CHAPTER 5
CROWD BEHAVIOUR ANALYSIS USING MACHINE LEARNING AND DEEP LEARNING APPROACHES

5.1 Introduction

In this chapter, two deep models are proposed for crowd behavior analysis. The first model, i.e., "MuST-POS: Multiscale Spatial-Temporal 3D Atrous-Net and PCA-guided OC-SVM for Crowd Panic Detection (CPD)". On the contrary, the second model, i.e., "TS-MDA: Two-Stream Multiscale Deep Architecture for Crowd Behavior Prediction (CBP)," is designed to understand and predict different crowd behavior scenes like normal, panic, congestion, fight, and abnormal activities. The proposed MuST-POS is designed using conventional machine learning and deep learning technique and also provide the solution using the One-Class Classification (OCC) approach. However, the proposed TS-MDA is solely designed using deep techniques, which solves the objective function as a multi-class classification (MCC)-based approach. Most of the existing crowd panic detection models have adopted conventional [160][161][164] and deep learning approaches [11], but the performance of such models is degraded due to a lack of handling human shape variations in the crowd videos, which is the motivation behind the proposed MuST-POS. The MuST-POS exploits both multiscale spatial and multiscale temporal features to form scale-invariant spatial-temporal features to handle crowd shape changes in the crowd videos.

Further, the existing MCC-based crowd behavior prediction models do not handle human shape variation and minimize background influence, resulting in poor

performance. In addition, the OCC-based approaches do not consider the dissimilarities between several anomalies and treat them as a single class. So, to address these issues, the second model, i.e., TS-MDA, is proposed. The details of these approaches are described in the subsequent sections.

5.2 MuST-POS: Multiscale Spatial-Temporal 3D Atrous-Net and PCA guided OC-SVM for Crowd Panic Detection

5.2.1 Proposed Method and Model

The proposed model utilizes deep learning as well as conventional machine learning concepts. The overall block diagram of the proposed model is illustrated in Figure 5.1, and the detail of the proposed MuST-POS model is displayed in Figure 5.2. The proposed model consists of three modules: (i) a deep model, i.e., Multiscale Spatial-Temporal 3D Atrous-Net (MuST-3AN), (ii) Principal Component Analysis (PCA), and (iii) One-Class Support Vector Machine (OC-SVM). The MuST-3AN is designed to learn the normal crowd behavior patterns by exploiting multiscale spatial-temporal features. The PCA reduces the dimensionality of the extracted multiscale features from the MuST-3AN, and the OC-SVM is used to predict the outliers, which are nothing but panic behaviors. The following subsections will explain the proposed work in details,

- Architecture Details of MuST-3AN.
- Pre-Processing.
- Multiscale Appearance (spatial) Temporal feature Extraction.
- Dimension Reduction.
- Crowd Panic Detection using OC-SVM.

5.2.1.1 Architecture Details of MuST-3AN

According to Figure 5.2, the proposed MuST-3AN mainly has two streams: The multiscale appearance stream (MAS) and the multiscale temporal stream (MTS). Both

MAS and MTS are built on the same number of 3D atrous blocks. Each of these two streams has seven such blocks but different in the number of kernels. The details of these blocks are mentioned in Table 5.1.

