



TS-MDA: two-stream multiscale deep architecture for crowd behavior prediction

Santosh Kumar Tripathy¹ · Harsh Kostha² · Rajeev Srivastava¹

Received: 5 September 2021 / Accepted: 27 June 2022 / Published online: 21 July 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

In recent years, crowd behavior prediction (CBP) has gained much attention from academics and helps to control crowd disasters. The CBP has been solved either as one-class classification (OCC) or multi-class classification (MCC) problems. The OCC-based CBP models learn the normal crowd behavior patterns and treat outliers as anomalies or abnormal crowd behaviors. Nevertheless, these models do not consider the differences in anomaly types and interpret them as one class. On the other hand, the MCC-based CBP models overcome such drawbacks. However, very few datasets and models have been proposed. The current state-of-the-art MCC-based CBP approaches exploit spatial–temporal features but lack in addressing two crucial challenges in the crowd scenes: (a) human-scale variation due to perspective distortion and (b) minimizing effects of cluttered background. To this end, an end-to-end trainable two-stream multiscale deep architecture has been proposed for MCC-based CBP. The first stream uses a deep convolution neural network to extract multiscale spatial features from the frames to handle human-scale variation. The second stream extracts multiscale temporal features from de-background frames using a multi-layer dilated convolution long short-term memory. The effect of the cluttered background has been minimized by extracting de-background frames by adopting a visual background extractor algorithm. The multiscale features from the two streams are concatenated and used to classify different crowd behaviors. The experiments are manifested on two large-scale crowd behavior datasets: MED and GTA. The experimental results show that the proposed model performs better than the state-of-the-art MCC-based CBP approaches.

Keywords Crowd behavior prediction · CNN · Dilated Conv-LSTM · Multiscale spatial–temporal features

1 Introduction

The exponential growth of the worldwide population causes a massive increase in crowd disasters. So, crowd behavior prediction becomes an important task to help crowd

monitoring, thereby minimizing crowd disasters. Thus, crowd behavior prediction (CBP) has recently become a crucial research area and gained much attention from academics and industrial AI. Based on the availability of the ground truth crowd behavior datasets, the current state-of-the-art approaches can be broadly categorized into two types: one-class classification (OCC) [1–21] and multi-class classification (MCC) models [22–24]. The OCC-based CBP approaches usually learn normal crowd behavior patterns, and the outlier is termed a crowd anomaly or abnormal crowd behavior. Such approaches neglect the dissimilarities between different types of abnormal crowd behaviors and treat them as one class.

In contrast, the MCC-based CBP approaches consider the disparities between several crowd behaviors and solve it as a multi-class problem. The MCC-based CBP is beneficial in identifying different types of crowd behaviors and helps control crowd disasters. Recently, research areas like crowd counting [25], crowd congestion-level analysis [26],

Communicated by Ichiro IDE.

✉ Santosh Kumar Tripathy
santoshktripathy.rs.cse18@iitbhu.ac.in

Harsh Kostha
harsh.koshta.che19@iitbhu.ac.in

Rajeev Srivastava
rajeev.cse@iitbhu.ac.in

¹ Computing and Vision Lab, Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, UP 221005, India

² Department of Chemical Engineering, Indian Institute of Technology (BHU), Varanasi, UP 221005, India

and OCC-based crowd abnormality detection [1–21] have been vastly explored. However, limited models have been proposed for MCC-based CBP [22–24]. The conventional machine learning [22] and deep learning [23, 24] models have been proposed for MCC-based CBP. Nevertheless, the deep learning approaches [23, 24] provide a better result than the conventional approach [22] but are limited in addressing the following challenges that exist in the crowd scenes,

- (a) Change in pedestrian’s shape due to perspective distortion. The Fig. 1 shows human shape change in the crowd scene where the person near the camera occupies more pixels than the person far from the camera.
- (b) Minimizing the effect of cluttered background.

Hence, an end-to-end trainable two-stream deep learning architecture (TS-MDA) for the MCC-based CBP has been proposed to fulfill the above research gap. The main contributions are as follows:

- (a) The first stream, also known as multiscale spatial stream (MSS), is designed using deep CNN to extract multiscale spatial features from the RGB frames. The multiscale spatial features are scale-invariant features that can handle human shape variation due to perspective distortion.
- (b) The cluttered background is tackled by adopting a universal background eliminator algorithm to obtain the foregrounds of the frames.
- (c) The second stream, i.e., multiscale temporal stream (MTS), is built using a deep dilated convolutional long–short-term memory (Conv-LSTM) network and obtain scale-invariant de-background temporal features from the foreground frames.

- (d) Finally, the multiscale spatial and de-background temporal features are fused and used to predict different crowd behaviors.
- (e) Extensive experiments are conducted on two publicly available large-scale crowd behavior datasets to show the model’s effectiveness.

The following sections are organized as follows: Sect. 2 discusses the literature review, Sect. 3 discusses the details about the proposed model, Sect. 4 describes the datasets and performance metrics, followed by a discussion of experiment and results analysis in Sect. 5, and Sect. 6 discusses the conclusion and future work.

2 Literature review

The state-of-the-art CBP approaches mainly focus on extracting spatial, frequency, or temporal features from the crowd scenes. The following subsections discuss the recent state-of-the-art OCC and MCC-based CBP models.

2.1 OCC-based crowd behavior prediction

The OCC-based CBP is also known for crowd anomaly or outlier detection. Conventional machine learning [1–4] and deep learning models [5–21] have been explored for OCC-based CBP. The traditional approaches exploit hierarchical features from spatial–temporal interest points [2], spatial–temporal features [3], frequency domain features [27], statistical properties [28], and also trajectories [4] to detect anomalous crowd behavior. On the other hand, deep models using CNNs, sequence to sequence models [11], generative models: encoder-decoders, generative adversarial networks (*GANs*) [18, 19], and hybrid models have also been

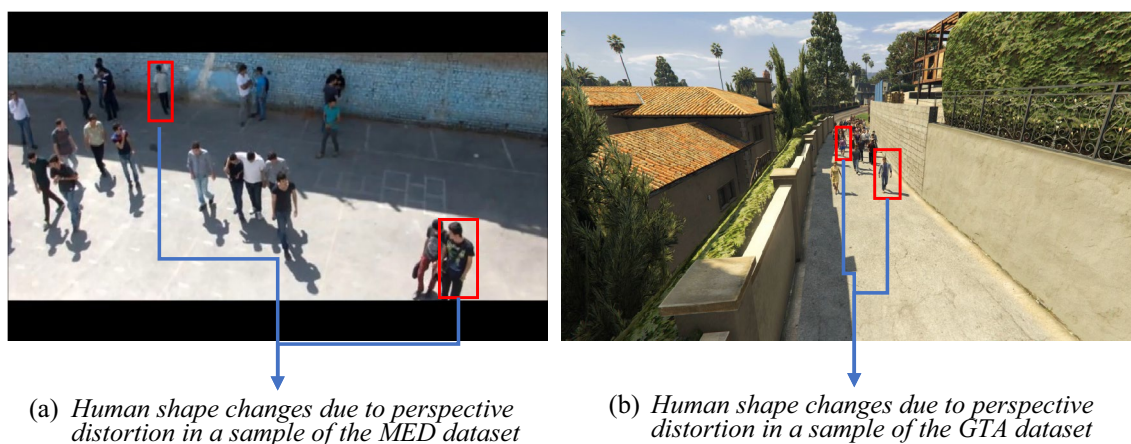


Fig. 1 Examples of human shape change due to perspective distortion in crowd scenes

proposed. These state-of-the-arts adopt One-Class Extreme Learning Machine (OC-ELM) [28], Gaussian classifier [12, 29], One-Class Support Vector Machine (OC-SVM) [6, 13, 15, 30], or reconstruction error [14, 18, 19, 21] for crowd outlier/anomaly detection.

Most of the deep learning techniques exploit crowd appearance and motion properties for anomaly detection. For example, Zhou et al. [5] extracted spatial–temporal features using a spatial–temporal CNN (ST-CNN) for crowd anomaly detection. Bouindour et al. [6] exploited a pre-trained Alex-Net [31] to extract spatial–temporal features from the crowd scenes followed by an OC-SVM for crowd outlier detection. Smeureanu et al. [7] utilized pre-trained CNN model to model crowd behavior patterns and utilized an OC-SVM for abnormal crowd behavior prediction. Similar work can be found by Bouindour et al. [9], where authors proposed a modified pre-trained residual 3D-CNN and extracted spatial–temporal features for crowd abnormality detection. On the other hand, Ravanbakhsh et al. [8] proposed a plug-and-play CNN model which extracts semantic and motion features to detect local crowd anomalous events. Similarly, Song et al. [10] proposed a 3D-CNN for crowd violence detection. Apart from CNN model sequential models like *LSTM* [32], bidirectional-LSTM (BDLSTM) [11] has also been explored for violent crowd detection. Jackson et al. [11] developed a BDLSTM model, which takes histogram of oriented gradient (HOG) features from the frames as input and learns temporal consistencies between them. Recently, Ammar et al. [32] proposed a real-time detection framework for crowd panic-like behavior detection. The authors proposed a hybrid kind of model in which they extracted handcrafted features from the crowd video followed by an LSTM model to capture the temporal dependencies between frames. The model is trained with normal frames. The frames will be treated as panic based on the prediction error. Xu et al. [13] developed a stacked de-noised auto-encoder to exploit spatial and temporal cues from crowd scenes and utilized an OC-SVM for crowd anomaly detection.

Similarly, convolutional auto-encoders [16, 17, 33, 34] are proposed to exploit spatial–temporal features for crowd anomaly detection. Apart from the encoder-decoder models, GANs are also explored. Ravanbakhsh et al. [18, 19] proposed a GAN to exploit motion patterns from the crowd scene, and the reconstruction errors are bounded to detect crowd anomalies. Deep hybrid models have also been explored for crowd anomaly detection. For example, Zhuang et al. [20] proposed a deep hybrid model containing Inception-V3 and stacked differential LSTM to identify violent crowd behavior. Yang et al. [21] proposed a hybrid model containing a CNN-based auto-encoder and LSTM for detecting crowd anomalies. However, the OCC-based CBP models treat different crowd anomalies as the same, i.e., one class, and do not consider their dissimilarities. In such cases, multi-class classification models are helpful.

