IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 March 2023, accepted 16 March 2023, date of publication 22 March 2023, date of current version 16 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3260403

RESEARCH ARTICLE

Depth Sensors-Based Action Recognition Using a Modified K-Ary Entropy Classifier

MOUAZMA BATOOL[1], SAUD S. ALOTAIBI[2], MOHAMMED HAMAD ALATIYYAH[3], KHALED ALNOWAISER[4], HANAN ALJUAID[5], AHMAD JALAL[1], AND JEONGMIN PARK[6]

[1]Department of Computer Science, Air University, Islamabad 44000, Pakistan
[2]Information Systems Department, Umm Al-Qura University, Makkah 24382, Saudi Arabia
[3]Department of Computer Science, College of Sciences and Humanities in Aflaj, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[4]Department of Computer Engineering, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[5]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[6]Department of Computer Engineering, Tech University of Korea, Siheung-si, Gyeonggi-do 15073, South Korea

Corresponding author: Jeongmin Park (jmpark@tukorea.ac.kr)

This work was supported by the MSIT (Ministry of Science and Information Communication Technology), South Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2018-0-01426) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). The authors acknowledge Princess Nourah bint Abdulrahman University Researchers supporting Project number (PNURSP2023R54), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT Surveillance system is acquiring an ample interest in the field of computer vision. Existing surveillance system usually relies on optical or wearable sensors for indoor and outdoor activities. These sensors give reasonable performance in a simulation environment. However, when used under realistic settings, they could cause a large number of false alarms. Moreover, in a real-world scenario, positioning a depth camera at too great a distance from the subject could compromise image quality and result in the loss of depth information. Furthermore, depth information in RGB images may be lost when converting a 3D image to a 2D image. Therefore, extensive surveillance system research is moving on fused sensors, which has greatly improved action recognition performance. By taking into account the concept of fused sensors, this paper proposed a novel idea of a modified K-Ary entropy classifier algorithm to map the arbitrary size of vectors to a fixed-size subtree pattern for graph classification and to solve complex feature selection and classification problems using RGB-D data. The main aim of this paper is to increase the space between the intra-substructure nodes of a tree through entropy accumulation. Hence, the likelihood of classifying the minority class as belonging to the majority class has been reduced. The working of the proposed model has been described as follows: First, the depth and RGB images from three benchmark datasets have been taken as the input for the model. Then, using 2.5D cloud point modeling and ridge extraction, full-body features, and point-based features have been retrieved. Finally, for the efficacy of the surveillance system, a modified K-Ary entropy accumulation classifier is optimized by the probability-based incremental learning (PBIL) algorithm has been used. In both qualitative and quantitative experimental results, the testing results have shown 95.05%, 95.56%, and 95.08% performance over SYSU-ACTION, PRECIS HAR, and Northwestern-UCLA (N-UCLA) datasets. The proposed system could apply to various real-world emerging applications like human target tracking, security-critical human event detection, perimeter security, internet security, public safety etc.

INDEX TERMS 2.5D cloud point, full body features, point-based features, probability-based incremental learning, RGB-D, K-Ary entropy accumulation.

I. INTRODUCTION

Human action recognition is a promising research field in the areas of mobile computing, context-aware computing, ambient assistive living, pervasive computing, and security

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

surveillance system [1]. Due to the intense requirements from the current technological progress and demands, it has recently attracted increasing attention in the latest technology development [2]. Sensor technologies have progressed significantly during the past decade and is actively explored in accurately recognizing activities, health status, and human behavior. Hence, these sensors may be just as capable of enhancing our quality of life as other common electronic gadgets like personal computers, smart phones, etc [3].

