Computers, Materials & Continua
DOI: 10.32604/cmc.2022.026666
Article

Tech Science Press

Smart Deep Learning Based Human Behaviour Classification for Video Surveillance

Esam A. AlQaralleh[1], Fahad Aldhaban[2], Halah Nasseif[2], Malek Z. Alksasbeh[3] and Bassam A. Y. Alqaralleh[2,*]

[1]School of Engineering, Princess Sumaya University for Technology, Amman, 11941, Jordan
[2]MIS Department, College of Business Administration, University of Business and Technology, Jeddah, 21448, Saudi Arabia
[3]CIS Department, Faculty of Information Technology, Al Hussein bin Talal University, Ma'an, 71111, Jordan
*Corresponding Author: Bassam A. Y. Alqaralleh. Email: b.alqaralleh@ubt.edu.sa
Received: 31 December 2021; Accepted: 11 February 2022

Abstract: Real-time video surveillance system is commonly employed to aid security professionals in preventing crimes. The use of deep learning (DL) technologies has transformed real-time video surveillance into smart video surveillance systems that automate human behavior classification. The recognition of events in the surveillance videos is considered a hot research topic in the field of computer science and it is gaining significant attention. Human action recognition (HAR) is treated as a crucial issue in several applications areas and smart video surveillance to improve the security level. The advancements of the DL models help to accomplish improved recognition performance. In this view, this paper presents a smart deep-based human behavior classification (SDL-HBC) model for real-time video surveillance. The proposed SDL-HBC model majorly aims to employ an adaptive median filtering (AMF) based pre-processing to reduce the noise content. Also, the capsule network (CapsNet) model is utilized for the extraction of feature vectors and the hyperparameter tuning of the CapsNet model takes place utilizing the Adam optimizer. Finally, the differential evolution (DE) with stacked autoencoder (SAE) model is applied for the classification of human activities in the intelligent video surveillance system. The performance validation of the SDL-HBC technique takes place using two benchmark datasets such as the KTH dataset. The experimental outcomes reported the enhanced recognition performance of the SDL-HBC technique over the recent state of art approaches with maximum accuracy of 0.9922.

Keywords: Human action recognition; video surveillance; intelligent systems; deep learning; security; image classification

1 Introduction

Human action recognition (HAR) and classification techniques have various applications that are helpful in day-to-day lives. Video surveillance is employed in smart supervision systems in banks, smart buildings, and parking lots [1]. Communication between machines and human is a major challenge, i.e., performed by many different methods namely hand gesture classification and speech recognition [2]. The process of video frames acquired from security camera with the help of recognizing and controlling abnormal behavior creates an automated care monitoring scheme as a human action detector [3]. Furthermore, the many elderly and sick people living alone and needing to be checked by constant surveillance triggers the need for an intelligent system that is beneficial and essential to monitor elder people. Various factors are essential in the efficacy of action detection systems like the background of the location, any abnormality condition, and detection time. The consequence of all the factors in the study of objects and the kind of behavior and actions identify the classification and recognition of the behavior [4]. Especially, in partial behavior, just the topmost part of the body is employed for recognizing hand gestures. Analysis of Human behavior from a captured video needs a preprocessing phase involving foreground and background recognition, also tracking individuals in successive frames.

Other important steps include feature extraction, appropriate model or classifier selection, and lastly the procedure of authentication, classification, and detection-based feature extraction. The initial phase for object behavior detection is recognizing the movement of the object in an image and its classification. The more commonly known method for the detection of moving objects is background subtraction [5]. The simplest method of background subtraction can be accomplished by comparing all the frames of the video with a static background. As stated, afterward the pre-processing phase, the automated recognition systems will include two major phases: feature extraction and action classification [6]. The most significant phases in the behavior analysis method are creating an appropriate feature vector and feature extraction. This process will create the primitive information for the classification.

Accurate recognition of action is one of the difficult processes to alter in clutter backgrounds and viewpoint variations. Hence, we can emphasize, that one of the most popular methods for HAR employs engineered motion [7] and texture descriptor evaluated about Spatio-temporal interest point. Additionally, many approaches follow the traditional method of pattern recognition [8]. This approach is depending on two major phases: learning classifier based on the attained feature and calculating difficult handcrafted features in the video frame. In real-time scenarios, it is uncommonly known that feature is significant to the task at hand because the selection of features is extremely problem-dependent [9].

