Engineering 5 (2019) 930–939

Contents lists available at ScienceDirect

Engineering

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

Research
Artificial Intelligence—Article

AED-Net: An Abnormal Event Detection Network

Tian Wang [a], Zichen Miao [a], Yuxin Chen [a], Yi Zhou [b], Guangcun Shan [a,c,*], Hichem Snoussi [d]

[a] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[b] Department of Electronic Engineering, Dalian Maritime University, Dalian 116026, China
[c] School of Instrumentation Science and Opto-electronic Engineering & International Research Institute for Multidisciplinary Science, Beihang University, Beijing 100191, China
[d] Institute Charles Delaunay-LM2S-UMR STMR 6281 CNRS, University of Technology of Troyes, Troyes 10010, France

ARTICLE INFO

Article history:
Received 22 May 2018
Revised 1 February 2019
Accepted 25 February 2019
Available online 25 May 2019

Keywords:
Abnormal events detection
Abnormal event detection network
Principal component analysis network
Kernel principal component analysis

ABSTRACT

It has long been a challenging task to detect an anomaly in a crowded scene. In this paper, a self-supervised framework called the abnormal event detection network (AED-Net), which is composed of a principal component analysis network (PCAnet) and kernel principal component analysis (kPCA), is proposed to address this problem. Using surveillance video sequences of different scenes as raw data, the PCAnet is trained to extract high-level semantics of the crowd's situation. Next, kPCA, a one-class classifier, is trained to identify anomalies within the scene. In contrast to some prevailing deep learning methods, this framework is completely self-supervised because it utilizes only video sequences of a normal situation. Experiments in global and local abnormal event detection are carried out on Monitoring Human Activity dataset from University of Minnesota (UMN dataset) and Anomaly Detection dataset from University of California, San Diego (UCSD dataset), and competitive results that yield a better equal error rate (EER) and area under curve (AUC) than other state-of-the-art methods are observed. Furthermore, by adding a local response normalization (LRN) layer, we propose an improvement to the original AED-Net. The results demonstrate that this proposed version performs better by promoting the framework's generalization capacity.

1. Introduction

Video studies have attracted an increasing amount of attention from researchers in the computer vision community in recent years. Lately, research into topics such as object tracking [1–3], gait recognition [4,5], and activity recognition [6–8] have achieved competitive results and demonstrated promise for the future.

Abnormal event detection, which involves detecting the specific frames in a video that contains an anomaly, is one of the hottest research issues in the video field. In comparison with the tasks mentioned above, abnormal event detection has greater significance for national security and for people's lives with the modernization of society, an increasing number of surveillance cameras are being deployed in various places, producing an enormous quantity of video every second. This much data is impossible for a human to deal with, or to determine any abnormal events contained within. However, missing even one anomaly in a surveillance video could result in unbearable loss. Thus, there is a need for the construction of an automatic video abnormality detector that can deal with millions of videos frames and can alert people in order to enable a timely and effective response when an anomaly happens.

Ref. [9] describes the many difficulties inherent in anomaly detection. Although it is simple enough to list a few types of abnormalities for a specific scene, such as the presence of a car or a bicyclist in a pedestrian crowd, it is impractical to enumerate all the abnormal events that could be possible within that scene; thus, there are countless positive classes in this classification task. Furthermore, due to a lack of abnormal samples—that is, video frames including abnormal events—the training set is severely imbalanced, making it infeasible to train a model for multi-class classification. All these difficulties indicate that the anomaly detection task is a one-class classification problem that is hard to handle.

Some methods have been suggested to deal with abnormal event detection. For example, Ref. [10] proposes a method based on histograms of the optical flow orientation descriptor. As the handcrafted feature descriptor in this case was constructed based on human experience, it did not represent the feature in a training process. Thus, it performs worse than current deep learning

* Corresponding author.
E-mail address: gcshan@buaa.edu.cn (G. Shan).

https://doi.org/10.1016/j.eng.2019.02.008

methods. Deep learning methods have been recently developed, as described in Refs. [11,12], largely due to the availability of big data and efficient hardware. Such methods are intensively applied in the computer vision field and have achieved great results. In Ref. [13], Wang et al. used a convolution neural network (CNN) for defect detection in product quality control. However, the original CNN for face recognition is not applicable to this task, because its training requires samples of different classes. Considering the success of the principal component analysis network (PCAnet) [14] in image classification, Ref. [15] proposes a PCAnet-based method to extract information from raw images for anomaly detection, with a one-class classifier constructed on a clustering algorithm. However, this method has natural limitations due to the K-medoids clustering algorithm, which has difficulty dealing with the high-dimensional features extracted by the PCAnet. In this paper, we propose a self-supervised network, termed the abnormal event detection network (AED-Net), to deal with the task of video anomaly detection, with only normal samples being provided as training data. Since PCAnet has been demonstrated as being able to extract features as an unsupervised model, it was chosen for our self-supervised AED-Net. In addition, a one-class classifier is used to handle the extracted high-dimensional features in order to determine the abnormality of frames.

To be more specific, this new self-supervised network uses optical flow maps as the input, because these maps are well-suited to representing motion. Next, high-level semantics of a crowd's situation can be extracted from the PCAnet. Subsequently, a simple but effective one-classifier kernel principal component analysis (kPCA) [16] is used to classify the high-dimensional features. Having the advantages of both networks, AED-Net is trained to understand each frame and conduct detection. More importantly, a local response normalization (LRN) layer (a technique used in CNN to aid generalization) is incorporated to improve the AED-Net. It is worth noting that this new network can be trained with unlabeled data and performs better in comparison with state-of-the-art methods in an abnormal detection task. Our new self-supervised network can effectively detect abnormal events even in crowded situations, which improves the detection results according to the experiments tested on the public Monitoring Human Activity dataset from University of Minnesota (UMN dataset) and Anomaly Detection dataset from University of California, San Diego (UCSD dataset).

The rest of this paper is organized as follows. Section 2 provides a brief review of related works, and Section 3 reviews the basic algorithms of our framework, the PCAnet and kPCA. Next, the complete architecture of AED-Net is elaborated in Section 4, along with our improvements to it. Section 5 illustrates and discusses the experimental results of the UMN [17] and UCSD [18] datasets. Finally, Section 6 presents our conclusions.

2. Related work

In general, traditional methods for anomaly detection can be divided into two major classes. The first class is based on trajectory, and has been widely used in abnormal events detection [19–22]. In Refs. [23–25], the authors extract the trajectory of normal events to indicate normal modes; trajectories that differ from the normal patterns are then considered abnormal. However, occlusion between moving objects affects the effectiveness of this method when it is applied to crowded scenes. To tackle this problem, a new model is suggested in Ref. [26] to deal with the interrelatedness of human behavior and to ameliorate the representation of objects' interaction. In Ref. [27], a discrete transformation is utilized to develop a reliable multi-target tracking algorithm that associates objects in different frames. However, the occlusion problem affects the results so much that the methods listed above do not address this issue in an effective way. Hence, the tracking strategy is not adopted in our work.

The spatiotemporal method falls into another category. Promising research on this method has been proposed. In Ref. [28], Wang et al. propose a covariance matrix as a feature descriptor, which encodes the optical flow and partial derivatives of adjacent frames. In Refs. [29–32], the authors model motion patterns with histograms of pixel changes. In Refs. [33–35], distributions of optical flow are used as the basic features, and models for detecting abnormal events are then built based on optic flow features. Ref. [36] proposes an approach to estimate the interaction between moving objects. Another study [9] uses a detector that combines time and space anomalies. The wavelet transform used in image processing can also be utilized to analyze motion [37,38]. In those cases delicate feature descriptors were designed manually, and tended to work well only under specified conditions. In our work, the features are extracted by a self-supervised network.

