

# Crowd Behavior Recognition Using Hybrid Tracking Model and Genetic algorithm Enabled Neural Network

Manoj Kumar and Charul Bhatnagar

*Department of Computer Engineering and Applications  
GLA University, Mathura, India  
manojkumar15es@gmail.com*

Received 12 March 2016

Accepted 28 September 2016

## Abstract

In the current era, crowd behavior analysis is important topic due to the significance of video surveillance in the public area. Literature presents a handful of works for crowd behavior detection and analysis. Even though, the complicated challenges such as, low quality video, wide variation in the density of crowds and difficult motion patterns pose a complicated challenges for the researchers in crowd behavior detection. In order to alleviate these issues, we develop a crowd behavior detection system using hybrid tracking model and integrated features enabled neural network. The proposed crowd behavior detection system estimate the direction of movement of objects as well their activity using proposed GLM-based neural network. The proposed GLM-based neural network integrates the LM algorithm with genetic algorithm to improve the learning process of neural network. The performance of the proposed crowd behavior detection algorithm is validated with five different video and the performance is extensively analyzed using accuracy. From research outcome, we proved that the proposed system obtained the maximum accuracy of 95% which is higher than the existing methods taken for comparison.

**Keywords:** Crowd video, crowd behavior, tracking, recognition, neural network

## 1. Introduction

Intelligent video surveillance has been became one of thekey research area in computer vision due to heightened security concerns. Even though many research topics are available in video surveillance, tracking and behavior analysis from the crowded video are challenging problem because it has a number of applications including eventmonitoring, behavior modeling, traffic control and security applications [1-4]. In dense crowds, tracking and behavior detection is a major problem [5, 6] because the large number of object is in close which makes it difficult to establish correspondences across frames. In recent years, a number of security agencies specialized in dense crowd management have emerged to respond to the need. Especially, this problem has started to draw attention of

the research community for automatic detection of abnormal crowd behaviors during public events [14]. Technically speaking, crowd behavior analysis can be divided into two tasks: (1) motion information extraction and (2) abnormal behavior modeling. The former usually amounts to crowd tracking. It is a process by which we estimate the speed, direction and location of crowd in a video sequence. Higher level models of crowd behavior can be used to detect anomalous events [15].

However, this effort is hampered by general difficulties of the anomaly detection problem [16]. One fundamental limitation is the lack of a universal definition of anomaly. For crowds, it is also infeasible to enumerate the set of anomalies that are possible in a given surveillance scenario [17]. This is compounded by the sparseness, rarity, and discontinuity of anomalous events, which limit the number of examples available to

train an anomaly detection system. In result, anomaly detection can be extremely challenging. While this has motivated a great diversity of solutions, it is usually quite difficult to objectively compare different methods. Typically, these combine different representations of motion and appearance with different graphical models of normalcy, which are usually tailored to specific scene domains. Abnormalities are themselves defined in a somewhat subjective form, sometimes according to what the algorithms can detect. In some cases, different authors even define different anomalies on common data sets [17].

In this paper, we considered five different videos which are marathon sequence as well as pedestrian video. These videos defined vehicle as the abnormality. By considering these video as input, this work presents crowd behaviour recognition system using integrated features and GLM-based neural network. At first, input video are given to hybrid tracking model which considers the EWMA model and motion model to track the objects of movements. Once the tracking path is identified, twelve different features are extracted to find the characteristics of the objects. These twelve features are then used to train the GLM-based neural network to classify the direction of movement and activity. The proposed GLM-based neural network integrates the LM algorithm and genetic algorithm for finding the optimal weights. Based on the trained weights, GLM neural network find the direction of movement and activity of objects. The paper is organized as follows: Section 2 reviews the literature. Section 3 presents the motivation and section 4 presents the proposed method for crowd behaviour detection. Section 5 presents the experimental results and conclusion is given in section 6.

## 2. Literature Review

Data-driven method for crowd tracking utilizes the learned pattern to handle fast and dense crowd flow. The problem here is more computational task is required in searching of motion pattern [7]. HajerFradiet *al.*[8] proposed crowd density map-based tracking method for monitoring persons in crowd in video surveillance data. This method has the advantage of using local features as an observation of a probabilistic density function. Inflexibility in person detections for high level environment change is the drawback here. Irshad Ali, Matthew N. Dailey [9] proposed the approach of

confirmation-by-classification method which detects and tracks multiple humans in high-density crowds in the presence of extreme occlusion. But detector's output is unreliable in this approach. Convolutional neural network-based method is proposed by Lijun Cao *et al.* in [10] which is more robust than the direct way of tracking but it requires large enough data and enough diversity. Discriminative structure prediction model which captures the interdependence of multiple influence factors is proposed in [11]. The problem here is the risk of labeling error.

Xing Hu *et al.* [20] proposed a LNND descriptor to represent the video event for anomaly detection in crowded scenes. Si Wu *et al.* [19] proposed a Bayesian framework for escape detection by directly modeling crowd motion in both the presence and absence of escape events. Specifically, they introduced the concepts of potential destinations and divergent centers to characterize crowd motion in the above two cases respectively, and construct the corresponding class-conditional probability density functions of optical flow. Chunyu Chen *et al.* [18] proposed an algorithm based on the acceleration feature to detect anomalous crowd behaviors in video surveillance systems. Different from the previous work that uses independent local feature, the algorithm explores the global moving relation between the current behavior state and the previous behavior state. Weixin Li *et al.* [17] have proposed an anomaly detector that spans time, space, and spatial scale, using a joint representation of video appearance and dynamics and globally consistent inference. They modeled crowded scenes with a hierarchy of MDT models, equated temporal anomalies to background subtraction, spatial anomalies to discriminant saliency, and integrated anomaly scores across time, space, and scale with a CRF. Shobhit Saxena *et al.* [15] have discussed the crowd feature selection and extraction and proposed a multiple-frame feature point detection and tracking based on the KLT tracker.

## 3. Motivation behind the Approach

**Problem definition:** Let assume that the input crowded video  $V$  which is represented as a sequence of frames,  $V \in (v_i; 1 \leq i \leq n)$ . The input video contains multiple objects and the intention is to track the moving path of every  $j^{th}$  object using sequence of frames. The

importance challenge considered here is to classify the every  $j^{th}$  object based on movement and behavior. The object's movement is to be classified as, left, right, front and back and the behavior of the object is to be classified normal or abnormal.