The aim of HAR is to collect and analyse data from various monitoring tools, such as sensors. The significant amount of time is needed to manually examine the visual data of different monitoring devices. Hence, in the modern era of computer vision, it is essential to use techniques to automate the visual semantics process of human activities [4]. The certain challenging factors such as, occlusion and unstable variation in shape and size of subject decrease the efficiency of action recognition task [5]. Moreover, human actions recorded with a various sensors, including depth sensors, smartphone sensors, RGB sensors, and others, to perform HAR are usually sensitive to changes in lighting and background clutter. Furthermore, it is impractical to use many cameras to achieve HAR [6]. Thus, with the recent advancement in vision-based technology, depth-based sensors, such as low-cost Kinect, have improved a lot in efficiency and quality. Also, the depth-based sensors have an advantage over RGB in that they can record data even in low-light conditions, and the data is also resistant to changes in color and texture [7]. Therefore in this paper depth-based sensors have been considered to achieve human activity recognition. There are numerous methods for determining view-invariant human behavior. One method is to use a multi-camera system to record human activity and then extract 3D features. Yet numerous 3D HAR modelling examples [8], [9], and [10] heightened the complexity of the recognition method. The second method involves applying the view transformation model (VTM) to the same views to transform features from different views [11]. VTM requires multi-view images in order to be generated, despite the fact that it has the advantage of multi-view recognition over multi-camera systems [12]. The third strategy is to combine the multi-view data obtained from several cameras using a multi-view fusion classification algorithm [13]. Unfortunately, the approach performs badly when views are drastically altered or self-occlusion occurs because it cannot be adequately represented [14]. Moreover, they have high computing costs or produce low-resolution bounding boxes, limiting their utility in situations where detail is required [15], [16], [17], [18], [19], [20], [21], [22]. Additionally, small spaces and 2D mapping approaches usually scale poorly to the human silhouette features extraction method [23].

To address the aforementioned issues, a 2.5D cloud point system is suggested in this paper, to create a 2D map using a static point cloud. Then 3D dynamic objects are supplemented on the map, resulting in a 2.5D map. Moreover, full body features and point-based body features enhance the efficiency of HAR. Furthermore, a novel K-Ary entropy accumulation optimized by probability-based incremental learning (PBIL) algorithm has been proposed to improve the recognition results of HAR.

In three datasets, we found that our method outperformed existing state-of-the-art methods in terms of recognition rates. This work's main contribution can be summarized as follows.

- The ground plane and the concept of voxel density are incorporated into an action recognition model for 2.5D to 2D mapping from 3D data input.
- We have looked into full body features and point-based body features for estimating numerous human silhouette areas, which is a critical challenge in a variety of real-world applications. The total accuracy of action identification was enhanced because of our precise feature extraction algorithms.
- By using the K-Ary entropy accumulation classifier, a novel technique has enhanced recognition accuracy. Additionally, a probability-based incremental learning classifier has produced optimal outcomes as an action recognition estimator.

THE LIST OF ACRONYMS AND SYMBOLS

Acronyms and Symbols	Abbreviation
RGB	Red Green Blue.
RGB-D	Red Green Blue Depth.
PBIL	Probability Based Incremental Learning.
VTM	View Transformation Model.
CNN	Convolutional Neural Network.
RNN	Recurrent Neural Network.
LSTM	Long Short-Term Memory.
2D	2 Dimensional.
2.5D	2.5 Dimensional.
3D	3 Dimensional.
BiLSTM	Bidirectional LSTM.
BGS	Background subtraction.
MRF	Markov Random Field.
KATH	K-Ary Tree Hashing Classifier.
KEC	K-Ary Entropy Classifier.
LOSO	Leave One Subject Out.
N-UCLA	Northwestern-UCLA.
HSV	Hue Saturation Value.
GHz	Giga Hertz.
GB	Giga Bytes.
RAM	Random Access Memory.
I	Original value of Image.
I_{min}	Minimum value of pixel.
I_{max}	Maximum value of pixels.
$omega_{target}$	Difference of upper and lower bound.
x	x-axis coordinate.
y	y-axis coordinate.
z	z-axis coordinate.

Acronyms and Symbols	Abbreviation
d	Depth value of image.
ε	Threshold.
c_0, c_1	Parameters of model.
u_0, v_0	Shifted parameters.
dis	Distortion function.
P	Point Cloud Data.
H	Hausdorff distance.
R_{data}	Ridge Data.
dt	Determine.
S	Human Skeleton.
dp	Depth pixel.
K	Number of nearest neighbors.
f	Frame.
dp	Depth Pixel.
R	Relation between nodes.
S	Set of nodes.
N	Number of nodes.
D	Degree of relation between nodes.
A	Image Region.
u, v,z	Vectors.
P	Point features.
s	Search rate.
l	Learning rate.
p	Population size.
C	Chromosomes.
v	Data Vector.
d	Euclidean Distance.
$x_1, x_2, \ldots, x_n$	vectors.
HW	Hamming weight.
a	Entropy Constant.
pdf	Probability density function.
k	Dimension vector space.
X_i	Subtree patterns of K-Ary.
pth	P^{th} number of Subtree Patterns.
w_{ip}, w_{jp}	Weight of Subtree Patterns.

