Computers, Materials & Continua
DOI: 10.32604/cmc.2023.034563
Article

Tech Science Press

HRNetO: Human Action Recognition Using Unified Deep Features Optimization Framework

Tehseen Ahsan[1,*], Sohail Khalid[1], Shaheryar Najam[1], Muhammad Attique Khan[2], Ye Jin Kim[3] and Byoungchol Chang[4]

[1]Department of Electrical Engineering, Riphah International University, Peshawar Rd, near Hajj Complex, I-14, Islamabad, Islamabad Capital Territory, 46000, Pakistan
[2]Departmnt of Computer Science, HITEC University, Taxila, 47080, Pakistan
[3]Department of Computer Science, Hanyang University, Seoul, 04763, Korea
[4]Center for Computational Social Science, Hanyang University, Seoul, 04763, Korea
*Corresponding Author: Tehseen Ahsan. Email: ahsantehseen69@gmail.com
Received: 20 July 2022; Accepted: 08 December 2022

Abstract: Human action recognition (HAR) attempts to understand a subject's behavior and assign a label to each action performed. It is more appealing because it has a wide range of applications in computer vision, such as video surveillance and smart cities. Many attempts have been made in the literature to develop an effective and robust framework for HAR. Still, the process remains difficult and may result in reduced accuracy due to several challenges, such as similarity among actions, extraction of essential features, and reduction of irrelevant features. In this work, we proposed an end-to-end framework using deep learning and an improved tree seed optimization algorithm for accurate HAR. The proposed design consists of a few significant steps. In the first step, frame preprocessing is performed. In the second step, two pre-trained deep learning models are fine-tuned and trained through deep transfer learning using preprocessed video frames. In the next step, deep learning features of both fine-tuned models are fused using a new Parallel Standard Deviation Padding Max Value approach. The fused features are further optimized using an improved tree seed algorithm, and select the best features are finally classified by using the machine learning classifiers. The experiment was carried out on five publicly available datasets, including UT-Interaction, Weizmann, KTH, Hollywood, and IXAMS, and achieved higher accuracy than previous techniques.

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

1 Introduction

Human action recognition (HAR) is an extensive research topic in the field of computer vision (CV) [1,2]. HAR applications include e-health, human-computer interaction (HCI), visual information understanding, and video surveillance [3]. Out of these applications, the most critical

application is video surveillance [4]. Human action recognition is mainly used to reduce crime rates and security purposes [5]. The primary reason for expanding HAR research is intelligent city cameras for surveillance [6]. In addition, human action recognition is critical in visual administration for detecting human activities in public places [7]. There are different types of human action; these actions are categorized into two classes: involuntary and voluntary [8].

The CV techniques are introduced in the literature for automated HAR in video sequences [9]. Manually processing video frames is difficult and takes more time [10,11]. Primarily the classical techniques are based on the point, texture, shape, and geometric features [12]. A couple of procedures depend on the temporal data of the human; before the extraction of features, few extract the silhouette features of humans [13], so the problems for traditional methods are not suitable for complex activities for better accuracy [14]. Traditional CV methods for feature extraction are less efficient and slower due to the training data and extracted features compared to new deep learning-based techniques [15]. These techniques include Shi-Tomasi Corner Detection (STCD), Harris Corner Detection (HCD), Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Rotated BRIEF (ORB), and Oriented fast [8,16,17].

The convolutional neural network (CNN) learns about the features directly. The first layer is used to extract deep local features, and the last layer, known as fully connected, is used to extract global features. Most studies show that using different pre-trained CNN models consumes less computational time during the training phase while achieving high accuracy in final image classification and video summarization. In [10], the author utilizes the transfer learning concept and trains seven pre-trained CNN models. The features are extracted using these pre-trained models by implementing the idea of transfer learning.

The researchers designed multiple deep learning techniques for HAR in video sequences as deep learning models became more prevalent [18,19]. Deep learning demonstrated capable results in CV for a variety of tasks, such as action recognition [20,21], gait recognition [22,23], medical [24,25], and a few more [10]. Data representation and learning are made by deep learning at different levels by creating new models [26] by simulating the human brain processing. These models use various preprocessing layers such as the convolutional layer, ReLu layer, pooling layer, F.C. layer, and Softmax layer [27]. Supervised learning, unsupervised learning, hierarchical probabilistic models, and neural network are different models in deep learning [28]. Many HAR deep learning techniques are introduced in the literature. However, there are still several challenges that reduce recognition accuracy. In this work, we effectively proposed an end-to-end deep learning framework for HAR. Our significant contributions are as follows:

- Performed a frames preprocessing step and trained two fine-tuned deep learning networks through deep transfer learning.
- Proposed a new Parallel Standard Deviation Padding Max Value (PSPMV) approach for deep features fusion.
- Proposed an improved tree seed optimization algorithm for best feature selection.

2 Related Work

Human action recognition (HAR) is categorized into two types: (i) machine learning algorithms have been used for the recognition; (ii) hand-crafted features are carried out for the recognition [29]. In [30] to represent relevant activities such as walking, running, and jogging, the Global and Local Zernike Moment (GLZM) is used as a Bag-of-Features (BoF). Furthermore, using the General Linear Model, global features are extracted and fused (GLM). The global features represent the

region of the human body where the activities are performed, whereas the local features represent the activity information. The Whitening transformation was used to preprocess fused features. Finally, the final classification is performed using a multiclass support vector machine (SVM). In [31] skeleton-based action recognition scheme is presented using the hierarchical recurrent CNN method. The skeletons are divided into five parts according to physical layouts. The Berkeley mhad, MSRAction3D, and HDM05 are the datasets utilized for experimental results and achieved an accuracy of 100%, 94.49%, and 96.92%, respectively. In [32], convolutional neural network (CNN) based sequential connections are presented with the most extended shortest memory (LSTM) network. The deep fusion framework efficiently exploits the spatial features of CNN models and the temporal features from the longest shortest memory (LSTM). The UCF-Sports, UCF11, and HMDB datasets are utilized for the evaluation and achieved accuracies of 99.1%, 94.6%, and 69.0%, respectively.

In [33], the authors presented a hybrid technique for HAR called HAREDNet. The proposed approach is based on a few essential steps such as (i) The Encoder-Decoder Network (EDNet) is utilized for deep features extraction; (ii) The iSIFT is utilized for the local feature extraction, Local Maximal Occurrence (LOMO), and improved Gabor (iGabor); (iii) the feature redundancy is reduced by using the Cross view Quadratic Discriminant Analysis (CvQDA) and (iv) features are fused using weighted fusion strategy. The presented technique is validated on several datasets such as UCF-101, NTU RGB+D, and HMDB51 with accuracies of 97.48%, 97.45%, and 80.58%. In [10], the authors presented a feature mapping, fusion, and selection method for HAR. Pre_trained CNN models: InceptionV3 and DenseNet201 are fine-tuned and extract features in the first step. The serial-based extended (SbE) fusion method is applied to combine the features. The Kurtosis-controlled Weighted k-nearest neighbor (KNN) is used to select the best features in the third step. Finally, supervised learning methods are used to classify the selected features. Three different datasets were used in the experiments: Hollywood, WVU, KTH, and IXMAS, and they achieved accuracies of 99.8%, 99.3%, 97.4%, and 99.9%, respectively. In [34], the authors presented deep learning and the Spatio-temporal technique for HAR. The training model uses the transfer learning technique to extract deep features. A decoder-encoder (DAE) process is used to learn spatial relationships, and a recurrent neural network (RNN) with an LSTM framework is used. In [8], the authors presented a SNSP method for HAR. The features are extracted by standard normal, slope, and parameter space. Several experiments are performed on KARD-Kinect Activity Recognition Dataset, UTD Multimodal Human Action Dataset, and SBU Kinect Interaction Dataset, achieving improved accuracy. The recent studies above focused on deep learning features and classification using different machine learning classifiers. The gaps in the past studies are the number of features passed to the classifiers for improved accuracy and the presence of redundant features. This article's proposed framework for HAR is a deep unified learning and an improved optimization algorithm.