**Challenges:** The classification of object's movement and behavior become a challenging issue due to the following reasons:

The influence on accurate tracking of objects due to the overlapping with other objects in crowd video is directly affects the estimation of direction as well as behavior.

The defining of characteristics to classify the abnormal event is very challenging because when an abnormal event happens, everyone will try to escape from the location.

As the video utilized for surveillance have lower quality, developing a automatic robust system which can should be noise adaptable is much required.

The important challenge on algorithmic part is the right selection and utilization of intelligence algorithms and image processing techniques to effectively classify the direction and behavior.

**Contributions of the paper:** The main contribution of the paper is discussed as follows:

The first contribution is to effectively identify the right number of feature set for preserving the characteristics about direction and behavior. Here, we define 12 features to preserve the direction and behavior dependent characteristics

The second contribution is to develop a hybrid learning algorithm for the classification of direction and behavior. Here, the existing Levenberg–Marquardt algorithm [22] is modified with the genetic algorithm and a new learning algorithm, called GLM is proposed.

#### 4. Proposed Method: Crowd Behavior Recognition Using Hybrid Tracking Model and Integrated Features Enabled Neural Network

This section presents the proposed crowd behavior recognition using hybrid tracking model and integrated features enabled neural network. The overall process of behavior detection can be performed using three important steps. In the first step, persons presented in the crowd video are tracked using the Zero-stopping constraint-based hybrid tracking model [25]. In the second step, the important physical parameters such as,

speed, directivity, variance and entropy are found out as quantitative values from the tracked objects. Then, these two features are utilized to detect the crowd behavior using neural network which is trained with the proposed learning algorithm. The proposed learning algorithm combines the Levenberg–Marquardt algorithm and genetic algorithm for better training process. The trained neural network with two output layers is utilized to detect the behaviors such as, direction of movement (four directions) and activity of person (normal or abnormal). The block diagram of the proposed crowd behavior recognition is given in figure 1.

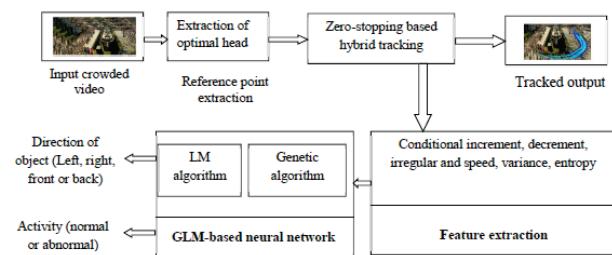


Figure 1. Block diagram

##### 4.1 Tracking of person from crowd video

The first step in the proposed method is to track the path of person from the input video. The tracking path is import to identify their characteristics to detect the direction of movement as well as behavior of person. Here, the tracking of person is performed using the Zero-stopping constraint-based hybrid tracking model given in [25]. In this model, the head object is estimated using the neighborhood search algorithm and the visual-based tracking is performed using HSIM measure. Similarly, the spatial tracking model is done using the EWMA model and both the tracking results are combined with zero stopping constraint.

**a) Reading of video and reference point:** At first, the input crowded video  $V$  is read out and the frames are extracted. Then, every frame of input video,  $V \in (v_i; 1 \leq i \leq n)$  is given as input to the hybrid tracking model to track the human object presented in the frames. Let us assume that the input video contains  $n$  number of frames. At first, reference points are randomly selected from the first frame  $v_1$  to find out the head part of human object presented in the first frame.

$$R_j = (x_j, y_j) ; 1 \leq x_j \leq M ; 1 \leq y_j \leq N$$

Where,  $M \times N$  is size of the image.

**b) Extraction of optimal head object:** Once the reference point  $R_j$  is extracted from the first frame, the head part is found out using the neighborhood-based estimation procedure. To estimate the head part of the reference point  $R_j$ , minimum bound rectangle is formed by setting the reference point as centre pixel and with height  $h$  and width of  $w$ . Then, reference point is moved along left and right direction in the same key frame to find the optimal reference point.

**c) Hybrid tracking model:** Once we detect the head part initially, the movement of the head part is found out using the motion-based estimation model and EWMA model. In the motion-based estimation, the tracking procedure is usually done by matching the objects in the current frame to next frame. Let assume that  $T_j^M$  is the tracked path of the  $j^{th}$  object by motion-based estimation procedure. The initial head location identified from the previous step is added to the tracked path.

$$T_j^M \ll (x_1^i, y_1^i) \in R_j^i$$

In the modeling equation defined in EWMA, the tracking path is identified is as follows:

$$T_j^E(t+1) = D_F * T_j^E(t) + (1 - D_F) * R_j^2(t)$$

$$T_j^E \in (x_j, y_j)$$

Where,  $T_j^E(t+1)$  is the tracking location of  $j^{th}$  object of  $t+1^{th}$  frame by EWMA model.  $T_j^E(t)$  is the tracking location of  $j^{th}$  object of  $t^{th}$  frame by EWMA model,  $D_F$  is factor and  $R_j(t)$  is reference location of  $j^{th}$  object of  $t^{th}$  frame.

Finally, the tracked output from motion-estimated model  $T_j^M$  and EWMA model  $T_j^E$  is integrated with the condition of zero-stopping constraint to obtain the tracking path of  $j^{th}$  object in input crowded video  $V$ .

$$T_j = \left\lceil \frac{1}{2} (T_j^E + T_j^M) \right\rceil ; \text{ if } |T_j^E| \neq \phi$$

$$T_j = |T_j^E| ; \text{ if } |T_j^E| = \phi$$

#### 4.2 Extraction of feature points from the tracking path

The tracked path of  $j^{th}$  object of the input video  $T_j$  is then utilized to find the feature points to find the direction of movement and behavior of the  $j^{th}$  object. The feature points should reflect the characteristics of the direction and behavior of the object. Here, we have

taken six different features such, conditional increment, conditional decrement, conditional irregular, speed, variance and entropy. Here, the first three features such as, conditional increment, conditional decrement, conditional irregular are taken to preserve the characteristics of direction of every objects. The later features such as, speed, variance and entropy are used to preserve the behavior of the crowd objects. Let us consider the tracking path of  $j^{th}$  object of the input video  $T_j$  can be represented as follows,

$$T_j = \left\{ \{l_1^x, l_1^y\}, \{l_2^x, l_2^y\}, \dots, \{l_{nx}^x, l_{ny}^y\} \right\}$$

