*Computers, Materials & Continua*
DOI:10.32604/cmc.2021.017800
***Article***

Tech Science Press

# Multi-Layered Deep Learning Features Fusion for Human Action Recognition

**Sadia Kiran[1], Muhammad Attique Khan[1], Muhammad Younus Javed[1], Majed Alhaisoni[2], Usman Tariq[3], Yunyoung Nam[4,*], Robertas Damaševičius[5] and Muhammad Sharif[6]**

[1]Department of Computer Science, HITEC University Taxila, Taxila, Pakistan
[2]College of Computer Science and Engineering, University of Ha'il, Ha'il, Saudi Arabia
[3]College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Khraj, Saudi Arabia
[4]Department of Computer Science and Engineering, Soonchunhyang University, Asan, Korea
[5]Faculty of Applied Mathematics, Silesian University of Technology, Gliwice, Poland
[6]Department of Computer Science, COMSATS University Islamabad, Wah Campus, Pakistan
[*]Corresponding Author: Yunyoung Nam. Email: ynam@sch.ac.kr
Received: 11 February 2021; Accepted: 29 March 2021

**Abstract:** Human Action Recognition (HAR) is an active research topic in machine learning for the last few decades. Visual surveillance, robotics, and pedestrian detection are the main applications for action recognition. Computer vision researchers have introduced many HAR techniques, but they still face challenges such as redundant features and the cost of computing. In this article, we proposed a new method for the use of deep learning for HAR. In the proposed method, video frames are initially pre-processed using a global contrast approach and later used to train a deep learning model using domain transfer learning. The Resnet-50 Pre-Trained Model is used as a deep learning model in this work. Features are extracted from two layers: Global Average Pool (GAP) and Fully Connected (FC). The features of both layers are fused by the Canonical Correlation Analysis (CCA). Then features are selected using the Shanon Entropy-based threshold function. The selected features are finally passed to multiple classifiers for final classification. Experiments are conducted on five publicly available datasets as IXMAS, UCF Sports, YouTube, UT-Interaction, and KTH. The accuracy of these data sets was 89.6%, 99.7%, 100%, 96.7% and 96.6%, respectively. Comparison with existing techniques has shown that the proposed method provides improved accuracy for HAR. Also, the proposed method is computationally fast based on the time of execution.

**Keywords:** Action recognition; transfer learning; features fusion; features selection; classification

## 1 Introduction

Human action Recognition (HAR) has incredible significance in numerous everyday applications, for instance, video surveillance [1], virtual reality, robotics, video analytics, and assistive living, etc. [2,3]. The movement of several body parts of a human being simultaneously can be referred to as an action [4,5]. According to the view of computer vision (CV), action recognition relates the observations such as video data with sequences [6]. A sequence of human actions accomplished by at least two actors in which one actor must be a person or an object is called interaction [7]. It has become a demanding task in CV to understand the human activities from videos. Automated recognition of an activity that being performed by human in a video sequences is the main capability of intelligent video system [8].

The main aim of action recognition is to supply useful information related to the subjects' habits. Also, they permit the system or robot to make users comfortable with interacting with them. Recognition and forecasting the occurrence of crimes could be done by interpreting human activities to assist the police or other agencies in reacting straightaway [9]. The proper recognition of human actions accurately is extremely difficult due to lots of problems, e.g., jumbled backgrounds, changing environmental conditions, and viewpoint differences [10].

HAR techniques from video sequences are usually classified into two types- template-based method and model-based method. In template-based method, lower and middle-level features are emphasized. In the model-based method, high-level features are emphasized [11,12]. In the past few years, a large number of feature extraction methods are introduced, especially spatial-temporal interest points (STIP) feature descriptor [13], motion extraction image (MEI) and motion history image (MHI) [14], spatiotemporal descriptors based on 3-D gradients [15], extend robust scale features (SURF) [16], 3D SIFT [17], histogram oriented gradients (HOG) 3D [15] and dense optical trajectories [18] have achieved fruitful results for HAR using video sequences [19]. Then classification of these extracted features is done using different machine learning (ML) classification methods such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), decision tree, linear discriminant analysis (LDA), Ensemble tree and neural networks, etc.

Compared to the techniques above, significant performance was achieved after the deep convolutional neural network (DCNN) in machine learning [20,21]. Several pre-trained deep models are presented in the literature, such as AlexNet, VGG, GoogleNet, ResNet, and Inception V3. DCNN models can act directly on the raw inputs without any preprocessing [22]. More complex features can be extracted with every supplementary layer. A major dissimilarity in the complexity of adjoining layers of model reduces with the proceeding of the data to the upper convolutional layers. In recent years, these deep learning (DL) models are utilized for HAR and show high accuracy [23]. But sometimes, when humans have performed complex actions similar to each other, these models diminish the performance.

Therefore, some researchers presented sequential techniques. In those techniques, they performed fusion and get better information for an entire video sequence. Afza et al. [24] presented a parallel fusion approach named length control features (LCF) and achieved improved performance. Xiaog et al. [25] presented a two-stream CNN model for action recognition. They focused on both optical flow-based generated images and temporal actions for better action recognition. Elharrous et al. [26] presented a combined model for both action classification and summarization. They initially extract the human silhouette and then extract the shape and temporal features. In summary these methods computed improved results but they did not focused on the computational time. The major challenges which are handling in this work are: i) Change of human variation and viewpoint due to camera conditions, camera being static or dynamic; ii) Consumption of more time during the training process, and iii) Selection of the most related features is a key problem for minimizing error rate of an automated system. In this article, we proposed a fusion based framework along with a reduction scheme for better computational time and improved accuracy. Our major contributions are as follows:

(a) Global frames preprocessing is employed using mean and standard deviation values in a fitness function.
(b) Transfer learning-based features are extracted from two successive layers of Resnet50, where the numbers of parameters are the same as the original model.
(c) Canonical correlation approach is implemented for deep features fusion of both successive layers.
(d) A threshold function is implemented based on Shannon Entropy for the selection of best features.
(e) Multiple supervised learning algorithms are implemented for classification and also conducted a fair comparison with existing techniques.

