Journal of Engg. Research Vol. 7 No. (4) December 2019 pp. 115-137

Taylor series based compressive approach and Firefly support vector neural network for tracking and anomaly detection in crowded videos

Avinash Ratre

Department of Electronics and Communication Engineering,
Delhi Technological University, India
avinashratre14@gmail.com

ABSTRACT

The application areas of multimedia content and computer vision analysis gains remarkable attention towards the motive to recognize the actions of the humans present in the video. Accordingly, crowd behavior analysis is an important topic owing to the significance of video surveillance in the public localities. This work introduces an anomaly detection (AD) model by introducing a tracking model and the optimization based classifier for the crowd video. The objects present in the video require tracking since the anomaly depends on the action of the object. This work proposes a hybrid tracking (HT) model by combining the Taylor series based predictive (TSP) tracking and the compressive tracking (CT) approach. The features are extracted from the tracked objects, and a feature vector is formed. Moreover, this work proposes the Firefly based support vector neural network (FSVNN) for the classification purpose. The weights of the proposed FSVNN classifier are trained with the genetic and the firefly algorithm. The performance of the proposed model is analyzed using three videos from the standard databases and is compared with the existing methods, such as self-organizing map (SOM), deep belief network (DBN), neural network (NN), and Firefly-based deep belief network (FDBN). The simulation results reveal that the proposed AD model with the FSVNN classifier attains overall better performance with the values of 0.97035, 1, and 0.96, for sensitivity, specificity, and accuracy, respectively than the comparative methods.

Keywords: Anomaly detection; crowd video; taylor series based predictive tracking; compressive tracking approach; firefly algorithm.

INTRODUCTION

Due to the increase in the video surveillance system, a large number of researchers in computer vision focus on excavating the valuable information from the monitored video data for public security. Problems, like representing the human behaviour in crowded scenes, recognizing the various human actions and, determining the associations of various events have more importance in research. Among them, anomaly detection (AD) from the crowded scenes has captured the interest of the researchers due to the advent of various technologies available in the multimedia applications (Arora & Singh, 2013 ; Ratre & Pankajakshan, V. 2018a,b). The aim of AD is to differentiate uncertain abnormal behaviours from certain normal ones accurately in real-time. Depending on the locality of anomaly, crowd behaviours are categorized into two groups, namely, local anomaly and global anomaly. In local anomaly, an abnormal behaviour is present in a local region of a frame and its surroundings are normal ones. In global anomaly, an abnormal behaviour is present in an entire frame; i.e., the frame contains several local anomalies. The different surveillance scenes contain different abnormal behaviours. Hence, the tasks of AD always vary for different surveillance scenes. As a result, several AD techniques have been introduced for various surveillance tasks. The study of the crowd behavior in the public places, such as parks, marathon race, movie theatres, mall, etc., is considered important to ensure the

safety of the public (Fradi *et al.,* 2017). Owners of the private and the public firms install a video surveillance system to monitor the activities of the persons (Laptev *et al.,* 2008; Narasimhan & Kamath, 2017). Public places have many persons moving in a group, and they are mentioned as the crowd (Du *et al.,* 2007). The people moving as a crowd in the video have different speeds and trajectories and do varying actions (Natarajan & Nevatia, 2008). To ensure the safety of the persons in the public, it is necessary to monitor the action of the public at each frame of the video (Sempena *et al.,* 2011; Soma & Gupta, 2017). The presence of the anomaly in the video can be monitored manually or automatically. The video obtained through the public monitoring of the persons is lengthy and, hence, requires constant manual monitoring for the detection of the anomaly. The manual processing requires more time and also leads to human errors (Narasimhan & Kamath, 2017). Hence, the AD model opts for the automation. The automation of the AD alerts the security while detecting the anomaly in the video (Chen *et al.,* 2017).

The persons moving in the crowd do actions, as walking, cycling, etc. The anomaly represents the abnormal behavior, such as intrusion, and fault of the person (Bensch *et al.,* 2017). Differentiation of the anomaly from the normal behavior acts as a prime concern of the AD model. Another challenge involved in the AD is the selection of the features for the detection. Traditional models have used trajectory-based algorithms and spatiotemporal domain-based algorithms (Zhou *et al.,* 2016) for the AD (Yuan *et al.,* 2017). Detection of the anomaly in the video is majorly classified as object based model and holistic based model (Fradi *et al.,* 2017). In the object based model, the crowds are segmented for the detection, while in the second approach, the crowd is considered as a whole. The use of the object based models is more suitable for the video with a small number of persons. The holistic based model analyzes the system as a whole, and thus, it is suitable to analyze the dynamics of the video frame than the action of the person.

The features representing the anomalies of the video have a complex structure, and hence suitable features need to be used (Narasimhan & Kamath, 2017). Literature works have included optimization based procedures for the AD. The literature works have used the Kernel-based classifiers (Sempena *et al.,* 2011), Support vector machines (SVM) (Natarajan & Nevatia, 2008) for the AD. Some works have included the classification algorithms based on the temporal approaches, namely, the Dynamic Time Warping (DTW) (Du *et al.,* 2007). The temporal based algorithms detect the anomalies in the video by analyzing the time or speed. Some of the works have performed the classification based on the Hidden Markov Models (HMM) (Laptev *et al.,* 2008) and Dynamic Bayesian Networks (DBN) (Foroughi *et al.,* 2008). The use of the SVM (Natarajan & Nevatia, 2008), Relevant Vector Machine (RVM) (Tipping, 2000), and Artificial Neural Networks (ANN) (Jun & Kim, 2012) has proved efficient in classifying the anomaly in the video. The K-nearest neighbor (K-NN) (Herbrich, R. 2002) based models also perform well in the crowded scenarios. Other NN based networks such as deep learning (Revathi & Kumar, 2017) and deep neural network (DNN) (Sabokrou *et al.,* 2017) provide better AD in the video.

This research work provides an innovative approach to the AD scheme. The proposed model performs the AD in three steps. These are 1) object detection, 2) object tracking, and 3) feature extraction. The proposed AD scheme primarily detects the presence of various objects in the crowded videos. Then, based on the proposed object tracking model, the movement of objects from one frame to another is detected. The proposed object tracking model uses the Taylor series based predictive (TSP) model (Yang *et al.*, 2001) and the compressive tracking (CT) model (Zhang *et al*., 2012). For tracking the objects in the frame, the proposed object tracking model uses the average results from both the TSP model and the CT model. The physical parameters, such as, mean and variance of the frame to frame distance traveled, speed, motion deviation, and frame occupancy, are extracted as quantitative values from the tracked objects. Along with that, the histogram based appearance features are also extracted from the tracked objects. Finally, the proposed FSVNN classifier detects the normal and the abnormal behavior of the crowds with the extracted features as its input. The metrics, sensitivity, specificity, accuracy, and the receiver operating characteristics (ROC) curve determine the performance of the proposed FSVNN classifier. The proposed model of AD has applications in various domains, like event detection in sensor networks, system health monitoring, fault detection, fraud detection, intrusion detection, and so on.

The major contribution of this work towards the AD is briefed below:

- Design of a novel hybrid approach for the object tracking. Here, the proposed Taylor series based compressive (TSC) approach finds the average of the results obtained from the Taylor series and the CT algorithm to track the objects in the video.
- Design of the FSVNN classifier by modifying the conventional support vector neural network (SVNN) classifier with the genetic algorithm (GA) and the Firefly algorithm (FF).

The rest of the paper is structured as follows: Section 1 introduces the AD scheme. Section 2 surveys various literary works dealing with the AD and different challenges involved in the AD purpose. Section 3 details about the proposed object tracking approach and the proposed FSVNN classifier. Section 4 contains the collection of simulation results of the proposed AD scheme. Section 5 concludes the work.

MOTIVATION

Literature survey

This section comes up with a survey of eight literature works compiling the AD in the crowd videos.

A. E. Gunduz *et al.* (2016) had proposed the density aware approach with the sparse features for the AD. The proposed model was trained with the sparse features as a separate HMM. Then, from each model AD was done. The proposed model detected the low-velocity and high-velocity anomalies without the training process. Besides, it does not detect the textural anomalies in the videos. H. Fradi *et al.* (2017) had proposed the AD model with the use of a set of visual descriptors. The proposed model used the spatio-temporal model for finding the visual descriptor set. The neighborhood interaction was based on the Delaunay triangulation. However, the work has utilized Crow Search Algorithm (CSA) for the optimal estimation, but only less concentration is given on the dynamic behavior of the crows.

C. Chen *et al.* (2015) had presented the AD algorithm with the acceleration feature. The acceleration feature had the gray-scale invariance properties for the three adjacent frames. A foreground extraction step used by the detection algorithm was designed by acceleration computation. The proposed detection approach utilized higher time for the computation. Soma & Gupta (2017) presented an approach for the abnormality detection with the input video image to be represented as feature matrices. The feature matrices are provided as the input maintains the spatial and temporal structure of the crowd video. The decomposition of the feature vectors was based on the low-rank and sparse components for the AD. The temporal and the spatial correlations between the components were not considered for evaluation. T. Chen *et al.* (2018) proposed the AD model with the low-level feature from the video. The motion energy model considered the Sum of Squared Differences (SSD) for the AD in the video. Sabokrou *et al.* (2017) had presented the fast AD model based on the localization of the video. They had proposed the cubic-patch-based model with the series of classifiers. The proposed model depends on the low likelihood of the video features and thus, performed the AD. The proposed model contains both the local and the global descriptors.

Narasimhan & Kamath (2017) presented the dynamic anomaly detection and localization system with the deep learning classifiers. The proposed model detected the local and the global anomalies present in the video by representing the frames in the video as the cubic patches. Sparse denoising autoencoder was employed in the model for reducing the computation speed of the model. S. Amraee *et al.* (2017) presented the AD system for the surveillance systems. The model had not considered the object detection and the object tracking for the feature extraction purpose. Besides, the training of the model was based on the HOG descriptors and a multivariate Gaussian model. The model assumed the background of an intended scene of the video as time invariant sequence.

