Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.039289

ARTICLE

HybridHR-Net: Action Recognition in Video Sequences Using Optimal Deep Learning Fusion Assisted Framework

Muhammad Naeem Akbar[1,*], Seemab Khan[2], Muhammad Umar Farooq[1], Majed Alhaisoni[3], Usman Tariq[4] and Muhammad Usman Akram[1]

[1]Department of Computer Engineering, National University of Sciences and Technology (NUST), Islamabad, 46000, Pakistan

[2]Department of Robotics, SMME NUST, Islamabad, 45600, Pakistan

[3]Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

[4]Management Information System Department, College of Business Administration, Prince Sattam bin Abdulaziz University, Al-Kharj, 16278, Saudi Arabia

*Corresponding Author: Muhammad Naeem Akbar. Email: naeemakbar@ceme.nust.edu.pk

Received: 20 January 2023 Accepted: 25 May 2023 Published: 08 October 2023

ABSTRACT

The combination of spatiotemporal videos and essential features can improve the performance of human action recognition (HAR); however, the individual type of features usually degrades the performance due to similar actions and complex backgrounds. The deep convolutional neural network has improved performance in recent years for several computer vision applications due to its spatial information. This article proposes a new framework called for video surveillance human action recognition dubbed HybridHR-Net. On a few selected datasets, deep transfer learning is used to pre-trained the EfficientNet-b0 deep learning model. Bayesian optimization is employed for the tuning of hyperparameters of the fine-tuned deep model. Instead of fully connected layer features, we considered the average pooling layer features and performed two feature selection techniques-an improved artificial bee colony and an entropy-based approach. Using a serial nature technique, the features that were selected are combined into a single vector, and then the results are categorized by machine learning classifiers. Five publically accessible datasets have been utilized for the experimental approach and obtained notable accuracy of 97%, 98.7%, 100%, 99.7%, and 96.8%, respectively. Additionally, a comparison of the proposed framework with contemporary methods is done to demonstrate the increase in accuracy.

KEYWORDS

Action recognition; entropy; deep learning; transfer learning; artificial bee colony; feature fusion

1 Introduction

Over the last decade, machine learning (ML) has emerged as one of the most rapidly growing fields in advanced computer sciences. Several studies in Activity Recognition have been conducted using machine learning and computer vision [1]. However, they encountered various types and similarities between multiple human actions, making it more difficult to identify the action accurately. Several

techniques for action recognition have been introduced in the past. These techniques belong to traditional ML methods such as Convolution Neural Networks (CNN) and sparse coding (SC). Few advanced ML techniques, including Long-term Short Memory (LTSM), Deep Convolutional Neural Networks (DCNN), and recurrent neural networks (RNN), have also been employed for action recognition with improved accuracy [2].

These advanced techniques are comprised of complex architectures that require a lot of memory and have limitations regarding computational resources for HAR applications. Real-world applications of HAR may include Human-Computer Interaction (HCI) and some intelligent video surveillance applications. Mobile Edge Computing (MEC) also contributes a lot of technology integration in the field of medicine. Automation in remote health care supervision is also one of the advantages of MEC. The technique is also applicable in action recognition. The services where HAR might be applicable may include content-centric summarization [3], sports video analysis and evaluation, and remote health monitoring applications for intelligent surveillance. Silhouette-based features can support robust detection of actions in a real-time environment [4]. Action recognition from video streams has advanced from analyzing the present to forecasting the coming action. It applies highly to surveillance, driverless cars, and entertainment [5].

EfficientNet Models [6] are state-of-the-art deep CNN (DCNN) models comprising meek yet highly potent compound scaling functions. The function can scale a baseline CNN to a target resource bound while maintaining model efficiency. EfficientNet is a scale-able model in terms of layer depth, width, and resolution, which makes it capable of performing better than other DCNNs, which include AlexNet, GoogleNet, and MobileNet. It has become an important and basic component of new computer vision research, especially in deep learning. In the proposed technique, EfficientNet [7] is used to extract the best features from multiple datasets, and these feature vectors are further processed. Transfer learning involves transferring information from the source domain (ImageNet) to the target domain [8]. Information is transferred to get the best features from the datasets. Fully connected layers are modified to account for the no number of classes in each dataset. The technique helps to create a high-performance method that uses pre-trained models [9].

Major Challenges and Contributions: Intangible ML and Data Mining (DM) techniques have been applied to solve numerous real applications. Feature fusion is a technique where extracted feature vectors from the training images are fused based on some pre-determined standard [10]. The fused vector has the best features with a high contribution. In supervised learning, the dataset is kept in two sets, training, and testing, depending upon the ratio set by the researcher. Training images are used to make the model learn, and then the proposed model is validated on testing images. Evaluation is done on pre-defined parameters [11]. The current deep learning systems mainly focus on hybridizing the latest and traditional deep learning methods. Most of the hybrid techniques managed to improve the accuracy, but their least focus was on reducing the time complexity. Computational time is a significant component, especially in action recognition problems, as the system needs to identify the correct action in a minimum time [12]. Some other factors that need to be sorted for better results include redundant and irrelevant or unimportant features.

In this work, we proposed a deep learning and Entropy controlled optimization algorithm-based framework for action recognition. The following are our main contributions:

- Fine-tuned EfficientNet-B0 deep learning model and training are performed on selected action recognition datasets using deep transfer learning. The deep model's training has been done with static hyperparameters.

- Entropy-controlled Artificial Bee Colony optimization algorithm is proposed for the best feature selection.
- Fusion is performed using a mean deviation-based serial threshold function.

2 Literature Review

Recently, HAR has grown in importance as a research field. The researchers have adapted several supervised and unsupervised learning methods for HAR applications [13]. It is essential to consider all available clues to analyze human behavior and predict the appropriate action later. Human action can also be identified using the blend of some traditional techniques with advanced deep learning methods. Traditional methods for action recognition may not produce the best result when used in isolation—a hybrid of conventional and advanced techniques performed better in several recent studies.

Masmoudi et al. [2] presented an unsupervised CNN that has overcome memory and computational issues to a greater extent. PCANet-TOP is an unsupervised convolutional PCANet architecture; it can learn spatiotemporal features from Three Orthogonal Planes (TOP). To reduce the dimensions of the learned features, whitening PCA has been used. They used a Support Vector Machine (SVM) to classify action. The presented techniques were assessed on Weizmann, royal institute (KTH), UCF Sports, and YouTube actions datasets, and the achieved accuracy on these datasets is 90%, 87.33%, 92.67%, and 81.40%, respectively. Results have proven that the presented principle component analysis (PCANet-TOP) model provides distinguishing and balancing features using TOP. It also enabled us to attain comparatively better results than the existing techniques. Ramya et al. [14] presented an algorithm based on distant transform and entropy features extracted from the human silhouettes. The first step was to attain the silhouettes, which were performed by using the correlation coefficient-based frame difference method. Then, the step was to extract features using Entropy and distance transform. This helped by facilitating the model with contour and deviation information. In the final step, the extracted features were given to neural networks to classify human actions. Datasets used to assess the presented model include Weizmann, KTH, and UCF50, and the achieved accuracy on them was 92.5%, 91.4%, and 80%, respectively. Researchers also observed that there is still room for improvement, and results can be improved by manipulating the training testing ratio in the future. The local variation features and fused shape features resulted in the better performance of the algorithm.