The rest of the manuscript is organized as follows. The proposed methodology includes frame normalization, deep learning features, features fusion, and selection, presented in Section 2. Results are presented in Section 3, which followed the Conclusion of Section 4.

## 2 Proposed Methodology

A unified model has been proposed in this article for HAR built on the fusion of multiple deep CNN layers features. Five core steps are involved in this work-database normalization, individual CNN features extraction through two successive layers, fusion information of both successive layers (AVG-Pool and FC1000), selection of best features, and finally classification through supervised learning method. The proposed method is evaluated on five datasets and also compared with existing techniques. Proposed architecture of this work has been illustrated in Fig. 1.

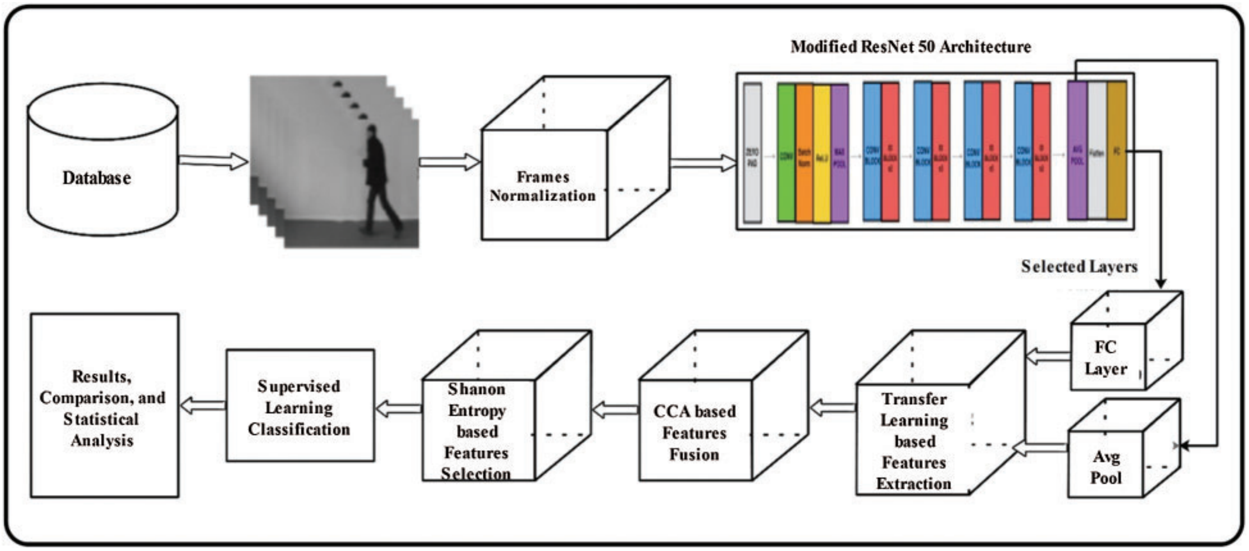
Modified ResNet 50 Architecture
Database
Frames Normalization
ZERO PAD
CONV
Batch Norm
ReLU
MAX POOL
CONV BLOCK
ID BLOCK x3
CONV BLOCK
ID BLOCK x3
CONV BLOCK
ID BLOCK x3
CONV BLOCK
ID BLOCK x3
AVG POOL
Flatten
FC
Selected Layers
FC Layer
Avg Pool
Transfer Learning based Features Extraction
CCA based Features Fusion
Shanon Entropy based Features Selection
Supervised Learning Classification
Results, Comparison, and Statistical Analysis

**Figure 1:** Proposed deep learning-based architecture for human action recognition

### 2.1 *Preprocessing*

The main objective of preprocessing is to bring improvement in image statistics. It represses unwanted distortions or improves some image features that are essential for additional processing.

In this work, we performed normalization of video frames. Normalization is a procedure frequently applied as a major aspect of data preparation for machine learning. The normalization process is comprised of global contrast normalization(GCN), local normalization, and histogram equalization [27].

**GCN:** In Global Contrast Normalization (GCN), each value of an image's pixels is subtracted from the mean value and then divided with predictable error. To avert images from possessing differing quantities of contrast is the main objective of GCN. Images having very little, however, not equal to zero contrast have smaller details and turn out to be problematic for action recognition. GCN can be described as:

$$Z'_{m,n,l} = s\frac{Z_{m,n,l} - \overline{Z}}{max\left\{\varepsilon, \sqrt{\lambda + \frac{1}{3\mathrm{rc}}\sum_{m=1}^{\mathrm{r}}\sum_{n=1}^{\mathrm{c}}\sum_{l=1}^{3}(Z_{m,n,l} - \overline{Z})^2}\right\}} \tag{1}$$

where $m$ represents a row, $n$ represents column, $l$ is a depth of color an $d$ mean intensity of the full image is represented by $\overline{Z}$. Then the local contrast is improved by employing bottom hat filtering and log transformation. In the end, the output of both local and global is combined in one matrix for the final resultant enhanced image. These resultant enhanced images are used for the training of a deep learning model for further processing.

