Computers, Materials & Continua
DOI: 10.32604/cmc.2022.028696
Article

Tech Science Press

A Hybrid Duo-Deep Learning and Best Features Based Framework for Action Recognition

Muhammad Naeem Akbar[1,*], Farhan Riaz[1], Ahmed Bilal Awan[1], Muhammad Attique Khan[2], Usman Tariq[3] and Saad Rehman[2]

[1]National University of Sciences and Technology (NUST), Islamabad, 46000, Pakistan
[2]Department of Computer Science, HITEC University, Taxila, Pakistan
[3]College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, Al-Kharaj 11942, Saudi Arabia
*Corresponding Author: Muhammad Naeem Akbar. Email: naeemakbar@ceme.nust.edu.pk

Received: 15 February 2022; Accepted: 06 April 2022

Abstract: Human Action Recognition (HAR) is a current research topic in the field of computer vision that is based on an important application known as video surveillance. Researchers in computer vision have introduced various intelligent methods based on deep learning and machine learning, but they still face many challenges such as similarity in various actions and redundant features. We proposed a framework for accurate human action recognition (HAR) based on deep learning and an improved features optimization algorithm in this paper. From deep learning feature extraction to feature classification, the proposed framework includes several critical steps. Before training fine-tuned deep learning models – MobileNet-V2 and Darknet53 – the original video frames are normalized. For feature extraction, pre-trained deep models are used, which are fused using the canonical correlation approach. Following that, an improved particle swarm optimization (IPSO)-based algorithm is used to select the best features. Following that, the selected features were used to classify actions using various classifiers. The experimental process was performed on six publicly available datasets such as KTH, UT-Interaction, UCF Sports, Hollywood, IXMAS, and UCF YouTube, which attained an accuracy of 98.3%, 98.9%, 99.8%, 99.6%, 98.6%, and 100%, respectively. In comparison with existing techniques, it is observed that the proposed framework achieved improved accuracy.

Keywords: Action recognition; deep learning; features fusion; features selection; recognition

1 Introduction

Human Action Recognition (HAR) was formally founded by Leonardo da Vinci (1452–1519), who was interested in human appearance and motion analysis, in order to teach students how to correctly draw people [1]. At the moment, most research is focusing on HAR, which has a wide range of applications such as TV production, entertainment, education, social studies, security [2], intelligent video surveillance [3], home monitoring, human–machine interfacing, video storage and retrieval [4], assistive living and assistant robots [5]. It covers a wide range of research topics, including human detection in video [6] human pose estimation, human tracking, and the analysis and comprehension of human activities [7]. In the last 10–15 years, there has been significant progress in HAR research, which has resulted in commercial products [8].

Action recognition may include a large variety of human action depending upon the requirements for various applications [9]. Like in security or surveillance applications running, Jumping, pushing, and punching is important however sitting, mobile calling also showed much concern [10]. Still, image-based HAR is for identifying the action of a person from a single image without considering temporal information [11]. Action representation and analysis-based HAR involves feature representation using feature extraction techniques and machine learning techniques [12]. Abnormal activity detection are used for video surveillance to prevent crime or for inspecting crime scene [13]. Action classification is a pattern recognition and machine learning problem and many techniques are introduced in the literature [14]. The famous techniques are graph-based, SVM [15], nearest neighbor [16], HMM, ELM, and named a few more [17]. Graph-based methods are used to classify input features of HAR that include Random Forest (RF), Geodesic Distance Isograph (GDI), to name a few [18]. The concept of Support Vector Machine (SVM) is to separate the data points using a hyperplane [19]. Non-linear data is classified by performing multi-class learning using a one-*vs.*-one SVM classifier having a polynomial kernel [20]. These techniques give the better results for smaller set of datasets but for the larger size of dataset, the accuracy goes down and computational time jumped up [21].

Deep Learning (DL) is a technique that instructs computers to perform the task similar to that of the naturally conducted tasks by a human brain. Convolutional Neural Network (CNN) [22], RNN, Long Short-Term Memory (LSTM), Deep Belief Network (DBN), as well as Generative Adversarial Network (GAN) are widely used networks for the action recognition task [23]. In CNN, maps are created using local neighbourhood information for each image that extracts the deep features [24]. Convolution, activation, pooling, fully connected, and output layer are all important layers in the CNN architecture. Because of the 1D vector size, features are extracted from fully connected layers. When the extracted features are insufficient for classification, a few researchers have used fusion techniques [25]. According to the analysis of recent studies, some redundant features are also added during the fusion process; thus, the researchers used features reduction and selection techniques. Only important features are chosen for the final classification using features selection techniques [26].

Several techniques have been introduced by computer vision researchers from last couple of years for human action recognition (HAR) [27]. They focused on both classical techniques and deep learning based techniques [28]. The classical techniques are based on the region of interest detection, traditional feature extraction such as shape, texture, and point, features reduction, and classification through machine learning methods [29]. The deep learning showed much improvement in the recent years for several applications and HAR is one of the most emerging applications [30]. Through deep learning, the researchers employed deep features from the dense layers and further optimized information through some feature selection techniques. A few researchers employed skeleton based information to get the movement of human in the video frames and then trained on deep learning networks. Shen et al. [31] presented a 3D skeleton based framework for HAR. The features are extracted through skeletons and trained a LSTM model. Then through time series data, skeleton points are composed in the network and obtained a complex LSTM network. The experimental process was conducted on three publicly available datasets and achieved improved recognition accuracy. Xie et al. [32] presented a temporal CNN based architecture for HAR. The CNN is opted at the first step for the information of inputs. After that, a novel cross-spatial temporal graph CNN is introduced to get the joints information. The modeling capability is enhanced after that by employing temporal attention layer. Three datasets are employed for the experimental process and indicated improved accuracy. Zhang et al. [33] suggested a new approach for action recognition which contains the fusion of CNN and BLSTM networks. In this method, an algorithm based on swarm intelligence is also introduced for recognizing the optimized hyperparameter of the deep neural networks. They tested their approach by using UCF101, UCF50, and KTH datasets that show improved recognition results. Nazir et al. [34] introduced the dynamic Spatio-temporal Bag of expression (DSTBoE) technique for HAR to overcome the problems related to occlusion, interclass variations, and view independence noticed in realistic scenes. In this technique, SVM is used for classification. They applied their approach by using UCF50, UCF11, KTH, and UCF sports datasets and obtained an accuracy of 94.10%, 96.94%, 99.21%, and 98.60%, respectively. Rahimi et al. [35] presented a kernelized Grassmann manifold learning-based method for HAR to overcome the issues related to outliers and noise in the given training data. They evaluated their approach by using UCF101, UTD-MHAD, MSR action 3D, UCF sport, and KTH datasets and attained higher accuracy on all used datasets.

Kumar et al. [36] introduced a novel Gated RNN method for action recognition. They evaluated their approach by using UCF Sports dataset and attained an accuracy of 96.8%. Kiran et al. [37] suggested a novel approach using deep learning for action classification. They used a pre-train deep model named resnet50. Global Avg pool and FC layer are used to extract deep features and then fuse them. Finally, the feature vector is used for classification by a classifier. They used KTH, UT-Interaction, UCF YouTube, UCF Sports, and IXMAS datasets for their approach and attained better accuracy. Li et al. [38] suggested a new residual network for HAR based on feature fusion and Global Avg pooling (GAP). They tested their technique by using CAVIAR, UCF101, UCF11, UT-Interaction datasets and attained accuracy above 90%. Ahmed et al. [39] suggested a novel pose descriptor for action recognition. They tested their approach by using HCA, CASIA, UCF11, and UT-Interaction datasets and obtained an accuracy of 88.72%, 98%, 99%, and 96.4% respectively.

The preceding methods concentrated on HAR's temporal data. They improved accuracy by using several publicly available datasets. Few scientists concentrated on the fusion process. The combination of multi-level features improves recognition accuracy while increasing computational time. The major challenges that researchers are still facing: (i) deep learning networks are used to extract the most relevant features; (ii) redundant and irrelevant features not only reduce accuracy but also increase computational time. In this paper, we proposed a new framework for HAR based on deep learning and a hybrid PSO algorithm. Our most significant contributions are as follows:

- Fine-tuned MobileNet V2 and DarkNet 53 pre-trained deep learning models on action recognition datasets using transfer learning. The weights of the 50% layers have been freeze instead of all layers except new FC layer.
- Features are extracted from middle layers (average pooling and convolution) and fused by employing canonical correlation analysis (CCA) approach. The CCA technique is refined by single threshold function called Non-Redundant Function.
- A hybrid particle swarm optimization algorithm is opted. The proposed hybrid algorithm is based on the crow search optimization output features.

The rest of the article is organized in the following order: Proposed methodology for HAR is presented in Section 2. This section includes the convolutional neural network (CNN) framework, deep learning features extraction, fusion of features, and selection of best features using hybrid PSO algorithm. Section 3 presents the experimental results and discussion of the proposed framework. Finally, the Section 4 concludes this article.

2 Proposed Methodology

The proposed method encompasses features extraction followed by features fusion and selection of optimized features which are used for classification by various supervised learning-based classifiers to identify the human actions. Fig. 1 illustrated the proposed framework of HAR. In this figure, it is illustrated that the initially original video frames are normalized and train fine-tuned deep learning models – MobileNet-V2 and Darknet53. The pre-trained deep models are utilized for the features extraction that is fused using canonical correlation approach. After that, an improved PSO based optimization algorithm is opted for best features selection. The selected features were subsequently utilized for classification of actions through different classifiers.