2.2 MCC-based crowd behavior prediction

A very few models have been explored as far as MCC-based CBP is concerned. The main reason is due to the lack of availability of more multi-class ground truth datasets. Nevertheless, recently, multi-class crowd behavior datasets like MED and GTA have been proposed. Conventional feature learning approaches [22] used HOG, HOF, MBH, dense trajectories, and HOT features to classify crowd behaviors. Dupont et al. [24] analyzed the performance of different deep models for the CBP. Models [24] using 3D-CNN and 3D-CNN + VGG were developed for crowd behavior prediction. Lazaridis et al. [23] proposed a two-stream deep learning architecture for the CBP. The heat map and the optical flow of the crowd scene are inputted into the first and the second stream, respectively. The two streams were developed using convolution layers and Conv-LSTM blocks. Nevertheless, the state-of-the-art MCC-based CBP approaches only focuses on extracting appearance and motion attributes, but the following challenges are yet to be addressed,

- (a) Human shape changes due to perspective distortion and
- (b) Several measures to minimize the effect of cluttered background.

3 Proposed model

A crowd behavior understanding model should handle the two most challenging situations in the crowd scene: human shapes variation due to perspective distortion and the effect of cluttered background. The state-of-the-art CBP models [22–24] deviate in handling such challenges.

Nevertheless, from the literature of related research domains like crowd counting, Sang et al. [35] proposed a scale-adaptive CNN (SA-CNN) for crowd counting in images and handles crowd shape change due to perspective distortion by aggregating features from convolution layers of different scales. On the other hand, the cluttered background can be removed from the scene by utilizing the universal background subtractor [36], i.e., the visual background eliminator (ViBE). Hence, by adopting the idea of SA-CNN [35] and utilizing the ViBE algorithm [36], a two-stream multi-scale deep architecture (TS-MDA) is proposed for the MCC-based CBP. The proposed model can handle human shape variations and minimize the cluttered background effects by extracting multiscale spatial and multiscale de-background temporal features from the scenes. The architecture of the proposed model is illustrated in Fig. 2. The proposed model constitutes of the following sub-modules:

- Preprocessing.
- Candidates for TS-MDA.

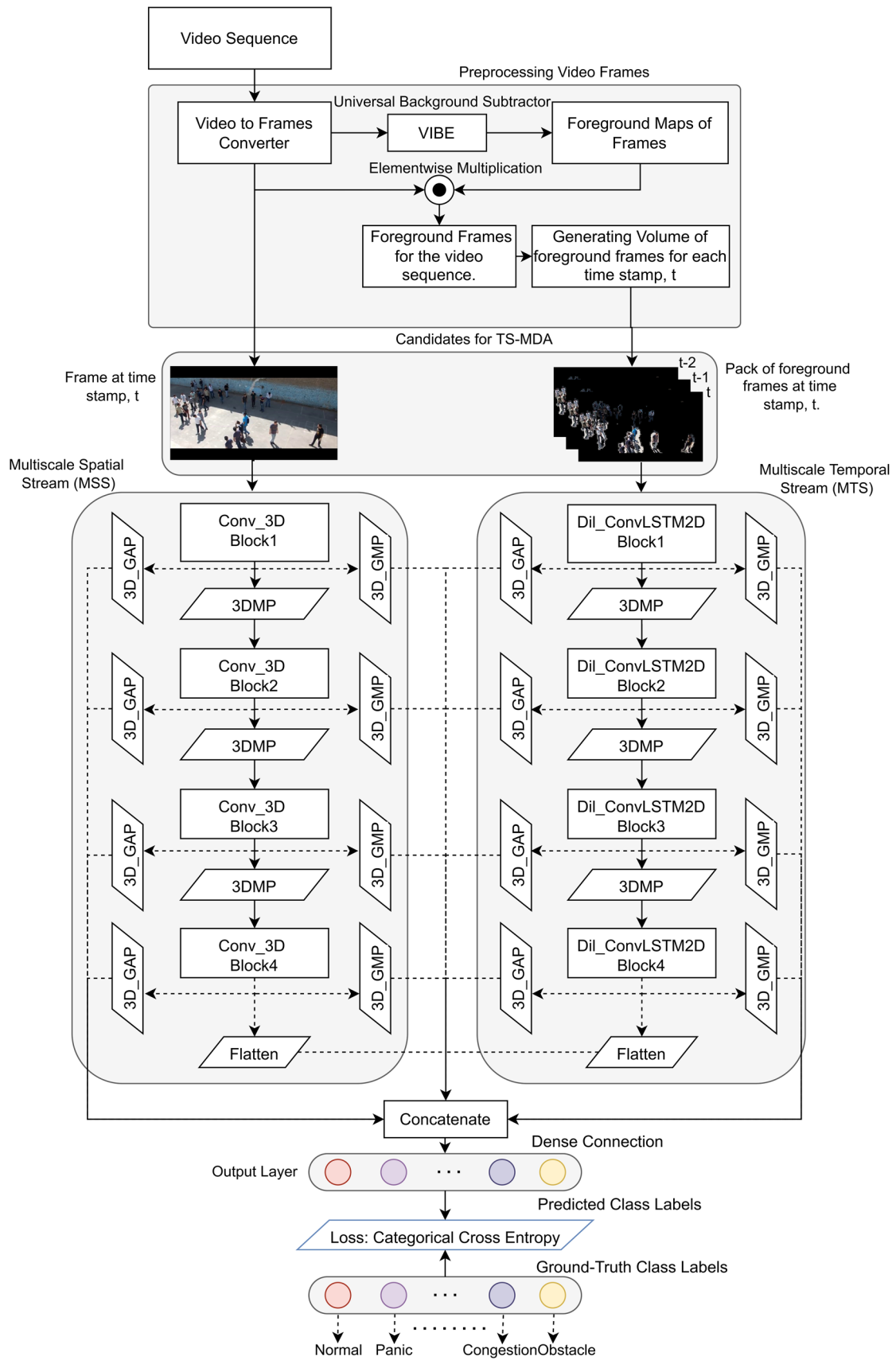


Fig. 2 Architecture of proposed TS-MDA

- Architecture details of the TS-MDA
- Multiscale feature extraction.
- Crowd behavior prediction
- Loss function and optimization.

3.1 Preprocessing

At first, the video sequence is preprocessed before training the TS-MDA. The preprocessing stage is followed by the generation of the candidate for training the TS-MDA. During preprocessing, the RGB frames are extracted from the video sequence. The extracted frames are resized into $[150 \times 150 \times 3]$. Let all the resized N number of frames be represented as $RF = \{rf_1, rf_2, \dots, rf_N\}$. After video to frame conversion, the foreground frames are extracted, explained in the following subsection.

3.1.1 Foreground image extraction

The cluttered background can affect the performance of any model. So, it would be better to eliminate the effects of background from the frames. The universal background eliminator (ViBE) [36] is a popular algorithm to model frame's background; thereby, the foreground maps from the frames can be extracted. Let $v^t(x)$ be the pixel corresponding to location x in the t th resized frame rf_t . According to ViBE [36], the background pixel of the t th frame is modeled by a set of Z background samples/pixels obtained from its previous frame. Let the background pixels from previous frame (rf_{t-1}) be represented as $M^{t-1} = \{v_1^{t-1}, v_2^{t-1}, \dots, v_Z^{t-1}\}$. Here, v_j^{t-1} for $j = 1$ to Z represents to the j th pixel of $(t-1)$ th frame which is classified as a background pixel and Z is the total number of background pixels of $(t-1)$ th frame. The pixel $v^t(x)$ can be a member of M^t by defining a sphere (let say $S_R(v^t(x))$) of radius R centered on $v^t(x)$ and then comparing M^{t-1} to the closest values within the set of samples. The $v^t(x)$ can be classified into background pixel if the cardinality ($\#$) of the intersection of M^{t-1} and $S_R(v^t(x))$ is greater than a threshold ($\#_{\min}$), and formally it can be written as,

$$\#\{S_R(v^t(x)) \cap M^{t-1}\}. \quad (1)$$

For time 0, the background samples are initialized randomly using uniform law and can be represented as:

$$M^0(x) = \{v^0(y) | y \in N_G(x)\}. \quad (2)$$

The background samples are updated for consecutive frames by updating M^t using the Equation-1. After obtaining the background pixels for the resized frame rf_t , the foreground pixels can be easily extracted. Let the foreground pixels represent the foreground maps by a set $FM = \{fm_1, fm_2, \dots, fm_N\}$. The foreground frames/

images are extracted from the scene by performing the elementwise multiplication between RF and FM . Let a set $FF = \{ff_1, ff_2, \dots, ff_N\}$ represent the foreground frames and is obtained by implementing Eq. (3):

$$ff_i = rf_i \odot fm_i, \quad \text{where } i = 1, 2, \dots, N. \quad (3)$$

Here, the symbol, i.e., \odot represents elementwise multiplication.

Now, the volume of foreground images at stamp t is obtained by stacking the foreground images from time stamp $t, t-1, t-2$. Let the set $VF = \{vf_1, vf_2, \dots, vf_N\}$ represent the volume of foreground frames for the dataset. Each of $vf_t = \text{Concatenate}([ff_t, ff_{t-1}, ff_{t-2}])$ for each $t = 1$ to N . Here, Concatenate() is the concatenation operation.

3.2 Candidates for TS-MDA

The main moto of the proposed model is to extract multiscale spatial features and multiscale temporal features from the MSS and the MTS, respectively. So, the MSS and the MTS input should be the frames and volume of frames, respectively. Again, to minimize the background effects, we used the volume of foreground images at timestamp t as input to the MTS. Hence, candidates for the TS-MDA are the resized frames (RF) and volume of foreground frames (VF) for MSS and MTS, respectively.

