCG datasets, and we compare the results with existing state-of-the-art approaches in the recently published papers.

4.2 MCG dataset

MCG dataset is a real-world Ground dataset collected from a sport stadium of Melbourne. We use videos from C2, C5 and C6 cameras installed at different corridors in MCG. Figure 6 shows the sample images obtained from the MCG dataset.

The MCG dataset is a real-world dataset collected from a sports stadium in Melbourne [5, 12]. It consists of different

camera views of actively moving crowds in the MCG corridors with people standing and loitering events. The data includes videos collected on four dates from six different cameras (C1 to C6) during Australian Football League (“footy”) matches at MCG, summing to approximately 31 hours of data [5, 12]. Specifically, we use 16-Sep-C2, 16-Sep-C5 and 16-Sep-C6 video sequences from C2, C5 and C6 cameras to evaluate our framework to detect loitering events. Figure 6 shows the sample images obtained from the MCG dataset.

4.2.1 Frame-level abnormal crowd detection

We consider loitering events abnormal in densely crowded scenes and aim to detect frame-level loitering events. A *frame* is defined as anomalous if there is at least one actual abnormal pixel (related to a loitering event) in the frame. We obtain the ROC curve and the AUC by changing the threshold η . The x -axis denotes the False Positive Rate (FPR), and the y -axis denotes True Positive Rate (TPR).

Fig. 5 Our proposed OP-RL model for learning optimal parameters in this work

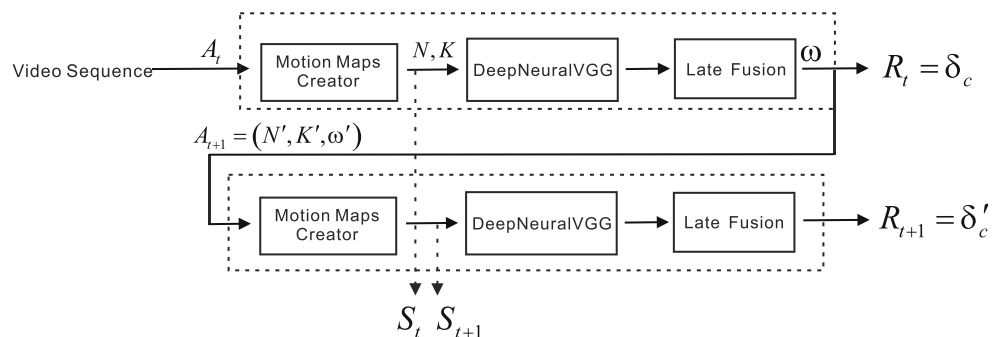


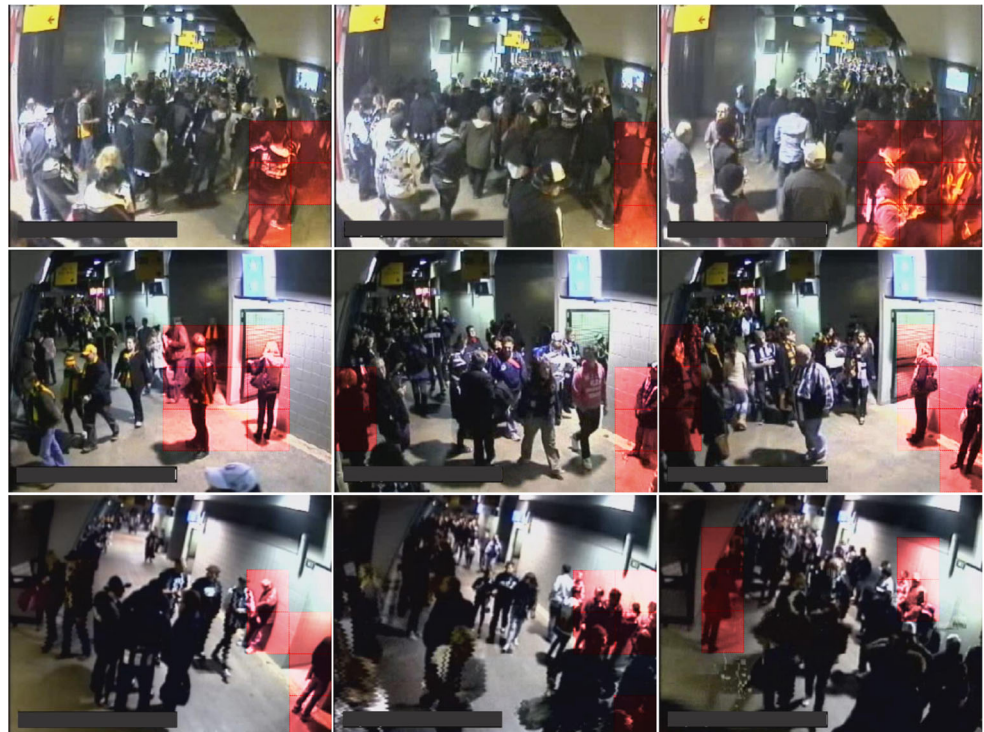
Table 2 The evaluation metric for parameter set