With its rapid development, deep learning has recently achieved outstanding results in the field of abnormal event detection. Unlike the features of manual design, features extracted by a deep learning network are obtained through a learning process. In the proposed AED-Net, a self-supervised learning method is proposed for abnormal event detection, which involves only normal samples being learned.

3. Self-supervised feature extraction and anomaly detection

Self-supervised learning is a learning paradigm in which there is no external supervised information—that is, labels—as ground truth beyond the data itself. Under this paradigm, the self-supervised learning method simply adopts the raw data as the material for training, which means that the model learns to extract latent supervised information in the data. Data categories are not employed in the training process.

The self-learning model is applicable to the anomaly detection task. Since we can only use normal data to train the model, no external supervised information is given to the model. Thus, the model must fully understand what a normal datum is from the input video clips, and then use it as supervised information to tune its parameters. Table 1 introduces the notations used in this paper.

3.1. PCAnet for feature extraction

Both traditional and deep learning methods have been applied for feature extraction from video frames. In Ref. [10], the global optical flow descriptor is used as the feature. However, optical flow only contains low-level motion information in the frames; high-level information features such as people's running pattern, or how many people are in the frame, cannot be represented by it. Thus, the deep learning method is used to deal with this high-level feature extraction problem. The most popular model is CNN, which stack layers and extract deeper and deeper features step by step. However, that particular model requires strong external supervised information, which is not provided in our task. Thus, we chose the PCAnet [14], an equivalent model in feature extraction that utilizes the power of deep learning without requiring external supervised information.

PCAnet [14] is a deep learning network that has been proposed within the prevailing trend of deep learning. Although it is simple in comparison with other popular deep learning networks, such as the deep CNN, PCAnet is capable enough to handle challenging tasks such as face recognition. Thus, this model was chosen for its efficiency and competitive ability in feature extraction.

Table 1
A description of the notations used in this paper.

Variable	Description
S	Raw surveillance video frame
I	Input of PCAnet (optical flow map)
$k_1,\ k_2$	Size of patches in PCAnet
X	Matrix consisting of all patches from an optical flow map at Stage 1 of PCAnet
S_i	Matrix containing all matrices $\bar{X}$ at the ith stage of PCAnet
K_j^i	jth filter of ith stage of PCAnet
C	Outputs of Stage 1 of PCAnet
Y	Matrix consisting of all patches from an optical flow map at Stage 1 of PCAnet
O	Outputs of Stage 2 of PCAnet
T	Integer-valued image after binarizing and encoding outputs of Stage 2
$\mathcal{F}$	Final feature of PCAnet
F	Inputs feature of kPCA classifier
$\mathcal{M}(F_i)$	Feature F mapped into higher-dimensional space
$\kappa(F_i,\ F_j)$	Scalar product of $\mathcal{M}(F_i)$ and $\mathcal{M}(F_j)$
W_j	Eigenvectors of covariance matrix of mapped feature in higher-dimensional space
V	Kernel matrix, $V_{i,j} = \kappa(F_i,\ F_j)$
$\bar{V}$	Kernel matrix, $\bar{V}_{i,j}$, which is the scalar product of $\bar{\mathcal{M}}(I_i)$ and $\bar{\mathcal{M}}(I_j)$
R	Reconstruction error, i.e., abnormality score
α	Eigenvectors of kernel matrix $\bar{V}$
p_i	Output value on the ith feature map
q_i	Normalized output of p_i
$\delta,\ n,\ \beta,\ \gamma$	Hyperparameters of LRN

PCAnet is a cascaded linear network. A typical two-stage PCAnet architecture is shown in Fig. 1. Because it is inspired by CNN, each stage of PCAnet consists of an independent principal component analysis (PCA) filter bank that must be learned in order to perform feature extraction work. Feature maps in the first stage are linearly cascaded to the next stage to extract higher-level features. As discussed by Chan et al. [14], the performance corresponding to the number of stages shows that although a two-stage network performs better than a one-stage network, networks with more than two stages have few advantages over a two-stage network; therefore, a two-stage PCAnet is sufficient for the task at hand for the benefit of computation efficiency.

A two-stage PCAnet was therefore used to extract features. In the training phase, at the beginning of Stage 1, an optical flow map I_i with the shape $h \times w$ is sampled around each pixel to small patches sized $k_1 \times k_2$, as shown in Fig. 1 with the upper gray arrows. Next, the samples, $\text{patch}(x_1), \text{patch}(x_2), ..., \text{patch}\left[x_{(h-k_1+1)\times(w-k_2+1)}\right]$, are vectorized and compose sample matrix $X = \left[x_1, x_2, x_{(h-k_1+1)\times(w-k_2+1)}\right]$. We then perform mean subtraction to X to obtain $\bar{X}$. (See Table 1 for a list of all the notations used in this paper.)

For N input optical flow maps, $I = \{I_1, I_2, ..., I_N\}$, PCAnet initially samples them to obtain the following:

$$S_1 = \left[\bar{X}_1, \bar{X}_2, ..., \bar{X}_N\right] \in \mathbb{R}^{k_1 k_2 \times N(h-k_1+1)\times(w-k_2+1)} \tag{1}$$

Next, PCAnet computes L_1 convolution kernels based on I by implementing PCA, as shown by the lower gray arrow in Fig. 1, to obtain the following:

$$K_l^1 = \text{vec2mat}_{k_1,k_2}\left[s_l\left(S_1 S_1^{\mathrm{T}}\right)\right] \in \mathbb{R}^{k_1 \times k_2},\ l = 1, 2, ..., L_1 \tag{2}$$

where $s_l\left(S_1 S_1^{\mathrm{T}}\right)$ denotes the lth principal eigenvector of $S_1 S_1^{\mathrm{T}}$, and vec2mat$(\cdot)$ maps a vector from $\mathbb{R}^{k_1 k_2}$ to a matrix $M \in \mathbb{R}^{k_1 \times k_2}$. At the end of Stage 1, the convolution operation is performed to extract features:

$$C_i^l = I_i * K_l^1,\ i = 1, 2, ..., N \tag{3}$$

where $*$ denotes two-dimensional (2D) convolution, C_i^l refers to the lth feature map of the ith input I_i, and the number of outputs in Stage 1 is $L_1 N$. Note that the boundary of I_i is zero-padded in order to ensure that the outputs have the same size as the input—that is, $h \times w$. As implied, in the test phase, PCAnet will directly perform a convolution operation on inputs I using kernels obtained from the training phase.