The remainder of the paper is organized in the following manner. The related work on action recognition using fused sensors for surveillance systems is reviewed in Section II. The process for developing the system is presented in Section III. Experimental data are given and examined in Section IV to provide additional insight into the current action recognition dilemma. Finally, in Section V, a conclusion is given, as well as recommendations for further work.

II. RELATED WORKS

The scientific literature now offers several effective but still constrained gait recognition techniques for optical sensors like RGB and RGB-D images. Even though the field of human activity recognition has been researched for more than 40 years [51], there is still a need for improvement. The following ongoing gait recognition research challenges have been listed in [52]: Inadequate training datasets, unequal predictability of frames, cluttered background, the performance of similar action in several different manners known as intra-class variation, and different activities are much alike known as inter-class variation.

A. ACTION RECOGNITION WITH RGB SENSORS

Archana and Hareesh [24] propose a real-time surveillance system based on the RGB dataset. The model has been implemented on 3D CNN and ResNet18 without using LSTM based attention model. Their model comprises three convolutional layers connected to a pre-initialized layer with a hard-coded kernel, two subsampling layers, and a fully connected layer that assigns network outputs to their respective activity classes. The main drawback of this system is that it requires a large dataset to avoid the overfitting issue in the training model. Lee and Ahn [25] suggested a real-time model for the classification of human actions using a single RGB camera. The CNN has been used as a classifier which is further integrated into the NVIDIA JETSON XAVIER mobile robot embedded board respectively. The model has been implemented on two open-source libraries, including 3D-baseline and OpenPose. The model has achieved 70% accuracy on the NTU-RGBD dataset. The complete process has taken 15 frames per second olver the embedded platform in a real-time system. Crasto et al. [26] train the action recognition-based RGB stream, the output of the 3D CNN model using a linear combination of standard cross-entropy loss and feature-based loss that influence the motion and appearance, as well as minimize the feature-based loss. The author named the output-trained stream as Motion-Augmented RGB Stream. The model has given an average accuracy of 72.0% on UCF101-1, Kinetics400, Something-Somethingv1, and HMDB51-1 datasets respectively. The model had notable performance of distinct actions, while its performance degraded when the actions were similar to sitting and standing actions. Nasir et al. [27] proposed a machine-learning technique to classify video data. The videos have been first pre-processed by extracting the segment of interests. Later on, feature descriptor mining was done using four different features including 3D Cartesian-plane Features, Geodesic Distance, n-way Point Trajectory Generation, and Joints MOCAP. Finally, a neuro-fuzzy classifier has been used to classify the data into different actions. The proposed model has been evaluated on Hollywood2 and HMDB-51 and has achieved an accuracy of 91.99% and 82.55% respectively. Jalal et al. [28] proposed an event detection model. The author has designed a pseudo-2D stick model based on extracting of full-body human silhouette features, followed by optimization and hierarchical classification. The sine, energy, and 3D Cartesian gradient features have been used for the feature extraction mechanism. The ray optimization and K-Ary tree hashing classifier have given an optimal performance of 90.48% over the UCF50 dataset.

