Hybrid Deep Learning Technique with One Class Svm for Anomaly Detection in Crowded Environment

N.K. PRIYADHARSINI[1*], R. KAVITHA[2*], A. KALIAPPAN[3*], DR. D. CHITRA[4*]

*[*1]Department of Computer Science and Engineering, P.A. College of Engineering and Technology, Pollachi, Coimbatore, India.*

*[*2]Department of Computer Science and Engineering, P.A. College of Engineering and Technology, Pollachi, Coimbatore, India.*

*[*3]Department of Computer Science and Engineering, P.A. College of Engineering and Technology, Pollachi, Coimbatore, India.*

*[*4]Department of Computer Science and Engineering, P.A. College of Engineering and Technology, Pollachi, Coimbatore, India.*

ABSTARCT

Nowadays, the analysis of abnormal events becomes more and more exhausting due to the divine use of surveillance cameras. Reliability of normal and abnormal samples is generally different in practice. In the existing work, developed machine learning and swarm intelligence based approaches for anomaly detection by extracting spatiotemporal features from video sequences. This proposed abnormal event detection mechanism is obvious, but the tracking result is less precise. Some of the reasons are low quality video, system noise, small object, and other factors. In order to improve the precisions of the tracked object, this research work proposed a new hybrid deep learning and robust segmentation method for better and faster tracking result. In the proposed approach, a video frame is subjected to a series of operations to extract most salient information from it. Initially, a 2D variance plane is constructed to encode local spatio-temporal variations around each pixel in a video frame. The Improved Particle Swarm Optimization algorithm is then applied to isolate the most salient regions based on motion information in the 2D variance plane. Grey Level Co-occurrence Matrix is applied to the extracted salient pixels in the video. And then the segmentation process is done with the help of fuzzy c-means clustering of successive frames are exploited for pattern matching in a simple feature space. Finally developed a hybrid deep learning based on a pre-trained Convolution Neural Network and One-class SVM is trained with spatial features for robust classification of abnormal shapes. Thus the experimental result suggests that the proposed anomaly detection techniques outperform the existing techniques in context of accuracy and time complexity.

Keywords: *Crowded Scenes, Improved Particle Swarm Optimization (IPSO) Algorithm, Grey Level Co-Occurrence Matrix (GLCM), Fuzzy C-Means Clustering (FCM), Convolution Neural Network (CNN).*

INTRODUCTION

The analysis of abnormal events becomes more and more exhausting due to the divine use of surveillance cameras [8]. Recently, government agencies, businesses, and even schools are turning toward video surveillance as a means to increase public security. Video surveillance has been a key component in ensuring security at airports, banks, and correctional institutions. The goal of visual surveillance is not only to use cameras instead of human eyes, but also to accomplish the entire surveillance task as automatically as possible using video analysis [23].

Shape recognition methods require a list of objects and behaviour patterns that are anomalous. Unfortunately, this is not always possible, especially suspicious

https://doi.org/10.37896/pd91.4/91442

activities cannot be known in advance [7]. An alternative approach is based on learning normal behaviour from a video sequence exhibiting regular activity and then flag moving objects.

Reliability of normal and abnormal samples is generally different in practice. A machine learning and swarm intelligence based approaches for anomaly detection is developed for extracting spatiotemporal features from video sequences. This proposed abnormal event detection mechanism is obvious, but the tracking result is less precise. Some of the reasons are low quality video, system noise, small object and other factors. In order to improve the precisions of the tracked object a new hybrid deep learning and robust segmentation method is proposed for better and faster tracking result.

RELATED WORK

A Hidden Markov Models(HMM) to learn the normal and abnormal patterns [5] of Unix processes. These patterns can be used to detect an anomalies and known intrusion. Using experiments on the mail-sending system call data, demonstrate that construct concise and accurate classifiers to detect intrusion action.

An effective HMM based intrusion detection system [2] that improves the modeling time and performance by only considering the privilege transition flows based on the domain knowledge of attacks. The training with the HMM method is significantly faster than the conventional method trained with all data, without loss of detection performance.

Predictive model [9] capable of discriminating between normal and abnormal behaviour of network traffic. In the training phase, special attention is given to the initialization and model selection issues, that makes the training phase particularly effective. This system is able to classify network traffic in proportion to the number of features used for training HMM.

A transaction based probabilistic model [22] is developed to combine HMMs and feature aided tracking. The method is able to detect the modelled pattern of an asymmetric threat with a high performance as compared to a maximum likelihood based data mining technique. Performance analysis shows that the detection of HMMs improves with increase in the number of states in an HMM.

An algorithm that regards the video frames containing caption as a bag [14]. It detects, localizes and extracts video caption frames using Multiple-Instance Learning (MIL) automatically. The results show that the method can detect, localize and extract video caption frames.

System that is capable of extracting and modelling several objects in videos [14], in addition allows user to interact within a continuous learning setup. Multi cue learning approach presents rule based event detection and multiple feature tracking.

Omar et al. (2013) introduced an intrusion detection method. The perturbations of normal behaviour indicate a presence of intended or unintended induced attacks, faults, defects and others. This work presents a detection of an anomalies for applying supervised and unsupervised methods for managing the problem of an anomaly detection.

Anomaly detection methodology for data with latent dependency structure. As a particular instantiation [15], derive a hidden Markov anomaly detector that extends the regular One-Class SVM. The approach is non-convex, through a difference of convex functions algorithm. The empirical evaluation on artificial and real data from the domains of computational biology and computational sustainability shows, that the approach can achieve significantly higher anomaly detection performance than the regular one-class SVM.

Abnormal behaviour detection method using causality analysis and sparse reconstruction [24]. Effective representation of multiple-object behaviour, low level visual features and causality features are adopted. The low-level visual features, that included trajectory shape descriptor, speeded up robust features and histograms of optical flow, are used to describe properties of individual behaviour and causality features obtained by causality analysis are introduced to depict the interaction information among a set of objects.

An approach for creating ensembles of previously trained convolutional neural networks [18]. The approach allows to increase the performance of the image classification. The approach is able to outperform the standard perceptron and single convolutional neural network.