Challenges

Various challenges pertained to the AD from the crowded video are briefed below:

- The major challenge involved in the AD scheme is the finding of the suitable features for the detection purpose (Gunduz *et al.,* 2016). Besides, the crowded videos have increased crowd density. Hence, for the improved accuracy, the detection scheme needs to overlook more crowd scenes/frames to detect the anomaly.
- The features representing the anomaly activity within the video have a complex structure and hence, training the classifier with the large sized features increases the complexity of the algorithm (Narasimhan & Kamath, 2017).

The proposed method extracts the histogram based features, which avoid hard decisions compared to edge based features. The integration of GA and the firefly algorithm with the SVNN classifier yields good results and decreases the complexity.

- The other factors affecting the AD in the crowded scenes are inter-object occlusion and the low resolution of the video frame (Soma & Gupta, 2017).
- Availability of training samples for the detection of the anomaly in each scenario is very scarce. The anomalies change frequently based on the video environment (Chen *et al.,* 2015).

The proposed model of AD works well even if the video has objects occlusion and low resolution frames.

- The crowd video with the varying actions, such as cycling, driving, and walking objects, has varying speed of movement within the frames. The proposed scheme needs to clearly distinguish the actions of the objects from the anomaly.

The proposed AD model clearly distinguishes the actions of the objects from the anomaly.

PROPOSED METHOD: TAYLOR SERIES BASED COMPRESSIVE APPROACH AND FIREFLY SUPPORT VECTOR NEURAL NETWORK FOR ANOMALY DETECTION

This work introduces the AD approach using the proposed FSVNN classifier from the crowd videos. Figure 1 shows the proposed AD model with the proposed object tracking model and the FSVNN classifier. Initially, the AD in the crowd video represents the abnormal activities. The proposed AD model gets each frame of the crowd video as the input. Then, the objects present in the video are detected by finding the center pixel location of the object. Tracking of the movement of objects from one frame to another is necessary to detect the activities of the objects in the video. This can be done through the proposed object tracking model. This model uses both the TSP and the compressive approach for the tracking of the movement of the objects from frame to frame. In the next step, the features of the tracking model for the training purpose are extracted. Various features, such as mean and variance of the frame to frame distance traveled, speed, motion deviation, frame occupancy and the histogram based features, are extracted from the tracking model. The proposed FSVNN classifier gets the features for the training phase and detects the presence of the anomaly in each frame of the video.

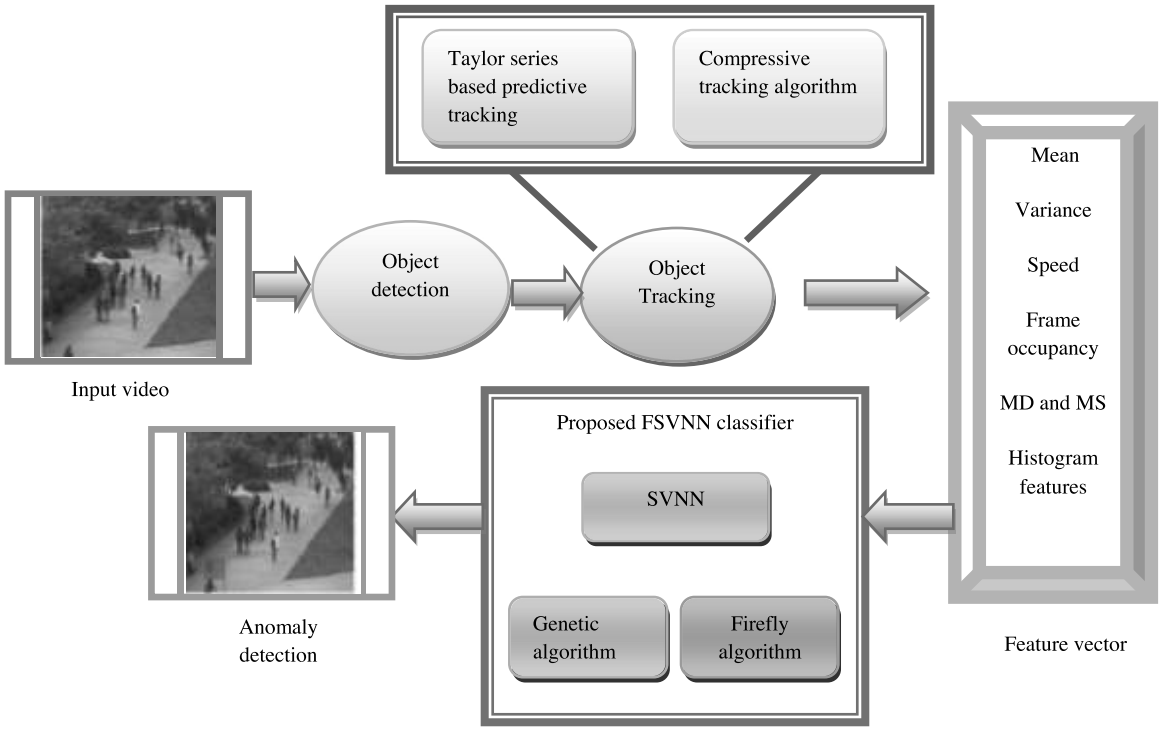


Figure 1. The proposed AD model in the crowd video based on the FSVNN classifier

Object detection

The initial step in the detection of the presence of the anomaly in the crowd video, is the detection of the objects in each frame of the video. The objects in the video indicate that the persons in the video are doing different activities. This work utilizes the video taken from the crowded environment. The crowded video contains many objects doing different activities, as walking, running, cycling, and so on. One object in the frame may overlap over the other object in the subsequent frames. Hence, the position of the objects in each frame needs to be detected for identifying the abnormal activities in the video. Consider the crowded video A with N frames. Equation (1) presents each frame present in the video A.

$$A = \{A_1, A_2, ..., A_a, ..., A_N\}; \quad 1 \le a \le N \tag{1}$$

where, the term A_a represents the a^{th} frame of the video A. The value, a varies between 1 and N. Consider the a^{th} frame of the video having M objects. Then, the objects of the a^{th} frame in the video are represented by the following expression,

$$O_a = \{O_a^1, O_a^2, ..., O_a^o, ..., O_a^M\}; 1 \le a \le N \ \ and \ \ 1 \le o \le M \tag{2}$$

where, the term O_a expresses the objects in the a^{th} frame. The term O_a^o indicates the o^{th} object in the a^{th} frame of the video A. The object's position in the frame is detected through the unsheathing of the object from the background of the video frame. The object detection uses a preset threshold for defining the presence of the object in the frame. The object's center position in the frame is located, and the difference between the positions of the object from one frame to another frame is found. Consider the object's center position in the a^{th} frame of the video as (x_a, y_a) and the object's position in the $(a+1)^{th}$ frame as (x_{a+1}, y_{a+1}). The difference in position of the object in the a^{th} and the $(a+1)^{th}$ frame is

expressed as $D(x,y)$. The presence of the object in the pixel location (x,y) depends on the preset threshold ξ and it is expressed by the following equation:

$$\begin{Bmatrix} if \quad D(x,y) \geq \xi \;\; ; \quad Location\ defines\ an\ object \\ else \;\; D(x,y) < \xi \;\; ; \quad Location\ is\ not\ an\ object \end{Bmatrix} \tag{3}$$

Object tracking based on the proposed Taylor series based compressive approach

The next step in the AD process is the tracking of the objects from one frame to another. In this paper, a hybrid object tracking model is proposed based on the predictive tracking model using the Taylor series (Yang *et al.,* 2001) and the spatial model that uses the compressive approach (Zhang *et al.,* 2012). The predictive tracking based on the Taylor series uses the second degree Taylor polynomial so that the accuracy in tracking can be improved. Meanwhile, the compressive approach utilized can track the object such that the image structure of objects is preserved.

Object tracking with the predictive approach: Taylor series

The TSP model predicts the motion of the object from one frame to another. The TSP model uses the second-degree polynomial for the prediction purpose. The position change of the object from the a^{th} frame to the $(a+1)^{th}$ frame based on the TSP model is indicated in the following equation:

$$T(a+1) \approx T(a) + \frac{T'(a)}{1!} + \frac{T''(a)}{2!} \tag{4}$$

where, the terms $T'(a)$ and $T''(a)$ express the first degree and the second degree polynomials. Equations (5) and (6) define the expression for $T'(a)$ and $T''(a)$ as

$$T'(a) = \frac{T(a) - T(a-Z)}{Z} \tag{5}$$

$$T''(a) = \frac{T'(a) - T'(a-Z)}{Z} = \frac{T(a) - 2T(a-Z) + T(a-2Z)}{Z^2} \tag{6}$$

where, the term Z represents the constant for the predictive approach. By applying the TSP approach, the position of the object o in each video frame is found. Equation (7) expresses the object's position in the video frames based on the TSP approach.

$$B_T = \left\{ \left(x^o_{T(1)}, y^o_{T(1)}\right), \left(x^o_{T(2)}, y^o_{T(2)}\right), ..., \left(x^o_{T(a)}, y^o_{T(a)}\right), ..., \left(x^o_{T(N)}, y^o_{T(N)}\right) \right\} \tag{7}$$

Where the term $\left(x^o_{T(a)}, y^o_{T(a)}\right)$ represents the o^{th} object's position in the a^{th} frame of the video based on the TSP model.