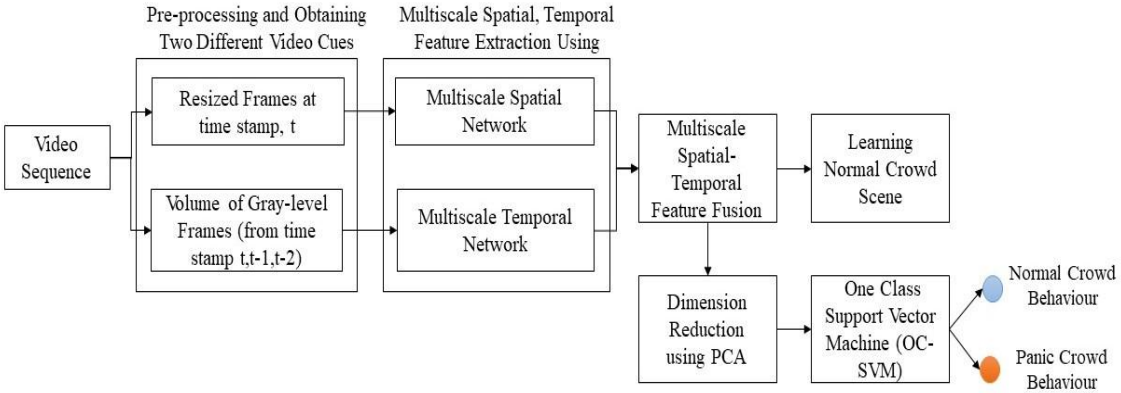

Figure 5.1: Overall block diagram of the proposed MuST-POS

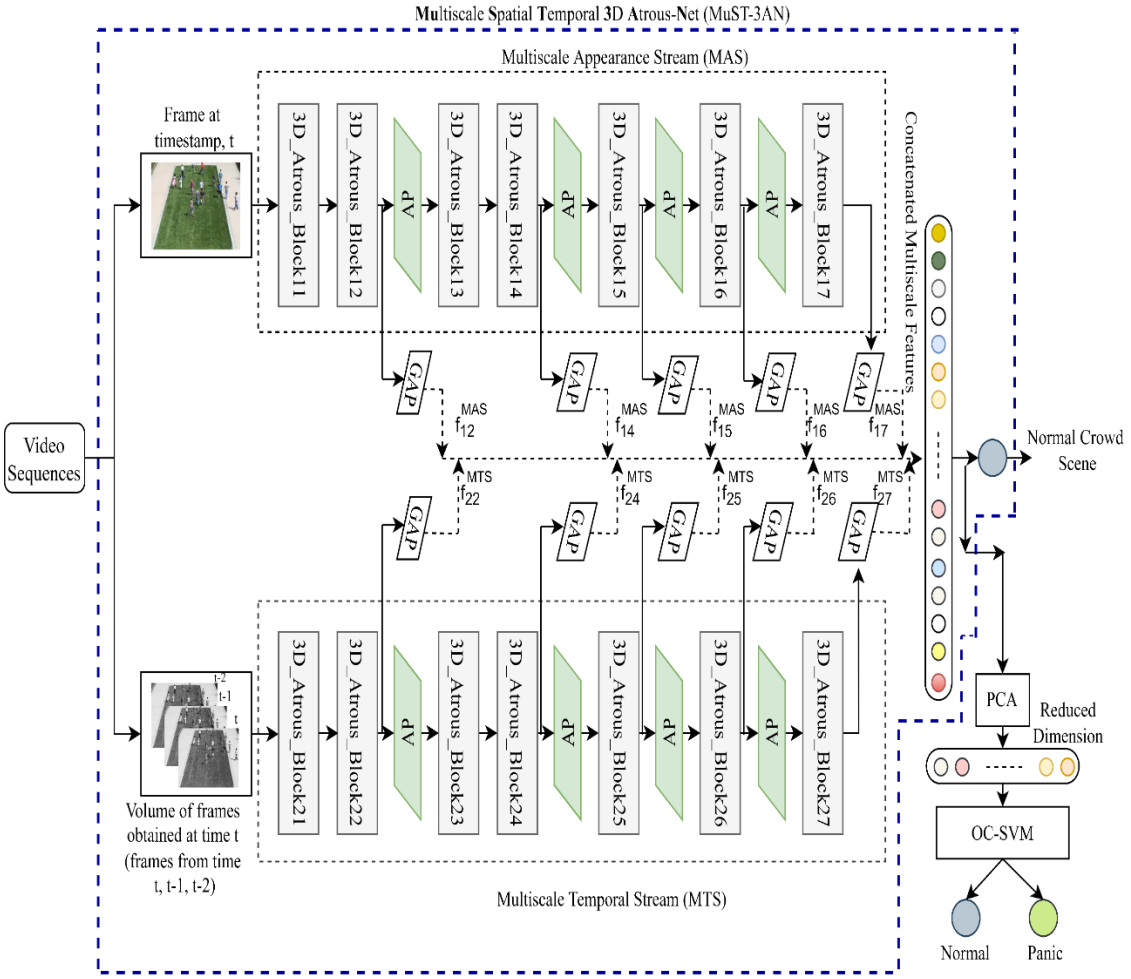

Figure 5.2: The architecture of the proposed MuST-POS

Table 5.1: Block details of the MuST-POS

Block Name	Layer Name	No. of Kernels	Kernel Size	Dilation Rate
3D_Atrous_Block11	Conv3D	32	5×5×5	3
	LeakyReLU	NA		
	BN			
3D_Atrous_Block12	Conv3D	64	5×5×5	3
	LeakyReLU	NA		
	BN			
3D_Atrous_Block13	Conv3D	128	4×4×4	2
	LeakyReLU	NA		
	BN			
3D_Atrous_Block14	Conv3D	256	3×3×3	2
	LeakyReLU	NA		
	BN			
3D_Atrous_Block15	Conv3D	300	3×3×3	1
	LeakyReLU	NA		
	BN			
3D_Atrous_Block16	Conv3D	300	3×3×3	1
	LeakyReLU	NA		
	BN			
3D_Atrous_Block17	Conv3D	320	2×2×2	1
	LeakyReLU	NA		
	BN			
3D_Atrous_Block21	Conv3D	16	5×5×5	3
	LeakyReLU	NA		
	BN			
3D_Atrous_Block22	Conv3D	32	5×5×5	3
	LeakyReLU	NA		
	BN			
3D_Atrous_Block23	Conv3D	64	4×4×4	2
	LeakyReLU	NA		
	BN			
3D_Atrous_Block24	Conv3D	128	3×3×3	2
	LeakyReLU	NA		
	BN			
3D_Atrous_Block25	Conv3D	256	3×3×3	1
	LeakyReLU	NA		
	BN			
3D_Atrous_Block26	Conv3D	320	3×3×3	1
	LeakyReLU	NA		
	BN			
3D_Atrous_Block27	Conv3D	340	2×2×2	1
	LeakyReLU	NA		
	BN			
GAP	Global Average Pooling3D	NA		
AP	AveragePooling3D	NA	2×2×2	NA

Each of the 3D_Atrous blocks has three different layers: a dilated Conv3D layer, a Leaky ReLU activation layer, and a batch normalization (BN) layer. Each stream has four 3D average pooling layers (AP) at different stages of the network for performing multiscale analysis. The AP layer downscale the incoming features map to its half of size. The network utilizes the 3D Global Average Pooling (GAP) at different levels of the network to extract the global average of feature maps. We have used zero padding in the Atrous and average pooling layers.

5.2.1.2 Pre-Processing

In the pre-processing stage, we have extracted frames and the volume of frames from the video. The frames are rescaled to $[200 \times 200 \times 3]$. Let the N number of rescaled frames be denoted by a set $S = \{s_1, s_2, \ldots\ldots, s_N\}$. Similarly, let the N number of volume of frames be denoted by a set $V = \{v_1, v_2, \ldots\ldots, v_N\}$. The volume of frames is obtained by stacking grayscale frames from timestamp $t, t-1, t-2$. Each volume element of set V is rescaled to $[200 \times 200 \times 3]$.

5.2.1.3 Multiscale Spatial-Temporal feature extraction

The MAS is designed to extract multiscale spatial features from the frame. The multiscale features can be used to handle scale variation due to perspective distortion. The set S is inputted into the MAS. The MAS has multi-stages of atrous 3D convolution layers. The reason for using the dilated 3D CNN is to cover a larger area on the image or feature map by keeping the kernel parameters the same as the normal convolution layer. For multiscale analysis, we have used four AP layers in four different stages of the MAS module. The activated feature maps from the 3D_Atrous_Block12, 3D_Atrous_Block14, 3D_Atrous_Block15, 3D_Atrous_Block16, and 3D_Atrous_Block17 are used for multiscale spatial/appearance feature analysis. The selected feature maps at different scales are fed into GAP layers for obtaining statistical features. Let the features obtained

from the GAP layers be denoted as $f_{ij}^{MAS}|_{i=1,j=2,4,5,6,7}$, here the variable $i\ and\ j$ represents the column index and layer index of the GAP layer.

The multiscale temporal features are extracted by using the MTS module. The structure of MTS is the same as the MAS except for the number of kernels. The set V is inputted into the MTS module. The activated feature maps from the 3D_Atrous_Block22, 3D_Atrous_Block24, 3D_Atrous_Block25, 3D_Atrous_Block26, and 3D_Atrous_Block27 are used for multiscale temporal feature analysis. GAP layers follow the selected multiscale temporal features to extract statistical features like mean from the input feature maps. Let the multiscale mean features are represented as $f_{ij}^{MTS}|_{i=2,j=2,4,5,6,7}$.

Now, these two multiscale spatial-temporal features are concatenated and represented as $f = [f_{ij}^{MAS}, f_{ij}^{MTS}]$. A single neuron follows the concatenated feature sets for modeling the normal video sequences. Let, the predicted output of the MuST-3AN network is represented by a set $P = \{p_1, p_2, \ldots., p_N\}$ and let the ground truth labels are represented as $G = \{g, g_2, \ldots., g_N\}$. We have only the normal crowd scenes for training, so we got only one class for training the deep multiscale network. Due to such a reason, we have used linear activation at the output layer. Let $\emptyset_{\text{MuST}-3\text{AN}}$ represent the learnable parameters of the MuST-3AN. We obtain the loss between the P and the G using the mean squared error. Let the following Equation 5.1 defines the mean squared error loss between the P and the G.

$$loss_{\text{MuST}-3\text{AN}} = \frac{1}{N}\sum_{k=1}^{N}(p_k - g_k)^2 \tag{5.1}$$

The minimization function of the proposed model can be represented using Equation 5.2.

$$\underset{\emptyset_{\text{MuST}-3\text{AN}}}{argmin}\ [loss_{\text{MuST}-3\text{AN}}] \tag{5.2}$$

To solve the above optimization problem, backpropagation with Adam optimizer [170] has been used. The model is trained with a learning rate of 0.001. We have adopted early stopping to halt the network and also to avoid overfitting.

5.2.1.4 Dimension Reduction

After training the $\mathrm{MuST}-3\mathrm{AN}$ on the normal crowd videos, we have extracted the fused multi-scale features (f) for panic detection for video frames. Let the multi-scale feature matrix for the video sequence is denoted as $\mathrm{F}^{\mathrm{N}\times\mathrm{L}}$, where N is the total number of sequences and the L represents the dimension of the multi-scale feature of each frame. The concatenated multi-scale spatial-temporal features (f) are one-dimensional vectors. The same feature vector can be given to the OC-SVM for crowd panic detection, but the overheads of computational complexity for OC-SVM could be increased. Hence, by considering this fact, we have used PCA [86] to reduce the dimension of the multi-scale spatial-temporal features for the CPD.

The PCA can be defined as an orthogonal transformation of a set of features of one coordinate system to features to another coordinate system in such a way that the descending order of variances (obtained by performing some scalar projection on the feature set) lie on the ascending order (1^{st}, 2^{nd} and so on) of coordinates. The principal decomposition of $F^{N\times L}$ can be formulated as mentioned in Equation 5.3 [86],

$$T = F \times W \tag{5.3}$$

Where F is the $N \times L$ multiscale feature matrix, and W is the $q \times q$ square matrix. The columns of W are the eigenvectors of $F^T F$. The W is also known as sphering transformation. Now, by admiring first l principal components (obtained from first l eigenvectors), we can have the truncated transformation of Equation 5.4 as,

$$T_l = F \times W_l \tag{5.4}$$

Here the dimension of T_l is $N \times l$. Empirically, we set the dimension of the PCA to 128.