Dataset	N	K	ω
MCG	7.0	6	0.6
UCSD Ped1	7.6	4	0.6
UCSD Ped2	7.6	4	0.6
Avenue	7.4	3	0.6
Subway	7.1	3	0.6

Also, we note that $FNR = 1 - TPR$. The FPR corresponds to the frame identified as anomalous when normal. In contrast, the TPR denotes the number of anomalous frames detected correctly as anomalous. Figure 7(a) shows the frame-level abnormal (loitering) events detected in video frames for Cameras C2, C5 and C6 of the MCG dataset. For the three video sequences (16-Sep-C2, 16-Sep-C5 and 16-Sep-C6), our proposed DeepSDAE has better AUC (86.4%, 74.6% and 79.3%) and lower EER (18.1%, 29.5% and 23.4%) when compared with DBN's AUC (85.4%, 70.1% and 79.1%) and EER (20%, 35% and 23%), indicating an improved frame-level anomaly detection results.

We use AUC and Equal Error Rate (EER) to evaluate this work quantitatively. Table 3 furnishes the performance comparison of DeepSDAE architecture with previous works. Furthermore, we can observe that the proposed architecture outperforms Deep Belief Nets (DBN) [6] when evaluated on the data from three cameras (C2, C5 and C6) from the MCG dataset.

Fig. 6 Sample results obtained by DeepSDAE on the real-world MCG dataset, where the red block denotes the anomalies. Loitering is the most common abnormal pattern in the MCG dataset



4.2.2 Pixel-level abnormal crowd detection

As before, we focus on detecting loitering events in densely crowded scenes. If we identify actual abnormal pixels account for more than 40% of the ground truth in the pixel level comparisons, we consider those pixels as true positive (TP) detection. In contrast, we consider it a false positive (FP) even if a single regular pixel is identified as abnormal. Figure 7(b) shows the ROC for pixel-level anomaly detection, and Table 3 furnishes the quantitative results. Figure 7(b) and Table 3 show that the performance of DeepSDAE is better than anomaly detection presented using DBN architecture [6]. For the three video sequences (16-Sep-C2, 16-Sep-C5 and 16-Sep-C6), our proposed DeepSDAE has better AUC (69%, 68.3% and 73.3%) and lower EER (34.8%, 37.7% and 28.5%) when compared with DBN's AUC (67.6%, 64.1% and 70.4%) and EER (35%, 40.2% and 30.7%), indicating a better anomaly detection results at pixel level.

4.3 UCSD Ped1 and Ped2 dataset

UCSD Ped1 dataset [8] contains pedestrian walkway video sequences. The dataset contains 34 and 36 videos available for training and testing, respectively. Each frame in the video sequences has 238×158 pixel resolutions. UCSD Ped2 dataset [8] contains crowds of people moving parallel to the camera plane. It comprises 16 video sequences (360×240 pixel resolutions) for training and 12 videos

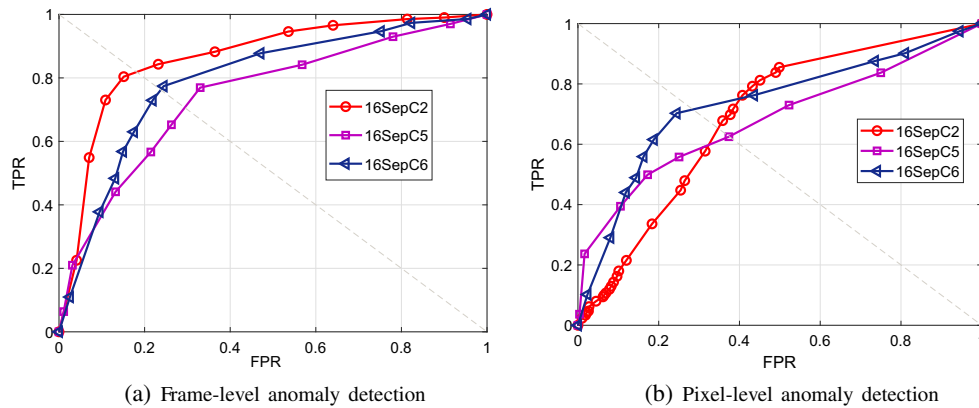


Fig. 7 The evaluation result of MCG dataset: (a) frame-level based abnormal event detection, (b) pixel-level based abnormal event detection

for testing. Because we do not require pre-training, we only use the test sequences of these two datasets. For both UCSD Ped1 and Ped2, the expected behavior includes people walking. The abnormal behavior includes anomalous motion patterns and non-pedestrian objects like cyclists, people with wheelchairs, cars, and skaters. We kept all the model parameters the same as the MCG dataset. Figure 8 shows the sample results we obtained from the proposed SDAE scheme. Figures 9 and 10 show the evaluation results for UCSD Ped1 and Ped2 datasets.