An efficient method for detection and localization of anomalies in videos [20]. Using Fully Convolutional Neural networks (FCNs) and temporal data, a pre-trained supervised FCN is transferred into an unsupervised FCN ensuring the detection of global anomalies in scenes. High performance in terms of speed and accuracy is achieved by investigating the cascaded detection as a result of reducing computation complexities. FCN-based architecture addresses two main tasks, feature representation and cascaded outlier detection.

Machine learning-based anomaly detection algorithm are employed to find malicious traffic [1] in a synthetically generated data set of Modbus or Transmission Control Protocol communication of a fictitious industrial scenario. The applied algorithms are SVM, Random Forest, K-NN and K-means clustering. Due to the synthetic data set, supervised learning is possible.

A method for abnormal behaviour detection is using deep learning [25]. The vision words are exploited to describe behaviour by the bag of words and in order to reduce feature dimensions, the Agglomerative Information Bottleneck approach is used for vocabulary compression. An adaptive feature fusion method is adopted to enhance the discriminating power of these features. A sparse representation is exploited to abnormal behaviour detection, that improve the detection accuracy.

Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection [6] in the context of social multimedia. A large-scale analysis over a Carnegie Mellon University (CMU) based dataset has been conducted to identify its performance in terms of detecting malicious events such as Identity theft, profile cloning and confidential data collection.

Inception-V3 neural network [27] the whole process consists of two parts 1) Inception-V3 neural network is used to extract the features of image samples. 2) The abnormal behaviour recognition model is established to classify the obtained features. Three-layer feedforward neural network algorithm is used to compare with other common machine learning classification models K-nearest neighbour algorithm, random forest, SVM and two-layer feed forward neural network.

A method based on maximization of the area under the ROC [4]. This method does not need manual labelling for estimation of the model. It carries out the estimation process in a semi-supervised fashion. Hierarchical detection of abnormal behaviours based on the priority of the behavioural features is also developed in the testing phase. The process stops if abnormal behaviour is detected.

The KRMEFE-RABEC technique includes three key processes [10] specifically pre-processing, feature extraction, and classification. Kringing Regressive Mapreduce pre-processing is accomplished for cleaning and transforming the raw data into a valuable and understandable format to minimize the complexity of the disease diagnosis. Renyi entropy-based feature extraction process is carried out to the pre-processed features for finding the robust features significant to seizures for accurate disease diagnosis. Rocchio Adaptive Base class boost ensemble technique is applied to early epileptic seizure recognition with the robust extracted features by constructing the weak learners.

PROPOSED METHODOLOGY

Hybrid Deep Learning Technique with one class SVM for Anomaly Detection

A video frame is subjected to a series of operations to extract most salient information from it. Initially, a 2D variance plane is constructed to encode local spatio-temporal variations around each pixel in a video frame. The Improved Particle Swarm Optimization (IPSO) algorithm is then applied to isolate the most salient regions based on motion information in the 2D variance plane. The segmentation process is done with the help of Improved Fuzzy C-Means Clustering (IFCM) of successive frames are exploited for pattern matching in a simple feature space. A GLCM is applied to the extract salient pixels in video. Finally developed a Hybrid Deep Learning (HDL) based on CNN and One-class SVM is trained with spatial features for robust classification of abnormal shapes. A 2D variance plane is computed from the optical flow field magnitude to extract more discriminative information from the proposed Histogram Of Swarms (HOS) descriptor. A 2D Variance plan extracts the features required for tracking.

2D variance plane

A 2D variance plane from the optical flow field magnitude is computed to extract more discriminative information in the frame. Figure 1 shows the optical flow field magnitude of a input frame.

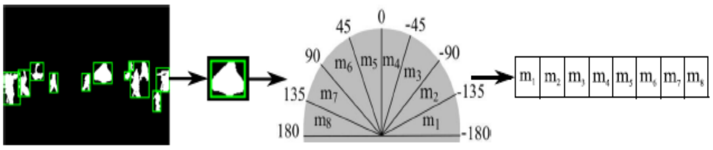

Figure 1 Optical Flow field Magnitude of Input Frame

Each location in the plane pertaining to a pixel in the original frame contains a value calculated by finding variance in the 5×5×5 spatio-temporal neighbourhood of the pixel. The numerical value assigned to each pixel shows deviation of optical flow from its surrounding in both space and time. A 2D variance plane is contracted for each frame using optical flow field magnitude. Each pixel in the variance plane represents the variance of optical flow field magnitude in local spatio temporal neighbourhood of that pixel [17]