B. ACTION RECOGNITION WITH DEPTH SENSORS

Popescu et al. [29] proposed human activity recognition model based on depth data as input to the system. The channel, temporal, and context information of the RGB-D data has been captured using a temporal fusion mechanism to mutually combined the input data. Finally, CNN has been applied to the resultant data to get the data's maximum likelihood score and classify the human activities accordingly. The system has achieved an accuracy of 94.38% on the PRECIS HAR benchmark dataset. Ke et al. [30] proposed temporal and spatial structural information for the feature extraction mechanism on depth data. The extracted features have been then fed into a multitasking convolutional neural network to learn the action recognition. The proposed model has been extensively tested on the Northwestern-UCLA dataset and has achieved an accuracy of 86.82%. Wang et al. [31] propose a student-teacher learning model based on a one-layer bidirectional LSTM (BiLSTM) to predict activity at the early stage of action. The BiLSTM model has a forward and backward layer and receives information from history to obtain latent features. The author has trained the model on RGB-D datasets of NTU RGB-D, SYSU-ACTION, and UCF-101 datasets and has achieved an accuracy of 60.97, 75.35, and 89.64 percent. The model has achieved progress by minimizing the global distribution of knowledge between student-teacher models. The main limitation of this model include that this model has gotten slower due to spatial-temporal features and therefore it has taken a large computation time for the training data also degraded the performance of the overall model. Zhang et al. [32] proposed an adaptive neural network built on a convolutional neural recurrent neural network (RNN), network (CNN), and long short-term memory (LSTM). This model worked by learning an adaptive technique in each network, followed by the prediction of important observation viewpoints, and performing the transformation for the classification of human activity. Moreover, the networks were fused to eliminate the overfitting problem in the training data.The proposed model has trained over the Northwester UCLA dataset and SYSU-ACTION datasets and has achieved an accuracy of 85.1% and 86.6% respectively. Although the deep fused neural network approach achieved better performance, the main drawback of this method is that it has just used the last layer's calculation is lost, along with a great deal of relevant data that the middle layer collected. Hence, it degraded the performance of the overall modal. Khalid et al. [15] proposed a semantic recognition system based on RGB-D images. The system has been composed of filtration, feature extraction, feature selection, and classification. The bilateral filtering has been used as a pre-processing mechanism on RGB-D datasets. Secondly, five feature extraction technique has been applied to the filtered data including Euclidean Distance Transform, Gaussian Mixture Model, Conditional Random Field, Fidual point, and 3D cloud point. Finally, Fisher's Linear Discriminant Analysis along with the K-ary tree hashing classifier has been applied for human action recognition. The proposed model has been validated over SYSU-Action dataset with an accuracy of 93.5%.

III. PROPOSED MODEL

The system's operation is detailed in this section. Background subtraction, feature extraction, feature selection (optimization), and classification are all parts of the system's operation. Each of the subsections listed above has been detailed in detail below. Fig. 1 depicts the overall system's workflow. The output of each area is depicted in the other figures. The data pre-processing is to obtain realistic silhouettes of human posture. The feature selection and feature extraction approaches minimize the dimensionality of feature space by removing irrelevant features from the extracted silhouette. Finally, classification has been done to sort data into groups based on similarities in their features, as shown in Fig. 1. The Fig. 1 depicts the overall system's workflow. The depth (RGB-D) images have been used as input to the pre-processing stage. In pre-processing step, silhouette extraction has been obtained using substitution and scaling operation. The resultant silhouette has been then used to extract full body features (ridge features, Markov random field (MRF)), and 2.5D cloud point features. The 2.5D cloud point features have been then used to extract point based features (spatial-temporal features, angular geometric features, and orientation based features). Later on, full body features and point-based body features have been taken as input to probability-based incremental learning. The optimized features have been finally fed as input to the novel K-Ary Entropy Accumulation classifier.

A. SILHOUETTE EXTRACTION

Background subtraction (BGS) is a critical step in a surveillance system. Many BGS methods have previously been provided, such as temporal medians of previous n frames [33], statistical approaches [34], self-organizing maps [35], [36], [37], and numerous features-based methods [38], [39], [40]. These BGS systems, however, have certain fundamental limitations because they used color spaces based on human perception (i.e., visible light), such as RGB, HSV, and YUV, where Y and UV denote brightness and chrominance, respectively [41], [42], [43], [44]. In general, those strategies are ineffective in color camouflage settings and are very sensitive to changes in lighting [45].

Here, preprocessing step has been accumulated through substitution operation [46] and scaling operation. The depth Kinect camera, is usually unable to acquire complete information on the depth pixels either due to occlusion or light-defusing obstacles [47]. Therefore, it has been interpolated using substitution operation. For each pixel in the depth image, the left and right neighboring pixels of the missing depth pixel has been searched. The missing pixel has been replaced with the larger of the two valid neighboring depth pixels [48].