### *2.2 Convolutional Neural Network*

A simple CNN model consists of the following layers: Convolution Layer, Pooling Layer, ReLu Layer, Batch Normalization Layer, Fully Connected Layer, and output layer. The details of each layer defined as follows:

The Convolutional layer received a volume of size $M_1 \times G_1 \times D_1$. This layer needs *four* variables/factors, e.g., $C$ Filters, their spatial range $E$, the stride $S$ and the quantity of zero padding $P$. It generates a capacity of size $M_2 \times G_2 \times D_2$ where

$$M_2 = (M_1 - E + 2P)/S + 1 \tag{2}$$

$$G_2 = (G_1 - E + 2P)/S + 1 \tag{3}$$

$$D_2 = C \tag{4}$$

It represents $E.E.D_1$ weights per filter with parameter sharing. The next layer is the pooling layer. The pooling layer's task is to lessen the image's spatial dimensions to minimize the number of variables and calculations within the linkage. It thus controls the problem of overfitting among layers. The pooling layer takes a capacity that has size $M_1 \times G_1 \times D_1$. This needs 2 variables, e.g., stride $S$, and their spatial range $E$. It generates a capacity that has $M_2 \times G_2 \times D_2$. Mathematically, it is defined as:

$$M_2 = (M_1 - E)/S + 1 \tag{5}$$

$$G_2 = (G_1 - E)/S + 1 \tag{6}$$

$$D_2 = D_1 \tag{7}$$

The next layer is a ReLU activation layer. It is a kind of activation function. Mathematically, it is described as $z = \max(0, y)$. This function described that the values with negative would be converted into zero (positive). A fully connected layer can be described as the calculation of some component of the output $y^{l+1}$(or $z$ note that $z$ is an *alias* for $y^{l+1}$) have to need of every component of the input $y^l$. This layer is also known as the feature layer. The last layer is softmax, which is used for classification. In this layer, an entropymax function is employed for the final decision.

### *2.3 Deep Learning Features*

Deep learning showed great success in machine learning in the last few years for several video surveillance and medical imaging applications. Video surveillance is a hot research area, but the researchers face major issues called imbalanced datasets and large datasets. In this article, we used a pre-trained deep learning model named ResNet-50. The common ResNet models usually perform the skipping of double or triple layers, which comprise (ReLu) and batch normalization [28]. The incentive for skipping over layers is to attach the output by the previous layer to the next coming layer. This will support reducing the vanishing gradient issue. Skipping helps simplify the network because it uses small no layers in the initial training stages. As a result, it accelerates the process of learning. As long as the network will learn the space of features, it slowly reinstates those skipped layers. A neural network in the absence of residual portion investigates additional space of feature that makes it unprotected. ResNet-50 is a convolutional neural network (CNN), originally trained on the ImageNet dataset, consisting of around 100 million images of 1000 classes. The depth of this network is 50 layers, and the input size is 224-by-224-by-3. The original number of layers and selected layers are presented in Tabs. 1 and 2.

**Table 1:** Number of layers of ResNet50 CNN model

| Image input layer | 1 |
|---|---|
| Convolution 2D layer | 53 |
| Batch normalization layer | 53 |
| ReLU layer | 49 |
| Max pooling 2D layer | 1 |
| Addition layer | 16 |
| Average pooling 2D layer | 1 |
| Fully connected layer | 1 |
| Softmax layer | 1 |
| Classification output layer | 1 |

We modified this ResNet50 architecture according to the number of classes. For this purpose, we removed the last layer and added a new layer that includes the number of action classes. We then train the modified model through transfer learning on 70% data, where the next 30% are used for the testing process. In the training process, we define the number of epochs 100, the learning rate is 0.0001, and mini-batch size is 64. The input size is the same as the input of the original deep model. We extract features from the last two feature layers named Average Pool Layer and Fully Connected Layer. The dimension of extracted features is $N \times 2048$ and $N \times 1000$, respectively. In the later stage, we fused features of both vectors into one vector for further processing.

### *2.4 Features Fusion Using CCA*

In multivariate statistical analysis, CCA has similar significance as principal component analysis (PCA) and linear discriminant analysis (LDA). It is the most important multi-data processing technique. CCA is conventionally utilized for analyzing the associations amongst two groups of variables [29]. It tries to find two groups of random variables so that these random variables presume the highest correlation over two groups of data. In contrast, the transformations inside every group of data are not correlated. Mathematically, it is formulated as follows:

**Table 2:** Description of selected layers of ResNet50 for feature extraction

| Layer | Description |
|---|---|
| Input layer | Name: 'input_1'; Input size: [224,224,3]<br>DataAugmentation: 'none'; Normalization: 'zerocenter' |
| Convolution | Name: 'conv1'; FilterSize: [7,7]; NumChannels: 3<br>NumFilters: 64; Stride: [2,2]; PaddingMode: 'manual'<br>PaddingSize: [3,3,3,3]; Weights: 4-D single<br>Bias: 1 x 1 x 64 single; WeightLearnRateFactor:1<br>WeightL2Factor:1; BiasLearnRateFactor:1<br>BiasL2Factor:0 |
| Average Pool layer | Name: 'avg_pool'; Poolsize: [7,7]; Stride: [7,7]<br>PaddingMode: 'manual'; PaddingSize: [0,0,0,0] |
| Fully Connected Layer | Name: 'fc1000'; InputSize: 2048; OutputSize: 1000<br>Weights: 1000 x 2048 single; Bias: 1000 x 1 single<br>WeightLearnRateFactor: 1; WeightL2Factor: 1<br>BiasLearnRateFactor: 1; BiasL2Factor: 0 |