This paper presents a smart deep learning-based human behavior classification (SDL-HBC) model for real-time video surveillance. The proposed SDL-HBC model majorly aims to employ an adaptive median filtering (AMF) based pre-processing to reduce the noise content. In addition, the capsule network (CapsNet) model utilized for the extraction of feature vectors and the hyperparameter tuning of the CapsNet technique takes place using the Adam optimizer. Finally, the differential evolution (DE) with stacked autoencoder (SAE) model is applied for the classification of human activities in the intelligent video surveillance system. The simulation result analysis of the SDL-HBC technique is carried out against two benchmark datasets namely KTH datasets.

2 Literature Review

Nikouei et al. [10] introduced a Single Shot Multi-Box Detector (SSD), lightweight Convolution Neural Networks (L-CNN), and depth-wise separable convolution. With narrowing down the classifier's search space for emphasizing human objects in surveillance video frames, the presented L-CNN method is capable of detecting pedestrians with reasonable computational workloads to an edge device. Nawaratne et al. [11] presented the incremental spatiotemporal learner (ISTL) for addressing limitations and challenges of anomaly localization and detection for real-time video surveillance. ISTL is an unsupervised DL method that employs active learning with fuzzy aggregation, to repetitively distinguish and update amongst new normality and anomalies which evolve.

Bouachir et al. [12] designed a vision-based methodology for automatically identifying suicide by hanging. These smart video surveillance systems operate by depth stream given by the RGB-D camera, nevertheless of illumination condition. The presented approach is depending on the exploitation of the body joint position for modeling suicidal behaviors. The static and dynamic pose features are estimated for effectively modeling suicidal behaviors and capturing the body joint movement. Wan et al. [13] developed a smartphone inertial accelerometer-based framework for HAR. The data are pre-processed by denoising, segmentation, and normalization for extracting valuable feature vectors. Furthermore, a real-time human activity classification-based CNN method has been presented that employed a CNN to local feature extraction.

Han et al. [14] presented an approach of data set remodeling by transporting parameters of ResNet-101 layers trained on the ImageNet data set for initializing learning models and adapting an augmented data variation method for overcoming the over-fitting problem of sample deficiency. To model structure improvements, a new deep 2-stream ConvNets was developed for action complexity learning. Ullah et al. [15] projected an improved and effective CNN-based method for processing data stream in real-time, attained from visual sensors of non-stationary surveillance environments. At first, the frame-level deep feature is extracted by a pre-trained CNN method. Then, an enhanced DAE is presented for learning temporal variations of the action from the surveillance stream.

3 The Proposed Model

In this study, a novel SDL-HBC technique has been derived for the recognition of human behavior in intelligent video surveillance systems. The proposed SDL-HBC technique aims to properly determine the occurrence of several activities in the surveillance videos. The SDL-HBC technique encompasses several stages of operations such as AMF based pre-processing, CapsNet based feature extraction, Adam optimizer-based hyperparameter tuning, SAE-based classification, and DE-based parameter tuning.

3.1 AMF Based Pre-Processing

Primarily, the AMF technique is used to pre-process the input image to eradicate the noise that exists in it [16]. The AMF technique makes use of the median value of the windows for replacing the intermediate pixels treated by the window. If the intermediate pixels are (Pepper) or (salt), it gets substituted using the intermittent value of the window. The AMF follows the replacement process with the median value of the window [17]. It generally operates in the following ways: The window gets arranged in ascending order. Then, the median value can be considered as the intermediate value next to the sorting process. Thus, the pixels can be substituted by the median value.

3.2 Feature Extraction Using Optimal CapsNet Model

At this stage, the preprocessed image is passed into the CapsNet model to derive the useful set of feature vectors. The CNN model can be utilized as an effective method for performing the 2D object recognition process. Because of the data routing process in the CNN model, the details, such as position and pose in the objects, are not considered. For resolving the issues of the CNN model, a new network model named CapsNet is derived. It is a deep network approach, which comprises a set of capsules. The capsule consists of a collection of neurons. The activation neuron indicates the feature of the elements that exist in the object. Every individual capsule is accountable to determine the individual element in the object and every capsule can integrate the capsules and compute the complete structure of the objects. The CapsNet comprises a multiple-layer network [18]. Fig. 1 showcases the framework of the CapsNet model.

