Computers and Electrical Engineering 99 (2022) 107805

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

HAREDNet: A deep learning based architecture for autonomous video surveillance by recognizing human actions☆

Inzamam Mashood Nasir [a,*], Mudassar Raza [a], Jamal Hussain Shah [a], Shui-Hua Wang [b], Usman Tariq [c], Muhammad Attique Khan [d,*]

[a] Department of Computer Science, COMSATS University Islamabad, Wah Campus, Wah Cantt, Pakistan
[b] Department of Mathematics, University of Leicester, Leicester, UK
[c] College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, Al-Kharaj, Saudi Arabia
[d] Department of Computer Science, HITEC University Taxila, Pakistan

ARTICLE INFO

Keywords:
Deep Convolutional Neural Network
Human Action Recognition
Weighted fusion
CvQDA
Encoder-Decoder CNN architecture

ABSTRACT

Human Action Recognition (HAR) is still considered as a significant research area due to its emerging real-time applications like video surveillance, automated surveillance, real-time tracking and resecue missions. HAR domain still have gaps to cover, i.e., random changes in human variations, clothes, illumination, and backgrounds. Different camera settings, viewpoints and inter-class similarities have increased the complexity of this domain. The above-mentioned challenges in uncontrolled environment have ultimately reduced the performances of many well-designed models. The primary objective of this research is to propose and design an automated recognition system by overcoming these afore-mentioned issues. Redundant features and excessive computational time for the training and prediction process has also been a noteworthy problem. In this article, a hybrid recognition technique called HAREDNet is proposed, which has a) Encoder-Decoder Network (EDNet) to extract deep features; b) improved Scale-Invariant Feature Transform (iSIFT), improved Gabor (iGabor) and Local Maximal Occurrence (LOMO) techniques to extract local features; c) Cross-view Quadratic Discriminant Analysis (CvQDA) algorithm to reduce the feature redundancy; and d) weighted fusion strategy to merge properties of different essential features. The proposed technique is evaluated on three (3) publicly available datasets, including NTU RGB+D, HMDB51, and UCF-101, and achieved average recognition accuracy of 97.45%, 80.58%, and 97.48%, respectively, which is better than previously proposed methods.

1. Introduction

People record and upload their own videos while performing different actions. This enriches data on internet to be categorized into most related action classes. Classifying and recognizing these actions manually is a tedious task, as it will require a lot of time and effort [1]. Thus, researchers tried to replace the manual process by automated human activity recognition systems. These automated systems used machine learning algorithms [2] to train models on training videos, which enables the proposed system for automatic recognition

☆ Reviews processed and recommended for publication to the co-Editor-in-Chief by Associate Editor Dr. Yudong Zhang.

* Corresponding authors.
E-mail addresses: imashoodnasir@gmail.com (I.M. Nasir), attique.khan@hitecuni.edu.pk (M.A. Khan).

https://doi.org/10.1016/j.compeleceng.2022.107805
Received 30 September 2021; Received in revised form 2 February 2022; Accepted 7 February 2022
Available online 11 February 2022

of actions in testing videos.

Many successful models have been implemented to recognize actions from dataset images having different illumination, changing viewpoint, intra-class differences and partial occlusions [3]. Generally, the frames of input images are compared with trained samples to compute correlation between it and extract the geometrical consistency between the input video and trained samples. This process is not feasible when the input videos have zooming, intra-class differences, camera motions and dynamic backgrounds. However, this progress is primarily achieved in controlled environment, where the data is collected in constant backgrounds, proper lighting, and surroundings. But when these successful models are tested in real environment, they face difficulties and wrong recognitions [4].

To understand these difficulties, following are few assumptions, which have been made in proposed techniques on controlled environment: a) Pre-processing Assumptions: For an automated system, the selection of related and appropriate features is a primary step. In many cases, a proper and accurate feature extraction one or sequence of pre-processing steps. These steps include segmentation, contrast enhancement, de-noising, de-blurring and removing occlusions. If these pre-processing steps fail, there are chances that the whole model will be affected; b) Data Assumptions: Statistical data is often used in machine learning methods for action recognition, where a trained classifier learns the features from training data. To make the classifier efficient, enough labelled training data is required. If this data is unavailable, insufficient or the data is acquired at run time by applying complex settings or different camera variations, the whole structure of trained model needs to be changed; and c) Model Assumptions: To efficiently train a model, the action is assumed to be a simplified learning objective. If an action is being represented by silhouettes, it can be assumed that the action can be represented by structural or boundary features. So, in that case, local structural or boundary features will be used to train a model, but of course, this assumption can be failed in many real-time applications .

In real-time action recognition applications, one or more assumptions may not be true. If an action in dynamic background needs to be recognized, foreground segmentation may not be available or may not be accurate. When an action needs to be retrieved from a video, the assumption of enough labelled data for training may not be true. Similarly, general assumption of action being a movement of body may not hold to model complex actions.

HAR domain still have gaps to cover, i.e., random changes in human variations, clothes, illumination, and backgrounds [5]. Different camera settings, viewpoints and inter-class similarities have increased the complexity of this domain. The above-mentioned challenges in uncontrolled environment have ultimately reduced the performances of many well-designed models. The primary objective of this research is to propose and design an automated recognition system by overcoming these afore-mentioned issues. In this article, a hybrid technique is proposed to recognize human actions. There are four (4) steps, including a) deep feature extraction using Encoder-Decoder Network (EDNet); b) local feature extraction using improved Scale-Invariant Feature Transform (iSIFT), improved Gabor (iGabor) and Local Maximal Occurrence (LOMO) feature descriptors; c) distance learning approach to find the relevance among features using Cross-view Quadratic Discriminant Analysis (CvQDA) algorithm; and d) weighted fusion strategy to select efficient and relevance portion of features. The proposed Human Action Recognition Encoder Decoder Network (HAREDNet) is tested on publicly available datasets including NTU RGB+D, HMDB51 and UCF-101and achieved improved results compared to previous techniques.

The rest of the article is as follows: proposed model is described in Section 2. Section 3 describes the experimental results, ablation analysis, and discussion by comparing the achieved results with state-of-the-art techniques. Section 4 concludes this article by mentioning the limitations and future directions of this work.

2. Literature review

HAR can be classified into two major types, a) recognition is carried out using the hand-crafted features and b) recognition is performed using machine learning-based methods. A Bag-of-Features (BoF) based Global and Local Zernike Moment (GLZM) technique was proposed by joining local and global predictors. Global features were insufficient to correctly represent similar activities, i.e., walking, running, and jogging. To overcome this issue, initially, local temporal features were calculated, and then local and global predictors were fused using General Linear Model (GLM) with various polynomial orders. The global features represented the human body's region performing activities, while local features represented the activity information. Whitening transformation was then applied to preprocess both local and global features. In the end, multi-class SVM was employed to recognize human activities from publicly available datasets Weizmann, UCF-sports, and KTH with accuracies of 98.90%, 86.40%, and 89.03%, respectively [6].

In another research, the authors introduced hierarchical RCNN to recognize the skeleton-based actions. These skeletons were divided in five different parts in accordance with the physical layouts. Method is implemented on three different datasets MSRAction3D, Berkeley MHAD and HDM05. Effective results are derived which are 94.49%, 100% and 96.92% respectively [7].

CNN is used to map the temporal relationships using LSTM networks. A deep fusion framework is utilized to effectively exploit the temporal features from LSTM and spatial features of CNN models. The proposed method is applied on UCF11, UCF-Sports and HMDB datasets and achieved accuracies of 94.6%, 99.1% and 69.0% respectively [8]. LSTM and CNNs of similar model complexities are used to extract the descriptors for HAR. The proposed model is tested on datasets UCF-101 and achieved 93.65 accuracy, while 66.2% accuracy was recorded on HMDB-51 dataset [9].

The existing approaches are quite impressive for human action recognition, but a very small portion highlights the use of deep learning in this domain. However, there are some weaknesses and certain limitations which encourage researchers to continue research on this active domain. In this study, the focus is to detect, recognize and rectify human actions and implement the proposed methods in real life applications. Few limitations of the literature are: 1) A single feature vector cannot possibly represent all kind of human action related properties [10]; 2) Although high precision and low false positive rate is achieved in some of the techniques [11], but these techniques are tested on homogenous datasets rather than using datasets having diversified variety of human actions; 3)

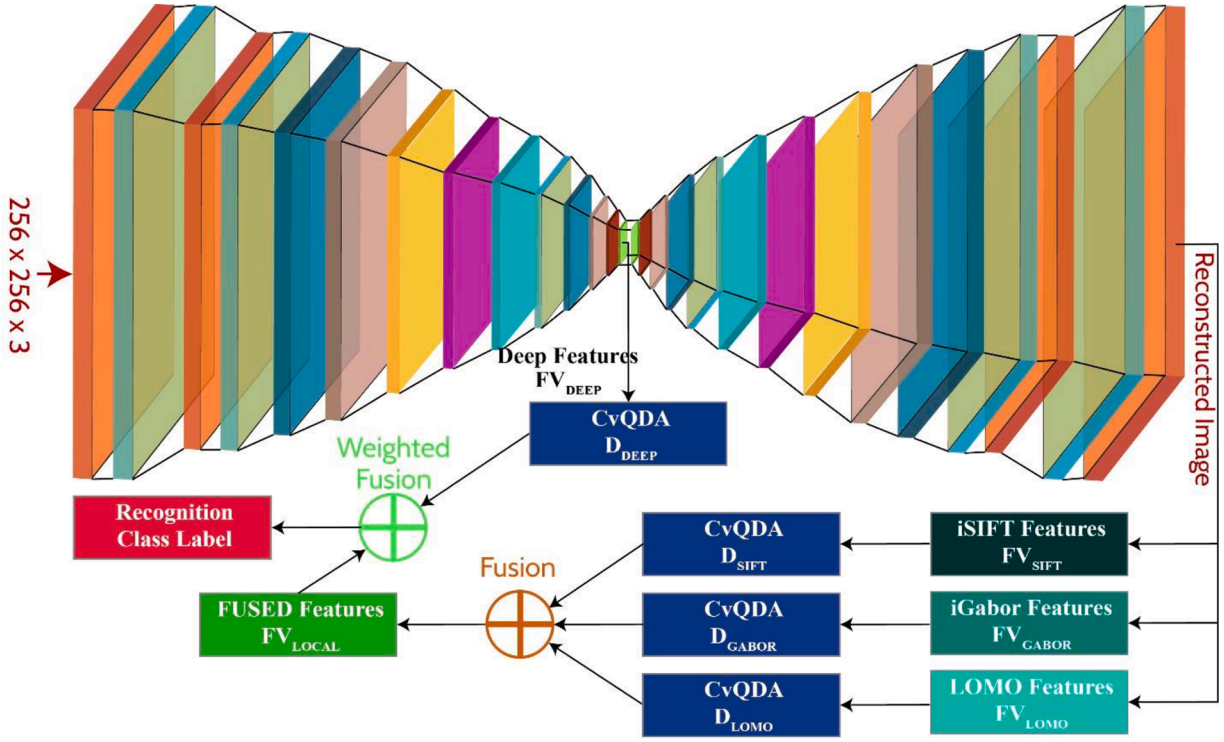

Fig. 1.. Proposed HAREDNet model.

