SIViP
DOI 10.1007/s11760-016-0935-0

ORIGINAL PAPER

An efficient system for anomaly detection using deep learning classifier

A. R. Revathi[1] · Dhananjay Kumar[2]

Received: 21 November 2015 / Revised: 6 July 2016 / Accepted: 8 July 2016

Abstract In this paper, a deep learning-based anomaly detection (DLAD) system is proposed to improve the recognition problem in video processing. Our system achieves complete detection of abnormal events by involving the following significant proposed modules a Background Estimation (BE) Module, an Object Segmentation (OS) Module, a Feature Extraction (FE) Module, and an Activity Recognition (AR) Module. At first, we have presented a BE (Background Estimation) module that generated an accurate background in which two-phase model is generated to compute the background estimation. After a high-quality background is generated, the OS model is developed to extract the object from videos, and then, object tracking process is used to track the object through the overlapping detection scheme. From the tracked objects, the FE module is extracted for some useful features such as shape, wavelet, and histogram to the abnormal event detection. For the final step, the proposed AR module is classified as abnormal or normal event using the deep learning classifier. Experiments are performed on the USCD benchmark dataset of abnormal activities, and comparisons with the state-of-the-art methods validate the advantages of our algorithm. We can see that the proposed activity recognition system has outperformed by achieving better EER of 0.75 % when compared with the existing systems (20 %). Also, it shows that the proposed method achieves 85 % precision rate in the frame-level performance.

Keywords Abnormal detection · EER · Deep learning · Background model · Video surveillance

✉ A. R. Revathi
revathiar0778@gmail.com

[1] SRM- Valliammai Engineering College, Chennai, India

[2] Department of Information Technology, Anna University, MIT, Chennai, India

1 Introduction

In the past few years, in the field of multimedia content and computer vision analysis, recognizing atomic human actions from the videos 'in the wild' has been the most important research topic [1]. Videos have rich structural information that can be applied for video indexing and retrieval. From the point of competent indexing and mining of videos, video content analysis is to find meaningful structures and samples from the visual data [2]. Video analysis tasks comprise video parsing, content indexing, abstraction, and representation. The early works focus on the low-level parsing, i.e., the video shot boundary detection [3]. The task of activity recognition [16–19] is to overpass the gap among the numerical pixel level data and a high-level abstract activity explanation. On the other hand, multiview video sequences are frequently captured under differing illumination and lighting conditions and multiple cameras may have dissimilar and unfamiliar parameters, e.g., positions, orientations, and zooming factors [4].

In the human activity classification in videos, many considerable methods have been introduced. For many years, Kernel-based classifiers have been famous in a large range of applications [5], which frequently lead to significantly improved presentation. Due to its dependable performance across many dissimilar tasks, SVM is the commonly employed algorithm together with the high-level video event classification [6,22]. On the other hand, the presentation of SVM classification is responsive to a few parameters. The dynamic time warping (DTW) [7], a technique for measuring resemblance among the two temporal sequences, which may differ in time or speed, is one of the most general temporal classification algorithms. In addition, some probability-based techniques by generative models (dynamic classifiers) are suggested such as hidden Markov models

Published online: 19 August 2016

(HMM) [8,9] and dynamic bayesian networks (DBN) [10]. Conversely, discriminative models (static classifiers) such as support vector machine (SVM) [11,12], relevant vector machine (RVM) [13,14], and artificial neural network (ANN) [15] can furthermore be applied in the video signal classification. As a result, other methods are suggested, such as the Kalman filter [16], binary tree [17], multidimensional indexing [18], and K-nearest neighbor (K-NN) [19]. Dissimilar classification algorithms regularly need dissimilar sets of appropriate feature representations. For video signal classification, the Bayesian neural network has more competence alternatively [20].

This paper develops an activity recognition system which classifies the abnormal/normal event from the crowded scenes. The organization of the proposed system is as follows.

(a) A two-phase Background Estimation Module (BE) is used to select the optimal background candidates for the generation of the updated background models.
(b) An object segmentation method is done from each video frame through Object Segmentation (OS) Module.
(c) Useful features are extracted in the activity recognition process from every tracked object through an FE module.
(d) The activities are classified via AR module using the deep learning classifier.

The rest of this paper is organized as follows: Sect. 2 gives a brief description of the literature survey. Section 3 explains the proposed activity recognition system. Result and discussion are discussed in Sect. 4. The conclusion is summed up in Sect. 5.

2 Review of related works

For human activity recognition, a lot of research has been executed. In the subsequent section, a few of the latest associated works regarding the human activity recognition are reassessed. The context-aware activity recognition and anomaly detection in video have been described by Zhu et al. [21]. They illustrated the promising results on the VIRAT Ground Dataset that reveal the advantages of joint modeling and the recognition of activities in a wide-area scene and the efficiency of the technique in anomaly detection. Cong et al. [32] have presented an abnormal video event detection system that considers both the spatial and temporal contexts. For anomaly measurements, we formulate the abnormal event detection as a matching problem, which was more robust than statistic model-based methods, especially when the training dataset is of limited size.