Where,  $\{l_1^x, l_1^y\}$  is  $x$  and  $y$  coordinate of the first

location of the  $j^{th}$  object. From the tracking path of the  $x$  coordinate, the differential path is computed by taking the successive difference of location. The formulae to compute the differential path is given as follows:

$$CI_k = l_{k+1}^x - l_k^x$$

The differential path is then utilized to compute the conditional increment, conditional decrement and conditional irregular. The conditional increment is the parameter to check whether the direction of movement is always in the positive direction and conditional decrement is about to check whether the direction of movement is always in the negative direction. The conditional irregular is about to obtain the values if the direction of movement is in irregular direction at every time. Accordingly, the following formula is used to compute the features.

$$f_1^x = \begin{cases} 1 & ; \text{ if } \forall CI_k > 0 \\ \text{else} & ; 0 \end{cases}$$

$$f_2^x = \begin{cases} 1 & ; \text{ if } \forall CI_k < 0 \\ \text{else} & ; 0 \end{cases}$$

$$f_3^x = \begin{cases} 1 & ; \text{ if } f_1 = 0 \ \& \ f_2 = 0 \\ \text{else} & ; 0 \end{cases}$$

The fourth feature we considered here is the speed of the object. As we know that the speed of the object is the ratio of distance travelled and the time duration. The speed of the crowd object from the tracking path is computed as follows

$$f_4^x = \frac{N_l * (l_1^x - l_{nx}^x)}{(nx / nT)}$$

Where  $N_l$  is mapping distance which means that the distance in meter corresponds to one pixel,  $l_1^x - l_{nx}^x$  is the total distance travelled in terms of pixel,  $nx$  is the number of frames travelled and  $nT$  is number of frames per second.

The fourth and fifth parameters are related to statistics which are also important to find the behavior

of the crowd object. The variance is computed as follows,

$$f_5^x = \frac{1}{nx} \sum_{k=1}^{nx} (l_k^x - \mu)$$

The entropy is computed as follows,

$$f_6^x = - \sum_{i=1}^{u(l^x)} (P_i \log P_i)$$

Where,  $u(l^x)$  is the unique data values in the location path  $l^x$ . The same six features are extracted from the y location of the tracking path. So, totally, 12 feature points are extracted from the tracking path to detect the direction and behavior of crowd object. The final feature points of  $j^{th}$  object are represented in feature as like follows:

$$F_j = [f_1^x, f_2^x, \dots, f_6^x, f_1^y, f_2^y, \dots, f_6^y]$$

#### 4.3 Direction and activity classification using GLM-based neural network

Once the features are extracted from the tracking path of every object from the input crowd video, the proposed GLM-based neural network is utilized to find the direction and activity of objects. Feed forward neural network [21] is one of the widely used artificial intelligence model to perform the classification using two important processes like, training and testing. In training process, the features are utilized to learn the neurons by finding the optimal weights and then, learnt weights are utilized to find the class label of the tracked objects about the direction of movement and activity of the object.

**a) Architecture of the adapted neural network:** The first step in neural network is an initialization of the neural network. The input and output layer should be initialized with appropriate requirement. Here, the input features are 12 and bit elements required to get these two outputs are six so the neural network is initialized with 12 inputs and six outputs. The first output is to find the direction of the object from the first four outputs and the second output is to find the activity of the object last two outputs. Then, the number of hidden layers is assigned to one and the number of hidden neurons is found out from the experimentation to fix up the exact number. Figure 2 shows the architecture of the adapted neural network.

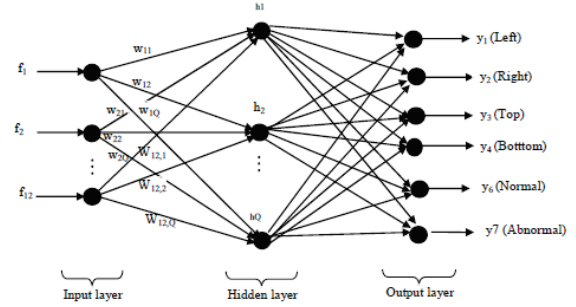


Figure 2. Architecture of the adapted neural network

#### b) Training by the proposed GLM algorithm:

After initializing the neural network architecture, the weight values should be initialized with random value. For the adapted neural network,  $12 \times Q$  weighted elements to connect the input layer and hidden layer and  $Q \times 7$  weighted elements to connect hidden layer with output layer. Also,  $Q$  bias weights in the hidden layer and six bias elements to connect the output layer. So, weight vector can be represented as,

$$W = \{w_i ; 1 \leq i \leq 42 + 7 \times Q\}$$

Where,  $w$  are node weights and bias weight. Initially, the weights are randomly initialized and then, weights are dynamically updated in every iteration  $t$ . The formulae used for updating the weights of every iterations using LM algorithm [22] is given as follows:

$$W_{t+1}^{LM} = W_t - [H + \mu * I]^{-1} * g$$

$$H = J^T * J$$

Where,  $\mu$  is the Levenberg's damping factor which ranges from 0 to 1.  $I$  is the identity matrix.  $J$  is the Jacobian matrix for the system which is obtained by taking the first-order partial derivatives of a vector-valued function. This can be computed by finding the partial derivatives of each output in respect to each weight, and has the form  $F(r, W)$  which,  $r$  is a nonlinear functions viewed for neural network.  $r$  is the feature vector given to the neural network,  $W$  are the weights of the network,  $X$  is total number of weights defined in the neural network, including node and bias weight. The gradient matrix of  $g$  is computed using the following equation.

$$g = J^T * E$$

Once weights are computed, the output of neural network is computed after applying weights to the neural network function.



$$Y^{LM} = F(r, W^{LM})$$

$F(r, W^{LM})$  is the network function computed for every feature vector of the training signal using the weight vector  $W$ .  $Y$  is the associate output vector approximated or predicted by the network. Using the neural network output and original ground truth,  $E^{LM}$  is computed containing the output errors for each input vector used on training the network.

$$E^{LM} = \frac{1}{N} \sum_{i=1}^N (Y^{LM}_i - Y_{G_i})$$

In the current iteration, the two weights such as,  $W_t$  and  $W_{t+1}^{LM}$  is used to find the new weights based on genetic algorithm [23]. From the genetic algorithm, we apply cross over and mutation operator to these two vectors to generate a weight of  $W_{t+1}^{GA}$ . Once the crossover point is defined, weight vector from beginning of chromosome to the crossover point is copied from  $W_t$ , the rest is copied from the  $W_{t+1}^{LM}$  to find a new weight vector. This new weight vector is utilized to perform mutation operation where, an element corresponding to the mutation point is updated with random value. The final weight vector obtained after mutation operator is known as,  $W_{t+1}^{GA}$ .