Object tracking with the compressive algorithm

The other model used in the proposed object tracking model is the CT algorithm. The CT algorithm considers both the low and the high dimensional features present in the frame. The object in the frame is tracked by applying the random matrix over the frame of the video. The random matrix involved in the tracking process is provided in Equation (8):

$$l = G.h \tag{8}$$

where, the term l represents the low dimensional space, $l \in \mathbb{R}^x$ in the video frame and the term h represents the

high dimensional space, $h \in \mathbb{R}^y$ in the video frame where $x \ll y$. The random matrix used for the CT is represented as $G \in \mathbb{R}^{xxy}$. The algorithmic steps involved in the CT algorithm are briefed below:

1) The a[th] frame of the video is provided as the input to the algorithm.

2) The low dimensionality features of the a[th] frame is extracted with the use of the set of image patches. The image patches are represented as $P_{\varsigma} = \left\{ r \middle\| B_C(r) - B_C(a-1) \right\| < \varsigma \right\}$ where, the term $B_C(a-1)$ represents the required tracking location of the $(a-1)^{th}$ frame.

3) The low dimensionality features are assigned as the input to the naive Bayes classifier (NBC). The NBC transforms each image patch into the log-likelihood ratio map which indicates chances that patches belong to object or non-object. The NBC, $C(l)$ provides the required tracking location B_C and it is represented as the sum of log-likelihood ratios of independent Gaussian distributed conditional distributions, $p(l_l | u = 1)$ and $p(l_l | u = 0)$ with their respective mean and variance parameters:

$$C(l) = \sum_{l=1}^{l1} \log\left(\frac{p(l_l | u = 1)}{p(l_l | u = 0)}\right) \tag{9}$$

where, the likelihood pdf are estimated directly/independently from the image patches, the term *u* represents the variable between 0 and 1, and *l*1 is the number of features.

4) In the next step, two sets of image patches are used for finding the tracking location, $B_C(a)$. The image patches are represented as, $P_{\psi} = \left\{ r \middle\| B_C(r) - B_C(a) \right\| < \psi \right\}$, and $P_{\zeta} = \left\{ r \middle| \varphi < \left\| B_C(r) - B_C(a) \right\| < \zeta \right\}$ with $\psi < \varphi < \zeta$.

5) Finally, the parameters of the classifier equation are updated based on the features obtained from the two image patches of previous step. The tracked location based on the CT algorithm is given as follows:

$$B_C = \left\{ \left(x^o_{C(1)}, y^o_{C(1)} \right), \left(x^o_{C(2)}, y^o_{C(2)} \right), ..., \left(x^o_{C(a)}, y^o_{C(a)} \right), ..., \left(x^o_{C(N)}, y^o_{C(N)} \right) \right\} \tag{10}$$

Tracking model based on the proposed Taylor series based compressive approach

The proposed TSC approach provides the final equation for the object tracking. The proposed TSC approach depends on the value of the tracking location obtained from both the TSP model and the CT algorithm. Equation (11) presents the equation for the object tracking by the proposed TSC approach. The proposed tracking model provides the accurate object's location in the video frame.

$$B = \frac{B_T + B_C}{2} \tag{11}$$

where, the term *B* presents the accurate object's location in the frame. The terms B_T and B_C are the tracked results obtained using Taylor series and compressive approaches.

Feature extraction

The feature extraction process extracts the necessary features from the objects tracked through the tracking model. In this work, features, such as mean, variance, speed, motion deviation, motion stopping, and frame occupancy, and relevant histogram based features as mean, variance, and color, are extracted from the tracked objects. The features extracted forms the feature vector. The features involved in the feature extraction procedure is explained as follows:

Mean of the frame to frame distance travelled

As the position of the object changes from frame to frame in the video, the distance between the object's location from one frame to another is computed. Then, the mean value of the distance travelled by each object between

two frames is measured and considered as a feature. The mean determining the frame to frame distance travelled is expressed as follows:

$$\rho(A_a, A_{a+1}) = \frac{\sum_{a=1}^{N} D}{N} \tag{12}$$

where, the term D represents the distance travelled by the object between the a^{th} and the $(a + 1)^{th}$ frame. The term N represents the total number of video frames.

Variance of the frame to frame distance travelled

Similarly, the variance of the distance measured is obtained for the object in each video frame. The variance of the frame to frame distance travelled by the object is given as follows:

$$\vartheta(A_a, A_{a+1}) = \frac{\sum_{a=1}^{N} (D - \rho)^2}{N} \tag{13}$$

where ρ is the mean.

Speed

The next feature involved in the feature extraction process is the speed. In the video, the objects move at varying speeds between the successive frames. The speed allows the researcher to determine the activity of the object. The speed of the movement by the object defines the ratio of the distance to the time. Equation (14) expresses the mathematical form of the speed.

$$Speed = \frac{D}{\delta} \tag{14}$$

where, the term D expresses the distance moved by the object from a frame to another, and the term δ represents the time utilized by the object to cover the distance D.

Motion deviation and motion stopping

The motion deviation (MD) and the motion stopping (MS) criteria features determine the change in the position of the object between the frames. MD defines the difference in the position of the object from the a^{th} and the $(a + 1)^{th}$ frame. Expression (15) defines the MD feature.

$$I = \sum_{a=1}^{N-1} \left\| (x_{a+1}, y_{a+1}) - (x_a, y_a) \right\| \tag{15}$$

where, the term (x_a, y_a) represents the center object's position in the a^{th} frame. The MS feature determine the position variation of the object from the previous frame, identifying the final object's location. Equation (16) gives the MS feature,

$$J = \sum_{a=1}^{N-1} \left\| (x_a, y_a) - (x_{a-1}, y_{a-1}) \right\| \tag{16}$$

where, (x_a, y_a) and (x_{a-1}, y_{a-1}) are the positions of the object in the a^{th} and $(a - 1)^{th}$ frame.

Frame occupancy

The frame occupancy feature defines the area occupied by the object in each frame. Since the objects move from one frame to another, the area occupied by the object in the frame differs from others. Measuring the area, the shape of

the object can be identified and thus, the object can be recognized easily. Hence, it is necessary to determine the area occupied by each object in the frame.

$$F^{O} = \sum_{b=1}^{K}\sum_{c=1}^{L} R_{bc} \tag{17}$$

$$R_{bc} = \begin{cases} 1 & ;if\ object \\ 0 & ;otherwise \end{cases} \tag{18}$$

Histogram based features

The histogram based features (Sergyan, 2008) like color, mean, and variance, are also extracted from the object. The histogram based features are briefed below.

i) Color

Color of an object is an important feature to be extracted, as it helps to determine whether the object is anomaly or not. Application of the probability histogram over the frame yields the required color histogram. Equation (19) presents the color histogram feature,

$$H(A_a) = \left[H^{Q} \quad H^{R} \quad H^{S} \right] \tag{19}$$

where, the term $H(A_a)$ indicates the color histogram of the frame A_a. The terms H^{Q}, H^{R} and H^{S} define the probability histogram of the red, the green, and the blue band, respectively.

ii) Mean

The brightness feature of the object in the frame is determined through the mean feature. The brightness of each pixel location of the object is determined through the calculation of the probability histogram and the gray level of the object. The mean of the object is represented as follows:

$$\overline{\mu}(A_a) = \sum_{k=0}^{i-1} k\, P(k) = \sum_{v=1}^{s}\sum_{w=1}^{s} \frac{A_a(v,w)}{s * s} \tag{20}$$

where, the term $A_a(v, w)$ represents the gray level of pixel position of the object in the a^{th} frame of the video and s is the size of the object in the frame. The term $P(k)$ represents the probability histogram.

iii) Variance

The contrast of the object is found through the variance measure. Equation (21) expresses the variance measure of the object.

$$\sigma^{2}(A_a) = \sum_{k=0}^{i-1} (\mu - \overline{\mu})^{2} P(k) \tag{21}$$

where, μ is gray level and $\overline{\mu}$ is the mean corresponding to the histogram features.

Construction of the feature vector

The feature vector is constructed with each features derived above. The feature vector is constituted by mean, variance, speed, frame occupancy, motion deviation and stopping criteria, and various histogram based features. The feature vector constructed with all the features is expressed below:

$$F = \left\{ F_1, F_2, ..., F_f, ..., F_m \right\} \tag{22}$$

where, the term F_f means the f^{th} feature in the feature vector. The features present in the feature vector vary between 1 and m.

Anomaly detection with the proposed FSVNN classifier

The next major step in the proposed work is design of the FSVNN classifier for the AD. The proposed FSVNN classifier uses the firefly algorithm (FF) (Arora & Singh, 2013) and Genetic Algorithm (GA) (Ludwig, *et al.,* 2014), in the SVNN classifier. The FF performs efficiently for the nonlinear and the multimodal problems. The FF has the flexible nature to be integrated with the other optimization algorithms. GA has better convergence for the solution with the larger space. Thus, the integration of GA and the FF for the optimization of the weights present in the SVNN classifier yields good results. The conventional SVNN classifier acts like a binary classifier. In the proposed FSVNN classifier, the weights for the optimization procedure are trained with the use of GA and the FF. The proposed FSVNN classifier utilizes a fitness function for the optimization process.

Fitness evaluation of the proposed FSVNN classifier

The proposed FSVNN classifier uses the maximization of fitness function for declaring the anomaly in the video frame. The fitness function depends on the Eigen value obtained from the weights of the proposed classifier. Equation (23) expresses the necessary maximisation fitness function of the proposed FSVNN classifier.

$$Fitness = \rho_{\max} + \rho_{\min} + \frac{\Gamma}{N}\sum_{a=1}^{N}\left|X_a - X_a^*\right| \tag{23}$$

where, the term X_a^* indicates the value of the ground truth information, the term X_a represents the classifier's output, the term Γ refers to the regularization factor and the term N represents the total video frames. The value of the $\rho_{\max}$ and $\rho_{\min}$ indicate the maximized and the minimized value of the Eigen function. The Eigen value for the fitness function is expressed by the following equation:

$$\rho = Eigen\left(W \times W^T\right) \tag{24}$$

$$\rho_{\max} = \max\left(\rho\right); \rho_{\min} = \min\left(\rho\right) \tag{25}$$

Where the term W represents the weight vector comprising of each weight in the proposed FSVNN classifier.