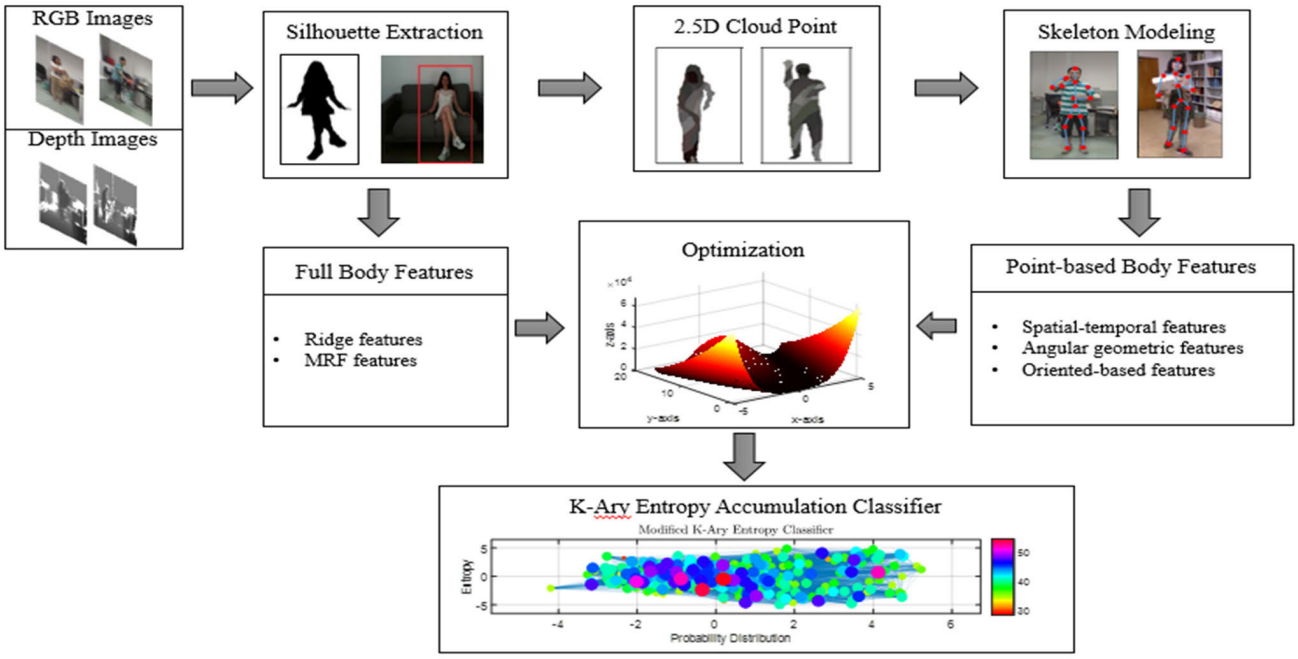

FIGURE 1. Block diagram of the proposed system based on RGB-D data for action recognition.

In the scaling operation as described in equation (1), the depth image has been linearly scaled within the range of 0 to 255;

$$I' = \frac{I - I_{min}}{I_{max} - I_{min}} omega_{target} + I'_{min} \tag{1}$$

where I refers to the original value and I' are the rescaled values of the depth image. The I_{min} and I_{max} depicts minimum and maximum values of pixels before scaling. The I_{min} depicts the lower bound of the depth image which has been set to 0. $omega_{target}$ is the difference between the upper and lower bound which is set to 25. Lastly, depth silhouette has been mapped to RGB images using an affine transformation to obtain silhouettes from RGB frames. The final results of pre-processing have been depicted in Fig. 2.

FIGURE 2. Silhouette extraction results of drink tea and sit-down activities of the PRECIS HAR dataset.

B. 2.5D CLOUD POINT MODELING

The streamlined depiction of a 3D surface exists in 2.5D data. The RGB image pixels (x,y) of a point on the body surface are separated by a depth value, called d(x,y), in 2.5D data [49]. As a result, Kinect 2.5D is a good compromise between depth and RGB images. These elements can be combined to define the geometry of a single object or the entire scene. The x, y, and z coordinates of each point in the point cloud indicate where the point is physically located in 3D [50].