5.2.1.5 Crowd Panic Detection using OC-SVM

The dimensionally reduced feature set is given in to the OC-SVM [183] for modelling the normal crowd scenes. To train the OC-SVM, we have got only feature maps of one class. Generally, the OC-SVM tries to maximally separate the distance from the hyperplane (dimensionally reduced feature space) to its origin. In this way, a binary function is learned, which captures a region that surrounds the input data of one normal crowd scene, and it returns one if the data points lie within this region otherwise it returns -1, also known as outliers or panic. To maximize the distance between the hyperplane of the reduced feature space and its origin, we must optimize the basic problem's quadratic programming, which is defined as in Equation 5.5 [183].

$$\min_{\omega,\rho} 0.5\|\omega\|_2 + \frac{1}{vN}\sum_{i=1}^{N}\xi_i - \rho \ \ such\ that\ \{\omega.\varphi(k_i) \geq \rho - \xi_i\ and\ \xi_i \geq 0 \quad (5.5)$$

Here k_i is the training data corresponding to i^{th} sequence of N normal sequences. The mapping function $\varphi()$ maps the primitive feature space to a higher dimension. The ω is normal to the hyperplane of the feature space. $v \in [0,1]$ is the upper bound of the outlier (panic sequences), and ξ is the relaxation variable.

5.2.2 Experimental Setup

The programming is written in python using Keras and TensorFlow. The model had been executed on an intel-i7 8th Generation Laptop with 16 GB RAM, 4 GB GPU, and on Google Colab. The batch size for all the datasets was set to 8. This paper adopted three approaches to avoid overfitting: first, the L_2 norm is used to regularize all the kernel weights; second, the early stopping is used to halt the network and third, data augmentation is done during training. The learning rate is set to 0.001. The regularized parameter (L_2) coefficient is set to 0.01. The momentum of batch normalization. (BN) and alpha of Leaky ReLU are set to 0.95 and 0.1, respectively. During training, data

augmentation has been used to avoid overfitting on the small datasets. During data augmentation, different patches of scale [200×200×3] are extracted from the original frames, and the volume of frames containing crowd scenes only. One-third of the extracted patches are rotated with 25^0, $30^0$and 45^0 randomly. The size of data augmentation contains 70% of the original training samples.

5.2.3 Result Analysis and Discussion

5.2.3.1 The UMN dataset

The comparisons of the performance of the proposed MuST-POS with other state-of-the-art panic detection approaches are illustrated in Table 5.2. All the eleven sequences of UMN contain around 7739 frames, including both normal and panic crowd behaviour. The MuST-POS achieves average detection accuracy of 99.40%, which is the same as DeepROD [11]. Singh *et al.* [148] achieved an average accuracy (Acc) of 99.20% on the UMN dataset which is the second best results as far as Table 5.2 is concerned.

Table 5.2: Comparison of results with state-of-the-art methods on the UMN dataset

Sequences	[161]		[160]		[11]		[164]		**Proposed Model**	
	ER	***Acc***	**ER**	***Acc***	**ER**	***Acc***	**ER**	***Acc***	**ER**	***Acc***
S1	1.40	98.60	1.00	99.00	0.90	99.00	1.44	98.60	0.48	99.52
S2	1.00	99.00	1.00	99.00	0.80	99.10	0.48	99.50	0.49	99.51
S3	2.00	98.00	3.00	97.00	0.00	100.00	2.91	97.00	0.55	99.45
S4	1.00	99.90	2.00	98.00	0.10	99.80	2.77	97.00	0.59	99.41
S5	2.00	99.70	1.00	99.00	0.10	99.90	2.60	97.00	0.66	99.34
S6	0.50	99.50	1.00	99.00	0.50	99.50	2.76	97.00	0.52	99.48
S7	1.20	98.80	1.00	99.00	0.30	99.60	2.23	97.80	0.45	99.55
S8	2.30	97.60	2.00	98.00	1.30	98.60	2.55	97.50	0.45	99.55
S9	0.30	99.70	1.00	98.00	0.10	99.80	0.00	100.00	0.46	99.54
S10	0.70	99.30	1.00	99.00	1.00	99.00	1.03	98.90	0.89	99.11
S11	0.50	99.50	1.00	99.00	0.30	99.60	0.62	99.40	1.00	99.00
Average	1.00	99.00	1.50	98.50	0.60	99.40	1.84	98.15	0.60	99.40

Figure 5.3: Examples of output of the proposed model on the UMN dataset. Figures (a), (b), (c) are normal sequences, and the model predicted as normal, figures(d), (e), (f) belong to starting of panic behavior, and the model predicted as panic and figures (g), (h).

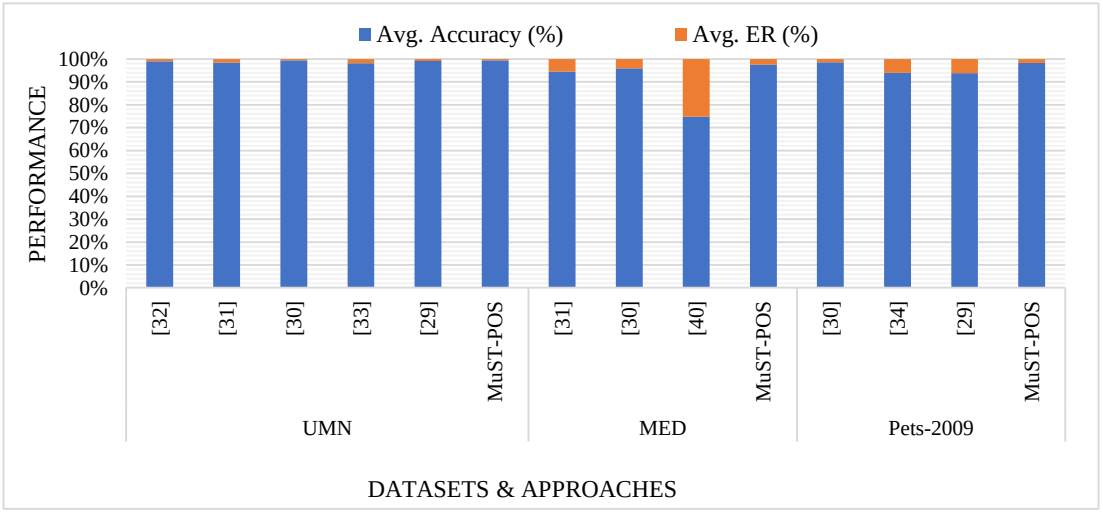

Figure 5.4: Comparison of average accuracy and average error rate between several approaches on three datasets.

So, by observing the performance of the proposed model on the UMN dataset, we can conclude that the model efficiently deals with the scale variation issue by extracting multiscale spatial-temporal features from the crowd scenes improves the performance of the model.

5.2.3.2 The MED Dataset

The performance analysis on the MED dataset is illustrated in Table 5.3 where the performance of the proposed model is compared with DeepROD [11]. Approaches like [2] and [160] also shown their performance on the MED dataset but these approaches only provided average accuracy on the MED dataset, so these data are not mentioned in Table 5.3.

Table 5.3: Comparison of results with state-of-the-arts on the MED dataset.

Sequences	Ammar *et al* [11]		**Proposed Model**	
	ER	***Acc***	**ER**	***Acc***
S1	4.00	95.50	1.54	98.46
S2	5.00	94.30	3.45	96.55
S3	2.00	97.70	2.28	97.72
S4	4.00	95.80	4.33	95.67
S5	6.00	93.40	1.36	98.64
S6	2.00	97.40	1.60	98.40
S7	0.40	99.50	5.33	94.67
S8	6.00	94.00	3.22	96.88
S9	1.70	98.30	2.49	97.51
S10	5.50	94.50	1.75	98.25
S11	9.00	91.00	1.99	98.01
Average	4.00	95.60	2.39	97.61

Models like [9, 150] and [1] obtain average detection accuracies of 95.60, 94.50, and 74.82, respectively. In comparison, the proposed model obtains a detection accuracy of 97.61% with a 2.39% error rate (ER). The proposed model tops the list as far as performance analysis is concerned.

The following Figure 5.5 shows results obtained on some of the samples of the MED dataset. Figure 5.4 illustrates a bar graph comparing the average accuracy and ER of several approaches on the MED dataset. The performance of the proposed model on the MED dataset shows the efficient utilization of multiscale spatial-temporal feature modeling for crowd panic detection. Thus, the model can handle the scale issue due to perspective distortion in the crowd panic video datasets. Figure 5.6 shows examples of results on few frames of the Pets-2009 dataset. Figure 5.4 illustrates a bar graph comparing the average accuracy and average ER of several approaches on the Pets-2009 dataset.