The matching algorithm for multiview video sequences has been described by Lee et al. [23], which offers dependable performance even when the multiple cameras have considerable dissimilar parameters, such as viewing angles and positions. In addition, Vijayakumar and Nedunchezhian [24] have progressed a video and audio features based on the event detection approach and explained about the success when used for the cricket sports video. A key frame detection-based approach has been executed by Mahesh et al. [25] toward the semantic event detection and categorization in cricket videos. The developed plan executes a top-down event detection and categorization by means of the hierarchical tree. Their classifiers illustrated tremendous results with exact detection and classification with reduced processing time. The real-time video event recognition using a semantic-based probabilistic has been described by San Miguel and Martinez et al. [26]. In particular, they applied Bayesian networks and probabilistically enlarged Petri Nets for recognizing, correspondingly, simple and complex events.

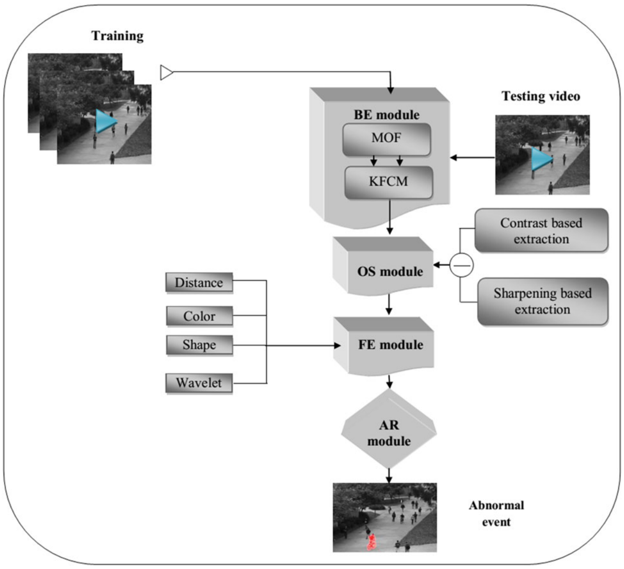

Fig. 1 Overall diagram for the proposed anomaly detection

3 The proposed method for activity recognition

The overall structure of the proposed system is illustrated in Fig. 1.

3.1 Background Estimation (BE) Module

At first, the proposed BE module designs two-phase background estimation method using MOF and Kernel fuzzy C-means (KFCM) clustering with the aim of constructing optimum background pixels for the background model.

The BE module is performed by two phases.

Phase 1 In this phase, the background is estimated using the best representation for the video shots called most occurrence of frequency (MOF), which carries both the spatial

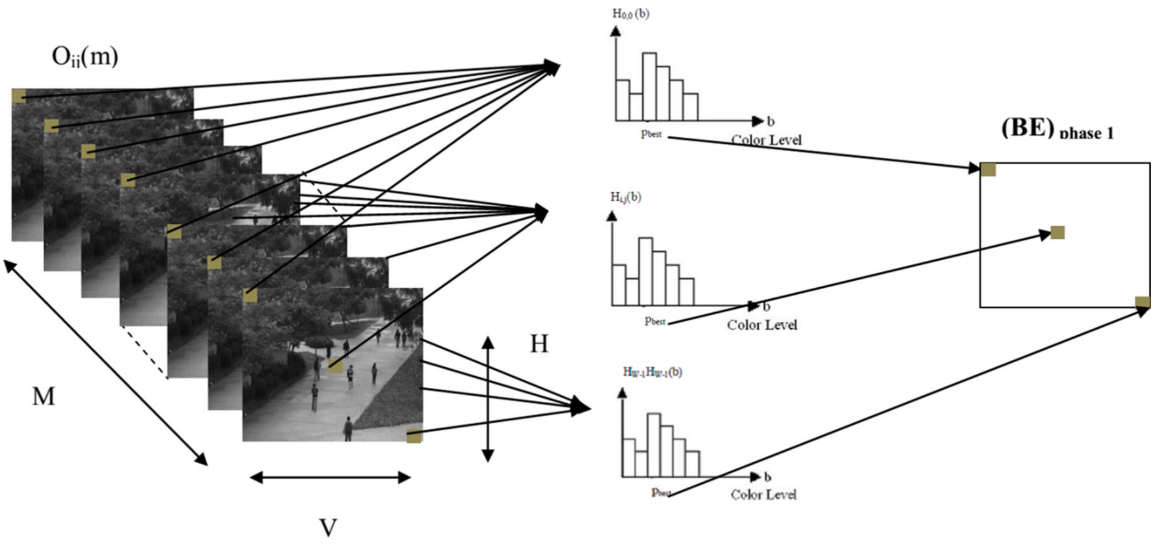

Fig. 2 MOF computation-based background detection in phase 1

and global information about the frames in the video shots to achieve a high activity recognition performance. This representation is based on the frequency of occurrence of the pixels at the same position in the frames within a shot. In other words, a frame is represented, whose value at each pixel position corresponds to that of the pixel with the largest frequency of occurrence. Figure 2 demonstrates the simplified diagram of how to obtain our proposed BE in phase 1. A histogram is formed based on the pixel values at each corresponding pixel position in the video. Then, the value at the pixel position in an MOF is the bin value whose most occurrence of frequency (MOF) or count is the maximum in the histogram. Therefore, the pixels of the MOF are calculated as follows

$$\mathrm{MOF}(x, y) = p_{\mathrm{best}}, 0 \leq i \leq U' - 1 \text{ and } 0 \leq i \leq V' - 1 \tag{1}$$

where

$U' \times V'$ is the size of a MOF.

p_{best} is chosen as follows:

$$p_{\mathrm{best}} = \arg\max_{p} \left\{ V'_{i,j}(p) \right\}, \quad \text{for } 0 \leq p \leq P \tag{2}$$

The process is repeated for each pixel and until the background image $B(x, y)$.