3.3 Architecture details

The overall architecture of the proposed model is illustrated in the following Fig. 1. The deep architecture contains two streams: a multiscale spatial-stream (MSS) and a multiscale temporal (MTS) stream. The MSS and MTS are inputted with the RF and VF , respectively. The MSS contains four stages of convolution 3D (Conv_3D) blocks. Each block contains a convolution 3D (Conv_3D) layer followed by a ReLU activation layer followed by a batch normalization (BN) layer. The details of the layers are mentioned in Table 1. The features maps are downscaled to its half after Conv_3D Block1, Conv_3D Block2, and Conv_3D Block3 using a 3D max-pooling layer (3DMP). 3D global max-pooling (3D_GMP) and 3D global average pooling (3D_GAP) are used after every activation layer of Conv_3D layers. Similarly, the MTS contains four stages of dilated ConvLSTM2D (Dil_ConvLSTM2D) blocks. The details of these blocks are mentioned in Table 1. Each block contains a dilated ConvLSTM2D layer, a Tanh activation layer, and a BN layer. The features maps are downscaled to its half after Dil_ConvLSTM2D Block1, Dil_ConvLSTM2D Block2, and Dil_ConvLSTM2D Block3 using 3DMP layers. The 3D_GMP and the 3D_GAP are used after every activation layer of dilated ConvLSTM2D layers. All the Conv_3D

Table 1 Details of the layers of the proposed model

| Blocks name | Layers name | No. of kernels | Kernel size | Dilation rate |
|--------------------------|-------------|----------------|-------------|---------------|
| Conv_3D Block1 | Conv_3D | 16 | (5, 5, 5) | |
| | ReLU | – | – | – |
| | BN | – | – | – |
| Conv_3D Block2 | Conv_3D | 64 | (4, 4, 3) | – |
| | ReLU | – | – | – |
| | BN | – | – | – |
| Conv_3D Block3 | Conv_3D | 128 | (3, 3, 3) | – |
| | ReLU | – | – | – |
| | BN | – | – | – |
| Conv_3D Block4 | Conv_3D | 256 | (3, 3, 3) | – |
| | ReLU | – | – | – |
| | BN | – | – | – |
| Dil_ConvLSTM2D Block1 | ConvLSTM2D | 25 | (3, 3) | (2, 2) |
| | Tanh | – | – | – |
| | BN | – | – | – |
| Dil_ConvLSTM2D Block2 | ConvLSTM2D | 40 | (3, 3) | (2, 2) |
| | Tanh | – | – | – |
| | BN | – | – | – |
| Dil_ConvLSTM2D Block3 | ConvLSTM2D | 60 | (2, 2) | (1, 1) |
| | Tanh | – | – | – |
| | BN | – | – | – |
| Dil_ConvLSTM2D Block4 | ConvLSTM2D | 80 | (2, 2) | (1, 1) |
| | Tanh | – | – | – |
| | BN | – | – | – |

layers, dilated ConvLSTM2D layers and max pooling layers are padded with zeros. We have used return sequence as “true” in dilated ConvLSTM2D layers. The feature maps from the activations of the fourth blocks of each layer are flattened. The flattened features maps are concatenated with features from all the 3D_GAP and the 3D_GMP, followed by a batch normalization layer that is fully connected (FC) with the output layer containing different neurons, each representing a particular crowd behavior. The activation of the output layer is SoftMax.

3.4 Multiscale spatial–temporal feature extraction and prediction

3.4.1 Multiscale spatial feature extraction

The proposed model utilizes Conv_3D layers to extract spatial features from the RGB frames. Although Li et al. [37] proposed a Conv_3D network where the convolution operation is performed along the temporal dimension for time-series data analysis, but the Conv_3D layer can also extract fine-grained spatial features by performing convolution across the channel dimension. For this, a slight change in the shape of the input needs to be done. For example, in the proposed model, the shape of the input image for Conv_3D

should be $[\text{batch_size} \times 150 \times 150 \times 3 \times 1]$ here, 3 defines the channel dimension (i.e., RGB), and batch_size defines the size of the batch of samples. We can use convolution 2D (Conv_2D) layers for spatial features extraction. However, the Conv_2D layer uses a 2D filter to perform convolution over each channel separately and then merges them into a single feature map. Thus, the same 2D kernel of shape $[a \times b]$ (here, a, b represent the number of rows and columns of the matrix) will be used for all the channels (R, G, B). So, the 2D kernel is not adaptive as far as learning is concerned for three channels. So, to keep this in mind, we have used Conv_3D layers with 3D kernels to perform convolution across the RGB channel.

The multiscale features can be used to deal with human-scale variation issues. The multiscale features for the spatial stream are obtained from the activations of different convolution layers. The multiscale features include:

- Statistical features like global mean and global max are obtained from each of the low-level activated feature maps of Conv_3D Block1.
- Statistical features like global mean and global max are also extracted from each of the mid-level activated feature maps of Conv_3D Block2 and Conv_3D Block3, respectively.

- High-level features correspond to activated feature maps of Conv3D_Block4 are extracted. The high-level features are in the form of multidimensional tensors and flattened into single-dimensional vectors.

All the extracted features from different scales of the spatial stream are concatenated. Let, $F^{Spatial}$ represents the concatenated multiscale features of the spatial stream.

3.4.2 Multiscale temporal feature extraction

The process of multiscale temporal feature extraction is the same as spatial-stream. The multiscale temporal features include:

- Statistical features like global mean and global max are extracted from the activated feature maps of Dil_ConvLSTM2D Block1, Dil_ConvLSTM2D Block2, Dil_ConvLSTM2D Block3.
- High-level temporal features corresponding to activated feature maps of Dil_ConvLSTM2D Block4 are also obtained. The high-level features are flattened into single-dimensional tensors.

The extracted features maps are then concatenated. Let a set $F^{Temporal}$ represents the multiscale temporal features.

3.4.3 Multiscale spatial–temporal feature fusion

The multiscale spatial features ($F^{Spatial}$) and temporal features ($F^{Temporal}$) are concatenated by simply appending one after another. Let, $F^{Concate} = \text{Concatenate}(F^{Spatial}, F^{Temporal})$ represents the concatenated multiscale features.

3.5 Crowd behavior prediction

The multiscale spatial–temporal features ($F^{Concate}$) are densely connected with the output layer. The output layer is used to predict the crowd behavior labels. The output layer contains different neurons, each representing a particular crowd behavior class like Panic, Fight, Congestion, Obstacle, Neutral, or Normal behaviors. The SoftMax activation is used in the output layer and it can be represented as:

$$Y_{CBP} = \bigcup_{p=1}^K [y_{p_{out}}] = \bigcup_{p=1}^K \text{SoftMax}(y_{p_{in}}) = \bigcup_{p=1}^K \left[\frac{e^{y_{p_{in}}}}{\sum_{p=1}^K e^{y_{p_{in}}}} \right] \tag{4}$$

Here, K resembles the number of available classes, the set $Y_{CBP} = \{y_{1_{out}}, y_{2_{out}}, y_{3_{out}}, \dots, y_{K_{out}}\}$ represents the predicted crowd behavior labels. The $y_{p_{in}} |_{p=1,2,3,\dots,K}$ refers to the weighted information transmitted from the concatenate layer to P th output neuron.

3.6 Loss function and optimization

Let \varnothing_{TS-MDA} , represents all the trainable parameters of the proposed model. Let $T_{i_{CBP}} = \{T_1, T_2, T_3, \dots, T_K\}^i$ be the ground truth labels of the i th crowd scene. The loss on the i th crowd scene is obtained using categorical cross-entropy between $T_{i_{CBP}}$ and $Y_{i_{CBP}}$. Let $L_i(T_{i_{CBP}}, Y_{i_{CBP}})$ be the Cross-Entropy loss on the i th crowd scene and can be represented as follows:

$$L_i(\varnothing_{TS-MDA}) = L_i(T_{i_{CBP}}, Y_{i_{CBP}}) = \left[- \sum_{p=1}^K T_p \log y_{p_{out}} \right]^i \tag{5}$$

Now, the problem becomes an optimization problem such that the loss between true and predicted distribution has to be minimized. The proposed work adopted mini-batch-based gradient decent approach using Adam optimization [38] method to minimize the loss function. The mini-batch-based optimization problem can be represented as:

$$\text{argmin}_{\varnothing_{TS-MDA}} [L(\varnothing_{TS-MDA})]^b \tag{6}$$

Here, b is the batch of samples. To minimize the above optimization problem, first, the mean of cumulative losses for a given batch b of samples of size Batch_Size is obtained:

$$[L(\varnothing_{TS-MDA})]^b = \frac{1}{\text{Batch_Size}} \sum_{i=1}^{\text{Batch_Size}} L_i(T_{i_{CBP}}, Y_{i_{CBP}}) \tag{7}$$

After finding the mean of cumulative of losses for a given batch of samples, b , the gradients of loss for the given batch are obtained as,

$$[\nabla L(\varnothing_{TS-MDA})]^b = \left[\nabla_{\varnothing_{TS-MDA}} L(\varnothing_{TS-MDA}) \right]^b \tag{8}$$

After finding the gradients of loss for the given batch b , the learnable parameters of the proposed TS-MDA are updated using the adaptive moment (Adam) [38] update rule. The Adam [38] optimizer utilizes the cumulative history of gradients to update the \varnothing_{TS-MDA} to solve the decay problem. For a given iteration itr , the cumulative history of gradients for a given batch b can be calculated using the following Eqs. (9–11):

$$m^b = \beta_1 \times m^{b-1} + (1 - \beta_1) \times [\nabla_{\varnothing_{TS-MDA}}]^b \tag{9}$$

$$v^b = \beta_2 \times v^{b-1} + (1 - \beta_2) \times \left([\nabla_{\varnothing_{TS-MDA}}]^b \right)^2 \tag{10}$$

$$\widehat{m}^b = \frac{m^b}{1 - \beta_1^{itr}} \quad \text{and} \quad \widehat{v}^b = \frac{v^b}{1 - \beta_2^{itr}} \quad (11)$$

where $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Now, the parameters are updated using the following Eq. (12) [38]:

$$\Phi_{TS-MDA} = \Phi_{TS-MDA} - \frac{\eta}{\sqrt{\widehat{v}^b} + \epsilon} \times \widehat{m}^b \quad (12)$$

Here, η is the learning rate. According to Adam *optimizer* [38], m^b and v^b are the weighted first- and second-order

moments, whereas \widehat{m}^b and \widehat{v}^b are their corrected moments obtained for a batch b . Furthermore, itr is the iteration number. The Algorithm-1 shows step-by-step processes used to optimize the proposed TS-MDA. The model is trained until the iteration (itr) reaches maximum iteration ($Max_Iteration$), or the early stopping criteria are satisfied. The patience parameter of the early stopping is set to 10.