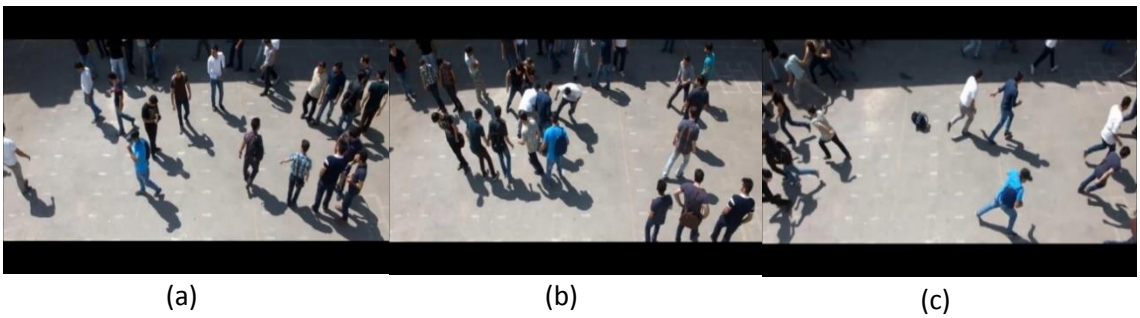

Figure 5.5: Examples of output of the proposed model on the MED dataset. Figure (a) is the normal sequence, and the model is predicting as normal, figure (b) shows to starting of panic behaviour, and the model is predicting as panic, and figure (c) shows the panic situations.

5.2.3.3 The Pets-2009 Dataset

The comparison of the performance of the proposed model with other state-of-the-art on the Pets-2009 dataset is illustrated in Table 5.4. The proposed model achieves 98.37% of accurate detection of crowd panic behavior. The error rate of the MuST-POS on Pets-2009 dataset is 1.63%. We followed the same training and testing criteria for the Pets-2009 dataset as mentioned in [9, 138, 157]. In contrast DeepROD [11] achieved an

average accuracy of 97.50 % with ER average ER of 1.40. Methods like [165] and [148] also evaluated their performance on the Pets-2009 dataset but only provided the average accuracy of 94.00 % and 93.80 % respectively. The proposed model achieves the highest accuracy and lowest false alarm rate as compared with recent approaches of crowd panic detection.

Table 5.4: Comparison of results with state-of-the-art methods on the Pets-2009 dataset.

Sequences	Ammar *et al.*[11]		**Proposed Model**	
	ER	***Acc***	***ER***	***Acc***
Time14–16	2.70	97.30	1.77	98.23
Time14–17	0.20	97.80	1.49	98.51
Average	1.40	97.50	1.63	98.37

(a) (b) (c)

Figure 5.6: Examples of output of the proposed model on the Pets-2009 dataset. Figure (a) is the normal sequence, and the model is predicting as normal, figure (b) shows to starting of panic behavior, and the model is predicting as panic, and figure (c) shows the panic frame and the model is predicting as panic.

5.2.3.4 Ablation Study

The ablation study shows the effects of each stream of the proposed model. The streams like MAS and MTS are the two critical modules designed to extract multi-scale spatial and multi-scale temporal features from the crowd panic videos. It is essential to show the influence of each module on panic detection. We split the network into two

modules: the multi-scale spatial 3D atrous net and PCA guided OC-SVM (MuS-POS), and the multi-scale temporal 3D atrous net and PCA guided OC-SVM (MuT-POS).

In addition, we have also experimented with single-scale analysis using single-scale PCA-guided OC-SVM (Single-Scale POS), where all the multi-scale connections are omitted. Similarly, to show the impact of PCA on CPD, we have examined the performance of MuST-POS without PCA on different datasets. Table 5.5 shows the results analysis of different cases of ablation study. The two streams perform differently on the UMN dataset, in which the average accuracies of MuS-POS and MuT-POS are 95.54% and 96.78%, respectively. In this case, the spatial stream performs better than the temporal stream. The performance is improved by combining these two streams (MuST-POS). The UMN dataset is challenging and contains different crowd panic situations in different environments.

Table 5.5: Comparison of results of different modules during ablation study

Datasets		**MuS-POS**		**MuT-POS**		**Single-Scale POS**		**MuST-POS (Without PCA)**		**MuST-POS**	
		Acc	$F1_{Score}$	Acc	$F1_{Score}$	Acc	$F1_{Score}$	Acc	$F1_{Score}$	Acc	$F1_{Score}$
The UMN	**S1**	91.36	94.45	95.36	97.00	95.04	96.79	98.24	98.86	99.52	99.69
	S2	95.28	97.06	97.22	98.27	96.49	97.83	98.43	99.02	99.51	99.70
	S3	96.53	96.97	97.08	97.46	96.72	97.14	98.36	98.57	99.45	99.52
	S4	97.22	98.33	96.78	98.08	97.07	98.24	98.53	99.13	99.41	99.65
	S5	97.00	97.77	97.26	97.96	95.83	96.89	98.56	98.92	99.34	99.51
	S6	94.99	96.80	96.20	97.57	94.12	96.24	98.96	99.34	99.48	99.67
	S7	97.20	98.31	97.87	98.72	95.64	97.36	98.88	99.32	99.55	99.73
	S8	94.45	96.05	95.35	96.70	91.45	93.91	98.95	99.92	99.55	99.68
	S9	95.74	97.47	96.80	98.10	93.46	96.10	99.24	99.54	99.54	99.72
	S10	94.67	96.88	96.30	97.82	91.71	95.15	97.92	98.79	99.11	99.48
	S11	96.53	98.06	98.39	99.09	95.29	97.35	98.39	99.10	99.00	99.44
	Average	95.54	97.10	96.78	97.88	94.80	96.63	98.58	99.13	99.40	99.61
The MED	**S1**	94.10	96.69	95.73	97.62	93.76	96.50	97.91	98.84	98.46	99.15
	S2	93.10	96.00	91.03	94.77	89.79	94.02	95.44	97.37	96.55	98.01

	S3	90.90	94.54	92.42	95.37	93.03	95.78	96.67	97.98	97.72	98.63
	S4	91.15	94.67	91.34	94.73	92.69	95.52	94.90	96.87	95.67	97.35
	S5	92.70	95.70	93.75	96.31	92.18	95.38	97.60	98.57	98.64	99.19
	S6	92.35	94.59	92.88	94.91	91.30	93.79	97.55	98.22	98.40	98.84
	S7	89.11	91.41	89.94	91.94	78.10	82.29	93.72	94.91	94.67	95.69
	S8	90.75	93.99	94.09	96.16	84.07	89.73	95.76	97.25	96.88	97.98
	S9	92.03	95.37	93.53	96.23	89.55	93.84	96.91	98.20	97.51	98.55
	S10	96.77	98.13	96.10	97.74	90.72	94.56	98.52	99.14	98.25	98.99
	S11	96.03	97.75	95.13	97.25	87.63	92.83	97.53	98.60	98.01	98.87
	Average	92.63	95.34	93.26	95.73	89.34	93.11	96.59	97.81	97.61	98.29
The Pets-2009	**Time14–16**	92.92	93.22	94.69	94.87	89.82	90.12	96.90	96.99	98.23	98.30
	Time14–17	91.83	94.78	94.81	96.74	88.14	92.38	97.03	98.14	98.51	99.07
	Average	92.37	94.00	94.75	95.80	88.98	91.25	96.96	97.56	98.37	98.68

On the other hand, the MED and Pets-2009 datasets have multiple crowd panic situations captured from a single environment. The MuS-POS and MuT-POS obtain average detection accuracies of 92.63% and 93.26%, respectively, on the MED dataset. However, in the Pets-2009 dataset, MuS-POS and MuT-POS achieve average detection accuracies of 92.37% and 94.75%, respectively.