Khan et al. [9] worked on a deep learning algorithm for HAR based on Kurtosis based weighted k-nearest neighbor (KNN). The architecture included four steps: feature extraction and mapping, kurtosis-based feature selection, serial-based feature fusion, and action identification. For feature extraction, two CNN models were used: DenseNet201 and Inception3. The classification was carried out on four different datasets: KTH, IXMAS, WVU, and Hollywood, with the obtained results being 99.3%, 97.4%, 99.8%, and 99.9%, respectively. It was discovered here that less features are included for the final classification aided in improving the algorithm's performance. Khan et al. [9] presented a Gated Recurrent Neural Network that has amplified computational competency. For action classification, researchers have used sequential data. Gaussian mixture model (GMM) and Kalman's filters were used to extract features. A novel approach based on hybrid deep learning methods was used for recognition. The GRUs aid in modeling the problem by the current sequential dependencies. Furthermore, graph regression neural network (GRNN) can be used to model problems with temporal relationships and time gaps between events. The method was tested using the KTH, UCF101, and UCF sports datasets.

Basak et al. [15] presented multiple ways to recognize action, including red, green, blue (RGB), depth, point cloud, infrared, etc. The choice of technique depends on the nature of the scenario

and the application for which it is being developed. A survey of the performance of various HAR techniques is presented. The study surveyed Fusion techniques, including the Fusion of RGB, depth, and skeleton modalities. Among the existing fusion techniques, the fusion of A/V modalities produced the best results in predicting actions. Aside from the fusion, co-learning techniques were thoroughly investigated. It was a technique for transferring learning by extracting knowledge from auxiliary modalities and applying it to learning another modality. Visual modalities such as RGB and depth are included in these co-learning techniques. Fu et al. [16] presented an algorithm to detect sports actions using deep learning methods, specifically the algorithm of clustering extraction. Athletic movements were first detected from deep learning techniques and then fused with sports-centered movements. CNN was applied on the sample set where non-athletic and negative images were provided to the network. The set was gradually enhanced with gathered false positive predictions, and the obtained results were then optimized using a clustering algorithm. The idea was to acquire athletes' training posture by analyzing the movements of their specific sport. The application was designed to assist sports trainers in giving professional training to athletes effectively and efficiently.

Liang et al. [17] developed a hybrid of CNN and Short-Term Long Memory (LTSM). Extensive testing has been carried out to determine the efficacy of the hybrid method. The paper also included a comparison of various deep-learning techniques. The researchers named their technique CNN + LTSM. First, the results demonstrated that the efficiency of learning algorithms differed marginally, but this did not affect the overall result. Second, it claimed that spatial, temporal interest point (STIP) could perform even better in the given conditions because it could extract interest points in video frames containing various human actions. Yue et al. [18] performed survey research on multiple robust and operative architectures for HAR and future action predictions. The study compared state-of-the-art methods for the recognition and prediction of actions. Recent models, efficient algorithms, challenges, popular datasets, evaluation criteria, and future guidelines were also presented with documented proofs. After detailed study and analysis, it was concluded that better datasets provide a foundation for better prediction of actions.

3 Methodology

In this section, a detailed methodology for the proposed architecture has been presented. The complete architecture consists of various steps, including feature extraction via transfer learning, using two optimizers, i.e., Artificial Bee Colony and Entropy-controlled feature selection, and serial-based feature fusion. The proposed HAR architecture is illustrated in Fig. 1.

3.1 Datasets

In this work, five publicly accessible datasets have been utilized for the experimental approach. The datasets include IXMAS [19], KTH [20], UT Interaction [20], UCF Sports [20], and Weizmann [20]. All these datasets have been well-known and used by several researchers in the last few years. The IXMAS and Weizmann have ten action classes, whereas the KTH and UT Interaction datasets have six action classes. UCF Sports action dataset contains 13 action classes.

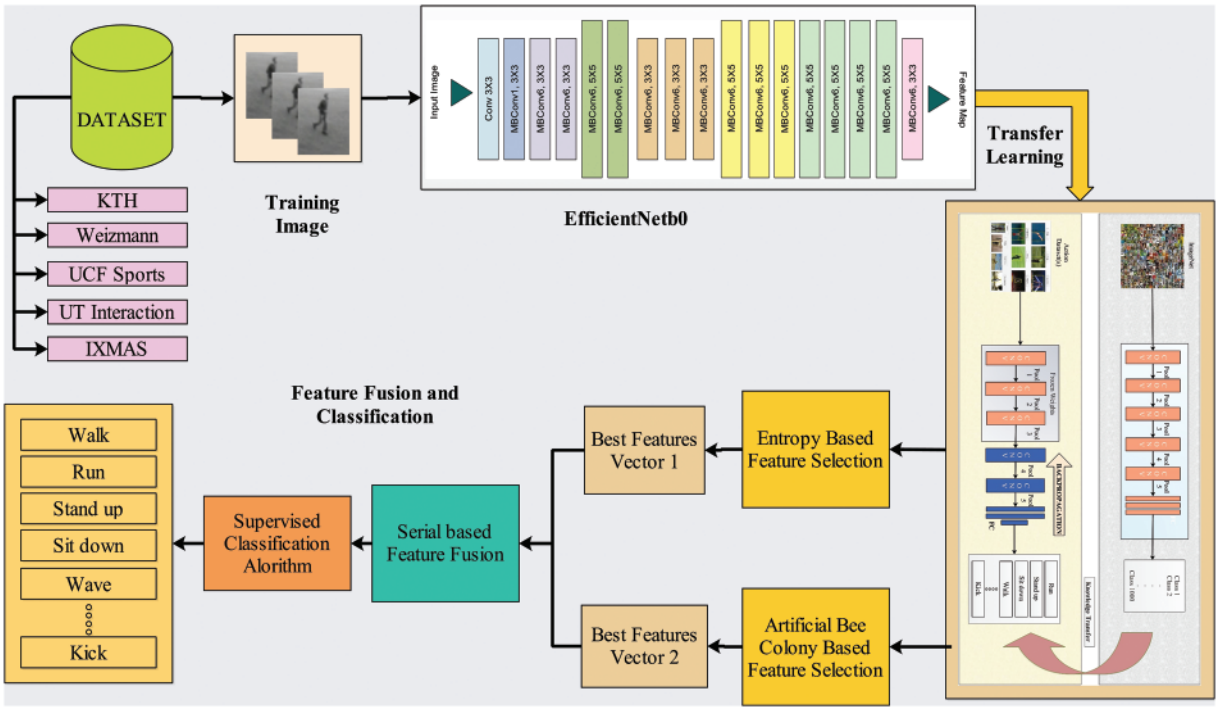

Figure 1: Visual illustration of the proposed framework for action recognition

3.2 Convolutional Neural Network (CNN)

In recent times, CNN has become immensely popular for image classification problems. Various studies are conducted to analyze the efficiency of CNN in spatial patterns that allow for extracting valuable features [21]. Recent trends in deep learning include spectral resolution, spatial grain, etc. CNN can apply to various problems in which classification, identification, and segmentation are at the top. The networks are useful for working on spatial patterns and enabling high spatial resolution data value. A variety of techniques for feature visualization by CNNs is helpful in the interpretation and allow learning from these models to improve its productivity. CNN is one of the novel techniques in machine learning that allows efficient and quick predictions for any given image. The network requires fewer parameters to learn than previously designed neural networks. A standard CNN has several layers, including the activation layer, i.e., ReLU (Rectified Linear unit) layer, the Pooling layer (Max, Avg, Min), the fully connected (FC) layer, and some other hidden layers. There exist a variety of CNNs, including AlexNet, GoogleNet, Inception, ResNet, and DenseNet. The general structure of a CNN with multi-layer architecture is illustrated in Fig. 2. The figure shows the complete design from input steam to final classification through the FC layer. Convolution layers are added to convolve the initial input and extract the required features. The extracted features are passed to multiple layers for further processing. After passing through different hidden layers, the network makes the final prediction. A simple architecture is illustrated in Fig. 2.