Algorithm-1 Optimizing the TS-MDA

Input: Resized frameset, $RF = \{rf_1, rf_2, \dots, rf_N\}$, foreground image set, $FF = \{ff_1, ff_2, \dots, ff_N\}$ where N is the total number of frames are inputted into MST and MTS respectively.

Ground-Truth Labels: The set T_{CBP} represents ground-truth crowd behavior labels for N number of frames.

Parameters: η , Φ_{TS-MDA} , *momentum*, β_1 , and β_2

Initialisation: $Max_Iteration = 2000$, $Batch_Size$ (Different for different dataset), $itr = 1$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\eta = 0.01$, regularize parameter of $L_2 = 0.01$ and *patience* = 10.

Output: Optimized TS-MDA

While *early-stopping* or $itr = Max_Iteration$ is satisfied, **do**

For each batch $b = 1$ to $\left\lceil \frac{N}{Batch_Size} \right\rceil$ **do**

For each sample i in batch b **do**

1. Find $F_i^{Spatial}$, $F_i^{Temporal}$, and F_i^{Concat} .
2. Find the predicted crowd behaviour label Y_{iCBP} using Equation-4
3. Find the Loss $L_i(\Phi_{TS-MDA})$ using Equation-5.

end for

4. Find the mean of cumulative of loss for the given batch b using Equations-7.
5. Find the gradients of the loss using Equation-8.
6. Obtain Cumulative History of gradients using Equation-9 to 11.
7. Update network parameter Φ_{TS-MDA} using Equation-12.

End For

6. $itr += 1$

End While

4 Datasets and performance metrics

4.1 Datasets

In our experiment, we have used two publicly available large-scale crowd behavior datasets, such as the MED dataset [22] and the grand theft auto v2 (GTA) [23] dataset. The MED dataset contains 31 crowd behavior sequences. There are five crowd behaviors like Neutral, Panic, Congestion, Fight, and Obstacle or Abnormal behavior. There are nearly 45,000 frames contained in the MED dataset. The

resolution of the original frames is of size $[480 \times 854 \times 3]$. Figure 3 shows few samples of the MED dataset. Authors [22] adopted leave-one-out validation on the MED dataset. The GTA dataset [23] contains 14 Crowd behavior sequences. Each video contains more than 3 min video length. The GTA dataset contains only three crowd behaviors like Normal, Panic, and Fight. The frames are recorded at 60 frames per second. The resolution of the frames is of size $[1080 \times 1920 \times 3]$. Authors [23] randomly selected ten video sequences for training and four for testing on the GTA dataset. Figure 4 shows few samples of the GTA dataset.

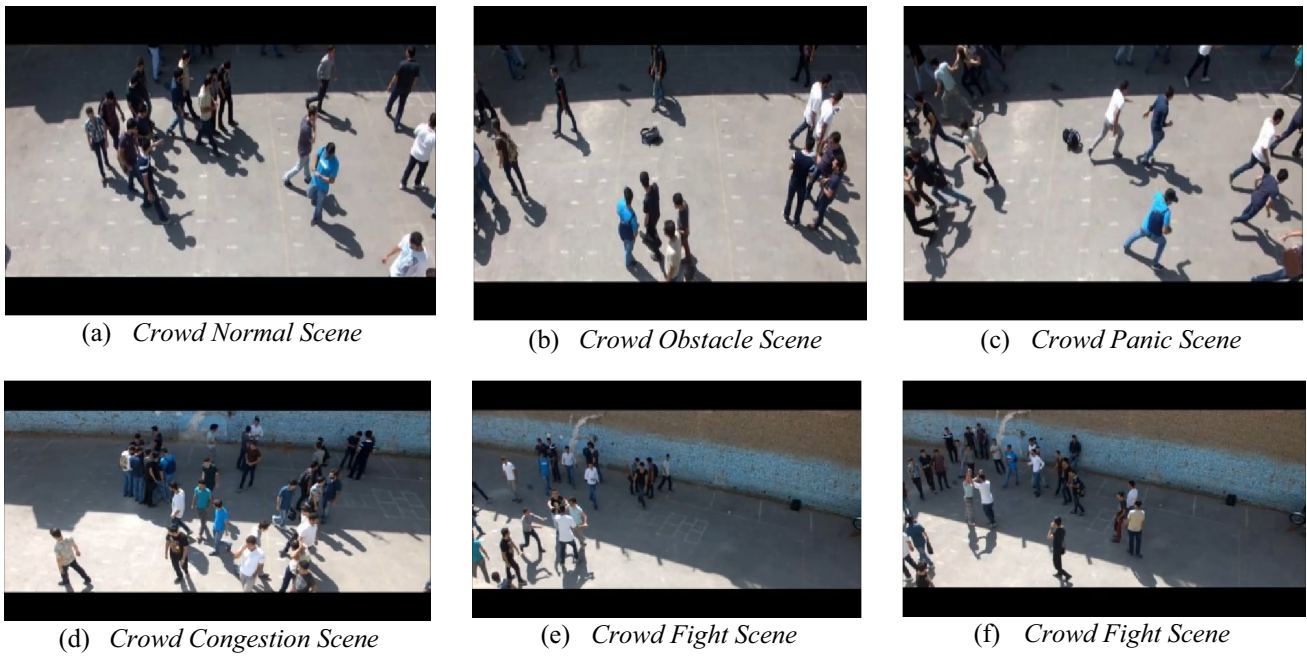


Fig. 3 Examples of different samples of the MED dataset



Fig. 4 Examples of different samples of the GTA dataset

Table 2 Confusion matrix

| Actual vs. predicted | Predicted classes | | Total number of instances |
|----------------------|-------------------|----------|---------------------------|
| | Positive | Negative | |
| Actual class | | | |
| Positive | TP | FN | $P = TP + FN$ |
| Negative | FP | TN | $N = FP + TN$ |

4.2 Performance metrics

Following the works [22–24] on the MED and GTA dataset, the proposed model has used confusion matrix and mean accuracy (Mean-Acc) as performance metrics. In addition to this, the proposed model also used the overall accuracy and precision of individual classes on the respective datasets. The following Table 2 shows the confusion matrix. Here, TP, TN, FP, and FN represent true positive, true negative, false

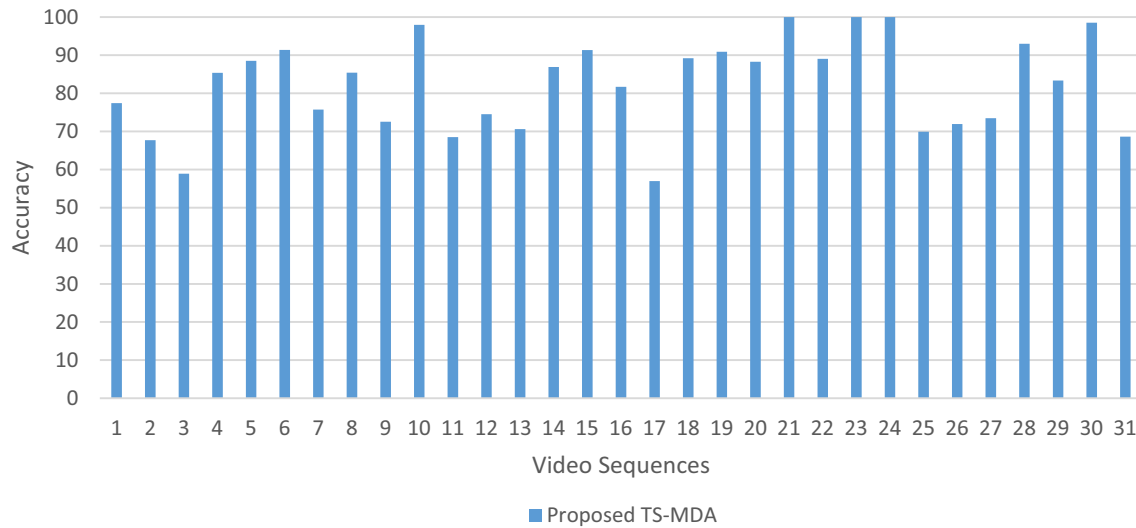


Fig. 5 Accuracies obtained by the proposed model using leave-one-sequence-out on the MED dataset

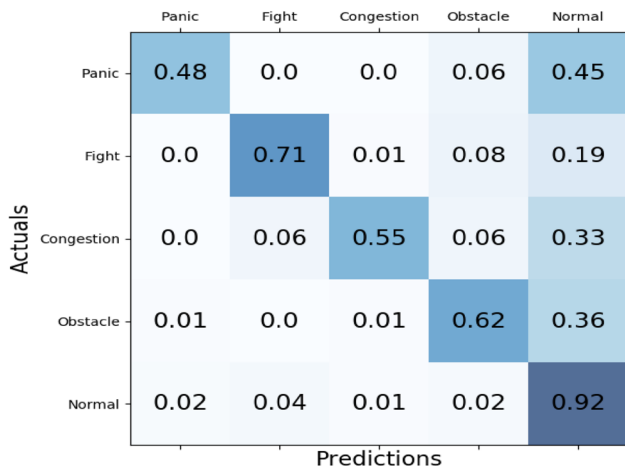


Fig. 6 Confusion matrix of the proposed model on the MED dataset

positive, and false negative. The Mean-Acc is obtained by dividing the sum of the accuracy of individual classes by the total number of classes, which is represented as in Eq. (13):

$$\text{Mean-Acc} = \frac{1}{K} \sum_{i=1}^K \text{Acc}_i \tag{13}$$

here K is the number of classes and Acc_i is the accuracy of the i th class which is calculated as, $\text{Acc}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}$. The precision of i th class can be obtained using Eq. (14):