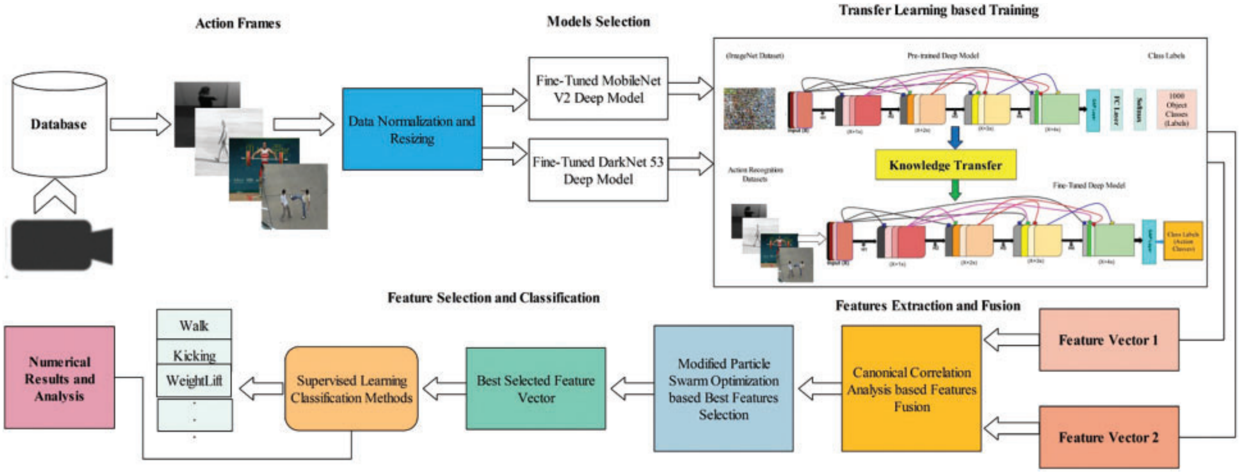

Action Frames
Models Selection
Transfer Learning based Training
Database
Data Normalization and Resizing
Fine-Tuned MobileNet V2 Deep Model
Fine-Tuned DarkNet 53 Deep Model
(ImageNet Dataset)
Pre-trained Deep Model
Class Labels
FC Layer
Softmax
1000 Object Classes (Labels)
Knowledge Transfer
Action Recognition Datasets
Fine-Tuned Deep Model
Class Labels (Action Classes)
Feature Selection and Classification
Features Extraction and Fusion
Feature Vector 1
Feature Vector 2
Canonical Correlation Analysis based Features Fusion
Modified Particle Swarm Optimization based Best Features Selection
Best Selected Feature Vector
Supervised Learning Classification Methods
Walk
Kicking
WeightLift
Numerical Results and Analysis

Figure 1: Proposed framework for human action recognition using deep learning and optimal features fusion

2.1 Datasets Description and Normalization

In this work, we utilized five publicly available datasets such as KTH [40], UT-Interaction [41], UCF Sports [42], Hollywood [43], and UCF YouTube [44]. The KTH Human activities dataset contains six activity classes: walking, running, jogging, boxing, waving, and clapping. The UT-Interaction consists of 6 action classes, namely pushing, pointing, kicking, punching, handshaking, and hugging. The UCF Sports dataset contains 13 action classes such as divingside, skateboarding front, run side, and named a few more having total 182 video sequences. The Hollywood dataset consists of 8 action classes, whereas the numbers of video sequences are above 600. The YouTube action dataset consists of 10 action classes such as basketball, walking, tennis swing, biking, swing, golf swing, and few more, whereas the number of video sequences are more than 1600. Each dataset video sequences are converted into frames and resized into 512 × 512 × 3. A few sample video frames are illustrated in Fig. 2.

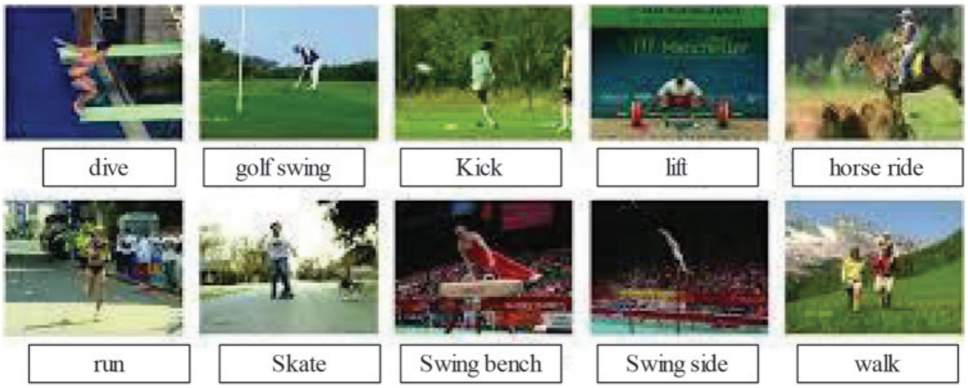


Figure 2: Sample frames of different human actions collected from UCF Sports dataset [42]

2.2 *Convolutional Neural Network*

Convolutional neural network (CNN) is a strong deep neural network for object recognition and image classification [45]. All the neurons in a CNN are connected in a feed forward fashion to the next layer's neurons [46]. A CNN model consists of several layers such as convolution, pooling, ReLu, and fully linked layers.

The convolution layer is responsible for recognizing and extracting local features derived from a sample of input image $A_p^{(s\times r\times p)}$ where, $A_p^{(s\times r\times p)} \in \tilde{\phi}_{fI}\,(i,j)$ and $s = r$ for square input. Consider the following image as an example of an input: $A_p = \{A_1, A_2, \ldots .A_n\}$, where the total number of training datasets is denoted by n. The result for each input image is $B_p = \{B_1, B_2, \ldots .B_n\}$, where $B_p \in \{1, 2, \ldots .Cl_n\}$ and Cl_n indicates the number of classes. A kernel slides over the input picture as $A_p^{(s\times r\times p)} * Z_p^{(\acute{s}\times\acute{r}\times\acute{p})}$ in the convolution layer. The following relation is used to extract local features:

$$\hat{U}_m^l = \text{б}\left(\sum\nolimits_{m=1}^{n} A_m^{l-1} \times \varphi_m^l + h_l^p\right) \tag{1}$$

where $\hat{U}_m^l$ generates a feature map for the layer, and the trainable parameters for layer are $l; \mathbb{C}_m^l + h_l^p$, $l; \varphi(.)$ denotes the function of activation.

A pooling layer, a non-linear down sampling approach, is also used in CNN. It effectively integrates two essential ideas, first is max pooling and second is convolution. The first stage aggregates a collection of maximal results for feature reduction as well as resistance to noise and volatility. The following equation describes the max-pooling configuration:

$$MxPool = max\left(y_{2m-1}^{l-1}, y_{2m}^{l-1}\right), l = 2\xi\forall\xi \in \mathbb{R} \tag{2}$$

where, $y_{2m-1}^{l-1} \in \cap U_m^l$ and $\mathbb{R}$ denote the real number weight values. A completely linked feed forward layer, fully connected follows the convolution and pooling layers. It works on the same principles as a conventional fully connected feed forward network, with the addition of a set of inputs and outputs.

$$\rho_k^l = Sig\left(\sum\nolimits_{m=1}^{n} \tilde{x}_m^{l-1} \times \Upsilon_{mk}^l + h_l^k\right) \tag{3}$$

where, $\tilde{x}_m^{l-1}$ is the output of last connected layer that includes 1D weight matrix.

2.3 Transfer Learning

Transfer learning (TL) is the process of reuse a pre-trained model for another task. The main purpose of TL is to reuse a CNN network for less number of training data. Mathematically, TL is defined as: A source domain is provided $\xi_{dm} = \left\{\left(p_1^{\xi}, q_1^{\xi}\right), \ldots.., \left(p_i^{\xi}, q_i^{\xi}\right), \ldots\ldots \left(p_n^{\xi}, q_n^{\xi}\right)\right\}$, where $\left(p_n^{\xi}, q_n^{\xi}\right) \in \mathbb{R}$; with specific learning objective, ξ_L, and Target domain $\psi_{dm} = \left\{\left(p_1^t, q_1^t\right), \ldots.., \left(p_i^t, q_i^t\right), \ldots\ldots \left(p_m^t, q_m^t\right)\right\}$, having learning task ψ_{dm}, $\left(p_n^t, q_n^t\right) \in \mathbb{R}$. The size of the training data is $((m, n)|n \ll m)$, and the corresponding labels are J_1^{ξ} and J_1^{ψ}. TL's major role is to enhance the target function ψ_{dm} learning ability and leveraging the information from the source ξ_{dm} and the target ψ_{dm}. Visually, the process of TL is presented in Fig. 3.

2.4 Deep Learning Features

In this work, we utilized two pre-trained deep learning models named- MobileNet-V2 and DarkNet-53 for deep features extraction. MobileNet-V2 [47] is a light weight CNN model having 5.2 million parameters. This network includes 53 deeper layers and accepts an input of dimension 224 × 224. This network includes residual layers that later converted high dimensional input into light dimensional output. This network includes convolutional layers, bottleneck layers, residuals, and fully connected layer. Originally, this network trained on ImageNet dataset having 1000 object classes. In this work, we fine-tuned this model and removed the FC layer. Then, a new layer is added and trained on action recognition datasets. The training is performed through TL. The working of TL is described in Fig. 3 and Section 2.3. Features are extracted from the global average pooling layer of the fine-tuned model and attained a feature vector of dimension $N \times 1280$. The DarkNet-53 [48] is light weight and much better than DarkNet19 and ResNet 101 deep models. This network accepts the input of an image 256 × 256. The total number of parameters of this network is 41.6 million. Originally, this network is also trained on ImageNet dataset having 1000 object classes. In the fine-tuning process, we removed the last layer named Conv53 and added a new layer named New_conv53. After that, the entire model is connected and trained through TL on action datasets. Features are extracted from the trained deep model of layer average pool having dimension $N \times 1024$. After that, the extracted features are fused using canonical correlation based approach.

