sensors

MDPI

Article

Human Action Recognition: A Paradigm of Best Deep Learning Features Selection and Serial Based Extended Fusion

Seemab Khan [1], Muhammad Attique Khan [1,*], Majed Alhaisoni [2], Usman Tariq [3], Hwan-Seung Yong [4], Ammar Armghan [5] and Fayadh Alenezi [5]

1 Department of Computer Science, HITEC University Taxila, Txila 47080, Pakistan; seemab.khan@hitecuni.edu.pk
2 College of Computer Science and Engineering, University of Ha'il, Ha'il 55211, Saudi Arabia; majed.alhaisoni@gmail.com
3 College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, Al-Kharaj 11942, Saudi Arabia; u.tariq@psau.edu.sa
4 Department of Computer Science & Engineering, Ewha Womans University, Seoul 120-750, Korea; hsyong@ewha.ac.kr
5 Department of Electrical Engineering, College of Engineering, Jouf University, Sakakah 72311, Saudi Arabia; aarmghan@ju.edu.sa (A.A.); fshenezi@ju.edu.sa (F.A.)
* Correspondence: attique.khan@hitecuni.edu.pk

Abstract: Human action recognition (HAR) has gained significant attention recently as it can be adopted for a smart surveillance system in Multimedia. However, HAR is a challenging task because of the variety of human actions in daily life. Various solutions based on computer vision (CV) have been proposed in the literature which did not prove to be successful due to large video sequences which need to be processed in surveillance systems. The problem exacerbates in the presence of multi-view cameras. Recently, the development of deep learning (DL)-based systems has shown significant success for HAR even for multi-view camera systems. In this research work, a DL-based design is proposed for HAR. The proposed design consists of multiple steps including feature mapping, feature fusion and feature selection. For the initial feature mapping step, two pre-trained models are considered, such as DenseNet201 and InceptionV3. Later, the extracted deep features are fused using the Serial based Extended (SbE) approach. Later on, the best features are selected using Kurtosis-controlled Weighted KNN. The selected features are classified using several supervised learning algorithms. To show the efficacy of the proposed design, we used several datasets, such as KTH, IXMAS, WVU, and Hollywood. Experimental results showed that the proposed design achieved accuracies of 99.3%, 97.4%, 99.8%, and 99.9%, respectively, on these datasets. Furthermore, the feature selection step performed better in terms of computational time compared with the state-of-the-art.

Keywords: human action recognition; deep learning; features fusion; features selection; recognition

Citation: Khan, S.; Khan, M.A.; Alhaisoni, M.; Tariq, U.; Yong, H.-S.; Armghan, A.; Alenezi, F. Human Action Recognition: A Paradigm of Best Deep Learning Features Selection and Serial Based Extended Fusion. *Sensors* **2021**, *21*, 7941. https://doi.org/10.3390/s21237941

Academic Editors: Tomasz Krzeszowski, Adam Świtoński, Michal Kepski and Carlos Tavares Calafate

Received: 4 November 2021
Accepted: 25 November 2021
Published: 28 November 2021

1. Introduction

Human action recognition (HAR) emerged as an active research area in the field of computer vision (CV) in the last decade [1]. HAR has applications in various domains including; surveillance [2], human-computer interaction (HCI) [3], video reclamation, and understanding of visual information [4], etc. The most important application of action recognition is video surveillance [5]. Governments use this application for intelligence gathering, reducing crime rate, for security purposes [6], or even crime investigation [7]. The main motivation of growing research in HAR is due to its use in video surveillance applications [8]. In visual surveillance, HAR plays a key role in recognizing the activities of subjects in public places. Furthermore, these types of systems are also useful in smart cities surveillance [9].

Human actions are of various types. These actions can be categorized into two broad classes, namely voluntary actions and involuntary actions [10]. Manual recognition of

these actions in real-time is a tedious and error-prone task; therefore, many CV techniques are introduced in the literature [11,12] to serve this task. Most of the proposed solutions are based on classical techniques such as shape features, texture features, point features, and geometric features [13]. A few techniques are based on the temporal information of the human [14], and a few of them extract human silhouettes before feature extraction [15].

Recently, deep learning has shown promising results in the field of computer vision (CV) [16]. Deep learning makes learning and data representation at multiple levels by mimicking the human brain processing [17] to create models. These models consist of multiple processing layers such as convolutional, ReLu, pooling, fully connected, and Softmax [18]. The functionality of a CNN model is to replicate the working of the human brain as it preserves and makes sense of multidimensional information. There exist multiple methods in deep learning, which include encompassing neural networks, hierarchical probabilistic models, supervised learning, and unsupervised learning models [19].

The HAR process is a challenging task as there are a variety of human actions in daily life. In order to tackle this challenge, deep learning models are utilized. The performance of a deep learning model is always based on the number of training samples [20]. In the action recognition tasks, several datasets are publicly available. These datasets include several actions such as walking, running, leaving a car, waving, kicking, boxing, throwing, falling, bending down, and many more.

Recently proposed systems mainly focus on the hybrid techniques; however, they do not focus on minimizing the computational time [21]. This is an important factor as most time surveillance is performed in real-time. Some of the other key challenges of HAR are as follows: (i) Query video sequences resolution is imperative for the recognition of the focal point in the most recent frame. The background complexity, shadows, lighting conditions, and outfit conditions extract irrelevant information using classical techniques of human action, which later results in inefficient action classification; (ii) with automatic activities recognition under multi-view cameras it is difficult to classify the correct human activities. Change in the motion variation captures the wrong activities under the multi-view cameras; (iii) imbalanced datasets impact the learning of a CNN. A CNN model always needs a massive number of training images for learning; and (iv) features extraction from the entire video sequences includes several irrelevant features, affecting the classification accuracy.

These challenges are considered in this work to propose a fully automated design using deep learning features fusion and best feature selection for HAR under the complex video sequences. The major contributions of this work are summarized as follows:

- Selected two pre-trained deep learning models and removed the last three layers. The new layers are added and trained on the target datasets (action recognition dataset). In the training process, the first 80% of the layers are frozen instead of using all the layers, whereas the training process was conducted using transfer learning.
- Proposed a Serial based Extended (SbE) approach for multiple deep learning features fusion. This approach fused features in two phases for better performance and to reduce redundancy.
- Proposed a feature selection technique named Kurtosis-controlled Weighted KNN (KcWKNN). A threshold function is defined which is further analyzed using a fitness function.
- Performed an ablation study to investigate the performance of each step in terms of advantages and disadvantages.

The rest of the manuscript is organized as follows: Related work is presenting in Section 2. The proposed design for HAR is presented in Section 3, which includes deep learning models, transfer learning, the selection of best features and fusion. Results of the proposed method are presented in Section 4 in terms of tables and confusion matrixes. Finally, Section 5 concludes this work.

2. Related Work

HAR has emerged as an impactful research area in CV from the last decade [22]. It is based on important applications such as visual surveillance [23], robotics, biometrics [24,25], and smart healthcare centers to name a few [26,27]. Several researchers of computer vision developed techniques using machine learning [28] for HAR. Most of these researches focused on deep learning due to its better performance and few of them used barometric sensors for activity recognition [29]. Rasel et al. [30] extracted the spatial features using acidometer sensors and classified using multiclass SVM for final activity recognition. Zhao et al. [31] introduced a combined framework for activity recognition. They combined short-term and long-term features for the final results. Khan et al. [32] combined the attention-based LSTM network with dilated CNN model features for the action recognition. Similarly, a skeleton based attention framework is presented by [33] for action recognition. Maheshkumar et al. [13] presented an HAR framework using both the shape and the OFF features [34]. The presented framework is the combination of Hidden Markov Model (HMM) and SVM. The shape and OFF features are extracted and used for HAR through the HMM classifier. The multi-frame averaging method was adopted for background extraction of the image. A discrete Fourier transform (DFT) was performed to reduce the magnitude on the length feature set from the middle to the body contour. In order to select features, the principal component analysis was implied. The presented framework was tested on videos recorded in real-time settings and achieved maximum accuracy. Weifeng et al. [35] presented a generalized Laplacian Regularized Sparse Coding (LRSC) framework for HAR. It was a nonlinear generalized version of graph Laplacian with a tighter isoperimetric inequality. A fast-iterative shrinkage thresholding algorithm for the optimization of ϱ-LRSC was also presented in this work. The input of the sparse codes learned by the ϱ-LRSC algorithm were placed into the support vector machine (SVM) for final categorization. The datasets used for the experimental process were unstructured social activity attribute (USAA) and HMDB51. The experimental results demonstrated the competence of the presented ϱ-LRSC algorithm. Ahmed et al. [36] presented an HAR model using a depth video analysis. HMM was employed to recognize regular activities of aged people living without any attendant. The first step was to analyze the depth maps through the temporal motion identification method using the segments of human silhouettes in a given scenario. Robust features were selected and fused together to find the gradient orientation change, intensity difference temporal and local movement of the body organs [37]. These fused features were processed via embedded HMM. The experimental process was conducted on three different datasets such as Online Self-Annotated [38], Smart Home, and Three Healthcare, and achieved the accuracies 84.4, 87.3, and 95.97%, respectively. Muhammed et al. [39] presented a smartphone inertial sensors-based framework for human activity recognition. The presented framework was divided into three steps: (i) extract the efficient features; (ii) the features were reduced using the kernel principal component analysis (KPCA) and linear discriminant analysis (LDA) to make them resilient; (iii) resultant features were trained via deep belief neural networks (DBN) to attain improved accuracy. The presented approach was compared with traditional expression recognition approaches such as typical multiclass SVM [40,41] and artificial neural network (ANN) and showed an improved accuracy.

Lei et al. [42] presented a light weight action recognition framework based on DNN using RGB video sequences. The presented framework was constructed using CNNs and LTSM units that was a temporal attention model. The purpose of using CNNs was to segment out the objects from the complex background. LTSM networks were used on spatial feature maps of multiple CNN layers. Three datasets, such as UCF-11, UCF Sports, and UCF-101, were used for experimental processes and achieved 98.45%, 91.89%, and 84.10%, respectively. Abdu et al. [43] presented an HAR framework based on deep learning. They considered the problem of traditional techniques which are not useful for the better accuracy of complex activities. The presented framework used a cross DBNN model that unites the SRUs with GRUs of the neural network. The SRUs were used to execute the

sequence multi-modal data input. Then GRUs were used to store and learn the amount of information that can be transferred from past state to future state. Zan et al. [44] presented an action recognition model that served the problem of multi-view HAR. The presented algorithm was based on adaptive fusion and category-level dictionary learning (AFCDL). In order to integrate dictionary learning, query sets were designed, and the regularization scheme was constructed for the adaptive weights assignment. Muhammad et al. [45] presented a new framework of 26-layered CNN for composite action classification. Two layers, the global average pooling layer and fully connected layer (FC) were used for feature extraction. The extracted features are classified using the extreme learning machine (ELM) and Softmax for final action classification. Four datasets named HMDB51, UCF Sports, KTH, and Weizmann were used for the experimentation process and showed better performance. Muhammad et al. [4] presented a new fully automated structure for HAR by fusing DNN and multi-view features. Initially, a pre-trained CNN named VGG19 was implied to take out DNN features. Horizontal and vertical gradients were used to compute multi-view features and vertical directional attributes. Final recognition was performed on the selected features via the Naive Bayes Classifier (NBC). Kun et al. [46] introduced an HAR model based on DNN that combines the convolutional layer with LSTM. The presented model was able to automatically extract the features and perform their classification with the standard parameters.