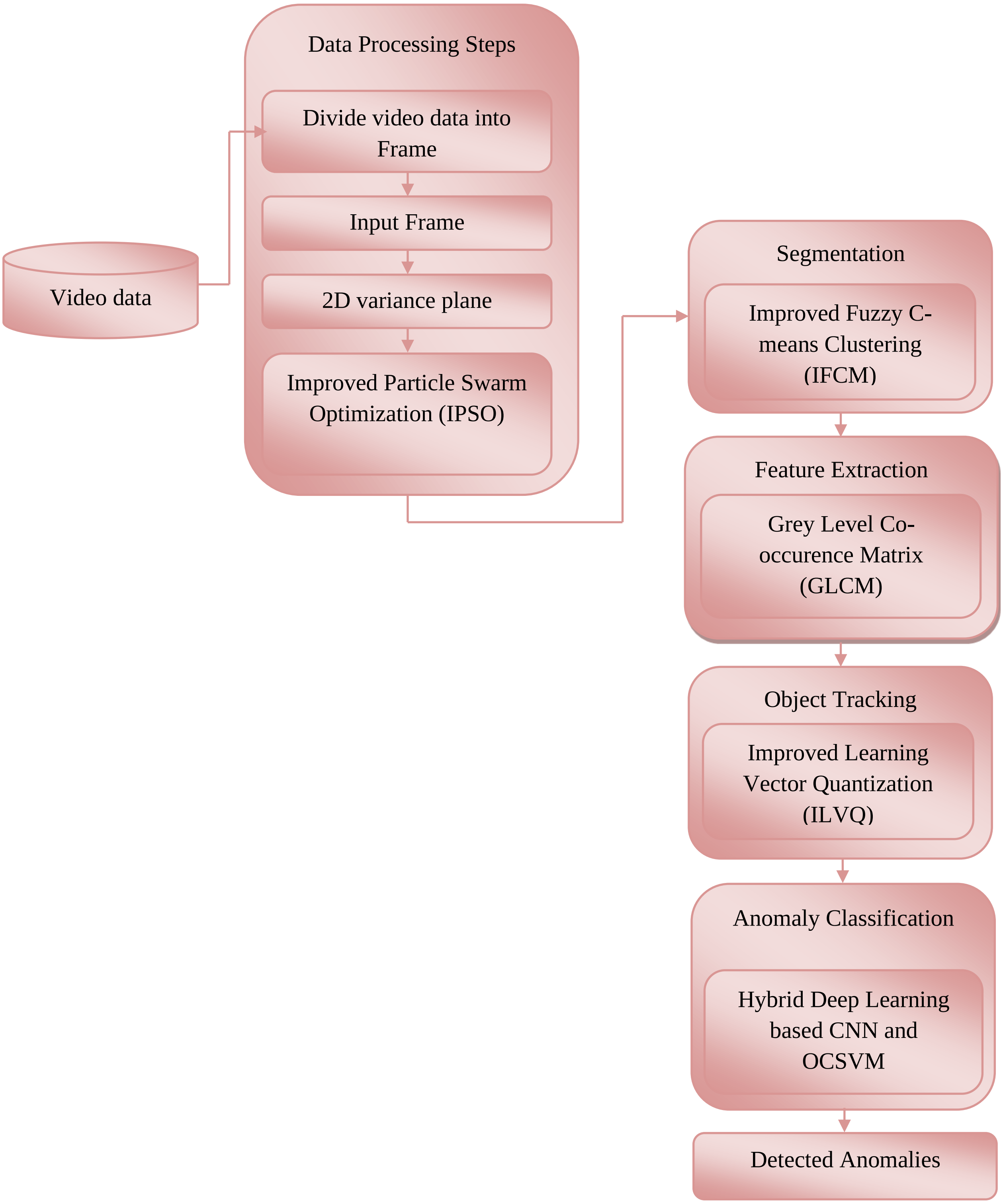
Data Processing Steps
Divide video data into Frame
Input Frame
2D variance plane
Improved Particle Swarm Optimization (IPSO)
Video data
Segmentation
Improved Fuzzy C-means Clustering (IFCM)
Feature Extraction
Grey Level Co-occurence Matrix (GLCM)
Object Tracking
Improved Learning Vector Quantization (ILVQ)
Anomaly Classification
Hybrid Deep Learning based CNN and OCSVM
Detected Anomalies

Figure 2 Overall process of HDL CNN Method

Improved Particle Swarm Optimization

Particle Swarm Optimization (PSO) uses a number of particles, it constitute a swarm moving around in the search space, to look for the best solution. Each particle is treated as a point in a D-dimensional space, that adjusts its flying according to its own flying experience as well as the flying experience of other particles [19]. The particles flight with a certain velocity in the D-dimensional space to find the optimal solution.

The velocity of particle i expresses as $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, the location of particle i expresses as $(x, x_{i2}, \dots, x_{iD})$,, the optimal location of particle i expresses as $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$, it is also called p_{best}.

The global optimum position of all particles expresses as $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$, it is also called g_{best}. Each particle in group has a fitness function to calculate the fitness value. In standard PSO, the velocity update formula of the dimension d shows in formulae (1) and (2):

$$v_{id} = w \times v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times (p_{gd} - x_{id}) \quad (1)$$

$$(X_{id} = x_{id} + v_{id}) \quad (2)$$

PSO parameters include: *Q* - Population Quantity, w - inertia weight, *C1* and *C2* are acceleration constants, v_{max} is the maximum velocity, G_{max} is the maximum number of iterations, rand ( ) and Rand ( ) are random functions with values in [0,1]. The value of *C1* and *C2* takes constant 2.

During optimization, the IPSO enhances information communication among populations and maintains population diversity to overcome the limitations of classical optimization algorithms in solving multiparameter, strong coupling, and nonlinear engineering optimization problems. These limitations include advanced convergence and the tendency to easily fall into local optimization. The parameters involved in the imported local-global information sharing term are analyzed and the principle of parameter selection for performance is determined. The performances of the IPSO and classical optimization algorithms are tested by using multiple sets of classical functions to verify the global search performance of the IPSO [13].

A IPSO variant that tries to improve the performance of PSO algorithm in finding better solutions while preserving both its simplicity and its fast convergence is developed. This divergence restriction factor is based on the introduction of a simple yet effective new operation in the iterative search process in order to enhance the algorithms ability in both exploring new areas of the search space that may contain better solutions and exploiting intermediate solutions. The starting point for the proposed variant is a modified PSO version based on parameter settings.

Divergence restriction factor K is introduced into PSO to ensure the best convergence. equation (3) presents the velocity formula:

$$v_{id} = K[v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times (p_{gd} - x_{id})] \quad (3)$$

Value *c1* and value *c2* used 2.05 that are same with Clerc's experiment. Here reserve four decimal places of K for experiment. Equation (4) is the specific velocity formula:

$$v_{id} = 0.7298 \times [v_{id} + 2.05 \times rand() \times (p_{id} - x_{id}) + 2.05 \times Rand() \times \left(p_{gd} - x_{id}\right) \quad (4)$$

In the early iterations, a particle in PSO needs to detect in a wide range to determine the likely location of the optimal solution. In later iterations, it needs to develop locally within a small range to determine the optimal point. Thus, *K* should take a larger value in the early and take a smaller value in the later. Simultaneously, *K* should become smaller slowly to the minimum in a longer period of late stage. This pattern of change is consistent with the con-cave function.