Stage 2 is conducted in almost the same way as Stage 1. In the training phase, each input C_i^l of C is sampled to patches. These patches are vectorized and compose matrix S_2 after mean subtraction is performed:

$$S_2 = \left[\bar{Y}_1^1, ..., \bar{Y}_1^{L_1}, \bar{Y}_2^1, ..., \bar{Y}_2^{L_1}, ..., \bar{Y}_N^1, ..., \bar{Y}_N^{L_1}\right] \in \mathbb{R}^{k_1 k_2 \times L_1 N(h-k_1+1)\times(w-k_2+1)} \tag{4}$$

where $\bar{Y}_j^l = \left[\bar{y}_{i,l,1}, \bar{y}_{i,l,2}, ..., \bar{y}_{i,l,(h-k_1+1)\times(w-k_2+1)}\right] \in \mathbb{R}^{k_1 k_2 \times (h-k_1+1)\times(w-k_2+1)}$ refers to the sample matrix of C_i^l. We then compute convolution kernels in Stage 2:

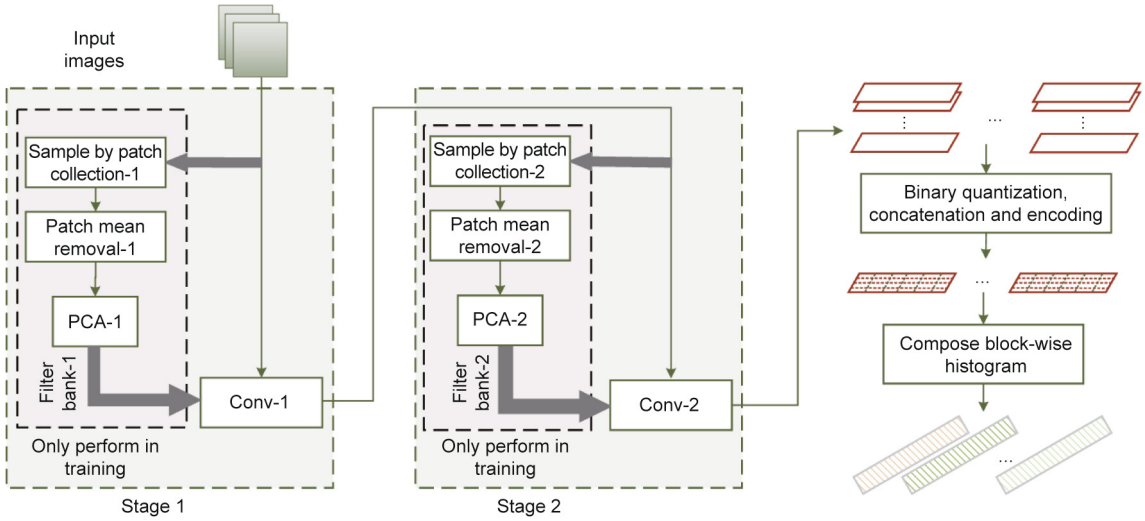

Fig. 1. Typical structure of the two-stage PCAnet used in our method. Conv: convolution.

$$K_m^2 = \text{vec2mat}_{k_1,k_2}\left[s_m\left(S_1 S_1^{\mathrm{T}}\right)\right] \in \mathbb{R}^{k_1 \times k_2},\ m = 1, 2, ..., L_2 \tag{5}$$

Finally, we obtain the outputs of Stage 2 by convolution:

$$O_i^{l,m} = C_i^l * K_m^2,\quad l = 1, 2, ..., L_1,\ m = 1, 2, ..., L_2,\ i = 1, 2, ..., N \tag{6}$$

The number of outputs in Stage 2 is $L_2 L_1 N$.

After Stage 2, we binarize the output by the Heaviside step function $H(\cdot)$, assigning 1 for positive entries and 0 for zero or negative entries. This enables the network to have nonlinearity. Thus, the network is capable of capturing high-level semantics in the optical flow maps. Each of these L_1 inputs of the second stage C_i^l has L_2 real-valued outputs in the second stage $O_i^{l,m}(m = 1, 2, ..., L_2)$. Around each pixel, there are L_2 binary bits; we can view them as a decimal number, converting the L_2 outputs $O_i^{l,m}$ to a single integer-valued image:

$$T_i^l = \sum_{m=1}^{L_2} 2^{m-1} H\left(O_i^{l,m}\right) \tag{7}$$

Finally, the output features of PCAnet are block histograms (with 2^{L_2} bins) computed based on all T_i^l. Note that one histogram does not represent the whole T_i^l, but a region of it. To do this, T_i^l is partitioned into B blocks and then used to calculate the histogram. A histogram is computed in each block. Next, all the histograms are concatenated into one vector, $\text{Bhist}\left(T_i^l\right)$. For single-input optical image I_i, the feature is as follows:

$$\mathcal{F}_i = \left[\text{Bhist}\left(T_i^1\right), ..., \text{Bhist}\left(T_i^{L_1}\right)\right]^{\mathrm{T}} \in \mathbb{R}^{(2^{L_2})L_1 B} \tag{8}$$

The local block can be either overlapping or non-overlapping. The latter setting is beneficial for detection except for face detection [14], so it is set to non-overlapping in this paper. Besides the overlapping choice, the hyperparameters of the PCAnet also include the filter size k_1, k_2, the number of filters in each stage $L_1 L_2$, and the block size for local histograms.

3.2. A self-supervised learning method for anomaly detection: kPCA

Because we can only utilize video sequence of normal scenes, and it is necessary to distinguish normal frames from abnormal frames with previously unknown anomalies, it is appropriate to class this task as one-class classification.

The common idea in one-class classification task is to train a classifier that encloses the training data—that is, the normal data—and thereby separate the abnormal data from the normal data. The support vector domain description (SVDD) classifier is a good example of this method. However, this classifier often generates a too-large decision boundary that hinders good performance. Using Gaussian process priors, Kemmler et al. [39] built a model for one-class classification that uses different measures derived from Gaussian process regression and approximate Gaussian classification. However, this model strongly relies on hyperparameter tuning of the re-parameterized kernel function.

In contrast, by learning the distribution of data, which is usually nonlinear, a kPCA classifier [16] can generate a decision boundary smoothly following the distribution of data, and tends to classify more accurately.

The structure of a kPCA classifier is shown in Fig. 2. The essential idea of this one-class classifier is that the features of normal frames have a similar distribution, while the features of abnormal frames have a very different distribution. Thus, after using PCA filters that were computed based on training features—that is, normal features—in order to perform PCA on both normal features and abnormal features, we were able to observe a clear difference in reconstruction error between normal features and abnormal features. The classification could then be conducted according to this disparity.

As discussed by Hoffmann [16], PCA cannot capture the nonlinear structure of input. Hence, kPCA is introduced to overcome this drawback, as it maps input $F_i \in \mathbb{R}^d$ to feature in higher-dimensional space: $\mathcal{M}(F_i) \in \mathbb{R}^n$ $(n > d)$. PCA is then performed in the feature space. Computation here only requires the scalar product of $\mathcal{M}(F_i)$—that is, $[\mathcal{M}(F_i) \cdot \mathcal{M}(F_b)]$. The scalar product is further replaced by the kernel function $\kappa(F_i, F_j)$ to perform the same task. Here, the kernel function uses the Gaussian kernel $\kappa(F_i, F_j) = \exp\left(-\frac{\|F_i - F_j\|^2}{2\sigma^2}\right)$. Furthermore, we obtain $\bar{\mathcal{M}}(F_i)$ from $\mathcal{M}(F_i)$ by performing mean subtraction, which can further represent W_j, the eigenvectors of the covariance matrix in higher-dimensional space. Thus, W_j can be expressed by $\mathcal{M}(F_i)$ as follows:

$$W_j = \sum_{i=1}^{N} \alpha_i^j \left[\mathcal{M}(F_i) - \frac{1}{N}\sum_{k=1}^{N} \mathcal{M}(F_k)\right] \tag{9}$$