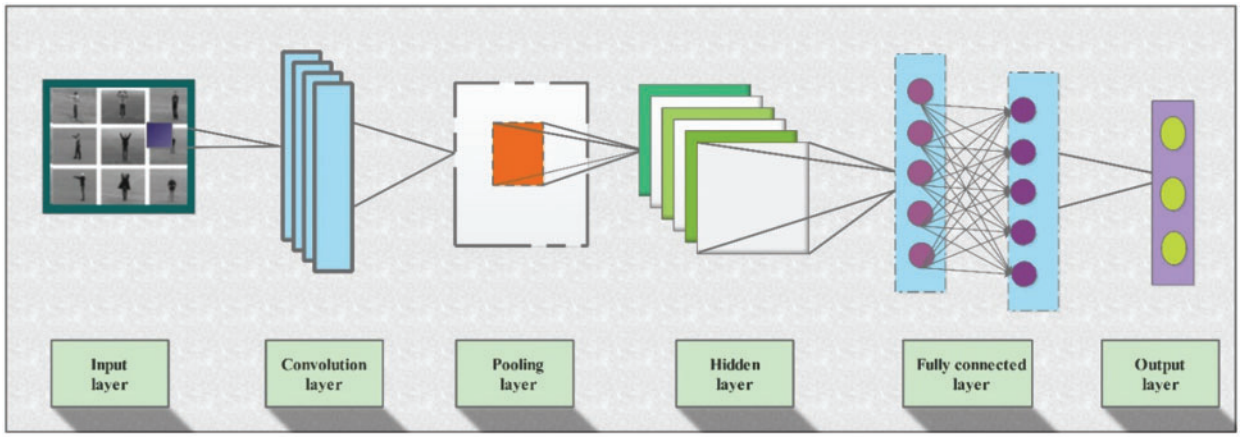

Figure 2: Detailed structure of a multi-layered convolutional neural network

3.3 EfficientNet-B0

EfficientNet is one of the best CNNs of recent times [22]. It is a family of prediction models from GoogleAI. It can scale up according to the number of parameters in the network. The model scales up with greater efficiency regarding the layer's depth, width, and resolution of the input image/video frame. It can scale up to a mix of the parameters mentioned above. To balance the dimensions of width, depth, and resolution, compound scaling is performed. These dimensions are scaled up on a fixed ratio. The mathematical representation of compound scaling is given below:

$$Depth = d = a^{\theta};\ Width = w = \omega^{\theta};\ resolution = r = r^{\theta} \tag{1}$$

Such that a $\cdot\ \omega^2 \cdot r^2 \approx 2$

$a \geq 1,\ \omega \geq 1,$ r $\geq$

The network also allows the creation of features instead of just feature extraction. These features can later be passed on to the classifier for predictions. The model outperformed all state-of-the-art networks of recent times, including ResNet, DenseNet, AlexNet, and others. In this research, the model is used on five different publically available datasets, and results are then compared on pre-defined criteria. Fig. 3 defines the complete network structure of an EfficientNet model.

3.4 Transfer Learning

Transfer learning is a modern invention in the field of ML. It is a technique where learned knowledge is transferred from a pre-trained model to a new network of the related domain [23]. Most ML algorithms are designed for designated applications, but in the case of transfer learning, the model can be reused for other applications after a little tweaking. Transfer learning modules depend on the ML algorithms being used for the predictions. Transfer methods can also be used as an extension of the algorithms already being used. Inductive learning is one of its dimensions where well-known algorithms are extended as Neural networks. These networks include Bayesian networks and Markov Logic networks. Another aspect is Reinforcement learning, where Q-learning and policy search algorithms are extended. Transfer learning aims to enhance the algorithm's performance in multiple aspects. Firstly, the initial performance by only transferring knowledge from the source domain. Secondly, time complexity can also be enhanced as the algorithm's final efficiency. Successful completion of knowledge transfer can greatly improve algorithm efficiency [24]. It also helps minimize

training time requirements as the information is transferred from a pre-trained network to the new one on a targeted domain. The technique requires less time for training and performs better than the other existing techniques. ImageNet is a sizable high-resolution image dataset that is typically utilized as the source domain in transfer learning. It is a dataset with 22,000 image categories and around 15 billion labels. Fig. 4 illustrates how knowledge is transferred from a source to a targeted domain. In this research, the targeted domain is action recognition. The source domain is defined as $D_s = \left\{\left(x_1^s, y_1^s\right), \ldots, \left(x_i^s, y_i^s\right), \ldots, \left(x_n^s, y_n^s\right)\right\}$ with a learning task L_d, L_s, $\left(x_m^d, y_m^d\right) \in \varphi$ and target domain $D_t = \left\{\left(x_1^t, y_1^t\right), \ldots, \left(x_i^t, y_i^t\right), \ldots, \left(x_n^t, y_n^t\right)\right\}$ with a learning task L_t, $\left(x_n^t, y_n^t\right) \in \varphi$, (m, n) will be training data size, where $n \ll m$ and y_i^s *and* y_i^t are the training data labels. The trained models have been employed for the features extraction. The layer, before fully connected, has been employed for feature extraction. A feature vector is obtained of dimensional $N \times 1280$ that is later optimized through Entropy controlled artificial bee colony (ABC).

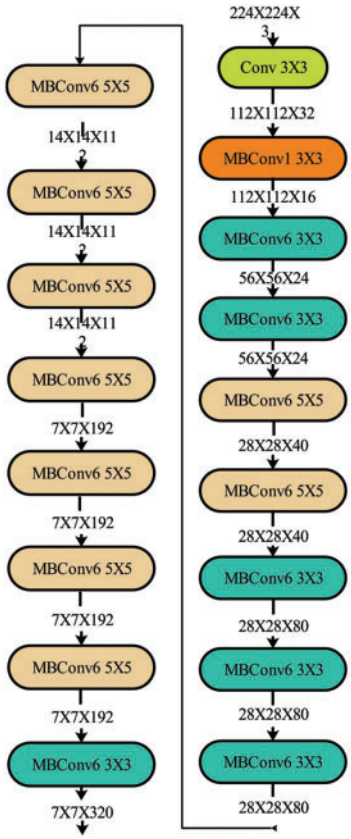

Figure 3: Detailed architecture of an EfficientNet-b0 deep learning model

3.5 Feature Selection

After feature extraction, the next step is to discard the features that do not contribute much to the performance. Next, the highest contribution feature is selected using two optimization algorithms, ABC and Entropy. In this section, the two algorithms are discussed in detail. Finally, from 1280 features extracted via EfficientNet-b0, the top 600 are selected in two separate feature vectors.