During single-scale analysis the Single-Scale POS obtains average accuracies and average F1-Score of <94.80%, 96.63%>, <89.34%, 93.11%> and <88.98%, 91.25%> on UMN, MED, and Pets-2009 datasets respectively. Figure 5.7 shows some panic samples contain crowd scale-variation, which are not detected by Single-Scale POS but are accurately detected by MuST-POS. On comparing the results of Single-Scale POS with Multi-Scale POS, it can be concluded that the multi-scale analysis is desirable to achieve better accuracy and address scale change due to perspective distortion.

(a) UMN-S1-Frame#560 (b)UMN-S7-Frame#4817 (c)Pets2009-S-17-Frame#82 (d)MED-S1-Frame#1562

Figure 5.7: Samples of panic situations which are detected as Normal by Single-Scale POS but are detected as Panic by the proposed MuST-POS

On the other hand, the MuST-OS (without PCA) obtains average accuracy and average F1-Scores of <98.58%, 99.13%>, <96.59%, 97.81%> and <96.96%, 97.56%> on the UMN, the MED, and the Pets2009 datasets respectively. The performance of MuST-OS (without PCA) is lower than the MuST-POS.

5.3 TS-MDA: Two-Stream Multiscale Deep Architecture for Crowd Behaviour Prediction

5.3.1 Proposed Method and Model

A crowd behavior understanding model should handle the two most challenging situations in the crowd scene: human shape variation due to perspective distortion and the effect of cluttered background. The state-of-the-art CBP models [1, 136, 137] deviate in handling such challenges. Nevertheless, from the literature on related research domains like crowd counting, Sang *et al.* [184] 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 [185], i.e., the visual background eliminator (ViBE). Hence, by adopting the idea of SA-CNN [184] and utilizing the ViBE algorithm [185], a two-stream multiscale deep architecture (TS-MDA) is proposed for the MCC-based CBP.

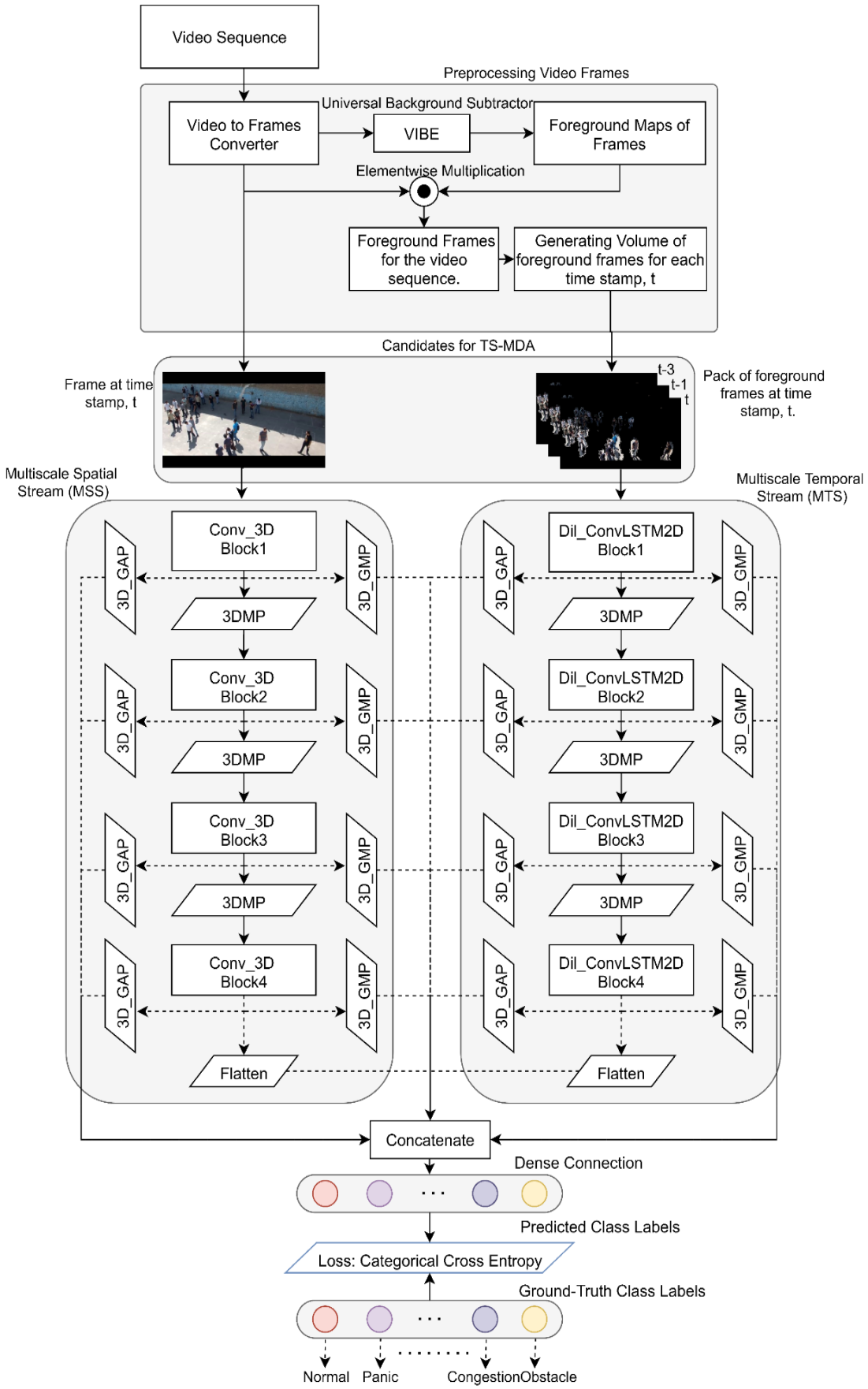

Figure 5.8: Architecture of proposed TS-MDA.

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 Figure 5.8.

The proposed model constitutes of the following sub-modules,

- Pre-processing.
- Candidates for TS-MDA.
- Architecture Details of the TS-MDA
- Multiscale Feature Extraction.
- Crowd Behavior Prediction.
- Loss Function and Optimization.

5.3.1.1 Pre-processing

At first, the video sequence is pre-processed before training the TS-MDA. The pre-processing stage is followed by the generation of the candidate for training the TS-MDA. During pre-processing, 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 $RF = \{rf_1, rf_2, \ldots\ldots, rf_N\}$. After video to frame conversion, the foreground frames are extracted, as explained in the following subsection.

5.3.1.1.1 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) [185] is a popular algorithm to model fame'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 [185] 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 $M^{t-1} = \{v_1^{t-1}, v_2^{t-1}, \dots\dots, v_Z^{t-1}\}$. 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 in Equation 5.6.

$$\#\{S_R(v^t(x)) \cap M^{t-1}\} \tag{5.6}$$

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

$$M^0(x) = \{v^0(y|y \in N_G(x))\} \tag{5.7}$$

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 $FM = \{fm_1, fm_2, \dots\dots, fm_N\}$. 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 Equation 5.7.

$$ff_i = rf_i \odot fm_i, where\ i = 1,2, \dots, N \tag{5.8}$$

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 timestamp t, $t-1$, $t-2$. Let the set $VF = \{vf_1, vf_2, \ldots . vf_N\}$ represent the volume of foreground frames for the dataset. Each of $vf_t = Concatenate([ff_t, ff_{t-1}, ff_{t-2}])\ for\ each\ t = 1\ to\ N$. Here, $Concatenate()$ is the concatenation operation.

5.3.1.2 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.

5.3.1.3 Architecture Details

The overall architecture of the proposed model is illustrated in the following Figure 5.8. 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 5.6. The features maps are downscaled to its half after Conv_3D Block1, Conv_3D Block2, and Conv_3D Block3 by 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.

Table 5.6: 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)	NA
	ReLU	NA		
	BN			
Conv_3D Block2	Conv_3D	64	(4,4,3)	NA
	ReLU	NA		
	BN			
Conv_3D Block3	Conv_3D	128	(3,3,3)	NA
	ReLU	NA		
	BN			
Conv_3D Block4	Conv_3D	256	(3,3,3)	NA
	ReLU	NA		
	BN			
Dil_ConvLSTM2D Block1	ConvLSTM2D	25	(3,3)	(2,2)
	Tanh	NA		
	BN			
Dil_ConvLSTM2D Block2	ConvLSTM2D	40	(3,3)	(2,2)
	Tanh	NA		
	BN			
Dil_ConvLSTM2D Block3	ConvLSTM2D	60	(2,2)	(1,1)
	Tanh	NA		
	BN			
Dil_ConvLSTM2D Block4	ConvLSTM2D	80	(2,2)	(1,1)
	Tanh	NA		
	BN			

The details of these blocks are mentioned in Table 5.6. 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 by using 3DMP layers. The 3D_GMP and the 3D_GAP are used after every activation layer of dilated ConvLSTM2D layers. All the Conv_3D

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 behaviour. The activation of the output layer is SoftMax.