It turns out that α^j, where $\alpha^j = \left[\alpha_1^j, \alpha_2^j, ..., \alpha_N^j\right]$, $j = 1, 2, ..., q$, is an eigenvalue of kernel matrix $\bar{V}$. Each component of $\bar{V}$—that is, $\bar{V}_{ij}$—is a scalar product of $\bar{\mathcal{M}}(F_i)$ and $\bar{\mathcal{M}}(F_j)$. Similarly, each component of kernel matrix V—that is, V_{ij}—is a scalar product of $\mathcal{M}(F_i)$ and $\mathcal{M}(F_j)$. Thus,

$$\bar{V}_{ij} = V_{ij} - \frac{1}{N}\sum_{a=1}^{N} V_{ia} - \frac{1}{N}\sum_{a=1}^{N} V_{aj} + \frac{1}{N^2}\sum_{a,b=1}^{N} V_{ab} \tag{10}$$

According to Hoffmann [16], given feature F_Z, the reconstruction error is calculated in feature space as follows:

$$R(I_z) = \left[\bar{\mathcal{M}}(F_z) \cdot \bar{\mathcal{M}}(F_z)\right] - \left\{\left[W\bar{\mathcal{M}}(F_z)\right] \cdot \left[W\bar{\mathcal{M}}(F_z)\right]\right\} \tag{11}$$

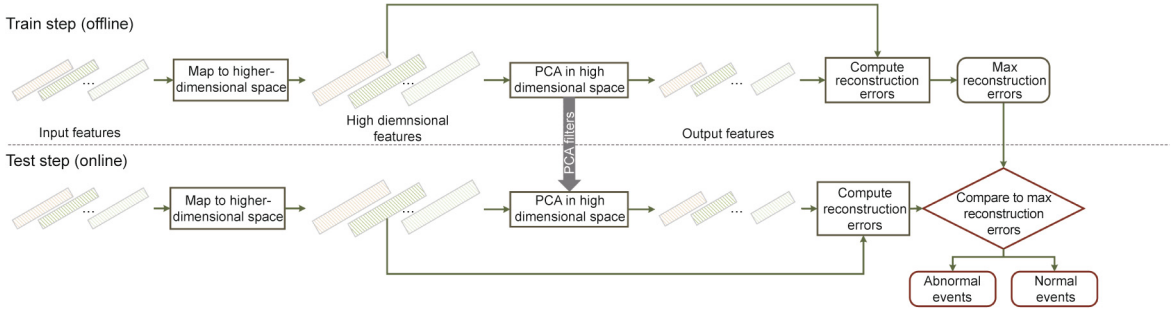

Fig. 2. The structure of a one-class classifier: kPCA.

where $W = (W_1; W_2; \ldots; W_q)$. The equation above can then be expressed more clearly, as follows:

$$\begin{aligned} R(F_z) &= \| \overline{\mathcal{M}}(F_z) \|^2 - \sum_{j=1}^{q} \left[\overline{\mathcal{M}}(F_z) \cdot W_j\right]^2 \\ &= V_{zz} - \frac{2}{N}\sum_{a=1}^{N} V_{za} + \frac{1}{N^2}\sum_{a,b=1}^{N} V_{ab} - \sum_{j=1}^{q}\left[P_j(F_z)\right]^2 \\ &= 1 - \frac{2}{N}\sum_{a=1}^{N} V_{za} + \frac{1}{N^2}\sum_{a,b=1}^{N} V_{ab} - \sum_{j=1}^{q}\left[P_j(F_z)\right]^2 \end{aligned} \tag{12}$$

In the equation above, $P_j(F_z)$ is expressed as follows:

$$\begin{aligned} P_j(F_z) &= \overline{\mathcal{M}}(F_z) \cdot W_j \\ &= \left[\overline{\mathcal{M}}(F_z) - \frac{1}{N}\sum_{a=1}^{N}\mathcal{M}(F_a)\right] \cdot \left[\sum_{i=1}^{N}\alpha_i^j \mathcal{M}(F_i) - \frac{1}{N}\sum_{i,b=1}^{N}\alpha_i^j \mathcal{M}(F_b)\right] \\ &= \sum_{i=1}^{N}\alpha_i^j\left[V_{zi} - \frac{1}{N}\sum_{a=1}^{N}V_{ia} - \frac{1}{N}\sum_{b=1}^{N}V_{zb} + \frac{1}{N^2}\sum_{a,b=1}^{N}V_{ab}\right] \end{aligned} \tag{13}$$

Hence, we obtain the desired form of measurement $R(I_z)$ to detect the anomaly.

The hyperparameters in this classifier are the number of eigenvectors q and the kernel width σ. Their values depend on the specific experiment environment.

Finally, given an input X and extracted feature F_x, we define the classifier as follows:

$$\text{status}(X) = \begin{cases} \text{anomaly } R(F_x) > \text{threshold} \\ \text{normality } R(F_x) \le \text{threshold} \end{cases} \tag{14}$$

The threshold above is the maximum reconstruction error computed in the training phase, as shown in Fig. 2.

4. Proposed AED-Net

Given the task of anomaly detection in video frames, we propose AED-Net, an integral self-supervised detection framework based on the self-supervised learning method that trains on normal data. To perform the feature extraction task based on input video frames, PCAnet is adopted as an effective network. For one-class classification, we then use kPCA, a particular one-class classifier, to determine the abnormality of the frames.

4.1. Optical flow computation

Initially, we obtain raw video frames, S. To detect the abnormal events in these frames, the moving area should first be separated from the static background in S in order to simplify the detection task. Optical flow, which represents the motion field between frames [40], is applicable to this motion extraction requirement.

The Horn–Schunck (H–S) method [41] can be used to compute optical flow. Considering the constraints of pixel value consistency and flow variety across the image, this method constructs an energy function and optimizes it to obtain optical flow in the form of u and v [41], which are the horizontal and vertical components of the optical flow. The constraint of smoothness is added to the function in order to mitigate the aperture problem. The proposed energy function is as follows:

$$E = \iint \left[\left(I_x u + I_y v + I_t\right)^2 + \alpha^2\left(\| \nabla u \|^2 + \| \nabla v \|^2\right)\right] \mathrm{d}x\mathrm{d}y \tag{15}$$

where E is the global energy; I_x, I_y, and I_t are the pixel values across the width direction, height direction, and time direction; and α is the hyperparameter controlling the smooth term.

Next, in order to process the optical flow feature as an image is processed, we visualize the optical flow u, v and obtain optical flow maps, I, using the Munsell Color System.

4.2. AED-Net

On an intuitive level, the anomaly detection task in our proposed AED-Net is to assign a score indicating abnormality to each frame of video. During a training phase, the largest reconstruction error should be set as the threshold for anomaly detection. Thus, in the testing phase, the abnormality of the test frames can be determined by comparing the score of the test frames with the threshold. To fulfill this task, we incorporate both PCAnet and kPCA to build AED-Net.

The framework of our proposed AED-Net is shown in Fig. 3, and the proposed algorithm of AED-Net is shown in **Algorithm 1**. First, optical flow maps, I are used as input of the whole framework for training and testing. Next, the PCAnet model is trained to learn to extract high-level information that better represents the situation of the scenes from the spatiotemporal features. Finally, utilizing the block-wise histograms as classification features extracted by PCAnet, kPCA is trained to learn the nonlinear data distribution of normal scenes and to determine the max normality score as the threshold computing by reconstruction error.