Weak segmentation of human from complex background, results in missing details in the action recognition process [12]; and 4) Less work has been based on deep learning methods, mostly rely on matured algorithms from other fields and attained good quality performance over conventional techniques like image processing or morphological operations [13]. Furthermore, a smaller number of training examples used for model learning due to non-availability of big data. However, synthetic text image data is available, but few approaches were reported to use such dataset. In the past decade, impressive work has been carried out on HAR. However, numerous proposed techniques had high false positive rate, computationally complex and highly time-consuming structure.

3. Human Action Recognition through Encoder-Decoder Network (HAREDNet)

Although Convolutional Neural Networks (CNNs) have proved their credibility to extract features by obtaining impressive results on various medical, agricultural, and related domains, the basic architecture of CNNs has not been efficient to discriminate the relation of sequences. In simple words, stand-alone CNN architecture cannot measure temporal dependencies of consecutive frames in HAR videos. Extracted features at different layers must possess a hierarchical architecture with explicit dependencies as these features cannot be taken as isolated and independent values.

To extract relevant features for HAR images, typical EDNet is extended to a hybrid model, which takes advantage of both CNNs and Recurrent Neural Networks (RNNs). The proposed hybrid model firstly extracts shallow sequence features using CNN architecture and then encodes these features into a single feature vector using RNN architecture by considering the dependency between extracted features at different levels. RNN-encoded features are termed as deeper features with the capability of extracting subjects from a video frame.

The proposed HAREDNet model is shown in Fig. 1. The encoder contains a) convolutional layers to extract features at each level and b) recurrent layers to encode the extracted features into a single representation of the subject. The decoder contains a) recurrent layers to predict and prioritize extracted features at each level and b) deconvolutional layers to reconstruct the segmented subjects from a given frame. The decoded image is then used to extract local features (i.e., iSIFT, iGabor, and LOMO). Both deep and local features are processed using the CvQDA technique followed by fusion of local features, and at the end, feature vector of local fusion is further fused with deep features for the recognition task.

In the encoder architecture, EF_1,EF_2,…, EF_n represents the extracted CNN features while in decoder architecture, DF_1,DF_2,…, DF_n represents the reconstructed CNN features to reconstruct the segmented subjects. In conventional RNN, the spatial dependency between adjacent pixels is learned using Long-Short Term Memory (LSTM) units as building blocks. Due to the complex training process of LSTM, Gated Recurrent Unit (GRU) was proposed, which performs equivalent to LSTM without having separate memory cells. It has been observed that GRUs have achieved evident performance as compared to LSTM with fewer parameters required. To further improve the efficiency and achieve maximum computational output of GRUs, a Bidirectional GRU (BiGRU) is defined as:

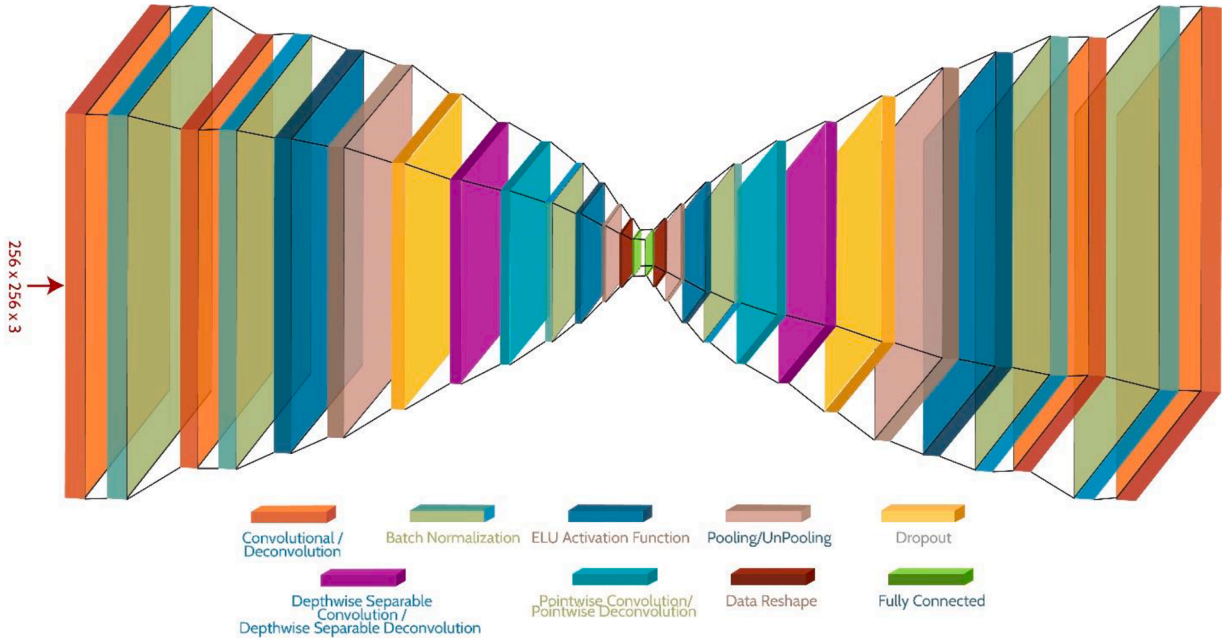

Fig. 2.. Structure of the proposed EDNet.

Table 1
Parameters of proposed EDNet.

Encoding					Decoding				
#	Layer	Params	Input	Output	#	Layer	Params	Input	Output
1	C	C = 32 *K*= (1193) P [0.64]	128 × 1 × 64 × 480	128 × 32 × 64 × 480	**34**	dC	C = 32 *K*= (1193) P [0.64]	128 × 1 × 64 × 480	128 × 32 × 64 × 480
2	BN	C = 32	128 × 32 × 64 × 480	128 × 64 × 64 × 480	**33**	BN	C = 32	128 × 32 × 64 × 480	128 × 64 × 64 × 480
3	C	C = 64 *K*= (64,1) P [0.0]	128 × 64 × 64 × 480	128 × 64 × 1 × 480	**32**	dC	C = 64 *K*= (64,1) P] 0.0]	128 × 64 × 64 × 480	128 × 64 × 1 × 480
4	BN	C = 64	128 × 64 × 1 × 480	128 × 64 × 1 × 480	**31**	BN	C = 32	128 × 64 × 1 × 480	128 × 64 × 1 × 480
5	ELUAF	–	128 × 64 × 1 × 480	128 × 64 × 1 × 480	**30**	ELUAF	–	128 × 64 × 1 × 480	128 × 64 × 1 × 480
6	P	[1,2]	128 × 64 × 1 × 480	128 × 64 × 1 × 192	**29**	uP	[1,2]	128 × 64 × 1 × 480	128 × 64 × 1 × 192
7	D	0.5	128 × 64 × 1 × 192	128 × 64 × 1 × 192	**28**	D	0.5	128 × 64 × 1 × 192	128 × 64 × 1 × 192
8	DSC	C = 64 *K*= (1,93) P [0.48]	128 × 64 × 1 × 192	128 × 64 × 1 × 192	**27**	DSdC	C = 64 *K*= (1,93) P [0.48]	128 × 64 × 1 × 192	128 × 64 × 1 × 192
9	PC	C = 32 *K*= (1,33) P [0.0]	128 × 64 × 1 × 192	128 × 32 × 1 × 192	**26**	PdC	C = 32 *K*= (1,33) P [0.0]	128 × 64 × 1 × 192	128 × 32 × 1 × 192
10	BN	C = 32	128 × 32 × 1 × 192	128 × 32 × 1 × 192	**25**	BN	C = 64	128 × 32 × 1 × 192	128 × 32 × 1 × 192
11	ELUAF	–	128 × 32 × 1 × 192	128 × 32 × 1 × 192	**24**	ELUAF	–	128 × 32 × 1 × 192	128 × 32 × 1 × 192
12	P	[1,16]	128 × 32 × 1 × 192	128 × 32 × 1 × 30	**23**	uP	[1,2]	128 × 32 × 1 × 192	128 × 32 × 1 × 30
13	DR	–	128 × 32 × 1 × 30	128 × 32 × 30	**22**	DR	–	128 × 32 × 1 × 30	128 × 32 × 30
14	FC	I: 32 O: 32	128 × 32 × 30	128 × 32 × 32	**21**	FC	I: 32 O: 32	128 × 32 × 30	128 × 32 × 32
15	ELUAF	–	128 × 32 × 32	128 × 32 × 32	**20**	ELUAF	–	128 × 32 × 32	128 × 32 × 32
16	BiGRU	I: 32 H: 32 L: 1	128 × 32 × 32	128 × 32 × 64	**19**	BiGRU		128 × 32 × 32	128 × 32 × 64
17	FC	I: 64 O: 32	128 × 32 × 64	128 × 32 × 32	**18**	FC	I: 32 O: 64	128 × 32 × 32	128 × 32 × 64

$$Y_F = (1 - B_L) \otimes Y_{L-1} + B_L \otimes \widehat{Y}_L \tag{1}$$

where, Y_F is the output for frame *F*, *L* is the current layer, B_L and $\widehat{Y}_L$ are updated gate matrices, Y_{L-1} is the output of the previous layer with $Y_0 = 0.024$ and $\otimes$ is matrix multiplication. If the input to the network is defined by I_c, weights for hidden and other than hidden layers are defined as V_c and W_c respectively and N_c is the bias value with *c* is the count of total feed-forward propagation layers in a network, then B_L is calculated as:

$$B_L = \mathscr{S}((W_c \times I_c) + (V_c \times Y_{L-1}) + N_c) \tag{2}$$

where, $\mathscr{S}$ is a sigmoid function. $\widehat{Y}_L$ is calculated as:

$$\widehat{Y}_L = \Gamma((W_h \times I_h) + V_h(Y_L \otimes Y_{L-1}) + N_h) \tag{3}$$

where, Γ is a tangent function, and *h* is the count of total backward propagation layers in a network. During the implementation, feed-forward and backward propagation layers calculate features by iteratively performing calculations on outputs of previous layers, weight matrices, and bias values. A single feature vector at the GRU of the network can be defined as:

$$F_B = B_L \boxplus \widehat{Y}_L \tag{4}$$

where, $\boxplus$ is a matrix concatenation function and *B* indicates total extracted features against a single frame. Feature vector of a whole network can be expressed by:

$$FV = (F_1, \ldots F_b, \ldots F_B) \tag{5}$$

where, *b* is some arbitrary central value, as different network layers can output different number of features. Structure of the proposed EDNet is shown in Fig. 2. To better understand human actions in a frame, spatial features are learned through convolutional layers, while temporal features are learned through the recurrent layers. Thus, extracted Feature Vector (FV) learns the complete information regarding a single frame, which is beneficial for real-time applications like person identification [14] and video surveillance [15]. The efficiency of proposed architecture is improved by updating GRU input for each batch during training phase.