3 Proposed Method

The proposed HAR framework is presented in Fig. 1 which represents four core steps: video frames preprocessing, deep learning models training and features extraction, features fusion using a serially extended approach, and finally, best features selection and classification. The main steps of the proposed framework are the fusion of the extracted features and the selection of the best features using an improved tree seed algorithm. The detail of this entire framework is given below subsection.

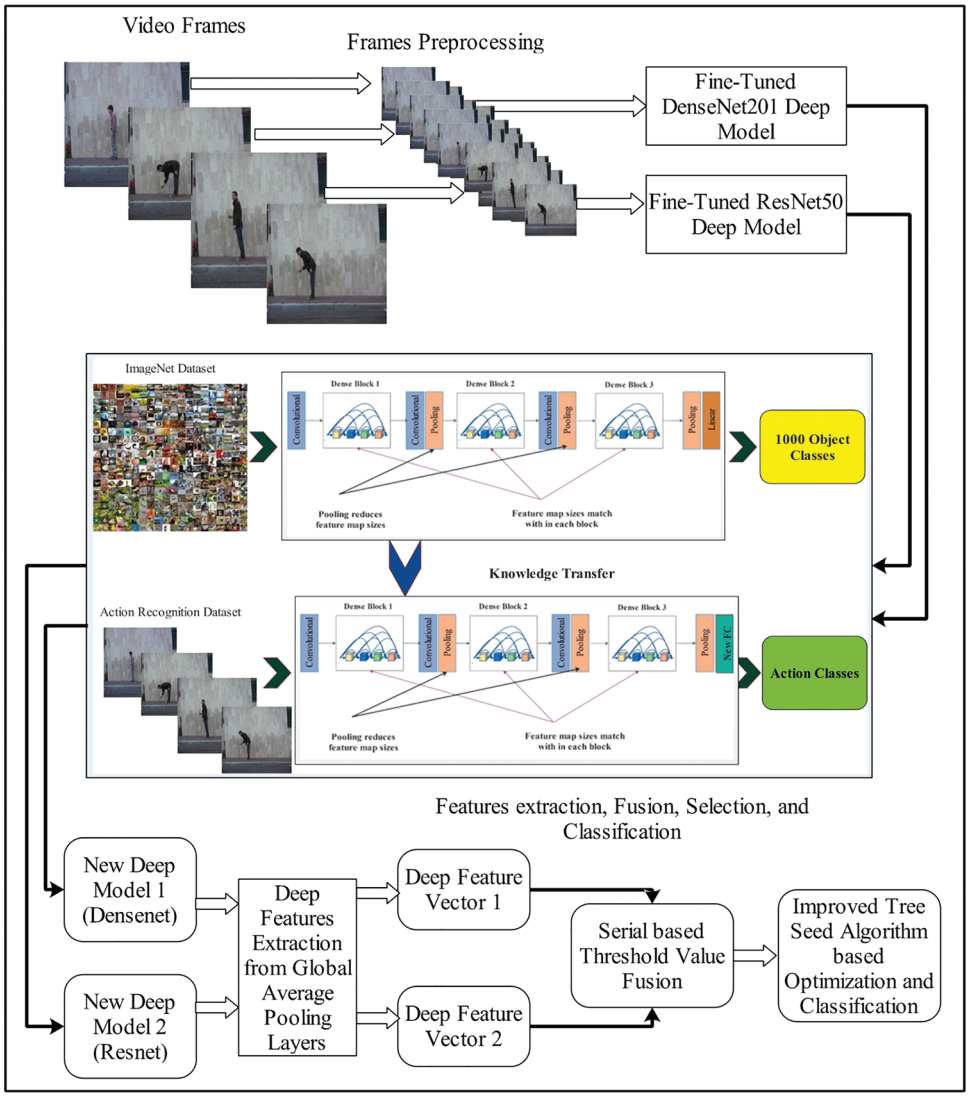

Figure 1: Proposed deep learning and features optimization based architecture for HAR

3.1 Video Frames Preprocessing

Preprocessing is an essential step in image processing because it allows you to normalize pixel values, resize images, and improve local contrast. In this work, we performed preprocessing step to resize the video frames in dimensions $256 \times 256 \times 3$. Initially, the size of each video frame was $512 \times 512 \times 3$. These normalized frames are used in the next step, named transfer learning-based training of CNN models.

3.2 Deep Transfer Learning

Transfer learning is reusing previously trained deep learning models for a new task. As shown in Fig. 2, a pre-trained CNN model was initially trained on the ImageNet dataset. This dataset contains 1000 object classes, but in the HAR task, the number of categories is different for each selected

dataset; therefore, we utilized the fine-tuned deep models (description is given below) and trained through transfer learning without freezing the weight of any hidden layer. Through this process, a newly introduced deep model is obtained that we finally utilize for the deep features extraction.