To avoid premature convergence, the divergence restriction factor should choose a convex function in the early iterations so that the particles can find optimal solution in a wide range. In the late period, it should choose a concave function so that the divergence restriction factor can change slowly to the minimum in order to develop locally [13]. It ensures convergence of the algorithm. According to this principle, the functional divergence restriction factor structuring on the basis of the cosine function is showed in equation (5)

$$K = \frac{\cos\left((\pi/G_{max}) \times T\right) + 2.5}{4} \quad (5)$$

where *T* is the number of iterations. Set G_{max}= *40*, the changing curve of value *K* appeared. The curve of *K* is convex function at first and transforms into a concave function at last. The value *K* is substituted in equation 5.1 and turns into equation (6).

$$v_{id} = \left(\frac{\cos\left((\pi \times T/G_{max})\right) \times 2.5}{4}\right) \times [v_{id} + 2 \times rand() \times (p_{id} - x_{id}) + 2 \times Rand() \times \left(p_{gd} - x_{id}\right)] \quad (6)$$

Segmentation using Improved Fuzzy C-means clustering

The Fuzzy C-Mean algorithm is one of the common algorithms that used to image segmentation by dividing the space of image into various cluster regions with similar images pixels values [15]. It is a clustering algorithm that enables data item to have a degree of belonging to each cluster by degree of membership. The algorithm is an iterative clustering method that produces an optimal c partition by minimizing the weighted within group sum of squared error objective function.

Let $X = \{x_1, \dots\dots, x_n\}$ be a dataset and let c be a positive integer greater than one. A partition of *X* into *c* clusters is represented by mutually disjoint sets $X_1, \dots., X_c$ such that $X_1 \cup \dots.. \cup X_c = X$ or equivalently by indicator function $\mu_1, \dots\dots \mu_c$ such that $\mu_i(x) = 1$ if *x* is in X_i and $\mu_i(x) = 0$ if *x* is not in X_i for all $i = 1, \dots\dots, c$. This is known as clustering *X* into *c* clusters $X_1 \dots.. X_c$ using $\{\mu_1, \dots\dots \mu_c\}$.

A fuzzy extension allows $\mu_i(x)$ taking values in the interval [0,1] such that $\sum_{i=1}^{c} \mu_i(x) = 1$ for all *x* in *X*. In this case, $\{\mu_1, \dots \dots \mu_c\}$ is called a fuzzy c-partition of *X*. Thus, the FCM objective function $\mathrm{J}_{\mathrm{FCM}}$ is defined in equation (7).

$$J_{FCM}(\mu, v) = \sum_{i=1}^{c} \sum_{j=1}^{n} \mu_{ij}^{m} d^2(x_j, v_i) \tag{7}$$

where $\mu = \{\mu_1, \dots \dots \mu_c\}$ is a fuzzy c-partition with $\mu_{ij} = \mu_i(x_j)$, the weighted exponent m is a fixed number greater than one establishing the degree of fuzziness, $v = \{v_1, \dots \dots v_c\}$ is the c cluster centers, and $d^2(x_j, v_i) = \|x_j - v_i\|^2$represents the Euclidean distance or its generalization. The FCM algorithm is an iteration through the necessary conditions for minimizing JFCM with the following update equation (8) and (9).

$$V_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{m} x_j}{\sum_{j=1}^{n} \mu_{ij}^{m}} \quad (i=1, \dots \dots c) \tag{8}$$

and

$$\mu_{i,j} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d(x_{j,}v_i)}{d(x_{j,}v_i)}\right) 2^{m-1}} \tag{9}$$

At each iteration, μ and v are updated using equation (8) and (9). The FCM algorithm iteratively optimizes $JFCM(\mu, v)$ until $|\mu(l+1) - \mu^l| \leq \varepsilon$ is the number of iterations. From the equation (8), it is clear that the objective function of FCM does not take into account any spatial dependence among *X* and consider each image pixel as an individual point. Also, the membership function in equation (10) is determined by $d^2(x_j, v_i)$, that measures the similarity between the pixel intensity and the cluster center. The closer the intensity values to the cluster center the higher the value of the membership. The membership function is highly sensitive to noise. If an image is affected by noise or other, the intensity of the pixels would change results in an incorrect membership and improper segmentation.

Distance community attraction factor is considered to exist between neighboring pixels to overcome the drawbacks. During clustering, each pixel attempts to attract its neighboring pixels toward its own cluster. This distance community attraction factor depends on two factors; the pixel intensities or feature attraction $\lambda(0 < \lambda < 1)$ and the spatial position of the neighbors or distance attraction $\xi(0 < \xi < 1)$, that also depends on the neighborhood structure. Considering this distance community attraction factor defined as in the equation (10)

$$d^2(x_j, v_i) = \|x_j - v_i\|^2 (1 - \lambda H_{ij} - \xi F_{ij}) \qquad (10)$$

H_{ij} represents the feature attraction and F_{ij} represents the distance attraction. The parameters λ and ξ adjust the degree of the two neighbourhood attractions. H_{ij} and F_{ij} computed in a neighborhood containing S pixels are given in equation (11), (12), and (13)

$$H_{ij} = \frac{\sum_{k=1}^{S} \mu_{ik}\, g_{ik}}{\sum_{k=1}^{S} g_{ik}} \qquad (11)$$

$$F_{ij} = \frac{\sum_{k=1}^{S} \mu_{ik}^2\, q_{jk}^2}{\sum_{k=1}^{S} q_{jk}^2} \qquad (12)$$

with

$$g_{jk} = |x_j - x_k|,\ q_{jk} = (a_j - a_k)^2 + (b_j - b_k)^2 \qquad (13)$$

where (a_j, b_j), and $(a_k - b_k)$ denote the coordinate of pixel j and k, respectively. It should be noted that a higher value of λ results in a stronger feature attraction and a higher value of ξ results in a stronger distance attraction. Optimized values of these parameters lead to the best segmentation results. Figure 3 shows the segmented image of the input frame.