The proposed EDNet uses different layers with distinct Parameters (Params) as Channels (C) and Kernels (K), Inputs (I), Outputs (O), Hidden Layers (HL), and number of layers (L). There are several layers, i.e., Convolutional (C), Batch Normalization (BN), ELU Activation Function (ELUAF), Pooling (P), Dropout (D), Depthwise Separable Convolution (DSC), Pointwise Convolution (PC), Data Reshape (DR....), Fully Connected (FC), Deconvolution (dC), UnPooling (uP), Depthwise Separable Deconvolution (DSdC) and Pointwise Deconvolution (PdC). Parameters of the proposed network are presented in Table 1.

Encoder part of the proposed network is also used to extract deep features by adding an extra DR layer, which reshapes the output of layer 13 into 744×1 for a single image. Reconstructed image from layer 34 is then forwarded to extract hand-crafted features. Characteristics like angle, curvature, and appearance are key elements to describe subjects in any frame. Therefore, to extract these sensitive characteristics, texture, edge, and color features like iSIFT, iGabor and LOMO are employed in this work.

3.1. iSIFT features

SIFT [16] is invariant to the rotation and scale of an image; thus, it can represent any object's local appearance at specific interest points. Presence of impact like illumination effect, noise, changes in viewpoints, inter-class and intra-class similarities reduce the effectiveness of traditional SIFT features. Thus, iSIFT features are proposed in this article, which are robust to these challenges. Suppose the scale space of an input image is denoted by $\aleph(idx, idy, \delta_a)$, which is obtained by a process of convolution over an input image $I(idx, idy)$ and gaussian of variable scale $\vartheta(idx, idy, \delta_a)$.

$$\aleph(idx, idy, \delta_a) = I(idx, idy) \bullet \vartheta(idx, idy, \delta_a) \tag{6}$$

here, $\bullet$ denotes the operation of convolution. Gaussian of variable scale with scale parameter δ_a is further defined as:

$$\vartheta(idx, idy, \delta_a) = \frac{1}{2\pi\delta_a^2} exp^{-\frac{idx^2 + idy^2}{2\delta_a^2}} \tag{7}$$

The stable locations of key points in $\vartheta(idx, idy, \delta_a)$ are computed by calculating the gaussian difference modified by multiplicative scalar *z*:

$$\vartheta(idx, idy, \delta_a) = \aleph(idx, idy, z\delta_a) - \aleph(idx, idy, \delta_a) \tag{8}$$

The Laplacian of Gaussian $\delta_a^2 \mathbb{C}^2 \vartheta$ for scale normalization form indifference of gaussian function. Therefore:

$$\vartheta(idx, idy, z\delta_a) - \vartheta(idx, idy, \delta_a) \approx (z - 1)\delta_a^2 e^2 \vartheta \tag{9}$$

This proves that the difference of gaussian function differs a constant *z* to normalize the Laplacian of variable scale by incorporating

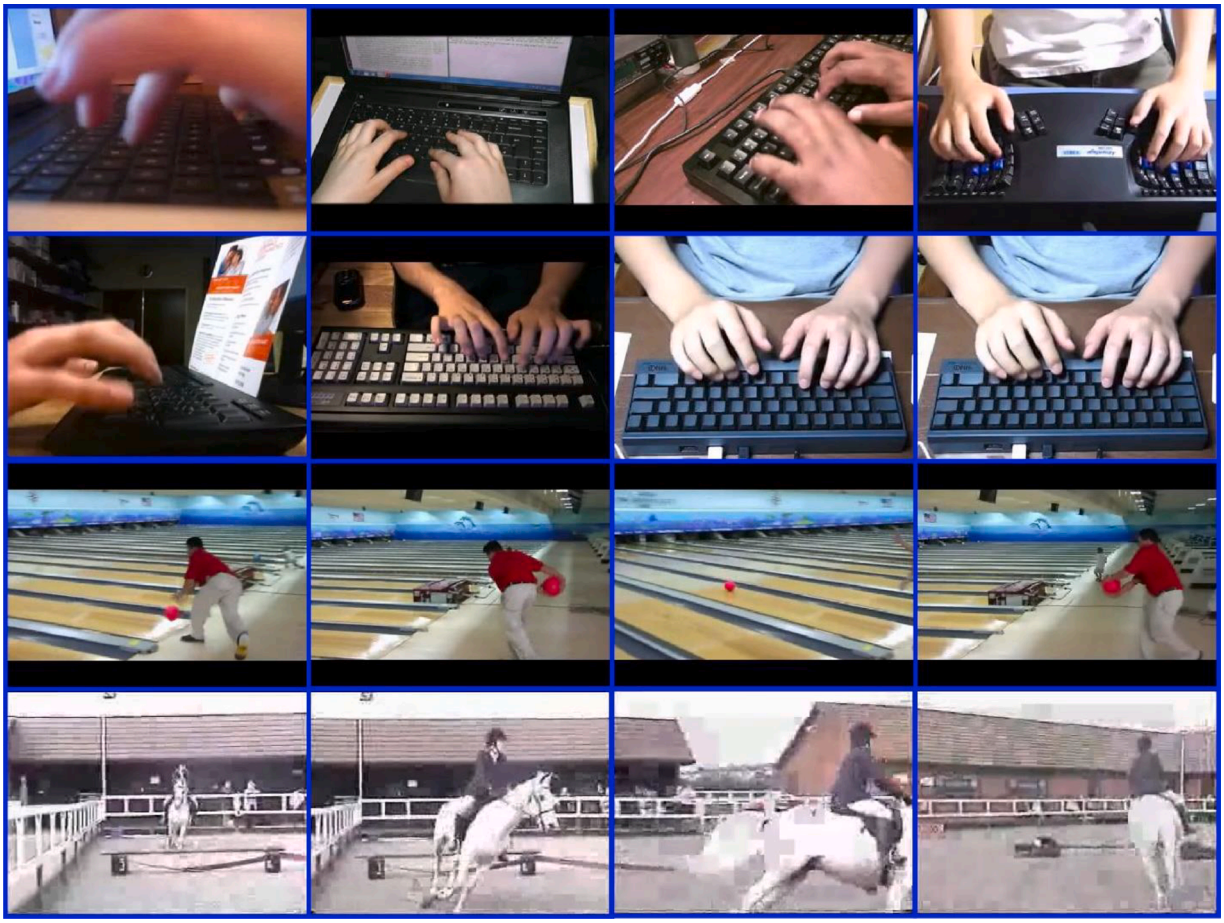

Fig. 3.. Intra-class variations and viewpoint variations in a single class (first two rows show intra-class variations while last two rows show different viewpoint variations for a single class).

scale normalization δ_a^2.

3.2. iGabor features

The algorithms of texture analysis are broadly utilized as wavelet transformations in many multi-resolution filtering techniques. The uncertainty in frequency and space of a two-dimensional joint image is minimized using a multi-resolution filtering technique called Gabor filtering [17]. Unlike the traditional Gabor features, iGabor finds the orientation-based edge and line detection using a combination of gabor filters, histogram equalization and convolution operation These extracted features characterizes the texture information of a given region. iGabor representations are effective for image understanding and subject recognition. To minimize the computational cost, iGabor features are simplified into three basic representations using gabor function $\gamma_{h,v}$ defined as:

$$\gamma_{h,v}(idx, idy) = \gamma_{\bar{c}}(\bar{r}) = \left(\frac{\|\bar{c}\|}{\varepsilon^2}\right)\left(exp^{-\frac{\|\bar{c}\|^2 \times \|\bar{r}\|}{2\varepsilon^2}}\right)\left(exp^{-x\bar{c}\times\bar{r}} - exp^{-\frac{\varepsilon^2}{2}}\right) \tag{10}$$

where, $\bar{r} = (idx, idy)$ denotes the spatial domain variable and $\bar{c}$ denotes the frequency vector to determine the direction and scale of gabor function and is calculated as $\bar{c} = c_e \times exp^{x\mu_v}$, where, $c_e = \frac{c_{max}}{p^h}$ and $c_{max} = \frac{\pi}{2}$. During the iGabor feature extraction for this work, the values of $p = 2$, $h = 1, 2, 3, 4, 5$, $\mu_z = \frac{\pi v}{8}$ and $x = v = \tau = 1, 2, 3, \ldots, 8$ are used. In the first of three basic representations of gabor, the magnitude of an input image is generated by performing the convolutions and sum it with gabor function in five (5) different horizontal directions with fixed scales. This is calculated as:

$$\gamma_H(idx, idy) = \left| I(idx, idy) \bullet \sum_{h=1}^{5} \gamma_h(idx, idy) \right| \tag{11}$$

here, $\gamma_H(idx,idy)$ denotes the five different outputs to represent the decomposition of an input image in horizontal directions. In the