Two groups of data $Z_1 \in \mathbb{R}^{m*p}$ *and* $Z_2 \in \mathbb{R}^{m*q}$ are given, CCA finds the linear combinations $Z_1V_1$ *and* $Z_2V_2$ which will enhance the couple-wise correlations over the two groups of data. $E_1$ *and* $E_2 \in \mathbb{R}^{m*b}, b \leq \min(rank(Z_1, Z_2))$, are identified as canonical variables and $V_1 \in \mathbb{R}^{p*b}$ *and* $V_2 \in \mathbb{R}^{q*b}$ are the canonical vectors. If another method is used, the procedure discovers an initial couple of canonical vectors $v_1^{(1)}$ *and* $v_2^{(1)}, \left(v_1^{(1)} \in \mathbb{R}^{p*1}, v_2^{(1)} \in \mathbb{R}^{q*1}\right)$, which will enhance the linear fusion of the two groups of data.

$$\max_{v_1^{(1),}v_2^{(1)}} corr(Z_1v_1^{(1)}, Z_2v_2^{(1)}) \tag{8}$$

To acquire the initial couple of canonical variables stated as:

$$e_1^{(1)} = Z_1v_1^{(1)} \text{ } and \text{ } e_2^{(1)} = Z_2v_2^{(1)} \tag{9}$$

The leftover $b-1$ canonical variables will be computed by using the same method. By applying further restrictions at matrices columns, i.e., $e_h^{(i)}\,(i = 1, \ldots, b,\ h = 1, 2)$, every group of data, the canonical variables are uncorrelated, and they possess zero mean and unit variance.

$$E_h^T E_h = I, \quad h = 1, 2. \tag{10}$$

$$E_h^T E_y = U_{h,y}, \quad h \neq y, \quad h, y = 1, 2, \tag{11}$$

$$U_{h,y} = diag(u_{h,y}^{(1)}, \ldots, u_{h,y}^{(b)}). \tag{12}$$

The issue of CCA may be presented as an optimization problem that uses "Lagrange multipliers," and the canonical covariates can be computed by resolving a general eigenvalue explanation [29]. Here the columns of $V_1$ and $V_2$ are the eigenvectors of the two matrices $S_{z_1}^{-1}S_{z_1,z_2}S_{z_2}^{-1}S_{z_2,z_1}$ and $S_{z_2}^{-1}S_{z_2,z_1}S_{z_1}^{-1}S_{z_1,z_2}$, where $S_{z_1,z_2}$ is the cross-correlation matrix of $Z_1$ *and* $Z_2$ ($S_{z_2,z_1} = S_{z1,z_2}^{T}$), and $S_{z_1}$ and $S_{z_2}$ are the autocorrelation matrices of $Z_1$ and $Z_2$, respectively. Hence, the final fusion is defined as:

$$Fused = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}_{N\times K} \tag{13}$$

where, $K$ represents the number of fused features, which are 1876 in this work, and $N$ represents the sample frames used for training and the testing.

### 2.5 Feature Selection

Feature selection is the process of selecting the best features for accurate classification within less execution time. In this work, a new threshold function is proposed to select best features based on the Shannon Entropy. Initially, we computed the entropy value of the fused vector by the following equation.

$$H = -\sum_{i=1}^{N} p_i \log(p_i) \tag{14}$$

where, $N$ represents the total number of features, $p_i$ is the probability of each feature, and $i$ represents the index of each feature in the fused vector. Then, we implemented a threshold function to select the best features. The criterion of the selection of best features is the fitness function, which is Fine KNN. We initialize 20 iterations, and after each iteration, the selected features are evaluated through the fitness function. In the end, the higher accuracy-based feature set is selected for the final classification. Mathematically, the threshold function is defined as follows:

$$Th = \begin{cases} Sel\,(i) & for\;\; Fused\,(i) \geq H \\ Ignore, & Elsewhere \end{cases} \tag{15}$$

The final selected features are passed in the supervised learning classifiers for final classification.

## 3 Experimental Results and Discussion

The proposed method is evaluated on four selected datasets as IXMAS, UCF Sports, UT Interaction, and KTH. Each classifier's performance is measured through the following parameters such as recall rate, precision rate, accuracy, FNR, and testing time. Also, the performance of Fine KNN is further compared with few other well-known classifiers such as Linear Discriminant (LDA), Quadratic SVM (QSVM), Cubic SVM (CSVM), Medium Gaussian SVM (MGSVM), and Weighted KNN (WKNN). The results of each classifier are presented below.

### *3.1 IXMAS Dataset*

The proposed recognition accuracy of the IXMAS dataset is presented in Tab. 3. Six different classifiers are utilized for recognition accuracy and selected the best one based on the accuracy performance. From this table, the highest accuracy is 89.6% achieved on Fine KNN, whereas the other parameter such as recall rate is 89.58%, the precision rate is 89.75%, FNR is 10.42%, and the classification computation time is 277.97 s. The next highest accuracy is 87.8% which is attained at Cubic SVM. The minimum accuracy of 79.8% is achieved on Weighted KNN along best recognition time of 194.89 s. The best accuracy of FKNN is proved by the confusion matrix given in Tab. 4. Besides, the computation time of each classifier is plotted in Fig. 2. As shown in this figure, the WKNN executes fast as compare to other classifiers.