We compared our quantitative results using DeepSDAE with the 13 recently published algorithms. Tables 4 and 5 show the quantitative results on the UCSD Ped1 and UCSD Ped2 datasets. Regarding the result on the UCSD Ped1 dataset, Our proposed DeepSDAE has AUC of 91.5% and 65.7% at detecting frame-level and pixel-level anomalies, outperforming the majority of the existing state-of-the-art approaches. Compared with the Feng [21], TCP [27] and sRNN-AE [38], DeepSDAE is inferior to these methods. Likewise, our approach has lower EER compared to most algorithms.

Regarding the result on the UCSD Ped2 dataset, Our proposed DeepSDAE has AUC of 88.9% and EER of 19.9% at detecting frame-level anomalies, outperforming the existing methods on the UCSD Ped2 dataset by at

least 1%, with lower EER. Especially, when it is compared with the Feng [21], TCP [27] and sRNN-AE [38], where DeepSDAE is inferior to these methods in UCSD Ped1 dataset, DeepSDAE can perform better in this case.

4.4 Subway surveillance and avenue datasets

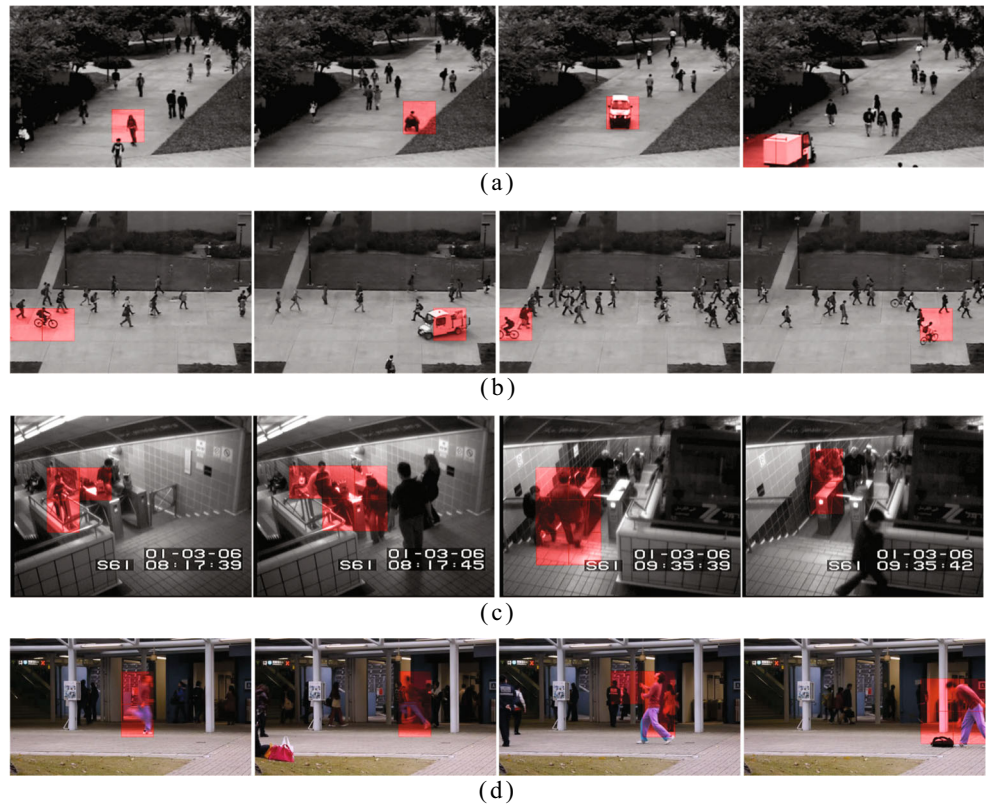
The Subway Surveillance dataset [11] is collected in a subway station, where anomalous behaviors correspond to people moving in the wrong directions. The entire dataset contains two video sequences at entry (144249 frames) and exit (64900 frames) gates. The Avenue Dataset [9] contains 16 training and 21 testing video clips acquired on The Chinese University of Hong Kong (CUHK) campus with a total of 30652 frames (15328 for training and 15324 for testing) [10].