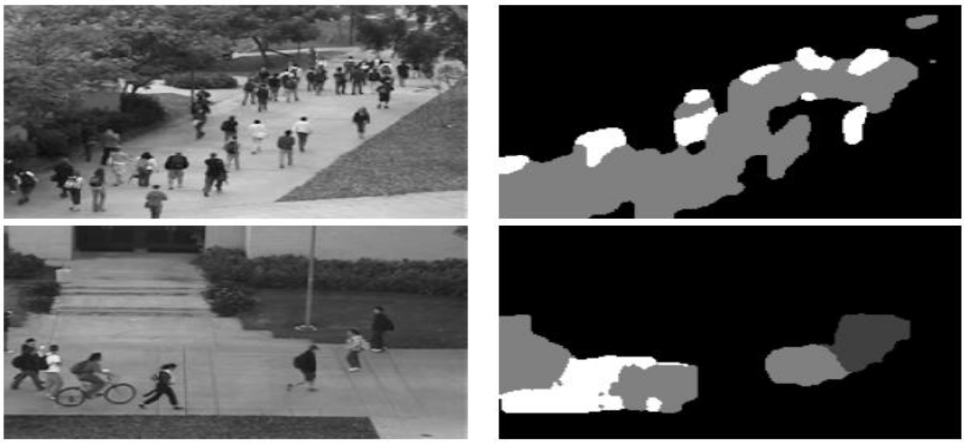

Figure 3 Segmented Image of Input Frame

https://doi.org/10.37896/pd91.4/91442

Grey Level Co-occurence Matrix

GLCM, texture properties can be captured. Using co-occurrence matrix, feature vector can be extracted in several ways. In this, contrast has been selected in extracting feature vector. Contrast is a measure of the local variations present in an image.

Classification using Deep Learning based on a CNN and One-class SVM

The CNN is designed to take advantage of two dimensional structures like 2D Images and capture local spatial patterns. This is achieved with local connections and tied weights. It consists of one or more convolution layers with pooling layers between them, followed by one or more fully connected layers, as in a standard multilayer perceptron [6]. In CNNs, kernels or filters are used to see required features are present in an image by convolution with the image. The size of the filters gives rise to locally connected structure that are each convolved with the image to produce feature maps. The feature maps are usually sub-sampled using mean or max pooling. The reduction in parameters is due to the fact that convolution layers share weights.

The reason behind parameter sharing is that make an assumption, that the statistics of a patch of a natural image are the same as any other patch of the image. This suggests that features learned at one location can also be learned for other locations. Learned feature detector anywhere in the image. This makes CNN ideal feature extractors for images. The CNN with many layers have been used for various applications especially image classification. Deep CNN have achieved the lowest error rates in image classification tasks. The CNNs, perform classification by extracting image features directly from raw images via tuning the parameters of the convolutional and pooling layer. The features extracted by CNN strongly depend on the size of the training dataset. If the training dataset is small, CNN tends to overfit after several epochs. Deep Convolution Neural Network (DCNN) with transfer learning have evolved. A hybrid model consisting of a Deep Convolutional feature extractor followed by a fast and accurate classifier, the One Class Support Vector Machine (OCSVM), for the purpose of avoiding over fitting problem in CNN for anomalies detection in the given images. Figure 4 shows the CNN kernels.

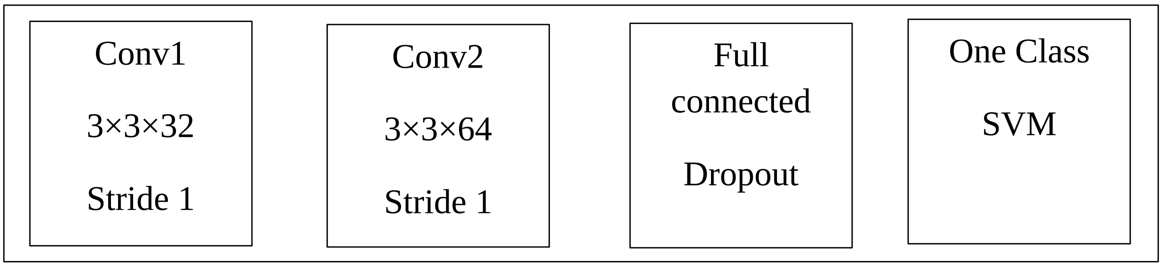

Figure 4 CNN with one class SVM

One Class Support Vector Machine

SVM have always been of interest in anomaly detection because of their ability to provide non-linear classification through a kernel function [3].

Consider the traditional two-class support vector machines in that are given as a set of n training instances $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. $x_i \in R^d$, where y_i is the class label of the x_i instance and $y_i \in [-1, +1]$. The linear SVMs classifier recovers an optimal separating hyperplane maximizing the margin of the classifier with the equation: $w^T x + b = 0$, with $w \in F$ and $b \in R$ two parameters determine the position of the decision hyperplane in feature space F its orientation is tuned by w and its displacement by b. The decision function is shown in equation (14).

$$f(x, w, b) = sign(w^T x + b) \in \{-1, +1\} \tag{14}$$

Already notice some over runs that will make considerable contribution to the classifier profile defined by the decision function. Its shown in equation (15).
where,

$$sign\ (w^T x + b) = \begin{cases} +1, & if\ (w^T x + b) = 0 \\ -1, & otherwise \end{cases} \tag{15}$$

The concept of SVMs is to find (w,b) such that the hyperplane is positioned at maximum distance of the nearest training samples of the two classes in order to reduce the generalization error. This distance defines the margin. SVMs have first been proposed for linearly separable classification tasks. SVM are extended to non-linearly separable classification problems. Some samples are allowed to violate the margin and a non-linear decision boundary can be obtained by projecting the data into a higher dimension space thanks to a non-linear function *Φ(x)*. Data points may not be linearly separable in their original space. If data points are lifted into a feature space *F* a hyperplane can separate them. Hyperplane is projected back into the input space, it has a non-linear shape. Preventing the SVM classifier from over-fitting noisy data, slack variables ξ are introduced to allow some data points to lie within the margin, and the parameter *C > 0* in equation 5.18 tunes the trade-off between the classification error on the training data and margin maximization. The objective function of SVM classifiers has the following minimization formulation shown in equation (16).

$$\underset{w, b, \xi_i}{min} \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} \xi_i \tag{16}$$

subject to $y_i(w^T \phi(x_i) + b \geq 1 - \xi_i$

$$\xi \geq 0, \mathrm{i} = 1, \ldots \ldots \mathrm{n}$$

The minimization problem is solved using Lagrange multipliers α_t, *i=1,……n*. The new decision function rule for a data point x is defined in equation (17).

$$f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b) \tag{17}$$

Every $\alpha_i>0$ is weighted in the decision function and thus supports the machine. Since SVMs are considered to be sparse, there are relatively few Lagrange multipliers with a non-zero value. The function $K(x, xi) = F(x)\ T\ F(xi)$ is known as the kernel function. Since the outcome of the decision function only relies on the dot-product of the vectors in the feature space F. It is not necessary to perform an explicit projection. As long as a function K provides the same results, it can be used instead. This is known as the kernel trick. Popular choices for the kernel function are linear, polynomial, and sigmoidal. Gaussian Radial Base Function is used. Its shown in equation (18)

$$K(x, x_i) = \exp\left(\frac{-\|x-x_i\|^2}{2\sigma^2}\right) \tag{18}$$

where $\sigma \in R$ is a kernel parameter and (x, x_i) is the dissimilarity measure. The power of the method comes from using kernel functions that enable it to operate in a high dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates.