**Table 3:** Proposed recognition accuracy of IXMAS dataset

| Classifier | Recall rate (%) | Precision rate (%) | Accuracy (%) | FNR (%) | Time (s) |
|---|---|---|---|---|---|
| LDA | 83.66 | 83.83 | 83.7 | 16.34 | 557.59 |
| QSVM | 86.00 | 86.25 | 86.1 | 14.00 | 1478.8 |
| CSVM | 87.83 | 88.00 | 87.8 | 12.17 | 1676.4 |
| MG SVM | 83.75 | 84.25 | 83.8 | 16.25 | 2060.6 |
| **Fine KNN** | **89.58** | **89.75** | **89.6** | **10.42** | 277.97 |
| W KNN | 79.75 | 80.58 | 79.8 | 20.25 | **194.89** |

**Table 4:** Confusion matrix of FKNN for IXMAS dataset. The action classes are checkwatch (CW), CrossArm (CA), ScratchHead (SH), TurnAround (TA), Wave (WV), getup (GU), kick (K), pickup (PU), point (P), punch (PN), SitDown (SD), and walk (W)

| Class | Recognition class | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CW | CA | SH | TA | WV | GU | K | PU | P | PN | SD | W |
| CW | **97%** | 1% | 1% | | | <1% | | | | | <1% | |
| CA | 5% | **93%** | 2% | | <1% | | | | | | | |
| SH | 3% | 4% | **92%** | | | 1% | | | | | <1% | |
| TA | 1% | 1% | 1% | **85%** | | 3% | | | | 1% | <1% | 6% |
| WV | <1% | <1% | 1% | <1% | **92%** | | 1% | <1% | <1% | 4% | <1% | <1% |
| GU | 1% | <1% | <1% | 3% | | **84%** | <1% | 2% | | 1% | 8% | <1% |
| K | <1% | | <1% | 1% | 1% | | **89%** | <1% | 2% | 6% | | <1% |
| PU | <1% | | | <1% | | 1% | 1% | **93%** | 1% | <1% | 2% | <1% |
| P | 1% | 1% | 1% | 1% | 1% | <1% | 3% | 1% | **89%** | 1% | <1% | |
| PN | 1% | <1% | <1% | <1% | 3% | | 5% | 1% | 3% | **86%** | <1% | <1% |
| SD | 3% | <1% | 2% | 1% | | 4% | | 2% | | <1% | **87%** | |
| W | 1% | | <1% | 12% | <1% | <1% | 1% | | | <1% | | **85%** |

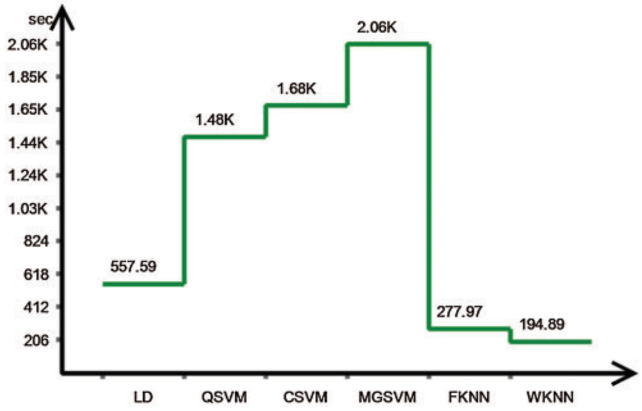

**Figure 2:** Testing Computational time for IXMAS dataset

### *3.2 UCF Sports Dataset*

The proposed recognition accuracy of the UCF Sports dataset is presented in Tab. 5. Six different classifiers are used for recognition accuracy and selected the best one based on the accuracy performance. From this table, the highest accuracy is 99.7% achieved on Linear Discriminant. In contrast, the other parameter such as recall rate is 99.76%, the precision rate is 99.76%, FNR is 0.24, and the classification computation time is 49.143 s. The next highest accuracy is 99.2% achieved on Quadratic SVM and cubic SVM. The minimum accuracy of 93.5% is achieved on Weighted KNN along best recognition time of 16.524 s. The best accuracy of LDA is further proved by a confusion matrix, presented in Tab. 6. Besides, the computation time of each classifier is plotted in Fig. 3. From this figure, it is illustrated that the WKNN executes fast as compare to other classifiers.

**Table 5:** Proposed recognition accuracy of UCF Sports dataset

| Classifier | Recall rate (%) | Precision rate (%) | Accuracy (%) | FNR (%) | Time (s) |
|---|---|---|---|---|---|
| **LDA** | **99.76** | **99.76** | **99.7** | **0.24** | 49.143 |
| QSVM | 99.23 | 99.23 | 99.2 | 0.77 | 112.9 |
| CSVM | 99.15 | 99.23 | 99.2 | 0.85 | 114.86 |
| MG SVM | 97.07 | 97.76 | 97.0 | 2.93 | 532.4 |
| Fine KNN | 98.30 | 98.30 | 98.2 | 1.70 | 19.365 |
| W KNN | 93.61 | 94.53 | 93.5 | 6.39 | **16.524** |

### *3.3 UT Interaction Dataset*

The proposed recognition accuracy of the UT Interaction dataset is presented in Tab. 7. Six different classifiers are used for recognition accuracy and selected the best one based on the accuracy performance. From this table, the highest accuracy is 96.7% achieved on Fine KNN, whereas the other parameters such as recall rate are 97%, the precision rate is 96.66%, and FNR is 3%. The next highest accuracy is 96.5% that is attained on the LDA classifier. The minimum noted the accuracy of 91.2% achieved on Weighted KNN along best recognition time of 14.604 s. The best accuracy of FKNN is proved by the confusion matrix given in Tab. 8. Also, the computation time of each classifier is plotted in Fig. 4. As shown in this figure, the WKNN computationally fast as compared to the rest of the classifiers.