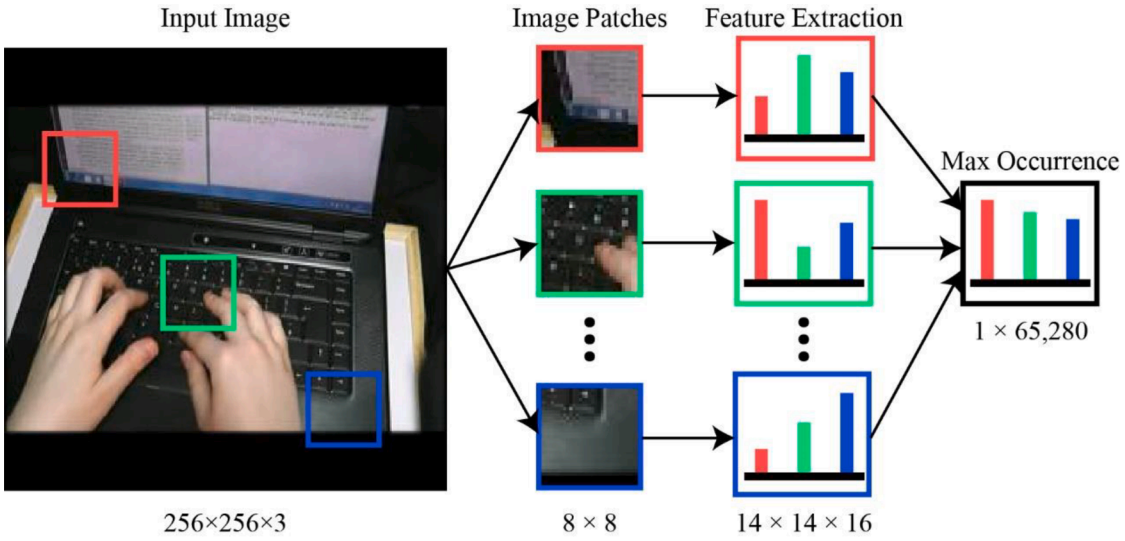

Fig. 4.. Procedure of extracting LOMO features.

second step, convolutions and summation are performed using gabor function in eight (8) different vertical directions with fixed scales as:

$$\gamma_V(idx, idy) = \left| I(idx, idy) \bullet \sum_{v=1}^{8} \gamma_v(idx, idy) \right| \tag{12}$$

here, $\gamma_V(idx,idy)$ are the outputs, which represent the decomposed input image in eight directions. In the last step, both horizontal and vertical decompositions are fused and convolved with the input image to form a mapped image $\gamma_{out}(idx,idy)$ as:

$$\gamma_{out}(idx, idy) = \left| I(idx, idy) \bullet \left[\left(\sum_{h=1}^{5} \gamma_h(idx, idy) \right) \otimes \left(\sum_{v=1}^{8} \gamma_v(idx, idy) \right) \right] \right| \tag{13}$$

3.3. LOMO features

HAR images are well-described through color and viewpoint. Different camera settings, illumination conditions, and capturing angles vary in an uncontrolled environment. Therefore, a single video can have multiple colors and viewpoint variations, as shown in Fig. 3. To reduce these impacts, LOMO features are employed in this work. Initially, all frames are preprocessed using the Retinex algorithm [18], which calculates and fixes the color and lightness effect. The frames are processed to keep color and light consistent throughout the whole video by processing shadowed regions and vivid colors. The multi-scale Retinex algorithm is implemented to combine the large-scale and small-scale retinex simultaneously for tonal rendition and dynamic range compressions. The algorithm achieves efficient subject visual representation for both dynamic range compressions and color consistency.

A total of two scales (surrounded and central retinex) are utilized with standard deviation $\sigma = 10$ and $\sigma = 30$. These scales are finalized after extensive experiments. Along with this, offset and gain parameters are automatically computed to stretch the intensities of pixels, linearly between 0 and 255. Color features of preprocessed frames are extracted by converting frames into HSV color space for calculating HSV color histogram. These color features are merged with Scale Invariant Local Ternary Pattern (SILTP) features to tackle the illumination changes. SILTP is an improved form of Local Binary Pattern (LBP), as it introduced robustness against frame noises, achieved invariance to changes in intensity and invariance in comparing tolerance.

Subjects in HAR usually have various viewpoints, i.e., a subject of the video in an uncontrolled environment may appear in frontal, side, or back view under different camera positions. Thus, identifying subjects in different and evolving viewpoints is a tedious task. To solve this issue, the usage of a sliding window is proposed to represent local information of the subjects in a frame. A sliding window of size 8×8, with a stride of 3 pixels, is used to identify local patches in a 256×256 frame. Each sliding window extracts two SILTP scales and a $14 \times 14 \times 16$-bin HSV histogram, where every bin illustrates the probability occurrence of a single pattern in a patch. Varying viewpoint changes are located by checking all patches at a particular horizontal position, and the probability of a single pattern is maximized among all patches. Fig. 4 represents the overall procedure of extracting LOMO features.

A multi-scale description of subjects is considered by building a triple-scale pyramid representation, which reduces the frame of size 256×256 by applying local max-pooling by three 3×3 pooling operations. After the concatenation of computed LOMO features, the final feature vector has $14 \times 14 \times 16$ color bins, two SILTP bins of size 4^3,15 horizontal and five vertical groups, which makes 65, 280 features for a single frame. Log transformation is applied to suppress bins with tremendous values and normalize SILTP and HSV

features to unit length.

3.4. Cross-view Quadratic Discriminant Analysis (CvQDA)

Cross-view Quadratic Discriminant Analysis (XQDA) was proposed to maintain the relation between the Bayesian face method and direct simplicity. A gaussian model was used to distribute the difference between interclass and intraclass features. In this article, CvQDA is proposed to calculate ratio of two gaussian distributions and derive the Levenshtein distance. Interclass and intraclass covariance matrices can be defined as:

$$CM_1 = \frac{1}{SP_1}\sum_{mat=1}\left(SM_i - SM_j\right)\left(SM_i - SM_j\right)^T \tag{14}$$

$$CM_2 = \frac{1}{SP_2}\sum_{mat=0}\left(SM_i - SM_j\right)\left(SM_i - SM_j\right)^T \tag{15}$$

where, SM_i and SM_j are random patches, mat is an indicational sum vector of SM_i and SM_j. If both SM_i and SM_j belongs to the same subject, $mat = 1$, else $mat = 0$. SP_1 indicates total similar sample pairs, while SP_2 indicates total dissimilar sample pairs. Feature vector FV is obtained optimizing the generalized Rayleigh quotient:

$$FV = \frac{W^T \circ CM_2 W}{W^T \circ CM_1 W} \tag{16}$$

where $\circ$ denotes the dot product. The Levenshtein distance (LD) of two viewpoints for same subject in FV can be calculated as:

$$LD\left(SM_i,\ SM_j\right) = \mathrm{FV} + \left[\left(SM_i,\ SM_j\right)^T W \times \left(\left(W^T CM_1 W\right)^{-1} - \left(W^T CM_2 W\right)^{-1}\right) \times W^T\left(SM_i,\ SM_j\right)\right] \tag{17}$$

3.5. Weighted fusion strategy

Deep features can have some noise due to evolving background in uncontrolled videos; thus, they may not be effective in locating subjects efficiently. Deep features also rely on substantial sample tags, while the local features are independent of these sample tags, which makes these features reliable to recognize viewpoint appearances. If all these features are fused, it can overcome their lack. After extracting deep features, iSIFT features, iGabor features, and LOMO features, CvQDA tries to calculate the feature distance between features of same frame using Eq. (17). To collaborate all traditional and deep features, the final distance calculated for all extracted feature types is used to sort and combine all features into a single feature vector as:

$$FV_{FUSED} = \Upsilon FV_{DEEP} + (1 - \Upsilon)(FV_{SIFT} + FV_{GABOR} + FV_{LOMO}) \tag{18}$$

here, Υ is a parameter to control the relative importance among the deep and local features. After extensive experiments, it is noted that setting $\Upsilon = 0.6$ and giving 60% weightage to deep features and 40% to local features have achieved maximum results.

4. Experiments and results

The proposed technique is validated through intensive experiments, where different performance matrices are used to verify the efficiency. In this section, experimental results on selected datasets are presented, and a comprehensive discussion is provided, where the proposed technique is compared with state-of-the-art techniques.

4.1. Datasets and experimental settings

The proposed technique is trained, tested, and validated using three (3) publicly available standard benchmark datasets.

NTU RGB+D (D1) [19] is a large dataset containing 56,880 videos across 60 classes in an uncontrolled environment. There are nine health-related classes (falling down, staggering, sneezing, etc.), 40 daily life classes (reading, eating, drinking, etc.), and 11 mutual action classes (hugging, kicking, punching, etc.). Videos are captured from 40 subjects of age between 10 and 35 years, using Microsoft Kinect v2. Videos are recorded with a resolution of 1920 × 1080 with 30 fps. This dataset has two modules, i.e., Cross-Subject (CS) and Cross-View (CV).

HMDB51 (D2) [20] is another largescale dataset for HAR in an uncontrolled environment. This dataset contains 51 action classes, divided into five types: a) general body movements (sit down, backhand flip, dive, etc.); b) General facial actions (talk, chew, laugh, etc.); c) Body movements for human interaction (sword fight, shake hands, punch, etc.); d) Body movements with object interaction (hit something, draw sword, brush hair, etc.); and e) Facial actions with object manipulation (drink, eat, smoke, etc.). There are 6766 video clips in 51 classes, where each class contains a minimum of 101 videos. Video clips have a frame rate of 30fps, while the height is fixed at 240 pixels. All videos have different widths as per the aspect ratio.