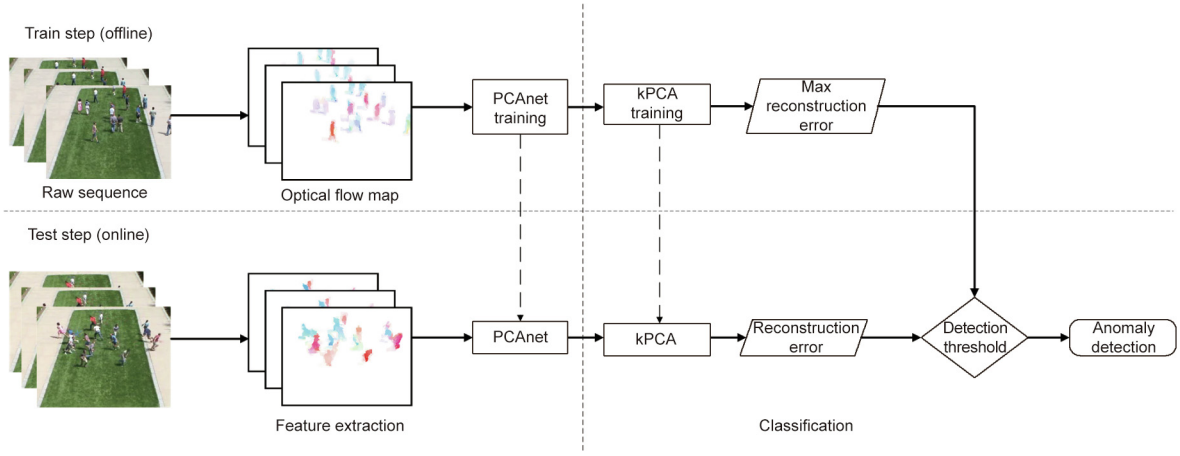

Fig. 3. Architecture of the whole framework.

During the test time, in order to minimize the influence of frames that carry little relevant information, foreground detection is first performed and frames in the test video clip that contain few people are removed. Next, k block-wise features are extracted by the PCAnet trained previously, and a test score is computed for every frame by kPCA. Finally, the test score is compared with the max normality score to determine whether the frame is abnormal.

Algorithm 1. AED-Net

Input:

Optical flow maps $I = \{I_1, I_2, ..., I_N\}$

Output:

Threshold of max normality score threshold, K^1, K^2, α

for $i = 1, 2, ..., N$ **do**

Sample I_i by patches and get X_i by vectorizing and concatenating all patches

end for

$\bar{X}_l = X_i - \frac{1}{N}(\sum X_i)$

$S_1 = \left[\bar{X}_1, \bar{X}_2, ..., \bar{X}_N\right]$

$K^1 = \underset{K^1}{\operatorname{argmin}} \| S_1 - K^1\left(K^1\right)^{\mathrm{T}} S_1 \|\text{s.t. } K^1\left(K^1\right)^{\mathrm{T}} = I_{L_1}$

$C = I * K^1,\ C = \left\{C_i^l\right\},\ i = 1, 2, ..., N,\ l = 1, 2, ..., L_1$

for $i = 1, 2, ..., N, j = 1, 2, ..., L_1$ **do**

Sample C_i^j by patches and get Y_i^j by vectorizing and concatenating all patches

end for

$\bar{Y}_l^j = Y_i^j - \operatorname{mean}\left(Y_i^j\right)$

$S_2 = \left[\bar{Y}_1^1, ..., \bar{Y}_1^{L_1}, \bar{Y}_2^1, ..., \bar{Y}_2^{L_1}, ..., \bar{Y}_N^1, ..., \bar{Y}_N^{L_1}\right]$

$K^2 = \underset{K^2}{\operatorname{argmin}} \| S_2 - K^2\left(K^2\right)^{\mathrm{T}} S_2 \|\text{s.t. } K^2\left(K^2\right)^{\mathrm{T}} = I_{L_2}$

$O_i^{l,m} = C_i^l * K_m^2,$
$l = 1, 2, ..., L_1,\ m = 1, 2, ..., L_2,\ i = 1, 2, ..., N$

$T_i^l = \sum_{m=1}^{L_2} 2^{m-1} H\left(O_i^{l,m}\right)$

$\mathcal{F}_i = \left[\text{Bhist}\left(T_i^1\right), ..., \text{Bhist}\left(T_i^{L_1}\right)\right]^{\mathrm{T}}$

for $i = 1, 2, ..., N$ **do**

  for $j = 1, 2, ..., N$ **do**

    $V_{i,j} = \kappa(\mathcal{F}_i, \mathcal{F}_j)$

  end for

end for

$\bar{V}_{i,j} = V_{i,j} - \frac{1}{N}\sum_{a=1}^{N} V_{ia} - \frac{1}{N}\sum_{a=1}^{N} V_{aj} + \frac{1}{N^2}\sum_{a,b=1}^{N} V_{ab}$

$\alpha = \underset{\alpha}{\operatorname{argmin}} \| \bar{V} - \alpha\alpha^{\mathrm{T}} \bar{V} \|\text{s.t. } \alpha\alpha^{\mathrm{T}} = I_q$

Initialize threshold = 0

for $i = 1, 2, ..., N$ **do**

  $R(\mathcal{F}_i) = 1 - \frac{2}{N}\sum_{a=1}^{N} V_{ia} + \frac{1}{N^2}\sum_{a,b=1}^{N} V_{ab} - \sum_{j=1}^{q}\left[P_j(\mathcal{F}_i)\right]^2$

and

  $P_j(\mathcal{F}_i) = \sum_{k=1}^{N} \alpha_k^j [V_{ik} - \frac{1}{N}\sum_{a=1}^{N} V_{ka} - \frac{1}{N}\sum_{b=1}^{N} V_{ib} + \frac{1}{N^2}\sum_{a,b=1}^{N} V_{ab}]$

  if $R(\mathcal{F}_i) >$ threshold **do**

    threshold $= R(\mathcal{F}_i)$

  end if

end for

4.3. Improved PCAnet with normalization technique

In the machine learning field, the generalization of an algorithm is an important but difficult task that measures the algorithm's performance on new data. Nowadays, the most popular and effective normalization technique in the deep learning field is batch normalization (BN) [42]. BN improves the network's generalization ability such that when given a sample as input, the output is determined by a whole mini-batch; thus, it never produces a deterministic output for a sample. The role of BN in elevating a model's generalization ability has been proven experimentally [42]. However, BN is not applicable to our self-supervised model because it has two trainable parameters in the implementation: γ and β. In AED-Net, we could not find ways to train these parameters. Besides, we do not feed data by mini-batches in our method. However, LRN, a light-weight normalization technique with no trainable parameters, is applicable to our task and achieved good results in the experiments.

Proposed by Krizhevsky et al. [43], the LRN scheme has been found to aid the generalization ability of a model. Response competition among contiguous outputs with the same spatial position is introduced. For an output value p_i on the ith feature map, the normalized output q_i can be calculated as follows:

$$q_i = \frac{p_i}{\left[\alpha + \delta\sum_{j\in nb(i,n)}(p_j)\right]^{\theta}}$$
$$nb(i,n) = \{j | j = \max(0, \tfrac{i-n}{2})...\min(N-1, \tfrac{i+n}{2})\} \quad (16)$$

where δ, n, α, θ are configurable parameters; δ denotes the weights on outputs of adjacent frames; α is the bias term for computational safety; θ controls the total magnitude of the normalization term; and n denotes how many adjacent frames are included in the normalization. The feature maps of a network are arranged once the network is initialized.

We introduced this scheme from CNN to PCAnet in order to improve the model's ability to generalize. It is added after computing the feature maps by convolution operation at each stage. In addition, the LRN parameters are all set intuitively before training and are not learnable, making the LRN suitable for our unsupervised framework.