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{14}$$

From Table 2, the overall accuracy on the dataset can be obtained using Eq. (15):

Table 3 Performance comparison with other state-of-the-art approaches on the MED dataset, bold letters represent best results

| Approaches | Classification accuracy (%) per individual behavior classes | | | | | Mean-ACC (%) | Accuracy (%) |
|-----------------|---|--------------|--------------|--------------|--------------|--------------|--------------|
| | Panic | Fight | Congestion | Obstacle | Normal | | |
| V3G-FC7 [24] | 80.72 | 37.41 | 31.18 | 47.25 | 71.35 | 53.58 | 62.71 |
| V3G-FC8 [24] | 53.23 | 29.89 | 27.32 | 42.35 | 32.16 | 36.99 | 33.82 |
| C3D-FC7 [24] | 84.72 | 32.93 | 16.16 | 29.61 | 92.69 | 51.22 | 73.52 |
| C3D-FC8 [24] | 57.32 | 25.89 | 17.22 | 25.51 | 46.64 | 34.50 | 40.59 |
| Trajectory [22] | – | – | – | – | – | 35.30 | – |
| HOG [22] | – | – | – | – | – | 38.80 | – |
| HOF [22] | – | – | – | – | – | 37.69 | – |
| MBH [22] | – | – | – | – | – | 38.53 | – |
| HOT [22] | 62.18 | 38.27 | 25.67 | 28.20 | 36.53 | 38.17 | – |
| DT [22] | 74.82 | 30.47 | 23.43 | 27.94 | 36.88 | 38.71 | – |
| Proposed | 48.20 | 71.32 | 54.50 | 61.70 | 91.51 | 65.45 | 81.26 |

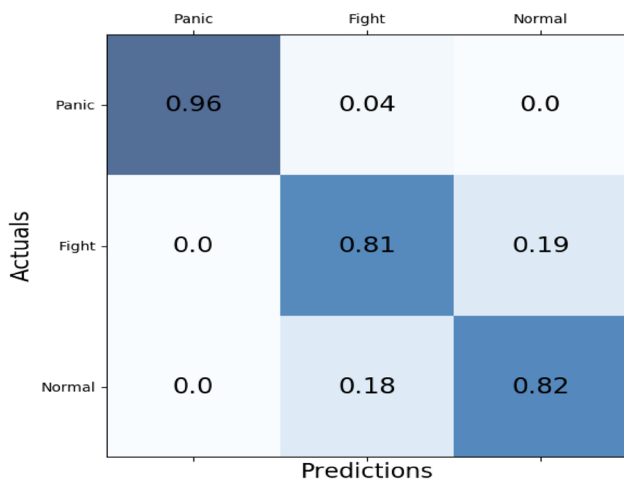


Fig. 7 Confusion matrix of the proposed model on the GTA dataset

Table 4 Performance comparison with state-of-the-art approach on the GTA dataset, bold letters represent best results.

| Approaches | Classification accuracy (%) per individual behavior classes | | | Mean-ACC (%) | Accuracy (%) |
|---------------------------|---|--------------|--------------|--------------|--------------|
| | Normal | Panic | Fight | | |
| Spatial-temporal net [23] | 83.80 | 61.20 | 28.90 | 71.70 | – |
| Proposed | 81.88 | 95.60 | 80.75 | 86.07 | 88.61 |

$$Accuracy = \frac{TP + TN}{T} \tag{15}$$

where T is the total number of test frames.

5 Experiment and results analysis

The program is written in Python using TensorFlow and Keras. The batch size, learning rate for the datasets are set to 128 and 0.01, respectively. Measures like early stopping and kernel regularization have been adopted to avoid overfitting the proposed model. The early stopping is used to halt the network to avoid overfitting the dataset. The patience value of early stopping is set to 10. The model is trained until the early stopping criteria are satisfied. The L_2 norm is used for kernel regularization, whose value is set to 0.01.

5.1 The MED dataset

The procedure for training and testing is followed as prescribed in [22], i.e., leave-one-sequence-out cross-validation

is performed to evaluate the model's performance. In each execution of leave-one-sequence-out, the train set contains 30 percent of video sequences of the entire training set covering all classes of samples. Figure 5 shows a graphical representation of the experimental results obtained on the MED dataset during leave-one-sequence-out. As shown in Fig. 5, the proposed model achieves accuracies of 77.43, 67.71, 58.91, 85.38, 88.54, 91.40, 75.73, 85.41, 72.53, 97.95, 68.5, 74.51, 70.62, 86.91, 91.33, 81.69, 56.98, 89.22, 90.89, 88.29, 100.00, 89.03, 100.00, 100.00, 69.92, 71.94, 73.46, 92.98, 83.34, 98.52 and 68.63% sequentially on 31 video sequences. Figure 6 shows the heatmap of the confusion matrix on the MED dataset. The proposed model achieves classification accuracies of 48.20, 71.32, 54.50, 61.70, and 91.51% on the Panic, Fight, Congestion, Obstacle, and Neutral or Normal crowd behaviors, respectively. The performance comparisons of the proposed model with state-of-the-art approaches are illustrated in Table 3. Deep models [24] like V3G-FC7, V3G-FC8, C3D-FC7, and C3D-FC8 are used for performance analysis. Similarly, conventional machine learning approaches like trajectory-based, HOG, HOF, MBH, HOT, and DT techniques are also used for performance comparison. It can be observed from Table 3 that the proposed model achieves the highest mean accuracy and overall accuracy of 65.45 and 81.26%, respectively. Hence, the proposed feature learning process performs better than the recent state-of-the-art approaches.

5.2 The GTA dataset

The experiment on the GTA dataset is demonstrated by following the same procedure as mentioned in [23]. The behavior sequences 2, 4, 11 and 12 are the test sequences which were selected randomly. The confusion matrix of the proposed model on the GTA dataset is illustrated in Fig. 7. The classification accuracies on the Normal, Panic, and Fight crowd behaviors are 81.88, 95.60, and 80.75%, respectively. The proposed model achieves an overall accuracy on the test samples of 88.61%. The performance comparison with the state-of-the-art approach is illustrated in Table 4. The spatial-temporal model [23] is the only model which experimented on the GTA dataset. The spatial-temporal model [23] achieves classification accuracies of 83.80, 61.20, and 28.90% on the Normal, Panic, and Fight crowd behaviors on the GTA dataset. The mean accuracy of the spatial-temporal model [23] is 71.70, whereas the proposed model achieves the mean accuracy of 86.07%. Hence, the proposed model performs better than the spatial-temporal model [23].

5.3 Results analysis against OCC methods

Apart from comparing the performance of the proposed TS-MDA against multi-class-based crowd behavior prediction

Fig. 8 Comparison of the OCC approaches against the proposed TS-MDA using leave-one-sequence-out on the MED dataset

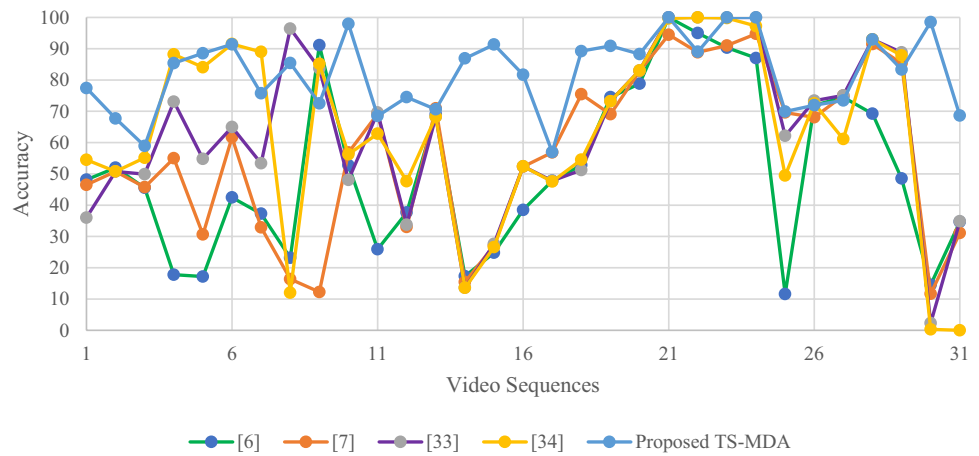


Table 5 Comparison of results of the proposed TS-MDA against the OCC-based approaches

| Dataset | | [6] | [7] | [33] | [34] | Proposed model |
|--------------------------|------------|--------|-------|--------|--------|----------------|
| MED | | | | | | |
| For one class approach | | | | | | |
| Per-class accuracy (%) | Normal | 90.67 | 72.92 | 88.86 | 90.99 | – |
| | Abnormal | 07.63 | 33.10 | 14.68 | 24.48 | – |
| Accuracy (%) | | 61.65 | 58.76 | 62.94 | 67.78 | – |
| For multi-class approach | | | | | | |
| Per-class accuracy (%) | Panic | – | – | – | – | 48.20 |
| | Fight | – | – | – | – | 71.32 |
| | Congestion | – | – | – | – | 54.50 |
| | Obstacle | – | – | – | – | 61.70 |
| | Normal | – | – | – | – | 91.51 |
| Accuracy (%) | | – | – | – | – | 81.26 |
| GTA | | | | | | |
| For one class approach | | | | | | |
| Per-class accuracy (%) | Normal | 0.00 | 0.00 | 100.00 | 0.00 | – |
| | Abnormal | 100.00 | 85.58 | 0.00 | 100.00 | – |
| Accuracy (%) | | 70.04 | 59.94 | 29.95 | 70.04 | – |
| For multi-class approach | | | | | | |
| Per-class accuracy (%) | Panic | – | – | – | – | 80.75 |
| | Fight | – | – | – | – | 95.60 |
| | Normal | – | – | – | – | 81.88 |
| Accuracy (%) | | – | – | – | – | 88.61 |