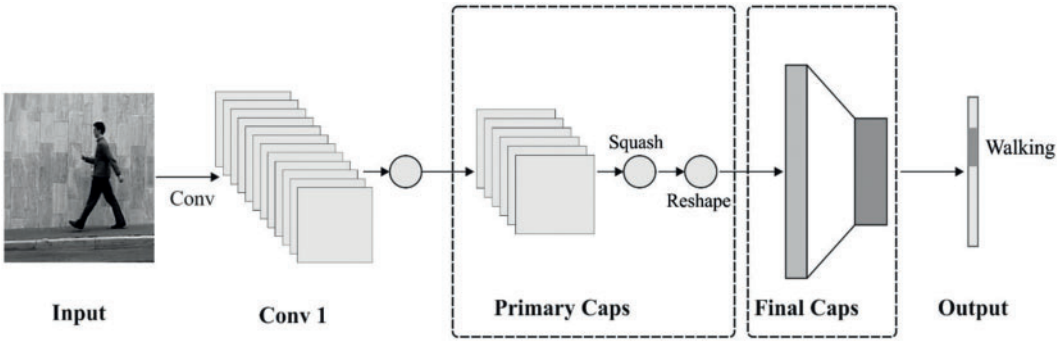

Figure 1: CapsNet structure

The length of the outcome u_j denotes the possibility of the occurrence of the respective element, and the direction of the vector u_i encodes different characteristics of the respective element. The prediction vector $\hat{u}$ signifies the belief that performs encoding of the relativity amongst the $i-th$ capsule in the low-level capsules and $j-th$ capsule in the high-level capsule by the use of a linear transformation matrix W_{ij}, as given below.

$$\hat{u}_{j|i} = W_{ij} \cdot u_i \tag{1}$$

The identified component occurrence and pose details can be used for predicting the entire existence and pose details. At the time of the training procedure, the network gets progressively learned in adjusting the transformation matrix of the capsule, paired via the respective relativity among the elements and the entire one in the objects. At the high-level capsule, the s_j and v_j denotes input and output of capsules j, correspondingly s_j signifies the total of the predicted vectors $\hat{u}_{j|i}$ with equivalent weight c_{ij} in low-level capsules i. In Eq. (2), c_{ij} indicates the coupling coefficient and can be computed using an iterative dynamic routing approach, where $\sum_j c_{ij} = 1$ and $c_{ij} \geq 0$. If $c_{ij} = 0$, there is no data transmission among the capsules i and j. When $c_{ij} = 1$, the details of capsule i can be sent to the high-level capsule j. As the output length indicates a probability value, a non-linear squash function can be utilized for ensuring that the short vector can be reduced nearer to the value of 0 and the long vector can be compacted to the value of 1. The squash function can be defined using Eqs. (2)–(4):

$$s_j = \sum_i c_{ij} \cdot \hat{u}_{j|i} \tag{2}$$

$$v_j = \frac{||s_j^2||}{1 + ||s_j^2||} \frac{s_j}{||s_j||} \tag{3}$$

$$c_{ij} = \frac{\exp(b_{ij})}{\Sigma_k \exp(b_{ik})}^{'b_{ij}} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j \tag{4}$$

If the low, as well as high-level capsules, are reliable with the prediction process, the value of c_{ij} is high and it gets reduced if they are unreliable [19]. By modifying the routing coefficients, the dynamic routing model gets ensured that the low-level capsule transmits the predictive vector to the high-level capsule, which is dependable with the prediction, therefore the output of the sub-capsule is transmitted to the precise parent capsule.

The Adam optimizer is used to optimally select the hyperparameter values of the CapsNet model. The Adam method is one of the widely employed techniques that alter the learning rate adoptively for all the parameters. This is an integration of distinct gradient optimization approaches. It is an exponentially decaying average of past squared gradient, i.e., RMSprop and Adadelta, as well as it takes the abovementioned gradients, i.e., analogous to Momentum.

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) g_t \tag{5}$$

$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t \odot g_t \tag{6}$$

whereas β_1 and β_2 represent the decay rates that are presented for following the default value. M_t and G_t is determined for estimating the mean of past gradient (initial moment) and the uncentered variation of past gradient (next moment), correspondingly. Since the decaying rate causes some bias problems, it is essential to perform the bias-correction task [20].

$$\begin{aligned} \hat{M} &= \frac{M_t}{1 - \beta_1^t} \\ \hat{G}_t &= \frac{G_t}{1 - \beta_2^t}. \end{aligned} \tag{7}$$

Hence, the upgrade value of Adam can be determined by Eq. (8)

$$\Delta\theta_t = -\frac{\alpha}{\sqrt{\hat{G} + \varepsilon}} \hat{M}_t \tag{8}$$

The gradient part of $\Delta\theta_t$ is described by

$$g_t^{'} = \frac{1}{\sqrt{\hat{G}_t + \varepsilon}} \hat{M}_t \tag{9}$$

$$\begin{aligned} \Delta\theta_t &= -\alpha \left(\frac{1}{\sqrt{\hat{G}_t + \varepsilon}} \hat{M}_t \right) \\ &= -\alpha g_t^{'}. \end{aligned} \tag{10}$$

Here, it is proven that each operation is depending on the past gradient of the present parameter that has no relation to the learning rate. Therefore, Adam has an effective performance through the learning rate method.