Recently, the development of deep learning models for HAR using high dimensional datasets has shown immense progress. Classical methods for HAR did not show satisfactory performance, especially for large datasets. In contrast, the modern techniques such as Long Short-Term Memory (LSTM), SV-GCN, and Convolution Neural Networks (CNNs) are showing improved performance and can be considered for further research to obtain an improvement in the accuracy.

3. Proposed Methodology

This section presents the proposed methodology for human action recognition in complex video sequences. The proposed design consists of multiple steps, including feature mapping, feature fusion, and feature selection. Figure 1 represents the proposed design of HAR. In this design, features are extracted from the two pre-trained models such as DenseNet201 and InceptionV3. The extracted deep features are fused using the Serial based Extended (SbE) approach. In the later step, the best features are selected using Kurtosis-controlled Weighted KNN. The selected features are classified using several supervised learning algorithms. Detail of each step is provided below.

3.1. Convolutional Neural Network (CNN)

CNN is an innovative technique in deep learning that makes the classification process fast and precise. CNN requires lesser parameters to train compared with the traditional neural networks [47]. A CNN model contains multiple layers where the convolution layer is an integral part. Few other layers contained in the CNN model are pooling layers (min, max, average), the ReLU layer, and some fully connected (FC) layers. The internal structure of a CNN has multiple layers as presented in Figure 2. This figure shows that video sequences are provided as input to this network. In the network, the initially convolutional layer is added to convolve input image features, which are later normalized in pooling and hidden layers. After that, FC layers are added to convert image features into 1D feature vector. The final 1D extracted features are classified in the last layer, which is known as the output layer.

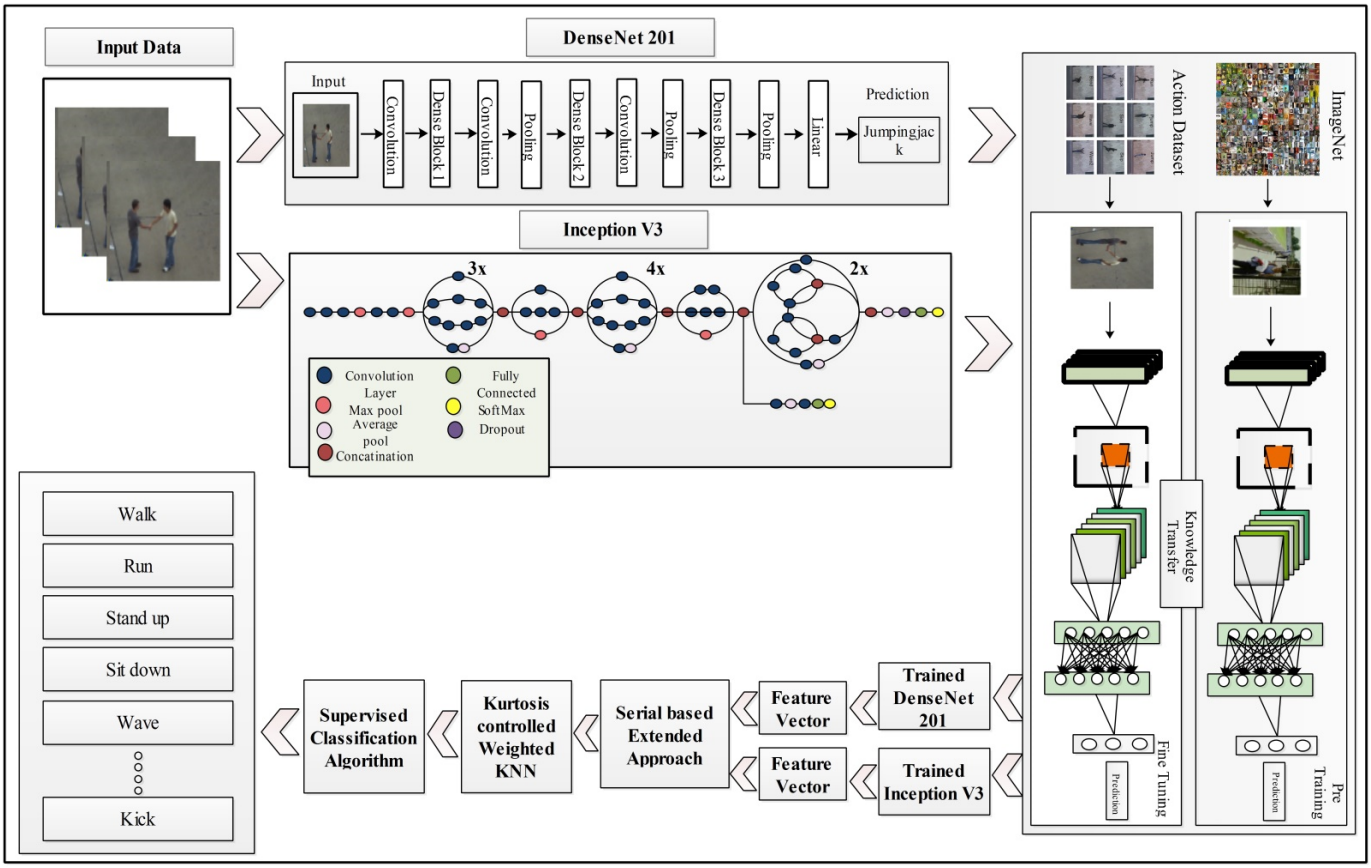

Figure 1. Illustration of a proposed design for HAR using deep learning.

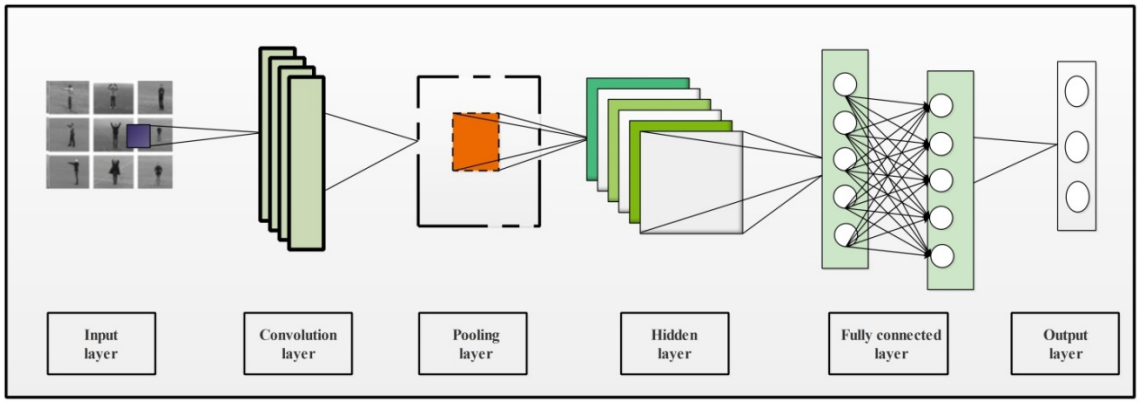

Figure 2. A simple architecture of CNN containing multiple layers for image classification.

3.2. Densenet201 Pre-Trained Deep Model

DenseNet is an advanced CNN model where every layer is directly connected with all the layers in subsequent order. These connections help to improve the flow of information in the network, as illustrated in Figure 3. This dense connectivity makes it a dense convolutional network commonly known as DenseNet [48]. Other than the improvement in the information flow, it caters to the vanishing gradient problems as well as it strengthens the feature prorogation process. DenseNet also allows for reusing the features and it reduces required parameters, which eventually reduces the computational complexity of the algorithm. Consider a CNN with ϕ number of layers and ϕ_l layer index has an input stream that starts with x_0. A nonlinear transformation function $F_\phi(.)$ is applied on each layer and it can be a combination of multiple functions such as BN, pooling convolution or

ReLU. In a densely connected network, each layer is connected to its subsequent layers. Output of the ϕ^{th} layer is represented by x_ϕ.

$$x_\phi = F_\phi\left(x_0, \ldots\ldots, x_{\phi-1}\right) \quad (1)$$

where $\left(x_0, \ldots\ldots, x_{\phi-1}\right)$ states the concatenation of the feature maps generated in layers $0, \ldots\ldots.., \phi - 1$.

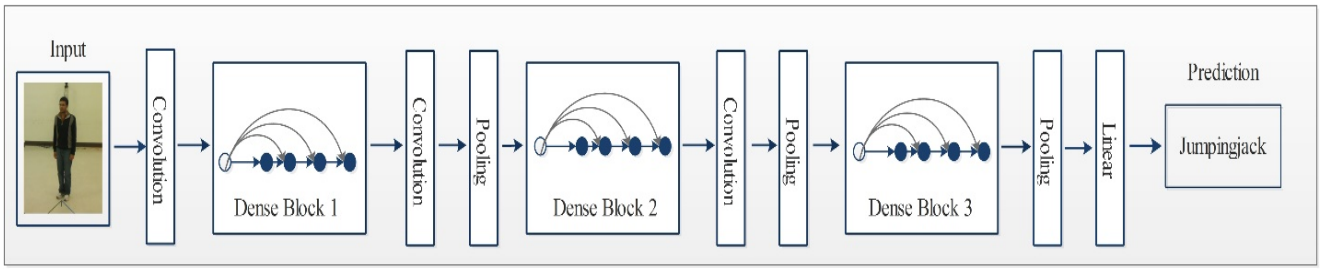

Figure 3. Network architecture of DenseNet201 for action recognition.

3.3. Inception V3 Pre-Trained Deep Model

InceptionV3 [49] is an already trained CNN model on the ImageNet dataset. It consists of 316 layers which include convolution layers, pooling layers, fully connected layers, dropout, and Softmax layers. The total number of connections in this model is 350. Unlike a traditional CNN that allows a fixed filter size in a single layer, InceptionV3 has the flexibility to use variable filter sizes and a number of parameters in a single layer which results in better performance. An architecture of InceptionV3 is shown in Figure 4.

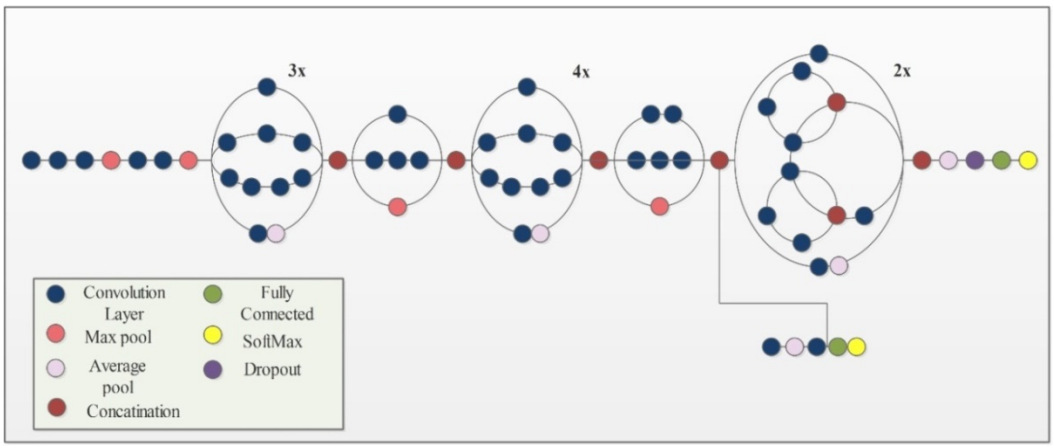

Figure 4. Network architecture of Inceptionv3 model.