Architecture of the proposed FSVNN classifier

This section provides the architecture of the proposed FSVNN classifier in Figure 2. The Proposed FSVNN classifier has three layers. These are input layer, hidden layer and the output layer. The features of each object in the frame are fed into the input to the proposed FSVNN classifier. Since there are m features, the proposed FSVNN has m input layers. The features provided as the input are trained with the weights provided in the hidden layer. The proposed FSVNN classifier provides weights at both the input and the hidden layer. The input layer provides m weights for each feature input and the hidden layer provides a single hidden layer weight. The output provided in the hidden and the output layer is modified with the addition of biases. The final equation to the proposed FSVNN classifier is expressed in the Equation (26).

$$\mathrm{X_a} = \mathrm{W_\chi} \times \log\mathrm{sig}\left[\left(\sum_{\mathrm{f}=1}^{\mathrm{m}} \mathrm{F_f} * \mathrm{W_f}\right) + \mathrm{q_1}\right] + \mathrm{q_2} \tag{26}$$

where, the term X_a refers to the output to be classified under the FSVNN classifier. The term W_χ represents the weight provided by the hidden layer and the term W_f represents the f^{th} weight of the input layer. The terms q_1 and q_2 indicate the bias present in the FSVNN classifier. The term F_f represents the input feature given to the classifier.

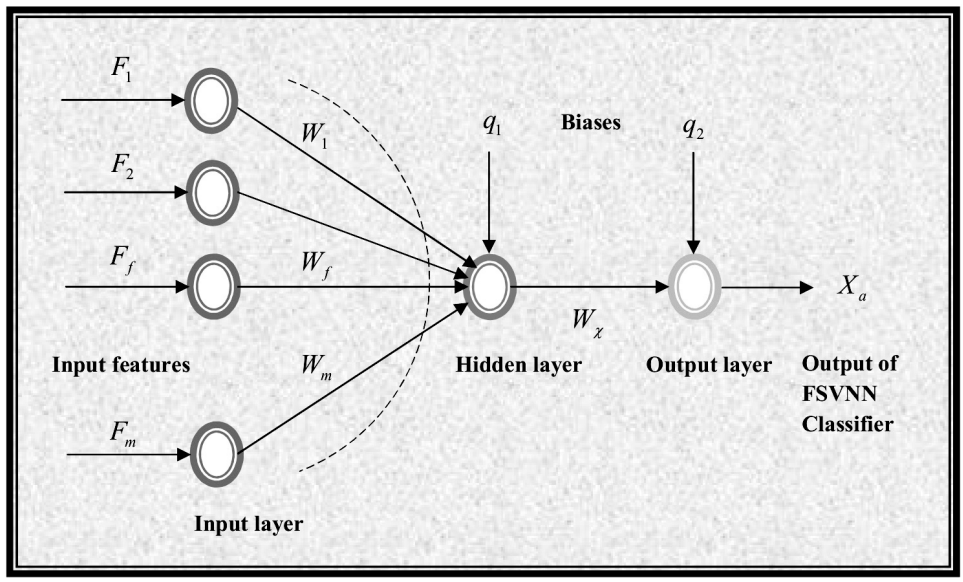


Figure 2. Architecture of the FSVNN classifier

3.4.3 Training phase: Training of the proposed FSVNN classifier with the features

The weights present in the proposed FSVNN classifier are trained with the features extracted from the tracked objects in the video frame. The *m* features extracted from the tracked objects in each video frame are fed into the input to the proposed FSVNN classifier. The weights at the input of the proposed classifier are trained with GA and the FF.

Algorithmic procedure of the FSVNN:

The general steps pertained to the training of the proposed FSVNN classifier to find the required input weights are explained as follows:

Step 1: Initialization: Initially, the weights present in the FSVNN classifier are randomly initialized. For the a^{th} video frame, *m* features are extracted. This requires *m* input weights for the AD. The randomly initialized weights are expressed as follows:

$$W^a = \left\{ W_f^a \quad ; 1 \le f \le m \right\} \tag{27}$$

where, W_f^a represents f^{th} weight for the a^{th} frame. The weights for the training process are initialized as the chromosomes.

Step 2: Evaluation of the fitness function: In this step, the fitness of the initialized weights is calculated. The fitness as the maximization function is expressed in the Equation (23).

Step 3: GA based weight update: GA finds the optimized weight based on three steps, such as selection, crossover, and mutation. The weight update based on the GA is expressed as follows:

$$W_a^{crossover}\left(t+1\right) = \alpha\, W_p\left(t\right) + \left(1-\alpha\right) W_q\left(t\right) \tag{28}$$

$$p = round\left[\left(E-1\right) \times \frac{e^{\eta\gamma_1} - 1}{e^{\eta} - 1} + 1 \right] \tag{29}$$

$$q = round\left[(E-1) \times \frac{e^{\eta\gamma_2} - 1}{e^{\eta} - 1} + 1 \right] \tag{30}$$

Where the terms p and q represent the indices of chromosomes. The term η refers to the selective pressure. The term α represents the constant for the weight update and E is the size of the population. γ_1 and γ_2 are random numbers in the range [0,1].

Step 4: FF based weight update: In this step, the weight of the proposed FSVNN classifier is updated based on the FF. The FF performs the updatation based on the attractiveness to another parameter. Equation (31) presents the weight update based on the Firefly.

$$\mathrm{W}_{\mathrm{a}}^{\mathrm{FF}}(\mathrm{t}+1) = \mathrm{W}_{\mathrm{a}}^{\mathrm{FF}}(\mathrm{t}) + \mathrm{Y}_{\mathrm{a}}(\mathrm{t}+1) \tag{31}$$

$$Y_a(t+1) = Y_a(t) + \delta_0 \times e^{-\beta d_{a,a1}^2} \left(Y_{a1}(t) - Y_a(t) \right) + \gamma K_a(t) \tag{32}$$

where, the term $W_a^{FF}(t+1)$ represents the updated weight at time $(t+1)$ based on the FF. The term $\mathrm{Y}_{\mathrm{a}}(\mathrm{t}+1)$ represents the required updated position based on the attractiveness of the fireflies. The term δ_0 represents the attractiveness parameter at $d_{a,a1} = 0$ and $K_a(t)$ represents the vector of the Gaussian distributed random numbers drawn at time t. The terms γ and β are randomization parameter and light absorption parameter, respectively.

Step 5: Weight update for the proposed FSVNN classifier: The proposed FSVNN classifier takes the weight update from both the GA and the FF on the basis of their fitness function. In this step, the fitness of the weights is computed for both algorithms. If the fitness of GA is better than the FF, then the weight of the GA is used for the next iteration. Otherwise, the weight of the FF is utilized. Equation (33) expresses the weight update equation of the proposed FSVNN algorithm.

$$W_a(t+1) = \begin{cases} W_a^{crossover}(t+1) & ; if\ Fitness\left[W_a^{crossover}(t+1)\right] > Fitness\left[W_a^{FF}(t+1)\right] \\ W_a^{FF}(t+1) & ; Otherwise \end{cases} \tag{33}$$

where, the term $Fitness\left[W_a^{crossover}(t+1)\right]$ represents the fitness of the weights obtained by the GA, and the term $Fitness\left[W_a^{FF}(t+1)\right]$ indicates the fitness of the weights obtained through the FF.

Testing phase: Detection of the anomaly in the video from the trained data

Based on the optimal weights provided by the training phase, the AD is done in the testing phase. Here, the features of the test frame are fed into the input of the trained FSVNN classifier. The proposed FSVNN classifier provides the information about the presence of the anomaly in the frame. The output of the proposed FSVNN classifier is represented as follows:

$$X_a^{test} = \begin{cases} 1 & ; \quad Anomaly \\ 0 & ; \quad no\ anomaly \end{cases} \tag{34}$$

Pseudo code of the proposed FSVNN classifier

Figure 3 exhibits the pseudo code of the proposed FSVNN classifier employed in the AD system. The proposed FSVNN classifier is initially trained with the extracted features of the tracked objects. Then, the weights of the classifier are trained with GA and the FF. Then the proposed FSVNN classifier provides the required classified output for the test video at the testing phase.

Sl. no	Algorithm: Pseudo code of the proposed FSVNN classifier
1	**Input: F= Feature vector**
2	**Output:** X_a^{test} = Output of the classifier
3	**Begin**
4	**//Training phase**
5	**For (t=0: t= max)**
6	**Initialize** the weights of the FSVNN classifier
7	$W^a = \{W_f^a \quad ; 1 \le f \le m\}$
8	Find the fitness of the weights using the fitness equation (23)
9	Update the weight $W_a^{crossover}(t+1)$ using GA
10	Update the weight $W_a^{FF}(t+1)$ using the FF
11	Find the fitness for the updated weights
12	**If** $\left(Fitness\left[W_a^{crossover}(t+1)\right] > Fitness\left[W_a^{FF}(t+1)\right]\right)$
13	Update the weight of the FSVNN using the $W_a^{crossover}(t+1)$
14	$W_a(t+1) = W_a^{crossover}(t+1)$
15	**Else**
16	Update the weight of the FSVNN using the $W_a^{FF}(t+1)$
17	**End if**
18	$W_a(t+1) = W_a^{FF}(t+1)$
19	Train the classifier with the updated weight $W_a(t+1)$
20	Find the output of the classifier based on the equation (26)
21	**End for**
22	**End**
23	**// Testing phase**
24	Provide the features of the test video
25	Find X_a^{test} based on the equation (34)
26	**End**

Figure 3. Pseudo code of the proposed FSVNN classifier for the AD from the crowded videos

RESULTS AND DISCUSSION

The simulation results of the proposed TSC based tracking and proposed FSVNN classifier are briefed in this section. The metrics, such as sensitivity (SEN), specificity (SPE), and accuracy (ACC), MOTP, and ROC curve, measure the performance of the FSVNN model.