3.3 Human Behavior Detection and Classification

During the detection and classification process, the SAE model receives the feature vectors as input and allot proper class labels to it. In this work, the SAE was introduced by autoencoder (AE) and Logistic Regression (LR) layers [21]. The AE is a building block of the SAE classification method. It is composed of a reconstruction or decoder stage (Layer 2 to 3) and an encoder stage (Layer 1 to 2). While W and W^T (transpose of W) represents weight matrix of b and b' mode are two different bias vectors of s can be defined by nonlinearity functions such as sigmoid function; y denotes a latent parameter of input layer x, and z is assumed as a prediction of x given y has a similar shape as x. Fig. 2 illustrates the architecture of the SAE technique.

$$y = s(Wx + b) \tag{11}$$

$$z = s(W^T y + b') \tag{12}$$

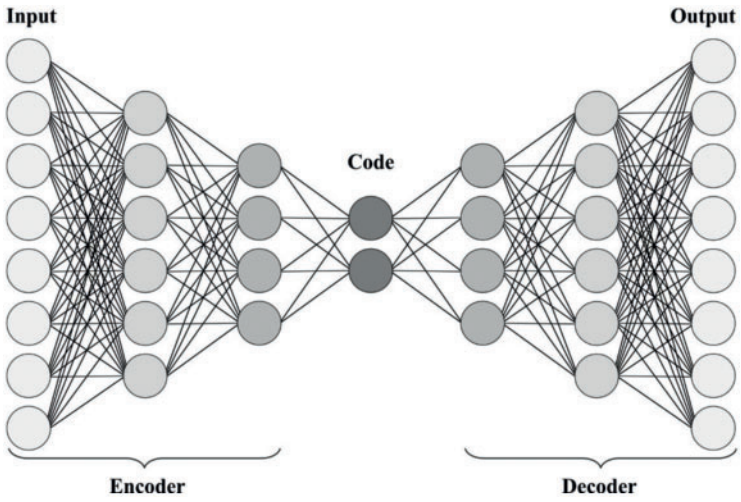

Figure 2: Structure of SAE

Various AE layer is stacked jointly in the unsupervised pretraining phase (Layer 1 to 4). The next representation $'y'$ processed as AE is applied employed as input for upcoming AE layers. Such layers undertake training as AE by minimizing reconstructed errors that are estimated simultaneously [22]. Then, reconstructed errors (loss function $L(x, z)$) are estimated in iteration. Here, it uses cross-entropy for measuring reconstruction error, in which x_k and z_k represents k^{rh} component of x and z, respectively.

$$L(x, z) = -\sum_{k=1}^{d} [x_k ln z_k + (1 - x_k) ln(1 - z_k)] \tag{13}$$

The reconstruction error is constrained under the GD application. The weight in Eqs. (11) and (12) must be upgraded as per the Eqs. (14)–(16), in which L represents a learning rate.

$$W = W - a\frac{\partial L(x, z)}{\partial W} \tag{14}$$

$$b = b - a\frac{\partial L(x, z)}{\partial b} \tag{15}$$

$$b' = b' - a\frac{\partial L(x,z)}{\partial b'} \tag{16}$$

Once the layer is pre-trained, a process is supervised under the fine-tuning stage.

3.4 Parameter Tuning Using DE Algorithm

In order to tune the weight and bias values of the SAE model, the DE algorithm is utilized and thereby improves the recognition performance. It is regarded as a population-based search approach that is initially developed by Price and Storn [23]. In the current work, a three-step adjusting method is proposed by the DE approach for solving an optimization issue. Indeed, the target of the presented technique is to enhance the model parameter of the PID-type FLC design. To perform this task, some amount of solution vectors are initialized randomly and iteratively upgraded by selection operator and genetic operator (crossover and mutation). Initially, the mutation operator is employed by a randomly selected solution (r_1, r_2 and r_3) vectors in DE population. Then, the variance among the two vectors (r_2 & r_3) multiplied by a scaling factor (F) is appended to the initial vector (r_1). Therefore, all the targeted solution X_i^G are transformed as to mutant solution vector y_i^{G+1}.

$$V_i^{G+1} = X_{r_1}^G(t) + F*(X_{r_2}^G - X_{r_3}^G), r_1 \neq r_2 \neq r_3 \neq i \tag{17}$$

Next, the crossover operators are employed for calculating a trial vector u_i^{G+1}. It can be performed by integrating the target solution vectors with the mutated vectors as follows