We keep the experimental parameters for all the components unchanged (same as the previous datasets). Figure 8 shows the example results. We evaluated the performance at the frame level for these two datasets because the Subway Surveillance dataset [11] and Avenue [9] does not provide the pixel-level based ground truth. Tables 6 and 7 show the quantitative results compared with other methods.

Table 3 Comparison of performance of deep belief nets (DBN) [6] and the proposed DeepSDAE on MCG dataset (cameras C2, C5 and C6)

Dataset	Approaches	Frame-level anomalies		Pixel-level anomalies	
		AUC	EER	AUC	EER
16-Sep-C2	DBN [6]	85.4%	20.0%	67.6%	35.0%
16-Sep-C5	DBN [6]	70.1%	35.0%	64.1%	40.2%
16-Sep-C6	DBN [6]	79.1%	23.0%	70.4%	30.7%
16-Sep-C2	DeepSDAE	86.4%	18.1%	69.0%	34.8%
16-Sep-C5	DeepSDAE	74.6%	29.5%	68.3%	37.7%
16-Sep-C6	DeepSDAE	79.3%	23.4%	73.3%	28.5%

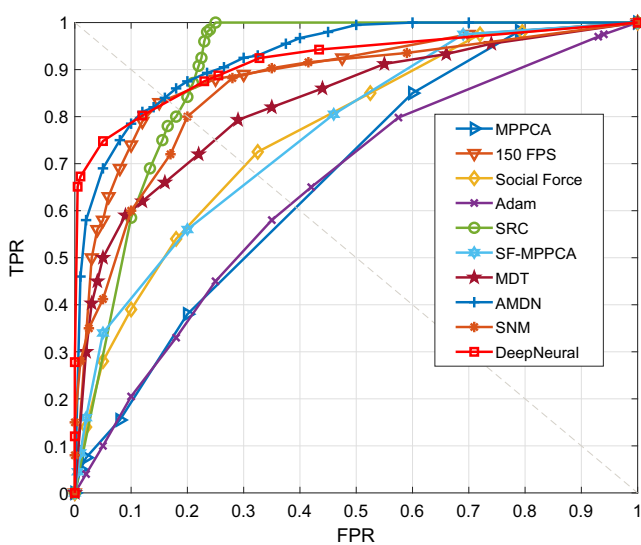
Fig. 8 Sample results of DeepSDAE on three benchmark datasets, where red coloured pixels denoting the detected anomalies in (a) UCSD Ped1 dataset [8], (b) UCSD Ped2 dataset [8], (c) Subway Surveillance dataset [11], and (d) Avenue dataset [9]



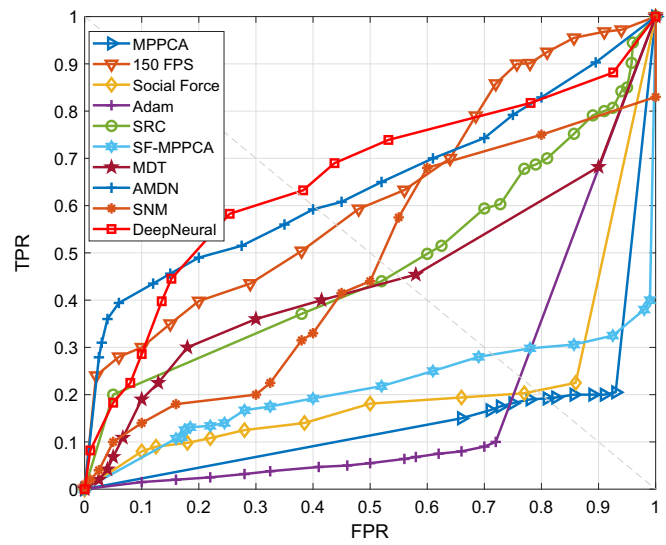
For the Subway Surveillance dataset, we can see that the performance of our approach is second only to the method proposed in [46], and it outperforms the rest of the existing approaches. In addition, our method produces the best performance in terms of EER. Furthermore, our DeepSDAE is also the second-best for the Avenue dataset

compared with the existing approaches. It achieves good performance than most of the past methods.