5.3.1.4 Multiscale Spatial-Temporal Feature Extraction and Prediction

The proposed model utilizes Conv_3D layers to extract spatial features from the RGB frames. Although Li *et al*. [186] 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 $[batch\_size \times 150 \times 150 \times 3 \times 1]$ 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 represents 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. 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. The multiscale spatial features ($F^{Spatial}$) and temporal features ($F^{Temporal}$) are concatenated by simply appending one after another. Let, $F^{Concate} = Concatenate\left(F^{Spatial}, F^{Temporal}\right)$ represents the concatenated multiscale features.

5.3.1.5 Crowd Behaviour 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,

$$Y_{CBP} = \cup_{p=1}^{K}[y_{p_{out}}] = \cup_{p=1}^{K} SoftMax(y_{p_{in}}) = \cup_{p=1}^{K}\left[\frac{e^{y_{p_{in}}}}{\sum_{p=1}^{K} e^{y_{p_{in}}}}\right] \quad (5.8)$$

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

5.3.1.6 Loss Function and Optimization

Let $\emptyset_{TS-MDA}$, represents all the trainable parameters of the proposed model. Let $T_{i_{CBP}} = \{T_1, T_2, T_3, .., T_K\}^i$ be the ground truth labels of the i^{th} crowd scene. The loss on the i^{th} crowd scene is obtained by 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.

$$L_i(\emptyset_{TS-MDA}) = L_i(T_{i_{CBP}}, Y_{i_{CBP}}) = \left[-\sum_{p=1}^{K} T_p \, log \, y_{p_{out}}\right]^i \quad (5.9)$$

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 [170] method to minimize the loss function. The mini-batch based optimization problem can be represented as,

$$\underset{\emptyset_{TS-MDA}}{argmin}[L(\emptyset_{TS-MDA})]^{b} \tag{5.10}$$

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.

$$[L(\emptyset_{TS-MDA})]^{b} = \frac{1}{Batch\_Size}\sum_{i=1}^{Batch\_Size} L_i\left(T_{i_{CBP}}, Y_{i_{CBP}}\right) \tag{5.11}$$

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,

$$[\nabla L(\emptyset_{TS-MDA})]^{b} = \left[\nabla_{\emptyset_{TS-MDA}} L(\emptyset_{TS-MDA})\right]^{b} \tag{5.12}$$

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) [170] update rule. The Adam [170] optimizer utilizes the cumulative history of gradients to update the $\emptyset_{TS-MDA}$ to solve the decay problem. For a given iteration itr the cumulative history of gradients for a given batch b can be calculated by using the following equations (5.13-5.16).

$$m^{b} = \beta_1 \times m^{b-1} + (1-\beta_1) \times [\nabla \emptyset_{TS-MDA}]^{b} \tag{5.13}$$

$$v^{b} = \beta_2 \times v^{b-1} + (1-\beta_2) \times ([\nabla \emptyset_{TS-MDA}]^{b})^2 \tag{5.14}$$

$$\widehat{m^{b}} = \frac{m^{b}}{1-{\beta_1}^{itr}}\ and\ \widehat{v^{b}} = \frac{v^{b}}{1-{\beta_2}^{itr}} \tag{5.15}$$

, where β_1=0.9 and β_2=0.999. Now, the parameters are updated by using the following Equation 5.16 [170].

$$\emptyset_{TS-MDA} = \emptyset_{TS-MDA} - \frac{\eta}{\sqrt{\widehat{v^{b}}+\epsilon}} \times \widehat{m^{b}} \tag{5.16}$$

Here, η is the learning rate. According to Adam optimizer [170], 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-5.1 Optimizing the TS-MDA

Input: Resized frameset, $RF = \{rf_1, rf_2, \ldots\ldots, rf_N\}$, foreground image set, $FF = \{ff_1, ff_2, \ldots. ff_N\}$ 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, $\emptyset_{TS-MDA}$, $momentum$, β_1, and β_2

Initialisation: $Max\_Iteration = 2000$, $Batch\_Size$ (Different for different dataset), $itr = 1, \beta_1 = 0.9$, β_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 $F_i^{Concate}$.
2. Find the predicted crowd behaviour label $Y_{i_{CBP}}$ using Equation 5.8.
3. Find the Loss i.e., $L_i(\emptyset_{TS-MDA})$ using Equation 5.9.

end for

4. Find the mean of cumulative of loss for the given batch b using Equation 5.11.
5. Find the gradients of the loss using Equation 5.12.
6. Obtain Cumulative History of gradients using Equations 5.13 to 5.15
7. Update network parameter $\emptyset_{TS-MDA}$ using Equation-12.

End For

6. $itr += 1$

End While

5.3.2 Experimental Setup

The program is written in Python by using TensorFlow and Keras. The batch size and 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. The programs have been executed on different computing nodes of the Param Shivay supercomputer.

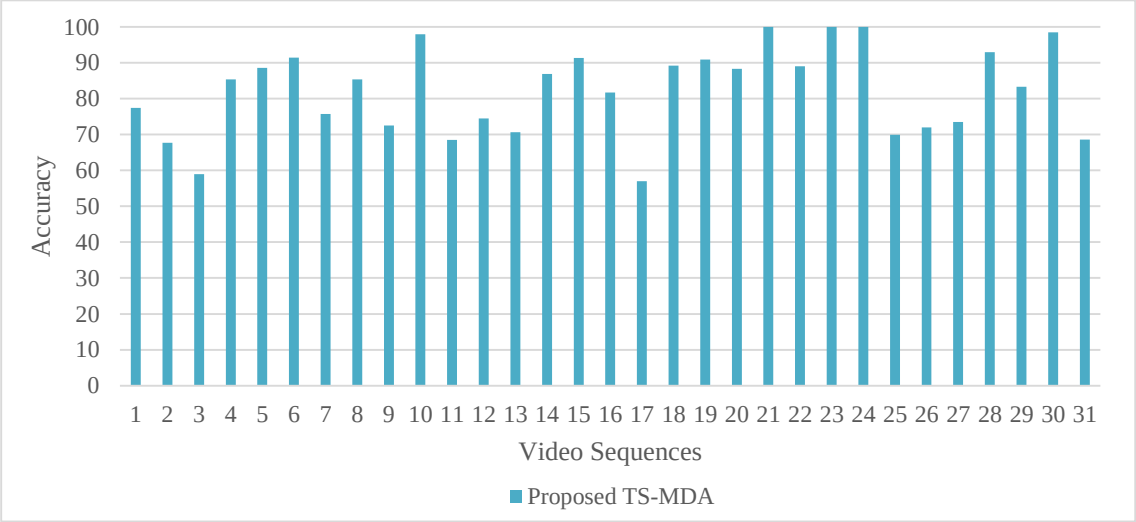

Figure 5.9: Accuracies obtained by the proposed model using leave-one-sequence-out on the MED dataset.

5.3.3 Results Analysis and Discussion

5.3.3.1 The MED dataset

The procedure for training and testing is followed as prescribed in [2], 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 5.9 shows a graphical representation of the experimental results obtained on the MED dataset during

leave-one-sequence-out. As shown in Figure 5.9, 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 5.10 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 5.7. The values in bold letters represent best results in Table 5.7. Deep models [147] 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 5.7 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.