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, \ (\text{rand}(j) \leq CR) \ or \ j = rand \ n(i) \\ \chi_{ij}^{G}, \ (rand \ (j) > CR) \ and \ j \neq rand \ n(i) \end{cases} \tag{18}$$

Whereas $j = 1, 2, \ldots, D$, rand (j) $\in [0, 1]$ denotes the *jth* parameter of a randomly generated value. *CR* indicates the crossover probabilities i.e., random vector ranges from zero to one. rand $n(i) \in \{1, 2, \ldots, D\}$ characterizes an arbitrary number that ensures u_i^{G+1} get at one component from $v_i^{G+1,}$ or else no new parent vector is produced, therefore the population remains the same. Lastly, in a selective section if as well as only if the trial vector u_i^{G+1} produces an effective fitness function value than x_i^G, then u_i^{G+1} is fixed to x_i^{G+1}, or else, the older vector x_i^G is maintained.

$$x_i^{G+1} = \begin{cases} u_i^{G+1} & (f(u_i^{G+1}) < f(x_i^G)) \\ x_i^G & (f(u_i^{G+1}) \geq f(x_i^G)) \end{cases} \tag{19}$$

The DE technique derives a fitness function to attain improved classification performance. It determines a positive integer to represent the better performance of the candidate solutions. In this study, the minimization of the classification error rate is considered as the fitness function, as given in Eq. (20). The optimal solution has a minimal error rate and the worse solution attains an increased error rate [24].

$$fitness(x_i) = ClassifierErrorRate(x_i)$$
$$= \frac{number \ of \ misclassified \ samples}{Total \ number \ of \ samples} * 100 \tag{20}$$

4 Performance Validation

The performance validation of the proposed model takes place using two benchmark datasets namely the KTH dataset. The former KTH dataset (available at https://www.csc.kth.se/cvap/actions/)

is an open-access dataset, comprising six kinds of video actions and a resolution of 160∗120. The videos are transformed into a set of 100 frames for every video.

This section investigates the result analysis of the SDL-HBC model on the test KTH dataset. Fig. 3 shows the confusion matrix of the SDL-HBC model on the applied KTH dataset. The figure reported that the SDL-HBC model has identified 99 instances under 'Boxing' class, 99 instances under 'Handclapping' class, 97 instances under 'Handwaving' class, 96 instances under 'Jogging' class, 97 instances under 'Running' class, and 98 instances under 'Walking' class.

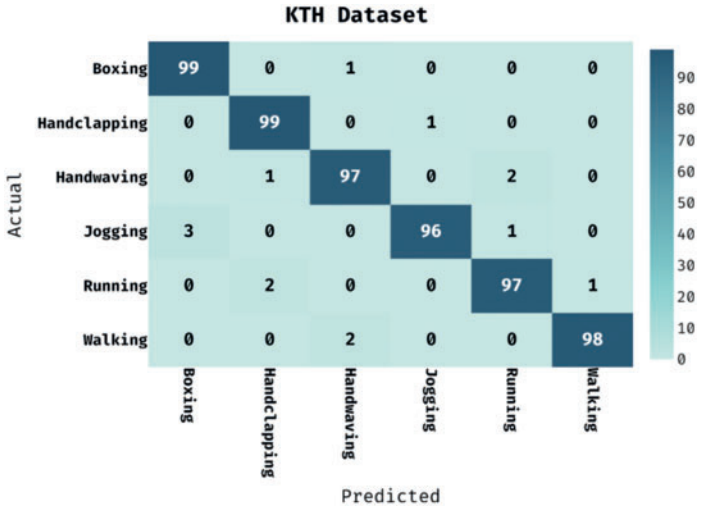

Figure 3: Confusion matrix analysis of SDL-HBC technique on KTH dataset

The performance validation of the SDL-HBC model on the test KTH dataset is offered in Tab. 1 and Figs. 4–6. The results demonstrate that the SDL-HBC model has attained effective recognition performance. For instance, under 'Boxing' class, the SDL-HBC model has resulted to $sens_y$, $spec_y$, $prec_n$, $accu_y$, and F_{score} of 0.9900, 0.9940, 0.9706, 0.9933, and 0.9802. Moreover, under the 'Handwaving' class, the SDL-HBC model has accomplished $sens_y$, $spec_y$, $prec_n$, $accu_y$, and F_{score} of 0.9700, 0.9940, 0.9700, 0.9900, and 0.9700. Furthermore, under the 'Walking' class, the SDL-HBC model has gained $sens_y$, $spec_y$, $prec_n$, $accu_y$, and F_{score} of 0.9800, 0.9980, 0.9899, 0.9950, and 0.9849. Moreover, the average result analysis of the SDL-HBC model can attain an improved average $sens_y$, $spec_y$, $prec_n$, $accu_y$, and F_{score} of 0.9767, 0.9953, 0.9768, 0.9922, and 0.9767 respectively.