5. Experiments

We carried out experiments on the UMN dataset [17] and the UCSD Ped1 and Ped2 datasets [18] for local abnormal event detection. These public datasets, which are open to the entire research community, were used to evaluate the proposed AED-Net with different criteria: the *frame-level criterion* and the *pixel-level criterion*. The UMN dataset was used to evaluate the model's capacity with the *frame-level criterion*, the UCSD Ped1 and Ped2 datasets were used to evaluate it with both *pixel-level criterion* and *frame-level criterion*. Both the evaluation criteria are based on truth-positive rates (TPR) and false-positive rates (FPR), in which "abnormal events" are denoted as "positive," while "normal status" are denoted as "negative." The results of the experiments were compared with other state-of-the-art methods, and demonstrated the superiority of our method.

5.1. Detection performance on the UMN dataset

The UMN dataset [17] is composed of three scenes—namely, a lawn, interior, and plaza—with a resolution of 240 × 320. All scenes are related to the escaping action of crowds. In this dataset [17], the evacuation behaviors of crowds are assigned as abnormal. We detect the anomalism of each frame, which is measured by frame-level criteria. Fig. 4 shows a couple of frames from each UMN scene. For computational efficiency, all optical flow maps extracted from the original video frames are resized to small sizes, which have been proved to contain sufficient information for detection.

Foreground detection is used in this experiment to avoid the disturbance of no-meaning frames. Frames that contain fewer than three whole human body motion shapes, as shown in Fig. 5, are detected directly in our work by measuring moving foreground blobs.

To improve the generalization ability of the AED-Net, a data augmentation technique is adopted in this experiment. An optical flow map is first resized to 120×160 and nine sub-maps sized 96×128 are cut from the resized map. Next, all ten maps (one of 120×160 and nine of 96×128) are resized uniformly to 24×32 for training and testing.

After removing interfering frames, we construct a training set and test set for each scene. 760 normal frames in the scene on the lawn are used for training, which forms a training set of 7600, while other normal and abnormal frames are used for testing. For the indoors scene and the plaza scene, the number of frames for training are 1100 and 1000, respectively.

For all three scenes, the hyperparameters in AED-Net are set as follows: the filter at each stage is sized as 3×3. Both stages have eight filters to reserved enough variance. The final block size is 8×8. The hyperparameters in the classifier, kernel size σ, and number of filters q, differ for each scene. They are set at $(1,\ 2800)$, $(1,\ 3800)$, and $(0.25,\ 4200)$ for $(\sigma,\ q)$ for the scene on the lawn, indoor, and on the plaza, respectively, after cross-validation. The receiver operating characteristic (ROC) curve, area under curve (AUC), and equal error rate (EER) are analyzed with the *frame-level criterion*. When plotting the ROC curve, the threshold for determining the anomalism of the frames is altered. The results, along with comparisons with other methods, are presented in Table 2 [9,15,23,34,36]. As shown in Table 2, our method achieves respectable results on frame-level anomaly detection as measured by both AUC and EER. Given the simplicity of whole framework, this result is remarkable, and is better than the state-of-the-art methods.

Table 2
Results comparison on the UMN dataset.

Method	AUC (%)	EER (%)	Ref.
Li et al.	99.5	3.7	[9]
Chaotic invariants	99.4	5.3	[23]
SF	94.9	12.6	[36]
Sparse	99.6	2.8	[34]
Bao et al.	—	2.6	[15]
Ours	**99.7**	**2.4**	—

Bold values indicate the present work study of this paper. SF: social force.

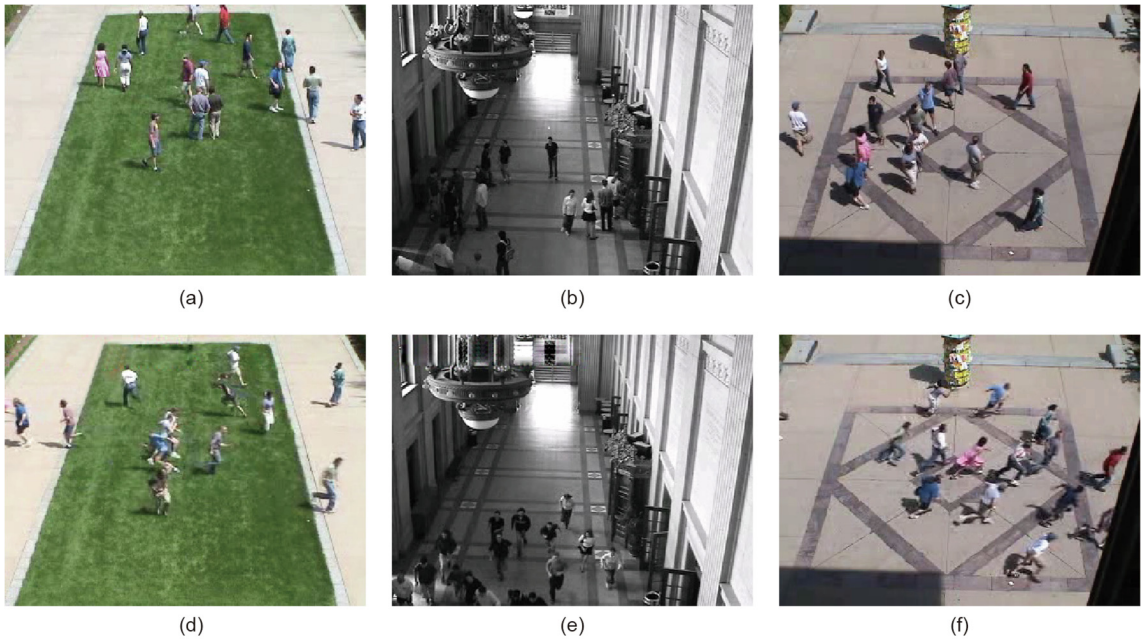

Fig. 4. Examples of video frame for three scenes. (a, d) show a scene on a lawn, (b, e) show an indoor scene, and (c, f) show a scene in a plaza. The evacuation behaviors of crowds (d–f) are assigned as abnormal.

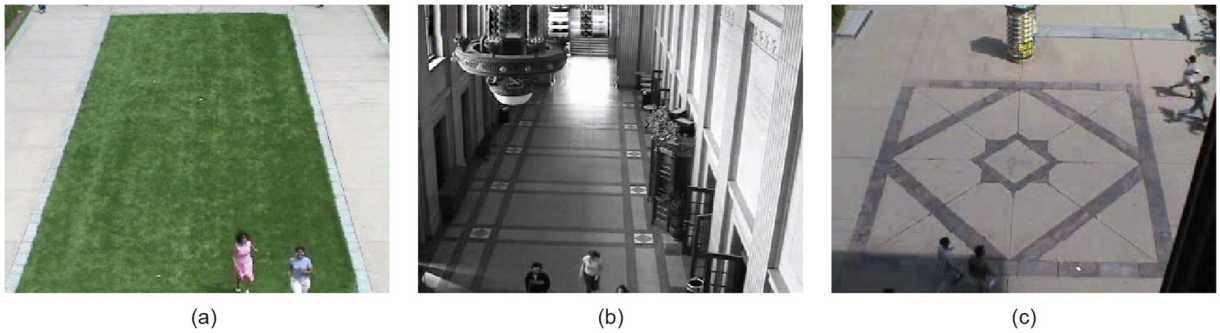

Fig. 5. Examples of abnormal video frames detected by considering the area of foreground in the frame due to its disturbance to detection. As before, (a) shows a scene on a lawn, (b) shows an indoor scene, and (c) shows a scene in a plaza.