The weights  $W_{t+1}^{GA}$  are again applied to neural network function to obtain the neural network output which is then used to compute error.

$$Y^{GA} = F(r, W^{GA})$$

$$E^{GA} = \frac{1}{N} \sum_{i=1}^N (Y^{GA}_i - Y_{G_i})$$

Two errors computed  $E^{LM}$  and  $E^{GA}$  are computed and the one which is having the less value is assigned as the error value ( $E_{t+1}$ ) for the current iteration and its corresponding weight is assigned as the final weights for the current iteration ( $W_{t+1}$ ).

The error of the current iteration  $E_{t+1}$  and previous iteration  $E_t$  is compared and if the value is decreased, then  $\mu$  is decreased with a factor  $\nu$ . If the error value is increased, then  $\mu$  is increased with a factor  $\nu$ . This process is repeated for  $T$  number of iteration and the final weights are taken as the trained weights which are then used for the direction and activity estimation of the crowd object. Figure 3 shows the pseudo code of the proposed GLM algorithm.

	GLM Algorithm
1	<b>Input:</b> Feature vector $F$ and Ground truth $Y_G$
2	<b>Output:</b> Final weight vector $W$
3	<b>Procedure</b>
4	<b>Start</b>
5	Initialize weight vector $W_0$
6	Find error $E$ after applying $W_0$ to $F(x, W)$
7	<b>While</b> ( $t < T$ )
8	Compute the Jacobian $J$
9	Compute the $H$ and $g$
10	Find new weight vector $W_{t+1}^{LM} = W_t - [H + \mu * I]^{-1} * g$
11	Find error $E^{LM}$ after applying $W_{t+1}^{LM}$ to $F(x, W)$
12	Find new weight vector $W_{t+1}^{GA}$ by applying cross over and mutation
13	Find error $E^{GA}$ after applying $W_{t+1}^{GA}$ to $F(x, W)$
14	If ( $E^{LM} < E^{GA}$ )
15	$W_{t+1} = W_{t+1}^{LM}$ and $E_{t+1} = E^{LM}$
16	else
17	$W_{t+1} = W_{t+1}^{GA}$ and $E_{t+1} = E^{GA}$
18	Endif
19	If ( $E_{t+1} < E_t$ )
20	Decrease $\mu$ with a factor $\nu$
21	Else
22	Increase $\lambda$ with a factor $\nu$
23	Endif
24	<b>Endwhile</b>
25	Return weight $W_{t+1}$
26	<b>End</b>

Figure.3. GLM algorithm

## 5. Results and Discussion

This section discusses the experimental results and the detailed evaluation of the proposed crowd behavior recognition system with existing techniques.

### 5.1 Dataset description

For the experimental validation, we utilized five dataset where, three dataset are taken from High Density Crowds Data Sets are taken from [12]. These three

datasets are named as, Marathon-1, Marathon-2 and Marathon-3. The other two videos are taken from UCSD datasets [13]. These two datasets are named as, UCSD-bicycle and UCSD-car.

*Marathon-1:* This sequence captures participants in a marathon from an overhead camera. It is a hard sequence due to the strict occlusion among the participants, and the similar looking outfits worn by most of the athletes. The sequence has 492 frames, but each athlete remains in the field of view, on average, for 120 frames. *Marathon-2:* This sequence also involves a marathon. However, the camera in this sequence is installed on a high-rise building. As a result, the number of pixels on each individual is fewer. In addition, there are severe illumination changes when athletes move into the shadow of the neighbouring buildings. This sequence has 333 frames. *Marathon-3:* The third sequence is exceptionally challenging due to two factors: 1) appearance drastically changes due to the U-shape of the path; 2) the number of pixels on target varies due to the perspective effect. The fewer pixels make it more difficult to resolve even partial occlusions. The sequence is 453 frames long. *UCSD-bicycle:* This video contains movement of multiple persons with bicycle and the sequence has 100 frames. *UCSD-car:* This video contains movement of multiple persons with car and the sequence has 100 frames.

### 5.2 Evaluation metrics

The performance of the classification in estimation of direction as well as activity detection is evaluated using classification accuracy. It is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

Where, True positive (TP) is correctly identified, False positive (FP) is incorrectly identified, True negative (TN) is correctly rejected and False negative (FN) is incorrectly rejected.

### 5.3 Experimental set up

The proposed crowd behavior recognition system is implemented using Matlab 8.3 (R2014a) with a system configuration of 4GB RAM Intel processor and 64 bit OS. Here, the size of hidden layer is fixed to one and the size of hidden neurons is obtained from the experimentation. The comparison is evaluated with the

four techniques. Here, tracking model like, particle filtering [24] and hybrid model [25] are considered and classification model like LM-based neural network and GLM-based neural network are considered. So, by integrating these techniques four methods such as, particle filtering+LM, particle filtering+GLM, hybrid model+LM, hybrid model+GLM are taken for experimental purposes.

### 5.4 Experimental Results

The sample results of the proposed method are explained in this section. Figure 4.a shows the sample frame from marathon 1. The marathon 1 contains 492 frames which have 100 objects. Figure 4.b shows the detected objects of marathon 1 from the input frame and it is marked with rectangular box. Once we detect the objects, the tracking is performed with hybrid model. The output of the tracking path for marathon 1 is given in figure 4.c. Similarly, figure 5, 6, 7 and 8 shows the sample intermediate results of the marathon 2, 3, UCSD-bicycle and UCSD-car.

### 5.5 Performance evaluation on detecting direction of movement

This section shows the performance evaluation of the proposed method in detecting the direction movement of 100 objects presented in marathon 1, 2 and 3. UCSD videos contain 9 objects. Figure 9.a shows the accuracy graph for marathon 1 in detecting the direction of 100 objects for the various numbers of hidden neurons ( $Q$ ). Here, number of hidden neurons is varied from 10 to 50 and accuracy parameter is obtained for four different methods. From the graph, the accuracy of particle filtering+LM, particle filtering+GLM, hybrid model+LM and hybrid model+GLM achieved 24%, 62%, 56% and 62 when the  $Q$  value is fixed to 40. The accuracy of the method is increased when  $Q$  value is increased and the accuracy of all the four methods is maximum when  $Q$  value is fixed to 50. The maximum accuracy of 95% is obtained for the hybrid model+GLM. Figure 9.b shows the accuracy graph of marathon 2 for various number of  $Q$  values. From figure 9.b, we proved that the hybrid model+GLM achieved the maximum accuracy for all the different  $Q$  values. The maximum accuracy reached by the hybrid model+GLM is 95% which is higher than all the existing methods.