Phase 2 In this phase, another background model $B'(x, y)$ is determined using KFCM clustering algorithm to improve the background model estimation. At first, the frame differences are calculated for each frame and the mean value is computed from these different frames. Then, Kernel fuzzy C-means clustering is applied to the main frame, and then, n-number of clusters are obtained. From this, the background model $B'(x, y)$ is obtained based on the distance to the centroid of a cluster, and the closest one is elected as the background model $B'(x, y)$ or which cluster is closest to the zero range that cluster is selected as $B'(x, y)$.

$$B'(x, y) = \arg\min_{c}(c_j) < 0 \tag{3}$$

where

c_j is the jth cluster center.

3.2 Object Segmentation (OS) and object tracking (OT) Module

How to segment object regions in video frames is described with the following steps.

(1) Initial object regions

After the background model is produced through the BE procedure, the absolute difference $\Delta_t(x, y)$ is generated by the absolute differential estimation between the background models $B(x, y)$, $B'(x, y)$ and current incoming video frame $I(x, y)$. Now, the two-level difference is performed to obtain accurate object using a BE module. Here, the current incoming video frame $I(x, y)$ is subtracted with the background model $B(x, y)$, and then, the remaining object is subtracted with the background model $B'(x, y)$ to get the optimal region. At the first level, the absolute differential estimation between $B(x, y)$ and current incoming video frame $I(x, y)$ is given as

$$D_1 = B(x, y) - I(x, y) \tag{4}$$

where, D_1 is the absolute difference for the first level, $B(x, y)$ is the background model (from phase 1 in BE module), $I(x, y)$ is the current incoming video frame
At the second level, the absolute differential estimation between $B'(x, y)$ and current incoming video frame $I(x, y)$ is given as

$$D_2 = B'(x, y) - I(x, y) \tag{5}$$

where, D_2 is the absolute difference for the second level, $B'(x, y)$ is the background model (from phase 2 in BE module).

(2) Contrast-based information extraction
In this step, the contrast-based measure is applied to the differential image to identify the contrast information. Here, the *imadjust* function [28] is applied to increase the contrast of the differential image by mapping the values of the input intensity image to the new values such that, by default, 1 % of the data is saturated at low and high intensities of the input data. After that, the adjusted image is converted to binary image and it is denoted by $B_C(x, y)$.

(3) Sharpening-based information extraction
On the other hand, the sharpening-based filter is applied to the differential image to detect sharp edges. Here, Laplacian filter is used in our approach. Laplacian filter is one of the commonly used image sharpening operators [28], and it can be described by the following formula.

$$\left.\begin{array}{l}\nabla^2 B_C(x, y) = B_C(x+1, y) + B_C(x-1, y) + \\ B_C(x, y+1) + B_C(x, y-1) - 4B_C(x, y)\end{array}\right\} \tag{6}$$

where, $B_C(x, y)$ represents the pixel value of an input image.

According to (6), the Laplacian operator reflects the variety of one pixel relative to its surrounding four pixels. When Laplacian operator is used to get the sharpened image, the sharpened pixel value is

$$g(x, y) = B_C(x, y) - k\nabla^2 B_C(x, y) \tag{7}$$

where, k is the parameter concerning with the diffusion effect.

After the sharpened image is obtained, this image is converted into binary image, and it is denoted by $B_S(u, v)$.

(4) Object extraction
The object is extracted from the difference between the contrast-based information image and the sharpening-based information image.

$$O_D(c, y) = B_C(x, y) - B_S(x, y) \tag{8}$$

The obtained object results $O_D(c, y)$ in formation of many segments, and some of them may be irrelevant. In order to remove these, those segments having less area are neglected. For this, a threshold is fixed and the area of each segment is compared against it. Area of the segment is the total number of pixels that lie in the corresponding segment. Suppose the threshold set is denoted by T_h and the area of the ith segment be represented by A_i, then the condition is defined such that:

$$R_s(c, y) \Leftarrow \left\{\begin{array}{l} f(A_i < T_h), \text{Then Remove } Ar_i \\ if(A_i \geq T_h), \text{Then Accept } A_i \end{array}\right\} \Leftarrow O_D(c, y) \tag{9}$$

where $R_s(c, y)$ is the resultant object (after removing noise region).

Removal of smaller areas also results in reducing noise. After an object extraction process, the object tracking process is performed for every frame. Each object in an image is tracked by searching for an object in subsequent image of the video clip that overlaps most with the given object. This is carried out using the neighborhood estimation among the frames.