2.5 Features Fusion

In this work, we opted canonical correlation analysis (CCA) [49] for deep features fusion. Consider two feature vectors $\alpha = [\alpha_1, \alpha_2, \alpha_3, \ldots.., \alpha_n] \in K_{pxn}$ $\beta = [\beta_1, \beta_2, \beta_3, \ldots.., \beta_n] \in K_{qxn}$. In this paper it is assumed that both α and β are centered and also scaled. They were normalized with $\|\alpha\|_v = 1$ and $\|\beta\|_v = 1$. CCA gets pair of linear transformations, one for each of the sets of variables, such that when the set of variables transformed, the corresponding coordinates are maximally correlated. Mathematically, the method computes two projection vectors $\mathrm{V}_\alpha \in K_p \sim and\ \mathrm{V}_\beta \in K_q$ such that CCA maximized.

$$\alpha = \frac{A_\alpha^t \alpha \beta^t A_\beta}{\sqrt{(A_\alpha^t \alpha \alpha^t A_\alpha)(A_\beta^t \beta \beta^t A_\beta)}} \tag{4}$$

As correlation coefficient is invariant, hence Eq. (4) can be written as:

$$\max = A_\alpha^t \alpha \beta^t A_\beta \tag{5}$$

$$s.t = \begin{cases} A_\alpha^t \alpha \alpha^t A_\alpha = 1 \\ A_\beta^t \beta \beta^t A_\beta = 1 \end{cases} \tag{6}$$

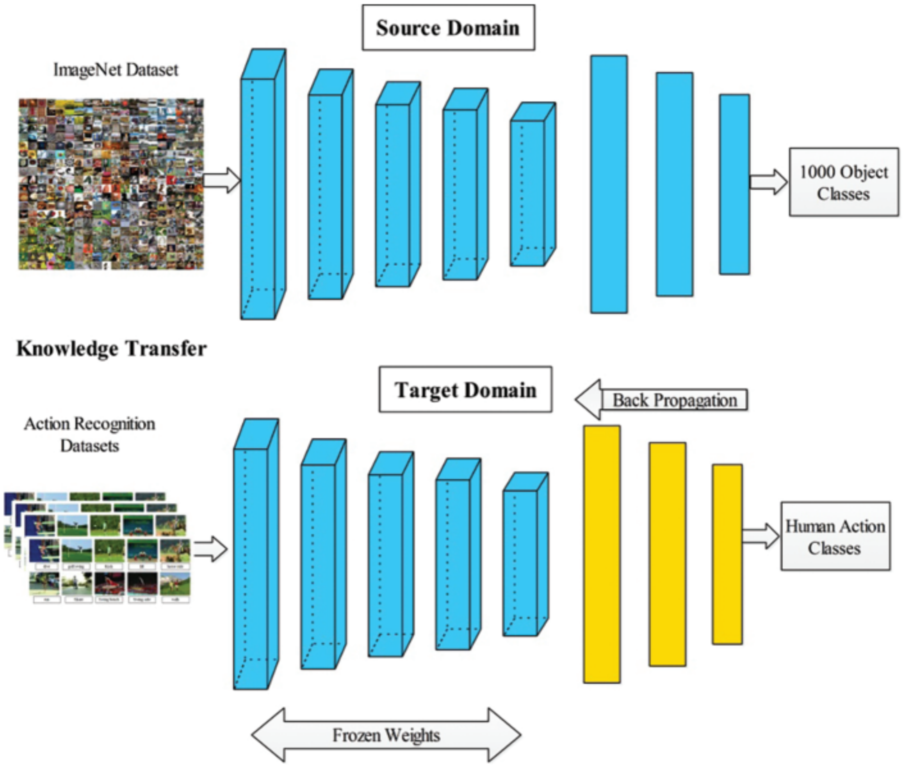


Figure 3: Transfer learning based fine-tuned CNN models training for HAR

This problem can be solved by using the Lagrange multiplier method that resorts to the generalized eigenvalue problem as follows:

$$\begin{pmatrix} 0 & \alpha\beta' \\ \beta\alpha' & 0 \end{pmatrix}\begin{pmatrix} A_\alpha \\ A_\beta \end{pmatrix} = \Omega\begin{pmatrix} \alpha & \alpha' & 0 \\ 0 & \beta & \beta' \end{pmatrix}\begin{pmatrix} A_\alpha \\ A_\beta \end{pmatrix} \tag{7}$$

If $\beta\beta'$ is non singular the above problem is equivalent to:

$$\alpha\beta'(\beta\beta')^{-1}\beta\alpha' A_\alpha = \Omega^2\alpha\alpha' A_\alpha \tag{8}$$

$$A_\beta = \frac{(\beta\beta')^{-1}\beta\alpha' A_\alpha}{\Omega} \tag{9}$$

Here $\frac{A\alpha}{\|\alpha^t A\alpha\|}$, $\frac{A_\beta}{\|\beta^t A_\beta\|}$ are called the canonical vectors. Based on the above formulations (Eqs. (4)–(9)), the final vectors are obtained by the following equation:

$$\begin{pmatrix} 0 & \alpha\beta' \\ \beta\alpha' & 0 \end{pmatrix}\begin{pmatrix} \widehat{A_\alpha} \\ \widehat{A_\beta} \end{pmatrix} = \Omega\begin{pmatrix} \alpha & \alpha' + xl & 0 \\ 0 & \beta & \beta' + yl \end{pmatrix}\begin{pmatrix} \widehat{A_\alpha} \\ \widehat{A_\beta.} \end{pmatrix} \tag{10}$$

The resultant fused vector is obtained of dimension $N \times 1742$ after applying Eq. (10). The resultant vector consists of several redundant features that was analyzed during the experimental process. Therefore, we opted an improved PSO based algorithm for best features selection.

2.6 Features Selection

In this work, we opted an improved feature selection algorithm named Particle Swarm with Crow Search Optimization (PSO-CSA). The main working of feature selection algorithm is based on the following three steps. In the first step, features are selected through PSO based algorithm [50]. In the second step, the resultant feature vector is passed to CSA for further refinement [51]. In the third step, best selected features are further refined using entropy based function for final selection. The Fine-KNN is utilized as a fitness function and error is computed as a loss of selected features. The final selected features are finally classified using machine learning classifiers. The working of algorithm is defined as follows:

Consider, we have M particle locations and velocities in a space of P dimensions offer solutions initiated in a random circumstance in the PSO algorithm. In iteration i, the solution of particle v is as indicated in Eq. (11). The current particle solution is then updated for the local and global domains, which are computed using Eqs. (12) and (13) respectively.

$$x_v^i = \left[x_{1v}^i, x_{1u}^i, \ldots, x_{v,p}^i\right], v - 1, 2, \ldots M \tag{11}$$

$$L_{v,u}^{i+1} = wgl_{v,u}^i + CO_1 y_1 \left(fbst_{v,u}^i - x_{v,u}^i\right) + CO_2 y_2 (gbst_u^i - x_{v,u}^i) \tag{12}$$

The terms CO_1 and CO_2 refer to cognitive and social factors, respectively, [0,1] are the random values y_1 and y_2. The inertia weight is denoted by wg, describes how the particle's prior velocity impacts the velocity of the subsequent iteration. The value of wg is determined by Eq. (13).

$$wg = wg_{max} - iter.\left(\frac{wg_{max} - wg_{min}}{max_{-iter}}\right) \tag{13}$$

$$x_{v,u}^{i+1} = x_{v,u}^i + L_{v,u}^{i+1} \tag{14}$$

The best attained features are represented by $x_{v,u}^{i+1}$ of dimension $N \times 1210$. This resultant vector is further passed to CSA for the further refinement in the selected features.

Consider, we have P dimension input PSO vector (population) and the population has M solutions (number of crows). The position of each crow v at iteration i is described by x vector as $x_v^i = \left[x_{1v}^i, x_{2v}^i, x_{3v}^i, \ldots, x_{pv}^i\right]$ for $v = 1, 2, 3, \ldots, M$, where x_v^i is the crow v's probable position solution in dimension p. If a crow v wants to take food from another crow u, one of two things could happen: (i) Crow u doesn't really track crow v but crow v will find crow v's food storage and update its position according to Eq. (15).

$$x_v^{(i+1)} = x_v^{(i)} + y_v * zL_u^{(i)} * (n_u^{(i)} - x_v^{(i)}) \tag{15}$$

where length of the flight is denoted by zL and y_v has been a random number belongs to $[0, 1]$. When crow u feels that crow v following her to find her food. In this case, the crow u travels at random to deceive the crow v. The two situations can be mathematically combined as follows Eq. (16):

$$x_v^{(i+1)} = \begin{cases} x_v^{(i)} + y_v * z_u^{(i)} * \left(n_u^{(i)} - x_v^{(i)}\right), & y_u \geq AW_v^i \\ Select\ a\ position\ at\ random, & Otherwise \end{cases} \tag{16}$$

where awareness probability of crow u at iteration i is represented by AW_v^i, y_v and y_u are the random numbers belong to [0, 1].The value of flight length has an impact on crow's ability to seek. High values of flight length make a significant contribution to global search and low values on the other hand help with local searches. During the algorithm's execution, each crow is evaluated using a well-defined fitness function (Fine-KNN). The crows then change their places based on the fitness score. Each new position is checked for viability. According to Eq. (17), the crow's memories are modified as follows:

$$n_v^{(i+1)} = \begin{cases} x_v^{(i+1)} ifz \left(x_v^{(i+1)}\right) \text{ is better than } \left(x_v^{(i)}\right) \\ n_v^{(i)}, \quad Otherwise \end{cases} \tag{17}$$

The resultant $x_v^{(i+1)}$ features of dimension $N \times 982$ are passed to entropy function and sort into descending order. From the sorted vector, the top 90% features are selected for final classification. In the classification phase, several classifiers are opted, mentioned in the Results section.

3 Results and Discussion

The results of proposed framework of HAR are presented in this section in terms of numerical values, time plots, and confusion matrixes. Six publicly available datasets are employed in this work for the experimental process- KTH, UT-Interaction, UCF Sports, Hollywood, IXMAS, and UCF YouTube. During the training process, several hyper parameters are considered for training of pre-trained models such as learning rate is 0.005, epochs are 100, drop out facto is 0.5, per epoch iterations are 30, and stochastic gradient descent (SGD) optimizer. The 50:50 approach is opted where the K-Fold cross validation is utilized and value of K = 10. The performance of each dataset is computed based on the several measures such as accuracy, time, and recall rate. The classification accuracy is computed using several machine learning classifiers such as LDA, SVM, KNN, and Bagged tree. The entire framework simulations are conducted on MATLAB2021a using Desktop computer having 16GB of RAM and 8GB of graphics card.