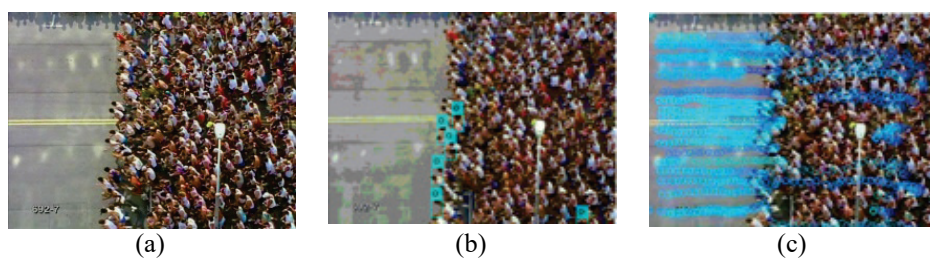


Figure 4: a) sample frame from marathon 1, b) detected object, c) tracking path

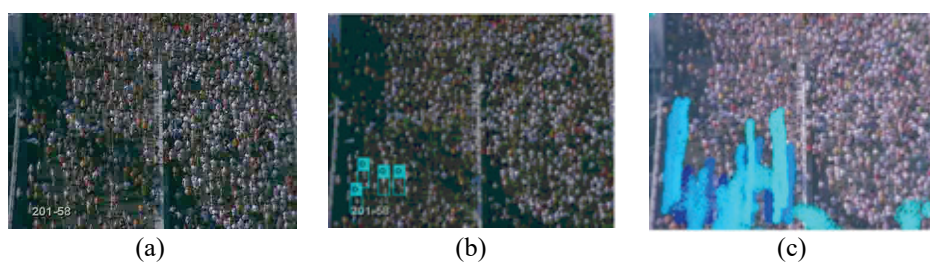


Figure 5: a) sample frame from marathon 2, b) detected object, c) tracking path

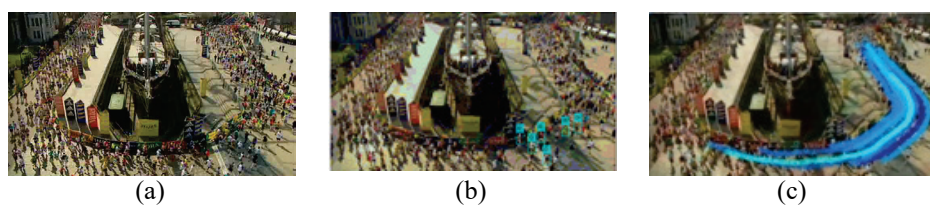


Figure 6: a) sample frame from marathon 3, b) detected object, c) tracking path

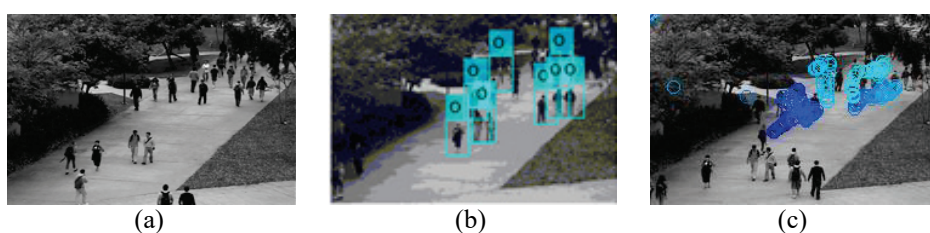


Figure 7: a) sample frame from UCSD-bicycle, b) detected object, c) tracking path