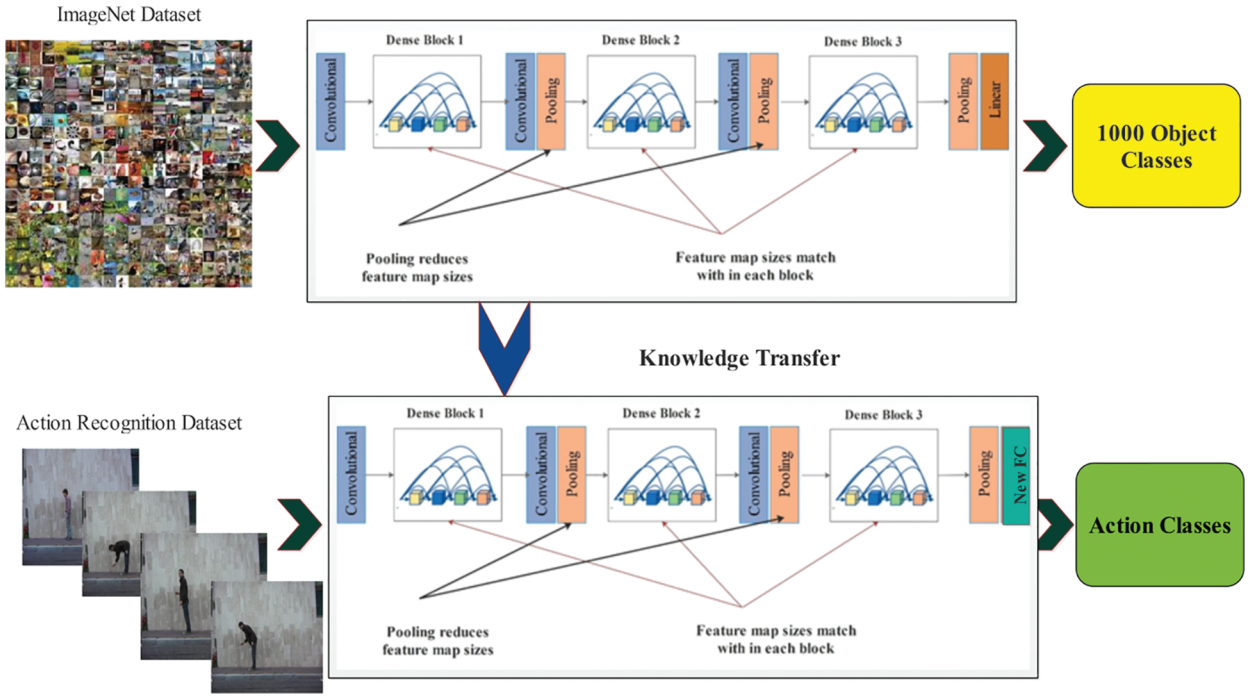

Figure 2: Workflow of deep transfer learning for training new deep model for HAR

3.3 Convolutional Neural Network

In the image classification task, CNN is popular due to its improved performance than traditional pattern recognition techniques. The convolutional, pooling and fully connected are the hidden layers used to extract the temporal and spatial features with the help of a filter applied to these layers [30,31]. A simple CNN architecture is the feed-forward artificial neural network (ANN) that is based on the three building blocks such as features are learned by the convolutional layer, the dimensions and the computational time are reduced through max-pooling (subsampling) layer, and classifications are done by the fully connected layer [35]. An architecture of a simple CNN model for the classification of images is shown in Fig. 3.

Convolutional Layer: Local features are extracted through the convolutional layer by finding the local connection among the sample of data coming from the input layer.

$$FV^{L} = \left(input_{x*x} + weight_{x*x}\right) + A \tag{1}$$

Here, $input_{x*x}$ represent the input data, the $weight_{x*x}$, represent the weight vector, x represents the filters and kernel size, and A represents the bias. Feature vectors are obtained by adding the pixels together and convolving the filter on the pixel of the images. The features of this layer are then passed to the activation layer called ReLu to resolve the non-linearity among features.

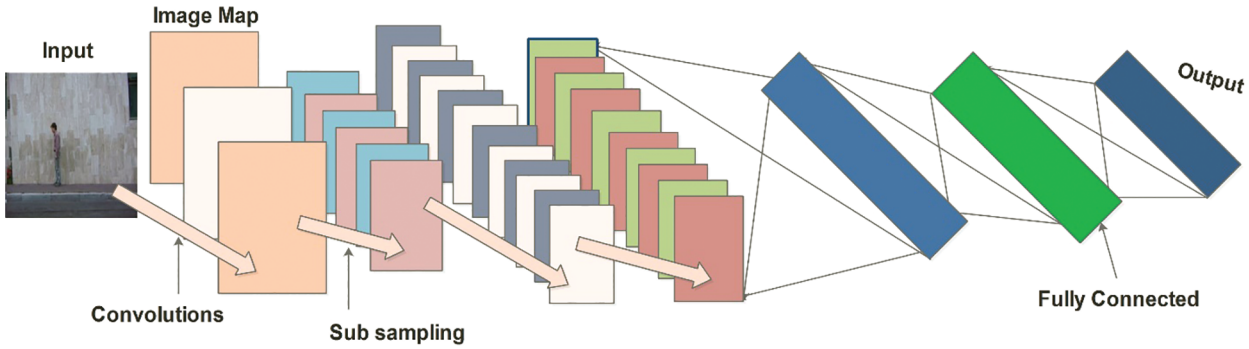

Figure 3: A simple architecture of CNN for HAR

Activation Layer: The ReLu activation layer assigns zero to the nonnegative values obtained from the convolutional layer. The main advantage of this activation layer is faster computational time than the other activation functions, such as leaky ReLu. Mathematically, this layer is described as follows:

$$\delta(R) = \max(0, R) \tag{2}$$

where R represents the element in the input vector.

Max-Pooling Layer: The max-pooling layer divided the feature map into small non-overlapping pooling kernels. It considers the maximum values of every kernel and passes them to the next layer. The max-pooling layer performed two main steps: (1) the data obtained from the previous layer is down-sampled and reduces the dimensions of the data, and (2) improvement in the model parameters for generalizability and less computational time.

Fully Connected Layer: The FC layer performed logical inference, and it converted the 3-D matrix into the 1-D vector by using fully convolutional operations. Mathematically, this layer is defined as:

$$Y_{z_o X1} = weight_{z_o X z_j}.X_{z_j X1}.A_{z_o X1} \tag{3}$$

where the input and output vector sizes are represented by Z_o and Z_j and Y is the output of FC layer.

Softmax Layer: The layer is utilized as a classification layer in the architecture of a CNN and used to find the probability of normalized classes $b(x^{(j)} = m^j|y^{(j)}; X)$.

$$\left(x^{(j)} = m^j|y^{(j)}; X\right) = \begin{bmatrix} (x^{(j)} = 1|y^{(j)}; X \\ (x^{(j)} = 2|y^{(j)}; X \\ . \\ . \\ . \\ (x^{(j)} = m|y^{(j)}; X \end{bmatrix} = \frac{1}{\sum_{i=1}^{m} v^{X_i^z} y^{(j)}} \begin{bmatrix} v^{X_1^z} y^{(j)} \\ v^{X_2^z} y^{(j)} \\ . \\ . \\ . \\ v^{X_m^z} y^{(j)} \end{bmatrix} \tag{4}$$

Here, m is the number of samples, $j = 1 \ldots m$ represents the weights which are replaced by X, and the input for the classifier is $v^{X_m^z} y^{(j)}$.

3.4 Deep Learning Features Extraction

Fine-Tuned DensNet201 CNN Model: The Dense convolutional network, also called DenseNet, has fewer parameters than several other pre-trained CNN models such as VGG19, VGG16, and many more. This network did not learn the redundant features map [36]. All layers in this network are in narrow styles, such as 12 filters, which add fewer sets of the new feature map. Every layer in this network

can directly access the hidden layers. As a result, the computational cost is reduced and made better for image classification. In this work, we fine-tuned this network according to the output classes of selected action recognition datasets. The last fully connected layer is removed, and added a new layer. As illustrated in Fig. 4, the fine-tuned model is trained on the action recognition dataset through T.L. concept. During the training process, several hyperparameters have been utilized, like a learning rate of 0.005, the mini-batch size of 16, epochs of 100, and the optimizer are Adam. After training the fine-tuned DenseNet201 model, the average pooling layer is utilized to extract deep features.