models, we have also compared the performance against one-class classification (OCC) or formally anomaly detection approaches. For this, we have implemented four different state-of-the-art approaches [6, 7, 33, 34] on the MED and GTA datasets. The training and testing procedures on MED and GTA for implementing OCC-based approaches are the same as mentioned in [22] and [23], respectively. Each dataset is divided into two classes: normal and abnormal, where the abnormal class consists of all the samples of non-normal classes like Panic, Congestion, Fight, Obstacle. However, while implementing these methods, we have followed all the

procedures mentioned in their papers, but a few things have been adopted, such as the input shape of $[236 \times 156 \times 3]$ is considered instead of $[235 \times 155 \times 3]$ for [33]. Again, the same threshold that showed the best result in [33] has been considered during one-class classification. The value of ν in [6] is not mentioned; therefore, we have implemented [6] with different values of ν , and it is observed that at $\nu=0.0001$, the model [6] performs better on the MED and the GTA datasets.

Figure 8 graphically compares the accuracy of the proposed TS-MDA and the OCC-based methods during

leave-one-out-sequence validation on the MED dataset. The proposed model performs better than state-of-the-arts except on a few video sequences of the MED dataset. For example, Bouindour et al. [6] performs better than the TS-MDA on video sequences 9, 22, and 27. Smeureanu et al. [7] have better performance than TS-MDA on video sequences 11, 13, 27, and 29. Ribeiro et al. [33] and Gutoski et al. [34] got better performance than TS-MDA on video sequences <8, 9, 11, 22, 26, 27, 29> and <4, 7, 9, 22, 26, 29> respectively. According to Fig. 5, only Gutoski et al. [34] got an accuracy of nearly zero on two sequences. Gutoski et al. [34] achieves 0.3 and 0.00% in the video sequences 30 and 31, respectively. Table 5 shows a comparison of results of TS-MDA against OCC-based models [6, 7, 33, 34]. Bouindour et al. [6], Smeureanu et al. [7], Gutoski et al. [34], and Ribeiro et al. [33] achieve overall detection accuracies of 53.25, 58.76, 67.78, and 62.94% on the MED dataset. On the other hand, the TS-MDA gets an accuracy of 81.26% on the MED dataset, which is relatively more than OCC-based approaches [6, 7, 33, 34]. One thing could be observed from Table 5 that the OCC-based approaches perform better in detecting Normal behavior sequences than abnormal

sequences on the MED dataset. However, the scenario is entirely changed for OCC-based methods on the GTA dataset. Approaches like Bouindour et al. [6] and Gutoski et al. [34] do not classify any of the "Normal" test sequences as "Normal" whereas the abnormal detection accuracies reached 100.00, 85.58, and 100.00% respectively.

However, Ribeiro et al. [33] achieve 100.00% accuracy for "Normal" sequences and 0.00% accuracy for abnormal sequences. Such accuracy deviation may be due to the computer graphics-based simulated dataset. Bouindour et al. [6], Smeureanu et al. [7], Gutoski et al. [34], and Ribeiro et al. [33] achieve the overall accuracies of 70.04, 59.94, 70.04, and 29.95% on the GTA datasets which are lower than the proposed TS-MDA model. Therefore, based on the above analysis, we can summarize that there are no OCC methods [6, 7, 33, 34] that confirm their effectiveness on the MED and the GTA datasets.

5.4 Ablation study

An ablation study on the proposed model has been performed to show the effectiveness of each of its main

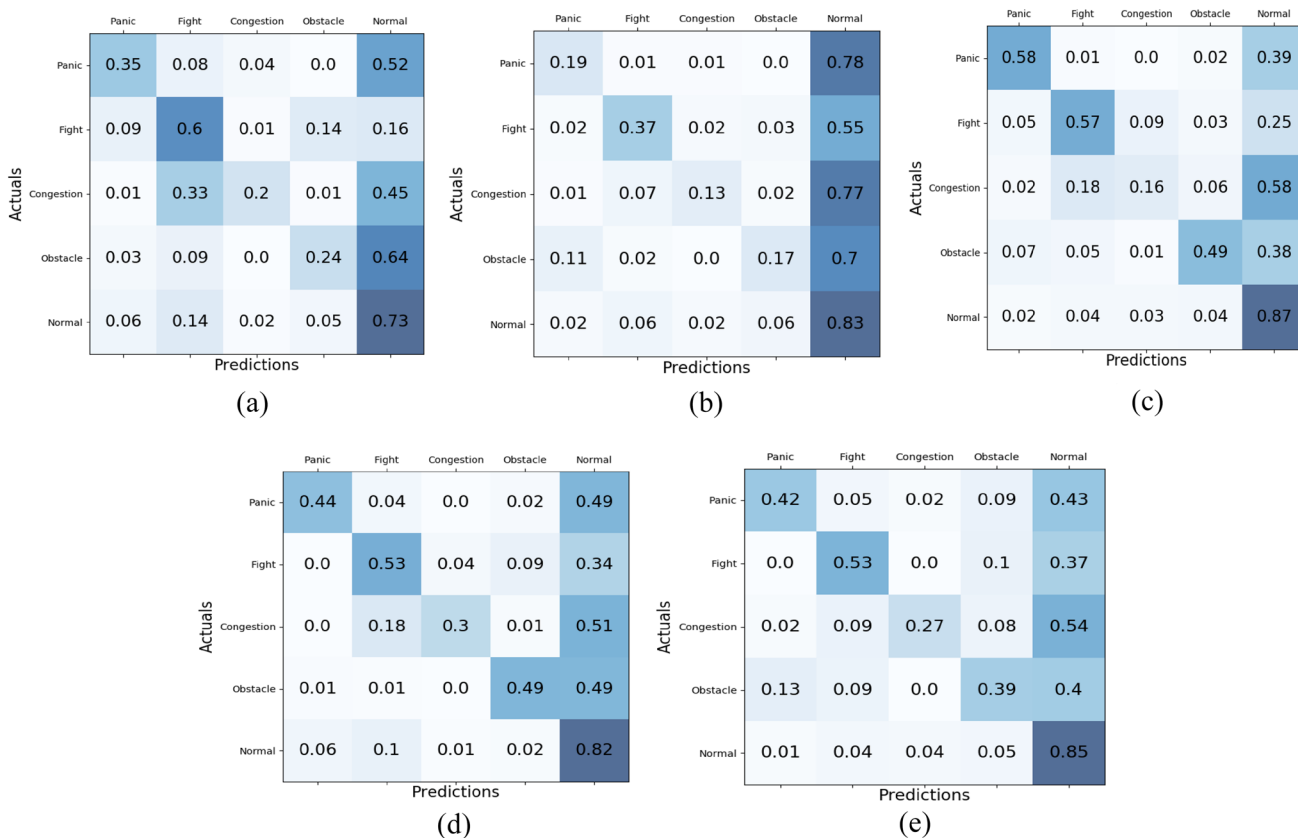


Fig. 9 Confusion metrics of different modules during ablation study on the MED dataset. The subfigures a–e are the confusion metrics of MSS, MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA modules, respectively