3.4. Transfer Learning Based Learning

Transfer learning is a well-known technique in the field of deep learning that allows the reusability of a pre-trained model on an advanced research problem [50]. A major advantage of using TL is that it requires less data as input and provides remarkable results. It aims to transfer knowledge from a source domain to a targeted domain, here the source domain refers to a pre-trained model with a very large dataset and the targeted domain is the proposed problem with limited labels [51]. In the source domain, usually a large high-resolution image dataset known as ImageNet is used [52,53]. It contains more than 15 billion labels and 1000 image categories. Image labels in ImageNet are saved according to the wordNet hierarchy, where each node leads to thousands of images belonging to that category. Mathematically, TL is defined as follows:

Given a source domain s_d, defined as:

$$s_d = \left\{ \left(x_1^d, y_1^d\right), \ldots, \left(x_i^d, y_i^d\right), \ldots, \left(x_n^d, y_n^d\right) \right\}$$

The learning task is L_d,L_s,$\left(x_m^d, y_m^d\right) \in \varphi$. The target domain is defined as:

$$s_t = \left\{ \left(x_1^t, y_1^t\right), \ldots, \left(x_i^t, y_i^t\right), \ldots, \left(x_n^t, y_n^t\right) \right\}$$

The learning task L_t, $\left(x_n^t, y_n^t\right) \in \varphi$, (m, n) will be the size of training data, where $n \ll m$ and y_i^d and y_i^t are the training data labels. Using this definition, both pre-trained models are trained on action datasets. During the training process, the learning rate was 0.01, the mini batch size is 64, the maximum epochs is 100 and the learning method is the stochastic gradient descent. After the fine-tuning process, the output of both models is the number of action classes.

3.5. Features Extraction

Features are extracted from the newly learned models called target models as shown in Figures 5 and 6. Figure 5 represents a DenseNet201 modified model. Using this model, features are extracted using the avg-pool layer. In the output, an $N \times 1920$ dimensional feature vector was obtained, denoted by $\vec{C}$, where N represents number of images in the target dataset.

Using the Inception V3 modified model (depicted in Figure 6), features are extracted from the average pool layer. On this layer, the dimension of the extracted deep feature vector is $N \times 2048$ and it is represented by $\vec{D}$, where N is the number of images in the target dataset.

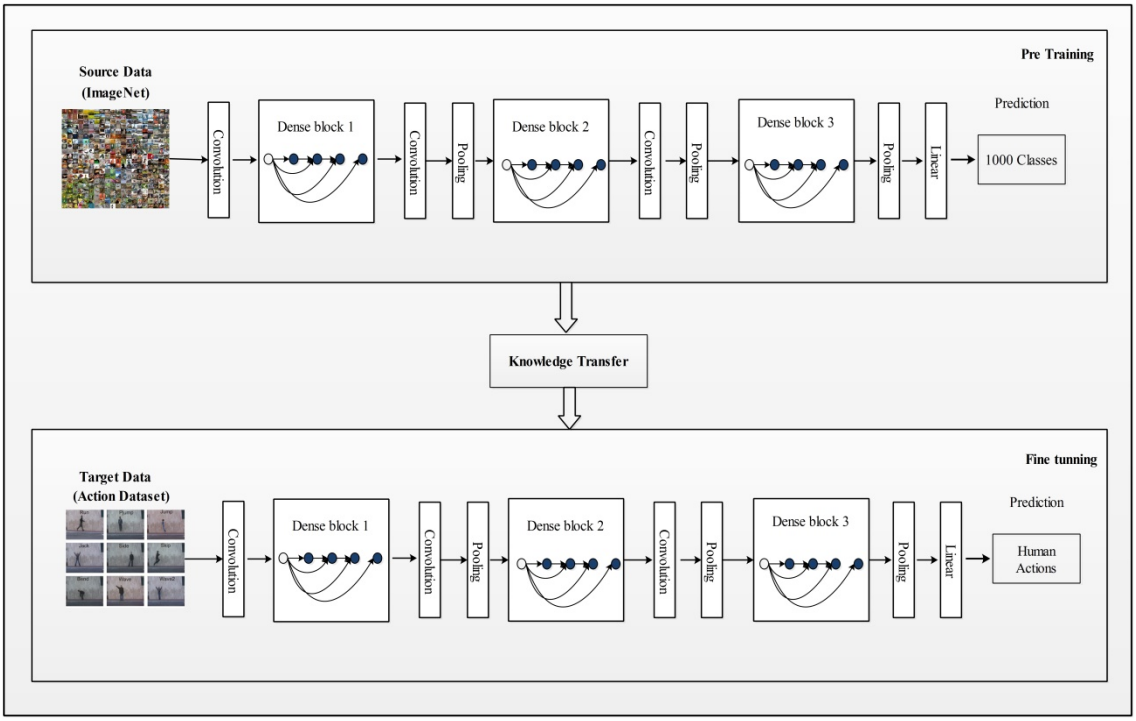

Figure 5. Target model (modified DenseNet201) for feature extraction.

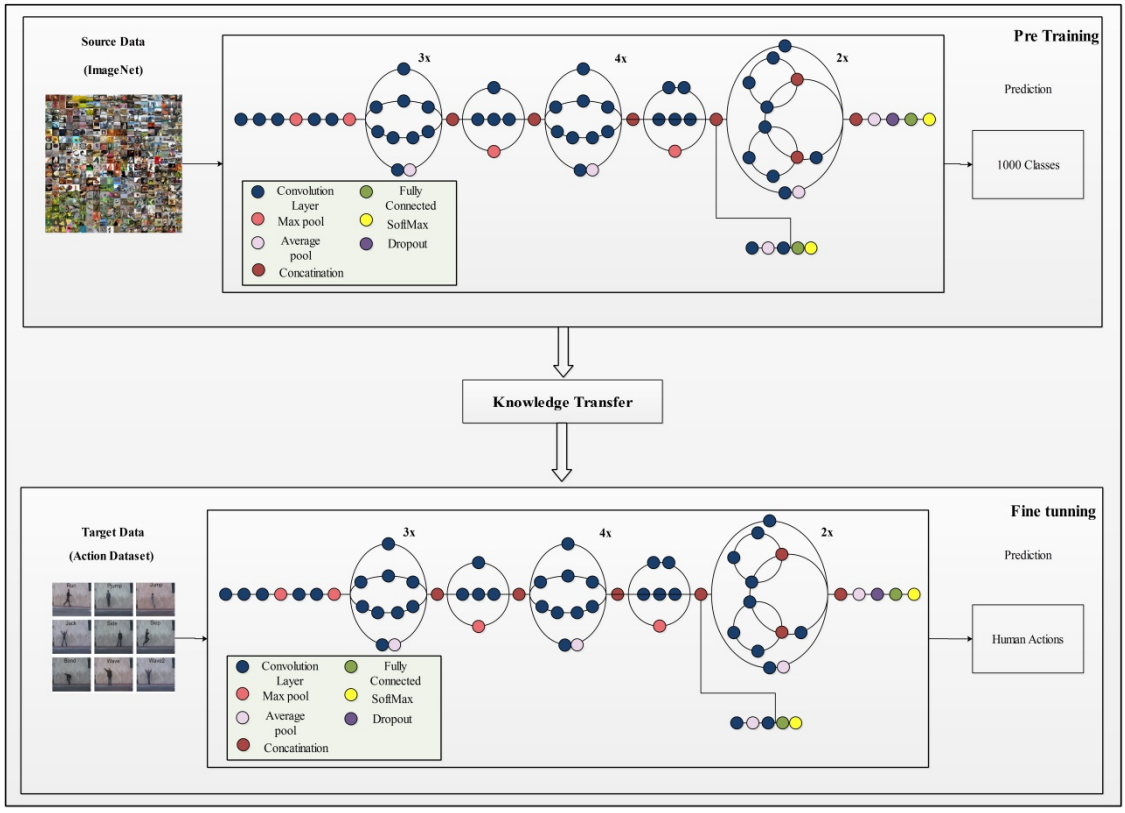

Figure 6. Target model (modified Inception V3) for feature extraction.

3.6. Serial Based Extended Fusion

The fusion of features is becoming a popular technique for improved classification results. The main advantage of this step is to improve the image information in terms of features. The improved feature space increases the classification performance. In the proposed work, a Serial based Extended (SbE) approach is implemented. In this approach, initially features are fused using a serial-based approach. The fused vectors are combined in a single feature vector and to obtain a feature vector of dimension $N \times 3968$ and denoted by β, considering two feature vectors $\vec{C}$ and $\vec{D}$ defined on the outline of sample space $\vec{Z}$. For an arbitrary sample $\delta \in \vec{Z}$, the equivalent two feature vectors are $j \in \vec{C}$ and $k \in \vec{D}$. The serial combined feature of δ can be defined as $= \begin{pmatrix} j \\ k \end{pmatrix}$. If feature vector $\vec{C}$ has n dimensions and feature vector $\vec{D}$ has m dimensions, then serial fused feature βwill have $(n + m)$ dimensions. After obtaining a βfeature vector, the features are sorted into descending order and the mean value is computed. Based on the mean value, the feature vector is extended in terms of the final fusion.

$$\mu() = \frac{1}{\mathbb{N}} \sum_{i=1}^{\mathbb{N}} (_i) \tag{2}$$

$$Fsn = \begin{cases} Fusion(i) & for\ _i \geq \mu \\ Discard, & ElseWhere \end{cases} \tag{3}$$

Here, $Fusion(i)$ is a final fused feature vector of dimension $N \times K$, where the value of K is always transformed according to the variation in the dataset. Later on, this fusion vector is analyzed using the experimental process and further refined using a feature selection approach.

3.7. Serial Based Extended Fusion

Feature selection is the process of the selection of subset features from the input feature vector [54]. It helps to improve the performance of the algorithm and also reduces the training time. In the proposed design, a new feature selection algorithm is proposed, Kurtosis-controlled Weighted KNN (KcWKNN). The proposed selection method works in the following steps: (i) input fused feature vector; (ii) compute Kurtosis value; (iii) define a threshold function; (iv) calculate fitness, and (v) select the feature vector.

The Kurtosis value is computed as follows:

$$Kr = \frac{\mu_4}{\delta^4} \tag{4}$$

$$\mu_4 = E\left[\left(\widehat{F}_i - E\left[\widehat{F}\right]\right)^n\right], \ \widehat{F}_i \in Fusion(i) \text{ and } n = 4 \tag{5}$$

$$\delta^4 = \sqrt{E\left[\left(\widehat{F}_i - \mu\right)^2\right]} \tag{6}$$

where K is the Kurtosis function, μ_4 is the fourth central moment, and δ is the standard deviation. Kurtosis is a statistical measure that we investigate to find how much the tails of the distribution deviate from the normal. Distributions with higher values are identified in this process. In this work, the main purpose of using Kurtosis is to obtain the higher tail values (outlier features) through the fourth moment that was later employed in the threshold function for the initial feature selection. By using the Kurtosis value, a threshold function is defined as follows:

$$Ts = \begin{cases} FS(i) & for \quad Fusion(i) \geq Kr \\ Ignore, & Elsewhere \end{cases} \tag{7}$$

The selected feature vector $FS(i)$ is passed into the fitness function WKNN for validation. Mathematically, WKNN is defined as follows:

Consider $\{(x_i, y_i)\}_{i=1}^{N} \in P$ as the training set where x_i is the p-dimensional training vector and y_i is its equivalent class labels set. To determine the label $\overline{y}$ of any $\overline{x}$ from the test set $(\overline{x}, \overline{y})$, the following mathematics takes place.