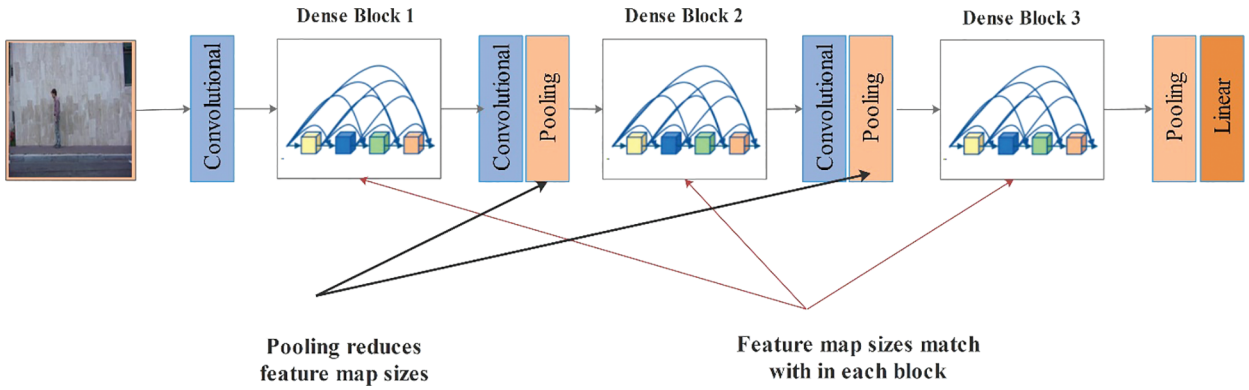

Figure 4: Architecture of DensNet-201pre-trained deep model

Fine-Tuned ResNet50: The residual network is also called (ResNet). This model introduced the residual connections between the layers, which helps preserve the knowledge gain, reduce the loss, and boost the performance during the training process. The ResNet50 model is trained on the 1000-class ImageNet dataset. We fine-tuned this model for HAR in this work. First, the fine-tuning new fully connected layer is added by removing the original fully connected layer. After that built the connections and trained the fine-tuned model through TL. The fine-tuned model is illustrated in Fig. 5. During the training process, several hyper parameters have been utilized mini-batch size is 16, the learning rate of 0.005, epochs are 100, the loss function is cross-entropy, and optimizer is stochastic gradient descent (SGD). Later, the average pooling layer is selected, and the activation function is applied for feature extraction. In the next step, extracted features of both networks are fused in a single vector for better information on subject actions.

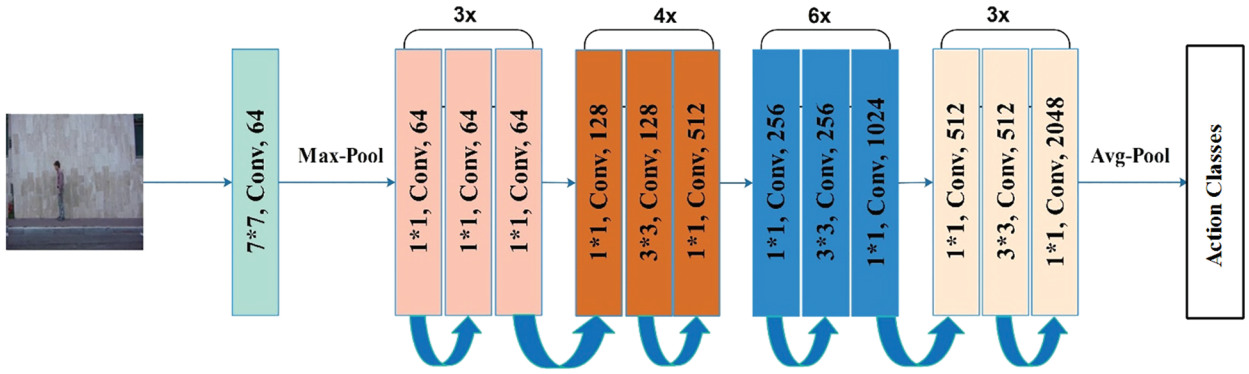

Figure 5: The main Architecture of the pre-trained deep model of ResNet-50

3.5 Features Fusion and Optimization

Features Fusion: In this work, we proposed a new features fusion approach named Parallel Standard Deviation Padding Max Value (PSPMV). Consider we have two extracted deep feature vectors of dimension $N \times K_1$ and $N \times K_2$ denoted by ϕ_1 and ϕ_2, respectively. Suppose the fused feature vector is denoted by ϕ_3 having dimension $N \times K_3$. The proposed fusion approach work in three core steps: (1) length finding, (2) equivalent dimension, and (3) maximum selection for fusion. In the first step, maximum and minimum dimensional feature vectors are defined as follows:

$$\phi_{max} = max\left(\phi_1, \phi_2\right) \tag{5}$$

$$\phi_{min} = min\left(\phi_1, \phi_2\right) \tag{6}$$

Based on the above equations, we consider ϕ_{max} feature vector and find the standard deviation value for the equivalent length of ϕ_{min}. The standard deviation value is utilized for padding instead of zero padding. Mathematically, the standard deviation of ϕ_{max} is computed as follows:

$$\sigma = \sqrt{\frac{\sum\left(\phi_{max}\left(i\right) - \overline{\phi_{max}}\right)}{n-1}} \tag{7}$$

After padding, the features of both vectors ϕ_1 and $\widetilde{\phi_2}$ are fused based on the following criteria, where $\widetilde{\phi_2}$ is updated vector after padding operation:

$$Fused = \left\{\phi_3\left\{k\right\} for\ Max\left(\phi_1(\mathrm{i}) \text{ and } \widetilde{\phi_2}(j)\right)\right\} \tag{8}$$

The resultant fused vector $\phi_3\{k\}$ is further optimized through an improved tree seed optimization algorithm. This step's main objective is to select important features and reduce the classification time.

Improved Tree Seed Algorithm-based Selection: The tree search algorithm is a natural phenomenon that deals with the relationship between the tree and its seeds [37]. It is the natural process by which the trees are spread on the surface by their seeds; over time, these seeds turn to the trees. If we assume the optimization problem's search space, the surface of a tree, then we can take the location of the seeds and trees to be the 'problem's solution. The location of the source is the most important optimization problem because this process strongly impacts the search. This search process is based on two mathematical equations. In the first one, the best location of tree population and the location of a tree which produces seed for the tree are considered to improve the local search. Hence, the new seed can be produced for a tree as follows:

$$M_{ab} = N_{ab} + \beta_{ab} \times \left(O_b - N_{ab}\right) \tag{9}$$

$$M_{ab} = N_{ab} + \beta_{ab} \times \left(N_{ab} - N_{cb}\right) \tag{10}$$

Here, the M_{ab} is the b^{th} dimension of the a^{th} seed that produces an a^{th} tree, N_{ab} is the b^{th} dimension of the a^{th} tree, O_b is a b^{th} dimension obtain for the best tree position so far, N_{cb} is a b^{th} dimensions of the c^{th} randomly tree is selected in the range of [1,1], a and c present different indices. In both equations above, the most significant thing is selection of new seed location (produced). In order to control the selection in the range of [0,1], search tendency (ST) has opted. If the ST has a higher mean, it indicates a strong local search and speedy convergence; however, if the ST value is lower than the mean, it indicates slow convergence but a strong global search. The ST parameters control the capability of exploitation and exploration of TSA. Therefore, the optimization problem using the initial position of the seed is defined as follows:

$$N_{ab} = P_{b,min} + c_{a,b} \times (D_{b,max} - P_{b,min}) \tag{11}$$

Here, $P_{b,min}$ is a search space of lower bound, $D_{b,max}$ is a search space of higher bound, $c_{a,b}$ are the random number produced for each dimension and location in the range of [0,1].

$$O = \min \{f\,(N_a)\}\, a = 1, 2, 3, \ldots\ldots\ldots N \tag{12}$$

where O shows the best solution from the selected population and N represents the total number of trees in the population. The resultant best-selected features of the tree search algorithm (TSA) are utilized in a maximum function to select the most optimum points as follows:

$$MO = max(O(k)), \quad k = 1, 2, 3, \ldots N \tag{13}$$

The selected max optimum features are passed in the fitness function and repeat the above step until the error is reduced to the minimum. Then, the selected features are passed to the supervised learning classifier for the final classification.