UCF-101 (D3) [21] dataset contains 101 action classes, which are categorized into five types: a) Sports (Throw Discus, Volleyball Spiking, Tennis Swing, etc.); b) Playing Musical Instruments (Playing Violin, Playing Tabla, Playing Piano, etc.); c) Human-Human

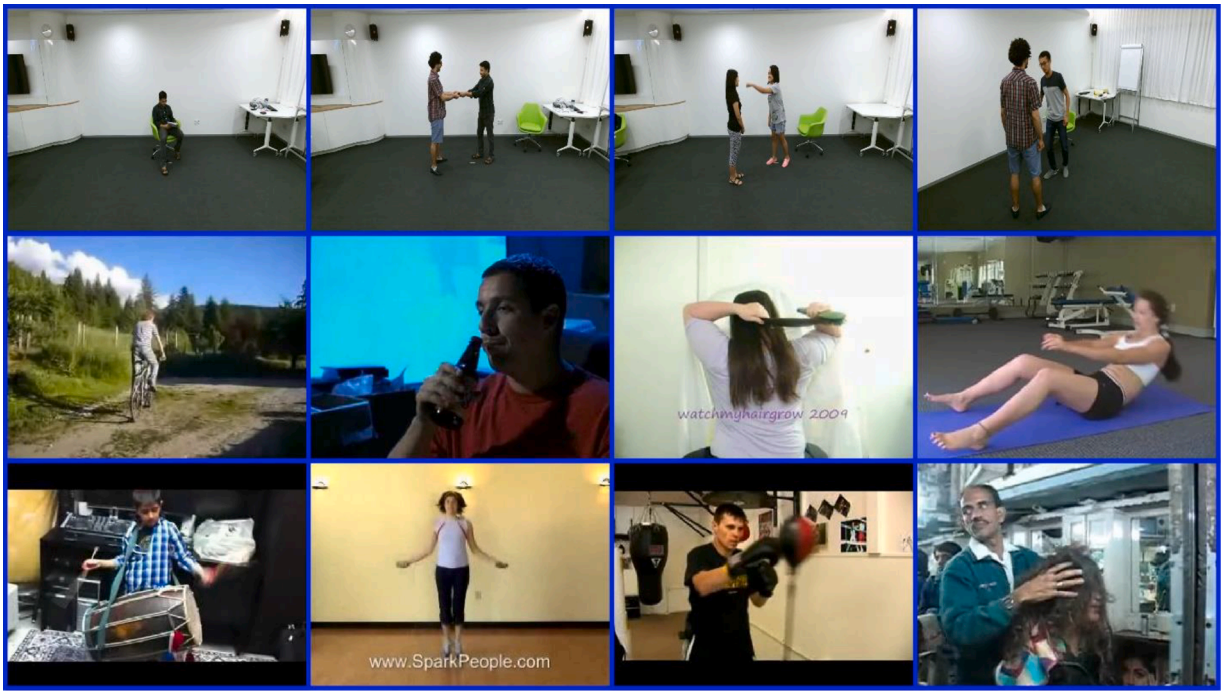

Fig. 5.. Sample images from selected dataset. First row D1: (left-to-right: writing, giving something to other person (gstop), punching, kicking other person (kop)); second row: D2 (left-to-right: ride_bike, drink, brush_hair, situp); and third row D3: (left-to-right: PlayingDhol (pd), JumpRope (jr), BoxingPunchingBag (bpb), HeadMassage (hm)).

Table 2
Recognition results of different classifiers on D1.

Classifier	Acc (%)		TT (m)	PT (s)
	CS	CV		
MSVM	**95.63 ± 1.49**	**99.26 ± 0.64**	**242 ± 36**	**0.45 ± 0.21**
ESD	76.94 ± 2.47	71.22 ± 2.43	279 ± 46	1.86 ± 0.43
FkNN	85.75 ± 2.72	89.02 ± 2.65	304 ± 19	0.75 ± 0.35
WkNN	80.84 ± 3.87	82.12 ± 3.79	260 ± 27	0.86 ±0.22
ES-kNN	60.97 ± 3.23	64.55 ± 3.49	426 ± 28	1.18 ±0.45
QSVM	63.73 ± 2.96	66.59 ± 2.52	376 ± 37	1.22 ± 0.26
LDA	90.28 ± 1.24	92.95 ± 1.98	444 ± 41	0.96 ± 0.27
BTree	74.43 ± 4.88	76.17 ± 4.62	341 ± 33	1.14 ± 0.38

Interaction (Salsa Spins, Military Parade, Band Marching, etc.); d) Body-Motion Only (Rope Climbing, Lunges, Jumping Jack, etc.); and e) Human-Object Interaction (Mixing Batter, Jump Rope, Hula Hoop, etc.). There are a total of 13,320 videos with a frame rate of 25fps and a resolution of 320 × 240. Fig. 5 shows sample images from D1, D2, and D3.

The proposed technique is trained on Core i7, 10th generation having CPU @ 3.9*GHz*and NVIDIA GeForce RTX 2060 Super 8 GB GDDR6 with 256 − *bit* Memory Bus and 1650*MHz* Boost Clock. MATLAB 2020a is used to train and test the proposed technique, where the minibatch size of 128, an initial learning rate of 0.00001, the momentum of 0.6, and maximum epochs of 750 are set. The learning rate is decreased after 7 epochs by the factor of 7.

A standard data-split approach of 70 − 15 − 15 for training, validation, and testing is employed. A total of six (6) performance evaluation matrices are used, including Accuracy (Acc), Correct Recognition Rate (CRR), Precision (Pre), Sensitivity (Sen), Training Time (TT), and Prediction Time (PT). A total of eight (8) classifiers, including different kernels of SVM and kNN are used to extract the results, i.e., Multi-class Support Vector Machine (MSVM), Ensemble Subspace Discriminant (ESD), Fine kNN (FkNN), Weighted kNN (WkNN), Ensemble Subspace kNN (ES-kNN), Quadratic SVM (QSVM), Linear Discriminant Analysis (LDA) and Bagged Tree (BTree). All experiments are performed at least five times using the same experimental setup.

4.2. Experimental results

The proposed model is evaluated on selected datasets using different experiments. Performance is measured by utilizing different classifiers and stand-alone (local and deep) features to note the proposed feature fusion and CvQDA algorithm's impact. The impact of

Table 3
Recognition results obtained through different features on D1.

Model	Feature Extractor	CvQDA		Acc (%)		TT (m)	PT (s)
		No	Yes	CS	CV		
Local Features	SIFT	✓		70.75 ± 4.92	73.29 ± 3.74	456 ± 12	1.37 ± 0.36
			✓	74.24 ± 2.22	77.76 ± 1.87	438 ± 37	1.21 ± 0.2
	iSIFT	✓		75.54 ± 4.68	78.37 ± 4.83	430 ± 25	1.14 ± 0.38
			✓	81.02 ± 3.41	85.08 ± 3.67	456 ± 15	1.48 ± 0.42
	Gabor	✓		53.24 ± 2.92	57.82 ± 2.87	358 ± 11	1.19 ± 0.39
			✓	58.59 ± 2.62	62.19 ± 1.95	390 ± 16	0.48 ± 0.22
	iGabor	✓		63.12 ± 4.23	68.65 ± 4.68	359 ± 49	1.15 ± 0.36
			✓	67.64 ± 3.11	69.35 ± 3.37	394 ± 34	1.03 ± 0.22
	LOMO	✓		79.59 ± 2.11	82.24 ± 1.97	224 ± 13	1.07 ± 0.33
			✓	81.97 ± 2.50	85.77 ± 2.63	292 ± 26	1.09 ± 0.41
Deep Features	EDNet	✓		79.55 ± 4.64	82.43 ± 4.63	**211 ± 12**	0.86 ± 0.58
			✓	83.28 ± 4.33	86.34 ± 4.67	234 ± 47	0.73 ± 0.45
Proposed	HAREDNet (w/o LOMO)	✓		83.87 ± 1.86	86.52 ± 1.36	357 ± 39	0.82 ± 0.25
			✓	86.87 ± 2.93	89.81 ± 2.61	374 ± 29	0.95 ± 0.36
	HAREDNet	✓		91.44 ± 2.44	94.49 ± 2.54	229 ± 28	0.76 ± 0.41
			✓	**95.63 ± 1.49**	**99.26 ± 0.64**	242 ± 36	**0.45 ± 0.21**

Table 4
Recognition results of different classifiers on D2.

Classifier	Acc (%)	CRR (%)	Pre (%)	Sen (%)	TT (m)	PT (s)
FkNN	**80.58 ± 2.19**	**82.45 ± 2.93**	**82.86 ± 2.11**	**81.21 ± 2.46**	**311 ± 14**	**0.41 ± 0.26**
ESD	64.1 ± 2.67	65.02 ± 2.74	64.88 ± 2.46	65.44 ± 2.44	392 ± 47	0.96 ± 0.49
MSVM	70.95 ± 3.71	71.25 ± 3.27	70.44 ± 3.81	69.76 ± 3.04	383 ± 38	0.84 ± 0.35
WkNN	65.86 ± 1.51	64.94 ± 1.76	63.28 ± 1.75	63.22 ± 1.99	370 ± 13	0.61 ± 0.53
ES-kNN	78.57 ± 2.51	79.75 ± 2.46	78.97 ± 2.82	79.92 ± 2.24	421 ± 11	0.76 ± 0.28
QSVM	74.86 ± 2.79	73.25 ± 3.66	76.27 ± 3.96	74.48 ± 3.27	360 ± 29	0.91 ± 0.41
LDA	75.62 ± 3.09	76.29 ± 4.41	76.28 ± 4.57	76.47 ± 4.97	383 ± 18	0.89 ± 0.28
BTree	65.31 ± 3.96	65.03 ± 4.23	66.63 ± 4.15	66.04 ± 4.29	326 ± 33	1.08 ± 0.26

Table 5
Recognition results obtained through different features on D2.

	Feature Extractor	CvQDA		Acc (%)	CRR (%)	TT (m)	PT (s)
		No	Yes				
Local Features	SIFT	✓		56.54 ± 2.17	58.21 ± 2.36	372 ± 35	0.73 ± 0.23
			✓	59.84 ± 3.74	62.36 ± 3.66	355 ± 41	1.1 ± 0.28
	iSIFT	✓		62.34 ± 4.49	63.69 ± 4.62	410 ± 45	1.02 ± 0.32
			✓	64.41 ± 1.86	63.58 ± 1.86	345 ± 37	0.63 ± 0.21
	Gabor	✓		66.88 ± 3.52	69.78 ± 3.07	354 ± 50	1.28 ± 0.48
			✓	70.25 ± 4.14	73.98 ± 3.53	364 ± 14	1.44 ± 0.28
	iGabor	✓		73.68 ± 4.34	73.64 ± 4.93	**277 ± 34**	0.83 ± 0.38
			✓	77.92 ± 1.51	76.74 ± 1.59	377 ± 46	1.26 ± 0.36
	LOMO	✓		66.91 ± 1.54	66.07 ± 1.38	439 ± 24	0.78 ± 0.39
			✓	68.71 ± 2.02	68.87 ± 1.67	370 ± 15	1.15 ± 0.38
Deep Features	EDNet	✓		67.64 ± 4.14	68.55 ± 3.86	332 ± 12	0.64 ± 0.33
			✓	69.78 ± 4.22	69.03 ± 3.72	312 ± 37	0.89 ± 0.45
Proposed	HAREDNet (w/o LOMO)	✓		72.69 ± 1.17	73.97 ± 1.31	434 ± 10	0.97 ± 0.38
			✓	75.12 ± 2.93	76.41 ± 3.32	375 ± 29	0.83 ± 0.55
	HAREDNet	✓		77.51 ± 4.68	77.25 ± 4.28	292 ± 11	0.54 ± 0.47
			✓	**80.58 ± 2.19**	**82.45 ± 2.93**	311 ± 14	**0.41 ± 0.26**

iSIFT and iGabor features is also noted for all selected dataset. While evaluating D1, a maximum average accuracy of 95.63% is achieved for CS and 99.26% for CV using the MSVM classifier. This classifier is trained in 242 min and predicted an input image with an average time of 0.45 s.