Table 1: Result analysis of SDL-HBC technique on KTH dataset

Methods	Sensitivity	Specificity	Precision	Accuracy	F-Score
Boxing	0.9900	0.9940	0.9706	0.9933	0.9802
Handclapping	0.9900	0.9940	0.9706	0.9933	0.9802
Handwaving	0.9700	0.9940	0.9700	0.9900	0.9700
Jogging	0.9600	0.9980	0.9897	0.9917	0.9746
Running	0.9700	0.9940	0.9700	0.9900	0.9700
Walking	0.9800	0.9980	0.9899	0.9950	0.9849
Average	0.9767	0.9953	0.9768	0.9922	0.9767

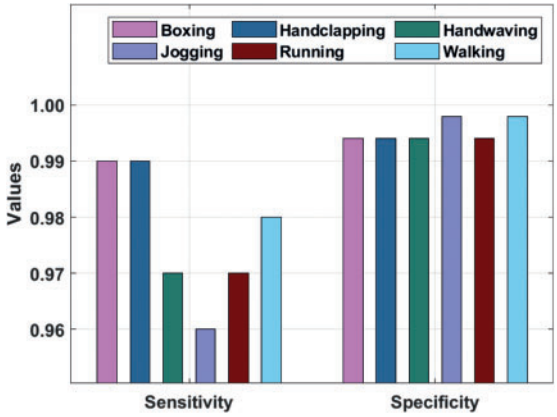

Figure 4: *Sens*$_y$ and *Spec*$_y$ analysis of SDL-HBC technique on KTH dataset

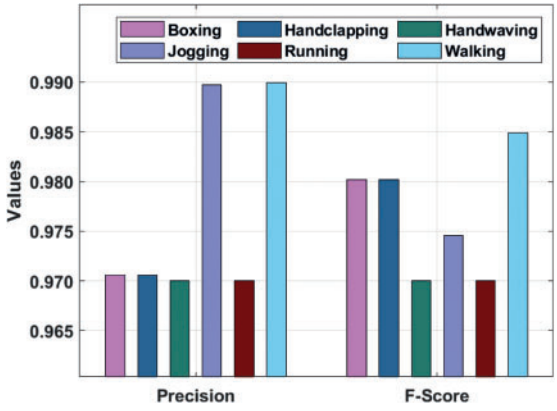

Figure 5: *Prec*$_n$ and F_{score} analysis of SDL-HBC technique on KTH dataset

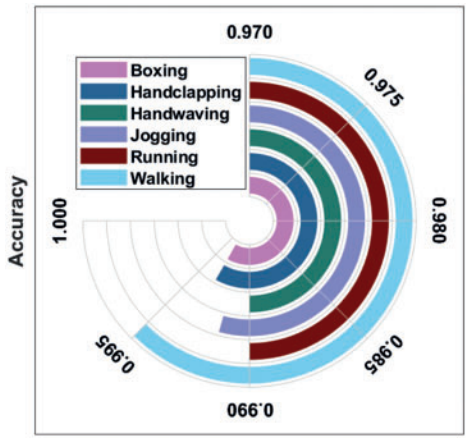

Figure 6: *Accu*$_y$ analysis of SDL-HBC technique on KTH dataset

Fig. 7 portrays the accuracy analysis of the SDL-HBC technique on the KTH dataset. The results demonstrate that the SDL-HBC approach has accomplished improved performance with increased training and validation accuracy. It is noticed that the SDL-HBC technique has gained improved validation accuracy over the training accuracy.

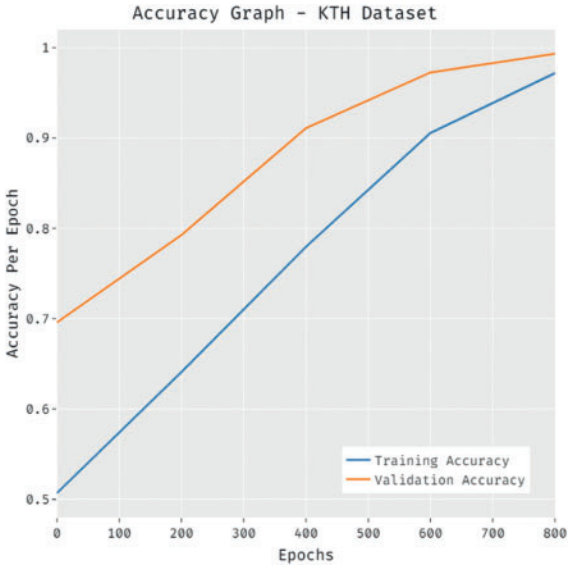

Figure 7: Accuracy graph analysis of SDL-HBC technique on KTH dataset

Fig. 8 depicts the loss analysis of the SDL-HBC technique on the KTH dataset. The results establish that the SDL-HBC system has resulted in a proficient outcome with the reduced training and validation loss. It can be revealed that the SDL-HBC technique has offered reduced validation loss over the training loss.