4 Results and Discussion

This section discusses the proposed HAR framework's experimental process with detailed numerical values and visual plots. The framework is tested on five publicly available datasets such as KTH, UT-Interaction, Weizmann, Hollywood, and IXMAS. Each dataset is divided into 50:50, which represent 50% of frames utilized for training and the rest 50% for testing. The K-Fold cross-validation is utilized for the whole testing process. Several classifiers are selected for the classification results, and the performance of every classifier is analyzed based on the following measures: accuracy, F1 Score, precision rate, recall rate, time, and area under the curve (AUC). The detailed numerical results of each dataset are given below subsection.

4.1 UT-Interaction Dataset Results

The numerical results of the proposed framework on the UT-Interaction dataset are given in Table 1. Ten classifiers, such as Narrow Neural Network, Medium Neural Network, and named a few more, are utilized for the classification purpose. The best accuracy achieved is 100% on Wide Neural Network (WNN), where the other performance measures like precision rate, recall rate, F1 Score, and time are 100%, 100%, 100%, and 18.7 (s). The second best-achieved accuracy on Medium Neural Network of 99.9%, with a recall rate of 99.8, a precision rate of 99.8, an F1 Score of 99.8%, and a testing classification time of 12.8 (s). The rest of the classifiers' accuracies are 99.8%, 99.7%, 99.2%, 99.9%, 98%, 98.8%, and 99.1%. The confusion matrix of WNN is illustrated in Fig. 6 and can be utilized to verify the best-achieved accuracy.

Table 1: The proposed HAR framework classification results on the UT-Interaction dataset

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Narrow neural network	99.8	99.8	99.8	99.8	15.6
Medium neural network	99.9	99.8	99.8	99.8	12.8
Wide neural network	**100**	**100**	**100**	**100**	**18.7**
Bilayered neural network	99.7	99.6	99.6	99.6	13.5
Trilayered neural network	99.2	99.2	99.1	99.2	24.8

(Continued)

Table 1: Continued

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Cubic SVM	99.9	99.8	99.8	99.8	38.1
Quadratic SVM	99.9	99.8	99.8	99.8	33.3
Cubic KNN	98.0	97.6	98.1	97.8	289.8
Cosine KNN	98.8	98.6	98.9	98.7	23.1
Boosted tree	99.1	98.8	98.9	98.8	181.1

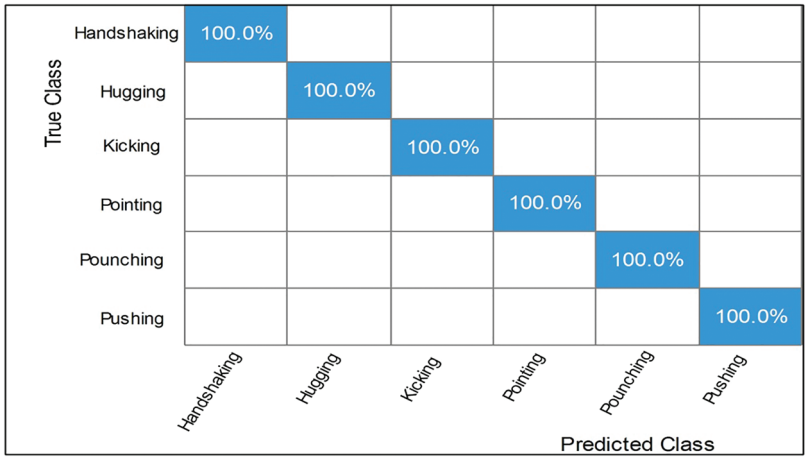

Figure 6: Confusion matrix of wide neural network on U.T. interaction dataset

4.2 Weizmann Dataset Results

The proposed framework results on Weizmann dataset are shown in Table 2. For the experimental process, several classifiers were used, similar to Table 1. The best accuracy achieved of 98.1% by Quadratic SVM, while the precision rate, recall rate, F1 Score, and time are 97.7%, 97.7%, 97.7%, and 85.8% (s). The second-best accuracy achieved on Cubic SVM of 98%, with a precision rate of 97.8, a recall rate of 97.6, an F1 score of 97.7%, and a time of 95 (s). Similarly, the rest of the classifiers attained 96.2%, 97.4%, 96.2%, 94.9%, 89.1%, 92.1%, and 91% accuracies, respectively. Fig. 7 illustrates the confusion matrix of QSVM, which confirms the accuracy of the proposed framework.

Table 2: The proposed HAR framework's classification results on the Weizmann dataset

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Narrow neural network	96.2	95.6	95.6	95.6	32
Medium neural network	97.4	97	97	97	19.5
Wide neural network	97.5	97	97.1	97	28.2
Bilayered neural network	96.2	95.5	95.6	95.5	32
Trilayered neural network	94.9	94.1	94.2	94.1	55.8
Cubic SVM	98	97.6	97.8	97.7	95

(Continued)

Table 2: Continued

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Quadratic SVM	**98.1**	**97.7**	**97.7**	**97.7**	**85.8**
Cubic KNN	89.1	87.6	88.8	88.1	438.3
Cosine KNN	92.1	90.7	91.5	91	31.6
Boosted tree	81.0	78.07	77.7	77.8	357.4