The comparison of different classifiers is shown in Table 2. D1 dataset is also evaluated by using local and deep features separately, as shown in Table 3. The CvQDA algorithm has significantly increased the recognition results for each experiment. LOMO features have also enhanced these results; as proposed, HAREDNet without LOMO features achieved average recognition accuracy of 83.87%, while with LOMO features, this accuracy is increased to 91.44%. Similarly, the proposed model without CvQDA has gained 86.87% average accuracy and 95.63% with CvQDA algorithm.

D2 dataset is evaluated using ACC, CRR, Pre, and Sen. Table 4 shows a comparison of different classifiers on D2, where FkNN has

Table 6
Recognition results of different classifiers on D3.

Classifier	Acc (%)	CRR (%)	Pre (%)	Sen (%)	TT (m)	PT (s)
MSVM	**97.48 ± 1.74**	**97.16 ± 1.71**	**96.37 ± 1.33**	**97.02 ± 1.82**	**328 ± 39**	**0.43 ± 0.24**
ESD	87.51 ± 3.94	88.16 ± 3.35	87.56 ± 3.38	86.84 ± 3.69	432 ± 49	0.92 ± 0.24
FkNN	82.82 ± 1.41	83.84 ± 1.69	83.42 ± 1.85	83.52 ± 1.89	443 ± 19	0.88 ± 0.31
WkNN	94.16 ± 1.95	93.27 ± 1.23	93.36 ± 1.03	93.02 ± 1.37	375 ± 39	0.57 ± 0.21
ES-kNN	90.89 ± 4.03	90.82 ± 4.63	90.05 ± 4.28	90.66 ± 4.47	511 ± 46	1.03 ± 0.23
QSVM	77.02 ± 3.11	77.22 ± 3.66	76.54 ± 4.15	78.08 ± 3.85	390 ± 21	0.89 ± 0.36
LDA	81.3 ± 4.27	80.62 ± 3.09	81.97 ± 3.36	80.07 ± 3.91	477 ± 29	0.73 ± 0.47
BTree	85.09 ± 1.55	84.99 ± 1.24	84.34 ± 1.62	86.69 ± 1.76	445 ± 39	0.64 ± 0.28

Table 7
Recognition results obtained through different features on D3.

	Feature Extractor	CvQDA No	CvQDA Yes	Acc (%)	CRR (%)	TT (s)	PT (s)
Local Features	SIFT	✓		73.31 ± 1.84	75.25 ± 2.11	324 ± 17	1.24 ± 0.45
			✓	76.48 ± 3.79	79.64 ± 2.67	319 ± 16	1.22 ± 0.22
	iSIFT	✓		79.14 ± 1.87	79.48 ± 1.36	368 ± 24	0.65 ± 0.26
			✓	82.77 ± 3.12	82.36 ± 3.48	459 ± 11	0.87 ± 0.46
	Gabor	✓		80.65 ± 3.53	83.02 ± 2.61	399 ± 37	2.14 ± 0.30
			✓	84.32 ± 2.75	86.41 ± 2.80	451 ± 48	1.39 ± 0.25
	iGabor	✓		85.67 ± 3.12	85.89 ± 2.73	494 ± 32	0.64 ± 0.26
			✓	86.93 ± 1.77	86.66 ± 1.76	377 ± 28	1.07 ± 0.42
	LOMO	✓		85.08 ± 3.93	85.14 ± 3.09	529 ± 24	0.95 ± 0.24
			✓	88.58 ± 4.02	88.83 ± 4.48	460 ± 45	0.74 ± 0.47
Deep Features	EDNet	✓		87.23 ± 3.22	87.21 ± 3.52	393 ± 16	0.85 ± 0.38
			✓	89.38 ± 2.06	90.78 ± 2.46	548 ± 27	1.21 ± 0.26
Proposed	HAREDNet (w/o LOMO)	✓		90.59 ± 1.05	90.06 ± 1.38	315 ± 16	0.92 ± 0.34
			✓	92.77 ± 2.97	92.83 ± 2.61	491 ± 33	0.72 ± 0.25
	HAREDNet	✓		95.44 ± 2.39	96.46 ± 2.08	**308 ± 44**	0.55 ± 0.38
			✓	**97.48 ± 1.74**	**97.16 ± 1.71**	328 ± 39	**0.43 ± 0.24**

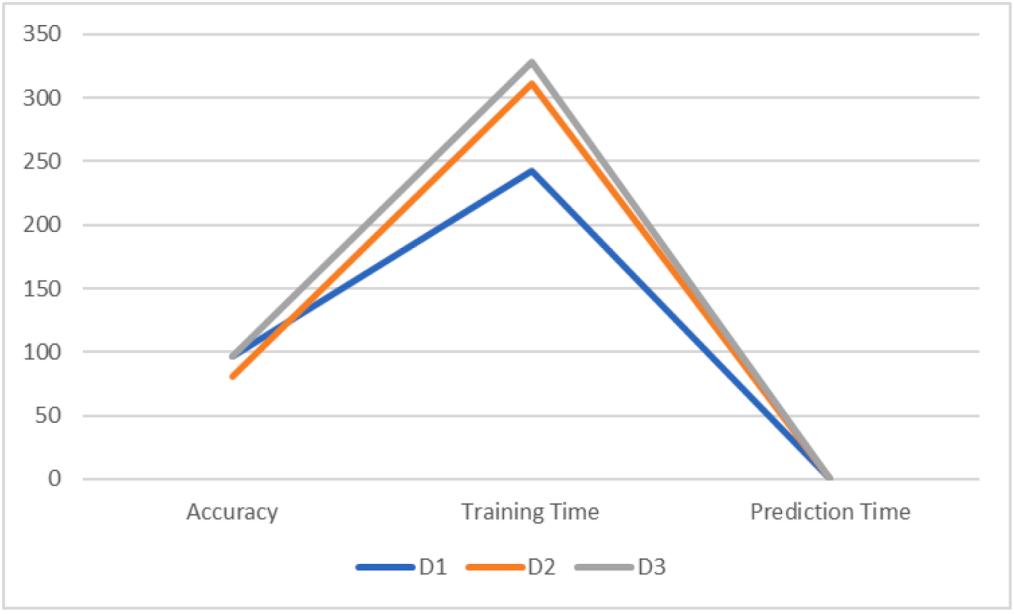

Fig. 6.. Comparison of accuracy, training time, and prediction time on D1, D2, and D3.

achieves maximum average recognition accuracy of 80.58%, CRR of 82.45%, Pre of 82.86%, Sen of 81.21% and TT of 311 min. FkNN predicts an input image an average of 0.41 s. The second highest accuracy is achieved on the ES-kNN classifier with 78.57%. The impact of LOMO features and the CvQDA algorithm is also evaluated and compared in Table 5. Without LOMO features, the proposed model obtained average recognition accuracy of 72.69%, and with LOMO features, this accuracy is increased to 77.51%. Similarly, the

Table 8
Impact of EDNet as compared to pre-trained CNN models.

Model	D1			D2			D3		
	Acc (%)	TT (m)	PT (s)	Acc (%)	TT (m)	PT (s)	Acc (%)	TT (m)	PT (s)
I	80.13	593.82	0.23	64.16	294.53	0.18	81.03	275.38	0.70
De	79.92	614.48	0.19	63.30	364.37	0.14	79.98	311.60	0.76
Da	78.49	491.59	0.27	61.82	248.51	0.23	76.33	297.92	0.64
EDNet	83.26	410.45	0.16	67.46	126.92	0.11	84.91	293.48	0.55

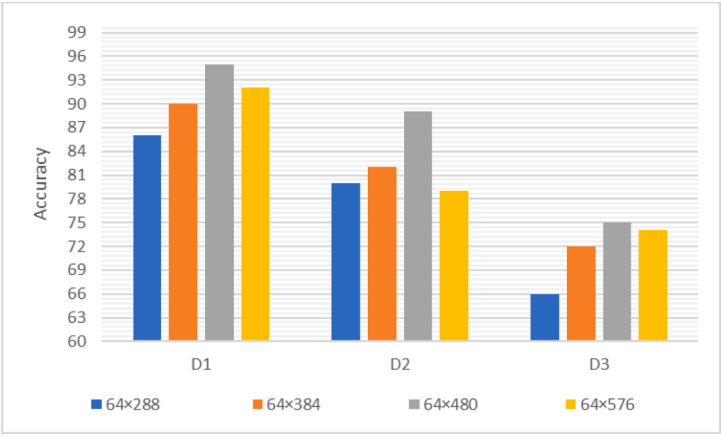

Fig. 7.. Comparison of adjusted input sizes for HAREDNet.

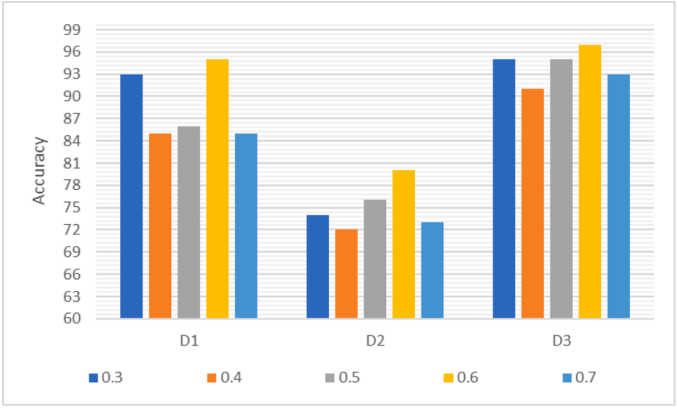

Fig. 8.. Comparison of weighted fusion for HAREDNet.

proposed model's accuracy without CvQDA remains 77.51%, while CvQDA increases this accuracy to 80.58%.

Evaluation of D3 is performed using the same performance measures used for D2. Table 6 presents a comparison of different classifiers on D3, where MSVM has achieves maximum average recognition accuracy of 97.48%, CRR of 97.16%, Pre of 96.37%, Sen of 97.02%, and TT of 328 min. MSVM predicts an input image in average of 0.43 s. The second highest accuracy is achieved on the WkNN classifier with 94.16%. The impact of LOMO features and the CvQDA algorithm is also evaluated and compared in Table 7. Without LOMO features, the proposed model obtained an average accuracy of 90.59%, and with LOMO features, this accuracy is increased to 95.44%. Similarly, the accuracy of the proposed model without CvQDA remains 95.44%, while CvQDA increases this accuracy to 97.48%. Fig. 6 shows an overall comparison of obtained accuracy, training time, and prediction time on all three datasets using the proposed model.