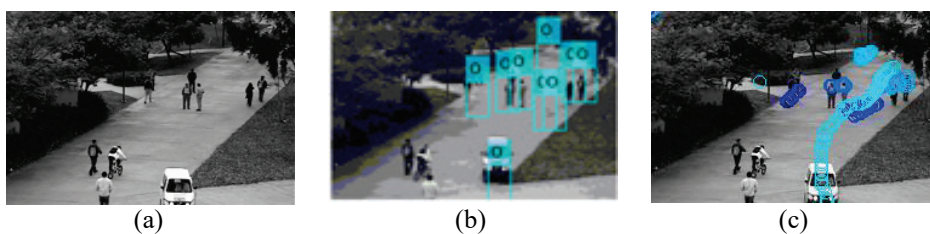


Figure 8: a) sample frame from UCSD-car, b) detected object, c) tracking path



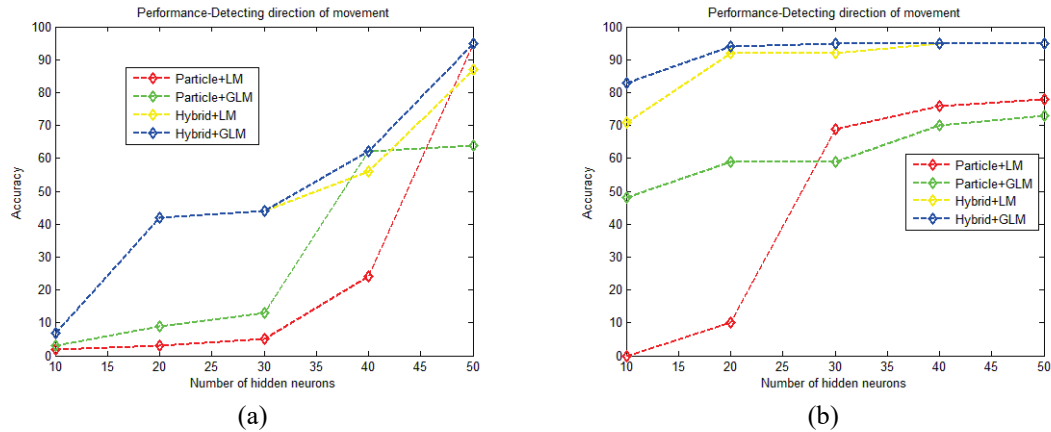


Figure 9. Accuracy graph a) Marathon 1, b) Marathon 2

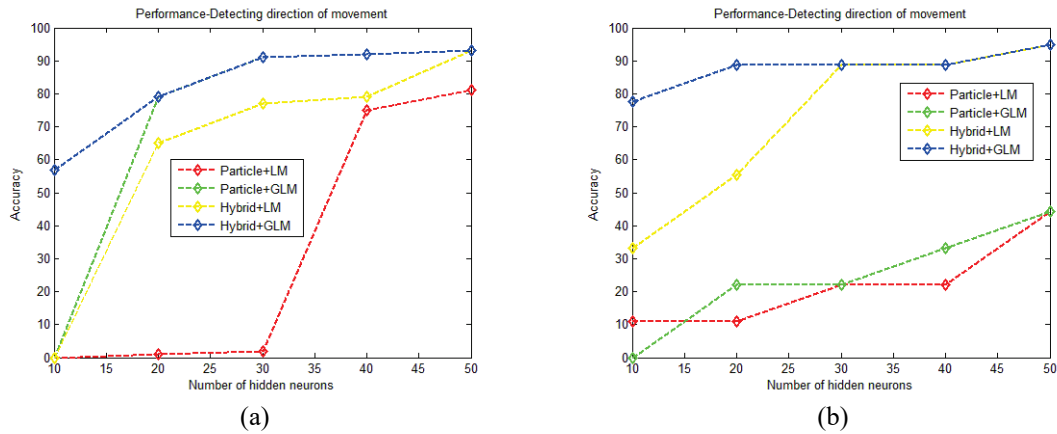


Figure 10. Accuracy graph a) Marathon 3, b) UCSD-bicycle

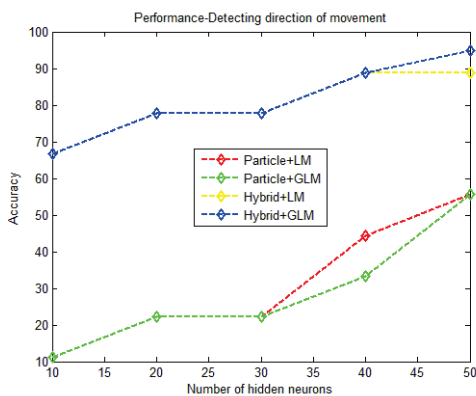


Figure 11. Accuracy graph for UCSD-car

Figure 10 shows the accuracy of the proposed method with existing methods in marathon 3. From the analysis of marathon 3, the maximum accuracy of particle filtering+LM, particle filtering+GLM, hybrid model+LM and hybrid model+GLM achieved 81%, 93%, 93% and 93% when the  $Q$  value is fixed to 50. When comparing the performance of all the four methods, the proposed hybrid model+GLM outperformed all the existing methods for the various  $Q$  values. Similarly, the performance comparison of UCSD-bicycle is given in figure 10.b. Here, the accuracy curve is increase when  $Q$  value is increased from 10 to 50. For the  $Q$  value of 40, the proposed hybrid model+GLM achieved the accuracy of 88% as compared with the particle filtering+LM which obtained the accuracy of 77%. Overall, the maximum accuracy reached by the proposed hybrid model+LM is 95% which is higher than other methods.

Figure 11 shows the accuracy graph for UCSD-car. The accuracy graph is plotted by varying the  $Q$  value from 10 to 50. The figure 11 shows the comparison of four different methods like, particle filtering+LM, particle filtering+GLM, hybrid model+LM and hybrid model+GLM in estimating the direction of movement of objects. When  $Q$  value is fixed to 50, the particle filtering+LM obtained 55% and particle filtering+GLM have obtained the accuracy value of 55%. For the same  $Q$  value, hybrid model+LM and hybrid model+GLM obtained the accuracy of 88% and 95% respectively. From the graph, we ensure that the proposed particle filtering+LM, particle filtering+GLM, hybrid model+LM and hybrid model+GLM outperformed the existing methods in all the  $Q$  values.

### 5.6 Performance evaluation on activity detection

This section shows the performance evaluation of the proposed method in activity detection of 100 objects presented in marathon 1, 2 and 3. UCSD-bicycle and UCSD-car contain 9 objects. The accuracy graph is plotted by varying the  $Q$  value from 10 to 50. The figure 12 shows the comparison of four different methods like, particle filtering+LM, particle filtering+GLM, hybrid model+LM and hybrid model+GLM in detecting the activity of objects whether it is normal or abnormal. From figure 12.a, the accuracy curve is increased when  $Q$  value is increased from 10 to 50. For the  $Q$  value of 10, the proposed hybrid model+GLM achieved the accuracy of 95%. From figure 13.b, the maximum accuracy reached by the proposed hybrid model+GLM is 95% which is higher than other methods in marathon 2.

Figure 13.a shows the accuracy graph for marathon 3. When  $Q$  value is fixed to 10, the particle filtering+LM obtained 37% and particle filtering+GLM have obtained the accuracy value of 95%. For the same  $Q$  value, hybrid model+LM and hybrid model+GLM obtained the accuracy of 84% and 95% respectively. From the graph, we ensure that the proposed particle filtering+LM, particle filtering+GLM, hybrid model+LM and hybrid model+GLM outperformed the existing methods in all the  $Q$  values. Similarly, the performance comparison of UCSD-bicycle is given in figure 13.b. For the  $Q$  value of 10, the proposed hybrid model+GLM achieved the accuracy of 88% as compared with the hybrid model+LM which obtained the accuracy of 77%.

Figure 14 shows the accuracy graph UCSD-car in detecting the activity of 9 objects for the various numbers of hidden neurons ( $Q$ ). Here, number of hidden neurons is varied from 10 to 50 and accuracy parameter is obtained for four different methods. From the graph, the accuracy of particle filtering+LM, particle filtering+GLM, hybrid model+LM and hybrid model+GLM achieved 33%, 33%, 88% and 88% when the  $Q$  value is fixed to 10. The accuracy of the method is increased when  $Q$  value is increased and the accuracy of all the four methods is maximum when  $Q$  value is fixed to 50. The maximum accuracy of 95% is obtained for the hybrid model+GLM.

### 5.7 Statistical validation

This section shows the average performance of the four methods in terms of accuracy for direction of movement and activity detection. Table 1 shows the average performance of the methods without applying any noisy information. From the table, we understand that the proposed Hybrid+GLM obtained better performance in all the five videos taken for the experimentation. The proposed Hybrid+GLM obtained the highest accuracy of 92.4% in the direction of movement for the Marathon 2 where, the existing Hybrid+LM obtained the accuracy of 89.0%. Similarly, when analysing the performance of the activity detection, the proposed Hybrid+GLM outperformed the existing methods in three videos but for the other two videos, the performance is same for the proposed Hybrid+GLM and existing Hybrid+LM. Also, the better performance reached by the proposed Hybrid+GLM is 95%.