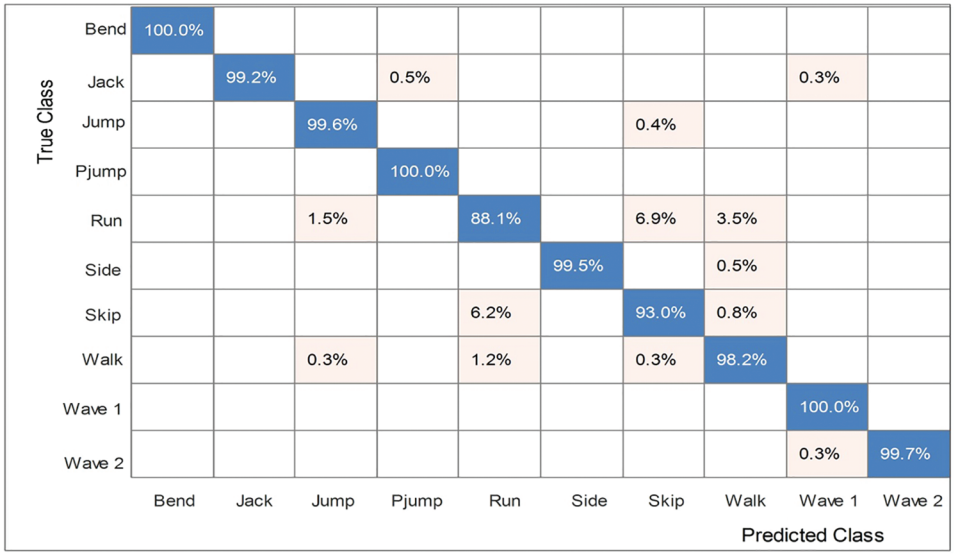

Figure 7: Confusion matrix of Quadratic SVM on Weizmann dataset

4.3 KTH Dataset Results

The proposed numerical results on the KTH dataset are shown in Table 3. Several classifiers, such as Cubic SVM, Narrow Neural Network, Medium Neural Network, and named a few more, are utilized for classification purposes. The maximum noted the accuracy of 99.6% on Quadratic SVM, while the other parameter like: 99.6 is the Recall rate, 99.6 is the Precision rate, F1 score is 99.6%, and the testing time is 152.7 (s). The rest of the classifier's accuracies are 99.4, 99.5, 99.4, 99.5, 98.4, 99.1, and 99.2, as given in the table below. Fig. 8 illustrates the confusion matrix of QSVM, which confirms the performance of the proposed framework.

Table 3: The proposed HAR 'framework's classification results on the KTH dataset

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Narrow neural network	99.4	99.4	99.4	99.4	74.6
Medium neural network	99.5	99.4	99.4	99.4	47.3
Wide neural network	99.5	99.4	99.4	99.4	67
Bilayered neural network	99.4	99.4	99.4	99.4	95.7

(Continued)

Table 3: Continued

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Trilayered neural network	99.5	99.4	99.4	99.4	122.6
Cubic SVM	99.6	99.6	99.6	99.6	177.7
Quadratic SVM	**99.6**	**99.6**	**99.6**	**99.6**	**152.7**
Cubic KNN	98.4	98.3	98.6	98.4	7290
Cosine KNN	99.1	99	99.1	99	315.5
Boosted tree	99.2	99.1	99.2	99.1	994

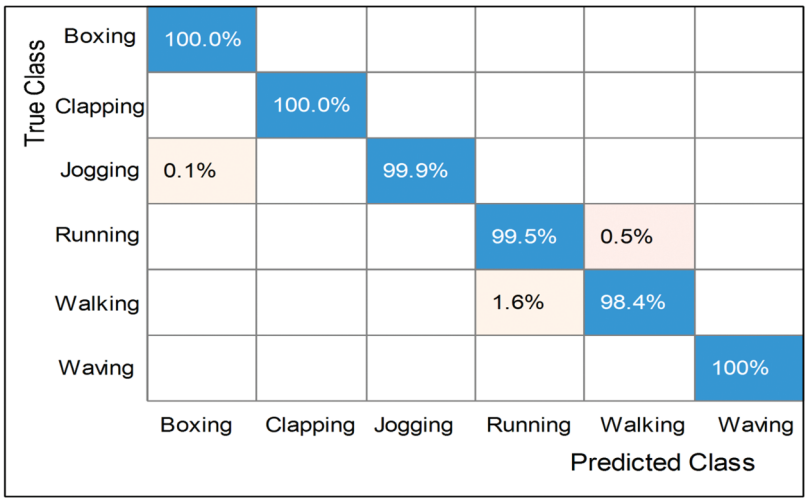

Figure 8: The confusion matrix of Quadratic SVM on the KTH dataset

4.4 Hollywood Dataset Results

Table 4 shows the result of the proposed framework on the Hollywood dataset. Ten classifiers, such as Quadratic SVM, Narrow Neural Networks, Medium Neural Networks, and named a few more, are utilized for classification purpose. The Medium Neural Network achieves the best accuracy of 99.9%, while the other parameters are: 99.9 is the precision rate, 99.8 is the recall rate, the F1 score is 99.8, and time is 67.2 (s). For the rest of the classifiers, the obtained accuracies are 99.8%, 99.8%, 99.9%, 99.9%, 99.4%, 99.2%, and 99.3%, respectively, as presented in the table below. The confusion matrix of Medium NN is also illustrated in Fig. 9, which confirms the accuracy of the proposed framework on the Hollywood dataset.

Table 4: The proposed HAR framework classification results on the Hollywood dataset

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Narrow neural network	99.9	99.7	99.9	99.8	72.8
Medium neural network	**99.9**	**99.8**	**99.9**	**99.8**	**67.2**
Wide neural network	99.9	99.8	99.9	99.8	88.4

(Continued)

Table 4: Continued

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Bilayered neural network	99.8	99.7	99.8	99.7	86.6
Trilayered neural network	99.8	99.5	99.6	99.5	154.1
Cubic SVM	99.9	99.8	99.8	99.8	439.9
Quadratic SVM	99.9	99.8	99.9	99.8	397.8
Cubic KNN	99.4	98.9	99.3	99	8573.7
Cosine KNN	99.2	99.2	99.2	99.2	9321
Boosted tree	99.3	99.2	99.3	99.2	8112.8