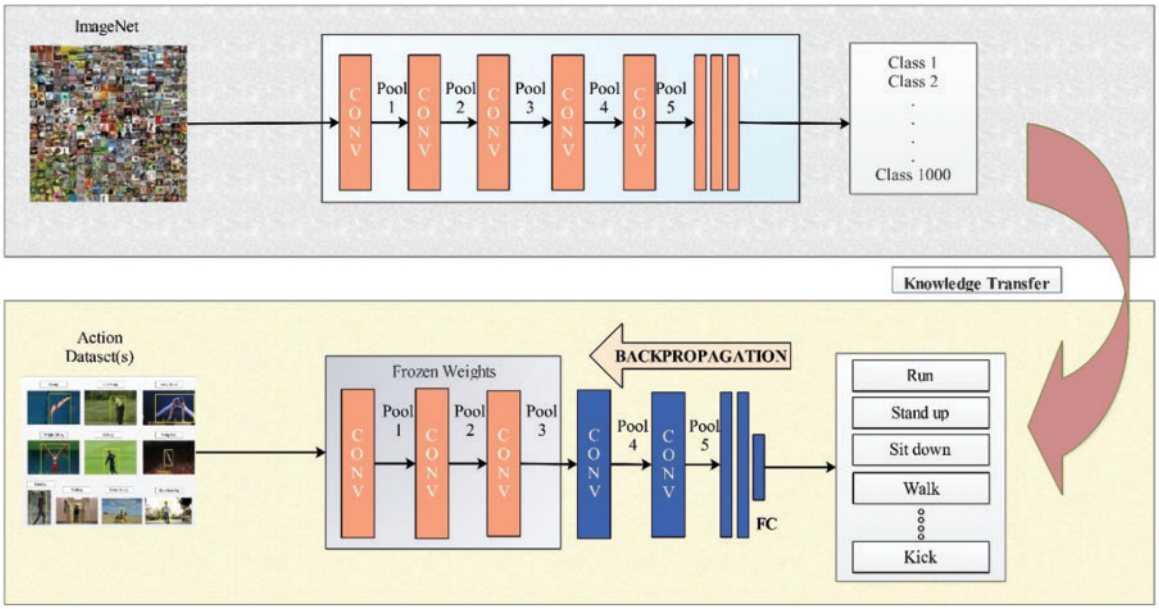

Figure 4: Illustration of transferring knowledge for action recognition

Artificial Bee Colony (ABC): Regarding the real-life bee colonies, ABC divides the bees into three groups: i) employed bees, ii) observer or onlooker bees, and iii) scout bees [25]. The job of the employed bees is to look for the food resource and convey the message to onlooker bees. On the given information, the onlookers choose to start exploring the nearby space of the food resource and find a new food resource. Employed bees with an improved food resource with already decided iterations get the scout status, and the new task for the scout is searching for a new food resource. ABC is employed in four fundamental steps:

The first step is an initialization, where the algorithm is set to produce random food resources. Each of them is defined as a vector in the search space; $x_i = x_{i,1}, x_{i,2}, x_{i.3}, \ldots, x_{i,n}$

$$x_{ij} = x_j^{max} \; R \; (0,1) \; \left(x_j^{max} - x_j^{min}\right) \tag{2}$$

where $i = \{1, 2, 3, \ldots, R\}$ and R is the number four resource which is equal to the number of employed bees or onlookers. $j = \{1, 2, 3, \ldots, \rho\}$ and ρ is the search space dimensions. x_{ij} is the jth dimension of x_i, R $(0, 1)$ is a random variable that uniformly distributes the search space. The minimum boundary value is x_j^{min} and maximum boundary value is x_j^{max} .

The second step is employed bees: Every employed bee is assigned a food resource, later modified by the bee itself after searching for a better resource. That is how knowledge is transferred from all the neighborhood except for the current location x_k. New food resource is located under Eq. (3);

$$x_{ij}^{'} = x_{ij} + \varphi_{ij} \left(x_{ij} - \; x_{kj}\right) \tag{3}$$

where x_i is the current food source location, φ_{ij} is a homogeneously distributed value within the given range $[-1, 1]$. After the initial position $x_i^{'}$ is found, the fitness value is assessed and equated with the x_i which is the current position. If $x_i^{'}$ is better than x_i, $x_i^{'}$ is replaced by x_i and this makes the algorithm enter its next iteration. The counter for the number of attempts for this iteration again resets to 0.

Otherwise x_i enters the next iteration with the same food resource value. The value of the counter, in this case, is upgraded to 1.

The third step focuses on onlooker bees. Each of the employed bees passes on the gathered information about their respective food resources to onlooker bees. Depending on the fitness value of the food resource, each onlooker bee selects a position, and for the selection, roulette wheel scheme is followed by the onlookers. They advocate that the better the source's fitness value, the higher the probability of selection. Probability is computed by Eq. (4).

$$\delta_i = \frac{fit_i}{\sum_{n=1}^{R} fit_n} \tag{4}$$

where fit_i is the fitness value of food resource x_i, After equating the probability of each location, a random number and (0, 1) is generated to govern the choice of food resource. If $\delta_i >$ rand (0, 1), x_i is selected as an employed bee in this step.

The last step caters to the scout bees; each food resource is initialized with 0. A counter contains the number of attempts. If the counter's value increases from the fixed value, the previous food resource will be discarded, and then a new food resource is assigned that is generated by Eq. (2).

Each food resource is added to a feature subset when the features are selected using ABC. Fitness value determines the quality of the food resource in the feature subset. Each source is represented in a binary string. One represents the selection, whereas 0 indicates the source is not selected.

Entropy-Based Selection: Entropy is the measure of uncertainty of the random variable λ. It measures the different probabilities among a set of limited values. Let λ be a random variable with a limited set of values having n values, such as $\{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$ and P is the set of a probability distribution. If a specified value λ_1 occurs with probability distribution $P(\lambda_i)$ such that $P(\lambda_i) \geq 0,\ i = 1, 2, 3 \ldots, n$ and $\sum_{i=1}^{n} (\lambda_i) = 1$, then the information amount is related to the known occurrences of λ_i can be defined as:

$$L(\lambda_i) = -\log P\ (\lambda_i) \tag{5}$$

This shows that the information generated in selecting a symbol $\lambda_i\ is - \log_2 P\ (\lambda_i)$ bits for a distinct source. On average, if the symbol λ_i is selected $n\ x\ P\ (\lambda_i)$ times in n selections, the average information gathered from n source outputs is given below:

$$-n\ x\ P\ (\lambda_1) \log P\ (\lambda_1) - n\ x\ P\ (\lambda_2) \log P\ (\lambda_2) - n\ x\ P\ (\lambda_3) \log P\ (\lambda_3), \ \ldots \tag{6}$$