3.3 Feature Extraction (FE) Module

In this module, features are extracted from the tracked objects. The seven features extracted include the distance measure (2 features), shape related (2 features), wavelet related (2 features), and the histogram.
Distance measure Distance measure takes the distance between the centroid of objects in subsequent frames (or images) in clip in x and y directions. Here, initially centroids of the objects are found out in each frame using regionprops MATLAB function. Let the objects in the frame be represented by Ob. Let the pixels inside the ith object Ob_i are represented by poi_j where $0 < j < n$ and the respective centroid is calculated as

$$\mathrm{cen}_i = \frac{\sum_{j=1}^{n} \mathrm{poi}_j}{n} \tag{10}$$

After finding out the centroid for each object for each frame, Euclidean distance is taken between the centroid of objects in subsequent frames. This distance found out forms a feature.
Shape-related features Another feature taken is the size of the bounding box of object in x and y directions. The minimum or the smallest bounding or enclosing box is a term used in geometry. In our case, the bounding box for the two dimensions (x and y) is found out. It refers to the box with the smallest measure within which all the pixels lie. It is also called minimum-perimeter bounding box. The area of the object also forms a feature which is the number of pixels enclosed inside the object boundary.

$$F_{\mathrm{W}}^{x,y} = \frac{N_{\mathrm{W}}(P)}{N(P)} \tag{11}$$

$$F_{\mathrm{B}}^{x,y} = \frac{N_{\mathrm{B}}(P)}{N(P)} \tag{12}$$

where $F_{\mathrm{W}}^{x,y}$ → Frequency of white pixel in x and y direction the size of bounding box

$F_{\mathrm{B}}^{x,y}$ → Frequency of black pixel in x and y

direction the size of bounding box

$N_{\mathrm{W}}(P)$ → Number of white pixel
$N_{\mathrm{B}}(P)$ → Number of white pixel
$N(P)$ → is Number of pixels

Wavelet-related features Wavelet-based features are extracted to additional computation of feature extraction process. Here, haar wavelet is used to extract the multirepresentation of the given image. One or more of the subbands such as CA, CV, CH, and CD can be used as a field for the feature computation; the CA subband has generally the best results because of the greater success. In our case, the haar wavelet is applied on the tracked object, and then, the mean and variance are computed from the CA subband. These mean and variance are selected as the wavelet feature. The general expression of the mean and variance formula is given by

$$\mu_{\mathrm{CA}} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} g(i, j) \tag{13}$$

$$\sigma_{\mathrm{CA}}^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} g^2(i, j) - \mu^2(i, j)) \tag{14}$$

Histogram-related features Histogram value is the fourth feature taken. A histogram is a graphical representation of the distribution of the image pixels. A histogram is a representation of the tabulated frequencies, shown as adjacent rectangles, erected over the discrete intervals, with an area proportional to the frequency of the observations in the interval. Histogram peak value is the maximum histogram value in the given interval. Histogram value is the histogram peak value divided by the total number of pixels in the object image after removing the background pixels.

$$H_n = \frac{\text{Number of pixels with intensity “}n\text{”}}{\text{total number of pixels}} n = 0, 1, 2, \ldots, L^{-}1 \tag{15}$$

3.4 Activity Recognition (AR) Module

Deep neural network (DNN) classifier [29] is employed as a solution for the activity recognition problem in this section. An artificial neural network model with the multiple layers of the hidden units and outputs is termed DNNs. In addition, it contains both the pre-training (using generative deep belief network or DBN) and fine-tuning stages in its parameter learning. To train the three sets of features is the core focus of our research in the training dataset, i.e., to find the right weight that can be applied to perfectly categorize the input features. In Fig. 3, an example of a DBN for classification is displayed. It contains an input layer which encloses the input units (called visible units), a number L of hidden layers, and at last an output layer which has one unit for each class regarded. The parameters of a DBN are the weights $W^{(j)}$ among the units layers $j-1$ and j and the biases $b^{(j)}$ of layer j.

Pre-training stage One of the most important problems for training deep neural network architectures is how to initialize these parameters. Arbitrary initialization causes optimization algorithms to discover poor local minima of the fault function effecting in low generalization. In order to work out the above problem based on the training of a sequence of Restricted Boltzmann Machines (RBMs), a novel algorithm has been brought in by Hinton et al. [29]. An RBM is a two-layer repeated neural network in which stochastic binary inputs are linked to stochastic binary outputs by symmetrically weighed connections. The first layer relates to inputs (visible units v) and the second layer to the concealed units h of the RBM. Unseen units can be regarded to act as feature detectors after RBM training, i.e., they create a compact representation of the input vector. In Fig. 3, an example of an RBM is specified.

An RBM is described by the energy function defined as

$$E(v, h) = -h^T W v - b^T v - c^T h \tag{16}$$

where W is the weight and b, c are the bias vectors for visible and hidden layers, respectively. The layer to layer conditional distributions

$$P(v_i = 1\,|h) = \sigma\left(b_i + \sum\nolimits_j W_{ji} h_j\right) \tag{17}$$

$$P(h_i = 1\,|v) = \sigma\left(c_i + \sum\nolimits_j W_{ji} v_j\right) \tag{18}$$

where, $\sigma(a) = 1/(1 + e^{-a})$ is the logistic function, providing outputs in the (0, 1) range. By sampling using the above probabilities, the output (0 or 1) of an RBM unit is determined.