(a) Compute the Euclidian distance e between $\overline{x}$ and each $(\overline{x}, \overline{y})$, formal given in Equation (8).

$$e(\overline{x}, \ x_i) = \overline{x} - x_{ii_o} \tag{8}$$

(b) Arrange all values in ascending order

(c) Assign a weight $\acute{\omega}_i$ to the ith nearest neighbor using Equation (9).

$$\acute{\omega}_i = \frac{1}{(e(\overline{x}, \ x_i))^2} \tag{9}$$

(d) Assign $\acute{\omega}_i = 1$ for the equally weighted KNN rule,

(e) The class label of $\overline{x}$ is assigned on the basis of majority votes from the neighbors by Equation (10).

$$\overline{y} = \text{argmax} \sum_{(x,y)\in P} \acute{\omega}_i, \ ^{\eta}(x = \overline{y}_i) \tag{10}$$

where x is the class label, $\overline{y}_i$ is the class label for ith nearest neighbor and $^{\eta}(.)$ is the Dirac-Delta function that takes value = 1 if its argument is true and 0 otherwise.

(f) Compute error.

The error is used as a performance measure, where the number of iterations is initialized as 50. This process is carried out until the error is minimized. Visually, the flow is shown in Figure 7, where it can be seen that the best selected features are finally classified

using supervised learning algorithms. Moreover, the complete work of the proposed design is listed in Algorithm 1.

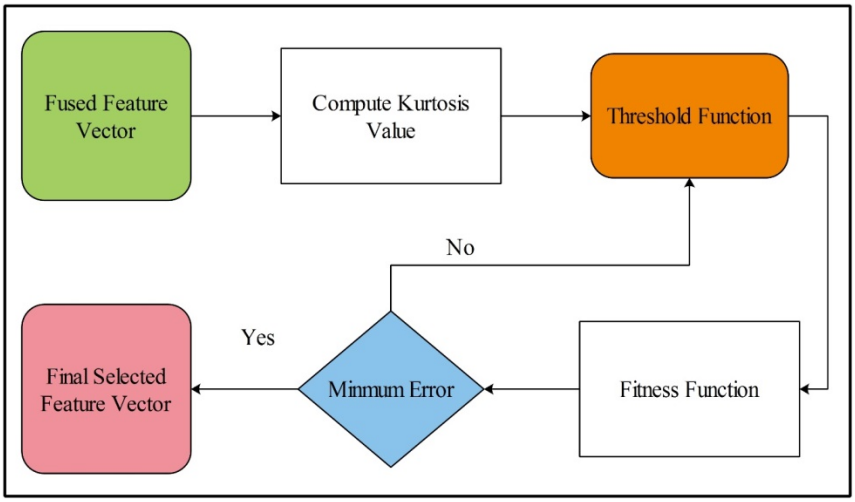

Figure 7. Proposed flow diagram of best feature selection.

Algorithm 1. The complete work of the proposed design.

Input: Action Recognition Datasets
Output: Predicted Action Class
Step 1: Input action datasets
Step 2: Load Pre-trained Deep Models;
- Densenet201
- Inception V3

Step 3: Fine Deep Models
Step 4: Trained Deep Models using TL
Step 5: Feature Extraction from Avg Pooling Layers
Step 6: SbE approach for Features Fusion
Step 7: Best Features Selection using Proposed KcWKNN
Step 8: Predict Action Label

4. Results and Analysis

The experimental process of the proposed method is presented in this section. Four publically available datasets such as KTH [3], Hollywood [38], WVU [39], and IXMAS [40] were used in this work for the experimental process. Each class of these datasets contains 10,000 video frames that are utilized for the experimental process. In the experimental process, 50% of video sequences are used for the training purpose, while the remaining 50% is utilized for the testing purpose. The K-Fold cross validation is adopted, where the value of $K = 10$. Results are computed on several supervised learning algorithms and select the best one is selected based on the accuracy value. All simulations are conducted on MATLAB2020a using a Personal Computer Corei7 with 16 GB of RAM and 8 GB Graphics card.

4.1. Results

A total of four experiments were performed on each dataset to analyze the performance of the middle step. These steps are: (i) performed classification using DenseNet201 deep features; (ii) performed classification using InceptionV3 deep model; (iii) performed classification using the SbE deep features fusion, and (iv) performed classification using KcWKNN-based feature selection.

Experiment 1: Table 1 presents the results of the specific DenseNet201 deep features on selected datasets. In this table, it is noted that the Cubic SVM achieved a better accuracy

of 99.3% on the KTH dataset. Other classifiers also achieved a better accuracy of above 94%. For the Hollywood action dataset, the best achieved accuracy is 99.9% for Fine KNN. Similar to the KTH dataset, the rest of the classifiers also performed better on this dataset. The best obtained accuracy for the WVU dataset is 99.8% for Cubic SVM. The rest of the classifiers also performed better and achieved an average accuracy of 97%. The best obtained accuracy of the IXAMAS dataset is 97.3% for Fine KNN.

Table 1. Classification accuracy on specific DenseNet201 deep model. The bold represents the best obtained values.

Classifier	Datasets Accuracy on DenseNet201 Deep Model			
	KTH	**Hollywood**	**WVU**	**IXAMAS**
Linear Discriminant	98.8	99.6	98.3	92.1
Linear SVM	98.0	98.3	97.1	86.6
Quadratic SVM	98.9	99.6	99.7	96.4
Cubic SVM	**99.3**	99.8	**99.8**	95.4
Medium Gaussian SVM	98.6	99.5	97.8	93.1
Fine KNN	98.7	**99.9**	99.3	**97.3**
Medium KNN	96.7	98.8	97.3	88.0
Cosine KNN	96.9	98.8	97.4	88.3
Weighted KNN	97.2	99.7	98.0	92.9
Ensemble Bagged Trees	89.6	98.2	94.5	82.9

Experiment 2: The results of InceptionV3 deep features are provided in Table 2. In this table, it is noted that the best achieved accuracy on the KTH dataset is 98.1%, for the Hollywood dataset it is 99.8%, for the WVU dataset it is 99.1%, and for the IXAMAS dataset it is 96%. From this table, it is observed that the performance of specific DenseNet201 features are better. However, during the computation of results, time significantly increases. Therefore, it is essential to handle this issue with consistent accuracy.

Table 2. Classification accuracy on specific InceptionV3 deep model. The bold represents the best obtained values.

Classifier	Datasets Accuracy on DenseNet201 Deep Model			
	KTH	**Hollywood**	**WVU**	**IXAMAS**
Linear Discriminant	96.6	98.8	96.5	87.3
Linear SVM	95.4	96.3	93.5	81.3
Quadratic SVM	97.6	99.3	99.0	92.1
Cubic SVM	**98.1**	99.5	**99.1**	93.6
Medium Gaussian SVM	97.0	99.3	97.7	91.2
Fine KNN	97.6	**99.8**	98.4	**96.0**
Medium KNN	95.00	98.1	94.8	83.8
Cosine KNN	95.6	98.5	95.1	84.7
Weighted KNN	95.9	99.1	95.8	90.0
Ensemble Bagged Trees	89.0	92.4	90.5	73.3

Experiment 3: After the experiments on specific feature sets, the SbE approach is applied for deep features fusion. The KTH dataset results are provided in Table 3. In this table, The highest performance is recorded for Cubic SVM with an accuracy of 99.3%. Recall

and precision are 99.3% and 99.43% respectively. Moreover, the noted time during the training process is 893.23 s. The second highest accuracy is achieved by a linear discriminant classifier of 99.2%. The rest of the classifiers also performed better. Compared with specific feature vectors, the fusion process results are more consistent. Figure 8 illustrates the true positive rates (TPRs)-based confusion matrix of Cubic SVM that confirms the value of the recall rate. In this figure, the highlighted diagonal values represent the true positive predictions, whereas the values other than the diagonal represent false negative predictions.

Table 3. Achieved results on KTH dataset after fusion of deep features using SbE approach. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	99.200	99.300	0.80	424.10	99.249	99.2
Linear SVM	98.400	98.616	1.60	487.10	98.508	98.4
Quadratic SVM	99.150	98.283	0.85	706.56	98.714	99.2
Cubic SVM	**99.300**	**99.433**	**0.70**	893.23	**99.366**	**99.3**
Medium Gaussian SVM	98.916	99.083	1.08	1445.8	98.999	98.9
Fine KNN	99.083	99.216	0.91	450.55	99.149	99.1
Medium KNN	96.700	97.233	3.30	447.37	96.965	96.8
Cosine KNN	97.516	97.716	2.48	459.33	97.616	97.5
Weighted KNN	97.483	97.916	2.51	447.59	97.699	97.6
Ensemble Bagged Trees	94.233	94.733	5.76	**192.96**	94.482	94.3

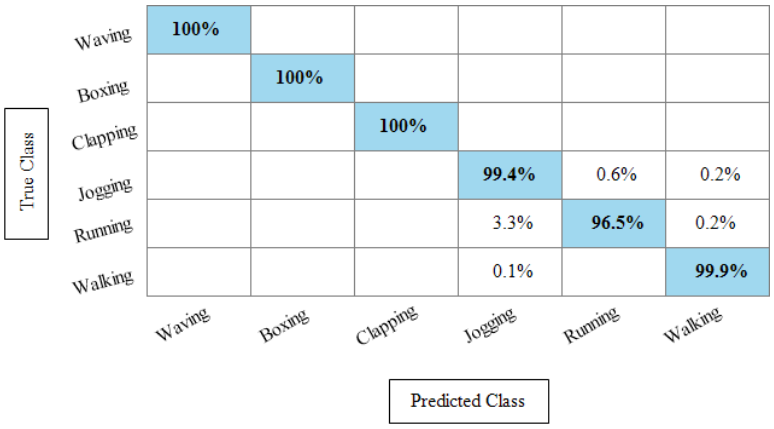

Figure 8. TPR-based confusion matrix of KTH dataset after fusion of deep features using SbE approach.

Table 4 represents the results of the Hollywood action dataset using the SbE approach. In this table, it is noted that the best accuracy is 99.9%, obtained by Fine KNN. Other performance measures such as recall rate, precision rate and F1 score values are 99.1825%, 99.8375%, and 99.5089%, respectively. The rest of the classifiers mentioned in this table performed better and achieved an average accuracy above 98%. Figure 9 illustrates the TPR-based confusion matrix of Fine KNN, where it is clear that each class prediction rate is above 99%. Moreover, compared with the specific deep features experiment on the Hollywood dataset, the fusion process shows more consistent results.

Table 4. Achieved results on Hollywood dataset after fusion of deep features using SbE approach. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	99.775	99.825	0.22	469.75	99.800	99.9
Linear SVM	99.887	99.25	1.11	734.42	99.567	99.2
Quadratic SVM	99.550	99.725	0.45	1065.4	99.637	99.7
Cubic SVM	99.575	99.775	0.42	1337.4	99.674	99.8
Medium Gaussian SVM	99.287	99.675	0.71	2227.1	99.480	99.7
Fine KNN	**99.182**	**99.837**	**0.18**	**447.76**	**99.508**	**99.9**
Medium KNN	98.500	99.0125	1.50	437.47	98.755	99.1
Cosine KNN	99.037	98.975	0.96	449.13	99.006	99.3
Weighted KNN	99.250	99.45	0.75	439.29	99.349	99.6
Ensemble Bagged Trees	94.425	97.562	5.57	209.63	95.968	96.7

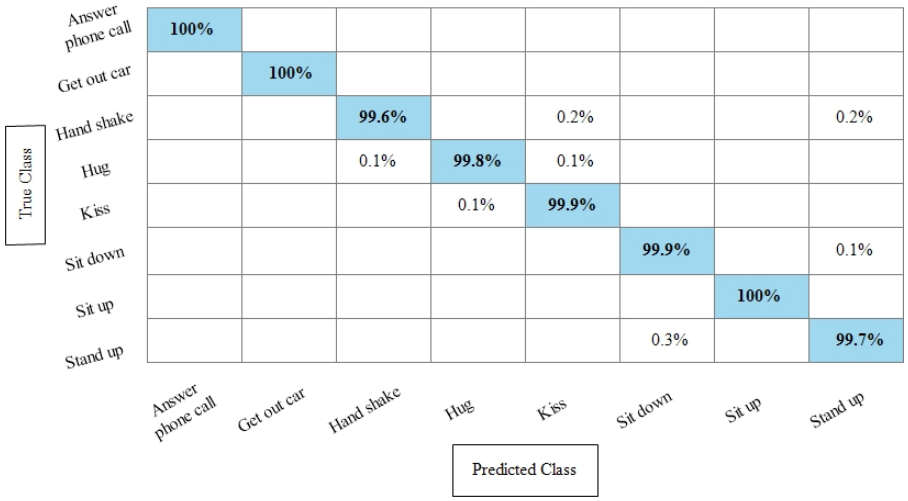

Figure 9. TPR based confusion matrix of Fine KNN using Hollywood dataset after fusion of deep features through SbE approach.