**Table 6:** Confusion matrix of LDA for UCF Sports dataset. The action classes are Golf-Swing-Back(GSB), Golf-Swing-Front (GSF), Golf-Swing-Side(GSS), Kicking-Front (KF), Kicking-Side(KS), Lifting(LF), Riding-Horse (RH), Run-Side(RS), SkateBoarding-Front (SBF), Swing-Bench (SB ), Swing-SideAngle (SSA), Walk-Front(WF), DivingSide(DS)

| Class | Recognition class | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GSB | GSF | GSS | KF | KS | LF | RH | RS | SBF | SB | SSA | WF | DS |
| GSB | **100%** | | | | | | | | | | | | |
| GSF | | **100%** | | | | | | | | | | | |
| GSS | | | **100%** | | | | | | | | | | |
| KF | | | | **100%** | | | | | | | | | |
| KS | | | | | **100%** | | | | | | | | |
| LF | | | | | | **100%** | | | | | | | |
| RH | | | | | | | **100%** | | | | | | |
| RS | | 1% | | | | | | **99%** | | | | | |
| SBF | | | | | | | | | **100%** | | | | |
| SB | | | | | | | | | | **99%** | | 1% | |
| SSA | | | | | | | | | | | **100%** | | |
| WF | | | | | | | | 1% | | | | **99%** | |
| DS | | | | | | | | | | | | | **100%** |

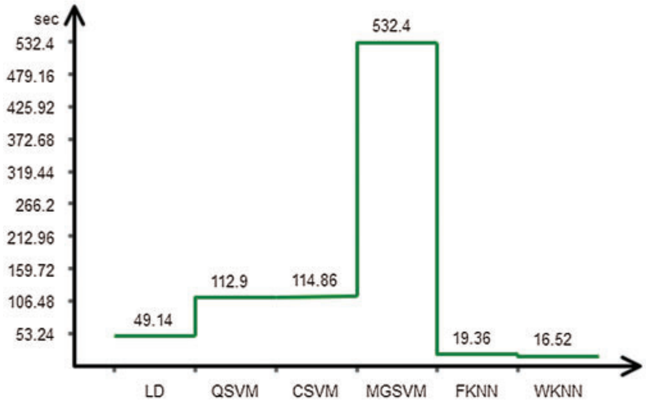

**Figure 3:** Testing computational time for UCF sports dataset

### *3.4 KTH Dataset*

The proposed recognition accuracy of the KTH dataset is shown in Tab. 9. Six different classifiers are used for recognition accuracy and selected the best one based on the accuracy performance. From this table, the highest accuracy is 96.6% achieved on FKNN. In contrast, the other parameter such as recall rate is 96.5%, the precision rate is 96.5%, FNR is 3.5%, and the classification computation time is 497.09 s. The next highest accuracy is 96.0% that is attained on Quadratic SVM. The minimum achieved accuracy is 91.7% for the weighted KNN classifier. The accuracy of Fine KNN is further proved by a confusion matrix, given in Tab. 10. Also, the computation time of each classifier is plotted in Fig. 5. From this figure, it is noted that the WKNN classifier executes fast as compared to the rest of the listed classifiers.

**Table 7:** Proposed recognition accuracy of UT Interaction dataset

| Classifier | Recall rate (%) | Precision rate (%) | Accuracy (%) | FNR (%) | Time (s) |
|---|---|---|---|---|---|
| LDA | 96.66 | 96.83 | 96.5 | 3.34 | 71.872 |
| QSVM | 95.83 | 96.00 | 96.0 | 4.17 | 67.558 |
| CSVM | 96.16 | 96.16 | 96.0 | 3.84 | 54.022 |
| MG SVM | 93.66 | 94.00 | 93.7 | 6.34 | 57.654 |
| **Fine KNN** | **97.00** | **96.66** | **96.7** | **3.00** | 18.523 |
| W KNN | 91.16 | 91.83 | 91.2 | 8.84 | **14.604** |

**Table 8:** Confusion matrix of proposed recognition accuracy for FKNN classifier. The action classes are handshaking (HS), hugging (HG), kicking (K), pointing (PT), punching (PN), and pushing (PU).

| Class | Recognition class | | | | | |
|---|---|---|---|---|---|---|
| | HS | HG | K | PT | PN | PU |
| HS | **99%** | | | 1% | | |
| HG | | **97%** | 1% | | 2% | 1% |
| K | 1% | | **97%** | | 2% | |
| PT | 1% | | 1% | **98%** | | |
| PN | 1% | | 2% | | **96%** | 1% |
| PU | 3% | 1% | 1% | 1% | 1% | **95%** |

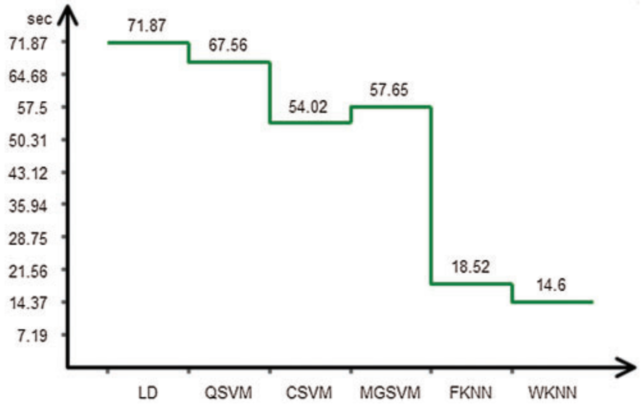

**Figure 4:** Recognition computational time for UT interaction dataset

**Table 9:** Proposed recognition accuracy of KTH dataset

| Classifier | Recall rate (%) | Precision rate (%) | Accuracy (%) | FNR (%) | Time (sec) |
|---|---|---|---|---|---|
| LDA | 95.00 | 95.00 | 95.1 | 5.00 | 541.06 |
| QSVM | 95.83 | 96.00 | 96.0 | 4.17 | 465.75 |
| CSVM | 94.66 | 94.33 | 94.6 | 5.34 | 497.09 |
| MG SVM | 94.50 | 94.66 | 94.5 | 5.50 | 767.32 |
| **Fine KNN** | **96.50** | **96.50** | **96.6** | **3.50** | 291.38 |
| W KNN | 91.66 | 91.83 | 91.7 | 8.34 | 227.27 |