3.1 Results

KTH Results: Tab. 1 presents the numerical results of proposed HAR framework on KTH dataset. In this table, Cubic SVM classifier attained the maximum accuracy of 98.30% in 240.89 (s). The recall rate is also computed of this classifier that is 0.98. The performance of this classifier can be further verified through a confusion matrix, illustrated in Fig. 4. This figure described that the correct predicted values, given in the diagonals. The accuracy for the rest of the classifiers is also computed, as described in this table that shows that the average accuracy is above 90%. Moreover, the computational time of each classifier is also noted, as plotted in Fig. 5. This figure shows that the LDA classifier execution time (14.62 s) is minimum than the rest of the classifiers. Moreover, the MGSVM classifier consumed higher time of 307.42 (s). Overall, the Cubic SVM classifier shows the better recognition performance.

Table 1: Proposed action recognition framework results on KTH dataset

Classifiers	Time (s)	TP	FN	Recall	Accuracy (%)
Linear Discriminant	14.62	563.70	36.20	0.94	93.80
Linear SVM	151.41	580.20	19.90	0.97	96.70
Fine KNN	180.78	543.50	56.60	0.91	90.90
Fine Tree	46.53	566.60	33.40	0.94	94.30
Gaussian Naïve Bayes	14.82	552.30	47.70	0.92	92.00
Cubic SVM	240.89	589.60	10.50	0.98	**98.30**
Quadratic SVM	189.84	588.40	11.60	0.98	98.10
Medium Gaussian SVM	307.42	569.60	30.40	0.95	95.00
Weighted KNN	181.25	420.40	179.60	0.70	71.30
Bagged Tree	124.39	558.40	41.60	0.93	93.20

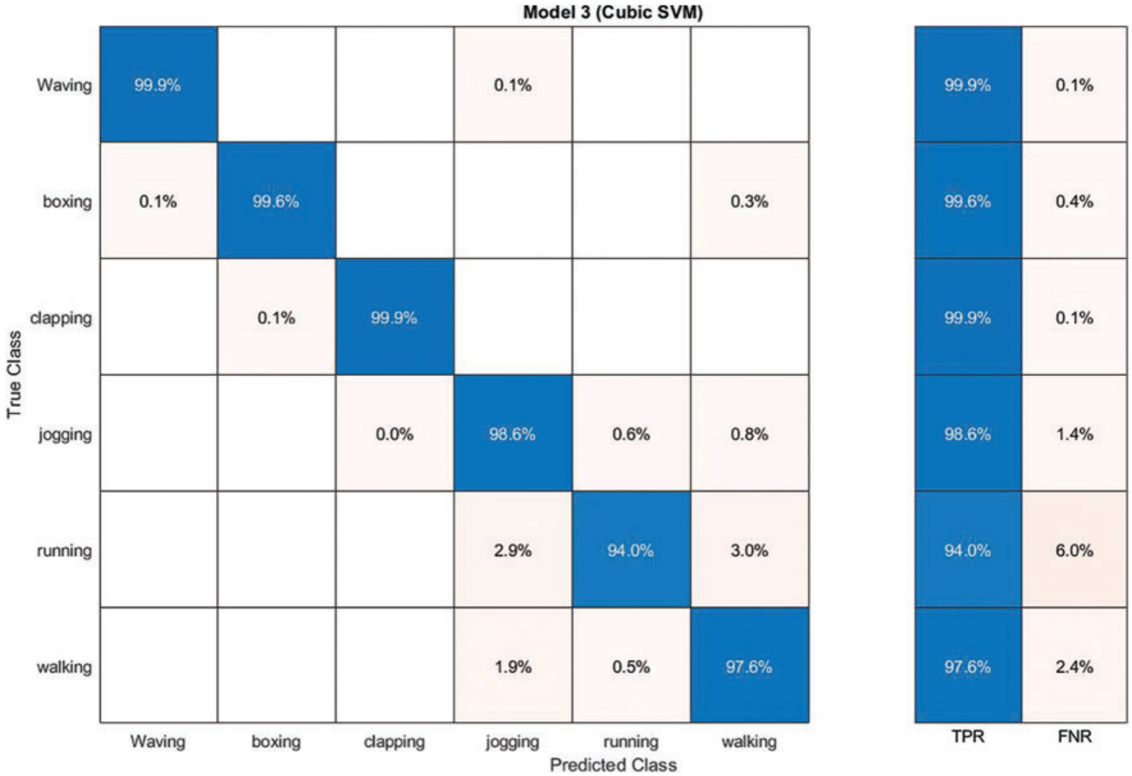


Figure 4: Confusion matrix of proposed framework on KTH dataset

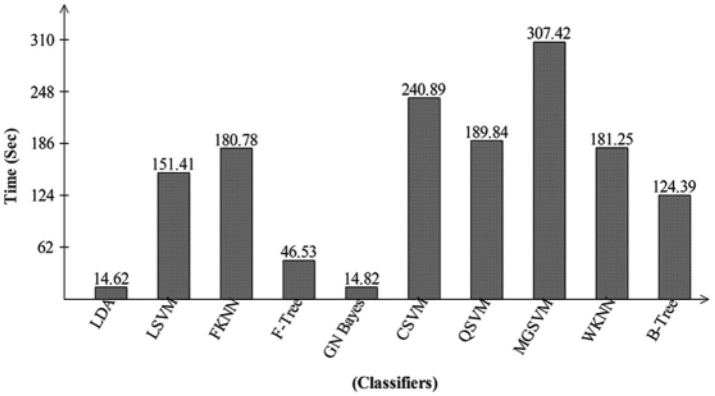


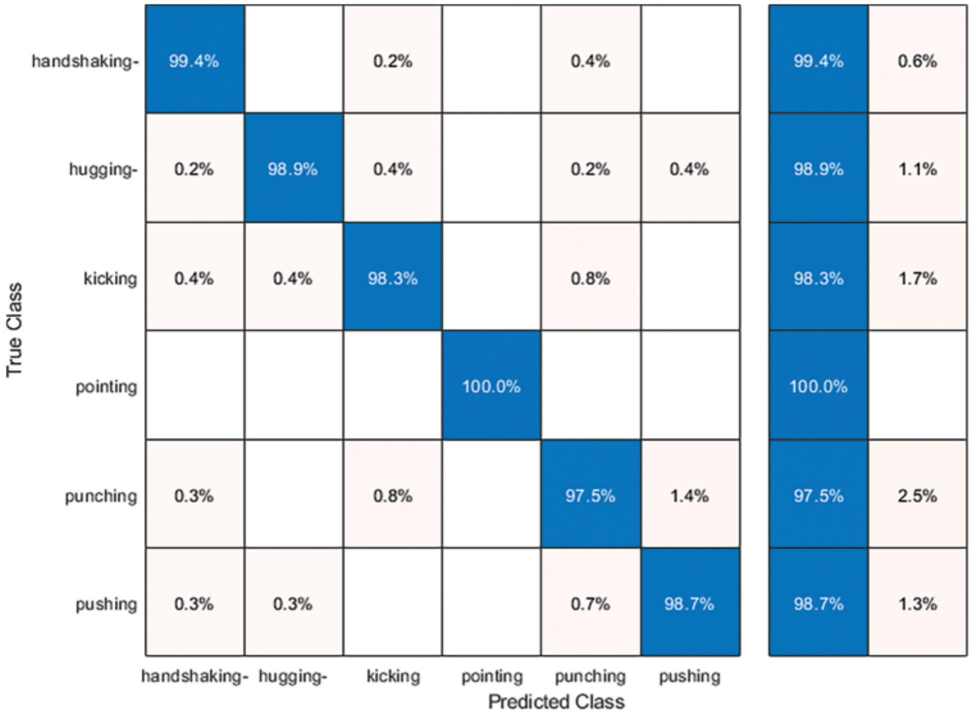
Figure 5: Computational time based comparison among selected classifiers on KTH dataset

UT-Interaction Results: Tab. 2 presents the numerical results of proposed HAR framework on UT-Interaction dataset. In this table, Fine KNN classifier attained the maximum accuracy of 98.90% in 21.22 (s). The recall rate is also computed of this classifier that is 0.99. The performance of this classifier can be further verified through a confusion matrix, illustrated in Fig. 6. This figure described that the correct predicted values, given in the diagonals. The accuracy for the rest of the classifiers is also computed, as described in this table that shows that the average accuracy is above 88%. Moreover, the computational time of each classifier is also noted, as plotted in Fig. 7. This figure shows that the LDA classifier execution time (17.25 s) is minimum than the rest of the classifiers. Moreover, the

Bagged Tree classifier consumed higher time of 58.34 (s). Overall, the Fine KNN classifier shows the better recognition performance.