5.2. Detection performance on the UCSD dataset

The UCSD dataset [18] contains video clips with a resolution of 158×238 obtained from a camera hung above pedestrian walkways. There are 34 training samples and 36 test samples in the Ped1 scene, and 16 training samples and 12 testing samples in the Ped2 scene, which includes people walking in different directions. The video clips that are labeled as abnormal have single anomalies such as a car, bicyclist, and so on. One of the frames with a car anomaly is shown in Fig. 6. Each video frame is partitioned into patches sized 12×16, which contain part of either walking people or the anomaly. These patches are then utilized as raw data. Assigning the anomalism of these patches is called "anomaly detection on pixel-level criteria" because it involves classifying the abnormality of a different section of the pixels of a frame.

Similar to previous experiments, foreground detection is performed here to avoid disturbance. After that, normal patches from the video frames containing the anomaly of a bicyclist are used as the training set, and abnormal patches from two frames of two video clips are used as the test set. The hyperparameters in AED-Net are set as follows: $k_1 = k_2 = 5$, $L_1 = L_2 = 7$, and block size 7×7 for experiments. The hyperparameters in the kPCA classifier are set as follows: $(0.8, 1350)$ for (σ, q).

Ped1 pixel-level and frame-level results, along with a comparison with other methods, are shown in Fig. 7 and Table 3 [9,18,28,34,36]. Ped2 pixel-level and frame-level results are shown in Table 4 [9,18,36]. In all the experiments, the proposed framework outperforms the state-of-the-art methods, especially in terms of AUC.

Fig. 6. Examples of frames of video clips containing an anomaly. (a) Frame of video clip with the anomaly of a bicyclist; (b) frame of video clip with the anomaly of a car.

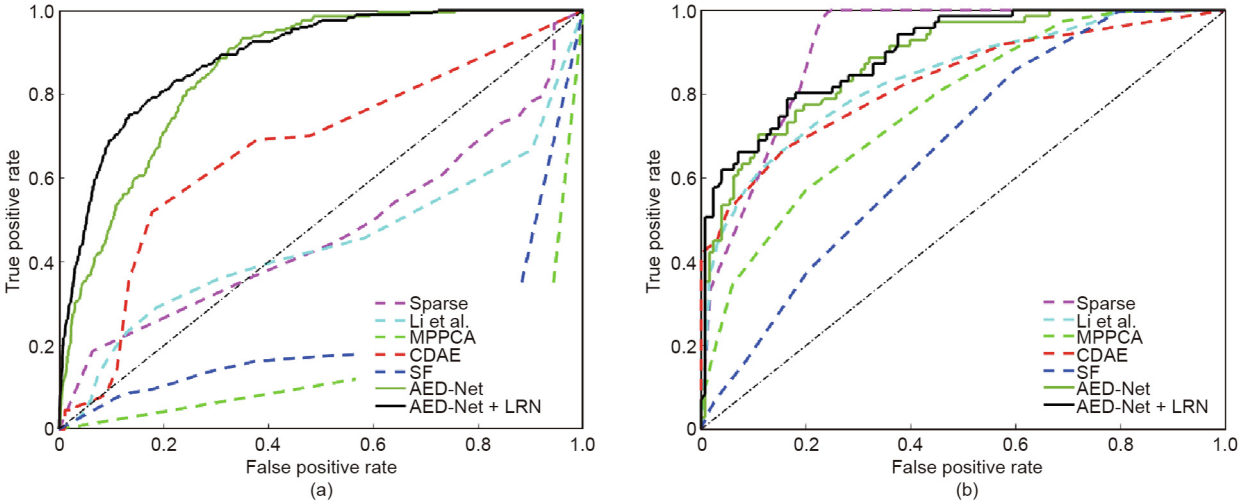

Fig. 7. Results for the Ped1 scene. (a) Pixel-level ROC for Ped1; (b) frame-level ROC for Ped1.

Table 3
Results comparison for the UCSD Ped1 scene.

Method	AUC (%)		EER (%)		Ref.
	Pixel level	Frame level	Pixel level	Frame level	
Li et al.	44.1	83.8	55.0	24.4	[9]
MPPCA	20.5	79.6	71.8	32.9	[18]
CDAE	65.8	82.9	36.9	26.8	[28]
Sparse	46.1	90.1	53.7	18.6	[34]
SF	17.9	67.0	79.0	39.2	[36]
AED-Net	86.1	88.2	22.9	22.6	–
AED-Net + LRN	88.9	89.7	19.4	19.1	–

MPPCA: mixture of probabilistic principal component analyzers; CDAE: covariance matrix of optical flow features for detection of abnormal events.

Table 4
Results comparison for the UCSD Ped2 scene.

Method	AUC (%)		EER (%)		Ref.
	Pixel level	Frame level	Pixel level	Frame level	
Li et al.	70.0	85.2	29.3	18.2	[9]
MPPCA	23.5	77.6	71.2	30.4	[18]
SF	29.1	71.6	80.2	42.3	[36]
AED-Net	88.9	90.2	22.4	20.3	—
AED-Net + LRN	91.3	89.6	16.8	15.9	—

5.3. Experiments on the improved AED-Net

After adding a LRN layer to the PCAnet, the whole framework was tested on the UCSD dataset, using the same experimental setting as the previous one used on the UCSD dataset. The hyperparameters of LRN were set as $\gamma = 2$, $\delta = 1 \times 10^{-4}$, where $n = 5$ and $\beta = 0.75$.

The results (shown in Fig. 7 and Tables 3 and 4) indicate that after the addition of the LRN, the whole framework shows better performance in detecting anomalies as measured by both AUC and EER. These findings indicate that this strategy improves our method by promoting its generalization ability.

6. Conclusion

In this work, we propose a simple but efficient framework, AED-Net, based on a self-supervised learning method. Raw data from surveillance video clips are used to calculate optical flow maps; their high-level features are then extracted by PCAnet, which is further used to determine the anomalism of local abnormal events and global abnormal events. The experimental results show that the framework performs well in detecting both global abnormal events and local abnormal events. Furthermore, after a LRN layer was added to address the overfitting problem, the performance of this framework improved. The framework achieves results that are better than state-of-the-art methods, indicating that it can effectively extract motion patterns from raw video and use them to detect anomalies.

Acknowledgements

This work is partially supported by the National Key Research and Development Program of China (2016YFE0204200), the National Natural Science Foundation of China (61503017), the Fundamental Research Funds for the Central Universities (YWF-18-BJ-J-221), the Aeronautical Science Foundation of China (2016ZC51022), and the Platform CAPSEC (capteurs pour la sécurité) funded by Région Champagne-Ardenne and FEDER (fonds européen de développement régional).