In this paper, a 2.5D point cloud model for a specific set of data was created using 2D data from the depth image and 3D data from the RGB image by computing all the 3D points from the measurements (x, y, d) in the depth image.

The depth image has been used to derive the first gait silhouettes. Then, 3D point cloud data were calculated using the gait silhouette and RGB image using equation (2).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{c_1 d + c_0} dis^{-1} \left(K^{-1} \begin{bmatrix} x + u_0 \\ y + v_0 \\ 1 \end{bmatrix}, k \right) \tag{2}$$

All of the gait point cloud data was normalized to 3D space before being used to build the 3D point cloud gait model for a specific viewpoint. Only one side surface area of the human body, known as a 2.5D point cloud model, is included in the gait point cloud data as shown in Fig. 3.

Given the size of the resulting point cloud data, Hausdorff distance [51] was used to further simplify it. The association between point cloud data P and its K nearest neighbors has been determined using the bounding box $\{K_1^P, K_2^P\}$ *and* $\left\{K_1^Q, K_2^Q\right\}$.

$$H = max_{i=1,2} min_{i=1,2} \left(\frac{\left\| K_i^P - K_j^Q \right\|}{\left\| K_i^P \right\| + \left\| K_j^Q \right\|} \right) \tag{3}$$

For P, the Hausdorff distance is given by $H^p = \max(H^Q)$, $Q = 1, 2, \ldots k$. In order to exclude the point cloud data that is less significant, a threshold ε has been chosen by computing the Hausdorff distance of each point inside the bounding box. The effectiveness of the simplification and the computing cost of the method are directly impacted by the threshold selection. A larger ε lowered the computing cost but had a lower degree of simplification, whereas a smaller ε had the reverse effect [52].

Before simplification, there are 25,862 point clouds in the output of the raw gait point cloud data. However, following simplification, the results can be seen in point cloud data with 13,286, 8,392, 6,381, 4,592, and 2,392 points, respectively [53]. The computations took 518, 432, 327, 273, and 228 ms to complete, respectively. We determined $\varepsilon = 10^{-4}$ with a mean computation time and sufficient simplification based on the experiment results [54]. The final 2.5D cloud pint findings are shown in Fig. 3.

FIGURE 3. The results of 2.5D modeling over the PRECIS HAR dataset.

C. SKELETON MODELING

Identifying key points of the human body have been initialized with the torso point. The torso point is the center point of the human body, and lead the main role in the outer shape estimation of human body pixels S_{px} [55]. The torso point has been calculated by taking the frame difference of video frames that have been formulated in equation (4).

$$S^f_{tp} = S^{f-1}_{tp} + \Delta S^{f-1}_{tp} \tag{4}$$

where, S^f_{tp} depicts the location of torso points tp on the human silhouette in video per frame f. Second, the human knee point has been calculated by taking the leg's middle point which is the center point between the hip and foot points [56]. The human knee point has been depicted in equation (5).

$$S^f_{sk} = \left(S^f_{sf} + \Delta S^f_{sh}\right)/2 \tag{5}$$

where, $S^f_{sk}, S^f_{sf}, \Delta S^f_{sh}$ depicts human knee, foot, and hip points respectively. Third, the elbow point in a human silhouette has been calculated by taking the center of shoulder and hand points [57]. The elbow point on a human's arm has been depicted in equation (6).

$$S^f_{se} = \left(S^f_{shn} + \Delta S^f_{ssd}\right)/2 \tag{6}$$

where, S^f_{se}, S^f_{shn}, ΔS^f_{ssd} depicts human elbow, hand, and shoulder points respectively. The fifteen key point's detection of the human body has been depicted in Fig. 4 and has been completely elaborated in Algorithm 1.

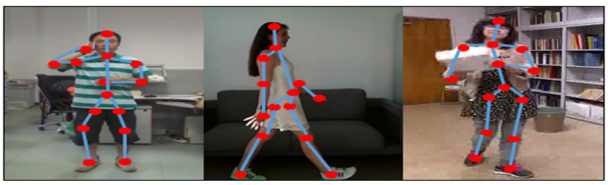

FIGURE 4. Examples of silhouette extraction process output over the SYSUACTION dataset.