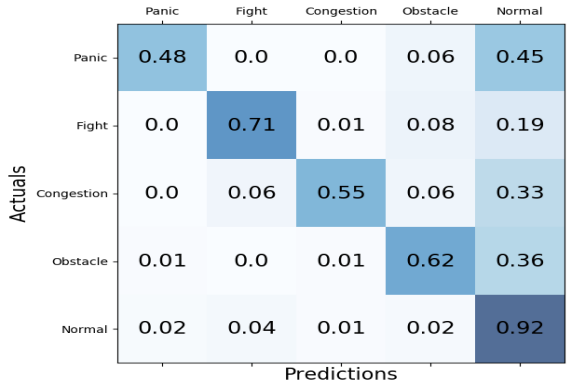

Figure 5.10: Confusion matrix of the proposed model on the MED dataset [2]

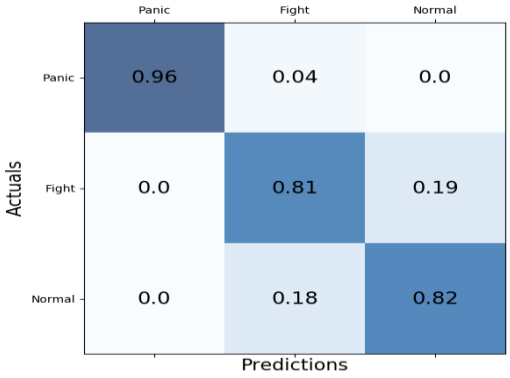

Figure 5.11: Confusion matrix of the proposed model on the GTA dataset [146]

Table 5.7: Performance comparison with other state-of-the-art approaches on the MED dataset

Approaches	**Classification accuracy (%) per individual behavior classes**					***Mean-ACC (%)***	***Accuracy (%)***
	Panic	**Fight**	**Congestion**	**Obstacle**	**Normal**		
V3G-FC7 [147]	80.72	37.41	31.18	47.25	71.35	53.58	*62.71*
V3G-FC8 [147]	53.23	29.89	27.32	42.35	32.16	36.99	*33.82*
C3D-FC7 [147]	**84.72**	32.93	16.16	29.61	**92.69**	51.22	*73.52*
C3D-FC8 [147]	57.32	25.89	17.22	25.51	46.64	34.50	*40.59*
HOT [2]	62.18	38.27	25.67	28.20	36.53	38.17	36.29
DT [2]	74.82	30.47	23.43	27.94	36.88	38.71	36.10
Proposed	48.20	**71.32**	**54.50**	**61.70**	91.51	**65.45**	**81.26**

5.3.3.2 The GTA dataset

The experiment on the GTA dataset [146] is demonstrated by following the same procedure as mentioned in [146]. The behavior sequences 2, 4, 11, and 12 are the test sequences that were selected randomly. The confusion matrix of the proposed model on the GTA dataset is illustrated in Figure 5.11. 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 5.8. The spatial-temporal model [146] is the only model which experimented on the GTA dataset [146].

The spatial-temporal model [146] 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 [146] 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 [146].

Table 5.8: Performance comparison with state-of-the-art approach on the GTA dataset. Values in bold letters represent best in the table.

Approaches	**Classification accuracy (%) per individual behavior classes**			***Mean-ACC (%)***	***Accuracy (%)***
	Normal	**Panic**	**Fight**		
Spatial-Temporal Net[146]	**83.80**	61.20	28.90	71.70	-
Proposed TS-MDA	81.88	**95.60**	**80.75**	**86.07**	**88.61**

5.3.3.3 Ablation Study

An ablation study on the proposed model has been performed to show the effectiveness of each of its main 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.
- 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 5.12: (a), (b), (c), (d) and (e) show the confusion matrix of the MSS, MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA modules, respectively, on the MED dataset [2]. Similarly, Figure 5.14: (a), (b), (c), (d) and (e) illustrate the confusion matrix of the MSS MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA on the GTA dataset [146]. Figure 5.13 shows a graphical comparison of different models used in the ablation study for the MED sequences [2]. According to Figure 5.13, 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 performs 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 5.9 and Table 5.10 on MED and GTA datasets respectively. Performance metrics like per-class accuracy, overall accuracy, and per-class precision are obtained for different modules.

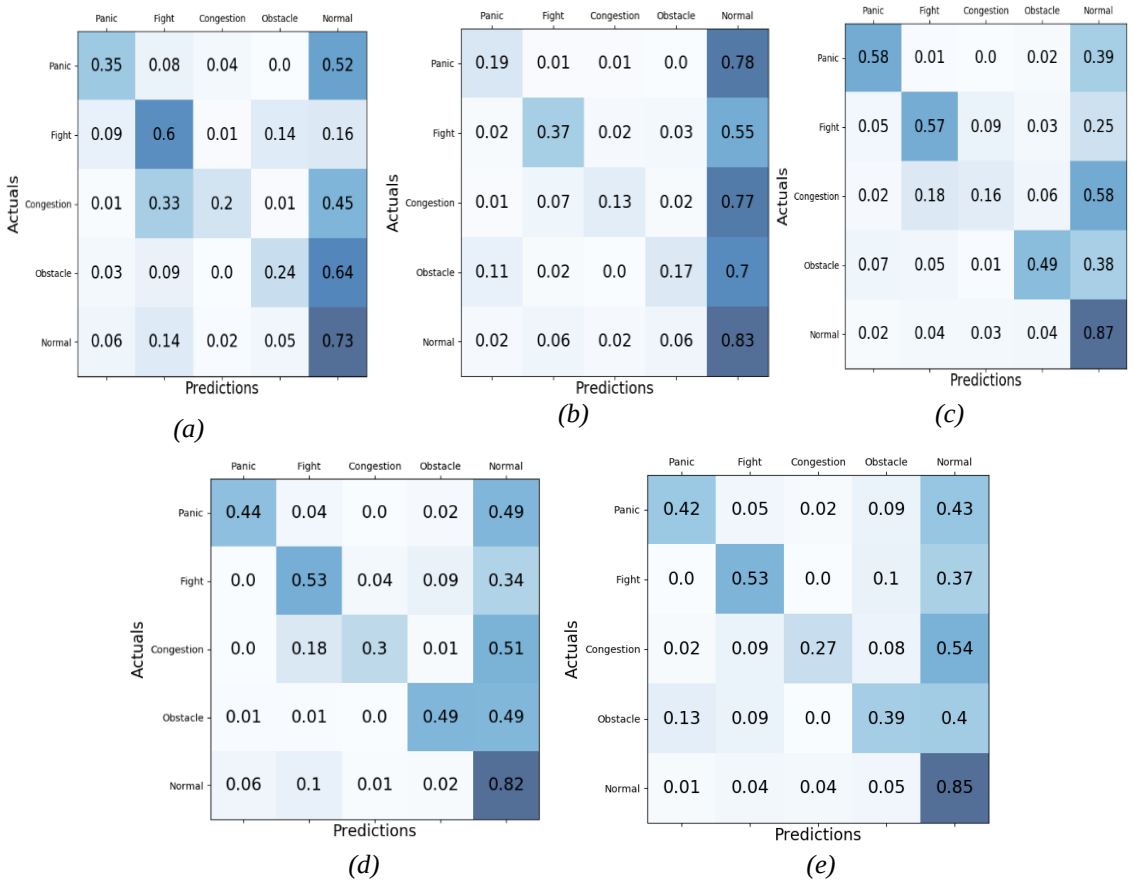

Figure 5.12: Confusion metrics of different modules during ablation study on the MED dataset [2]. *The subfigures (a), (b), (c), (d), and (e) are the confusion metrics of MSS, MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA modules, respectively.*

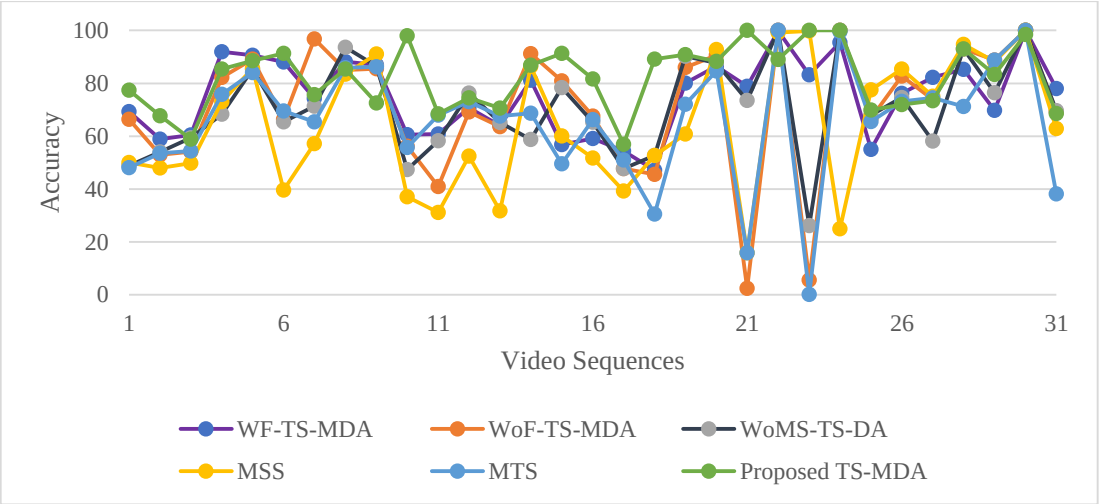

Figure 5.13: Comparison of accuracies of different models during ablation study using leave-one-sequence-out cross-validation on the MED dataset.