**Table 10:** Confusion matrix of Fine KNN on KTH dataset. The action classes are boxing (BX), Handclapping (HC), handwaving (HW), Jogging (JG), running (R), walking (W)

| Class | Recognition class | | | | | |
|---|---|---|---|---|---|---|
| | BX | HC | HW | JG | R | W |
| BX | **100%** | | | | | |
| HC | | **>99%** | < 1% | | | |
| HW | | 1% | **99%** | | | |
| JG | | | | **93%** | 5% | 3% |
| R | | | | 6% | **92%** | 2% |
| W | | | | 3% | 1% | **96%** |

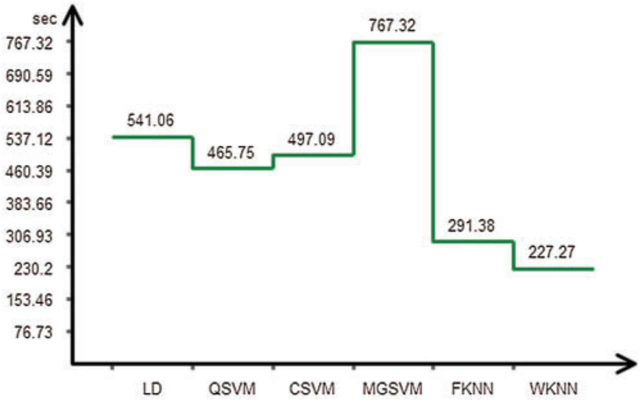

**Figure 5:** Recognition computational time for KTH dataset

Finally, we discussed our proposed method performance in the form of numeric values and graph plots. The numerical results are given in Tabs. 3–10. The results presented in these tables are validated through different performance matrices such as Recall rate, Precision rate, Accuracy, FNR, and Time. Based on the results, the Fine KNN showed better performance. However, the computational time of Weighted KNN is better. The computational time of each dataset is plotted in Figs. 2–5. But based on the accuracy, the Fine KNN is much better. Finally, we compare the proposed method accuracy with some recent techniques, as presented in Tab. 11. From this table, it is showed that the proposed accuracy is much better as compared to the existing techniques.

**Table 11:** Proposed method comparison with existing techniques

| Reference | Dataset | Accuracy (%) |
|---|---|---|
| [30] | IXMAS | 89.22 |
| [31] | IXMAS | 88.76 |
| **Proposed** | **IXMAS** | **89.6** |
| [31] | UCF sports | 90.2 |
| [32] | UCF sports | 92.10 |
| **Proposed** | **UCF sports** | **99.7** |
| [33] | KTH | 89.86 |
| [34] | KTH | 92.25 |
| [35] | KTH | 94.3 |
| **Proposed** | **KTH** | **96.6** |
| [36] | UT Interaction | 87.5 (with 20% training) |
| **Proposed** | **UT Interaction** | **96.7** |

## 4 Conclusion

A new method for the recognition of human actions is presented in this deep learning research work. There are few important steps to the proposed method. In the first step, pre-processing is applied, and video frames are resized according to the target model's input. The pre-trained ResNet50 model is used in the next step and is trained using transfer learning. Employing TL, features are extracted from two successive layers and fused using canonical correlation analysis (CCA). The fused feature vector consists of irrelevant information that is selected using the Shanon Entropy approach. Finally, the selected features are classified using supervised learning classifiers, and the best of them are selected based on the accuracy value. A few well-known datasets are used to evaluate the proposed method and have achieved remarkable accuracy. Based on the accuracy, we conclude that the features extracted through deep learning give better results when handling large-scale datasets. It is also noted that the merging of multilayer features produces better results. But this step affects the efficiency of the system. As a result, the selection process provided more accuracy and also minimizes overall time. In future studies, more complex datasets such as HMDB51 and UCF101 will be considered to evaluate the proposed method.

**Funding Statement:** This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

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

## References

[1] N. Jaouedi, N. Boujnah and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 3, pp. 447–453, 2020.

[2] M. A. Khan, T. Akram, M. Sharif, M. Y. Javed, N. Muhammad *et al.,* "An implementation of optimized framework for action classification using multilayers neural network on selected fused features," *Pattern Analysis and Applications*, vol. 21, no. 4, pp. 1–21, 2018.

[3] A. Sharif, K. Javed, H. Gulfam, T. Iqbal, T. Saba *et al.,* "Intelligent human action recognition: A framework of optimal features selection based on euclidean distance and strong correlation," *Journal of Control Engineering and Applied Informatics*, vol. 21, no. 32, pp. 3–11, 2019.

[4] M. A. Khan, I. Haider, M. Nazir, A. Armghan, H. M. J. Lodhi *et al.,* "Traditional features based automated system for human activities recognition," in *2020 2nd Int. Conf. on Computer and Information Sciences*, Sakaka, SA, pp. 1–6, 2020.