Table 2 shows the average performance of four methods by applying noisy information. Here, random noise with density level of 0.2 is applied in the input videos of all the frames and the performance values are obtained. From the table 2, we clearly proved that the proposed Hybrid+GLM obtained the better performance in all the videos. The maximum accuracy for direction of movement is 91.97% which is achieved by the proposed Hybrid+GLM but the existing Hybrid+LM achieved only 88.02%. Similarly, the performance activity detection is far better for the proposed Hybrid+GLM method in three videos by obtaining the maximum accuracy of 94.68%.

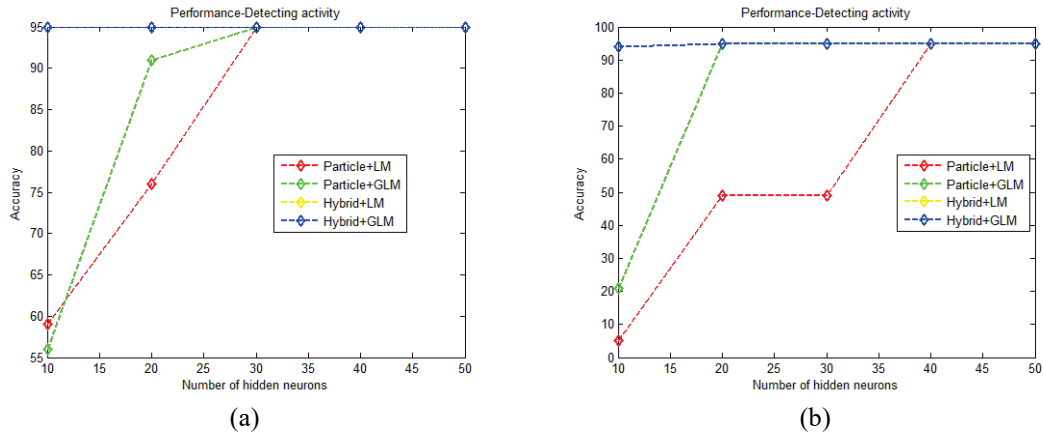


Figure 12. Accuracy graph a) Marathon 1, b) Marathon 2

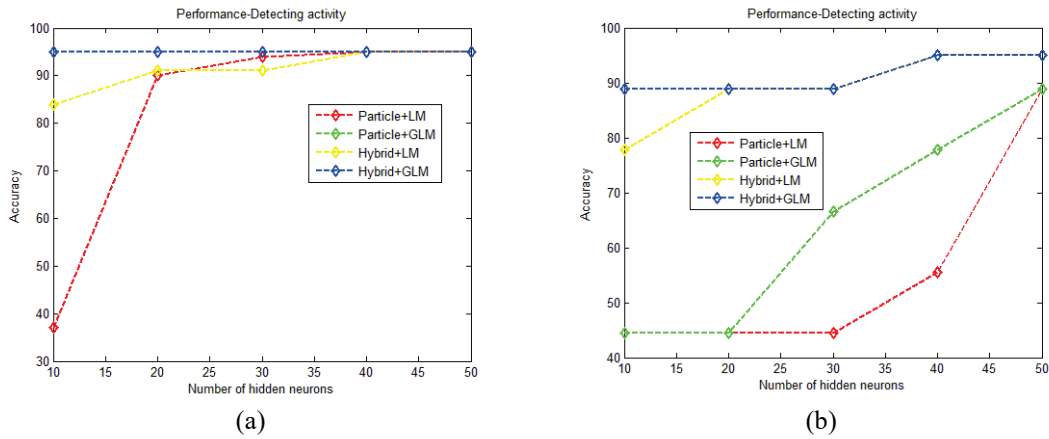


Figure 13. Accuracy graph a) Marathon 3, b) UCSD-bicycle

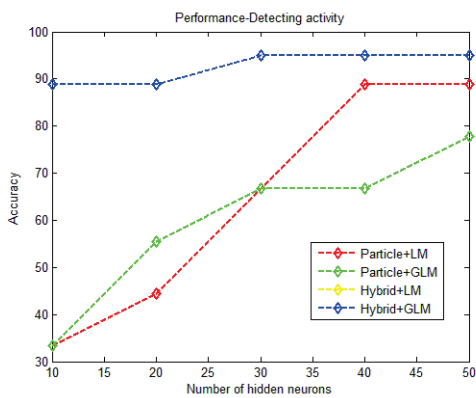


Figure 14. Accuracy graph for UCSD-car

Table 3 shows the computation time of the methods (in sec) for all the five videos. Here, the proposed Hybrid+GLM required only 18.1 sec to find the paths in Marathon 1 but the existing Hybrid+LM requires 18.9 sec to find the paths. This clearly shows the efficiency of the proposed method is better than the existing method. Due to the optimal finding of weights, the training phase is reduced for the proposed method which signifies the better performance in terms of computation time. Similarly, the proposed Hybrid+GLM require 11.5 sec, 17.2 sec, 5.6 sec and 6.8 sec for the Marathon 2, Marathon 3, UCSD-bicycle, UCSD-car videos.

Table 1: Average performance of the methods without noise

		Particle+LM	Particle+GLM	Hybrid+LM	Hybrid+GLM
Accuracy of direction of movement	Marathon 1	25.8000	30.2000	47.2000	<b>50.0000</b>
	Marathon 2	46.6000	61.8000	89.0000	<b>92.4000</b>
	Marathon 3	31.8000	71.0000	62.8000	<b>82.4000</b>
	UCSD-bicycle	22.2222	24.4444	72.3333	<b>87.8889</b>
	UCSD-car	31.1111	28.8889	80.0000	<b>81.2222</b>
Accuracy of activity detection	Marathon 1	84.0000	86.4000	<b>95.0000</b>	<b>95.0000</b>
	Marathon 2	58.6000	80.2000	<b>94.8000</b>	<b>94.8000</b>
	Marathon 3	82.2000	95.0000	91.2000	<b>95.0000</b>
	UCSD-bicycle	55.5555	64.4444	89.1111	<b>91.3333</b>
	UCSD-car	64.4444	60.0000	<b>92.5560</b>	<b>92.5556</b>