Experimental setup

The experimentation of the proposed FSVNN model requires a PC with Windows 10 OS, 4 GB memory, and Intel i3 processor. The simulation work is completed using the MATLAB tool, and various simulation results are obtained.

Database description

The experimentation of the proposed AD model with the FSVNN classifier, uses three videos from the standard databases. Video 1 for the experimentation is utilized from the UCSD-Peds1 dataset (Dataset, 2017a). Video 1 contains the moving of the pedestrians, bicycles, carts, etc. along the walkway. Video 2 is obtained from the dataset (Dataset, 2017b), which comprises persons running a marathon race. Video 3 (Dataset, 2017c) provides the moving persons, vehicles along the road, obtained from the publicly available resource.

	Initial AD duration in seconds	Ground truth object number of anomalous activities
Video 1	2.00	9
Video 2	1.07	7
Video 3	0.90	12

Evaluation metrics

The evaluation metrics, such as SEN, SPE, and the ACC analyze the quantitative performance of the proposed FSVNN classifier involved in the AD. The mathematical expression of each metric involved in the performance analysis is briefed below,

Sensitivity: SEN metric measures the true positives provided by the AD model. The expression for the SEN regarding the true positive and the false negative is expressed below:

$$Sensitivity = \frac{True^{+}}{True^{+} + False^{-}} \tag{35}$$

where, $True^{+}$ is the true positives, and $False^{-}$ is the false negatives.

Specificity: The SPE measure provides the evaluation of the true negatives achieved by the AD model. The expression for the SPE is expressed as follows:

$$Specificity = \frac{True^{-}}{True^{-} + False^{+}} \tag{36}$$

where, $True^{-}$ is the true negatives, and $False^{+}$ is the false positives.

Accuracy: The ACC metric measures the closeness of the model to achieve the true value. Equation (37) expresses the ACC metric.

$$Accuracy = \frac{True^{+} + True^{-}}{True^{+} + True^{-} + False^{+} + False^{-}} \tag{37}$$

Multiple Object Tracking Precision (MOTP): The MOTP metric for the AD scheme defines the ratio of the total number of matched object hypothesis in the video frame to the total number of matches created in the frame.

$$MOTP = \frac{\sum_a Hypothesis^N}{\sum_a Matches^N} \tag{38}$$

Comparative models

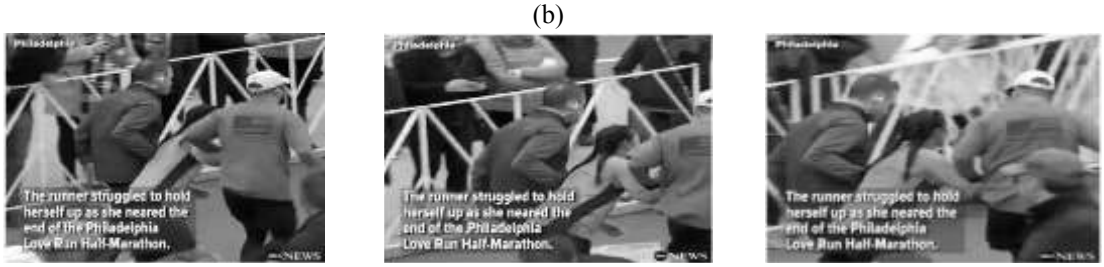
The proposed AD model with the FSVNN classifier is compared with the conventional models, such as Self-organizing Map (SOM) (Giovanis, 2010), Deep Belief Network (DBN) (Hu, *et al.,* 2016), Neural network (NN) (Giovanis, 2010), and Firefly based DBN (FDBN) (Applied FF to train DBN). The comparative models are analyzed using the performance metrics. The comparative models have the same experimental settings as the proposed model while obtaining the experimental results.

Experimental results of the proposed FSVNN classifier

Figure 4 shows various experimental AD results of the proposed FSVNN classifier based AD system. The experimentation of the proposed work is done with the use of three videos, i.e., video 1, video 2, and video 3. Video 1 presents the video of the several pedestrians walking on the sideways. Video 2 presents the persons running the marathon. Video 3 presents cyclists, pedestrians moving on the road. Figures 4.a, 4.c, and 4.e show the input sample provided to the proposed AD scheme from video 1, video 2, and video 3, respectively. Figure 4.b presents the AD in the video 1. The movement of the cyclist on the walkway is detected as the anomaly by the proposed scheme for the video 1. Figure 4.d provides the various anomalies present in video 2. Similarly, Figure 4.f provides the anomalies of video 3.

(a)

(b)



(c)

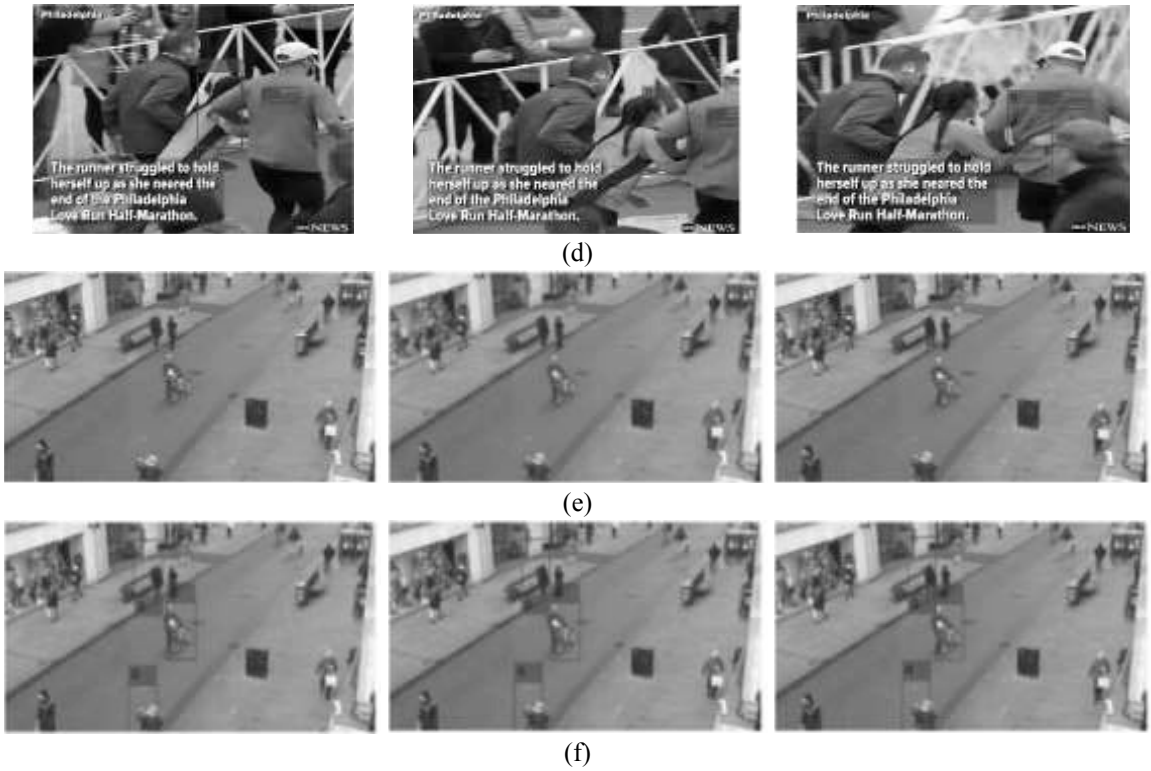


Figure 4. Experimental AD results of the proposed FSVNN classifier (a) Sample input images of the video 1, (b) Results of AD of (a), (c) Sample input images of the video 2, (d) Results of AD of (b), (e) Sample input images of the video 3, (f) Results of AD of (e)

Performance analysis of the proposed TSC based tracking approach and proposed FSVNN classifier

This section provides the performance analysis of the proposed TSC based approach for tracking in terms of MOTP and proposed FSVNN classifier for the AD in terms of metrics such as, SEN, SPE, and ACC. The performance of the proposed FSVNN classifier is also done through ROC curve analysis.

MOTP Analysis for TSC based tracking approach

The MOTP analysis is done between the varying number of objects in the video and the MOTP metric. In this work, the proposed hybrid tracking (HT) model is based on the Taylor series model and the CT model. The proposed HT model is analyzed by comparing the results with the compressed sensing (CS) approach (Zhang *et al.*, 2012) and the moving average (MA) model (Ratre *et al*., 2018b). Figure 5 shows the performance of the proposed HT Model in terms of MOTP curve. Figure 5.a presents the MOTP analysis for the video 1. For video 1 with two objects, the existing CS and the MA algorithms attain the MOTP value of 0.96954 and 0.8959. The proposed HT model attains the MOTP value of 0.98223 for video 1 with two objects. When video 1 has ten objects, the existing CS and the MA model attain the MOTP value of 0.9741 and 0.95228. The proposed HT model attains the MOTP value of 0.98934 for video 1 with ten objects.

Figure 5.b presents the performance analysis of the proposed HT model for video 2. For video 2 with two objects, the existing CS and the MA algorithm attain the MOTP value of 0.97 and 0.907. The proposed HT model attains the MOTP value of 0.98 for video 2 with two objects. When video 2 has 20 objects, the existing CS and the MA model attain the MOTP value of 0.97 and 0.980097. The proposed HT model attains the MOTP value of 0.98 for video 2

with 20 objects. Figure 5.c presents the performance analysis of the proposed HT model for video 3. For video 3 with ten objects, the existing CS and the MA algorithm attain the MOTP value of 0.98832 and 0.96371. The proposed HT model attains the MOTP value of 0.99353 for video 3 with ten objects. When video 3 has 20 objects, the existing CS and the MA model attain the MOTP value of 0.97749 and 0.95674. The proposed HT model attains the MOTP value of 0.98854 for video 3 with 20 objects. Here, the proposed HT model attains the maximum MOTP value more than that of the existing methods, since it takes the advantages of both TSP tracking and the CT approach. Also, the proposed HT model extracts the histogram based features, which avoid hard decisions compared to edge based features.