Table 2: Proposed action recognition framework results on UT-Interaction dataset

Classifiers	Time (s)	TP	FN	Recall	Accuracy (%)
Linear Discriminant	17.25	550.10	49.80	0.92	92.50
Linear SVM	31.77	553.30	46.70	0.92	93.30
Fine KNN	21.22	592.80	7.20	0.99	**98.90**
Fine Tree	23.84	338.30	261.70	0.56	59.30
Gaussian Naïve Bayes	17.49	455.80	144.20	0.76	77.40
Cubic SVM	37.73	592.30	17.70	0.97	97.30
Quadratic SVM	33.66	577.20	22.80	0.96	96.60
Medium Gaussian SVM	35.26	571.90	28.10	0.95	96.00
Weighted KNN	20.17	565.00	25.00	0.96	94.80
Bagged Tree	58.34	459.80	140.20	0.77	80.30



	handshaking-	hugging-	kicking	pointing	punching	pushing		
handshaking-	99.4%		0.2%		0.4%		99.4%	0.6%
hugging-	0.2%	98.9%	0.4%		0.2%	0.4%	98.9%	1.1%
kicking	0.4%	0.4%	98.3%		0.8%		98.3%	1.7%
pointing				100.0%			100.0%	
punching	0.3%		0.8%		97.5%	1.4%	97.5%	2.5%
pushing	0.3%	0.3%			0.7%	98.7%	98.7%	1.3%

Figure 6: Confusion matrix of proposed framework on UT Interaction dataset

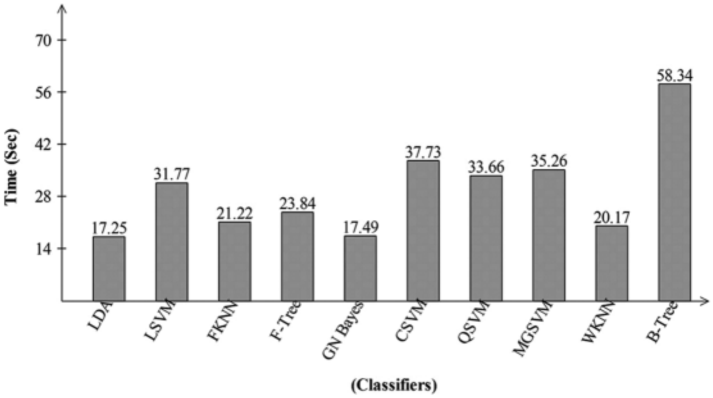


Figure 7: Computational time based comparison among selected classifiers on UT-Interaction dataset

UCF Sports Results: Tab. 3 presents the numerical results of proposed HAR framework on UCF Sports dataset. In this table, LDA classifier attained the maximum accuracy of 99.80% in 27.23 (s). The recall rate is also computed of this classifier that is 1.00. The performance of this classifier can be further verified through a confusion matrix, illustrated in Fig. 8. This figure described that the correct predicted values, given in the diagonals. The accuracy for the rest of the classifiers is also computed, as described in this table that shows that the average accuracy is above 90%. Moreover, the computational time of each classifier is also noted, as plotted in Fig. 9. This figure shows that the LDA classifier execution time (27.23 s) is minimum than the rest of the classifiers. Moreover, the MGSVM classifier consumed higher time of 516.16 (s). Overall, the LDA classifier shows the better recognition performance.

Table 3: Proposed action recognition framework results on UCF Sports dataset

Classifiers	Time (s)	TP	FN	Recall	Accuracy (%)
Linear Discriminant	27.23	1,296.00	4.00	1.00	**99.80**
Linear SVM	237.35	1,282.20	17.80	0.99	99.50
Fine KNN	195.75	995.30	304.70	0.77	83.30
Fine Tree	95.38	1,207.60	92.40	0.93	96.40
Gaussian Naïve Bayes	41.02	1,183.20	116.80	0.91	88.90
Cubic SVM	444.01	1,247.90	52.10	0.96	98.60
Quadratic SVM	370.86	1,279.00	21.00	0.98	99.40
Medium Gaussian SVM	516.16	1,101.10	198.90	0.85	92.00
Weighted KNN	191.96	591.30	708.70	0.45	50.10
Bagged Tree	264.48	1,237.90	62.10	0.95	97.90

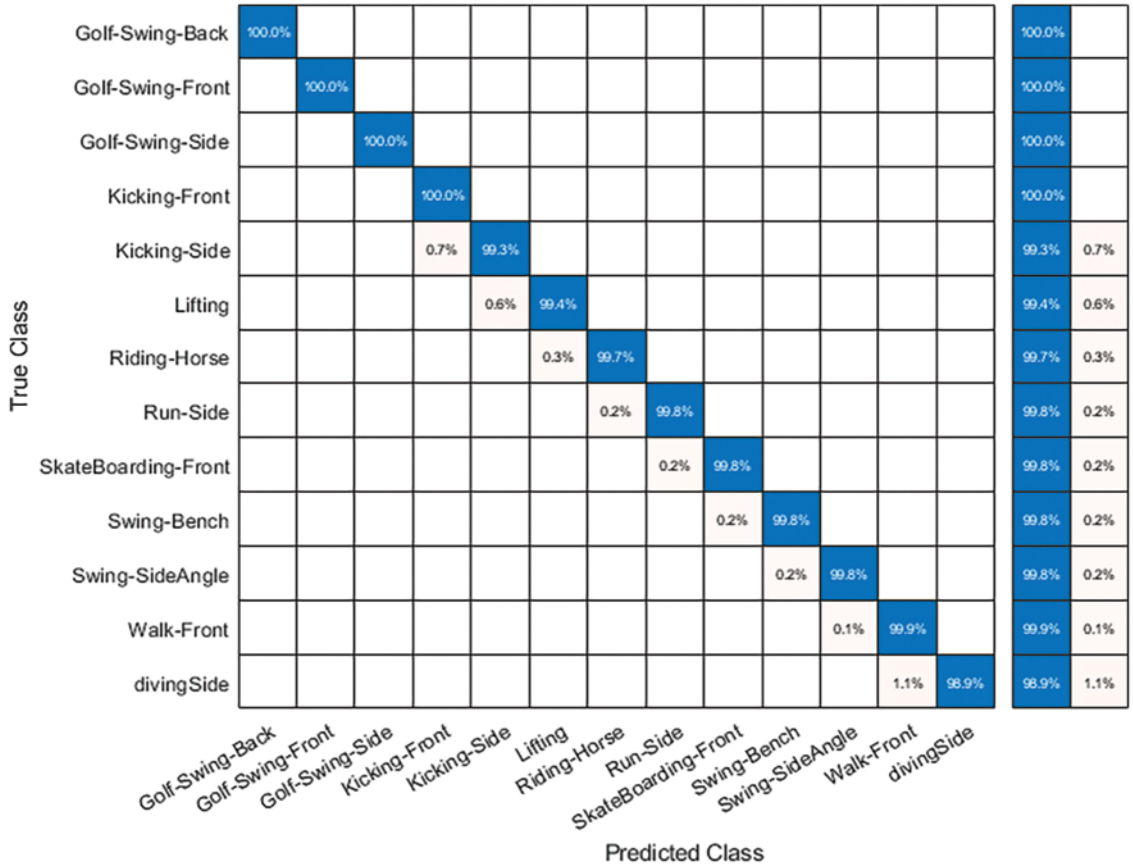


Figure 8: Confusion matrix of proposed framework on UCF Sports dataset

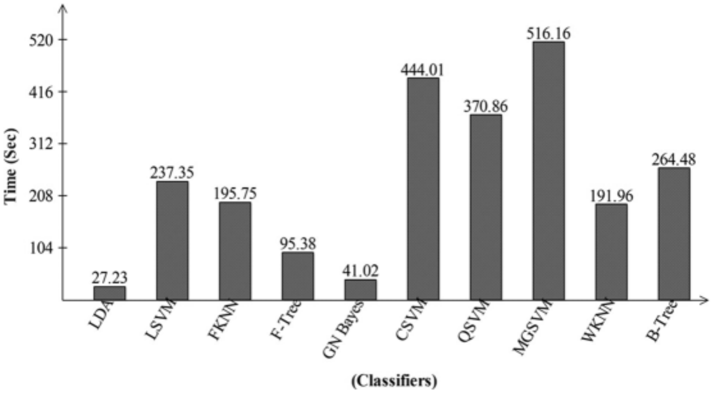


Figure 9: Computational time based comparison among selected classifiers on UCF Sports dataset

Hollywood Results: Tab. 4 presents the numerical results of proposed HAR framework on Hollywood dataset. In this table, LDA classifier attained the maximum accuracy of 99.60% in 30.93 (s). The recall rate is also computed of this classifier that is 0.99. The performance of this classifier can be further verified through a confusion matrix, illustrated in Fig. 10. This figure described that the correct predicted values, given in the diagonals. The accuracy for the rest of the classifiers is also computed, as described in this table that shows that the average accuracy is above 90%. Moreover, the computational time of each classifier is also noted, as plotted in Fig. 11. This figure shows that the LDA classifier execution time (30.93 s) is minimum than the rest of the classifiers. Moreover, the Cubic SVM classifier consumed higher time of 730.73 (s). Overall, the LDA classifier shows the better recognition performance.

Table 4: Proposed action recognition framework results on Hollywood dataset

Classifiers	Time (s)	TP	FN	Recall	Accuracy (%)
Linear Discriminant	30.93	793.30	6.70	0.99	**99.60**
Linear SVM	247.91	791.30	8.70	0.99	99.50
Fine KNN	274.32	778.60	21.40	0.97	98.10
Fine Tree	94.83	661.60	138.40	0.83	88.50
Gaussian Naïve Bayes	32.73	635.50	164.50	0.79	77.80
Cubic SVM	730.73	790.00	10.00	0.99	99.50
Quadratic SVM	609.73	792.80	7.20	0.99	99.60
Medium Gaussian SVM	602.87	771.30	28.70	0.96	98.20
Weighted KNN	195.92	551.90	248.10	0.69	73.80
Bagged Tree	219.22	721.60	78.40	0.90	93.90