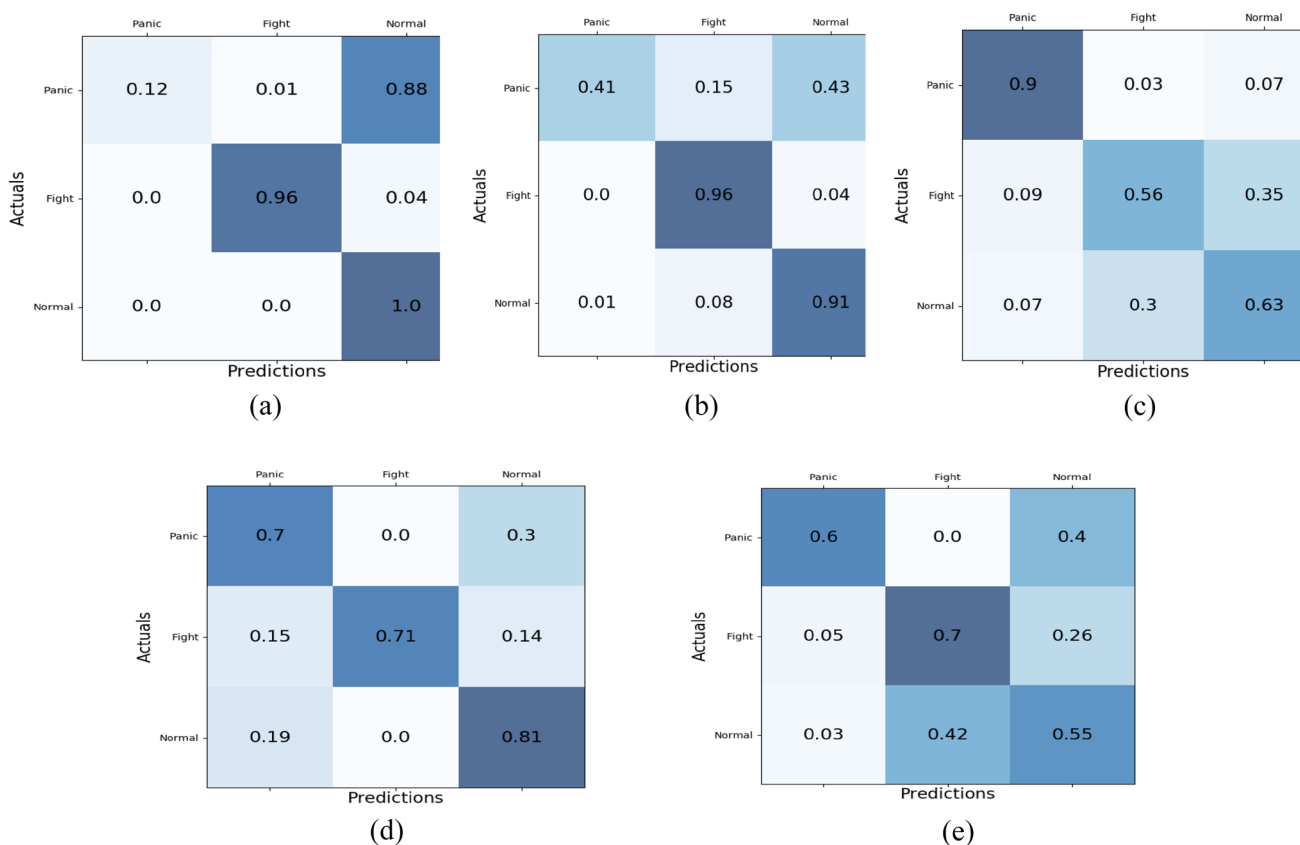
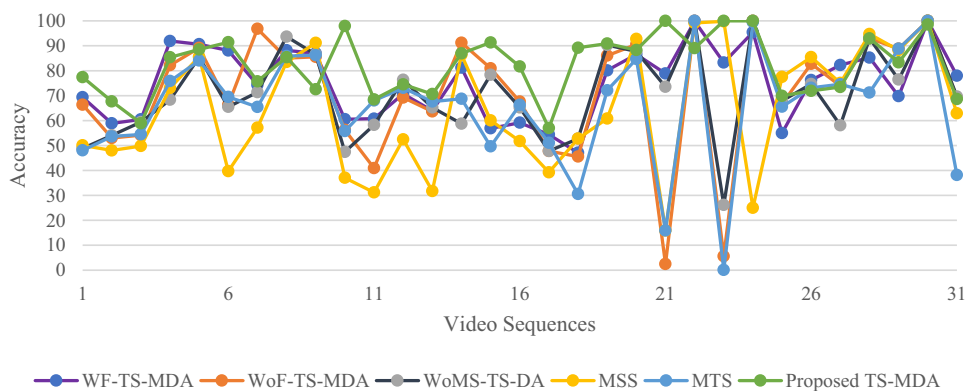


Fig. 10 Confusion metrics of different modules during ablation study on the GTA dataset. The subfigures a–e are the confusion metrics of MSS, MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA modules, respectively

Fig. 11 Comparison of accuracies of different models during ablation study using leave-one-sequence-out cross-validation on the MED dataset



modules. The proposed model contains two main modules: the multiscale spatial stream (MSS) and the multiscale temporal stream (MTS). Experiments are conducted by considering the MSS and MTS individually for crowd behavior classification. Apart from the MSS and MTS, other possible models have been obtained based on the inputs given to the two streams and the multiscale feature fusion. These possible models are,

- With foreground maps applied to inputs of the TS-MDA (WF-TS-MDA): in this case, the foreground maps are applied to the inputs of the two streams of the TS-MDA.
- Without foreground maps applied to inputs of the TS-MDA (WoF-TS-MDA): In this model, no foreground maps are applied to the inputs of two streams.

Table 6 Performance comparison of different modules of the proposed model

| Datasets | Approaches | Per class accuracy (in %) | | | | | Per-class precision (in %) | | | | | Overall accuracy (in %) |
|----------------|--------------------|---------------------------|-------|------------|----------|--------|----------------------------|--------|------------|----------|--------|-------------------------|
| | | Panic | Fight | Congestion | Obstacle | Normal | Panic | Fight | Congestion | Obstacle | Normal | |
| The MED | WF-TS-MDA | 57.94 | 57.34 | 7.24 | 48.54 | 87.17 | 53.60 | 59.36 | 22.10 | 64.04 | 81.16 | 73.17 |
| | WoF-TS-MDA | 44.45 | 53.01 | 30.00 | 49.31 | 82.08 | 37.31 | 43.15 | 69.11 | 69.83 | 76.71 | 69.00 |
| | WoMS-TS-DA | 41.50 | 52.60 | 27.40 | 38.60 | 85.43 | 43.30 | 56.28 | 39.59 | 47.24 | 79.20 | 70.11 |
| | The MSS module | 35.41 | 60.30 | 20.36 | 23.73 | 73.27 | 25.16 | 35.35 | 47.10 | 35.56 | 76.98 | 60.99 |
| | The MTS module | 19.38 | 37.42 | 13.40 | 17.43 | 83.43 | 25.95 | 47.38 | 31.37 | 31.36 | 70.43 | 63.03 |
| The GTA | The proposed model | 48.20 | 71.28 | 54.50 | 61.70 | 91.51 | 66.73 | 75.26 | 78.28 | 68.46 | 84.96 | 81.26 |
| | WF-TS-MDA | 90.13 | 55.98 | - | - | 62.80 | 92.14 | 51.23 | - | - | 64.28 | 75.33 |
| | WoF-TS-MDA | 70.47 | 70.77 | - | - | 80.87 | 80.50 | 100.00 | - | - | 57.78 | 73.64 |
| | WoMS-TS-DA | 59.65 | 69.78 | - | - | 54.80 | 94.51 | 51.60 | - | - | 39.25 | 60.16 |
| | The MSS module | 11.73 | 96.17 | - | - | 98.12 | 100.00 | 98.12 | - | - | 39.89 | 54.51 |
| The MTS module | 41.58 | 95.89 | - | - | 91.12 | 98.52 | 64.61 | - | - | 54.68 | 66.89 | |
| | The Proposed model | 95.59 | 80.75 | - | - | 81.88 | 100.00 | 67.59 | - | - | 86.30 | 88.61 |

- Without multiscale feature fusion on the TS-MDA (WoMS-TS-MDA): here, no multiscale features are fused in the Concatenate layer of the proposed TS-MDA.

Figure 9a–e shows the confusion matrix of the MSS, MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA modules, respectively, on the MED dataset. Similarly, Fig. 10a–e illustrates the confusion matrix of the MSS, MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA on the GTA dataset. Figure 11 shows a graphical comparison of different models used in the ablation study for the MED sequences.

According to Fig. 11, the proposed model shows better accuracy trend as compared to other models. However, the performance of the proposed model degraded on few sequences. For example, different modules, such as WF-TS-MDA, WoF-TS-MDA, WoMS-TS-DA, MSS, MTS, perform better than the proposed model on the sequences <3, 4, 5, 8, 9, 22, 26, 27, 30, 31>, <5, 7, 9, 14, 20, 22, 26, 27, 29, 30>, <3, 8, 9, 12, 22, 26, 30>, <9, 20, 22, 25, 26, 27, 28, 29, 30> and <8, 9, 22, 26, 27, 29, 30> respectively. Apart from this, the performance comparison of different modules is illustrated in Table 6. Performance metrics like per-class accuracy, overall accuracy, and per-class precision are obtained for different modules. The models like WF-TS-MDA, WoF-TS-MDA, WoMS-TS-MDA, MSS and MTS achieve overall accuracies of 73.17, 69.00, 70.11, 60.99, 63.03 on MED and 75.33%, 73.64, 60.16, 54.51, 69.89% on GTA dataset respectively. However, the proposed TS-MDA achieves better performance as compared with individual modules. From the confusion matrixes i.e., Figs. 9a, b and 10a, b, the MTS module tends to classify anomaly frames into normal frames on the MED dataset, but the same trend is not seen in the GTA dataset; this may be due to several reasons. First, the MED dataset is more realistic than the GTA dataset and contains more anomaly classes than the GTA dataset. Second, the more the different types of anomaly classes, the more similar will be the motion patterns compared to the “Normal” class, and thus, the MTS module tends to classify a more significant number of anomaly frames as Normal frames. However, among several modules used in the ablation study, the WF-TS-MDA performs better in the MED and GTA datasets. Multiscale features and minimizing the background effects are essential for crowd behavior modeling. The model without multiscale features, i.e., WoMS-TS-MDA, performs poorly compared to TS-MDA. Hence the multiscale features are essential for crowd behavior modeling. Similarly, the effect of foreground maps is also observed. The model without foreground maps (WoF-TS-MDA) achieves much less accuracy than WF-TS-MDA and TS-MDA. This is because the inputs to the MSS and MTS are affected by cluttered backgrounds.

Now, as far as the decision on applying foreground maps to both the streams, it has been observed that the proposed model (TS-MDA) with foreground maps applied to the MTS stream performs better than WF-TS-MDA. Therefore, it can be summarized that the proposed TS-MDA effectively handles scale variation issue and also utilize the de-background temporal features for crowd behavior modeling. There is another issue which need to be discussed as far as the difference of accuracies for the two datasets. This occurs due to: first, the MED dataset is a real-world dataset having five different types of crowd behaviors, whereas the GTA dataset is computer graphics (CG) data, which contains only three crowd behavior classes and second, the more the number of behavior classes, the more similar the appearance and motion patterns between them will be. Hence, it will be challenging to achieve better performance as far as the MED dataset is concerned. Nevertheless, the proposed model achieves better performance as far as state-of-the-art approaches are concerned.