Then, we explain how to train an RBM and how it is applied in the construction of an RBN. Initially we must highlight that RBM training is unsupervised. Specified a training example, we disregard its class label and we spread it stochastically through the RBM. In Eq. (18), the outputs of the hidden units follow the conditional distribution given. Our next sample from this distribution, hence producing a binary vector. This vector is circulated in the opposite direction through the RBM [from the hidden units to visible units using (17)] which effects in a confabulation (reconstruction) of the unique input

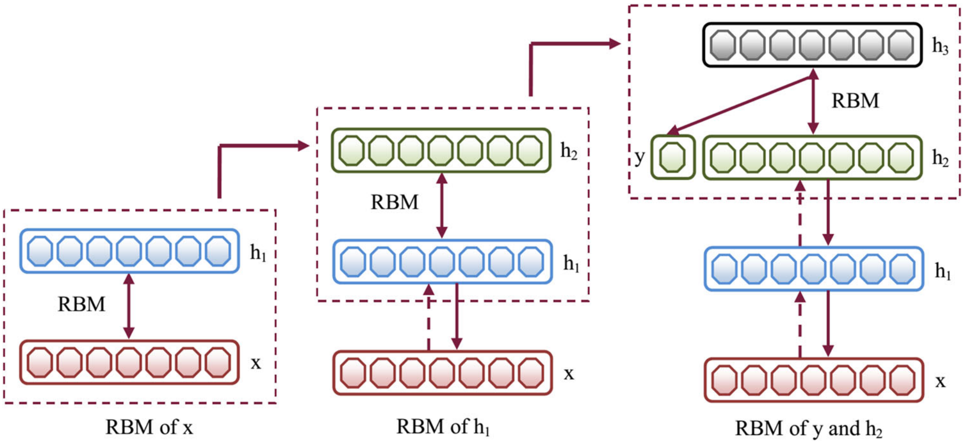

Fig. 3 Deep belief network with 3 hidden layers $h_1, h_2, and h_3$, one input layer x, and one output layer y

data. At last, the condition of the hidden units is revised by propagating this confabulation through the RBM.

The above procedure is executed continually for all the examples of the training set, and after that, the revision of the parameters happens as follows.

$$\Delta W_{ji} = \eta \left(\left\langle v_i h_j \right\rangle_{\text{data}} - \left\langle v_i h_j \right\rangle_{\text{reconstruction}} \right) \tag{19}$$

$$\Delta b_i = \eta \left(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{reconstruction}} \right) \tag{20}$$

$$\Delta c_j = \eta \left(\left\langle h_j \right\rangle_{\text{data}} - \left\langle h_j \right\rangle_{\text{reconstruction}} \right) \tag{21}$$

Procedure for pre-training the DNN

- Initially, the visible unit v is initialized to a training.
- Next, we update the hidden units using Eq. (18)
- In the same way, we update the visible units using Eq. (17)
- Re-update the hidden units and the reconstructed visible unit using the same equation used in step 2
- Perform the weight updates

$$\Delta w_{ij} \; \alpha \; \left\langle v_i h_j \right\rangle_{\text{data}} - \left\langle v_i h_j \right\rangle_{\text{reconstruction}}$$

One more RBM can be stacked at the top of it to make a multilayer model once the RBM is trained. Each time another RBM is stacked, the input visible layer is initialized and the values for the units in the already trained RBM layers are assigned using the current weights and biases. The final layer of the already trained layers is used as input to the new RBM. The new RBM is then trained in the procedure above, and then, this whole process can be repeated until some desired stopping criterion is met. In our DNN, 253 input layers, 3 hidden layers, and 2 output layers are used. The obtained deep network weights are used to initialize a fine-tuning stage.

Anomaly detection through the fine-tuning stage

The anomaly detection is carried out using the deep learning classifier described in the previous subsection. The fine-tuning stage is done as the ordinary back propagation algorithm. For classification tasks, a class layer is defined as abnormal (integer value 1) and normal (integer value 0) is added at the top of the network. Each neuron of this layer is activated for each class label while others are deactivated. The back propagation begins from the weights obtained in the pre-training stage. The top layer activations are obtained for each training set example, which is obtained in the forward path, and then, the error signal between the obtained activations and required targets is back propagated on the network for weights adjustment. The testing video is given to the deep learning classifier, where the trained weight is used as hidden layer weight for the testing stage. Finally, the decision is generated whether the test video belongs to the abnormal event or not.

4 Experiments and comparison

In this section, we report experimental results from the activity reported by the recognition approach detailed in the previous section.

4.1 Dataset description

The dataset used for the performance evaluation of the proposed system on the UCSD pedestrian dataset [27] is a well-annotated publicly available dataset for the evaluation of abnormal detection and localization in crowded scenes. The dataset was obtained with a stationary camera mounted at an elevation at a resolution of 238 × 158 with 10 fps, overlooking pedestrian walkways. The dataset contains different crowd densities, and the anomalous patterns are the presence of non-pedestrians on a walkway, and the anomalies include bicyclists, skaters, small carts, and people in wheelchairs. Videos were split into two subsets: Ped_1 and Ped_2, each corresponding to a different scene. Videos recorded from each scene were split into various clips each of which has around 200 frames. Ped_1 contains 34 training clips and

36 testing clips, while Ped_2 contains 16 training clips and 14 testing clips. For each clip, the ground truth annotation includes a binary flag per frame, indicating whether an anomaly is present in that frame.

4.2 Evaluation metrics and measurement

The receiver operating characteristic (ROC) curve is used to measure the accuracy of multiple threshold values. The ROC consists of the true positive rate (TPR) and the false positive rate (FPR), of which TPR determines a classifier or a diagnostic test performed on classifying positive instances correctly among all the positive samples available during the test, and FPR, on the other hand, defines how many incorrect positive results occur between all the negative samples available during the test.

To test the effectiveness of our proposed algorithm, two different levels of measurements are applied for evaluation, i.e., *pixel level* and *frame level*. Both are based on true positives (TPR) and false positive rates (FPRs) denoting 'an anomalous event' as 'positive' and 'the absence of anomalous events' as 'negative.'