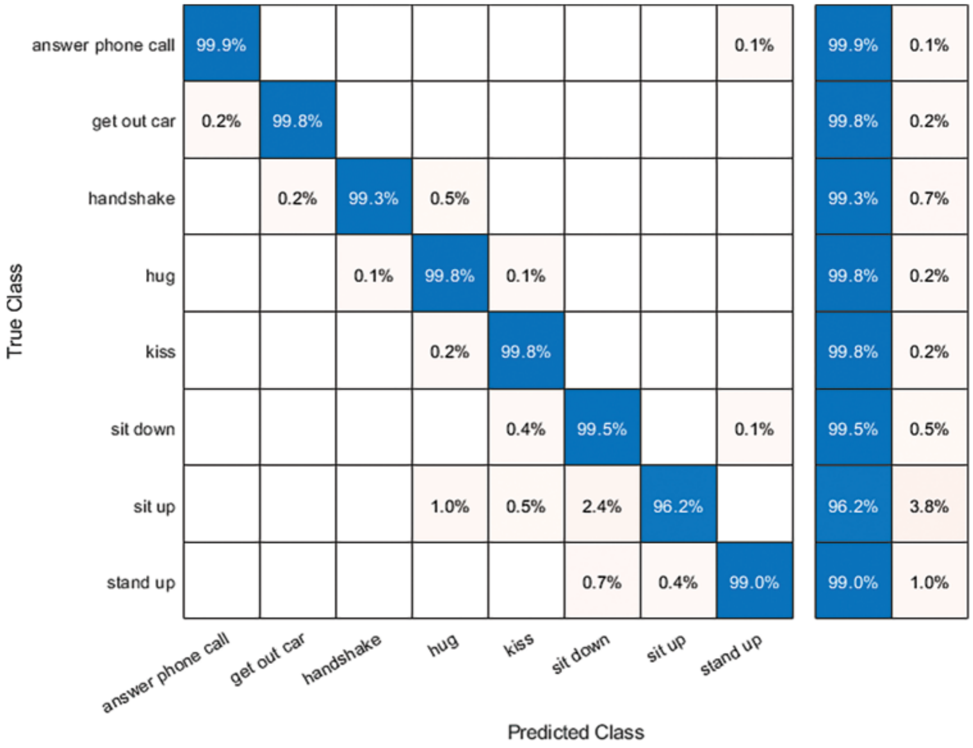


Figure 10: Confusion matrix of proposed framework on Hollywood dataset

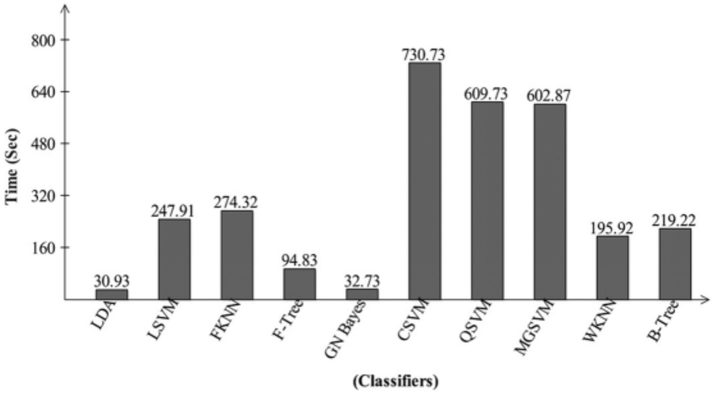
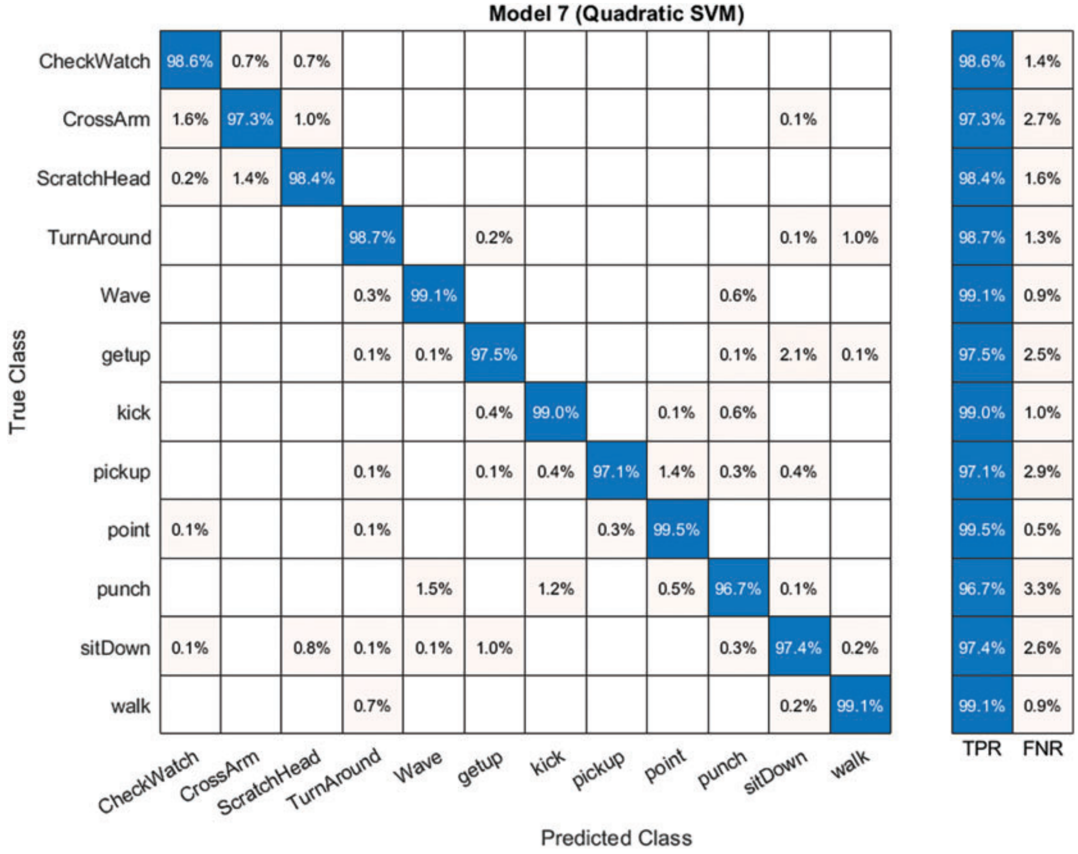


Figure 11: Computational time based comparison among selected classifiers on Hollywood dataset

IXMAS Results: Tab. 5 presents the numerical results of proposed HAR framework on IXMAS dataset. In this table, Cubic SVM classifier attained the maximum accuracy of 98.60% in 585.83 (s). The recall rate is also computed of this classifier that is 0.99. The performance of this classifier can be further verified through a confusion matrix, illustrated in Fig. 12. This figure described that the correct predicted values, given in the diagonals. The accuracy for the rest of the classifiers is also computed, as described in this table that shows that the average accuracy is above 86%. Moreover, the computational time of each classifier is also noted, as plotted in Fig. 13. This figure shows that the LDA classifier execution time (28.00 s) is minimum than the rest of the classifiers. Moreover, the MGSVM classifier consumed higher time of 767.46 (s). Overall, the Cubic SVM classifier shows the better recognition performance.

Table 5: Proposed action recognition framework results on IXMAS dataset

Classifiers	Time (s)	TP	FN	Recall	Accuracy (%)
Linear Discriminant	28.00	1,165.60	34.40	0.97	97.20
Linear SVM	267.06	1,172.30	27.70	0.98	97.80
Fine KNN	262.53	1,148.10	51.90	0.96	95.70
Fine Tree	135.04	742.90	457.10	0.62	64.20
Gaussian Naïve Bayes	37.55	811.90	388.10	0.68	67.90
Cubic SVM	585.83	1,182.30	17.70	0.99	**98.60**
Quadratic SVM	474.30	1,178.40	21.60	0.98	98.30
Medium Gaussian SVM	767.46	1,166.10	33.90	0.97	97.40
Weighted KNN	263.68	834.10	365.90	0.70	68.40
Bagged Tree	301.48	966.00	234.00	0.81	81.50



	CheckWatch	CrossArm	ScratchHead	TurnAround	Wave	getup	kick	pickup	point	punch	sitDown	walk	TPR	FNR
CheckWatch	98.6%	0.7%	0.7%										98.6%	1.4%
CrossArm	1.6%	97.3%	1.0%								0.1%		97.3%	2.7%
ScratchHead	0.2%	1.4%	98.4%										98.4%	1.6%
TurnAround				98.7%		0.2%					0.1%	1.0%	98.7%	1.3%
Wave				0.3%	99.1%					0.6%			99.1%	0.9%
getup				0.1%	0.1%	97.5%				0.1%	2.1%	0.1%	97.5%	2.5%
kick						0.4%	99.0%		0.1%	0.6%			99.0%	1.0%
pickup				0.1%		0.1%	0.4%	97.1%	1.4%	0.3%	0.4%		97.1%	2.9%
point	0.1%			0.1%				0.3%	99.5%				99.5%	0.5%
punch					1.5%		1.2%		0.5%	96.7%	0.1%		96.7%	3.3%
sitDown	0.1%		0.8%	0.1%	0.1%	1.0%				0.3%	97.4%	0.2%	97.4%	2.6%
walk				0.7%							0.2%	99.1%	99.1%	0.9%

Figure 12: Confusion matrix of proposed framework on IXMAS dataset

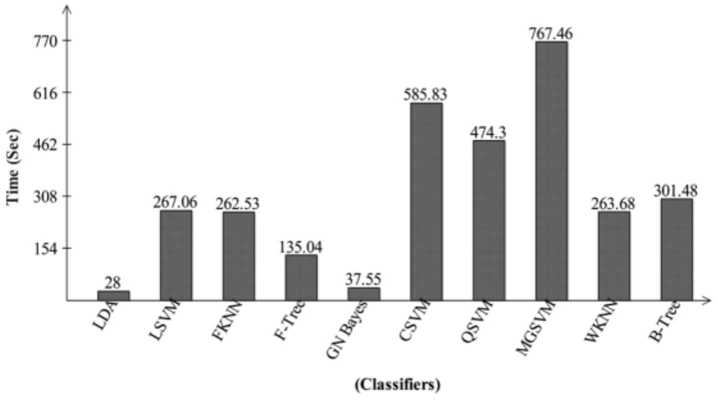


Figure 13: Computational time based comparison among selected classifiers on IXMAS dataset

UCF YouTube Results: Tab. 6 presents the numerical results of proposed HAR framework on UCF YouTube dataset. In this table, Cubic SVM classifier attained the maximum accuracy of 100% in 225.83 (s). The recall rate is also computed of this classifier that is 1.00. The performance of this classifier can be further verified through a confusion matrix, illustrated in Fig. 14. This figure described that the correct predicted values, given in the diagonals. The accuracy for the rest of the classifiers is also computed, as described in this table that shows that the average accuracy is above 95%. Moreover, the computational time of each classifier is also noted, as plotted in Fig. 15. This figure shows that the LDA classifier execution time (28.00 s) is minimum than the rest of the classifiers. Moreover, the Quadratic SVM classifier consumed higher time of 474.46 (s). Overall, the Cubic SVM classifier shows the better recognition performance.