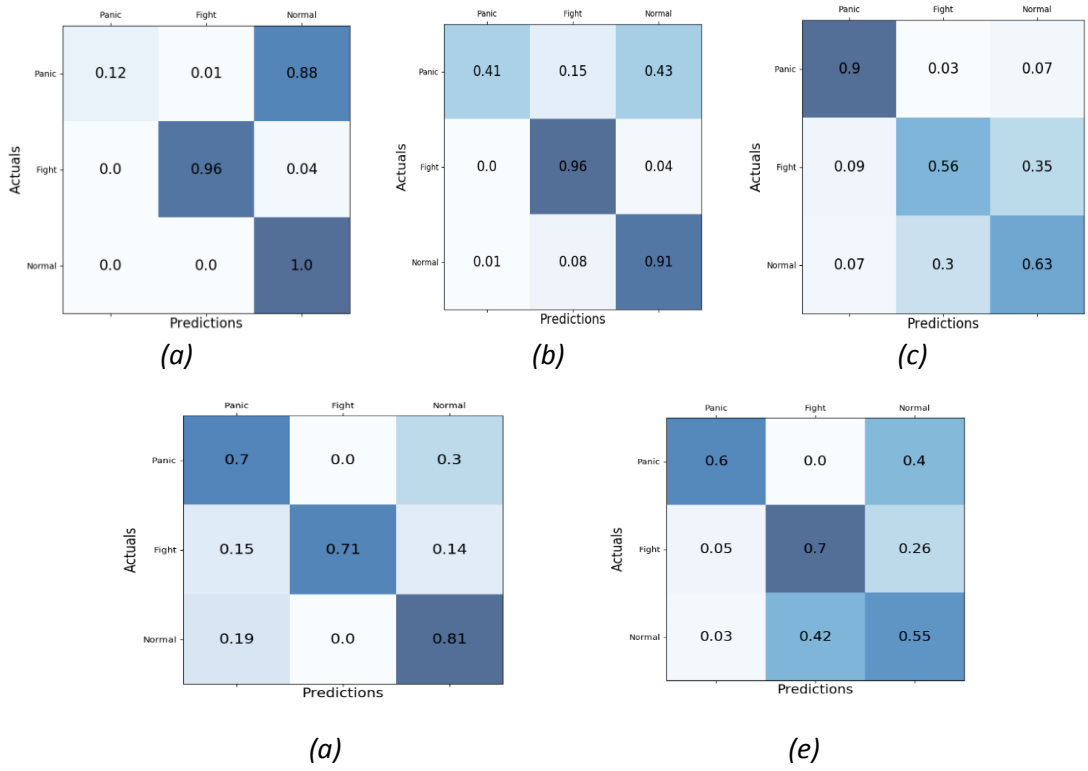

Figure 5.14: Confusion metrics of different modules during ablation study on the GTA dataset [146]. The subfigures (a), (b), (c), (d), and (e) are the confusion metrics of MSS, MTS, WF-TS-MDA, WoF-TS-MDA, and WoMS-TS-MDA modules, respectively.

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 [2] and 75.33%, 73.64%, 60.16%, 54.51%, 69.89% on GTA dataset [146] respectively.

However, the proposed TS-MDA achieves better performance as compared with individual modules.

Table 5.9: Comparative analysis of results of different modules of the proposed TS-MDA during ablation study on the MED dataset [2]. *Values in bold letters represent best in the table.*

Performance Metrics		**Modules used in Ablation Study**					
		WF-TS-MDA	**WoF-TS-MDA**	**WoMS-TS-DA**	**The MSS Module**	**The MTS Module**	**The Proposed Model**
Per Class Accuracy (in %)	**Panic**	57.94	44.45	41.50	35.41	19.38	48.20
	Fight	57.34	53.01	52.60	60.30	37.42	71.28
	Congestion	7.24	30.00	27.40	20.36	13.40	54.50
	Obstacle	48.54	49.31	38.60	23.73	17.43	61.70
	Normal	87.17	82.08	85.43	73.27	83.43	91.51
Per-Class Precision (in %)	**Panic**	**53.60**	37.31	43.30	25.16	25.95	66.73
	Fight	**59.36**	43.15	56.28	35.35	47.38	75.26
	Congestion	**22.10**	69.11	39.59	47.10	31.37	78.28
	Obstacle	**64.04**	69.83	47.24	35.56	31.36	68.46
	Normal	**81.16**	76.71	79.20	76.98	70.43	84.96
Over all Accuracy		**73.17**	69.00	70.11	60.99	63.03	81.26

Table 5.10: Comparative analysis of results of different modules of the proposed TS-MDA during ablation study on the GTA dataset [146]. Values in bold letters represent best in the table

Performance Metrics		**Modules used in Ablation Study**					
		WF-TS-MDA	**WoF-TS-MDA**	**WoMS-TS-DA**	**The MSS Module**	**The MTS Module**	**The Proposed Model**
Per Class Accuracy (in %)	**Panic**	90.13	70.47	59.65	11.73	41.58	95.59
	Fight	55.98	70.77	69.78	96.17	95.89	80.75
	Normal	62.80	80.87	54.80	98.12	91.12	81.88
Per-Class Precision (in %)	**Panic**	**92.14**	80.50	94.51	100.00	98.52	100.00
	Fight	**51.23**	100.00	51.60	98.12	64.61	67.59
	Normal	**64.28**	57.78	39.25	39.89	54.68	86.30
Over all Accuracy		**75.33**	73.64	60.16	54.51	66.89	88.61

From the confusion matrixes i.e., Figure 5.12: (a), (b) and Figure 5.14: (a), (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 [146]; this may be due to several reasons. First, the MED dataset [2] is more realistic than the GTA dataset [146] 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 [2] and GTA datasets [146]. 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 [2] is concerned. Nevertheless, the proposed model achieves better performance as far as state-of-the-art approaches are concerned.

5.4 Conclusion

In this chapter, two models for CBA were proposed. The first model (MuST-POS) adopted the OCC approach to predict normal and panic crowd behaviors, whereas the

second model (TS-MDA) performed multiclass crowd behavior classification. Both models handled human shape variations due to perspective distortion in the crowd video by exploiting scale-invariant features. In addition, the TS-MDA minimized the effect of background influence by utilizing deep features from the foreground image of the frames. To show the efficacy of the proposed models, extensive experiments, comparative results analysis, and ablation studies were performed on the publicly available benchmark datasets. The MuST-POS achieved 99.40%, 97.61%, and 98.37% of average detection accuracies on the UMN, the MED Panic, and Pets-2009 Panic datasets. The results of MuST-POS outperform recent state-of-the-art approaches. The TS-MDA achieved an accuracy of 81.26% and 88.61% on the MED and GTA datasets and performed far better than the recent state-of-the-art. The TS-MDA also performed better when compared to OCC-based approaches.

In this chapter, two deep models were proposed for the task CBA. The experimented results outperform the state-of-the-art. In the next chapter, an important task of CA, i.e., multitasking CA using a deep learning approach, will be discussed.