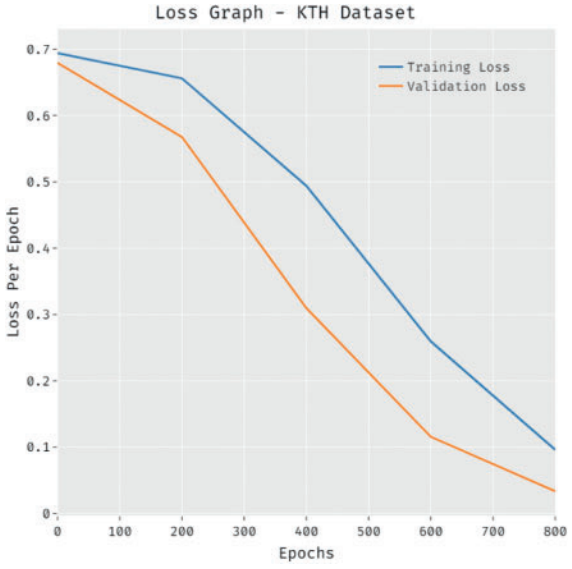

Figure 8: Loss graph analysis of SDL-HBC technique on KTH dataset

Finally, a comparative $accu_y$ analysis of the SDL-HBC model with recent approaches takes place in Fig. 9 and Tab. 2. The results show that the GMM+KF and GRNN techniques have attained lower $accu_y$ values of 0.7110 and 0.8600 respectively. In line with, the SVM-3DCNN, CNN-CAE, DTR-DNN, and GMM+KF+GRNN techniques have resulted in moderately closer $accu_y$ values of 0.9034, 0.9249, 0.9500, 0.9560, and 0.9630 respectively. However, the presented SDL-HBC model has accomplished maximum recognition performance with the $accu_y$ of 0.9922.

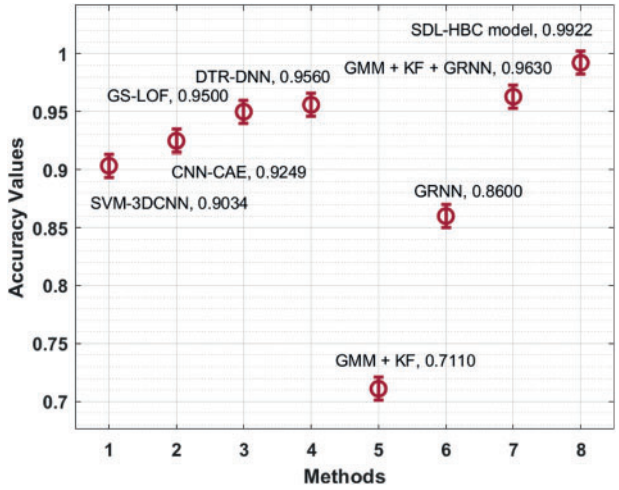

Figure 9: Comparative analysis of SDL-HBC technique on KTH dataset

Table 2: Comparative analysis of SDL-HBC technique in terms of accuracy on KTH dataset with existing approaches

Methods	Accuracy
SVM-3DCNN	0.9034
CNN-CAE	0.9249
GS-LOF	0.9500
DTR-DNN	0.9560
GMM + KF	0.7110
GRNN	0.8600
GMM + KF + GRNN	0.9630
SDL-HBC model	0.9922

5 Conclusion

In this study, a novel SDL-HBC technique has been derived for the recognition of human behavior in intelligent video surveillance systems. The proposed SDL-HBC technique aims to properly determine the occurrence of several activities in the surveillance videos. The SDL-HBC technique encompasses several stages of operations such as AMF based pre-processing, CapsNet based feature extraction, Adam optimizer-based hyperparameter tuning, SAE-based classification, and DE-based parameter tuning. The utilization of the Adam optimizer and DE algorithm results in improved

classification performance. The simulation result analysis of the SDL-HBC technique is carried out against two benchmark datasets namely KTH and UCF Sports datasets. The experimental results reported the enhanced recognition performance of the SDL-HBC technique over the recent state of art approaches. Therefore, the SDL-HBC technique can be considered an effective tool for intelligent video surveillance systems. As a part of the future scope, the performance of the SDL-HBC technique can be boosted by the design of hybrid DL models.