Table 5 presents the results of the WVU dataset using the SbE fusion approach. The highest accuracy is achieved through Linear Discriminant which is 99.8%, where the recall rate, precision rate, and F1 score are 99.79%, 99.78%, and 99.78%, respectively. Quadratic SVM and Cubic SVM performed second best and achieved an accuracy of 99.7% for each. The rest of the classifiers also performed better and gained the average accuracy of above 99%. Figure 10 illustrated the TPR based confusion matrix of the WVU dataset for the Linear Discriminant classifier. This figure showed that the correct prediction rate of each classifier is more than 99%. Compared with this accuracy of WVU on specific features, it is noticed that the fusion process provides consistent accuracy.

Table 5. Achieved results on WVU dataset after fusion of deep features using SbE approach. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR (%)	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	**99.79**	**99.78**	**0.21**	**2073.1**	**99.785**	**99.8**
Linear SVM	97.74	97.77	2.26	2567.7	97.755	97.7
Quadratic SVM	99.56	99.56	0.44	2824.5	99.560	99.6
Cubic SVM	99.56	99.57	0.44	2267	99.565	99.6
Medium Gaussian SVM	98.56	98.34	1.66	2749	98.449	98.3
Fine KNN	97.0	97.03	3.00	3486	97.015	97.0
Medium KNN	87.15	88.34	12.8	3933.5	87.741	87.2
Cosine KNN	87.98	89.01	12.1	2825.4	88.492	88.0
Weighted KNN	90.89	91.51	9.11	2716.7	91.198	90.9
Ensemble Bagged Trees	94.08	94.12	5.92	965.78	94.100	94.1

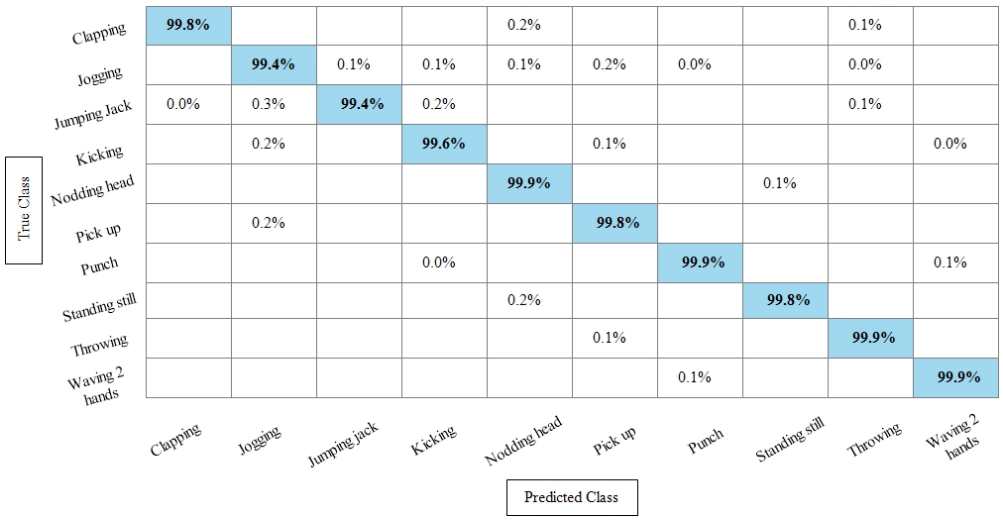

Figure 10. TPR-based confusion matrix of Linear Discriminant classifier after fusion of deep features using SbE approach.

Table 6 presents the results of the IXMAS dataset after SbE features fusion. In this table, it can be seen that the highest accuracy is achieved through Fine KNN of 97.4%, where the recall rate, precision rate, and F1 score are 97.18%, 97.25%, and 97.21%, respectively. Cubic SVM performed second best and achieved an accuracy of 97.3%. The rest of the classifiers also performed better and attained an average accuracy above 93%. Figure 11 illustrates the TPR-based confusion matrix of the Fine KNN for the IXMAS dataset using the SbE approach.

Overall, the results of the SbE approach are improved and are consistent compared with the specific deep features (see results in Tables 1 and 2). However, it is observed that the computational time increases during the fusion process. For a real-time system, this time needs to be minimized. Therefore, a feature selection approach is proposed.

Table 6. Achieved results on IXMAS dataset after fusion of deep features using SbE approach. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR (%)	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	96.460	96.310	3.54	508.35	96.384	96.5
Linear SVM	91.030	91.230	8.97	1428	91.129	91.3
Quadratic SVM	96.670	96.680	3.33	936.8	96.675	96.7
Cubic SVM	97.216	97.225	2.78	390.9	97.220	97.3
Medium Gaussian SVM	96.016	96.066	3.98	840.3	96.041	96.1
Fine KNN	**97.180**	**97.250**	**2.82**	**570.56**	**97.215**	**97.4**
Medium KNN	88.360	88.890	11.6	560.06	88.624	88.9
Cosine KNN	89.141	89.516	10.8	559.83	89.328	89.7
Weighted KNN	92.475	92.625	7.52	543.5	92.549	92.8
Ensemble Bagged Trees	80.291	81.550	19.7	284.31	80.915	81.4

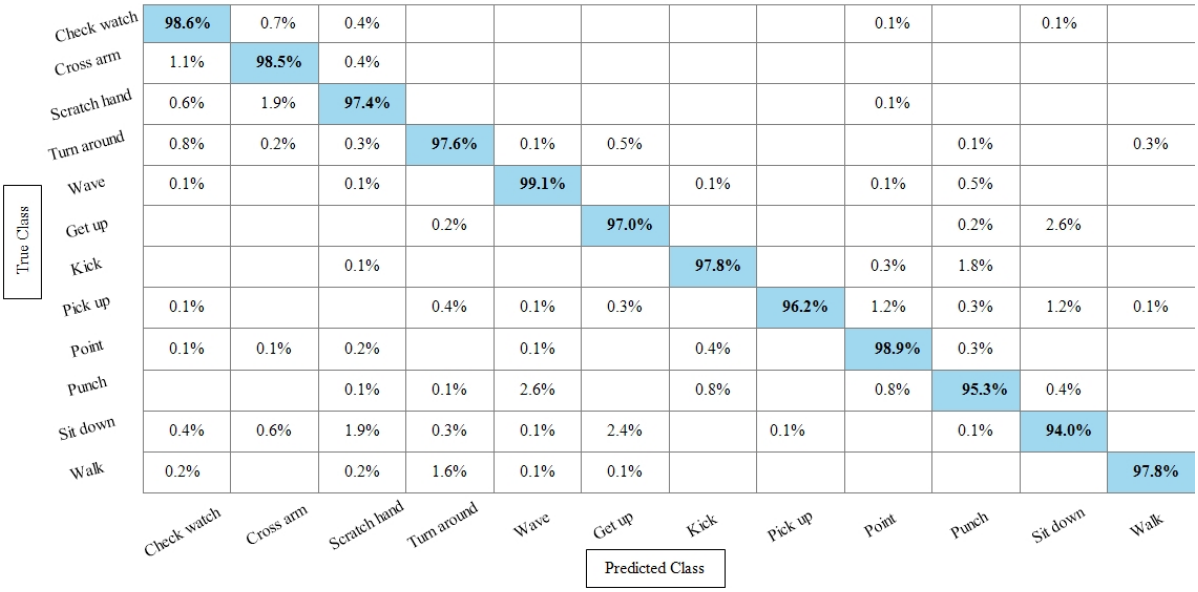

Figure 11. TPR-based confusion matrix of Fine KNN after fusion of deep features using SbE approach.

Experiment 4: In this experiment, the best features are selected using Kurtosis-controlled WKNN and provided to the classifiers. Results are provided in Tables 7–10. Table 7 presents the results of the proposed feature selection algorithm on the KTH dataset. In this table, the highest obtained accuracy is 99%, achieved by Cubic SVM. Other performance measures such as recall, precision and F1 score are 98.1666%, 99.1166% and 99.016%, respectively. Figure 12 illustrates the TPR-based confusion matrix of the Cubic SVM for the best feature selection process. In comparison with Table 3 results, it is noted that the accuracy of Cubic SVM decreases (0.3%), while the computational time expressively declines. The computational time of the Cubic SVM in the fusion process was 893.23 s, which is reduced after the feature selection process to 451.40 s. This shows that the feature selection process not only maintains the recognition accuracy but also minimizes the computational time.

Table 7. Achieved results on KTH dataset after best feature selection using KcWKNN. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR (%)	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	98.080	98.516	1.92	87.805	98.297	98.1
Linear SVM	97.633	97.933	2.36	255.42	97.783	97.7
Quadratic SVM	98.600	98.866	1.40	360.10	98.733	98.7
Cubic SVM	**98.916**	**99.116**	**1.09**	**451.40**	**99.016**	**99.0**
Medium Gaussian SVM	98.2833	98.483	1.71	687.37	98.383	98.3
Fine KNN	98.616	98.833	1.38	237.93	98.724	98.7
Medium KNN	95.483	96.366	4.51	231.39	95.922	95.7
Cosine KNN	97.016	97.183	2.98	230.18	97.099	97.0
Weighted KNN	96.233	97.000	3.76	222.90	96.615	96.4
Ensemble Bagged Trees	94.150	93.716	5.8	140.57	93.632	94.2

Table 8. Achieved results on Hollywood dataset after best feature selection using KcWKNN. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR (%)	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	99.087	99.450	0.912	88.375	99.268	99.4
Linear SVM	97.937	98.687	2.062	323.99	98.311	98.6
Quadratic SVM	99.262	99.587	0.737	439.41	99.424	99.5
Cubic SVM	99.387	99.675	0.612	501.67	99.531	99.7
Medium Gaussian SVM	98.587	99.500	1.412	910.78	99.041	99.5
Fine KNN	**99.812**	**99.837**	**0.187**	**213.33**	**99.825**	**99.8**
Medium KNN	97.225	98.550	2.775	224.52	97.883	98.5
Cosine KNN	98.325	98.862	1.675	221.19	98.593	98.9
Weighted KNN	98.575	99.412	1.425	215.89	98.992	99.2
Ensemble Bagged Trees	87.050	94.287	12.95	126.72	90.524	97.7

Table 9. Achieved results on WVU dataset after best feature selection using KcWKNN. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR (%)	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	98.50	98.53	1.50	241.48	98.515	98.5
Linear SVM	96.51	96.57	3.49	293.2	96.539	96.5
Quadratic SVM	**99.37**	**99.38**	**0.63**	**1064.6**	**99.375**	**99.4**
Cubic SVM	99.43	99.44	0.57	1124.0	99.435	99.4
Medium Gaussian SVM	98.24	98.25	1.76	1363.7	98.245	98.2
Fine KNN	96.55	96.59	3.45	1365.1	96.570	96.5
Medium KNN	86.80	87.98	13.2	1322.0	87.386	86.8
Cosine KNN	87.61	88.73	12.39	1316.2	88.166	87.6
Weighted KNN	90.33	91.07	9.67	1236.8	90.698	90.3
Ensemble Bagged Trees	94.71	94.75	5.29	423.37	94.730	95.7

Table 10. Achieved results on IXMAS dataset after best feature selection using KcWKNN. The bold represents the best obtained values.