The reinforcement learning-based parameter setting model helps the DeepSDAE to find the optimal learning parameters. Hence, SDAE combined with the VGG16 optimised model (reduced trainable parameters) speeds up



(a) Frame-level evaluation result



(b) Pixel-level evaluation result

Fig. 9 Evaluation result on UCSD Ped1 dataset: (a) frame-level anomaly detection results, and (b) pixel-level anomaly detection results

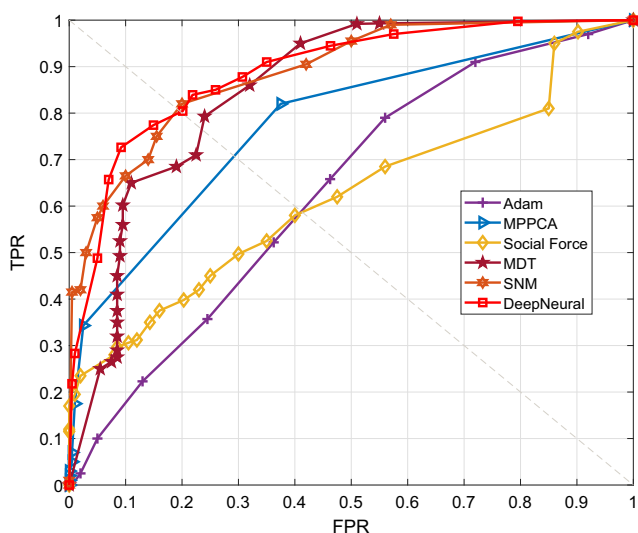


Fig. 10 ROC Curve for frame-level anomaly detection on UCSD Ped2 dataset. Pixel-level anomaly detection for this dataset is not available because of the complexity involved and lack of available comparisons in the literature

the training, and as a result, they converge quickly. Our experimental results confirm this finding and are better than other methods. Deep-Q-Learning provides a further optimisation to improve in learning the parameters. Our

Table 4 Comparison of results of DeepSDAE with other existing algorithms evaluated on the UCSD Ped1 dataset

Approaches	Frame-level		Pixel-level	
	AUC	EER	AUC	EER
MPPCA [39]	67.0%	40.0%	19.0%	44.1%
150 FPS [10]	91.8%	15.0%	63.8%	43.0%
SF [40]	76.8%	31.0%	21.3%	71.0%
Adam [11]	64.9%	38.0%	19.7%	76.0%
SRC [41]	86.0%	19.0%	45.3%	54.0%
SF- MPPCA [17]	76.9%	32.0%	21.3%	71.0%
MDT [17]	81.8%	25.0%	44.1%	58.0%
AMDN [20]	92.1%	16.0%	67.2%	40.1%
Compact Feature Sets [42]	81.8%	21.3%	56.9%	39.5%
Feng et al. [21]	92.8%	11.5%	69.1%	36.3%
Turchini et al. [43]	78.1%	24%	62.2%	37%
TCP [27]	94.6%	9.8%	64.1%	41.2%
SNM [44]	85.5%	20%	47.7%	51.2%
FFP [18]	90.6%	17.1%	62.1%	43.8%
sRNN-AE [38]	92.0%	15.3%	66.9%	37.4%
DeepSDAE	91.5%	16.6%	65.7%	37.5%

The results include frame-level and pixel-level detection results using AUC and EER

experiments suggest that the Deep-Q-Learning improves the overall AUC and ROC in detecting anomalies.

The proposed framework can perform well in detecting movement anomalies, but it cannot be used as a general approach in diverse types of anomaly detection. Our future work will include exploring late fusion mechanisms to different anomaly detection scenarios and generalising the framework for other crowd settings. In addition, we will explore whether we can have a further optimised VGG-16 model or have a lightweight model (such as MobileNets [48]) to improve overall computational time for real-time video surveillance applications.

5 Conclusion

We proposed an end-to-end deep learning model called DeepSDAE to detect anomalous crowd behavior. This framework is a hybrid deep learning architecture comprising motion maps, a four-layer SDAE, an optimised VGG16 model, and a PSVM. Our framework uses a late fusion mechanism to combine decisions from PSVM channels for detecting crowd anomalies. We derive optimal parameters by modelling the crowd flow process as the MDP and solving them using Deep Q-learning. DeepSDAE is a novel

Table 5 Comparison of results of DeepSDAE with other existing algorithms evaluated on the UCSD Ped2 dataset

Approaches	Frame-level	
	AUC	EER
Adam [11]	63.0%	40.8%
MPPCA [39]	77.0%	29.8%
SF [40]	63.0%	40.5%
MDT [17]	85.0%	23.5%
SF- MPPCA [17]	71.0%	36.0%
ConvLSTM-AE [22]	88.10%	–
SNM [44]	87.9%	20%
Compact Feature Sets [42]	84%	19.4%
Turchini et al. [43]	80.5%	19.3%
Feng et al. [21]	88.5%	16.9%
TCP [27]	88.1%	18.3%
FFP [18]	87.4%	22.9%
sRNN-AE [38]	88.3%	18.7%
DeepSDAE	88.9%	19.9%