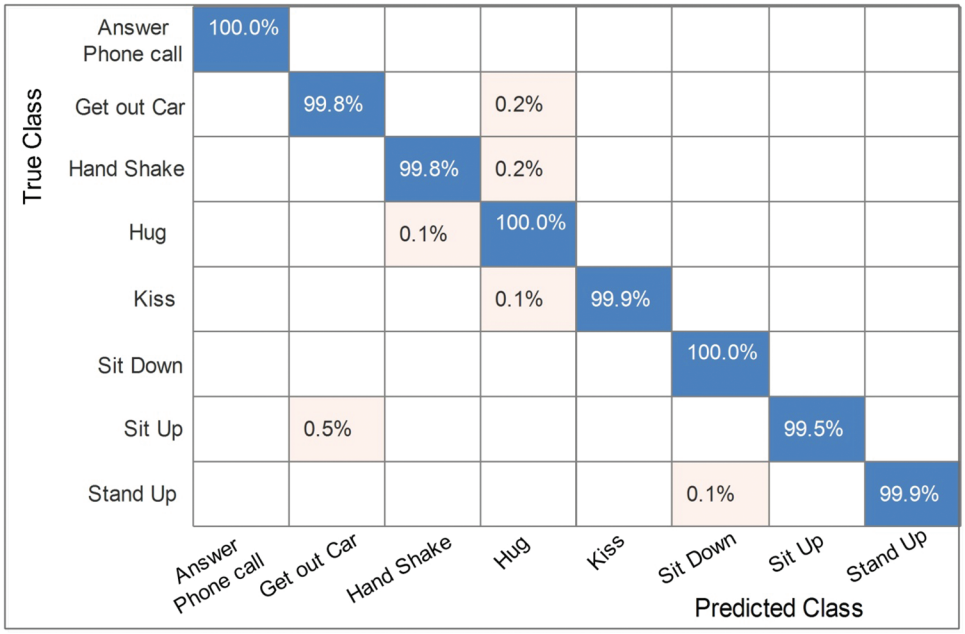

Figure 9: Confusion matrix of Medium N.N. for Hollywood dataset

4.5 IXAMS Dataset Results

Table 5 shows the result of the proposed framework on the IXAMS dataset. Ten classifiers, such as Quadratic SVM, Narrow Neural Networks, Medium Neural Networks, and named a few more, are utilized for classification purpose. The highest 97.5% accuracy is achieved on Cubic SVM, while the other parameter like precision rate, recall rate, time, and F1 score are 97.4, 97.4, 1726.5 (s), and 97.4%. The Quadratic SVM achieves the second-best accuracy of 96.9%. The rest of the classifiers achieved accuracies of 90.3%, 93.9%, 88.2%, 87.3%, 87.2%, 86.5%, and 94.3%, respectively. The confusion matrix is also presented in Fig. 10, confirming the proposed framework's accuracy on Cubic SVM. Finally, the comparison of the proposed method to the state-of-the-art method is made based on accuracy, as represented in Table 6. This table represents that the proposed framework accuracy is improved than the existing techniques on selected datasets.

Table 5: The proposed HAR framework classification results on the IXAMS dataset

Classifiers	Accuracy (%)	Recall rate	Precision rate	F1 score	Time (s)
Narrow neural network	90.3	89.9	89.9	89.9	195.2
Medium neural network	93.9	93.7	93	93.3	100.8
Wide neural network	95.4	95.2	95.2	95.2	138.6
Bilayered neural network	88.2	87.7	87.8	87.7	342.3
Trilayered neural network	87.3	86.7	86	86.3	720.4
Cubic SVM	**97.5**	**97.4**	**97.4**	**97.4**	**1726.5**
Quadratic SVM	96.9	96.9	96.9	96.9	1324.9
Cubic KNN	87.2	86.8	87.3	87	7535.7
Cosine KNN	86.5	86.5	86.7	86.6	8325.5
Boosted tree	94.3	94.3	94.3	94.3	6375.3

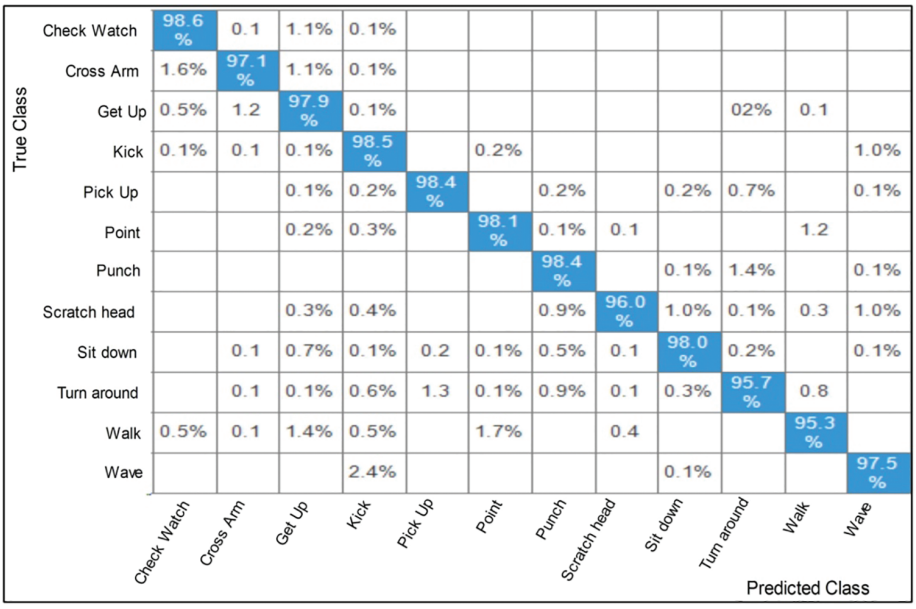

Figure 10: The confusion matrix of Cubic SVM for IXMAS dataset

Table 6: Comparison of proposed method accuracy with state of the art techniques

Method	Year	Dataset	Accuracy (%)
[38]	2020	KTH	94.70
[39]	2022	KTH	98.3
[40]	2019	IXAMS	97.0
[41]	2021	IXAMS	84.8
[42]	2021	UT-Interaction	96.7

(Continued)

Table 6: Continued

Method	Year	Dataset	Accuracy (%)
Proposed		UT-Interaction	**100.0**
		Weizmann	**98.1**
		KTH	**99.6**
		Hollywood	**99.8**
		IXAMS	**97.5**

5 Conclusion

This article for HAR presents a unified framework based on deep learning and an improved tree seed algorithm. The proposed method's primary focus is the fusion of deep learning features and selecting the best of them using an improved optimization algorithm. The experiment was carried out on five publicly available datasets, yielding improved accuracies of 100.0%, 98.1%, 99.6%, 99.8%, and 97.5%, respectively. The efficiency of the proposed fusion method is demonstrated based on the obtained results. However, this method requires a significant amount of computational time; as a result, classification accuracy improves, and the improved optimization algorithm alleviates this issue. On the other hand, redundant information in this work increases the computational time and decreases classification accuracy. Future work may investigate the features fusion technique by involving some optimization algorithms. Moreover, the latest dataset shall be considered for the validation of the proposed framework [33]. In the future, multiple angles shall be considered for action recognition. There will be a privacy issue from several angles but some IoT controlled deep learning frameworks may be useful [43].

Funding Statement: This work was supported by "Human Resources Program in Energy Technology" of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resources from the Ministry of Trade, Industry & Energy, Republic of Korea. (No. 20204010600090).

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

References

[1] M. S. Islam, K. Bakhat, R. Khan, N. Naqvi and M. M. Islam, "Applied human action recognition network based on SNSP features," *Neural Processing Letters*, vol. 2, no. 5, pp. 1–14, 2022.

[2] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun and G. Wang, "Human action recognition from various data modalities: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 6, pp. 1–9, 2022.

[3] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *International Journal of Computer Vision*, vol. 130, no. 21, pp. 1366–1401, 2022.

[4] V. Mazzia, S. Angarano, F. Salvetti and M. Chiaberge, "Action transformer: A self-attention model for short-time pose-based human action recognition," *Pattern Recognition*, vol. 124, no. 2, pp. 108487, 2022.

[5] M. Ahmed, M. Ramzan, H. U. Khan and S. Iqbal, "Real-time violent action recognition using key frames extraction and deep learning," *Computers, Material and Continua*, vol. 70, no. 1, pp. 1–16, 2021.

[6] T. T. Zin, Y. Htet, Y. Akagi, H. Tamura and K. Kondo, "Real-time action recognition system for elderly people using stereo depth camera," *Sensors*, vol. 21, no. 2, pp. 5895, 2021.

[7] A. Farnoosh, Z. Wang, S. Zhu and S. Ostadabbas, "A bayesian dynamical approach for human action recognition," *Sensors*, vol. 21, no. 7, pp. 5613, 2021.