Table 6: Proposed action recognition framework results on UCF YouTube dataset

Classifiers	Time (s)	TP	FN	Recall	Accuracy (%)
Linear Discriminant	28.00	997.00	3.00	1.00	99.70
Linear SVM	267.06	998.60	1.40	1.00	99.90
Fine KNN	262.53	999.90	0.10	1.00	100.00
Fine Tree	135.04	642.90	357.10	0.64	64.40
Gaussian Naïve Bayes	37.55	962.10	37.90	0.96	96.20
Cubic SVM	225.83	999.80	0.20	1.00	**100.00**
Quadratic SVM	474.30	999.80	0.20	1.00	100.00
Medium Gaussian SVM	467.46	999.70	0.30	1.00	100.00
Weighted KNN	263.68	999.30	0.70	1.00	99.90
Bagged Tree	301.48	957.40	42.60	0.96	95.70

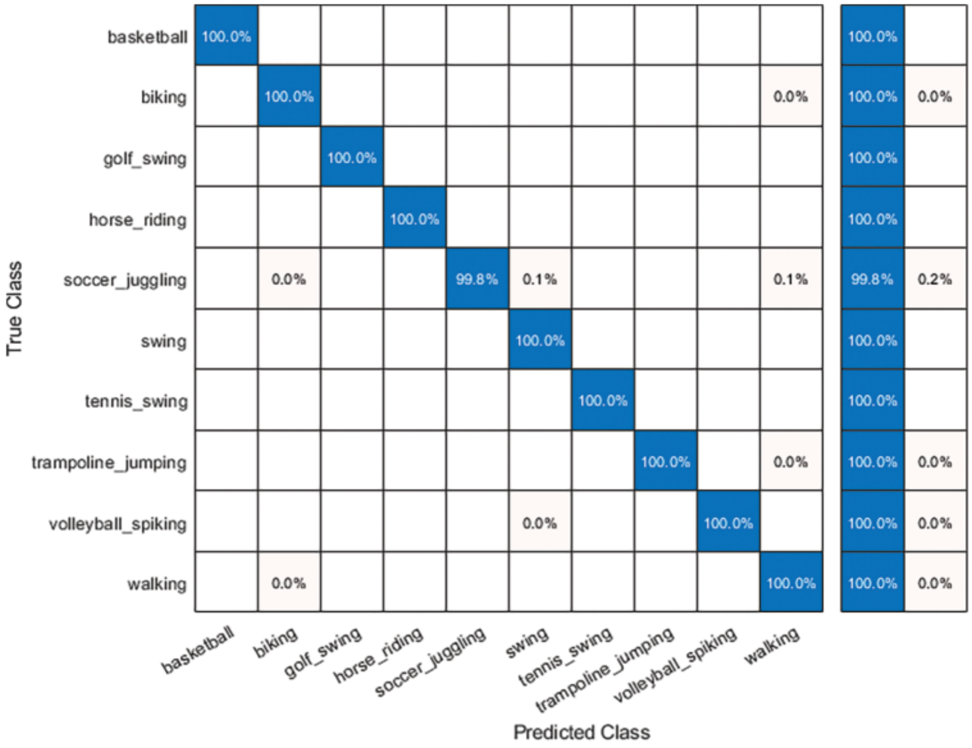


Figure 14: Confusion matrix of proposed framework on UCF YouTube dataset

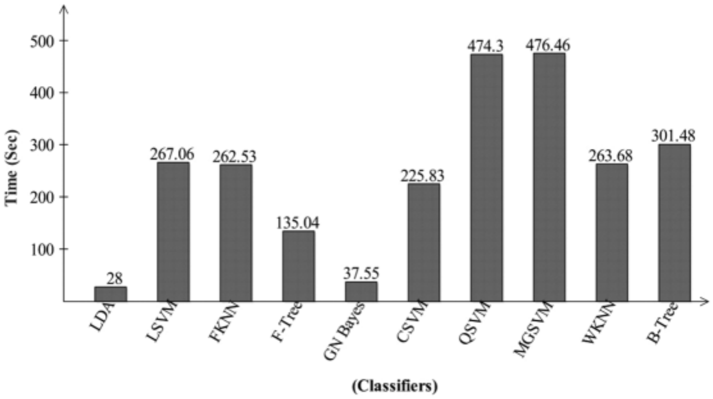


Figure 15: Computational time based comparison among selected classifiers on UCF YouTube dataset

3.2 Discussion and Comparison

Fig. 1 represents the detailed architecture of HAR using deep learning and optimization method. Experimentation has been carried out for 6 different datasets. For each dataset, ten different classifiers have been applied to ascertain performance in detail and results are displayed in the form of Tabs. 1–6, which include accuracy and execution time of all classifiers. For UCF sports and Hollywood datasets, LDA classifier turns out to be the best performing option giving accuracy above 99%, alongside fastest execution time (less than 31 s). Whereas for 3 datasets (KTH, IXMAS and UCF YouTube), Cubic SVM showed best accuracy which remained above 98%, however, execution time was best with LDA classifier (less than 28 s). KNN classifier was the best performing classifier for only 1 dataset (UT-Interaction), where accuracy of 98.9% was achieved but execution time remained least for LDA classifier (14.62 s). Analyzing overall performance, it is revealed that LDA is the best performing option for all datasets primarily in terms of execution time and accuracy, whereas, cubic SVM proves to be second best choice with a good accuracy but longer execution time. At the end, a comparison is conducted with state of the art (SOTA) techniques, given in Tab. 7.

Table 7: Comparison of proposed framework with SOTA techniques

Reference	Year	Dataset	Accuracy (%)
[36]	2021	UCF sports	96.8
[52]	2021	UCF sports	99.3
[53]	2021	UCF sports	92.67
Proposed	**-**	**UCF sports**	**99.8**
[37]	2021	UT-Interaction	96.7
[39]	2021	UT-Interaction	96.4
Proposed	**-**	**UT-Interaction**	**98.9**
[7]	2020	IXMAS	95.2
[54]	2019	IXMAS	88.05
Proposed	**-**	**IXMAS**	**98.6**
[12]	2021	Hollywood	99.2

(Continued)

Table 7: Continued

Reference	Year	Dataset	Accuracy (%)
Proposed	**-**	**Hollywood**	**99.6**
Proposed	**-**	**KTH**	**98.3**
Proposed		**UCF YouTube**	**100**

4 Conclusion

Human action recognition has been an active research topic in recent years, owing to recent advances in the fields of machine learning and deep learning. Computer vision researchers have introduced a number of techniques, with a focus on both classical and deep learning-based techniques. Due to similar actions and a large number of video sequences, traditional techniques did not perform well. We proposed a new framework in this paper that is based on the fusion of deep learning features and an improved PSO-based algorithm. The proposed framework was tested on six action datasets and found to be more accurate. Based on the results, we concluded that the fusion framework achieved higher accuracy, but the process lengthens the computational time. The long computational time is alleviated further by an improved PSO algorithm. The trapezoidal rule-based PSO algorithm will be modified and used for feature selection in the future. Furthermore, recent deep learning, LSTM, and reinforcement learning techniques will be considered for HAR in the future [55–57].

Acknowledgement: We are thankful to National Institute of Science and Technology for overall support.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] M. Sharif, T. Akram, M. Y. Javed, T. Saba and A. Rehman, "A framework of human detection and action recognition based on uniform segmentation and combination of euclidean distance and joint entropy-based features selection," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 3, pp. 1–18, 20172017.

[2] W. Sun, G. Dai, X. Zhang, X. He and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 1–8, 2021.

[3] T. Akram, M. Sharif, M. Y. Javed, N. Muhammad and M. Yasmin, "An implementation of optimized framework for action classification using multilayers neural network on selected fused features," *Pattern Analysis and Applications*, vol. 22, no. 11, pp. 1377–1397, 2019.

[4] M. Dzabraev, M. Kalashnikov, S. Komkov and A. Petiushko, "Mdmmt: Multidomain multimodal transformer for video retrieval," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 3354–3363, 2021.

[5] M. Sharif, T. Akram, M. Raza, T. Saba and A. Rehman, "Hand-crafted and deep convolutional neural network features fusion and selection strategy: An application to intelligent human action recognition," *Applied Soft Computing*, vol. 87, no. 2, pp. 105986, 2020.

[6] F. Afza, M. Sharif, S. Kadry, G. Manogaran and T. Saba, "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image and Vision Computing*, vol. 106, no. 10, pp. 104090, 2021.

[7] K. Javed, S. A. Khan, T. Saba, U. Habib and J. A. Khan, “Human action recognition using fusion of multiview and deep features: An application to video surveillance,” *Multimedia Tools and Applications*, vol. 4, no. 2, pp. 1–27, 2020.

[8] P. Pareek and A. Thakkar, “A survey on video-based human action recognition: Recent updates, datasets, challenges, and applications,” *Artificial Intelligence Review*, vol. 54, no. 21, pp. 2259–2322, 2021.

[9] Z. Ren, Q. Zhang, X. Gao, P. Hao and J. Cheng, “Multi-modality learning for human action recognition,” *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16185–16203, 2021.

[10] I. Azhar, M. Sharif, M. Raza and H. S. Yong, “A decision support system for face sketch synthesis using deep learning and artificial intelligence,” *Sensors*, vol. 21, no. 17, pp. 8178, 2021.

[11] H. Wang, B. Yu, K. Xia, J. Li and X. Zuo, “Skeleton edge motion networks for human action recognition,” *Neurocomputing*, vol. 423, no. 2, pp. 1–12, 2021.

[12] S. Khan, M. Alhaisoni, U. Tariq, H. S. Yong and A. Armghan, “Human action recognition: A paradigm of best deep learning features selection and serial based extended fusion,” *Sensors*, vol. 21, no. 1, pp. 7941, 2021.