The results include frame-level anomaly detection using AUC and EER

Reinforcement learning-Deep learning model to detect crowd anomalies, where RL is firstly introduced to explore the optimal parameter set. The experimental evaluation on four datasets (MCG, UCSD Ped1 and Ped2, Avenue, and Subway Surveillance) show that our proposed DeepSDAE

Table 6 Comparison of results of DeepSDAE with other existing algorithms evaluated on the Subway dataset

Approaches	Entrance		Exit	
	AUC	EER	AUC	EER
SRC [41]	83.3%	24.4%	80.2%	26.4%
MDT [17]	90.8%	16.7%	89.7%	16.4%
FCNs [26]	90.1%	17.4%	89.7%	16.2%
LSTM-AE [45]	93.3%	–	87.7%	–
NMC [46]	91.8%	–	94.2%	–
DeepSDAE	90.9%	16.5%	91.1%	15.2%

The evaluation results include anomaly detection results at entry and exit of the Subway Surveillance dataset [11]

Table 7 Comparison of results of DeepSDAE with other existing algorithms evaluated on the avenue dataset

Approaches	AUC	EER
sRNN [19]	81.7%	– %
150 FPS [10]	80.3%	27.5%
CAE [22]	70.2%	25.1%
FFP [18]	84.9%	–
DAF [47]	84.6%	–
AMDN [20]	78.0%	26.6%
LSTM-AE [45]	75.33%	–
ConvLSTM-AE [22]	77.0%	–
NMC [46]	88.9%	–
sRNN-AE [38]	83.1%	–
DeepSDAE	87.3%	20.3%

The results include anomaly detection frame-level anomaly detection in terms of AUC and EER

surpasses existing approaches in detecting anomalies (frame level or pixel level) in crowded scenes.

Acknowledgments The authors are very grateful to Editor and the anonymous reviewers for their valuable comments and suggestions that improved the presentation and quality of this paper highly. This work was supported by the Natural Science Foundation of China under Grants 12201523, and also supported by the Fundamental Research Funds for the Central Universities under Grants No. 2682021CX078.

References

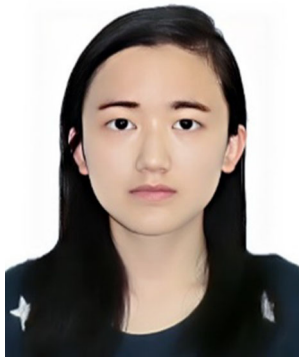
1. Varadarajan J, Odobez J-M (2009) Topic models for scene analysis and abnormality detection. In: 2009 IEEE 12th international conference on computer vision workshops, pp 1338–1345
2. Luff P, Heath C, Jirotko M (2000) Surveying the scene: technologies for everyday awareness and monitoring in control rooms. *Interact Comput* 13(2):193–228
3. Aggarwal JK, Cai Q (1999) Human motion analysis: a review. *Comput Vis Image Underst* 73(3):428–440
4. Krüger V, Kragic D, Ude A, Geib C (2007) The meaning of action: a review on action recognition and mapping. *Adv Robot* 21(13):1473–1501
5. Rao AS, Gubbi J, Rajasegarar S, Marusic S, Palaniswami M (2014) Detection of anomalous crowd behaviour using hyperspherical clustering. In: 2014 International conference on digital image computing: techniques and applications (DICTA), pp 1–8
6. Yang M, Rajasegarar S, Erfani SM, Leckie C (2019) Deep learning and one-class svm based anomalous crowd detection. In: 2019 International joint conference on neural networks (IJCNN). IEEE, pp 1–8