6 Conclusion and future work

This paper proposes a two-stream multiscale deep architecture (TS-MDA) for multi-class classification-based crowd behavior prediction (MCC-based CBP). The motivation behind the design of the proposed model is to handle human shape change due to perspective distortion and minimizing background influences during CBP. The former challenge is handled by extracting multiscale features from the crowd scene, and the latter is addressed by adopting the universal background subtractor, i.e., ViBE algorithm, and obtaining foreground images from the frames. The first stream, i.e., the MSS of the proposed model, can extract multiscale spatial/appearance features from the scene, whereas the second stream, i.e., the MTS, extracts multiscale temporal features from the volume of foreground of frames. All the multiscale features are concatenated and used to classify crowd behaviors. For experiment analysis, two large-scale crowd behavior datasets: MED and GTA, have been used. The proposed model achieves 81.26 and 88.61% accuracy on the MED and GTA datasets. By observing the performance comparisons from Tables 3 and 4, it can be concluded that the proposed model performs better than the state-of-the-art approaches. The performance of the proposed model is also compared to the OCC-based approaches (illustrated in Table 5), which confirms that no OCC methods show their effectiveness on the MED and the GTA datasets. An ablation study has also been conducted to show the effect of individual streams of TS-MDA. The ablation study was also conducted to show the effectiveness of the proposed model based on the foreground maps applied to the inputs and the influence of multiscale features. It can be observed from Table 6 that the

proposed TS-MDA performs better than other modules used in the ablation study for both the MED and GTA datasets. Hence, it can be concluded that the proposed model can provide better accuracy while dealing with the scale variation issue and minimizing the background influence. The future work will focus on improving the performance of the MCC-based CBP further.

Acknowledgements The support and the resources provided by 'PARAM Shivay Facility' under the National Supercomputing Mission, Government of India at the Indian Institute of Technology, Varanasi, are gratefully acknowledged.

References

1. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 FPS in MATLAB. *Proc IEEE Int. Conf. Comput. Vis.* (2013). <https://doi.org/10.1109/ICCV.2013.338>
2. Cheng, K.W., Chen, Y.T., Fang, W.H.: Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* (2015). <https://doi.org/10.1109/CVPR.2015.7298909>
3. Saligrama, V., Chen, Z.: Video anomaly detection based on local statistical aggregates. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* (2012). <https://doi.org/10.1109/CVPR.2012.6247917>
4. Lamba, S., Nain, N.: Detecting anomalous crowd scenes by oriented Tracklets' approach in active contour region. *Multimed. Tools Appl.* **78**, 31101–31120 (2019). <https://doi.org/10.1007/s11042-019-07806-8>
5. Zhou, S., Shen, W., Zeng, D., et al.: Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes. *Signal Process. Image Commun.* **47**, 358–368 (2016). <https://doi.org/10.1016/j.image.2016.06.007>
6. Bouindour, S., Hittawe, M.M., Mahfouz, S., Snoussi, H.: Abnormal event detection using convolutional neural networks and 1-Class SVM classifier. 1–6 (2018). <https://doi.org/10.1049/ic.2017.0040>
7. Smeureanu, S., Ionescu, R.T., Popescu, M., Alexe, B.: Deep appearance features for abnormal behavior detection in video. In: *Image Analysis and Processing—ICIAP 2017* (2017)
8. Ravanbakhsh, M., Nabi, M., Mousavi, H., et al.: Plug-and-play CNN for crowd motion analysis: an application in abnormal event detection. In: *Proc—2018 IEEE Winter Conf. Appl. Comput. Vision, WACV 2018-Janua*, pp. 1689–1698. <https://doi.org/10.1109/WACV.2018.00188> (2018)
9. Bouindour, S., Snoussi, H., Hittawe, M., et al.: An on-line and adaptive method for detecting abnormal events in videos using spatio-temporal ConvNet. *Appl. Sci.* **9**, 757 (2019). <https://doi.org/10.3390/app9040757>
10. Song, W., Zhang, D., Zhao, X., et al.: A novel violent video detection scheme based on modified 3D convolutional neural networks. *IEEE Access* **7**, 39172–39179 (2019). <https://doi.org/10.1109/ACCESS.2019.2906275>
11. Dinesh Jackson, S.R., Fenil, E., Gunasekaran, M., et al.: Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM. *Comput. Netw.* **151**, 191–200 (2019). <https://doi.org/10.1016/j.comnet.2019.01.028>
12. Sabokrou, M., Fathy, M., Hoseini, M., Klette, R.: Real-time anomaly detection and localization in crowded scenes. *IEEE Comput.*

- Soc. Conf. Comput. Vis. Pattern Recognit. Work (2015). <https://doi.org/10.1109/CVPRW.2015.7301284>
13. Xu, D., Ricci, E., Yan, Y., et al.: Learning deep representations of appearance and motion for anomalous event detection. Proc. Br. Mach. Vis. Conf. (2015). <https://doi.org/10.5244/C.29.8>
 14. George, M., Jose, B.R., Mathew, J., Kokare, P.: Autoencoder-based abnormal activity detection using parallelepiped spatio-temporal region. IET Comput. Vis. **13**, 23–30 (2018). <https://doi.org/10.1049/iet-cvi.2018.5240>
 15. Tran, H.T.M., Hogg, D.: Anomaly detection using a convolutional autoencoder. Winner-take-all (2017)
 16. Chong, Y.S., Tay, Y.H.: Abnormal event detection in videos using spatiotemporal autoencoder. Lect. Notes Comput. Sci. (Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.) **10262**, 189–196 (2017). https://doi.org/10.1007/978-3-319-59081-3_23
 17. Sabokrou, M., Fayyaz, M., Fathy, M., Klette, R.: Deep-cascade: cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes. IEEE Trans. Image Process. **26**, 1992–2004 (2017). <https://doi.org/10.1109/TIP.2017.2670780>
 18. Ravanbakhsh, M., Nabi, M., Sangineto, E., et al.: Abnormal event detection in videos using generative adversarial nets. In: ICIP, pp. 1577–1581. (2017). <https://doi.org/10.1109/ICIP.2017.8296547>
 19. Ravanbakhsh, M., Sangineto, E., Nabi, M., Sebe, N.: Training adversarial discriminators for cross-channel abnormal event detection in crowds. In: Proc—2019 IEEE Winter Conf. Appl. Comput. Vision, WACV, 2019, pp. 1896–1904. <https://doi.org/10.1109/WACV.2019.00206> (2019)
 20. Zhuang, N.: Convolutional DLSTM for crowd scene understanding. <https://doi.org/10.1109/ISM.2017.19> (2017)
 21. Yang, B., Cao, J., Wang, N., Liu, X.: Anomalous behaviors detection in moving crowds based on a weighted convolutional autoencoder-long short-term memory network. IEEE Trans. Cogn. Dev. Syst. (2018). <https://doi.org/10.1109/TCDs.2018.2866838>
 22. H. Rabiee, J. Haddadnia, H. Mousavi, M. Kalantarzadeh, M. Nabi and V. Murino, Novel dataset for fine-grained abnormal behavior understanding in crowd. In: 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 95–101 (2016). <https://doi.org/10.1109/AVSS.2016.7738074>
 23. Lazaridis, L., Dimou, A., Daras, P.: Abnormal behavior detection in crowded scenes using density heatmaps and optical flow. Eur. Signal Process. Conf. (2018). <https://doi.org/10.23919/EUSIPCO.2018.8553620>
 24. Dupont, C., Tobias, L., Luvison, B.: Crowd-11: a dataset for fine grained crowd behaviour analysis. In: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work 2017-July, pp. 2184–2191. <https://doi.org/10.1109/CVPRW.2017.271> (2017)
 25. Sindagi, V.A., Patel, V.M.: HA-CCN: hierarchical attention-based crowd counting network. IEEE Trans. Image Process. **29**, 323–335 (2020). <https://doi.org/10.1109/TIP.2019.2928634>
 26. Tripathy, S.K., Srivastava, R.: A real-time two-input stream multi-column multi-stage convolution neural network (TIS-MCMS-CNN) for efficient crowd congestion-level analysis. Multimed. Syst. **26**, 585–605 (2020). <https://doi.org/10.1007/s00530-020-00667-4>
 27. Aldissi, B., Ammar, H.: Real-time frequency-based detection of a panic behavior in human crowds. Multimed. Tools Appl. **79**, 24851–24871 (2020). <https://doi.org/10.1007/s11042-020-09024-z>
 28. Singh, G., Khosla, A., Kapoor, R.: Crowd escape event detection via pooling features of optical flow for intelligent video surveillance systems. Int. J. Image Graph Signal Process. **11**, 40–49 (2019). <https://doi.org/10.5815/ijigsp.2019.10.06>
 29. Sabokrou, M., Fayyaz, M., Fathy, M., et al.: Deep-anomaly: fully convolutional neural network for fast anomaly detection in crowded scenes. Comput. Vis. Image Underst. **172**, 88–97 (2018). <https://doi.org/10.1016/j.cviu.2018.02.006>
 30. Huang, S., Huang, D., Zhou, X.: Learning multimodal deep representations for crowd anomaly event detection. Math Probl Eng (2018). <https://doi.org/10.1155/2018/6323942>
 31. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: the all convolutional net (2014)
 32. Ammar, H., Cherif, A.: DeepROD: a deep learning approach for real-time and online detection of a panic behavior in human crowds. Mach. Vis. Appl. (2021). <https://doi.org/10.1007/s00138-021-01182-w>
 33. Ribeiro, M., Lazzaretti, A.E., Lopes, H.S.: A study of deep convolutional auto-encoders for anomaly detection in videos. Pattern Recognit. Lett. **105**, 13–22 (2018). <https://doi.org/10.1016/j.patrec.2017.07.016>
 34. Gutoski, M., Marcelo, N., Aquino, R., et al.: Detection of video anomalies using convolutional autoencoders and one-class support vector machines. In: XIII Brazilian Congr. Comput. Intell. 2017 (2017)
 35. Sang, J., Wu, W., Luo, H., et al.: Improved crowd counting method based on scale-adaptive convolutional neural network. IEEE Access **7**, 24411–24419 (2019). <https://doi.org/10.1109/ACCESS.2019.2899939>
 36. Barnich, O., Van Droogenbroeck, M.: ViBe: a universal background subtraction algorithm for video sequences. IEEE Trans. Image Process. **20**, 1709–1724 (2011). <https://doi.org/10.1109/TIP.2010.2101613>
 37. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 221–231 (2013). <https://doi.org/10.1109/TPAMI.2012.59>
 38. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. 1–15 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.