Table 2: Average performance of the methods with noise

		Particle+LM	Particle+GLM	Hybrid+LM	Hybrid+GLM
Accuracy of direction of movement	Marathon 1	24.9853	30.1025	47.0424	<b>49.8581</b>
	Marathon 2	45.6942	61.5215	88.0294	<b>91.9782</b>
	Marathon 3	31.6730	70.4531	61.8428	<b>81.4843</b>
	UCSD-bicycle	21.3088	23.4869	71.8480	<b>87.0967</b>
	UCSD-car	30.4787	27.9240	79.1997	<b>80.2627</b>
Accuracy of activity detection	Marathon 1	83.3443	85.6423	<b>94.2940</b>	94.1765
	Marathon 2	58.5643	79.4569	<b>94.1682</b>	94.1052
	Marathon 3	81.3509	94.6078	90.9231	<b>94.6829</b>
	UCSD-bicycle	54.6215	63.7890	89.0649	<b>90.3831</b>
	UCSD-car	63.7657	59.8288	92.4584	<b>92.5211</b>

Table 3: Average computation time of the methods (in sec)

	Particle+LM	Particle+GLM	Hybrid+LM	Hybrid+GLM
Marathon 1	20.1	20.9	18.9	<b>18.1</b>
Marathon 2	23.5	23.8	11.8	<b>11.5</b>
Marathon 3	19.8	20.3	18.1	<b>17.2</b>
UCSD-bicycle	8.9	9.1	<b>5.5</b>	5.6
UCSD-car	9.2	10.7	6.9	<b>6.8</b>

## 6. Conclusion

This paper presented a crowd behavior detection system using hybrid tracking model and GLM-based neural network. Here, two level of estimation was done for identifying the direction of movement for the objects as well as the activity recognition of objects. These two levels of estimation are performed using GLM-based neural network which is the hybridization of LM algorithm with genetic algorithm. Also, twelve different features were taken to preserve characteristics of objects in estimating the direction and activity. For the experimentation, we have considered the five different videos from crowd sequence as well as anomaly events to analyze the proposed method using classification accuracy. From results, we proved that the proposed system obtained the maximum accuracy of 95% which

is higher than the existing methods taken for comparison. In future, the proposed method can be extended to find the optimal architecture for neural network using heuristics methods.

## References

1. M. Hofmann, M. Haag and G. Rigoll, Unified hierarchical multi-object tracking using global data association, *in: PETS*, (2013).
2. M. Pätzold and T. Sikora, Real-time person counting by propagating networks flows, *in: AVSS*, (2011), pp. 66–70.
3. M.D. Breitenstein, F. Reichlin, B. Leibe, E. K. Meier and L.V. Gool, Robust tracking-by-detection using a detector confidence particle filter, *in: ICCV*, (2009).
4. V. Eiselein, D. Arp, M. Pätzold and T. Sikora, Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors, *in: AVSS*, (2012).



5. Z. Wu, N. Hristov, T. Hedrick, T. Kunz and M. Betke, Tracking a large number of objects from multiple views, *IEEE International Conference on Computer Vision*, (2009).
6. X.Song, X.Shao, H.Zhao, J.Cui, R.Shibasaki and H.Zha, An online approach: learning-semantic-scene-by-tracking and tracking-by-learning-semantic-scene, *IEEE Conference on Computer Vision and Pattern Recognition*, (2010).
7. M. Rodriguez, J. Sivic, I. Laptev and J.Y. Audibert, Data-driven Crowd Analysis in Videos, in *proceedings of 2011 IEEE International Conference on Computer Vision (ICCV)*, (2011), pp. 1235 - 1242.
8. H.Frady, V. Eiselein, J.L. Dugelay, I. Keller and T. Sikora, Spatio-temporal crowd density model in a human detection and tracking framework, *Signal Processing: Image Communication*, **31** (2015)100–111.
9. I. Ali and M.N. Dailey, Multiple human tracking in high-density crowds, *Image and Vision Computing*, **30** (2012)966–977.
10. L. Cao, X. Zhang, W.Ren, K. Huang, Large scale crowd analysis based on convolutional neural network, *Pattern Recognition*, (2015).
11. X. Liu, D. Tao, M. Song, L. Zhang, J. Bu, and C. Chen, Learning to Track Multiple Targets, *IEEE transactions on neural networks and learning systems*, **26**(5) (2014)1060 - 1073.
12. Tracking in High Density Crowds Data Set from <http://cvc.ucf.edu/data/tracking.php>.
13. UCSD datasets <http://www.svl.ucsd.edu/projects/anomaly/dataset.htm>.
14. B. Zhan, D.N. Monekosso, P. Remagnino, S.A. Velastin and L.Q. Xu.Crowd analysis: a survey, *Machine Vision and Applications*, **19**(5)( 2008) 345-357.
15. S.Saxena, F.Br'emond, M,Thonnat and R.Ma, Crowd Behavior Recognition for Video Surveillance, in *proceedings of 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, **5259** (2008) 970-981.
16. V. Chandola, A. Banerjee and V. Kumar, Anomaly Detection: A Survey, *ACM Computing Surveys*, **41**(3)(2009).
17. W.Li, V.Mahadevan and N.Vasconcelos, Anomaly Detection and Localization in Crowded Scenes, *IEEE transactions on pattern analysis and machine intelligence*, **36**(1)( 2014) 18-32.
18. C. Chen, Y. Shao and X. Bi, Detection of Anomalous Crowd BehaviorBased on the Acceleration Feature, *IEEE sensors journal*, **15**(12)(2015)7252-7261.
19. S. Wu, H.S. Wong and Z. Yu, A Bayesian Model for Crowd EscapeBehavior Detection, *IEEE transactions on circuits and systems for video technology*, **24**(1)(2014), 85-98.
20. X. Hu, S. Hu, X. Zhang, H. Zhang and L.Luo, Anomaly Detection Based on Local Nearest Neighbor Distance Descriptor in Crowded Scenes, *The Scientific World Journal*, (2014).
21. Tahmasebi and Hezarkhani, A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation, *Computers & Geosciences*, **42**(2012) 18–27.
22. M.T. Hagan and M.B. Menhaj, Training Feedforward Networks with the Marquardt Algorithm, *IEEE transactions on neural networks*, **5**(6) (1994) 989-993.
23. C. Z.Janikow and Z.Michalewicz, An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms, in *Proceedings of the Fourth International Conference on Genetic Algorithms*, (1991),pp. 31–36.
24. A.Bera, D. Wolinski, J.Pettre and D. Manocha, Real-time Crowd Tracking using Parameter Optimized Mixture of Motion Models, *Computer Vision and Pattern Recognition*, Cornell library, (2014).
25. M.Kumar and C.Bhatnagar, Zero-stopping constraint-based hybrid tracking model for dynamic and high dense crowd videos, communicated.