One Class SVM (OCSVM) are used to separate the data of one specific class, the target class, from other data. OCSVM are trained with positive examples only, data points from the target class. In the feature space F, OCSVM method basically separates all the data points from the origin by a hyperplane and it maximizes the distance of this hyperplane to the origin. This results in a binary function that captures the region of the input space where the training data lives. Thus the function returns +1 in a small region capturing the training data points and −1 elsewhere. The quadratic programming minimization function is slightly different from the original stated by equation (19).

$$\min_{w,\xi_i,\rho} \frac{\|w\|^2}{2} + \frac{1}{\eta n}\sum_{i=1}^{n} \xi_i - \rho \tag{19}$$

subject to $(w, \phi(x_i)) \geq \rho - \xi_i$

$$\xi \geq 0, i = 1, \ldots \ldots n$$

The new regularization parameter η instead of C in the original formulation given in equation (19) The range of C is from zero to infinity, but η is always between [0, 1]. η characterizes the solution in a nice interpretable way: (1) it sets an upper bound on the fraction of outliers, e.g. the training examples regarded out-of-class, (2) and it sets a lower bound on the number of training examples used as support vectors. Again by using Lagrange techniques and using a kernel function for the dot-product calculations, the decision function becomes as in equation (20) and (21).

$$f(x) = sign((w\phi(x_i)) - \rho) \tag{20}$$

https://doi.org/10.37896/pd91.4/91442

$$= sign\left(\sum_{i=1}^{n}\alpha_i K\,(x,x_i)-\rho\right) \tag{21}$$

OC-SVMs thus create a hyperplane characterized by w and ρ which has maximal distance from the origin in the feature space *F*, hence separating all the data points from the origin.

The problem of fitting the hyper-parameters of OC-SVM automatically is addressed. In the case of OC-SVM, this amounts to choose the kernel parameter γ and the regularization parameter η. A pair (γ_i,η_j) is defined as a learning configuration. For this purpose, to run OC-SVM for several learning configurations and select the best configuration by evaluating the diversity function. This main improvement of one-class SVMs is with respect to the slack variables. As illustrated in Figure 4, a non-zero slack variable ξi allows a point *xi* to lie on the other side of the decision boundary. In the case of robust one-class SVMs, the slack variables are proportional to the distance to the centroid. This allows points that are distant from the center to have a large slack variable. Since the slack variables are fixed, they are dropped from the minimization objective.

On the other hand, this causes the decision boundary to be shifted towards the normal points. It loses part of the interpretability of the results as there is no restriction on the number of points that can appear on the other side of the decision boundary. Theoretically, all the points can be labelled as outlying using equation (20) and consequentially, the majority could have a score greater than 1.0 using equation (21).

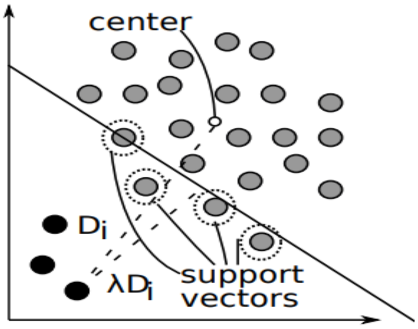

Figure 5 Modifying the slack variables for one-class SVM.

Figure 5 illustrates the slack variables are modified. Points that are further away from the center of the data are allowed to have a larger slack variable. The decision boundary is shifted towards the normal points and the outliers are no longer support vectors. The objective of the one-class SVMs is stated in equation (22). The slack variables are dropped from the minimization objective as shown in equation (23). They only appear in the constraints as $\widehat{D_\iota}$, whereas Q is the diversity function.

$$min_{w,p}\frac{\|w\|^2}{2}-\rho \tag{22}$$

subject to $w^T\phi(x_i)\geq\rho-\lambda*\widehat{D_\iota}$

https://doi.org/10.37896/pd91.4/91442

The slack variable $\widehat{D_\iota}$ is computed using equation (24). It represents the distance to the centroid in the kernel space. Since the diversity function *(Q)* is implicitly defined by the kernel, equation (24) cannot directly be used. This approximation is summarized in equation (26). The expression $\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$ is a constant and hence it can be dropped. The normalized distance $\widehat{D_\iota}$ appears in the optimization Objective using equation (24).

$$D_i = \left\| \phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i) \right\|^2 \tag{23}$$

$$\widehat{D_\iota} = \frac{D_i}{D_{max}} \tag{24}$$

$$\approx Q(x_i, x_i) - \frac{2}{n}\sum_{j=1}^{n} Q(x_i, x_j) \tag{25}$$

`

The dual objective of the one-class SVM can be summarized as follows:

$$min_\alpha \frac{\alpha^T Q^\alpha}{2} + \lambda D^T \alpha \tag{26}$$

$$\text{subject to } 0 \leq \alpha \leq 1, e^T\alpha = 1 \tag{27}$$

It can be seen that this modification to the dual objective of the one-class SVM objective in equation (27) and hence it can be incorporated easily in the original solver. Figure 6 shows the Anomalies detected using HDL Scheme.