Funding Statement: This research was funded by the Deanship of Scientific Research at the University of Business and Technology, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 4, pp. 447–453, 2020.

[2] J. D. P. Giraldo, A. A. R. Torres, A. M. Á. Meza, and G. C. Dominguez, "Relevant kinematic feature selection to support human action recognition in MoCap data," in *Int. Work-Conf. on the Interplay Between Natural and Artificial Computation*, Springer, Cham, pp. 501–509, 2017.

[3] G. V. Kale and V. H. Patil, "A study of vision-based human motion recognition and analysis," *International Journal of Ambient Computing and Intelligence*, vol. 7, no. 2, pp. 75–92, 2016.

[4] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 1653–1660, 2014.

[5] H. Wang, A. Kläser, C. Schmid and C. Liu, "Action recognition by dense trajectories," in *CVPR 2011*, Colorado Springs, CO, USA, pp. 3169–3176, 2011.

[6] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[7] M. Latah, "Human action recognition using support vector machines and 3D convolutional neural networks," *International Journal of Advances in Intelligent Informatics*, vol. 3, no. 1, pp. 47, 2017.

[8] C. Geng and J. Song, "Human action recognition based on convolutional neural networks with a convolutional auto-encoder," in *2015 5th Int. Conf. on Computer Sciences and Automation Engineering (ICCSAE 2015)*, Sanya, China, pp. 933–938, 2016.

[9] I. Jegham, A. B. Khalifa, I. Alouani and M. A. Mahjoub, "Vision-based human action recognition: An overview and real-world challenges," *Forensic Science International: Digital Investigation*, vol. 32, pp. 200901, 2020.

[10] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B. Y. Choi *et al.,* "Real-time human detection as an edge service enabled by a lightweight CNN," in *2018 IEEE Int. Conf. on Edge Computing (EDGE)*, San Francisco, CA, pp. 125–129, 2018.

[11] R. Nawaratne, D. Alahakoon, D. D. Silva and X. Yu, "Spatiotemporal anomaly detection using deep learning for real-time video surveillance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 393–402, 2020.

[12] W. Bouachir, R. Gouiaa, B. Li and R. Noumeir, "Intelligent video surveillance for real-time detection of suicide attempts," *Pattern Recognition Letters*, vol. 110, pp. 1–7, 2018.

[13] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.

[14] Y. Han, P. Zhang, T. Zhuo, W. Huang, and Y. Zhang, "Going deeper with two-stream ConvNets for action recognition in video surveillance," *Pattern Recognition Letters*, vol. 107, pp. 83–90, 2018.

[15] A. Ullah, K. Muhammad, I. U. Haq and S. W. Baik, "Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments," *Future Generation Computer Systems*, vol. 96, pp. 386–397, 2019.
[16] H. SShuka, N. Kumar and R. P. Tripathi, "Median filter-based wavelet transforms for multilevel noise," *International Journal of Computer Applications*, vol. 107, no. 14, pp. 11–14, 2014.
[17] Z. Gao, "An adaptive median filtering of salt and pepper noise based on local pixel distribution," in *2018 Int. Conf. on Transportation & Logistics, Information & Communication, Smart City (TLICSC 2018)*, Chengdu City, China, pp. 473–483, 2018.
[18] S. Sabour, F. Nicholas, and G. E. Hinton. "Dynamic routing between capsules," arXiv preprint arXiv:1710.09829, 2017.
[19] H. Chao, L. Dong, Y. Liu, and B. Lu, "Emotion recognition from multiband EEG signals using CapsNet," *Sensors*, vol. 19, no. 9, pp. 2212, 2019.
[20] C. Zhang, M. Yao, W. Chen, S. Zhang, D. Chen *et al.,* "Gradient descent optimization in deep learning model training based on multistage and method combination strategy," *Security and Communication Networks*, vol. 2021, pp. 1–15, 2021.
[21] J. Xu, L. Xiang, and Q. Liu, "Stacked sparse autoencoder (SSAE) for nuclei detection of breast cancer histopathology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 1, pp. 119–130, 2016.
[22] G. Liu, H. Bao and B. Han, "A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–10, 2018.
[23] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
[24] E. Y. Bejarbaneh, A. Bagheri, B. Y. Bejarbaneh, S. Buyamin, and S. N. Chegini, "A new adjusting technique for PID type fuzzy logic controller using PSOSCALF optimization algorithm," *Applied Soft Computing*, vol. 85, pp. 105822, 2019.