$$n\ x\ P\ (\lambda_n) \log P\ (\lambda_n)$$

Mathematically, Entropy is the distribution function of a random variable λ which depends on the probabilities. Hence, Entropy $E(\lambda)$ is the mean value and can be determined by the following equation:

$$E(\lambda) = -\sum_{i=1}^{n} P\ (\lambda_i) \log P\ (\lambda_i) \tag{7}$$

3.6 Feature Fusion

In the feature selection phase, two feature vectors generated by ABC and Entropy are fused, and then the features are passed on to classification algorithms to assess their performance. The algorithm takes two vectors of 600 features each and combines them serial-wise in a single feature vector. It has $N \times 1200$ denoted by F_f. The approach is known as Serial-based Extended (SbE) for feature fusion. It can result in an improvement of the results as the improved feature vector enhances the performance of the classification algorithm. Considering two feature vectors $\mathrm{ABC} = \underset{A}{\rightarrow}$, $Entropy = \underset{E}{\rightarrow}$ Defined an

outline of features space $\underset{U}{\rightarrow}$, for an uninformed sample space $\beta \in \underset{U}{\rightarrow}$ and the equivalent two feature vectors are $a \in \underset{A}{\rightarrow}$ and $e \in \underset{E}{\rightarrow}$. The serial-wise fused vector of β is denoted as $\Bbbk = \begin{pmatrix} i \\ j \end{pmatrix}$. If the feature vector $\underset{A}{\rightarrow}$ has p dimensions and the feature vector $\underset{E}{\rightarrow}$ has q dimensions, then the fused feature vector $\Bbbk$ will be containing $(p+q)$ dimensions. Once $\Bbbk$ is formed, the residing features are arranged in ascending order, and their mean is computed. The final fusion is formed after the mean value is extracted.

$$\vartheta\left(\Bbbk\right) = \frac{1}{n}\sum\nolimits_{i=1}^{M}\left(\Bbbk_i\right) \tag{8}$$

$$F_f = \begin{cases} Fusion\,(i) & for\ \Bbbk_i \geq \vartheta \\ Discard, & Elsewhere \end{cases} \tag{9}$$

where *Fusion* (i) is the resultant of two feature vectors fused with $M \times J$. The value of J is modified in accordance with the variation in the training images.

4 Results and Discussion

This section focuses on the experiments performed and the analysis of the achieved results after extensive experimentation. In addition, performance measures and evaluation criteria are also discussed in the same section. A total of five datasets were chosen for use in this work; information on the datasets are given in Section 3.1. The results for each dataset are tabulated, and a complete analysis is provided along with the confusion matrix. 50% of the total images in the dataset were used for training, with the remaining 50% used for model validation. K fold cross-validation, where K equals 10. The criteria for evaluation include the achieved accuracy and the computational time (S). The entire experiment is conducted on MATLAB2021b using a Personal Desktop Computer with 16 GB of RAM and an 8 GB graphics card.

4.1 KTH Dataset Results

Extensive experimentation is performed during the study on different standardized datasets. There are 6 six classes of this dataset. The entire dataset is split into 50:50 for training and testing. Table 1 presents the results of this dataset which obtained the highest accuracy by Cubic SVM (CSVM) of 98.4% and a computational time is 157.6 S. In the second step, ABC optimization is used, and selected the best features. For this experiment, the CSVM obtained the highest accuracy of 98.6%, and the recorded computational time was 83.412 S. Then, the entropy-controlled weighted KNN-based selection technique is employed for selecting the best features in descending order. This experiment obtained the best accuracy of 98.6% on CSVM, whereas the computational time was 73.534 S. In the last step, both selected features are fused via the SBE feature fusion technique. As a result, the CSVM obtained the best accuracy of 98.7%, which is improved to the previous experiments; however, the testing time is increased.

The accuracy of CSVM can be checked through a confusion matrix, illustrated in Fig. 5.

Table 1: Achieved results on the KTH dataset. *Linear discriminant analysis (LDA)

Classifiers	Features				Measures	
	EfficientNet	Feature selection		Feature fusion	Accuracy (%)	Time (S)
		ABC	Entropy			
LDA	✓				97.3	36.486
		✓			95.9	15.539
			✓		96.2	19.772
				✓	96.8	21.521
L SVM	✓				97.1	142.9
		✓			96.3	74.185
			✓		96.3	69.604
				✓	96.8	77.027
Q SVM	✓				98.4	157.60
		✓			98.2	76.654
			✓		98.2	67.992
				✓	98.4	81.655
C SVM	✓				98.6	183.20
		✓			98.6	83.412
			✓		98.6	73.534
				✓	98.7	92.799
MG SVM	✓				97.8	303.04
		✓			97.7	130.39
			✓		97.8	105.59
				✓	97.8	129.28
F KNN	✓				97.6	163.22
		✓			97.8	76.699
			✓		97.7	75.739
				✓	96.4	144.09

4.2 Weizmann Dataset Results

Weizmann dataset results are presented in this section as numerical and confusion matrix. In the first experiment, features are extracted from the original EfficientNet model and performed classification. As a result, the CSVM obtained the best accuracy of 96.5%, whereas the noted computational time is 45.678 S. The ABC optimizer is applied in the second experiment, and the best features are selected. The best-selected features are classified using several classifiers and obtained the best accuracy of 96.4%. For this experiment, the computational time is reduced to 26.65 S, previously 45.678 (S). In the third experiment, entropy-based features were selected and obtained the best accuracy of 96.7%, whereas the computational time was 23.758 (S). In the last experiment,

SbE-based fusion was performed and obtained the best accuracy of 96.8%, which is improved to the previous experiments (seen in Table 2). Overall, the CSVM outperformed this dataset. Also, the fusion process's computational time is extended, but accuracy is also improved. In addition, the CSVM confusion matrix, which can be used to confirm the proposed accuracy, is shown in Fig. 6.

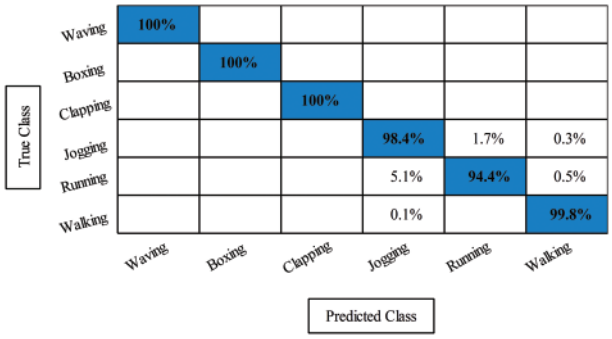

Figure 5: Confusion matrix for feature fusion on cubic SVM classifier on KTH dataset

Table 2: Achieved results on Weizmann dataset

Classifiers	Features				Measures	
	EfficientNet	Feature selection		Feature fusion	Accuracy (%)	Time (S)
		ABC	Entropy			
LDA	✓				96.1	20.56
		✓			96.2	16.772
			✓		95.5	6.356
				✓	96.8	15.293
L SVM	✓				94.2	37.642
		✓			94.0	23.23
			✓		93.6	22.873
				✓	94.1	28.909
Q SVM	✓				96.3	43.166
		✓			96.4	23.924
			✓		96.4	24.19
				✓	96.2	30.471
C SVM	✓				96.5	45.678
		✓			96.4	26.65
			✓		96.7	23.758
				✓	96.4	31.310
MG SVM	✓				94.0	47.595
		✓			94.2	26.691
			✓		94.6	24.946
				✓	94.4	31.589
F KNN	✓				93.4	18.707
		✓			93.5	9.502