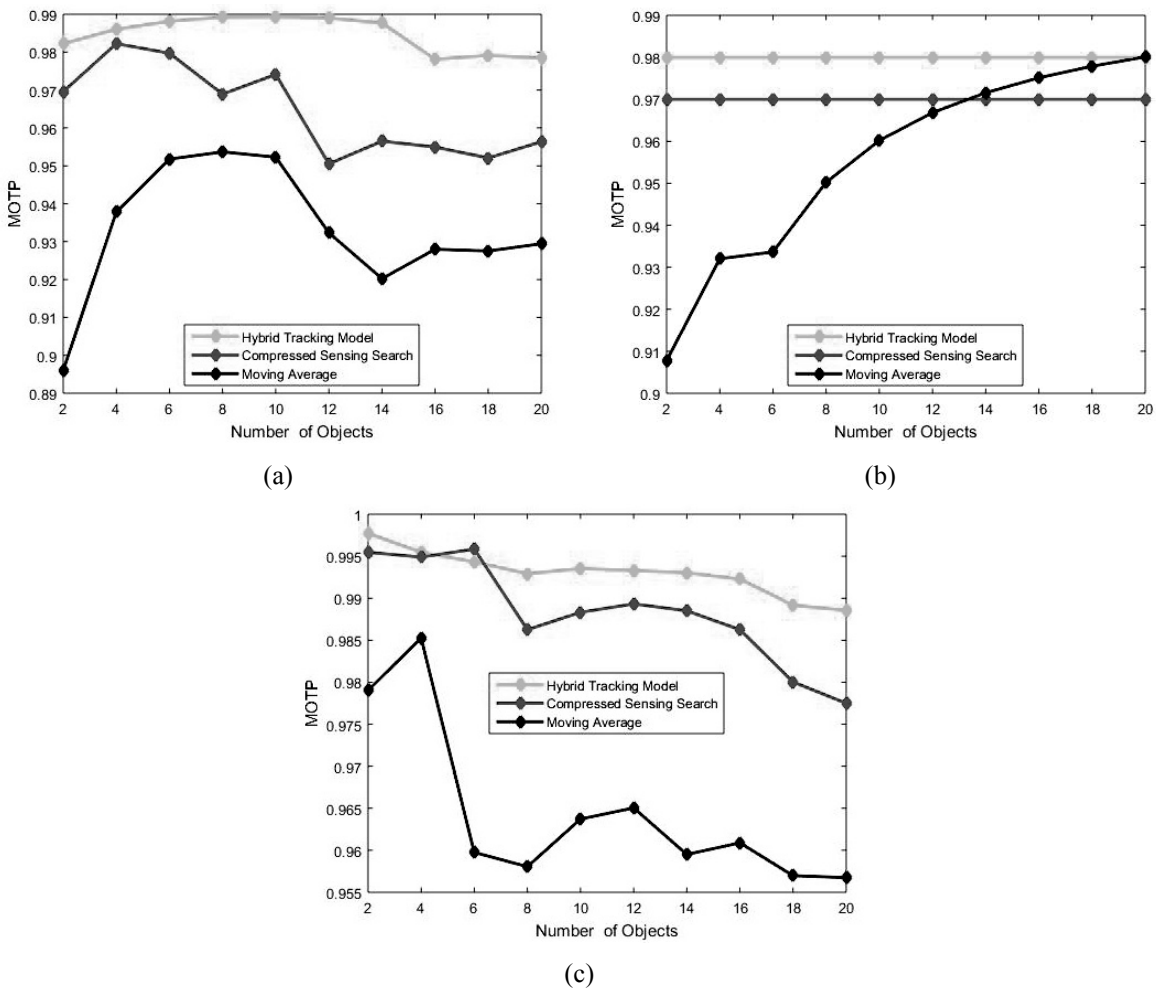


Figure 5. Performance of the proposed HT Model based on the MOTP curve for (a) video 1, (b) video 2, and (c) video 3.

Comparative Analysis

Here, the performance of the proposed FSVNN method is analyzed and compared with the existing methods, such as SOM, DBN, NN, and FDBN.

ROC curve Analysis

The performance of the proposed FSVNN classifier used for the AD is analyzed using the ROC curve. The ROC curve analysis is done between the True Positive Rate (TPR) and the False Positive Rate (FPR). Figure 6.a presents

the ROC curve for video 1. For the FPR value of 0.1, the conventional SOM, DBN, NN, and FDBN models attain the TPR value of 0.38764, 0.4256, 0.6025, 0.6525, and 0.7025. For the proposed FSVNN model, TPR is increased to 0.7025. When the FPR value is 0.5, the TPR value of the conventional SOM, DBN, NN, and FDBN models is 0.5805, 0.61822, 0.8025, and 0.8525. The proposed FSVNN model attains the improved TPR value of 0.9025 for the FPR value of 0.5. Figure 6.b presents the ROC curve for video 2. For the FPR value of 0.5, the conventional SOM, DBN, NN, and FDBN models attain the TPR value of 0.52675, 0.56175, 0.6878, and 0.8525. For the FPR value of 0.5, the proposed FSVNN classifier attains TPR of 0.9025. Figure 6.c presents the ROC curve analysis of the proposed model for video 3. For the FPR value of 0.5, the conventional SOM, DBN, NN, and FDBN models attain the TPR value of 0.602, 0.642, 0.8025, and 0.8525, respectively. Here, FSVNN classifier performs better than competing classifiers.

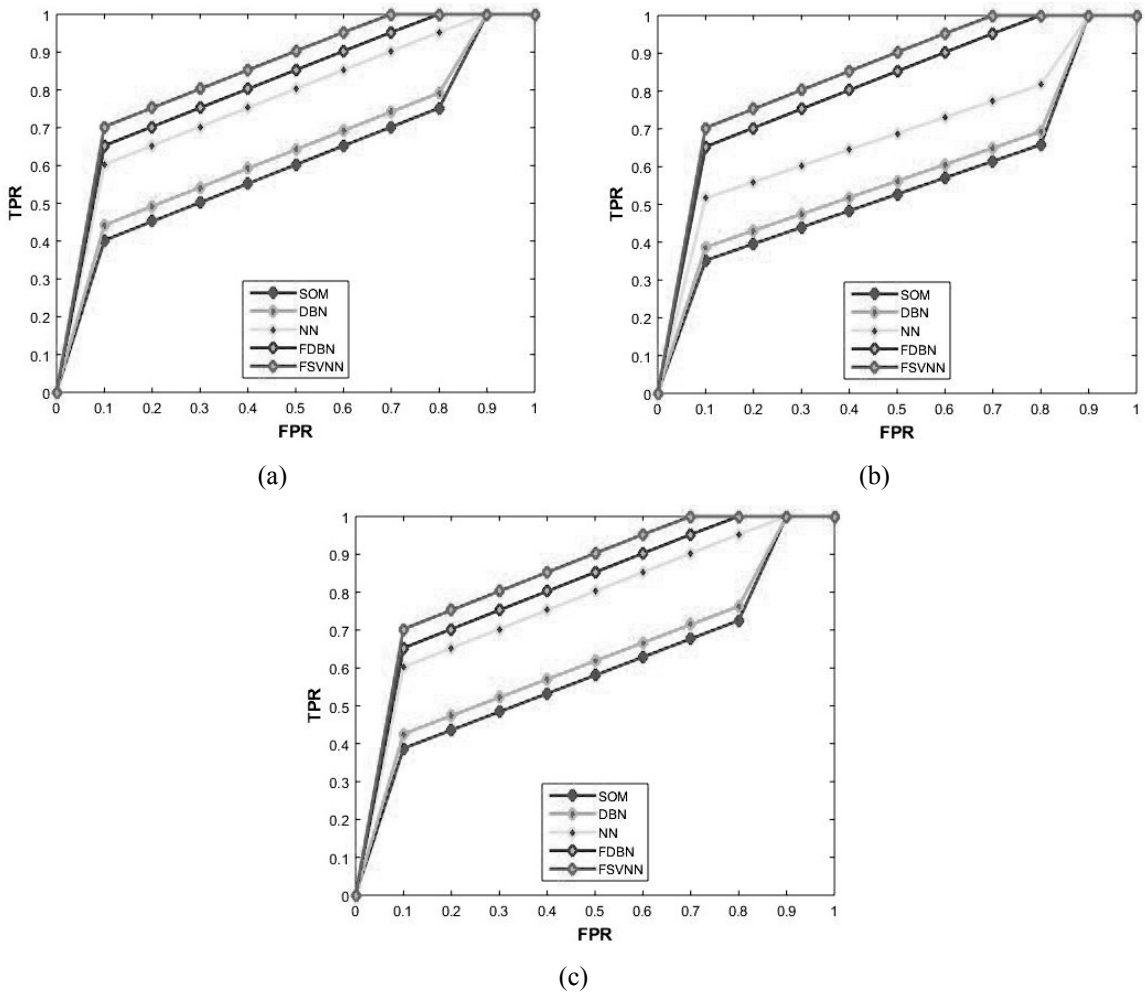


Figure 6. Performance of the proposed FSVNN classifier based on the ROC curve for (a) video 1, (b) video 2, and (c) video 3.