[13] W. Sun, L. Dai, X. Zhang, P. Chang and X. He, “RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring,” *Applied Intelligence*, vol. 21, no. 7, pp. 1–16, 2021.

[14] W. Peng, J. Shi and G. Zhao, “Spatial temporal graph deconvolutional network for skeleton-based human action recognition,” *IEEE Signal Processing Letters*, vol. 28, no. 17, pp. 244–248, 2021.

[15] M. Sharif, F. Zahid, J. H. Shah and T. Akram, “Human action recognition: A framework of statistical weighted segmentation and rank correlation-based selection,” *Pattern Analysis and Applications*, vol. 23, no. 12, pp. 281–294, 2020.

[16] T. Akram, M. Sharif, N. Muhammad, M. Y. Javed and S. R. Naqvi, “Improved strategy for human action recognition; experiencing a cascaded design,” *IET Image Processing*, vol. 14, no. 17, pp. 818–829, 2019.

[17] S. Li, Q. Cao, L. Liu, K. Yang and S. Liu, “GroupFormer: Group activity recognition with clustered spatial-temporal transformer,” in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, NY, USA, pp. 13668–13677, 2021.

[18] L. Gan and F. Chen, “Human action recognition using APJ3D and random forests,” *Journal of Software*, vol. 8, no. 2, pp. 2238–2245, 2013.

[19] U. M. Nunes, D. R. Faria and P. Peixoto, “A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier,” *Pattern Recognition Letters*, vol. 99, no. 1, pp. 21–31, 2017.

[20] S. Siddiqui, K. Bashir, M. Sharif, F. Azam and M. Y. Javed, “Human action recognition: A construction of codebook by discriminative features selection approach,” *International Journal of Applied Pattern Recognition*, vol. 5, no. 17, pp. 206–228, 2018.

[21] F. Saleem, M. Alhaisoni, U. Tariq, A. Armghana and F. Alenezi, “Human gait recognition: A single stream optimal deep learning features fusion,” *Sensors*, vol. 21, no. 17, pp. 7584, 2021.

[22] M. Zahid, F. Azam, M. Sharif, S. Kadry and J. R. Mohanty, “Pedestrian identification using motion-controlled deep neural network in real-time visual surveillance,” *Soft Computing*, vol. 4, no. 2, pp. 1–17, 2021.

[23] Y. D. Zhang, M. Alhusseni, S. Kadry, S. H. Wang and T. Saba, “A fused heterogeneous deep neural network and robust feature selection framework for human actions recognition,” *Arabian Journal for Science and Engineering*, vol. 11, no. 2, pp. 1–16, 2021.

[24] K. Jabeen, M. Alhaisoni, U. Tariq, Y. D. Zhang and A. Hamza, “Breast cancer classification from ultrasound images using probability-based optimal deep learning feature fusion,” *Sensors*, vol. 22, no. 2, pp. 807, 2022.

[25] M. Alhaisoni, A. Armghan, F. Alenezi, U. Tariq and Y. Nam, “Video analytics framework for human action recognition,” *Computers, Material and Continua*, vol. 69, no. 1, pp. 1–15, 2021.

[26] S. Kiran, M. Y. Javed, M. Alhaisoni, U. Tariq and Y. Nam, “Multi-layered deep learning features fusion for human action recognition,” *Computers, Material and Continua*, vol. 71, no. 2, pp. 1–15, 2021.

[27] A. Sarkar, A. Banerjee, P. K. Singh and R. Sarkar, "3D human action recognition: Through the eyes of researchers," *Expert Systems with Applications*, vol. 4, no. 1, pp. 116424, 2022.

[28] L. Shi, Y. Zhang, J. Cheng and H. Lu, "Action recognition via pose-based graph convolutional networks with intermediate dense supervision," *Pattern Recognition*, vol. 121, no. 17, pp. 108170, 2022.

[29] G. Zhang, G. Huang, H. Chen, C. M. Pun and W. K. Ling, "Video action recognition with key-detail motion capturing based on motion spectrum analysis and multiscale feature fusion," *The Visual Computer*, vol. 21, no. 4, pp. 1–18, 2022.

[30] Y. A. Andrade-Ambriz, S. Ledesma, M. A. Ibarra-Manzano, M. I. Oros-Flores and D. L. Almanza-Ojeda, "Human activity recognition using temporal convolutional neural network architecture," *Expert Systems with Applications*, vol. 191, no. 1, pp. 116287, 2022.

[31] X. Shen and Y. Ding, "Human skeleton representation for 3D action recognition based on complex network coding and LSTM," *Journal of Visual Communication and Image Representation*, vol. 82, no. 14, pp. 103386, 2022.

[32] Y. Xie, Y. Zhang and F. Ren, "Temporal enhanced graph convolution network for skeleton-based action recognition," *IET Computer Vision*, vol. 4, no. 1, pp. 1–21, 2022.

[33] L. Zhang, C. P. Lim and Y. Yu, "Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization," *Knowledge-Based Systems*, vol. 220, no. 5, pp. 1–20, 2021.

[34] S. Nazir, M. H. Yousaf, J. C. Nebel and S. A. Velastin, "Dynamic spatio-temporal bag of expressions model for human action recognition," *Sensors*, vol. 19, no. 2, pp. 1–15, 2019.

[35] S. Rahimi, A. Aghagolzadeh and M. Ezoji, "Human action recognition based on the Grassmann multi-graph embedding," *Signal, Image and Video Processing*, vol. 13, no. 2, pp. 271–279, 2018.

[36] B. S. Kumar, S. V. Raju and H. V. Reddy, "Human action recognition using a novel deep learning approach," *Materials Science and Engineering*, vol. 1042, no. 4, pp. 1–21, 2021.

[37] S. Kiran, M. Younus Javed, M. Alhaisoni, U. Tariq, Y. Nam *et al.,* "Multi-layered deep learning features fusion for human action recognition," *Computers, Materials & Continua*, vol. 69, no. 2, pp. 4061–4075, 2021.

[38] J. Li, Y. Han, M. Zhang, G. Li and B. Zhang, "Multi-scale residual network model combined with Global Average Pooling for action recognition," *Multimedia Tools and Applications*, vol. 21, no. 6, pp. 1–31, 2021.

[39] W. Ahmed, M. H. Yousaf, A. Yasin and M. Maqsood, "Robust suspicious action recognition approach using pose descriptor," *Mathematical Problems in Engineering*, vol. 2021, no. 9, pp. 1–12, 2021.

[40] I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld, "Learning realistic human actions from movies," in *2008 IEEE Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 1–8, 2020.

[41] M. Mahmood, A. Jalal and M. Sidduqi, "Robust spatio-temporal features for human interaction recognition via artificial neural network," in *2018 Int. Conf. on Frontiers of Information Technology*, Islamabad, Paksitan, pp. 218–223, 2018.

[42] K. Soomro and A. R. Zamir, "Action recognition in realistic sports videos," *Computer vision in Sports*, vol. 4, no. 1, pp. 181–208, 2018.

[43] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conf. on Computer Vision*, London, UK, pp. 510–526, 2016.

[44] A. Nadeem, A. Jalal and K. Kim, "Accurate physical activity recognition using multidimensional features and Markov model for smart health fitness," *Symmetry*, vol. 12, no. 3, pp. 1766, 2020.

[45] M. Arshad, U. Tariq, A. Armghan, F. Alenezi and M. Younus Javed, "A Computer-aided diagnosis system using deep learning for multiclass skin lesion classification," *Computational Intelligence and Neuroscience*, vol. 21, no. 7, pp. 1–23, 2021.

[46] S. Baek, J. Jeon, B. Jeong and Y. S. Jeong, "Two-stage hybrid malware detection using deep learning," *Human-Centric Computing And Information Sciences*, vol. 11, no. 13, pp. 2021, 2021.

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 4510–4520, 2018.

[48] M. Shahud, J. Bajracharya, P. Praneetpolgrang and S. Petcharee, "Thai traffic sign detection and recognition using convolutional neural networks," in *2018 22nd Int. Computer Science and Engineering Conf.*, Toranto, Canada, pp. 1–5, 2018.

[49] A. Adeel, M. Sharif, F. Azam, J. H. Shah and T. Umer, "Diagnosis and recognition of grape leaf diseases: An automated system based on a novel saliency approach and canonical correlation analysis based multiple features fusion," *Sustainable Computing: Informatics and Systems*, vol. 24, no. 11, pp. 100349, 2019.

[50] D. Wang, D. Tan and L. Liu, "Particle swarm optimization algorithm: An overview," *Soft Computing*, vol. 22, no. 4, pp. 387–408, 2018.

[51] G. I. Sayed, A. E. Hassanien and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Computing and Applications*, vol. 31, no. 14, pp. 171–188, 2019.

[52] F. Afza, M. Sharif, S. Kadry, G. Manogaran and T. Saba, "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image and Vision Computing*, vol. 106, no. 21, pp. 1–26, 2021.

[53] A. Abdelbaky and S. Aly, "Human action recognition using three orthogonal planes with unsupervised deep convolutional neural network," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 20019–20043, 2021.

[54] V. A. Chenarlogh and F. Razzazi, "Multi-stream 3D CNN structure for human action recognition trained by limited data," *IET Computer Vision*, vol. 13, no. 17, pp. 338–344, 2019.

[55] I. A. Khan, N. Moustafa, I. Razzak and M. Tanveer, "XSRU-IoMT: Explainable simple recurrent units for threat detection in Internet of Medical Things networks," *Future Generation Computer Systems*, vol. 127, no. 4, pp. 181–193, 2022.

[56] S. A. Khan, S. Hussain and S. Yang, "Contrast enhancement of low-contrast medical images using modified contrast limited adaptive histogram equalization," *Journal of Medical Imaging and Health Informatics*, vol. 10, no. 5, pp. 1795–1803, 2020.

[57] S. Yakhchi, A. Behehsti, Sm Ghafari, I. Razzak and M. Orgun, "A convolutional attention network for unifying general and sequential recommenders," *Information Processing & Management*, vol. 59, no. 2, pp. 102755, 2022.