(Continued)

Table 2 (continued)

Classifiers	Features				Measures	
	EfficientNet	Feature selection		Feature fusion	Accuracy (%)	Time (S)
		ABC	Entropy			
			✓		93.6	9.284
				✓	93.1	18.259

Figure 6: Confusion matrix for feature selection on cubic SVM classifier on Weizmann dataset

4.3 UCF Sports Dataset

The results of the UCF sports dataset have been described in this section. Table 3 presents the results of the UCF Sports dataset for all four experiments. In the first experiment, EfficientNet-based deep features are extracted and performed the classification. As a result, more than one classifier has been obtained the best accuracy of 100%, whereas the computational time of the LDA classifier is a minimum of 43.237 S. In the second step, ABC based optimization is performed, and selected the best features. The selected features are passed to the classifiers and obtain the best accuracy of 100%, whereas the time is reduced to 17.403 S. In the third experiment, Entropy-based best features were selected, and CSVM and FKNN obtained the best accuracy of 100%. In the last step, fusion is performed, and 100% accuracy is obtained, consistent with the other experiments but computationally slow. Moreover, Fig. 7 shows the LDA classifier's confusion matrix that can be utilized to verify the classification accuracy.

Table 3: Achieved results on UCF sports dataset

Classifiers	Features				Measures	
	EfficientNet	Feature selection		Feature fusion	Accuracy (%)	Time (S)
		ABC	Entropy			
LDA	✓				100	43.237
		✓			100	17.403
			✓		99.9	15.304
				✓	100	18.375
L SVM	✓				100	180.96
		✓			99.9	83.445
			✓		99.9	78.689
				✓	99.9	91.986
Q SVM	✓				100	229.52
		✓			99.9	104.78
			✓		100	92.583
				✓	100	115.79
C SVM	✓				100	234.38
		✓			100	113.88
			✓		99.9	94.788
				✓	99.9	122.41
MG SVM	✓				99.6	312.21
		✓			100	139.29
			✓		99.8	130.7
				✓	99.7	146.76
F KNN	✓				100	112.48
		✓			100	51.889
			✓		100	49.090
				✓	100	96.718

4.4 IXMAS Dataset

Results from the IXMAS dataset are displayed as a confusion matrix and as numerals in this section. In the first experiment, features are extracted from the original EfficientNet model and performed classification. As a result, the Fine KNN obtained the best accuracy of 96.7%, whereas the noted computational time is 189.79 S. The ABC optimizer is applied in the second experiment, and the best features are selected. The best-selected features are classified using several classifiers and obtained the best accuracy of 96.7%. As a result, this experiment's computational time is reduced to 97.538 S, previously 189.79 S. In the third experiment, entropy-based features were selected and

obtained the best accuracy of 97%, which improved, whereas the computational time was 88.911 S. In the last experiment, SbE-based fusion is performed and obtained the best accuracy of 96.9% (as seen in Table 4). This experiment consumed more time than the first three, but the accuracy was stable. In addition, the CSVM confusion matrix is shown in Fig. 8, and it can be used to check the proposed accuracy.

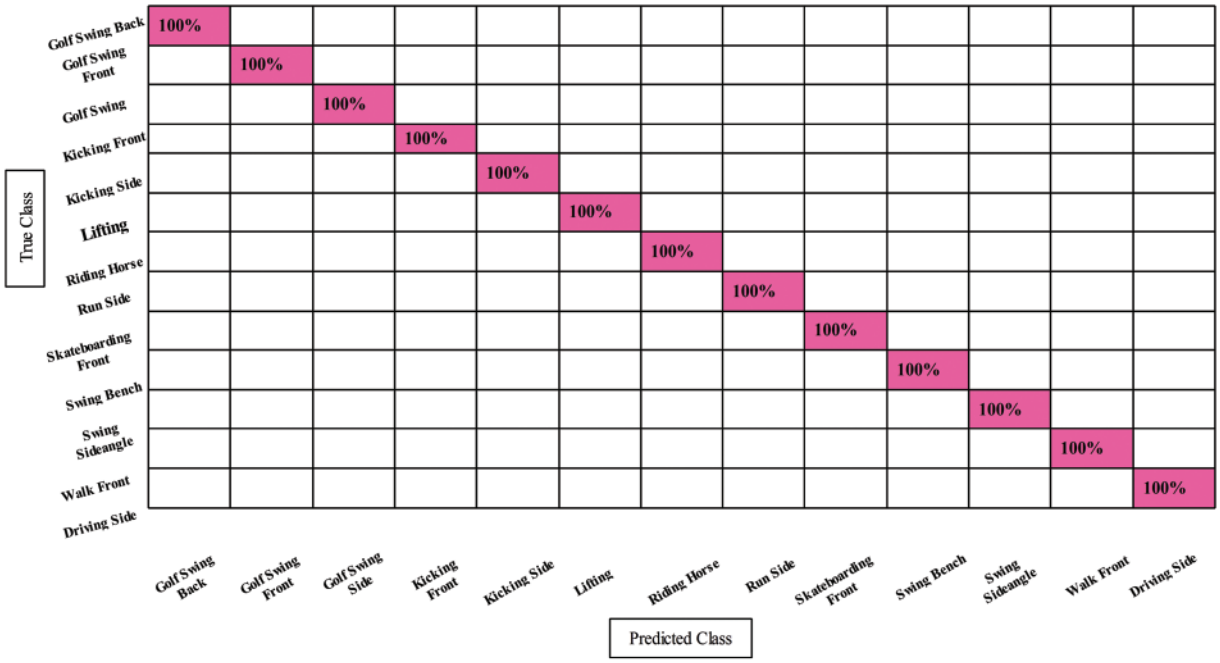

Figure 7: Confusion matrix for feature selection on liner discriminant classifier on UCF sports dataset

Table 4: Achieved results on the IXMAS dataset

Classifiers	Features				Measures	
	EfficientNet	Feature selection		Feature fusion	Accuracy (%)	Time (S)
		ABC	Entropy			
LDA	✓				87.7	57.952
		✓			81.0	18.525
			✓		81.3	19.134
				✓	84.9	20.506
L SVM	✓				86.1	424.6
		✓			82.9	190.79
			✓		81.8	173.56
				✓	84.5	202.01
Q SVM	✓				93.8	484.57
		✓			93.0	227.46
			✓		93.8	200.83
				✓	93.9	241.1

(Continued)

Table 4 (continued)

Classifiers	Features				Measures	
	EfficientNet	Feature selection		Feature fusion	Accuracy (%)	Time (S)
		ABC	Entropy			
C SVM	✓				95.4	542.75
		✓			95.1	250.0
			✓		95.4	209.4
				✓	95.7	253.03
MG SVM	✓				91.5	854.28
		✓			90.8	336.23
			✓		91.9	270.06
				✓	91.9	337.29
F KNN	✓				96.7	189.79
		✓			96.7	97.538
			✓		97.0	88.911
				✓	96.9	199.52