Analysis for 80 % training data

Figure 7 shows the performance analysis and comparative evaluation of the proposed FSVNN classifier for 80 % training data. Figure 7.a shows the SEN analysis for the proposed FSVNN classifier. The conventional SOM, DBN, NN, and FDBN models attain the SEN value of 0.93103, 0.89655, 0.89655, and 0.93103, respectively, for video 1. The proposed FSVNN classifier attains the overall improved value of SEN value of 0.97034 for video 1. Figure 7.b

shows the SPE analysis of the proposed FSVNN classifier. For video 1, the conventional SOM, DBN, NN, and FDBN models attain the SPE value of 0.5, 0.5, 0.5, and 1, respectively, for video 1. The proposed FSVNN model attains the SPE value of 1 for the three videos. Figure 7.c presents the performance analysis of the proposed model based on the ACC metric. For video 1, the conventional SOM, DBN, NN, and FDBN models attain the ACC value of 0.90322, 0.8709, 0.87096, and 0.93548, respectively. The proposed FSVNN model attains ACC value of 0.96 for the 80 % training data for video 1.

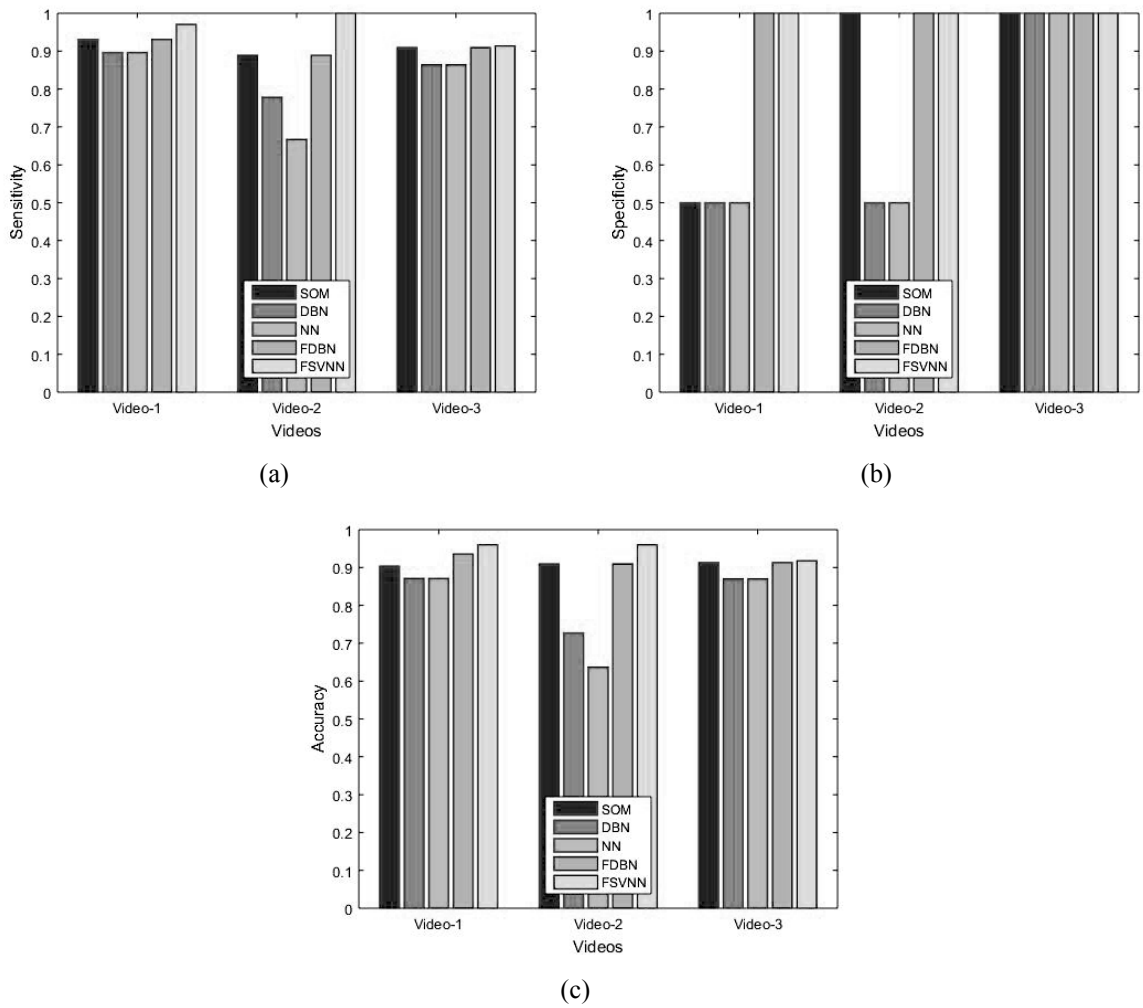


Figure 7. Performance of the proposed FSVNN classifier for 80 % training data
(a) SEN, (b) SPE, and (c) ACC.

Analysis for 90 % training data

Figure 8 shows the performance analysis and comparative evaluation of the proposed FSVNN classifier for 90% training data. Figure 8.a shows the SEN analysis for the proposed FSVNN classifier. The conventional SOM, DBN, NN, and FDBN models attain the SEN value of 0.93103, 0.89655, 0.89655, and 0.93103, respectively, for video 1. The proposed FSVNN classifier attains the overall improved value of SEN value of 0.93568 for video 1. Figure 8.b shows the SPE analysis of the proposed FSVNN classifier for 90 % training data. Increasing the training to 90 % shows that the proposed model attains negligible changes in the SPE analysis for each input video. The proposed

FSVNN model achieved overall high SPE value of 1 for video 1, video 2, and video 3, respectively. Figure 8.c presents the performance analysis of the proposed model based on the ACC metric for 90 % training data. For video 3, the conventional SOM, DBN, NN, and FDBN models attain the ACC value of 0.913043, 0.86956, 0.86956, and 0.91304, respectively. The proposed FSVNN model attains ACC value of 0.917608 for 90 % training of video 3.

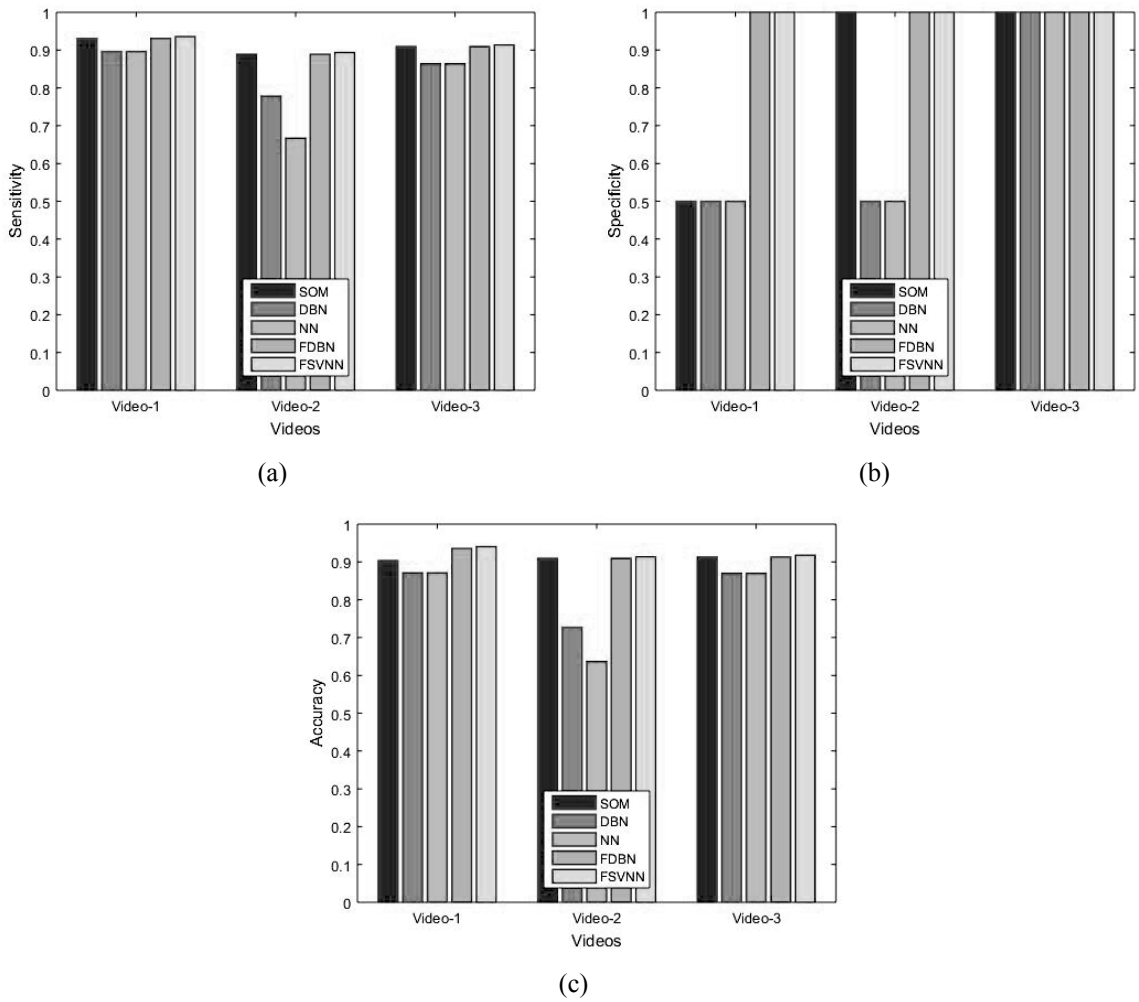


Figure 8. Performance of the proposed FSVNN classifier for 90 % training data (a) SEN, (b) SPE, and (c) ACC.

Discussion

The performance of the proposed AD scheme with the FSVNN classifier and the HT model is analyzed and compared with that of the various comparative models. From Table 1, it is demonstrated that the proposed AD with the FSVNN classifier performs better than the other comparative models. Table 1 contains the values of metrics obtained by the comparative models for the 80 % training of the video 1. The proposed AD scheme with the FSVNN classifier attains SEN, SPE, and ACC values of 0.97035, 1, and 0.96, respectively.