Classifier	Recall Rate (%)	Precision Rate (%)	FNR (%)	Time (s)	F1 Score (%)	Accuracy (%)
Linear Discriminant	91.583	91.516	8.41	119.8	91.549	91.7
Linear SVM	88.050	88.400	11.95	714.13	88.224	88.5
Quadratic SVM	95.008	95.083	4.99	634.7	95.045	95.1
Cubic SVM	95.783	95.866	4.21	239.4	95.824	95.9
Medium Gaussian SVM	94.466	94.933	5.53	475.5	94.699	94.6
Fine KNN	**97.075**	**96.991**	**2.92**	**290.69**	**97.033**	**97.1**
Medium KNN	86.383	86.925	13.61	266.24	86.653	86.9
Cosine KNN	88.066	88.233	11.93	270.74	88.149	88.5
Weighted KNN	90.975	91.966	9.02	263.74	91.468	91.2
Ensemble Bagged Trees	83.433	85.108	16.5	175.78	84.261	84.8

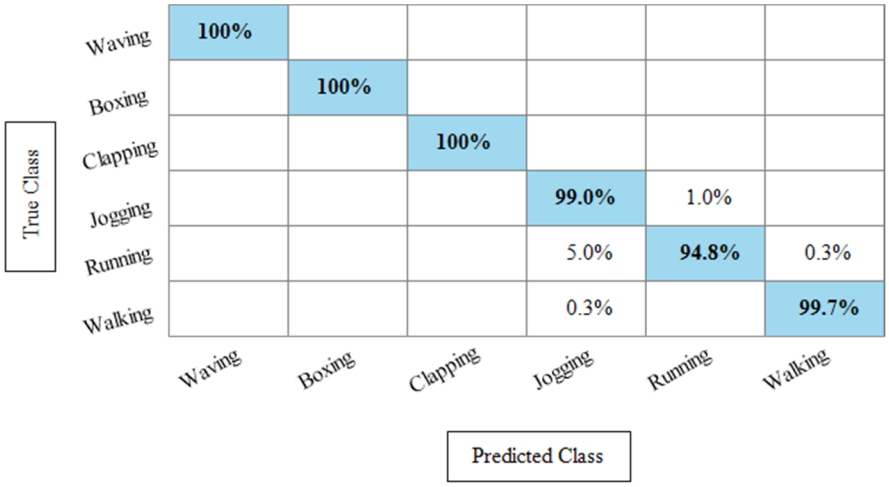

Figure 12. TPR based confusion matrix of Cubic SVM after best feature selection using KcWKNN.

Table 8 presents the best feature selection results on the Hollywood Action dataset and achieved best accuracy by Fine KNN of 99.8%. The other calculated measures such as recall rate, precision rate, and F1 Score are 99.812%, 99.837%, and 99.82%, respectively. For the rest of the classifiers, the average accuracy is above 98% (can be seen in this table). Figure 13 illustrates the TPR-based confusion matrix of Fine KNN for this experiment. The diagonal values in this experiment show the correct predicted values. Comparison with Table 4 shows that the classification accuracy is still consistent, whereas the computational time is significantly reduced. The computational time at the fusion process was 447.76 s, whereas after the selection process, it is reduced to 213.33 s. This shows that the selection of best features using KcWKNN performed significantly better.

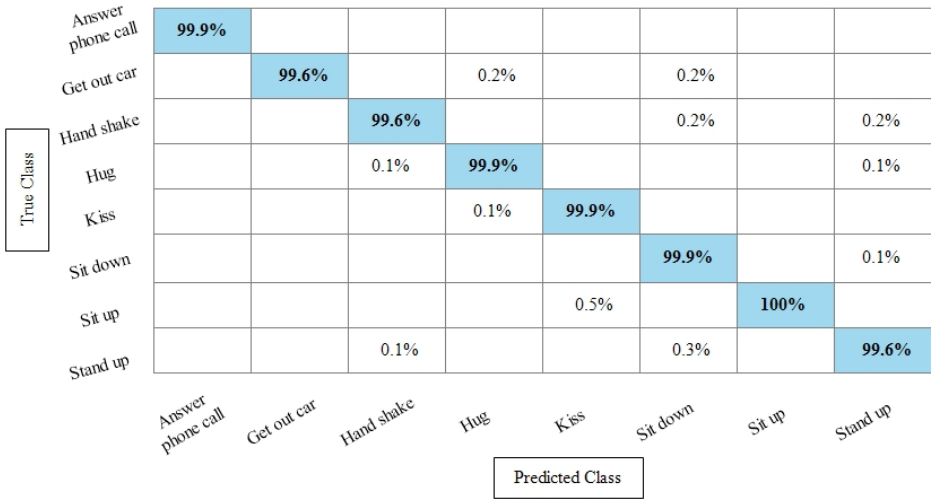

Figure 13. TPR based confusion matrix of Fine KNN after best feature selection using KcWKNN.

Table 9 presents the results of the WVU dataset after the best feature selection using KcWKK. In this table, Quadratic SVM and Cubic SVM performed best with the accuracy of 99.4%, where the recall rate is 99.37% and 99.43%, respectively, and the precision rate is 99.38% and 99.44%, respectively and the F1 score is 99.375%, and 99.43%, respectively. Figure 14 shows the TPR-based confusion matrix of the Cubic SVM for this experiment. This figure shows that the prediction rate of each class is above 99%. Moreover, in comparison with Table 5 (fusion results), the computational time of this experiment on the WVU dataset is almost half and accuracy is still consistent. This shows that the KcWKNN selection approach performed significantly well.

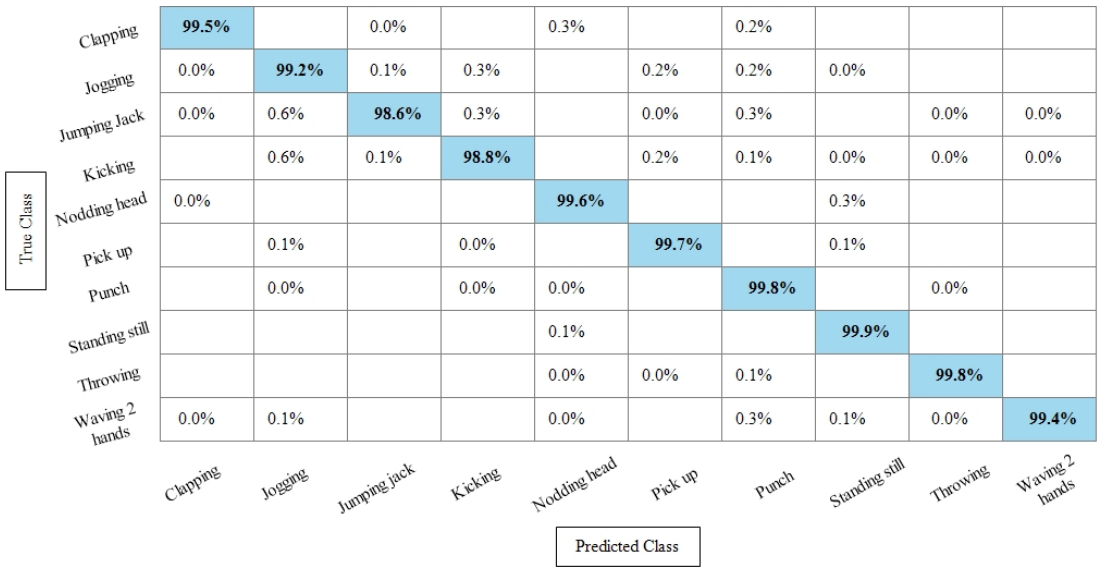

Figure 14. TPR-based confusion matrix of Cubic SVM after best feature selection using KcWKNN.

The results of the KcWKNN-based best features selection on the IXMAS dataset are provided in Table 10. In this table, it is noted that the Fine KNN attained best accuracy of 97.1%, whereas the recall rate, precision rate, and F1 score are 97.075%, 96.9916%, and 97.033%, respectively. Figure 15 illustrates the TPR-based confusion matrix of the Fine KNN for this experiment. The correct prediction value of each class is provided in the diagonal of this figure. Compared with Table 6, this experiment reduces the computational time while maintaining the recognition accuracy.

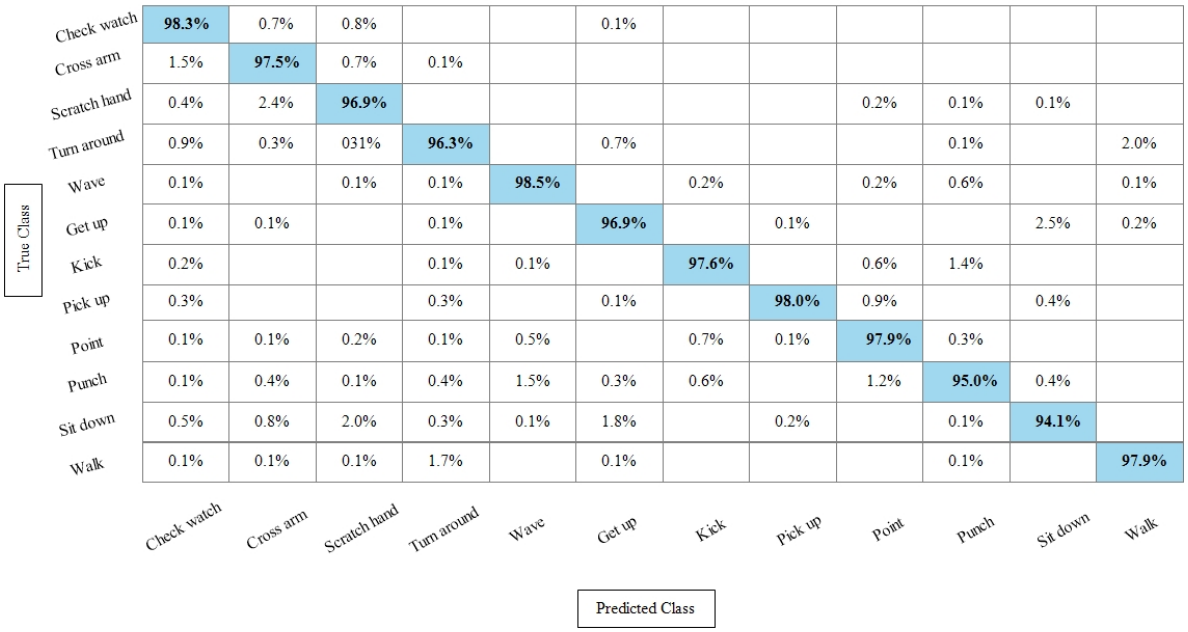

Figure 15. TPR-based confusion matrix of Fine KNN after best feature selection using KcWKNN.