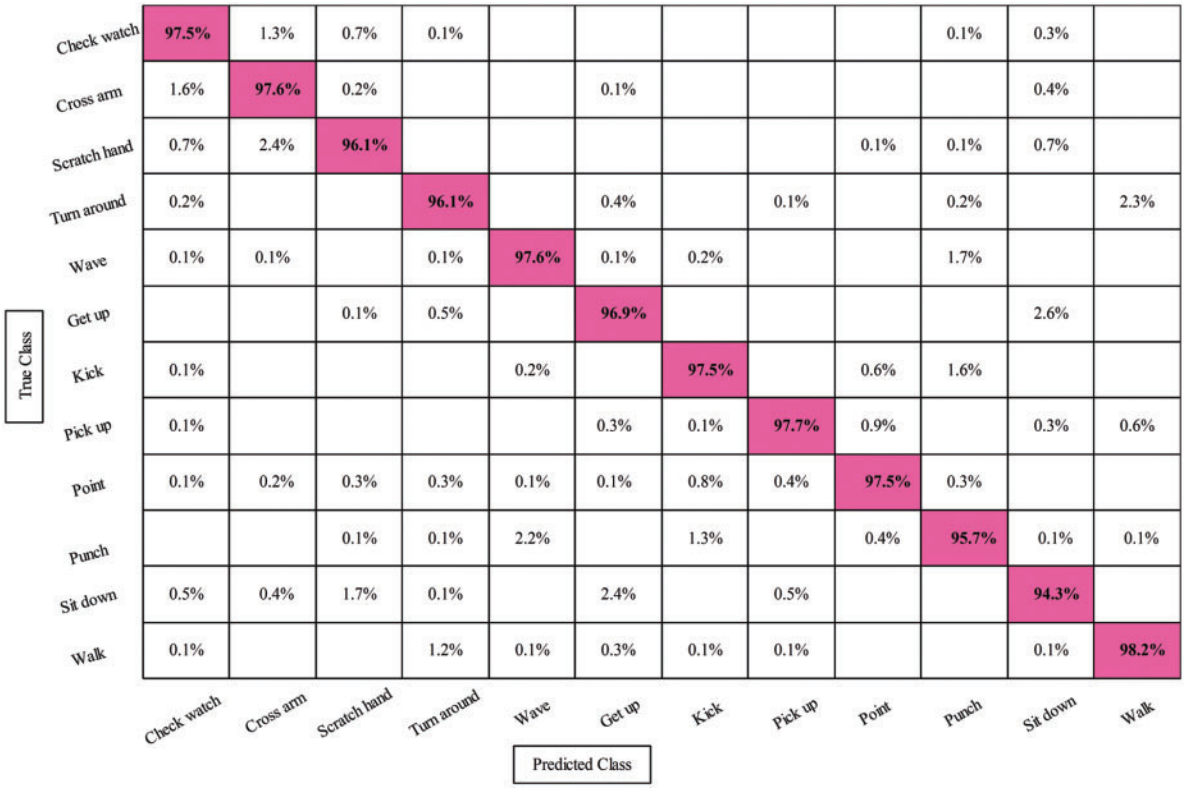

Figure 8: Confusion matrix for feature selection on fine KNN classifier on IXMAS dataset

4.5 UT Interaction Dataset

This section contains the findings from the UT Interaction dataset. Table 5 presents the results of the UT Interaction dataset for all four experiments. In the first experiment, EfficientNet-based deep features are extracted and performed the classification. The best-obtained accuracy for this experiment is 99.7% on the Fine KNN classifier, whereas the computational time is 16.643 S. In the second experiment, Fine KNN obtained the best 96.7% accuracy and the computational time of 7.343 S. From this, it is noted that the computational time is reduced, but accuracy is also dropped. In the third experiment, CSVM obtained the best accuracy of 99.6%, whereas the computational time was 11.113 S. This experiment's performance is better than the first two experiments. In the last experiment, fusion was performed and obtained the best accuracy of 99.7% with a computational time of 15.382 s. Overall, the CSVM performed well for this dataset. Fig. 9 shows this dataset's confusion matrix that can be utilized to verify the accuracy of Fine-KNN after the fusion process.

Table 5: Classification accuracy of UT interaction dataset

Classifiers	Features				Measures	
	EfficientNet	Feature selection		Feature fusion	Accuracy (%)	Time (S)
		ABC	Entropy			
LDA	✓				99.0	23.085
		✓			98.9	4.675
			✓		98.6	4.803
				✓	99.0	10.536
L SVM	✓				97.2	26.947
		✓			96.2	10.709
			✓		96.3	9.830
				✓	96.7	15.121
Q SVM	✓				99.6	23.779
		✓			99.4	12.473
			✓		99.2	10.737
				✓	99.6	15.355
C SVM	✓				99.7	24.739
		✓			99.5	11.887
			✓		99.6	11.113
				✓	99.7	15.382
MG SVM	✓				98.1	27.884
		✓			98.2	12.915
			✓		98.1	10.556
				✓	98.5	15.342
F KNN	✓				99.7	16.643
		✓			99.7	7.343
			✓		99.4	6.842
				✓	99.6	14.667

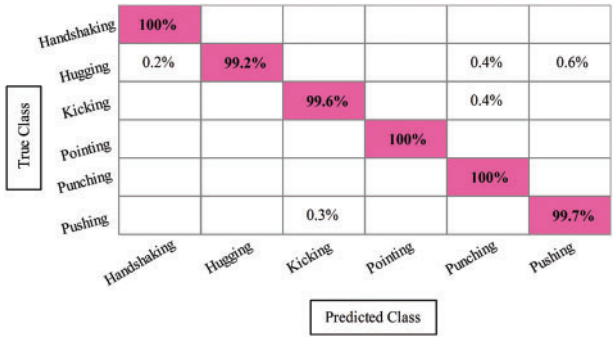

Figure 9: Confusion matrix of fine KNN classifier on UT interaction dataset

Finally, a thorough comparison is made with current methods, as shown in Table 6. In this table, several methods are listed; for each method, it is noted that they used several classifiers. Finally, we only use relevant data sets to compare the proposed accuracy. It can be seen from the accuracy values listed in this table that the proposed HAR framework has demonstrated increased accuracy.