[5] H. Arshad, M. I. Sharif, M. Yasmin, J. M. R. Tavares, Y. D. Zhang *et al.,* "A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition," *Expert Systems*, vol. 5, pp. e12541, 2020.

[6] Y.-D. Zhang, S. A. Khan, M. Attique, A. Rehman and S. Seo, "A resource conscious human action recognition framework using 26-layered deep convolutional neural network," *Multimedia Tools and Applications*, vol. 21, no. 4, pp. 1–23, 2020.

[7] M. A. Khan, K. Javed, S. A. Khan, T. Saba, U. Habib *et al.,* "Human action recognition using fusion of multiview and deep features: an application to video surveillance," *Multimedia Tools and Applications*, vol. 7, pp. 1–27, 2020.

[8] M. Sharif, F. Zahid, J. H. Shah and T. Akram, "Human action recognition: A framework of statistical weighted segmentation and rank correlation-based selection," *Pattern Analysis and Applications*, vol. 23, no. 1, pp. 281–294, 2020.

[9] T. Akram, M. Sharif, N. Muhammad, M. Y. Javed and S. R. Naqvi, "Improved strategy for human action recognition; Experiencing a cascaded design," *IET Image Processing*, vol. 14, pp. 818–829, 2019.

[10] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[11] C. Chen, R. Jafari and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2017.

[12] K. Akila and S. Chitrakala, "Highly refined human action recognition model to handle intraclass variability & interclass similarity," *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 20877–20894, 2019.

[13] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2–3, pp. 107–123, 2005.

[14] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[15] A. Klaser, M. Marszałek and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC 2008-19th British Machine Vision Conf.*, NY, USA, pp. 1–10, 2008.

[16] G. Willems, T. Tuytelaars and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *European Conf. on Computer Vision*, Berlin, Heidelberg, pp. 650–663, 2008.

[17] P. Scovanner, S. Ali and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. of the 15th ACM International Conference on Multimedia*, NY, USA, pp. 357–360, 2007.

[18] H. Wang, A. Kläser, C. Schmid and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[19] M. A. Khan, T. Akram, M. Sharif, M. Y. Javed, N. Muhammad *et al.,* "An implementation of optimized framework for action classification using multilayers neural network on selected fused features," *Pattern Analysis and Applications*, vol. 22, no. 4, pp. 1377–1397, 2019.

[20] A. Ullah, K. Muhammad, I. U. Haq and S. W. Baik, "Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments," *Future Generation Computer Systems*, vol. 96, no. 2, pp. 386–397, 2019.

[21] M. Rashid, M. Alhaisoni, S.-H. Wang, S. R. Naqvi, A. Rehman *et al.,* "A sustainable deep learning framework for object recognition using multilayers deep features fusion and selection," *Sustainability*, vol. 12, pp. 5037, 2020.

[22] L. Wang, P. Koniusz and D. Q. Huynh, "Hallucinating bag-of-words and fisher vector IDT terms for CNN-based action recognition," arXiv preprint arXiv: 1906.05910, 2019.

[23] M. Ma, N. Marturi, Y. Li, A. Leonardis and R. Stolkin, "Region-sequence based six-stream CNN features for general and fine-grained human action recognition in videos," *Pattern Recognition*, vol. 76, no. 26, pp. 506–521, 2018.

[24] F. Afza, M. Sharif, S. Kadry, G. Manogaran, T. Saba *et al.,* "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image and Vision Computing*, vol. 106, no. 27, pp. 104090, 2021.

[25] Q. Xiong, J. Zhang, P. Wang, D. Liu and R. X. Gao, "Transferable two-stream convolutional neural network for human action recognition," *Journal of Manufacturing Systems*, vol. 56, no. 1, pp. 605–614, 2020.

[26] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, A. Bouridane and A. Beghdadi, "A combined multiple action recognition and summarization for surveillance video sequences," *Applied Intelligence*, vol. 51, no. 2, pp. 690–712, 2021.

[27] D. A. Pitaloka, A. Wulandari, T. Basaruddin and D. Y. Liliana, "Enhancing CNN with preprocessing stage in automatic emotion recognition," *Procedia Computer Science*, vol. 116, pp. 523–529, 2017.

[28] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, pp. 770–778, 2016.

[29] N. M. Correa, T. Adali, Y.-O. Li and V. D. Calhoun, "Canonical correlation analysis for data fusion and group inferences," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 39–50, 2010.

[30] C. Wang, G. Hu and Y. Liu, "Multi-views action recognition on deep learning and K-SVD," *Journal of Physics*, vol. 20, pp. 62015, 2019.

[31] K. Kiruba, E. D. Shiloah and R. R. C. Sunil, "Hexagonal volume local binary pattern (H-VLBP) with deep stacked autoencoder for human action recognition," *Cognitive Systems Research*, vol. 58, no. 2, pp. 71–93, 2019.

[32] Y. Yi, P. Hu and X. Deng, "Human action recognition with salient trajectories and multiple kernel learning," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 17709–17730, 2018.

[33] K. Charalampous and A. Gasteratos, "On-line deep learning method for action recognition," *Pattern Analysis and Applications*, vol. 19, no. 2, pp. 337–354, 2016.

[34] Z. Guo, B. Wang and Z. Xie, "A novel 3D gradient LBP descriptor for action recognition," *Transaction on Information and Systems*, vol. 100, pp. 1388–1392, 2017.

[35] A.-A. Liu, Y.-T. Su, W.-Z. Nie and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 102–114, 2016.

[36] S. P. Sahoo and S. Ari, "On an algorithm for human action recognition," *Expert Systems with Applications*, vol. 115, no. 26, pp. 524–534, 2019.