Finally, a detailed analysis was conducted among all experiments in terms of accuracy and time. From Tables 1–10, it is observed that the accuracy value is improved after the proposed fusion process and the time is reduced. However, the noted time was still high and must be reduced further; therefore, a feature selection technique is proposed and time is significantly reduced compared with the original extracted deep features and fusion step (plotted in Figures 16–19). In the selection process, a little change occurred in the accuracy value, but on the other side, a high fall is noted in the computational time.

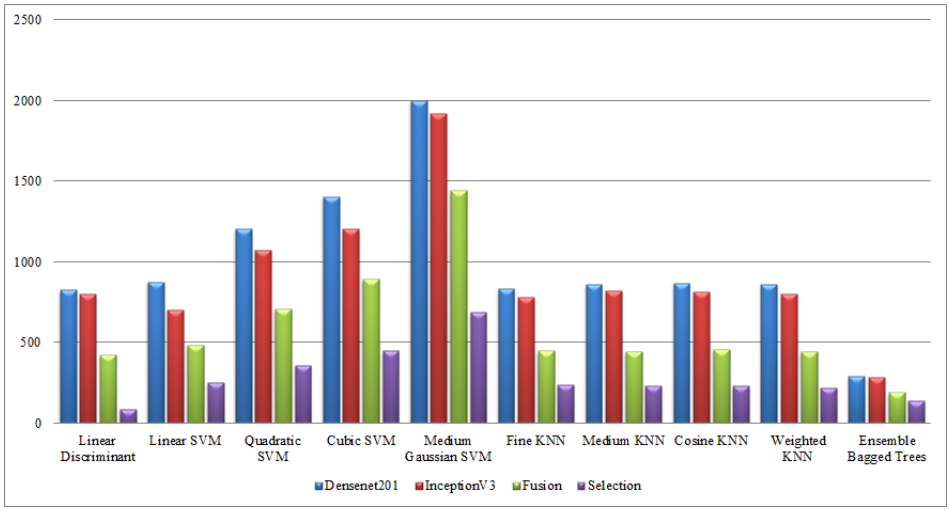

Figure 16. Computational time-based comparison of middle steps on KTH dataset.

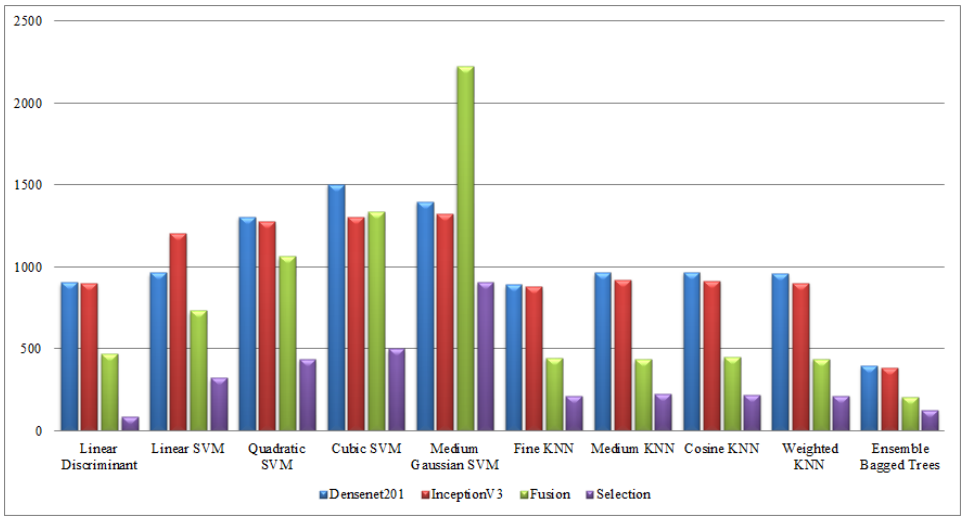

Figure 17. Computational time-based comparison of middle steps on Hollywood dataset.

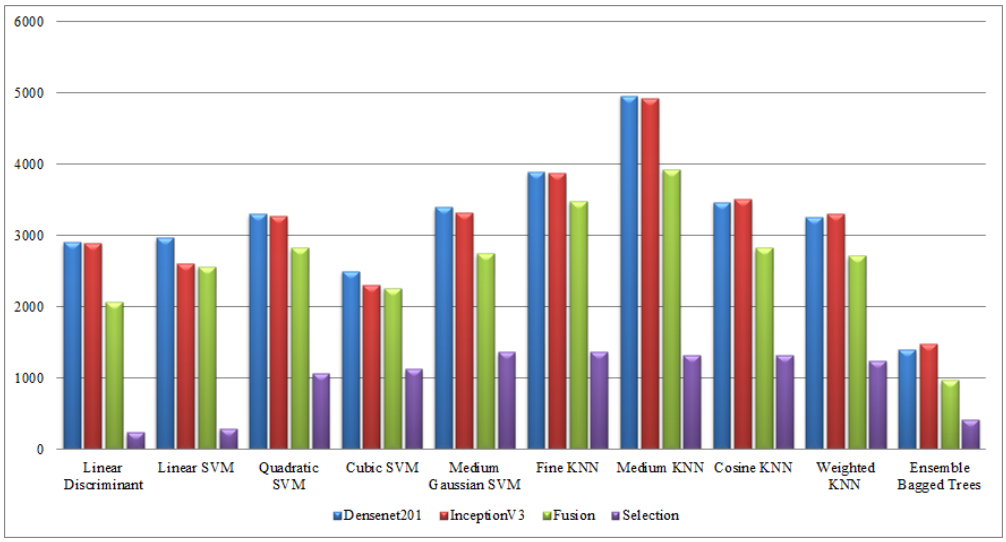

Figure 18. Computational time-based comparison of middle steps on WVU dataset.

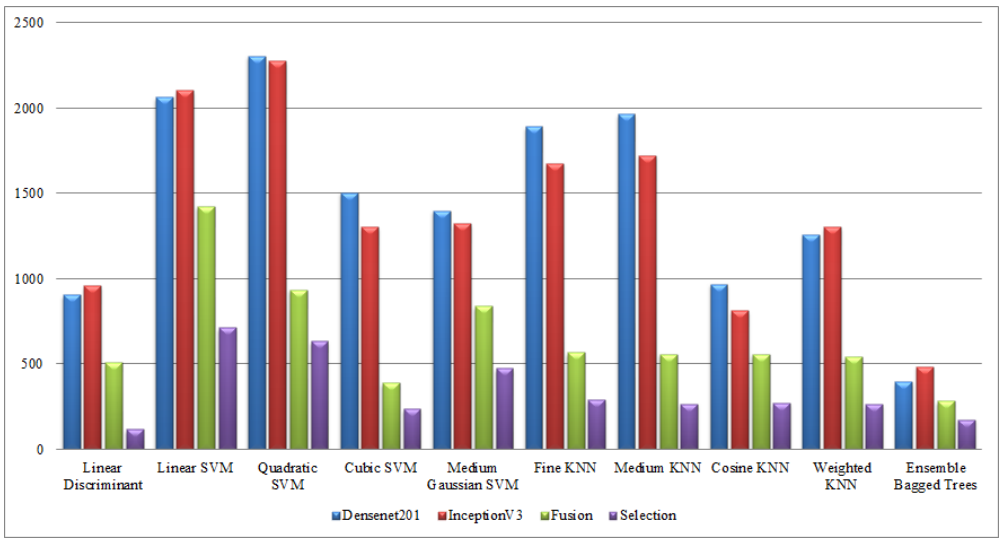

Figure 19. Computational time-based comparison of middle steps on IXMAS dataset.

4.2. Comparison with SOTA

Overall, the feature selection process maintains the classification accuracy while significantly reducing the computational time. A comparison with some recent techniques was also conducted as provided in Table 11. This table shows that the proposed design results are significantly improved. The main strength of the proposed design is the fusion of deep features using the SbE approach and best feature selection using KcWKNN.

Table 11. Comparison of the proposed design with existing techniques in terms of accuracy. The bold represents the best obtained values.

Reference	Dataset	Accuracy (%)
Muhammad et al. [45], 2020	KTH	98.30
Proposed method	**KTH**	**99.00**
Muhammad et al. [4], 2020	IXMAS	95.20
Amir et al. [55], 2021	IXMAS	87.48
Proposed method	**IXMAS**	**97.10**
Muhammad et al. [56], 2020	WVU	99.10
Muhammad et al. [57], 2019	WVU	99.90
Proposed method	**WVU**	**99.40**
Evan et al. [58], 2008	Hollywood	91.80
Proposed method	**Hollywood**	**99.20**

5. Conclusions

HAR has gained a lot of popularity in recent years. Multiple techniques have been used for the accurate recognition of human actions. The problem is to correctly identify the action in real-time and from multiple perspectives. In this work, a design is proposed where the key aim is to improve the accuracy of the HAR process in the complex video sequences using advanced deep learning techniques. The proposed design consists of four steps, namely feature mapping, feature fusion, feature selection, and classification. Two modified deep learning models, DenseNet201 and InceptionV3 were used for feature mapping. Fusion and selection were performed using the serial-based extended approach and Kurtosis-controlled Weighted KNN approach, respectively. The results were obtained after extensive experimentation on state-of-the-art action datasets. Based on the results, it is concluded that the proposed design performed better than the existing techniques in terms of accuracy as well as computational time. Cubic SVM and Fine KNN classifiers were top performers on the proposed HAR method. The key limitation of this work is the computational time that was noted during the original deep extracted features. This step increases the computational time that is not suitable for the real-time applications. As a future study, we intend to test the proposed design on relatively complex action datasets such as HMDB51 and UCF101. Moreover, the recent deep learning models can also be considered for feature extraction and will study the less complexity feature fusion and selection algorithms.