Compliance with ethics guidelines

Tian Wang, Zichen Miao, Yuxin Chen, Yi Zhou, Guangcun Shan, and Hichem Snoussi declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Kalal Z, Matas J, Mikolajczyk K. P–N learning: bootstrapping binary classifiers by structural constraints. In: Proceedings of the Computer Vision and Pattern Recognition; 2010 Jul 13–18; San Francisco, CA, USA. New York: IEEE; 2010. p. 49–56.
[2] Rui C, Martins P, Batista J. Exploiting the circulant structure of tracking-by-detection with kernels. In: Proceedings of the European Conference on Computer Vision; 2012 Oct 7–13; Florence, Italy. Berlin: Springer; 2012. p. 702–15.
[3] Zhang K, Zhang L, Yang MH. Fast compressive tracking. IEEE Trans Pattern Anal Mach Intell 2014;36(10):2002–15.
[4] Thapar D, Nigam A, Aggarwal D, Agarwal P. VGR-net: a view invariant gait recognition network. In: Proceedings of the IEEE International Conference on Identity, Security, and Behavior Analysis. 2018 Jan 11–12; Singapore. New York: IEEE; 2018.
[5] Wu Z, Huang Y, Wang L, Wang X, Tan T. A comprehensive study on cross-view gait based human identification with deep CNNs. IEEE Trans Pattern Anal Mach Intell 2017;39(2):209–26.
[6] Bojanowski P, Bach F, Laptev I, Ponce J, Schmid C, Sivic J. Finding actors and actions in movies. In: Proceedings of the IEEE International Conference on Computer Vision. 2013 Dec 1–8; Sydney, Australia. New York: IEEE; 2013.
[7] Cricri F, Roininen MJ, Leppanen J, Mate S, Curcio IDD, Uhlmann S, et al. Sport type classification of mobile videos. IEEE Trans Multimed 2014;16(4):917–32.
[8] Wang T, Chen Y, Zhang M, Chen J, Snoussi H. Internal transfer learning for improving performance in human action recognition for small datasets. IEEE Access 2017;5:17627–33.
[9] Li W, Mahadevan V, Vasconcelos N. Anomaly detection and localization in crowded scenes. IEEE Trans Pattern Anal Mach Intell 2014;36(1):18–32.
[10] Wang T, Snoussi H. Detection of abnormal visual events via global optical flow orientation histogram. IEEE Trans Inf Forensics Security 2014;9(6):988–98.
[11] Pan Y. Heading toward artificial intelligence 2.0. Engineering 2016;2(4):409–13.
[12] Xing EP, Ho Q, Xie P, Wei D. Strategies and principles of distributed machine learning on big data. Engineering 2016;2(2):179–95.
[13] Wang T, Chen Y, Qiao M, Snoussi H. A fast and robust convolutional neural network-based defect detection model in product quality control. Int J Adv Manuf Technol 2018;94(9–12):3456–71.
[14] Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y. PCANet: a simple deep learning baseline for image classification? IEEE Trans Image Process 2015;24(12):5017–32.
[15] Bao T, Karmoshi S, Ding C, Zhu M. Abnormal event detection and localization in crowded scenes based on PCANet. Multimedia Tools Appl 2017;76(22):23213–24.
[16] Hoffmann H. Kernel PCA for novelty detection. Patt Recog 2007;40(3):863–74.
[17] University of Minnesota [Internet]. Detection of unusual crowd activity; 2006 [cited 2019 February 24]. Available from: http://mha.cs.umn.edu/proj_events.shtml.
[18] Mahadevan V, Li W, Bhalodia V, Vasconcelos N. Anomaly detection in crowded scenes. In: Proceedings of the Computer Vision and Pattern Recognition; 2010 Jun 13–18; San Francisco, CA, USA. New York: IEEE; 2010.
[19] Stauffer C, Grimson WEL. Learning patterns of activity using real-time tracking. Trans Pami 2000;22(8):747–57.
[20] Zhang T, Lu H, Li SZ. Learning semantic scene models by object classification and trajectory clustering. In: Proceedings of the 2009 IEEE Conference on the Computer Vision and Pattern Recognition; 2009 Jun 20–25; Miami, FL, USA. New York: IEEE; 2009.
[21] Siebel NT, Maybank SJ. Fusion of multiple tracking algorithms for robust people tracking. In: Proceedings of the European Conference on Computer Vision; 2002 May 28–31; Copenhagen, Denmark; 2002.
[22] Cui X, Liu Q, Gao M, Metaxas DN. Abnormal detection using interaction energy potentials. In: Proceedings of the Computer Vision and Pattern Recognition; 2011 Jun 20–25; Colorado Springs, CO, USA. New York: IEEE; 2011.
[23] Wu S, Moore BE, Shah M. Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. In: Proceedings of the Computer Vision and Pattern Recognition; 2010 Jun 13–18; San Francisco, CA, USA. New York: IEEE; 2010.
[24] Khalid S. Activity classification and anomaly detection using m-mediods based modelling of motion patterns. Pattern Recognit 2010;43(10):3636–47.
[25] Rasheed N, Khan SA, Khalid A. Tracking and abnormal behavior detection in video surveillance using optical flow and neural networks. In: Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops; 2014 May 13–16; Victoria, BC, Canada. New York: IEEE; 2014.
[26] Zhang Y, Qin L, Ji R, Yao H, Huang Q. Social attribute-aware force model: exploiting richness of interaction for abnormal crowd detection. IEEE Trans Circ Syst Video Tech 2015;25(7):1231–45.
[27] Yuan Y, Fang J, Wang Q. Online anomaly detection in crowd scenes via structure analysis. IEEE Trans Cybern 2015;45(3):548–61.

[28] Wang T, Qiao M, Zhu A, Niu Y, Li C, Snoussi H. Abnormal event detection via covariance matrix for optical flow based feature. Multimedia Tools Appl 2018;77(13):17375–95.
[29] Benezeth Y, Jodoin PM, Saligrama V, Rosenberger C. Abnormal events detection based on spatio-temporal co-occurrences. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009 Jun 20–25; Miami, FL, USA. New York: IEEE; 2009.
[30] Benezeth Y, Jodoin PM, Saligrama V. Abnormality detection using low-level co-occurring events. Pattern Recognit Lett 2011;32(3):423–31.
[31] Jiménez-Hernández H, González-Barbosa JJ, Garcia-Ramírez T. Detecting abnormal vehicular dynamics at intersections based on an unsupervised learning approach and a stochastic model. Sensors 2010;10(8):7576–601.
[32] Kosmopoulos D, Chatzis SP. Robust visual behavior recognition. IEEE Signal Process Mag 2010;27(5):34–45.
[33] Adam A, Rivlin E, Shimshoni I, Reinitz D. Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Trans Pattern Anal Mach Intell 2008;30(3):555–60.
[34] Cong Y, Yuan J, Liu J. Sparse reconstruction cost for abnormal event detection. In: Proceedings of the Computer Vision and Pattern Recognition; 2011 Jun 20–25; Colorado Springs, CO, USA. New York: IEEE; 2011. p. 3449–56.
[35] Zhao B, Li F, Xing EP. Online detection of unusual events in videos via dynamic sparse coding. In: Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition. 2011 Jun 20–25; Colorado Springs, CO, USA. New York: IEEE; 2011. p. 3313–20.
[36] Mehran R, Oyama A, Shah M. Abnormal crowd behavior detection using social force model. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009 Jun 20–25; Miami, FL, USA. New York: IEEE; 2009.
[37] Bhatnagar G, Wu QMJ, Raman B. Discrete fractional wavelet transform and its application to multiple encryption. Inform Sci 2013;223:297–316.
[38] You X, Du L, Cheung Y, Chen Q. A blind watermarking scheme using new nontensor product wavelet filter banks. IEEE Trans Image Process 2010;19(12):3271–84.
[39] Kemmler M, Rodner E, Wacker ES, Denzler J. One-class classification with Gaussian processes. Patt Recog 2013;46(12):3507–18.
[40] Burton A, Radford J. Thinking in perspective: critical essays in the study of thought processes. North Yorkshire: Methuen; 1978.
[41] Horn BK, Schunck BG. Determining optical flow. Artif Intell 1981;17(1–3):185–203.
[42] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. 2015. arXiv:1502.03167.
[43] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems; 2012 Dec 3–6; Lake Tahoe, NV, USA. New York: ACM; 2012.