7. Erfani SM, Rajasegarar S, Karunasekera S, Leckie C (2016) High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recogn* 58:121–134
8. (2013). UCSD anomaly detection dataset. <http://www.svcl.ucsd.edu/projects/anomaly/dataset.html>. Last Accessed 26 Feb 2022
9. (2013). Avenue dataset for abnormal event detection. <http://www.cse.cuhk.edu.hk/leojia/projects/detectabnormal/dataset.html>. Last Accessed 26 Feb 2022
10. Lu C, Shi J, Jia J (2013) Abnormal event detection at 150 fps in matlab. In: *ICCV*, pp 2720–2727
11. Adam A, Rivlin E, Shimshoni I, Reinitz D (2008) Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans Pattern Anal Mach Intell* 30(3):555–560
12. Rao AS, Gubbi J, Marusic S, Palaniswami M (2015) Estimation of crowd density by clustering motion cues. *Vis Comput* 31(11):1533–1552
13. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533
14. Mo X, Monga V, Bala R, Fan Z (2014) Adaptive sparse representations for video anomaly detection. *IEEE Trans Circuits Syst Video Technol* 24(4):631–645
15. Bird N, Atev S, Caramelli N, Martin R, Masoud O, Papanikolopoulos N (2006) Real time, online detection of abandoned objects in public areas. In: *ICRA 2006*. IEEE, pp 3775–3780
16. Mohammadi S, Perina A, Kiani H, Murino V (2016) Angry crowds: detecting violent events in videos. In: *European conference on computer vision*. Springer, pp 3–18
17. Mahadevan V, Li W, Bhalodia V, Vasconcelos N (2010) Anomaly detection in crowded scenes. In: *2010 IEEE computer society conference on computer vision and pattern recognition*, pp 1975–1981
18. Liu W, Luo W, Lian D, Gao S (2018) Future frame prediction for anomaly detection—a new baseline. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 6536–6545
19. Luo W, Liu W, Gao S (2017) A revisit of sparse coding based anomaly detection in stacked RNN framework. In: *Proceedings of the IEEE international conference on computer vision*, pp 341–349
20. Xu D, Ricci E, Yan Y, Song J, Sebe N (2015) Learning deep representations of appearance and motion for anomalous event detection. [arXiv:1510.01553](https://arxiv.org/abs/1510.01553)
21. Feng Y, Yuan Y, Lu X (2016) Deep representation for abnormal event detection in crowded scenes. In: *2016 ACM on multimedia conference*, pp 591–595
22. Hasan M, Choi J, Neumann J, Roy-Chowdhury AK, Davis LS (2016) Learning temporal regularity in video sequences. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 733–742
23. Chong YS, Tay YH (2017) Abnormal event detection in videos using spatiotemporal autoencoder. In: *International symposium on neural networks*. Springer, pp 189–196
24. Dubey S, Boragule A, Gwak J, Jeon M (2021) Anomalous event recognition in videos based on joint learning of motion and appearance with multiple ranking measures. *Appl Sci* 11(3):1344
25. Morais R, Le V, Tran T, Saha B, Mansour M, Venkatesh S (2019) Learning regularity in skeleton trajectories for anomaly detection in videos. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 11996–12004
26. Sabokrou M, Fayyaz M, Fathy M, Moayed Z, Klette R (2018) Deep-anomaly: fully convolutional neural network for fast anomaly detection in crowded scenes. *Comput Vis Image Underst* 172:88–97
27. Ravanbakhsh M, Nabi M, Mousavi H, Sangineto E, Sebe N (2018) 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)*, pp 1689–1698
28. Lu X, Wang W, Ma C, Shen J, Shao L, Porikli F (2019) See more, know more: unsupervised video object segmentation with co-attention siamese networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 3623–3632
29. Lu X, Wang W, Shen J, Crandall D, Luo J (2020) Zero-shot video object segmentation with co-attention siamese networks. *IEEE transactions on pattern analysis and machine intelligence*
30. Mishra SR, Mishra TK, Sarkar A, Sanyal G (2020) Detection of anomalies in human action using optical flow and gradient tensor. In: *Smart intelligent computing and applications*. Springer, pp 561–570
31. Mishra SR, Mishra TK, Sanyal G, Sarkar A, Satapathy SC (2020) Real time human action recognition using triggered frame extraction and a typical cnn heuristic. *Pattern Recogn Lett* 135:329–336
32. Jafari MH, Luong C, Tsang M, Gu AN, Van Woudenberg N, Rohling R, Tsang T, Abolmaesumi P (2021) U-land: uncertainty-driven video landmark detection. *IEEE Trans Med Imaging* 41(4):793–804
33. Shao J, Loy CC, Wang X (2016) Learning scene-independent group descriptors for crowd understanding. *IEEE Trans Circuits Syst Video Technol* 27(6):1290–1303
34. Ghafoori Z, Rajasegarar S, Erfani SM, Karunasekera S, Leckie CA (2016) Unsupervised parameter estimation for one-class support vector machines. In: *Pacific-asia conference on knowledge discovery and data mining*. Springer, pp 183–195
35. Snoek CG, Worring M, Smeulders AW (2005) Early versus late fusion in semantic video analysis. In: *Proceedings of the 13th annual ACM international conference on multimedia*, pp 399–402
36. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1–9
37. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
38. Luo W, Liu W, Lian D, Tang J, Duan L, Peng X, Gao S (2019) Video anomaly detection with sparse coding inspired deep neural networks. *IEEE Trans Pattern Anal Mach Intell* 43(3):1070–1084
39. Kim J, Grauman K (2009) Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In: *2009 IEEE conference on computer vision and pattern recognition*, pp 2921–2928
40. Reddy V, Sanderson C, Lovell BC (2011) Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In: *CVPRW*. IEEE, pp 55–61
41. Cong Y, Yuan J, Liu J (2011) Sparse reconstruction cost for abnormal event detection. In: *CVPR*. IEEE, pp 3449–3456
42. Leyva R, Sanchez V, Li C-T (2017) Video anomaly detection with compact feature sets for online performance. *IEEE Trans Image Process* 26(7):3463–3478
43. Turchini F, Seidenari L, Bimbo AD (2017) Convex polytope ensembles for spatio-temporal anomaly detection. In: *International conference on image analysis and processing*. Springer, pp 174–184
44. Chaker R, Al Aghbari Z, Junejo IN (2017) Social network model for crowd anomaly detection and localization. *Pattern Recogn* 61:266–281
45. Luo W, Liu W, Gao S (2017) Remembering history with convolutional lstm for anomaly detection. In: *2017 IEEE international*