4.3 Performance evaluation of activity recognition

Performance evaluation of any anomaly detection method can be conducted either in the frame or in the pixel level. Some testing results are shown in Fig. 4, where our method can detect abnormal events such as cycles and cars. In Fig. 4, (i, v) sample frames of normal actions for the two scenes containing only walking pedestrians (ii, iii, iv, vi, vii, and viii) abnormality detection in the query videos are highlighted in red. Figure 5 shows the (a) frame- and (b) pixel-level precision/recall performance for the different thresholds (Figs. 6, 7).

The first and third row shows some samples ground truth and the corresponding detected outputs for UCSD1. The second and fourth row shows some samples ground truth and corresponding detected outputs for UCSD2, where abnormality detection in the query videos is highlighted in red. Precision and recall performance (frame and pixel level) for Ped_1 and Ped_2 are presented in Table 1. Analyzing Table 1, the proposed method is achieved about 85 % precision rate in the frame-level performance in Ped_1 dataset. On the other hand, the proposed method is achieved about 82 % precision rate in the frame-level performance in Ped_2 dataset.

4.4 Comparative analysis

In this section, our proposed method is compared against the state-of-the-art methods, such as DTM [31], sparse [32], and STMCAD [30] by using pixel-level and frame-level measurements.

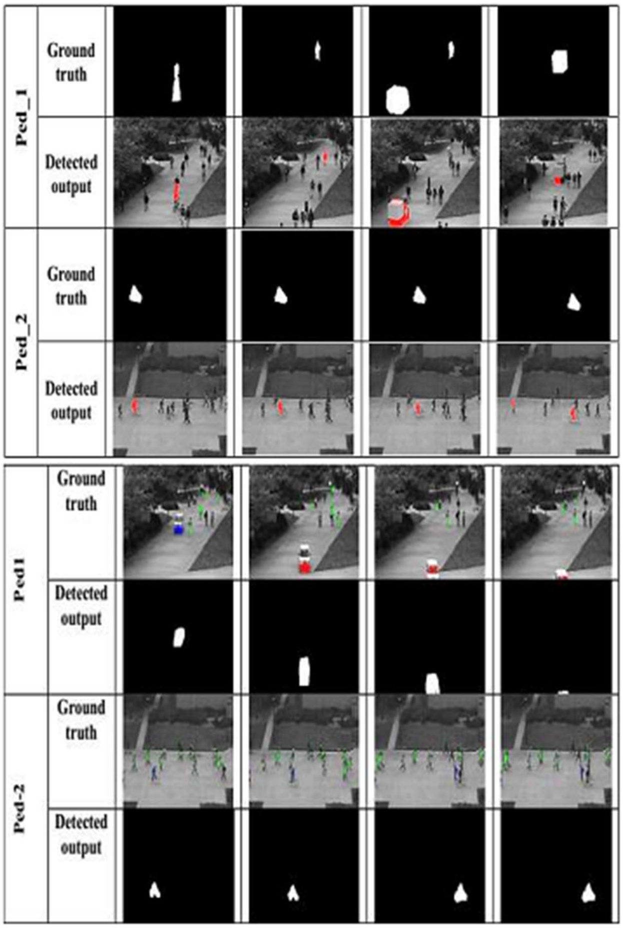

Fig. 4 Examples of abnormal detections the UCSD1 and 2 pedestrian datasets

Dataset 1 (Ped_1) In Table 2, the performance is evaluated using different criteria: for the equal error rate (EER), our approach is 0.75 %, which is higher than the STMCAD algorithm [30], DTM [31], and sparse-based algorithm [32]; for the rate of detection (RD), our method gives the detection rates of 70.21 %, and the STMCAD algorithm [30] gives only the detection rate of 53 %; and for the area under the curve (AUC), ours make a big improvement of 70.21 %, but the existing method makes only 47.1 %.

Dataset 2 (Ped_2) In Table 3, for the equal error rate (EER), our approach is 18 %, which is higher than the STMCAD algorithm (23 %), and for the area under the curve (AUC), STMCAD algorithm makes a bit improvement of 86.8 %, but the proposed method makes only 80 %. Therefore, it exhibits that our method out performs than the STMCAD algorithm [30].