4.3. Ablation study

To understand the proposed model's efficiency, it is analyzed in terms of EDNet, parameter analysis, and different descriptors' roles. The analysis is carried out on all selected datasets by randomly selecting training and testing samples.

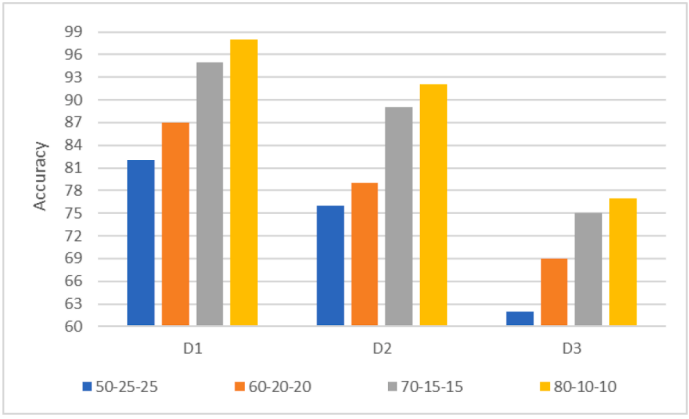

Fig. 9.. Comparison of data split ratio for HAREDNet.

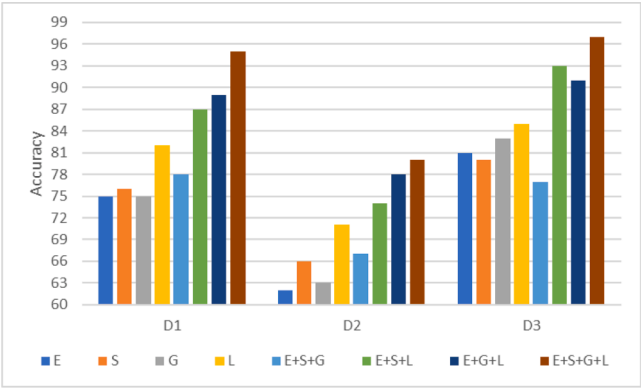

Fig. 10.. Comparison of fusing different descriptors (E: EDNet, S: iSIFT, G: iGabor, L: LOMO).

4.3.1. Role of EDNet

The proposed EDNet is compared with pre-trained models i.e., InceptionV3 (I), DenseNet201 (De), and DarkNet53 (Da) [22]. The impact is shown in Table 8. In terms of Acc, TT and PT prove that the proposed EDNet is well suited for HAR in an uncontrolled environment compared to pre-trained models. These results are achieved while utilizing selected CNN models as feature extractors and forwarded these features to several classifiers, where Q-SVM performed better. EDNet has achieved improved accuracy and reduced training and prediction time on all datasets.

4.3.2. Parameter analysis

Parameters of CNN architecture are analyzed for the reliability and validity of the proposed model. The input image of size $256 \times 256 \times 3$ is resized from 64×288 to 64×576 with the help of interpolation technique. The comparison of adjusted input sizes in terms of accuracy is presented in Fig. 7, where one random class is selected from all three datasets. It is noted that the size 64×480 has achieved the highest accuracy among other sizes. Further increase or decrease in this size is significantly reducing the performance. It should be noticed that the CNN architecture allows the kernel size of first Conv and DSC layer to be adjusted according to input data size.

The effect of weighted fusion is also analyzed, where the value of Υ is examined. During the experiments, the value of Υ is varied between 0.3 to 0.7. The impact of these values is shown in Fig. 8. These results show that the relevance of deep features has more impact on recognizing HAR than local features.

Lastly, the impact of splitting training, testing, and validation of data is noted, where the selected approach of $70 - 15 - 15$ is compared with $50 - 25 - 25$, $60 - 20 - 20$, and $80 - 10 - 10$. The noted impact is shown in Fig. 9, where the accuracy is increased when 80% of data is used for training purposes. But for a fair evaluation of the proposed model, a standard approach of $70 - 15 - 15$ is selected.

4.3.3. Role of different descriptors

The impact of different descriptors is analyzed, where deep features are fused with local features. This analysis is performed by using stand-alone features as well as employing different combinations. LOMO features have a significant impact on recognizing

Table 9
Comparison with previous techniques on D1.

Method	CS (%)	CV (%)
JOLO-GCN [23]	93.80	98.10
MS-G3D Net [24]	91.50	96.20
Multimodal graph convolutional subnetwork [25]	91.50	95.00
HAREDNet	**95.63**	**99.26**

Table 10
Comparison with previous techniques on D2 and D3.

Method	Accuracy	
	D2	D3
ResNext101+DA [26]	74.33	95.83
STCAN [27]	75.17	96.20
TSM-ResNet50 [28]	73.40	96.40
BraVe [29]	77.80	95.80
HAREDNet	**80.58**	**97.48**

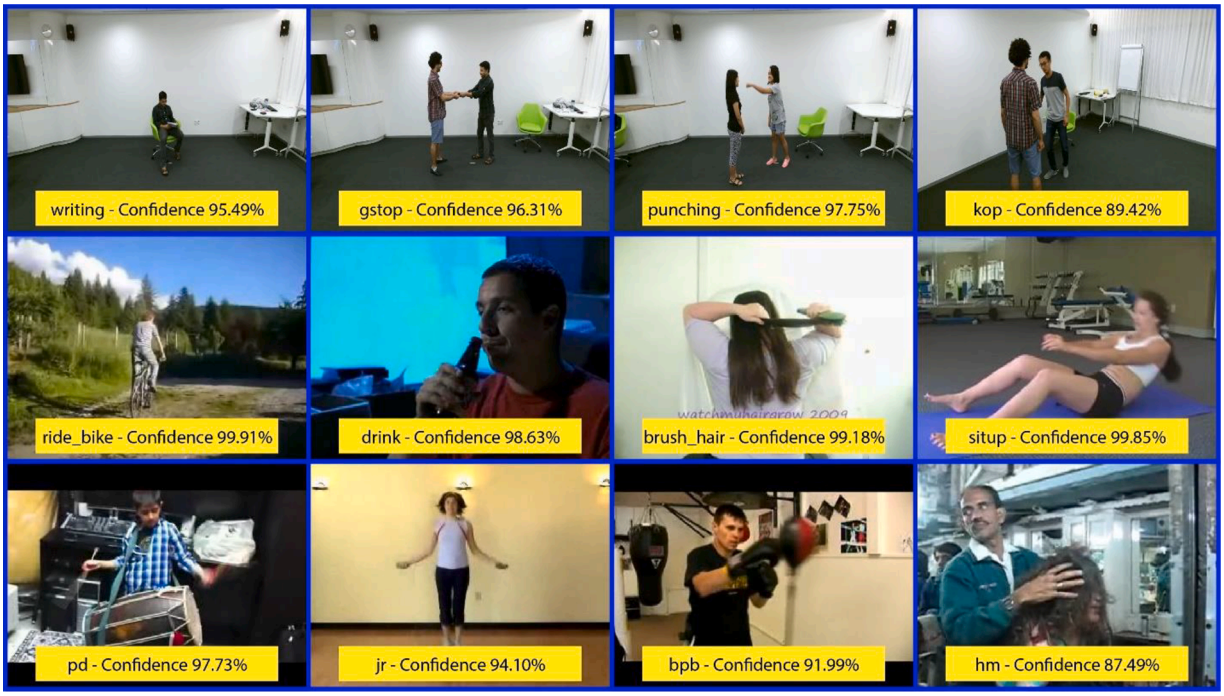

Fig. 11.. Correctly labeled data using HAREDNet.

human actions as compared to other local feature descriptors. When combined with iSIFT and iGabor, LOMO enhanced their efficiency by 13.26% and 8.34%, respectively. Maximum results are achieved by combining all deep and hand-crafted features. This impact is shown in Fig. 10.

4.3.4. Comparison with the state-of-the-art

Cai et al. [23] proposed a novel architecture called JOLO-GCN to estimate human actions utilizing pose skeletons and joint-centered low-level features combined into a CNN network. Joint-aligned-optical Flow Patches (JFP) were used to monitor joints for every joint's local motion to extract visual features as joint-centered pivotal. The technique was tested on D1 and achieved an accuracy of 93.80% for CS and 98.10% for CV.

Liu et al. [24] proposed a model for disentangling multi-scale graph convolutional layers and unifying spatio-temporal convolution layers as G3D. This increases the effectiveness of long-range modeling by aggregating schemes disentangling the relevance of nodes for nearest neighbors. THE proposed G3D technique was tested on D1 and achieved an accuracy of 91.50% for CS and 96.20% for CV.

In another technique, Yu et al. [25] proposed the usage of graph convolutional subnetwork for learning skeleton representations.