- conference on multimedia and expo (ICME). IEEE, pp 439–444
46. Ionescu RT, Smeureanu S, Popescu M, Alexe B (2018) Detecting abnormal events in video using narrowed motion clusters. arXiv:1801.05030
47. Smeureanu S, Ionescu RT, Popescu M, Alexe B (2017) Deep appearance features for abnormal behavior detection in video. In: International conference on image analysis and processing. Springer, pp 779–789
48. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Meng Yang received the Ph.D degree in Artificial Intelligence from University of Melbourne, Australia, in 2019. Since 2019, she has been with the School of Mathematics, Southwest Jiaotong University, where she is currently an associate professor. Her research interests include machine learning, reinforcement learning, data mining and computer vision.



Shucong Tian received the B.S. degree in mathematics from China West Normal University, Nanchong, China, in 2019. And he received the M.S. degree in mathematics from Southwest Jiaotong University, Chengdu, China, in 2022. He is currently pursuing the Ph.D. degree in information security from Southwest Jiaotong University. His research interests include reinforcement learning and deep learning.



Aravinda S. Rao is a Research Fellow in the Department of Electrical and Electronic Engineering, The University of Melbourne, Australia. He is also the Theme Coordinator of Internet of Things Sensors (Construction Tech) at Building 4.0 CRC and a Senior Member of the IEEE. His research interests include Computer Vision, Machine Learning, Deep Learning, Internet of Things (IoT).



Sutharshan Rajasegarar received his Ph.D. from The University of Melbourne, Australia. He is currently a Senior Lecturer with the School of Information Technology, Deakin University, Burwood, Australia. He previously worked as a Research Fellow at the Department of Electrical and Electronics Engineering, The University of Melbourne, as a researcher in Machine Learning with the National ICT Australia (NICTA/Data61), and as a

visiting researcher at University of Surrey, UK. His current research interests include anomaly/outlier detection, distributed machine learning, Artificial Intelligence (AI), signal processing, health analytics, computer vision, wireless communications, cyber security, sports analytics and Internet of Things (IoT).




Marimuthu Palaniswami is a Fellow of the Institute of Electrical and Electronic Engineering (IEEE) and an internationally recognised expert in Internet of Things (IoT), Sensor Networks, Automated Learning, and Computational Intelligence in large-scale complex systems. He is a named Distinguished Lecturer of the IEEE Computational Intelligence Society over the period 2013–2015.



Zhengchun Zhou received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in information security from Southwest Jiaotong University, Chengdu, China, in 2001, 2004, and 2010, respectively. From 2012 to 2013, he was a Postdoctoral Member with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. From 2013 to 2014, he was a Research Associate with the Department of Computer

Science and Engineering, the Hong Kong University of Science and Technology. Since 2001, he has been with the Department of Mathematics, Southwest Jiaotong University, where he is currently a Professor. His research interests include sequence design and coding theory. Dr. Zhou was the recipient of the National Excellent Doctoral Dissertation Award in 2013 (China).

Affiliations

Meng Yang¹ · Shucong Tian¹ · Aravinda S. Rao² · Sutharshan Rajasegarar³ · Marimuthu Palaniswami² · Zhengchun Zhou¹ 

Meng Yang
yangmeng@swjtu.edu.cn

Shucong Tian
sctian@my.swjtu.edu.cn

Aravinda S. Rao
aravinda.rao@unimelb.edu.au

Sutharshan Rajasegarar
srajas@deakin.edu.au

Marimuthu Palaniswami
palani@unimelb.edu.au

- ¹ School of Mathematics, Southwest Jiaotong University, Chengdu, 610000, China
- ² Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, 3010, VIC, Australia
- ³ School of IT, Deakin University, Geelong, 3125, VIC, Australia