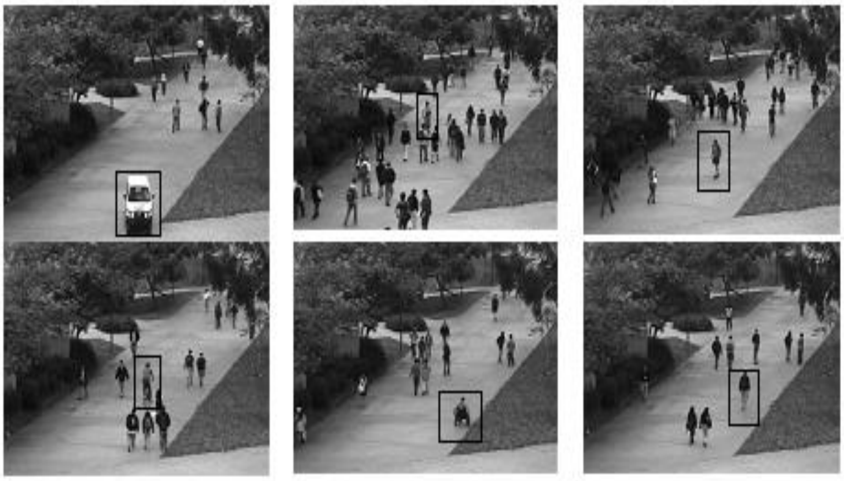

Figure 6 Detected Anomalies – HDL

EXPERIMENTAL RESULTS

The experimental findings were discussed in detail in every section. All tests with 1.8GHz Pentium IV processor were carried out on PC utilizing MATLAB. The HDL algorithm is examined for texture features on video sequence dataset UCSD Ped1. Comparing the efficiency of the proposed scheme with existing schemes such as EANN, KSVM, SVM, SNM, HMM and OF the best combination of features is evaluated based on the parameter value. Parameters are specified based on consistency, accuracy and precision. Several necessary measures have been used in this work to estimate the efficiency of the system of tracking objects based on HDL. Measurements such as Accuracy, Precision and Recall can be used. Let True Positive (*TP*) imply that actual event is anomalous and predicted as anomaly, False Positive (*FP*) actual event is normal and predicted as anomaly, True Negative (*TN*) shows actual event is normal and predicted as normal and False Negative (*FN*) cites actual event is anomalous and predicted as normal.

Table 4.1 shows the Confusion matrix of predicted and actual class using HDL method.

Table 4.1 Confusion Matrix for HDL Scheme

Actual VS Predicted	Predicted Class	
Actual Class	Abnormal	Normal
Abnormal	TP	FN
Normal	FP	TN

Table 4.2 demonstrates the overall numerical assessment of all Anomaly Detection schemes. It illustrates the high performance achieved by the proposed HDL compared to other schemes such as EANN, KSVM, SVM, SNM, HMM and OF. Owing to the efficacious the proposed HDL has high detection rate compare to the existing methods.

Table 4.2 Comparative analysis of HDL scheme using UCSD Dataset

Performance matrices	**Accuracy (%)**	**Precision (%)**	**Recall (%)**
HDL(Proposed III)	97	95.87	96.24
EANN(Proposed II)	95	93	92
KSVM(Proposed I)	94	91	90
SVM	90	85.324	85.21
SNM	86	82	81
HMM	80	81	82.33
OF	77	74	72

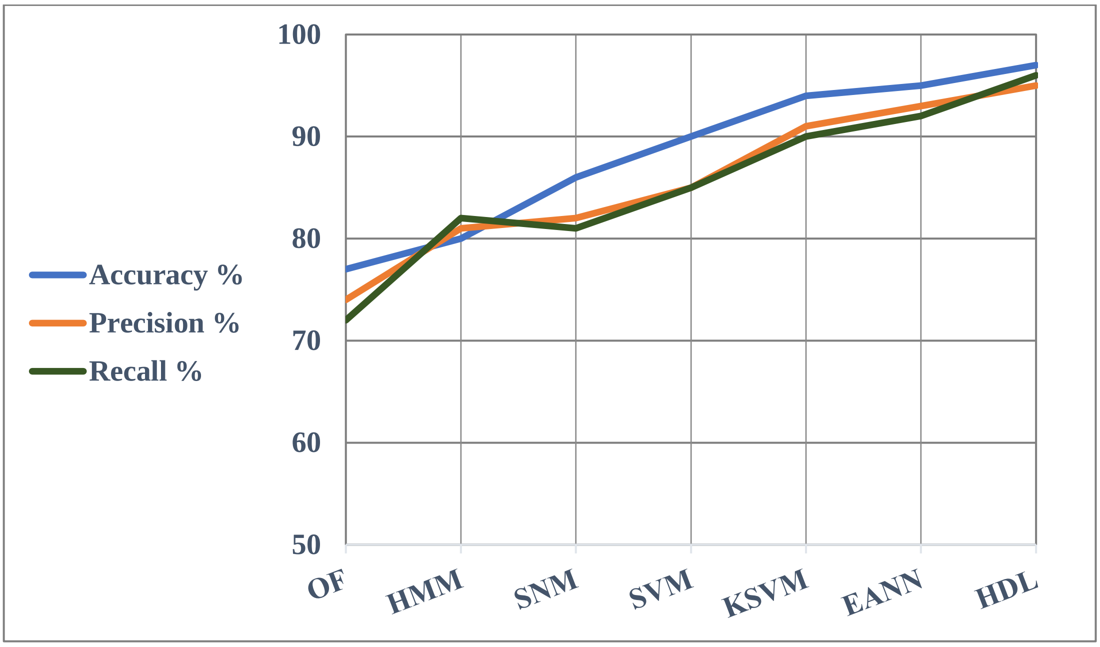

Figure 7 Performance comparison of HDL scheme for UCSD dataset

The Figure 7 illustrate the Accuracy, Precision and Recall comparison between the proposed and existing methods.

The experimental findings discussed in detail are tested with 1.8GHz Pentium IV processor on PC utilizing MATLAB. The processing time for HDL Scheme which includes object detection, Tracking and anomaly detection is shown in Table 4.3.

Table 4.3. Processing Time of HDL Scheme

HDL (Proposed method III)	UCSD
Processing Time(s)	28

SUMMARY

In a deep learning approach to detect anomalies in surveillance videos the major contribution is introducing informative features based on segmentation and using an automatically updated threshold to detect abnormal events. Initially the Improved Particle Swarm Optimization algorithm is applied to isolate the most salient regions based on motion information in the 2D variance plane. A Grey level Co-occurrence Matrix is applied to the extracted salient pixels in the video. The segmentation process is done with the help of Improved Fuzzy C-Means of successive frames are exploited for pattern matching in a simple feature space. A hybrid deep learning based on Convolution Neural Network and One-class Support Vector Machine is trained with spatial features for robust classification of anomalies. Experiments show that the proposed Hybrid Deep Learning method outperforms available methods in terms of prediction Accuracy, Precision and Recall.