[8] M. Bilal, M. Maqsood, S. Yasmin and S. Rho, "A transfer learning-based efficient spatiotemporal human action recognition framework for long and overlapping action classes," *The Journal of Supercomputing*, vol. 78, no. 21, pp. 2873–2908, 2022.
[9] Y. D. Zhang, S. A. Khan, M. Attique and A. Rehman, "A resource conscious human action recognition framework using 26-layered deep convolutional neural network," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 35827–35849, 2021.
[10] 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. 3, pp. 79–91, 2021.
[11] 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. 59–86, 2020.
[12] M. H. Kolekar and D. P. Dash, "Hidden markov model based human activity recognition using shape and optical flow based features," in *2016 IEEE Region 10 Conf. (TENCON)*, Mumbai, India, pp. 393–397, 2016.
[13] T. Krzeszowski, K. Przednowek, K. Wiktorowicz and J. Iskra, "The application of multiview human body tracking on the example of hurdle clearance," *Sport Science Research and Technology Support*, vol. 22, no. 1, pp. 116–127, 2016.
[14] A. Kushwaha, A. Khare and M. Khare, "Human activity recognition algorithm in video sequences based on integration of magnitude and orientation information of optical flow," *International Journal of Image and Graphics*, vol. 22, no. 1, pp. 2250009, 2022.
[15] Y. D. Zhang, M. Allison, 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. 13, no. 2, pp. 1–16, 2021.
[16] N. O'Mahony, S. Campbell, A. Carvalho and S. Harapanahalli, "Deep learning vs. traditional computer vision," *Science and Information*, vol. 11, no. 4, pp. 128–144, 2019.
[17] T. Akram, M. Sharif, M. Y. Javed and N. Muhammad, "An implementation of optimized framework for action classification using multilayers neural network on selected fused features," *Pattern Analysis and Applications*, vol. 22, no. 6, pp. 1377–1397, 2019.
[18] A. Karpathy, G. Toderici, S. Shetty, T. Leung and R. Sukthankar, "Large-scale video classification with convolutional neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 1725–1732, 2014.
[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," ArXiv, vol. 2, no. 1, pp. 1–6, 2014.
[20] K. Aurangzeb, I. Haider, T. Saba, K. Javed and T. Iqbal, "Human behavior analysis based on multi-types features fusion and Von nauman entropy based features reduction," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 2, pp. 662–669, 2019.
[21] A. Sharif, K. Javed, H. Gulfam, T. Iqbal and T. Saba, "Intelligent human action recognition: A framework of optimal features selection based on Euclidean distance and strong correlation," *Journal of Control Engineering and Applied Informatics*, vol. 21, no. 2, pp. 3–11, 2019.
[22] H. Arshad, R. Damaševičius, A. Alqahtani, S. Alsubai and A. Binbusayyis, "Human gait analysis: A sequential framework of lightweight deep learning and improved moth-flame optimization algorithm," *Computational Intelligence and Neuroscience*, vol. 22, no. 1, pp. 1–21, 2022.
[23] H. Arshad, M. Sharif, M. Yasmin and M. Y. Javed, "Multi-level features fusion and selection for human gait recognition: An optimized framework of Bayesian model and binomial distribution," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 6, pp. 3601–3618, 2019.
[24] M. Azhar, K. Ibrar, A. Alqahtani, S. Alsubai and A. Binbusayyis, "COVID-19 classification from chest x-ray images: A framework of deep explainable artificial intelligence," *Computational Intelligence and Neuroscience*, vol. 22, no. 1, pp. 31–46, 2022.

[25] K. Muhammad, S. H. Wang, S. Alsubai, A. Binbusayyis and A. Alqahtani, "Gastrointestinal diseases recognition: A framework of deep neural network and improved moth-crow optimization with DCCA fusion," *Human-Centric Computing and Information Sciences*, vol. 12, no. 5, pp. 1–16, 2022.
[26] S. Kadry, M. Alhaisoni, Y. Nam, Y. Zhang and V. Rajinikanth, "Computer-aided gastrointestinal diseases analysis from wireless capsule endoscopy: A framework of best features selection," *IEEE Access*, vol. 8, no. 2, pp. 132850–132859, 2020.
[27] C. Liang, D. Liu, L. Qi and L. Guan, "Multi-modal human action recognition with sub-action exploiting and class-privacy preserved collaborative representation learning," *IEEE Access*, vol. 8, no. 1, pp. 39920–39933, 2020.
[28] S. Nazir, Y. Qian, M. Yousaf, S. A. Velastin Carroza and E. Izquierdo, "Human action recognition using multi-kernel learning for temporal residual network," *Sensors*, vol. 2, no. 6, pp. 1–21, 2019.
[29] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 2, pp. 1–9, 2022.
[30] S. Aly and A. Sayed, "Human action recognition using bag of global and local Zernike moment features," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 24923–24953, 2019.
[31] Y. Du, W. Wang and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 1110–1118, 2015.
[32] H. Gammulle, S. Denman, S. Sridharan and C. Fookes, "Two stream lstm: A deep fusion framework for human action recognition," in *2017 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, N.Y., USA, pp. 177–186, 2017.
[33] I. M. Nasir, M. Raza, J. H. Shah, S. H. Wang and U. Tariq, "HAREDNet: A deep learning based architecture for autonomous video surveillance by recognizing human actions," *Computers and Electrical Engineering*, vol. 99, no. 4, pp. 10–28, 2022.
[34] Z. Gao, P. Wang, H. Wang, M. Xu and W. Li, "A review of dynamic maps for 3D human motion recognition using ConvNets and its improvement," *Neural Processing Letters*, vol. 52, no. 7, pp. 1501–1515, 2020.
[35] C. Bailer, T. Habtegebrial and D. Stricker, "Fast feature extraction with CNNs with pooling layers," *ArXiv*, vol. 7, no. 2, pp. 1–8, 2018.
[36] G. Huang, Z. Liu and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, NY, USA, pp. 4700–4708, 2017.
[37] M. S. Kiran, "TSA: Tree-seed algorithm for continuous optimization," *Expert Systems with Applications*, vol. 42, no. 11, pp. 6686–6698, 2015.
[38] Q. Meng, H. Zhu, W. Zhang, X. Piao and A. Zhang, "Action recognition using form and motion modalities," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3, pp. 1–16, 2020.
[39] M. N. Akbar, F. Riaz, A. B. Awan and S. Rehman, "A hybrid duo-deep learning and best features based framework for action recognition," *Computers, Materials & Continua*, vol. 73, no. 4, pp. 2555–2576, 2022.
[40] D. Purwanto, R. R. A. Pramono, Y. T. Chen and W. H. Fang, "Three-stream network with bidirectional self-attention for action recognition in extreme low resolution videos," *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1187–1191, 2019.
[41] H. B. Naeem, F. Murtaza, M. H. Yousaf and S. A. Velastin, "T-VLAD: Temporal vector of locally aggregated descriptor for multiview human action recognition," *Pattern Recognition Letters*, vol. 148, no. 11, pp. 22–28, 2021.
[42] 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. 69, no. 2, pp. 1–15, 2021.
[43] Z. Liang, M. Yin, J. Gao, Y. He and W. Huang, "View knowledge transfer network for multi-view action recognition," *Image and Vision Computing*, vol. 118, no. 31, pp. 104357, 2022.