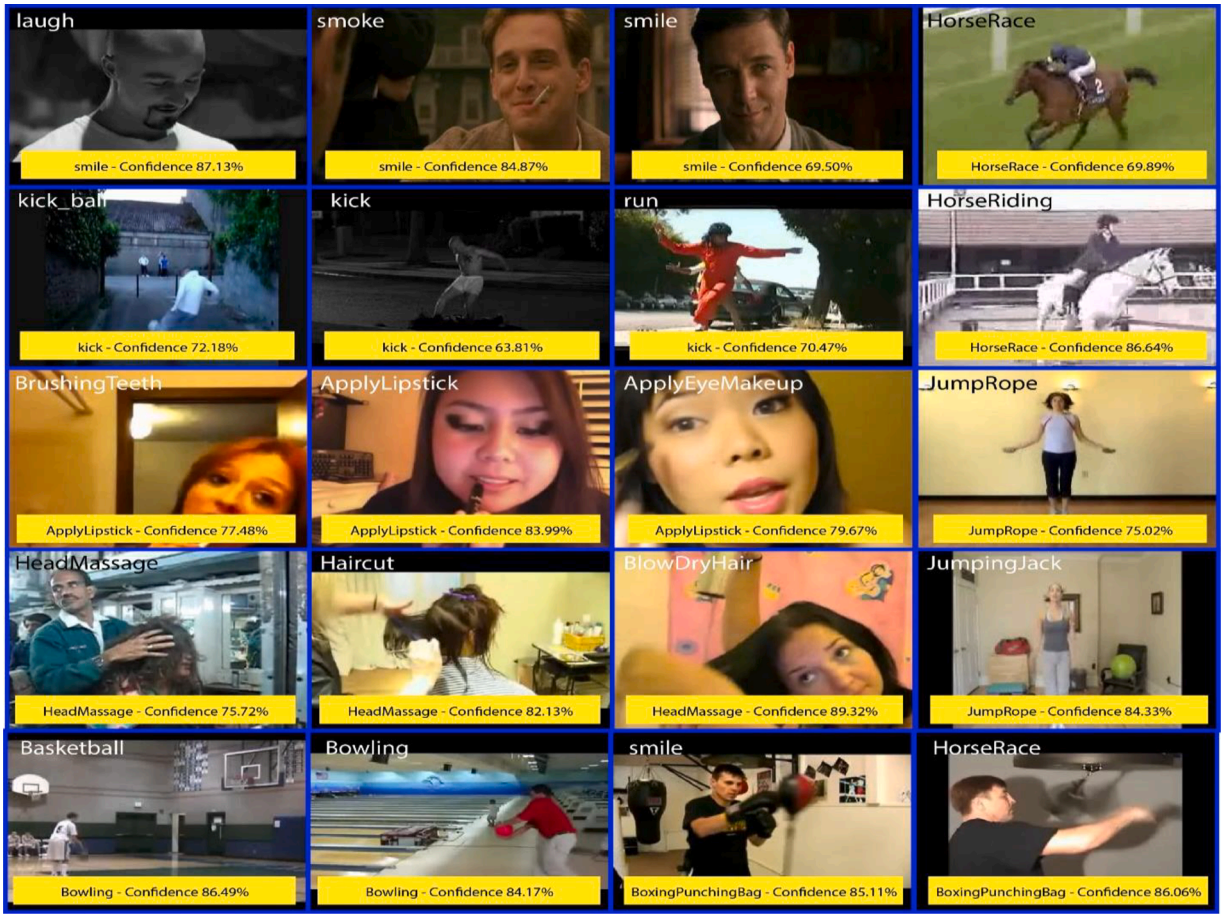

Fig. 12.. Wrong predictions using HAREDNet.

Spatial-temporal regions of interest are used from RGB videos to extract features of skeleton modalities. The model was evaluated on D1 dataset for achieving 91.50% and 95.00% accuracy for CS and CV respectively. Comparison of proposed model with state-of-the-art techniques is presented in Table 9.

Kim et al. [26] proposed an attention module to aim at human actions only by ignoring background and other non-action objects. Triplet loss was employed, which differentiated active and non-active features, while attention modules enhanced feature representation ability by utilizing the channel and spatial domains of action images. The proposed model was tested on the D2 dataset to achieve 95.83% accuracy and D3 to achieve 74.33% accuracy. Chen et al. [27] proposed Spatial-Temporal Channel-wise Attention Network (STCAN) to extract compelling features by recalibrating channel-wise features' responses. STCAN was based on a two-stream architecture with Channel-wise Attention Unit (CAU) to extract temporal and spatial features. CAU module was proved helpful for calculating dependencies among different channels to calculate and predict weight distribution of features. This technique was evaluated on D2 and D3 datasets and achieved 75.17% and 96.20% accuracies, respectively.

Liu et al. [28] proposed a computationally tractable technique to make a cluster of temporal dimensions and video frame activations based on similarity, which was used to aggregate frames into more miniature representations. This technique was validated on D2 and D3 and achieved a classification accuracy of 73.40% and 96.40%, respectively.

Recasens et al. [29] proposed a self-supervised architecture, which considered a narrow temporal window in one view while understood video content in other. The architecture learned generalization from a narrow view to the overall content of the video. It included different backbones, which enabled the usage of different modalities and augmentations in understanding broad view. The proposed architecture obtained 77.80% and 95.80% classification accuracy on D2 and D3 datasets. This comparison in tabular form is presented in Table 10.

From these results, the proposed HAREDNet has outperformed state-of-the-art techniques on all selected datasets. The inclusion of LOMO features and the CvQDA algorithm has improved the efficiency of the overall model. EDNet has also proved its novelty by achieving better results than pre-trained CNN models. The fusion of local and deep features has further enhanced the overall performance by achieving better results than previous techniques.

Fig. 11 shows some correctly labeled images, while Fig. 12 shows wrongly recognized images. It is visible that the wrong recognitions are due to intra-class similarity in a dataset, which has ultimately reduced the performance of the proposed model. The wrong

predictions are due to similar images in different classes. In the first row, first three images are like smile, but all these images belong to different classes. This can be improved by implementing distance functions.

5. Conclusion

This article aims to present a real-time HAR model, which can effectively and efficiently recognize human action in an uncontrolled environment. The hybrid technique proposed in this article recognizes human actions by performing four (4) steps, including a) deep feature extraction using EDNet; b) local feature extraction using improved iSIFT, iGabor and LOMO feature descriptors; c) distance learning approach to find the relevance among features using CvQDA algorithm; and d) weighted fusion strategy to select efficient and relevance portion of features. A comprehensive comparison of the proposed methodology in every aspect, from the purpose of usage to the integration properties, is also presented. Ablation analysis is also presented to review the proposed model. Experimental results show that the proposed model performs better than previous techniques on all datasets, i.e., NTU RGB+D, HMDB51, and UCF-101. The proposed algorithm, once trained, can recognize human actions with efficiency and effectiveness. One limitation of this work is that it has a high chance of wrong predictions due to intra-class similarity, as shown in Fig. 12. Another limitation is that it has higher chances of reduced recognition results if more than four subjects are in a single frame. In the future, a solution to these limitations can further increase the efficiency of this model. The proposed model can also be implemented and test on other recognition domains. Different distance functions can also be implemented to reduce the intra-class similarities.

Declaration of Competing Interest

The authors declare no conflict of interest.

Funding

This research has not received any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Shi L, Zhang Y, Cheng J, Lu H. Action recognition via pose-based graph convolutional networks with intermediate dense supervision. Pattern Recognit 2022; 121:108170.
[2] Guha R, Khan AH, Singh PK, Sarkar R, Bhattacharjee D. CGA: a new feature selection model for visual human action recognition. Neural Comput Appl 2021;33: 5267–86.
[3] Khan S, Khan MA, Alhaisoni M, Tariq U, Yong H-S, Armghan A, et al. Human action recognition: a paradigm of best deep learning features selection and serial based extended fusion. Sensors 2021;21:7941.
[4] Khan MA, Zhang Y-D, Khan SA, Attique M, Rehman A, Seo S. A resource conscious human action recognition framework using 26-layered deep convolutional neural network. Multimed. Tools Appl 2021;80:35827–49.
[5] Khan MA, Zhang Y-D, Allison M, Kadry S, Wang S-H, Saba T, et al. A fused heterogeneous deep neural network and robust feature selection framework for human actions recognition. Arab J Sci Eng 2021:1–16.
[6] Aly S, Sayed A. Human action recognition using bag of global and local Zernike moment features. Multimed Tools Appl 2019;78:24923–53.
[7] Du Y, Wang W, Wang L. Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1110–8.
[8] Gammulle H, Denman S, Sridharan S, Fookes C. Two stream lstm: a deep fusion framework for human action recognition. In: 2017 IEEE winter conference on applications of computer vision (WACV); 2017. p. 177–86.
[9] Sun L, Jia K, Chen K, Yeung D-Y, Shi BE, Savarese S. Lattice long short-term memory for human action recognition. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2147–56.
[10] Vishwakarma DK. A two-fold transformation model for human action recognition using decisive pose. Cogn Syst Res 2020;61:1–13.
[11] Gao Z, Xuan H-Z, Zhang H, Wan S, Choo K-KR. Adaptive fusion and category-level dictionary learning model for multiview human action recognition. IEEE IoT J 2020;6:9280–93.
[12] Mathe E, Maniatis A, Spyrou E, Mylonas P. A deep learning approach for human action recognition using skeletal information. GeNeDis 2018. Springer; 2020. p. 105–14.
[13] Chaudhary S, Murala S. Deep network for human action recognition using Weber motion. Neurocomputing 2019;367:207–16.
[14] N.L. Baisa, Z. Jiang, R. Vyas, B. Williams, H. Rahmani, P. Angelov, et al., "Hand-based person identification using global and part-aware deep feature representation learning," *arXiv preprint arXiv:2101.05260,*2021.
[15] Xu J. A deep learning approach to building an intelligent video surveillance system. Multimed Tools Appl 2021;80:5495–515.
[16] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vision 2004;60:91–110.
[17] Huang L-L, Shimizu A, Kobatake H. Classification-based face detection using Gabor filter features. In: *Sixth IEEE international conference on automatic face and gesture recognition, 2004*. Proceedings; 2004. p. 397–402.
[18] Land EH, McCann JJ. Lightness and retinex theory. Josa 1971;61:1–11.
[19] Shahroudy A, Liu J, Ng T-T, Wang G. Ntu rgb+ d: a large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 1010–9.
[20] Jhuang H, Garrote H, Poggio E, Serre T, Hmdb T. A large video database for human motion recognition. In: Proc. of IEEE international conference on computer vision; 2011. p. 6.
[21] K. Soomro, A.R. Zamir, and M. Shah, "UCF101: a dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402,*2012.
[22] J. Redmon, "Darknet: open source neural networks in c," ed, 2013.
[23] Cai J, Jiang N, Han X, Jia K, Lu J. JOLO-GCN: mining Joint-Centered Light-Weight Information for Skeleton-Based Action Recognition. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision; 2021. p. 2735–44.
[24] Liu Z, Zhang H, Chen Z, Wang Z, Ouyang W. Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2020. p. 143–52.
[25] B.X. Yu, Y. Liu, and K.C. Chan, "Skeleton focused human activity recognition in rgb video," *arXiv preprint arXiv:2004.13979,*2020.
[26] Kim DH, Anvarov F, Lee JM, Song BC. Metric-based attention feature learning for video action recognition. IEEE Access 2021;9:39218–28.

[27] Chen L, Liu Y, Man Y. Spatial-temporal channel-wise attention network for action recognition. Multimed Tools Appl 2021:1–20.
[28] X. Liu, S.L. Pintea, F.K. Nejadasl, O. Booij, and J.C. van Gemert, "No frame left behind: full video action recognition," *arXiv preprint* arXiv:2103.15395,2021.
[29] A. Recasens, P. Luc, J.-B. Alayrac, L. Wang, F. Strub, C. Tallec, et al., "Broaden your views for self-supervised video learning," *arXiv preprint* arXiv: 2103.16559,2021.