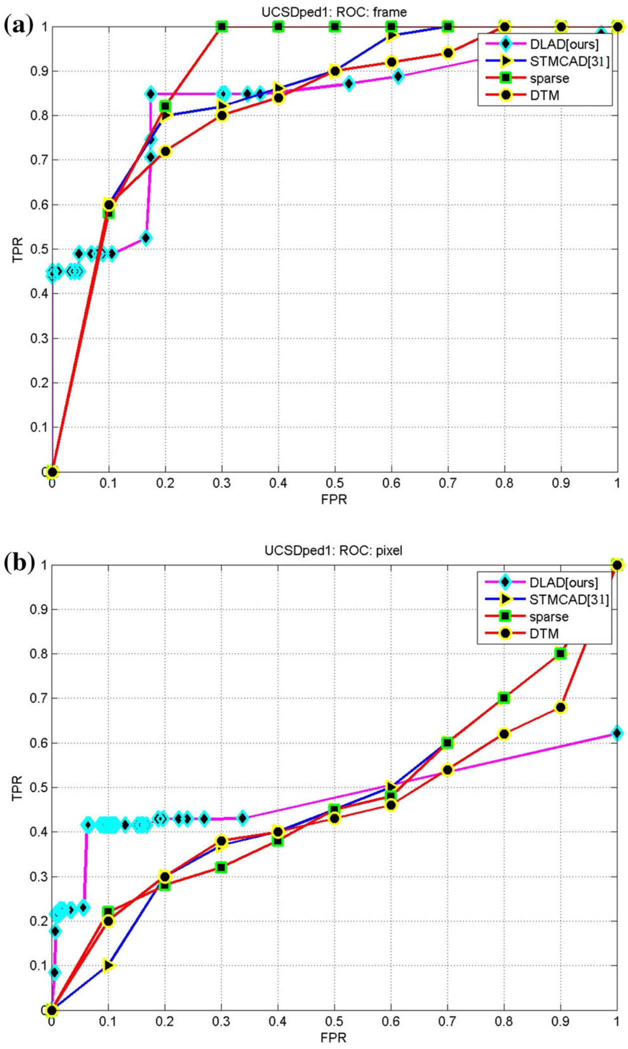

Fig. 5 Performance evaluation results of UCSD Ped_1 dataset. **a** Frame-level ROC for Ped_1 dataset, **b** pixel-level ROC for Ped_1 dataset

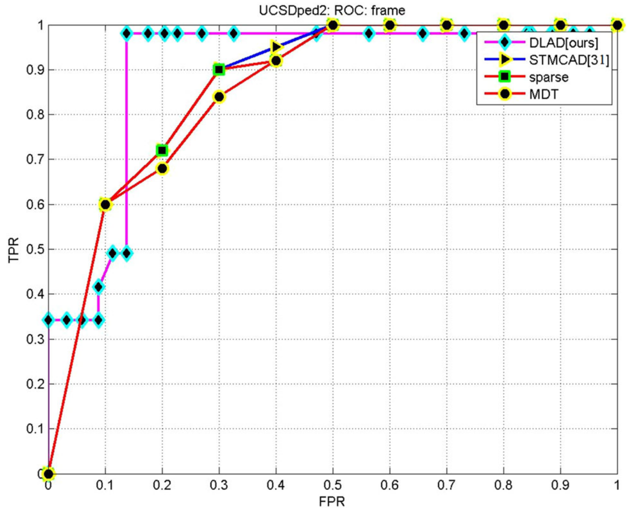

Fig. 6 Frame-level ROC for Ped_2 dataset

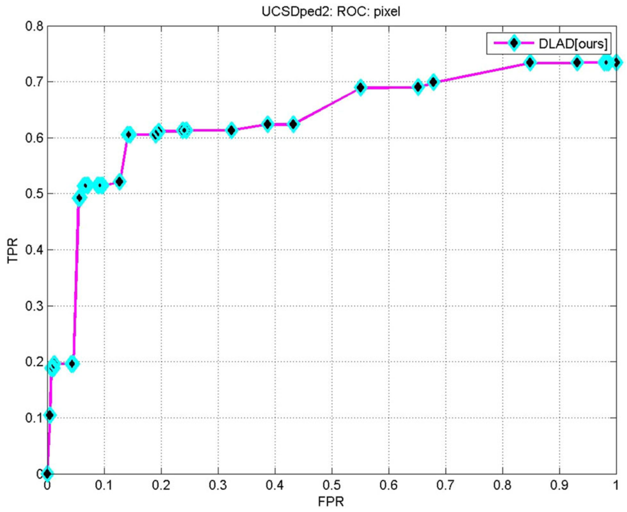

Fig. 7 Pixel-level ROC for Ped_2 dataset

Table 1 Frame-and pixel-level precision/recall performance

Measure	Ped_1		Ped_2	
	Frame (%)	Pixel (%)	Frame (%)	Pixel (%)
Precision	85	5	82	25
Recall	50	48	87	62

Table 2 Activity recognition performance of Ped_1

Dataset	Methods	Frame level	Pixel level	AUC (%)
		EER (%)	RD (%)	
Ped_1	Proposed	*0.75*	*40*	*55.45*
	STMCAD [30]	23	47	47.1
	Sparse [32]	19	46	46.1

Table 3 Activity recognition performance of Ped_2

Dataset	Methods	Frame level EER (%)	AUC (%)
Ped_2	Proposed	*18*	*72.64*
	STMCAD [30]	23	86.8
	Sparse [32]	25	86.1

5 Conclusion

In this paper, we have proposed a system to detect the abnormal events through a diverse set of features and deep learning classifier. Our system comprised the following significant modules: a Background Estimation (BE) Module, an Object Segmentation (OS) Module, a Feature Extraction (FE) Module, and an Activity Recognition (AR) Module. Initially, we developed a BE (Background Estimation) module that generated an accurate background in which two-phase model is generated to compute the background estimation. After a high-quality background was generated, the OS model was developed to extract the object from the videos, and then, the object tracking process was used to track the object through the overlapping detection scheme. From the tracked objects, the FE module was extracted some useful features such as shape, wavelet, and histogram. Finally, the proposed OR module is categorized as abnormal or normal event using deep learning classifier. The experiment shows that our proposed method achieved a better error rate of 0.75 % than the existing method (20 %). In future, we will work on extending the proposed system to more video applications. Moreover, we plan to apply efficient feature selection methods to reduce the feature dimension while keeping its discriminative, robustness, and constancy individuality.