Table 1. Comparative evaluation of the proposed AD with the FSVNN

Performance metrics	Comparative models for the video 1				
	SOM	**DBN**	**NN**	**FDBN**	**Proposed FSVNN**
SEN	0.931034	0.896552	0.896552	0.931034	**0.970345**
SPE	0.5	0.5	0.5	1	**1**
ACC	0.903226	0.870968	0.870968	0.935484	**0.96**

CONCLUSION

This paper presents the AD model by proposing the HT model and the FSVNN classifier. The proposed AD model primarily develops a HT model based on the TSP model and CT approach for tracking the objects in the video. Then, various features, such as speed, mean, variance, motion deviation and stopping, frame occupancy and several histogram based features, are extracted from tracked objects in the frames. Then the features are fed to the input layers of the proposed FSVNN classifier for the training purpose. The proposed FSVNN classifier modifies the existing SVNN classifier with the GA and the FF. The simulation of the proposed AD model based on the FSVNN classifier utilizes three videos from the standard datasets. The performance of the proposed model is analyzed and compared with the conventional models, such as SOM, DBN, NN, and FDBN, respectively. The MATLAB simulation results demonstrate that the proposed AD model with the FSVNN classifier attains the effective performance with the values of 0.97035, 1, and 0.96, for SEN, SPE, and ACC, respectively.

REFERENCES

Amraee, S., Vafaei, A., Jamshidi, K. & Adibi, P. 2018. Anomaly detection and localization in crowded scenes using connected component analysis, *Multimedia Tools and Applications,* **77**(12): 14767-14782.

Arora, S. & Singh, S. 2013. The Firefly optimization algorithm: convergence analysis and parameter selection, *International Journal of Computer Applications*, **69**(3): 48-52.

Bensch, R., Scherf, N., Huisken, J., Brox, T. & Ronneberger, O. 2017. Spatiotemporal deformable prototypes for motion anomaly detection, *International Journal of Computer Vision,* **122**(3): 502-523.

Chen, C., Shao, Y. & Bi, X. 2015. Detection of anomalous crowd behavior based on the acceleration feature, *IEEE Sensors Journal,* **15**(12): 7252-7261.

Chen, T., Hou, C., Wang, Z. & Chen, H. 2018. Anomaly detection in crowded scenes using motion energy model, *Multimedia Tools and Applications*, **77**(11): 14137-14152.

Dataset., 2017a. Dataset 1 (UCSD ped1) from "http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm".

Dataset., 2017b. Dataset 2 (Philadelphia Love Run Half Marathon) from "https://youtu.be/TvViJO7Pt44".

Dataset., 2017c. Dataset 3 (Town centre dataset-Univ. of Oxford) from "https://medium.com/@madhawavidanapathirana/real-time-human-detection-in-computer-vision-part-2-c7eda27115c6"

Du, Y., Chen, F. & Xu, W. 2007. Human interaction representation and recognition through motion decomposition, *IEEE Signal Processing Letters,* **14**: 952-955.

Foroughi, H., Naseri, A., Saberi, A. & Yazdi, H.S. 2008. An Eigen space-based approach for human fall detection using integrated time motion image and neural network, *In Proceedings of the IEEE 9th International Conference on Signal Processing (ICSP),* 1499– 1503, Beijing, China.

Fradi, H., Luvison, B. & Pham, Q.C. 2017. Crowd behavior analysis using local mid-level visual descriptors, *IEEE Transactions on Circuits and Systems for Video Technology,* **27**(3): 589-602.

Giovanis, E. 2010. Application of logit model and self organizing maps (SOMs) for the prediction of financial crisis periods in US economy, *Journal of Financial Economic Policy.* **2**(2): 98-125.

Gunduz, A.E., Ongun, C., Temizel, T.T. & Temizel, A. 2016. Density aware anomaly detection in crowded scenes, *IET Computer Vision,* **10**(5): 374-381.

Herbrich, R. 2002. Learning Kernel Classifier: Theory and Algorithm, *The MIT Press, Cambridge,* 384 pages, ISBN: 9780262256339.

Hu, X., Hu, S., Huang, Y., Zhang, H. & Wu, H. 2016. Video anomaly detection using deep incremental slow feature analysis network, *IET Computer Vision* **10**(4): 258-265.

Jun, B. & Kim, D. 2012. Robust face detection using local gradient patterns and evidence accumulation, *Pattern Recognition,* **45**(9): 3304-3316.

Laptev, L., Marszalek, M., Schmid, C. & Rozenfeld, B. 2008. Learning realistic human actions from movies, *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1–8, Anchorage, AK, USA.

Ludwig, O., Nunes, U. & Araujo, R. 2014. Eigen value decay: A new method for neural network regularization, *Neurocomputing*, **124**: 33-42.

Narasimhan, M.G. & Kamath, S. 2018. Dynamic video anomaly detection and localization using sparse denoising autoencoders, *Multimedia Tools and Applications*, **77**(11): 13173-13195.

Natarajan, P. & Nevatia, R. 2008. Online, real-time tracking and recognition of human actions, *In Proceedings of IEEE Workshop on Motion and Video Computing (WMVC), Copper Mountain,* 1– 8, Copper Mountain, CO, USA.

Ratre, A. & Pankajakshan, V. 2018a. Tucker tensor decomposition based tracking and Gaussian mixture model for anomaly localization and detection in surveillance videos, *IET Computer Vision,* **12**(6): 933-940.

Ratre, A. & Pankajakshan, V. 2018b. Tucker visual search-based hybrid tracking model and fractional Kohonen self-organizing map for anomaly localization and detection in surveillance videos, *The Imaging Science Journal,* **66**(4): 195-210.

Revathi, A.R. & Kumar, D. 2017. An efficient system for anomaly detection using deep learning classifier, *Signal, Image and Video Processing,* **11**(2): 291-299.

Sabokrou, M., Fathy, M., Moayed, Z. & Klette, R. 2017a. Fast and accurate detection and localization of abnormal behavior in crowded scenes, *Machine Vision and Applications,* **28**(8): 965-985.

Sabokrou, M., Fayyaz, M., Fathy, M. & Klette, R. 2017b. Deep-Cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes, *IEEE Transactions on Image Processing*, **26**(4): 1992-2004.

Sempena, S., Maulidevi, N. & Aryan, P.R. 2011. Human action recognition using dynamic time warping, *In Proceedings of the IEEE International Conference on Electrical Engineering and Informatics (ICEEI),* 1–5, Bandung, Indonesia.

Sergyan, S. 2008. Color histogram features based image classification in content-based image retrieval systems, *2008 6th International Symposium on Applied Machine Intelligence and Informatics,* 221-224, Herlany, Slovakia.

Soma, B. & Gupta, V. 2017. Abnormality detection in crowd videos by tracking sparse components, *Machine Vision and Applications,* **28**(1-2): 35-48.

Tipping, M.E. 1999. The relevance vector machine, *Advanced Neural Information Processing System,* 652–658.

Yang, Q., Zhang, H.H. & Zhang, H. 2001. Taylor series prediction: a cache replacement policy based on second-order trend analysis, *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Maui, HI, USA,* 7.

Yuan, Y., Wang, D. & Wang, Q. 2017. Anomaly detection in traffic scenes via spatial-aware motion reconstruction, *IEEE Transactions on Intelligent Transportation Systems,* **18**(5): 1198-1209.

Zhang, K., Zhang, L. & Yang, M. 2012. Real-time compressive tracking, *In Proceeding ECCV'12 Proceedings of the 12th European Conference on Computer Vision, Florence,* Italy, 864-877.

Zhou, S., Shen, W., Zeng, D., Fang, M., Wei, Y. & Zhang, Z. 2016. Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes, *Signal Processing: Image Communication*, 47: 358-368.

Submitted: 20/11/2017
Revised: 02/07/2018
Accepted: 03/07/2018

سلسلة تايلور القائمة على نهج التتبع الضاغط والشبكة العصبية لمتجهات الدعم المعتمدة على خوارزمية اليراعات المضيئة لتتبع والكشف عن الخلل في أشرطة الفيديو الجماهيرية

أفيناش راتر
قسم هندسة الإلكترونيات والاتصالات، جامعة دلهي التكنولوجية، طريق باوانا، دلهي، الهند

الخلاصة

تحظى مجالات تطبيق محتوى الوسائط المتعددة وتحليل مشاهد الكمبيوتر باهتمام ملحوظ نحو التعرف على حركات الأشخاص الموجودين داخل الفيديو. وفقاً لذلك، يُعد تحليل سلوك الجماهير موضوعاً مهماً نظراً لأهمية المراقبة بالفيديو في الأماكن العامة. يقدم هذا العمل نموذجاً للكشف عن الخلل من خلال تقديم نموذج تتبع ومُصَنِف يرتكز على تحسين مقاطع الفيديو الجماهيرية. تتطلب الأجسام الموجودة في الفيديو تتبعاً لأن الخلل يعتمد على حركة تلك الأجسام. يقترح هذا العمل نموذج تتبع هجين من خلال الجمع بين سلسلة تايلور القائمة على التتبع التنبئي ونهج التتبع الضاغط. تم استخراج السمات من الأجسام التي تم تتبعها، وتم تشكيل متجه الخصائص. علاوة على ذلك، يقترح هذا العمل الشبكة العصبية لمتجهات الدعم المعتمدة على خوارزمية اليراعات المضيئة (FSVNN) بغرض التصنيف. تم تجريب أوزان مصنف FSVNN المُقترح على الخوارزمية الجينية وخوارزمية اليراعات المضيئة. تم تحليل أداء النموذج المُقترح باستخدام ثلاثة مقاطع فيديو من قواعد البيانات القياسية ومقارنتها بالطرق الحالية، مثل SOM وDBN وNN وFDBN. من نتائج المحاكاة، تبين أن النموذج المُقترح للكشف عن الخلل مع مصنف FSVNN قد حققا أداءً أفضل بشكل عام من الطرق المقارنة بقيم 0.97035 و1 و0.96 للحساسية والنوعية والدقة على التوالي.