Table 6: Comparison of the proposed method's accuracy with the existing techniques

Reference	Dataset/Technique	Accuracy (%)
Masmoudi et al. [2]	PCA net	
	KTH	87.33
	UCF sports	92.67
	UCF II (youtube)	81.40
	Weizmann	90
Ramya et al. [14]	Entropy base feature selection	
	KTH	91.4
	UCF 50	80
	Weizmann	92.5
Abdelbaky et al. [26]	PCA net	
	KTH	93.33
	UCF sports	90
Guha et al. [27]	Cooperative genetic algorithm	
	UCF II (youtube)	84.06
	HMDBI 51	53.87
	UCI HAR	95.79
	KTH	100
	Weizmann	86.75
Kumar et al. [28]	Gated recurrent neural networks	
	KTH	96.72
	UCF sports	89.98
	UCF 101	90.31

(Continued)

Table 6 (continued)

Reference	Dataset/Technique	Accuracy (%)
Afza et al. [13]	Optical flow algorithm	
	Weizmann	97.9
	KTH	100
	UCF sports	94.5
	UCF II (youtube)	99.3
Zhang et al. [10]	26-layered DCNN	
	HMDBI 51	81.4
	UCF sports	99.2
	KTH	98.3
	Weizmann	98.7
Khan et al. [9]	SBE approach	
	KTH	99.3
	WVU	99.8
	IXMAS	97.4
	Hollywood	99.9
Muhamad et al. [29]	Dilated CNN/LTSM	
	UCF II (youtube)	98.3
	UCF Sports	99.1
	j-HMDB	80.2
KTH		98.7%
UT Interaction		99.7%
Weizmann		96.7%
UCF Sports		100%
IXMAS		97.0%

5 Conclusion

Action recognition has been gaining popularity in recent years due to its vast range of real-life applications. In this work, we proposed a deep learning and fusion of optimized features framework for the classification of accurate action recognition. The proposed framework consists of several serial steps. The pre-trained EfficientNet deep model was fine-tuned and trained on the selected action datasets using deep transfer learning in the first step. Then, features are extracted from the average pooling layer and computed the results. Based on the computed results, we analyzed several redundant features. Therefore, we performed two feature selection techniques and selected the best features. Then, the selected features are classified, and improved accuracies are obtained for all selected datasets. Also, the time was significantly reduced, which was this framework's main strength. In the last, the fusion of selected features is performed to enhance the accuracy, but this step also increases the computational time, which is a drawback of this approach. In the future, we will consider this problem and propose a more optimized fusion approach.

Acknowledgement: Not applicable.

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

Author Contributions: Software: M.N and S.K; Methodology: M.N, S.K, and M.U.F; Validation: M.A and M.N; Supervision: M.U.F and U.A; Writting and Review: U.T, M.N, and S.K; Project Administration: U.A and U.T; Conceptualization: M.A and U.T; Verification: U.A and M.U.F; Funding: M.N, S.K, U.A, and M.U.F.

Availability of Data and Materials: The datasets used in this work are publically available for the research purpose.

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

References

[1] M. Mittal, L. M. Goyal and S. Roy, "A deep survey on supervised learning based human detection and activity classification methods," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 27867–27923, 2021.

[2] Y. Masmoudi, M. Ramzan, S. A. Khan and M. Habib, "Optimal feature extraction and ulcer classification from WCE image data using deep learning," *Soft Computing*, vol. 26, no. 5, pp. 7979–7992, 2022.

[3] A. Ben Hamida, M. Koubaa, H. Nicolas and C. B. Amar, "Video surveillance system based on a scalable application-oriented architecture," *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 17187–17213, 2016.

[4] 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. 2, pp. 2555–2576, 2022.

[5] I. M. Nasir, M. Raza, J. H. Shah and M. A. Khan, "HAREDNet: A deep learning based architecture for autonomous video surveillance by recognizing human actions," *Computers and Electrical Engineering*, vol. 99, no. 2, pp. 107805, 2022.

[6] G. Marques, D. Agarwal and D. Torre, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Sensors*, vol. 96, no. 2, pp. 106691, 2020.

[7] S. Choi, C. Fang, D. Haddad and M. Kim, "Predictive modeling of charge levels for battery electric vehicles using CNN EfficientNet and IGTD algorithm," *Methods*, vol. 1, no. 1, pp. 1–21, 2022.

[8] Y. D. Zhang, M. Allison, S. Kadry 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. 4, no. 2, pp. 1–16, 2021.

[9] S. Khan, M. Alhaisoni, U. Tariq and A. Armghan, "Human action recognition: A paradigm of best deep learning features selection and serial based extended fusion," *Sensors*, vol. 21, no. 4, pp. 7941, 2021.

[10] Y. D. Zhang, S. A. Khan, M. Attique and S. Seo, "A resource conscious human action recognition framework using 26-layered deep convolutional neural network," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 35827–35849, 2021.

[11] M. E. Issa, A. M. Helmi, M. A. Al-Qaness and R. Damaševičius, "Human activity recognition based on embedded sensor data fusion for the Internet of Healthcare Things," *Healthcare*, vol. 21, no. 3, pp. 1084, 2022.

[12] A. Mehmood, S. Kadry, N. A. Almujally, M. Alhaisoni and J. Balili, "TS2HGRNet: A paradigm of two stream best deep learning feature fusion assisted framework for human gait analysis using controlled environment in smart cities," *Future Generation Computer Systems*, vol. 3, no. 7, pp. 1–23, 2023.

[13] 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, pp. 104090, 2021.

[14] P. Ramya and R. Rajeswari, "Human action recognition using distance transform and entropy based features," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8147–8173, 2021.

[15] H. Basak, R. Kundu, P. K. Singh and R. Sarkar, "A union of deep learning and swarm-based optimization for 3D human action recognition," *Scientific Reports*, vol. 12, no. 4, pp. 1–17, 2022.
[16] M. Fu, Q. Zhong and J. Dong, "Sports action recognition based on deep learning and clustering extraction algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, pp. 1–21, 2022.
[17] C. Liang, J. Lu and W. Q. Yan, "Human action recognition from digital videos based on deep learning," in *2022 the 5th Int. Conf. on Control and Computer Vision*, NY, USA, pp. 150–155, 2022.
[18] R. Yue, Z. Tian and S. Du, "Action recognition based on RGB and skeleton data sets: A survey," *Neurocomputing*, vol. 4, no. 2, pp. 1–21, 2022.
[19] S. Nigam, R. Singh, M. K. Singh and V. K. Singh, "Multiview human activity recognition using uniform rotation invariant local binary patterns," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 5, pp. 1–19, 2022.
[20] 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. 5, pp. 2259–2322, 2021.
[21] G. Yao, T. Lei and J. Zhong, "A review of convolutional-neural-network-based action recognition," *Pattern Recognition Letters*, vol. 118, no. 21, pp. 14–22, 2019.
[22] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Int. Conf. on Machine Learning*, NY, USA, pp. 6105–6114, 2019.
[23] 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. 22, pp. 2873–2908, 2022.
[24] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 4, pp. 1345–1359, 2009.
[25] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm and Evolutionary Computation*, vol. 36, no. 5, pp. 27–36, 2017.
[26] A. Abdelbaky and S. Aly, "Two-stream spatiotemporal feature fusion for human action recognition," *The Visual Computer*, vol. 37, no. 4, pp. 1821–1835, 2021.
[27] R. Guha, A. H. Khan, P. K. Singh and D. Bhattacharjee, "CGA: A new feature selection model for visual human action recognition," *Neural Computing and Applications*, vol. 33, no. 4, pp. 5267–5286, 2021.
[28] B. S. Kumar, S. V. Raju and H. V. Reddy, "Human action recognition using a novel deep learning approach," *Materials Science and Engineering*, vol. 5, no. 7, pp. 012031, 2021.
[29] K. Muhammad, A. Ullah, A. S. Imran, M. Sajjad and G. Sannino, "Human action recognition using attention based LSTM network with dilated CNN features," *Future Generation Computer Systems*, vol. 125, no. 7, pp. 820–830, 2021.