References

1. Liu, J., Yu, Q., Javed, O., Ali, S., Tamrakar, A., Divakaran, A., Cheng, H., Sawhney, H.: Video event recognition using concept attributes. *IEEE Workshop on Application of Computer Version* (2013)
2. Wang, X., Zhang, X.P.: An ICA mixture hidden conditional random field model for video event classification. IEEE Trans. Circuits Syst. Video Technol. **23**(1), 46–59 (2012)
3. Hanjalic, A.: Content-Based Analysis of Digital Video. Kluwer, Dordrecht (2004)
4. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: theory and experiment. IEEE Trans. Pattern Anal. Mach. Intell. **16**(9), 920–932 (1994)
5. Herbrich, R.: Learning Kernel Classifier: Theory and Algorithm. The MIT Press, Cambridge (2002)
6. Natarajan, P., Nevatia, R.: Real time tracking and recognition of human actions. In: *Proceedings of IEEE Workshop on Motion and Video, Computing*, pp. 1–8 (2008)
7. Sempena, S., Maulidevi, N., Aryan, P.R.: Human action recognition using dynamic time warping. In: *IEEE International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 1–5 (2011)
8. Duong, T.V., Bui, H.H., Phung, D.Q., Venkatesh, S.: Activity recognition and abnormality detection with the switching hidden semi-Markov model. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 838–845 (2005)
9. Natarajan, P., Nevatia, R.: Online, real-time tracking and recognition of human actions. In: *Proceedings of IEEE Workshop on Motion and Video Computing (WMVC)*, Copper Mountain, pp. 1–8 (2008)
10. Du, Y., Chen, F., Xu, W.: Human interaction representation and recognition through motion decomposition. IEEE Signal Process. Lett. **14**, 952–955 (2007)
11. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: *Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR)*, vol. 3, pp. 32–36 (2004)
12. Laptev, L., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8 (2008)
13. Foroughi, H., Naseri, A., Saberi, A., Yazdi, H.S.: An eigen space-based approach for human fall detection using integrated time motion image and neural network. In: *Proceedings of IEEE 9th International Conference on Signal Processing (ICSP)*, pp. 1499–1503 (2008)
14. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res. **1**, 211–244 (2001)
15. Tipping, M.E.: The relevance vector machine. Adv. Neural Inf. Process. Syst. pp. 652–658 (2000)
16. Bodor, R., Jackson, B., Papanikolopoulos, N.: Vision-based human tracking and activity recognition. In: *Proceedings of the 11th Mediterranean Conference on Control and Automation*, vol. 1, pp. 18–20 (2003)
17. Ribeiro, P.C., Santos-Victor, J.: Human activity recognition from video: modeling, feature selection and classification architecture. In: *Proceedings of the International Workshop on Human Activity Recognition and Modeling (HAREM)*, Oxford, UK, vol. 1, pp. 61–70 (2005)
18. Ben-Arie, J., Wang, Z., Pandit, P., Rajaram, S.: Human activity recognition using multidimensional indexing. IEEE Trans. Pattern Anal. **24**(8), 1091–1104 (2002)
19. Kumari, S., Mitra, S.K.: Human action recognition using DFT. In: *Proceedings of the Third IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pp. 239–242 (2011)
20. Linbo, W., Jieyu, Z.: Video image segmentation based on Bayesian learning. J. Image Graph. 1073–1078 (2005)
21. Zhu, Y., Nayak, N.M., Roy-Chowdhury, A.K.: Context-aware activity recognition and anomaly detection in video. IEEE J. Sel. Top. Signal Process. **7**(1), 91–101 (2013)
22. Lin, W., Chu, H., Wu, J., Sheng, B., Chen, Z.: A heat-map-based algorithm for recognizing group activities in videos. IEEE Trans. Circuits Syst. Video Technol. **23**(11), 1980–1992 (2013)
23. Lee, S.Y., Sim, J.Y., Kim, C.S., Lee, S.U.: Correspondence matching of multi-view video sequences using mutual information based similarity measure. IEEE Trans. Multimed. **15**(8), 1719–1731 (2013)
24. Vijayakumar, V., Nedunchezhian, R.: Event detection in cricket video based on visual and acoustic features. J. Glob. Res. Comput. Sci. **3**(8), 26–29 (2012)
25. Goyani, M.M., Dutta, S.K., Raj, P.: Key frame detection based semantic event detection and classification using hierarchical approach for cricket sport video indexing. Adv. Comput. Sci. Inform. Technol. Commun. Comput. Inform. Sci. **131**, 388–397 (2011)
26. San Miguel, J.C., Martinez, J.M.: A semantic-based probabilistic approach for real-time video event recognition. Comput. Vis. Image Underst. **116**(9), 937–952 (2012)
27. UCSD Anomaly Detection Dataset. http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm
28. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using Matlab. Pearson Education, Inc., Upper Saddle River (2004)
29. Hinton, G.E., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
30. Cong, Y., Yuan, J., Tang, Y.: Video anomaly search in crowded scenes via spatio-temporal motion context. IEEE Trans. Inform. Forensics Secur. **8**(10), 1590–1599 (2013)
31. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N: Anomaly detection in crowded scenes. In: CVPR, vol. 249, pp. 250, (2010)
32. Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3449–3456 (2011)