REFERENCE

1. Anton, SD, Kanoor, S, Fraunholz, D & Schotten, HD 2018, 'Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set', in Proceedings of the 13th international conference on availability, reliability and security, pp. 1-9.

2. Cho, S-B & Park, H-J 2003, 'Efficient anomaly detection by modeling privilege flows using hidden Markov model', Computers & Security, vol. 22, no. 1, pp. 45-55.

3. Choi, Y-S 2009, 'Least squares one-class support vector machine', Pattern Recognition Letters, vol. 30, no. 13, pp. 1236-40.

4. Feizi, A 2020, 'Hierarchical detection of abnormal behaviors in video surveillance through modeling normal behaviors based on AUC maximization', Soft Computing, vol. 24, no. 14, pp. 10401-13.

5. Gao, B, Ma, H-Y & Yang, Y-H 2002, 'Hmms (hidden markov models) based on anomaly intrusion detection method', in Proceedings. International Conference on Machine Learning and Cybernetics, vol. 1, pp. 381-385.

6. Garg, S, Kaur, K, Kumar, N & Rodrigues, JJ 2019, 'Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective', IEEE Transactions on multimedia, vol. 21, no. 3, pp. 566-78.

7. Hung, T-Y, Lu, J & Tan, Y-P 2013, 'Cross-scene abnormal event detection', in 2013 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2844-7.

8. Javanbakhti, S & Zinger, S 2012, 'Fast abnormal event detection from video surveillance', in Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), pp. 144-150.

9. Joshi, SS & Phoha, VV 2005, 'Investigating hidden Markov models capabilities in anomaly detection', in Proceedings of the 43rd annual Southeast regional conference-Volume 1, pp. 98-103.

10. Kaliappan. A & Chitra. D (2021), “Kringing Regressive Map reduce Entropy Feature Extraction based Rocchio Adaptive Boost Ensemble Classifier for Early Disease Diagnosis with Big Data”, Journal of Dynamic Systems and Applications, Dynamic Publishers Inc., US, vol.30, no. 6, pp. 964-980.

11. Kaliappan. A & Chitra. D (2020), “Analysis of Big Data Analytics in Healthcare sector: Applications and Tools”, Journal of Computational and Theoretical Nanoscience, American Scientific Publishers, US, vol.17, no.12, pp. 5605-561.

12. Kavitha R, Chitra. D and Priyadharsini N. K., “Video Event Recognition Using Conditional Random Fields” in the International Journal Annals of the Romanian Society for Cell Biology, ISSN:1583-6258, Vol. 25, Issue 4, 2021, Pages. 6565 – 6573. Scopus Indexed.

13. Kavitha R and Chitra. D, “An Extreme Learning Machine and Action Recognition Algorithm for Generalized Maximum Clique Problem in Video Event Recognition” in the Journal of Dynamic Systems and Applications, Volume no 30, Issue no 8, 2021, ISSN 2693-5295, page no 1228 – 1249

14. Liu, H, Zhou, C, Shen, J, Li, P & Zhang, S 2010, 'Video caption detection algorithm based on multiple instance learning', in 2010 Fifth International Conference on Internet Computing for Science and Engineering, pp. 20-4.

15. Omar, S, Ngadi, A & Jebur, HH 2013, 'Machine learning techniques for anomaly detection: an overview', International Journal of Computer Applications, vol. 79, no. 2, pp. 33-41.

16. Pal, NR, Pal, K, Keller, JM & Bezdek, JC 2005, 'A possibilistic fuzzy c-means clustering algorithm', IEEE transactions on fuzzy systems, vol. 13, no. 4, pp. 517-30.

17. Priyadharsini, N, K, & Chitra, D (2021), ‘A machine learning and swarm intelligence based approaches for anomaly detection in extremely crowded scenes’, Journal of Dynamic Systems and Applications, vol. 30, no. 8, pp. 1250 – 1272, doi.org/10.46719/ dsa20213083.

18. Plotnikov, D, Sopov, E & Panfilov, I 2018, 'An approach for automating the design of convolutional neural networks', in IOP Conference Series: Materials Science and Engineering, vol. 450, pp. 1-9.

19. Poli, R, Kennedy, J & Blackwell, T 2007, 'Particle swarm optimization', Swarm intelligence, vol. 1, no. 1, pp. 33-57.

20. Sabokrou, M, Fayyaz, M, Fathy, M, Moayed, Z & Klette, R 2018, 'Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes', Computer Vision and Image Understanding, vol. 172, pp. 88-97.

21. Schuster, R, Schulter, S, Poier, G, Hirzer, M, Birchbauer, J, Roth, PM, Bischof, H, Winter, M & Schallauer, P 2011, 'Multi-cue learning and visualization of unusual events', in 2011 IEEE International Conference on Computer Vision Workshops, November 6 – 13, Spain, pp. 1933-40.

22. Singh, S, Tu, H, Donat, W, Pattipati, K & Willett, P 2008, 'Anomaly detection via feature-aided tracking and hidden Markov models', IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 39, no. 1, pp. 144-59.

23. Tabia, H, Gouiffes, M & Lacassagne, L 2012, 'Motion modeling for abnormal event detection in crowd scenes', in International symposium on signal, Images, Video and communications, University of Valenciennes, France, pp. 1-4.

24. Wang, J & Xia, L-m 2017, 'Abnormal behavior detection by causality analysis and sparse reconstruction', Journal of Central South University, vol. 24, no. 12, pp. 2842-52.

25. Wang, J & Xia, L 2019, 'Abnormal behavior detection in videos using deep learning', Cluster Computing, vol. 22, no. 4, pp. 9229-39.

26. Xu, L, Ren, JS, Liu, C & Jia, J 2014, 'Deep convolutional neural network for image deconvolution', Advances in neural information processing systems, vol. 27, pp. 1790-8.

27. ZhanLi, L & JiaWei, Y 2019, 'Abnormal Behavior Recognition Based on Transfer Learning', in Journal of Physics: Conference Series, vol. 1213, pp. 1-7.