Author Contributions: Conceptualization, S.K., M.A.K. and A.A.; methodology, S.K., M.A.K. and M.A.; software, S.K. and M.A.K.; validation, M.A., U.T. and H.-S.Y.; formal analysis, U.T. and H.-S.Y.; investigation, U.T. and M.A.; resources, M.A.K. and U.T.; data curation, H.-S.Y. and A.A.; writing—original draft preparation, S.K. and M.A.K.; writing—review and editing, M.A., U.T. and F.A.; visualization, A.A. and F.A.; supervision, M.A.K. and H.-S.Y.; project administration, F.A. and A.A.; funding acquisition, H.-S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partially supported by Ewha Womans University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, D.; Lee, I.; Kim, D.; Lee, S. Action Recognition Using Close-Up of Maximum Activation and ETRI-Activity3D LivingLab Dataset. *Sensors* **2021**, *21*, 6774. [CrossRef]
2. Mishra, O.; Kavimandan, P.S.; Tripathi, M.; Kapoor, R.; Yadav, K. Human Action Recognition Using a New Hybrid Descriptor. In *Advances in VLSI, Communication and Signal Processing*; Springer: Singapore, 2021.
3. Chen, X.; Xu, L.; Cao, M.; Zhang, T.; Shang, Z.; Zhang, L. Design and Implementation of Human-Computer Interaction Systems Based on Transfer Support Vector Machine and EEG Signal for Depression Patients' Emotion Recognition. *J. Med. Imaging Health Inform.* **2021**, *11*, 948–954. [CrossRef]
4. Javed, K.; Khan, S.A.; Saba, T.; Habib, U.; Khan, J.A.; Abbasi, A.A. Human action recognition using fusion of multiview and deep features: An application to video surveillance. *Multimed. Tools. Appl.* **2020**, 1–27. [CrossRef]
5. Liu, D.; Xu, H.; Wang, J.; Lu, Y.; Kong, J.; Qi, M. Adaptive Attention Memory Graph Convolutional Networks for Skeleton-Based Action Recognition. *Sensors* **2021**, *21*, 6761. [CrossRef] [PubMed]
6. Ahmed, M.; Ramzan, M.; Khan, H.U.; Iqbal, S.; Choi, J.-I.; Nam, Y.; Kady, S. Real-Time Violent Action Recognition Using Key Frames Extraction and Deep Learning. *Comput. Mater. Continua* **2021**, *69*, 2217–2230. [CrossRef]
7. Wang, J.; Cao, D.; Wang, J.; Liu, C. Action Recognition of Lower Limbs Based on Surface Electromyography Weighted Feature Method. *Sensors* **2021**, *21*, 6147. [CrossRef]
8. Zin, T.T.; Htet, Y.; Akagi, Y.; Tamura, H.; Kondo, K.; Araki, S.; Chosa, E. Real-Time Action Recognition System for Elderly People Using Stereo Depth Camera. *Sensors* **2021**, *21*, 5895. [CrossRef] [PubMed]
9. Farnoosh, A.; Wang, Z.; Zhu, S.; Ostadabbas, S. A Bayesian Dynamical Approach for Human Action Recognition. *Sensors* **2021**, *21*, 5613. [CrossRef] [PubMed]
10. Buehner, M.J. Awareness of voluntary and involuntary causal actions and their outcomes. *Psychol. Conscious. Theory Res. Pract.* **2015**, *2*, 237. [CrossRef]
11. Hassaballah, M.; Hosny, K.M. Studies in Computational Intelligence. In *Recent Advances In Computer Vision*; Hassaballah, M., Hosny, K.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2019.
12. Sharif, M.; Akram, T.; Raza, M.; Saba, T.; Rehman, A. Hand-crafted and deep convolutional neural network features fusion and selection strategy: An application to intelligent human action recognition. *Appl. Soft Comput.* **2020**, *87*, 105986.
13. Kolekar, M.H.; Dash, D.P. Hidden markov model based human activity recognition using shape and optical flow based features. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), Singapore, 22–25 November 2016.
14. Hermansky, H. TRAP-TANDEM: Data-driven extraction of temporal features from speech. In Proceedings of the 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721), St Thomas, VI, USA, 30 November–4 December 2003.
15. Krzeszowski, T.; Przednowek, K.; Wiktorowicz, K.; Iskra, J. The Application of Multiview Human Body Tracking on the Example of Hurdle Clearance. In *Sport Science Research and Technology Support*; Cabri, J., Pezarat-Correia, P., Vilas-Boas, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2016.
16. Hassaballah, M.; Awad, A.I. *Deep Learning In Computer Vision: Principles and Applications*; CRC Press: Boca Raton, FL, USA, 2020.
17. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**. [CrossRef] [PubMed]
18. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
19. Palacio-Niño, J.-O.; Berzal, F. Evaluation metrics for unsupervised learning algorithms. *arXiv* **2019**, arXiv:1905.05667.
20. Kiran, S.; Khan, M.A.; Javed, M.Y.; Alhaisoni, M.; Tariq, U.; Nam, Y.; Damaševičius, R.; Sharif, M. Multi-Layered Deep Learning Features Fusion for Human Action Recognition. *Comput. Mater. Cont.* **2021**, *69*, 4061–4075. [CrossRef]
21. Khan, M.A.; Alhaisoni, M.; Armghan, A.; Alenezi, F.; Tariq, U.; Nam, Y.; Akram, T. Video Analytics Framework for Human Action Recognition. *Comput. Mater. Cont.* **2021**, *68*, 3841–3859.
22. Sharif, M.; Akram, T.; Yasmin, M.; Nayak, R.S. Stomach deformities recognition using rank-based deep features selection. *J. Med. Econ.* **2019**, *43*, 329.
23. Saleem, F.; Khan, M.A.; Alhaisoni, M.; Tariq, U.; Armghan, A.; Alenezi, F.; Choi, J.; Kadry, S. Human Gait Recognition: A Single Stream Optimal Deep Learning Features Fusion. *Sensors* **2021**, *21*, 7584. [CrossRef] [PubMed]
24. Khan, A.; Javed, M.Y.; Alhaisoni, M.; Tariq, U.; Kadry, S.; Choi, J.; Nam, Y. Human Gait Recognition Using Deep Learning and Improved Ant Colony Optimization. *Comput. Mater. Cont.* **2022**, *70*, 2113–2130. [CrossRef]
25. Mehmood, A.; Tariq, U.; Jeong, C.-W.; Nam, Y.; Mostafa, R.R.; Elaeiny, A. Human Gait Recognition: A Deep Learning and Best Feature Selection Framework. *Comput. Mater. Cont.* **2022**, *70*, 343–360. [CrossRef]
26. Wang, H.; Yu, B.; Xia, K.; Li, J.; Zuo, X. Skeleton Edge Motion Networks for Human Action Recognition. *Neurocomputing* **2021**, *423*, 1–12. [CrossRef]
27. Bi, Z.; Huang, W. Human action identification by a quality-guided fusion of multi-model feature. *Future Gener. Comput. Syst.* **2021**, *116*, 13–21. [CrossRef]
28. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process* **2020**, *138*, 106587. [CrossRef]

29. Manivannan, A.; Chin, W.C.B.; Barrat, A.; Bouffanais, R. On the challenges and potential of using barometric sensors to track human activity. *Sensors* **2020**, *20*, 6786. [CrossRef] [PubMed]
30. Ahmed Bhuiyan, R.; Ahmed, N.; Amiruzzaman, M.; Islam, M.R. A robust feature extraction model for human activity characterization using 3-axis accelerometer and gyroscope data. *Sensors* **2020**, *20*, 6990. [CrossRef]
31. Zhao, B.; Li, S.; Gao, Y.; Li, C.; Li, W. A Framework of Combining Short-Term Spatial/Frequency Feature Extraction and Long-Term IndRNN for Activity Recognition. *Sensors* **2020**, *20*, 6984. [CrossRef] [PubMed]
32. Muhammad, K.; Ullah, A.; Imran, A.S.; Sajjad, M.; Kiran, M.S.; Sannino, G.; Albuquerque, V.H.C. Human action recognition using attention based LSTM network with dilated CNN features. *Future Gener. Comput. Syst.* **2021**, *125*, 820–830. [CrossRef]
33. Li, C.; Xie, C.; Zhang, B.; Han, J.; Zhen, X.; Chen, J. Memory attention networks for skeleton-based action recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [CrossRef] [PubMed]
34. Im, W.; Kim, T.-K.; Yoon, S.-E. Unsupervised Learning of Optical Flow with Deep Feature Similarity. In *Computer Vision—ECCV 2020. ECCV 2020*; Lecture Notes in Computer Science; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; Volume 12369.
35. Liu, W.; Zha, Z.-J.; Wang, Y.; Lu, K.; Tao, D. p-Laplacian regularized sparse coding for human activity recognition. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5120–5129. [CrossRef]
36. Jalal, A.; Kamal, S.; Kim, D. A Depth Video-based Human Detection and Activity Recognition using Multi-features and Embedded Hidden Markov Models for Health Care Monitoring Systems. *Int. J. Interact. Multimed. Artif. Intell.* **2017**, *4*, 54. [CrossRef]
37. Effrosynidis, D.; Arampatzis, A. An evaluation of feature selection methods for environmental data. *Ecol Inform.* **2021**, *61*, 101224. [CrossRef]
38. Melhart, D.; Liapis, A.; Yannakakis, G.N. The Affect Game AnnotatIoN (AGAIN) Dataset. *arXiv* **2021**, arXiv:2104.02643.
39. Hassan, M.M.; Uddin, M.Z.; Mohamed, A.; Almogren, A. A robust human activity recognition system using smartphone sensors and deep learning. *Future Gener. Comput. Syst.* **2018**, *81*, 307–313. [CrossRef]
40. Joshi, A.B.; Kumar, D.; Gaffar, A.; Mishra, D. Triple color image encryption based on 2D multiple parameter fractional discrete Fourier transform and 3D Arnold transform. *Opt. Lasers. Eng.* **2020**, *133*, 106139. [CrossRef]
41. Cervantes, J.; Garcia-Lamont, F.; Rodríguez-Mazahua, L.; Lopez, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* **2020**, *408*, 189–215. [CrossRef]
42. Wang, L.; Xu, Y.; Cheng, J.; Xia, H.; Yin, J.; Wu, J. Human action recognition by learning spatio-temporal features with deep neural networks. *IEEE Access* **2018**, *6*, 17913–17922. [CrossRef]
43. Gumaei, A.; Hassan, M.M.; Alelaiwi, A.; Alsalman, H. A hybrid deep learning model for human activity recognition using multimodal body sensing data. *IEEE Access* **2019**, *7*, 99152–99160. [CrossRef]
44. Gao, Z.; Xuan, H.-Z.; Zhang, H.; Wan, S.; Choo, K.-K.R. Adaptive fusion and category-level dictionary learning model for multiview human action recognition. *IEEE Internet Things J.* **2019**, *6*, 9280–9293. [CrossRef]
45. Khan, M.A.; Zhang, Y.-D.; Khan, S.A.; Attique, M.; Rehman, A.; Seo, S. A resource conscious human action recognition framework using 26-layered deep convolutional neural network. *Multimed. Tools. Appl.* **2020**. [CrossRef]
46. Xia, K.; Huang, J.; Wang, H. LSTM-CNN architecture for human activity recognition. *IEEE Access* **2020**, *8*, 56855–56866. [CrossRef]
47. Rashid, M.; Sharif, M.; Raza, M.; Sarfraz, M.M.; Afza, F. Object detection and classification: A joint selection and fusion strategy of deep convolutional neural network and SIFT point features. *Multimed. Tools. Appl.* **2019**, *78*, 15751–15777. [CrossRef]
48. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE Press: Piscataway, NJ, USA.
49. Hussain, N.; Sharif, M.; Khan, S.A.; Albesher, A.A.; Saba, T.; Armaghan, A. A deep neural network and classical features based scheme for objects recognition: An application for machine inspection. *Multimed. Tools. Appl.* **2020**, 1–23. [CrossRef]
50. Akram, T.; Zhang, Y.-D.; Sharif, M. Attributes based skin lesion detection and recognition: A mask RCNN and transfer learning-based deep learning framework. *Pattern Recognit. Lett.* **2021**, *143*, 58–66.
51. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, USA, 23–28 June 2014.
52. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image databas e. In Proceedings of the 2009 IEEE conference on computer vision and pattern recognition, Miami, FL, USA, 20–25 June 2009.
53. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *NIPS* **2012**, *25*, 1097–1105. [CrossRef]
54. Naheed, N.; Shaheen, M.; Khan, S.A.; Alawairdhi, M.; Khan, M.A. Importance of features selection, attributes selection, challenges and future directions for medical imaging data: A review. *Comput. Sci. Eng.* **2020**, *125*, 314–344. [CrossRef]
55. Nadeem, A.; Jalal, A.; Kim, K. Automatic human posture estimation for sport activity recognition with robust body parts detection and entropy markov model. *Multimed. Tools. Appl.* **2021**, *22*, 1–34. [CrossRef]
56. Sharif, M.; Zahid, F.; Shah, J.H.; Akram, T. Human action recognition: A framework of statistical weighted segmentation and rank correlation-based selection. *Pattern Anal. Appl.* **2020**, *23*, 281–294. [CrossRef]

57. Akram, T.; Sharif, M.; Javed, M.Y.; Muhammad, N.; Yasmin, M. An implementation of optimized framework for action classification using multilayers neural network on selected fused features. *Pattern Anal. Appl.* **2019**, *22*, 1377–1397.
58. Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.