Algorithm 1 Key Body Points of Human Silhouette

Input: H_sil: human silhouette, H_sh; human shape,, H = height,, W = width, L = left, R = right

Output: 15 key points detection: I_hd = head, I_nk = neck, I_shr1, I_shr2 = shoulders, I_elb1, I_elb2 = elbows, I_hnd1, I_hnd2 = hands, I_md = mid, I_hip1, I_hip2 = hips, I_kne1, I_kne2 = knees, and I_fet1, I_fet2 = feet.

do

1. I_hd = Get_head_point(search(H_sil))
2. I_nk = Head_end_point(I_hd)
3. I_md = mid $(H, W)/2$
4. $[I\_hip1, I\_hip2]$ = search($[I\_md]$ && $[L, R]$)
5. $[I\_fet1, I\_fet2]$ = Get_bottom_point(H_sh)
6. $[I\_kne1, I\_kne2]$ = mid($[I\_md]$, $[I\_fet1, I\_fet2]$)
7. $[I\_shr1, I\_shr2]$ = search($[I\_hd, I\_nk]$ && $[L, R]$)
8. $[I\_hnd1, I\_hnd2]$ = search($[I\_nk]$&& $[L, R]$)
9. $[I\_elb1, I\_elb2]$ = search($[I\_hnd1, I\_hnd2]$ && $[I\_shr1, I\_shr2]$)

end

While(largest region of H_sil found)

return15 key body points: I_hd, I_nk, I_md, $I\_hip1, I\_hip2, I\_fet1, I\_fet2, I\_kne1, I\_kne2$, $I\_shr1, I\_shr2, I\_hnd1, I\_hnd2, I\_elb1, I\_elb2$

D. FEATURES EXTRACTION

In this section, key attributes from full-body features and point-based features have been figured out from the extracted silhouette. The full body features such as ridge and Markov random field (MRF) have been used to robustly and efficiently analyze the key features of full body silhouette due to their selective representation of the body skeleton [58]. While, point-based features, spatial-temporal features, angular geometric features, and oriented-based features have been formulated [59]. The full body features and point-based features then later fed into population-based incremental learning (PBIL) and self-annotated K-Ary entropy classifier. The proposed model has given significant performance over the existing state-of-the-art models.

1) RIDGE

The features of ridge body components are ridge data for feature extraction and binary edge extraction. We have employed depth silhouettes to extract features from binary edges during binary edge extraction. These edges have undergone the distance transform processing to produce distance maps. In spite of the fact that these maps have been calculated to discover local maximal that offer one or more ridge data inside of binary edges for ridge data production. [61].

In order to quantify the local statistical values of the intensities of depth silhouettes' nearest neighbors, window searching has been employed to extract the binary edge data around those objects. Hence, an enclosed body structure and a sufficient edge connection are produced. The binary edge extraction has been depicted as;

$$B_{edge}(dt) = \left\{dp_c \in dt | \exists dp_i, \left|dp_i - dp_c\right| > \delta_E\right\}, \quad dp_i \in \left\{dp_{c-1}, dp_{c+1}, dp_{c-w}, dp_{c+w}\right\} \tag{7}$$

where the center depth pixel dp_c has been evaluated for intensity by comparing it to its corresponding adjacent pixels dp_i. The distance transform, which produces distance maps, also processes binary edges further.

Second, distance maps have been employed in the production of ridge data $R_{data}(dt)$ to calculate the local maximum of the related edges and provide ridge data, a chain of pixels [62]. Such ridge data have been surrounded by binary edges that mimic the human body's skeleton.

$$R_{data}(dt) = \left\{ dp_c \in dt \left| \frac{\sum_{j=1}^{n} D_M(dp_i)}{(n)(D_M(dp_c))} < \delta_R \right. \right\} \tag{8}$$

where D_M is the distance map values, which compare the values of the center pixels to those of the surrounding pixels. The diagrammatic form of the distance map-based binary edge silhouettes and ridge data is shown in Fig. 5. Such ridge data might reflect the skeleton's position and remove the acoustically noisy edge data.

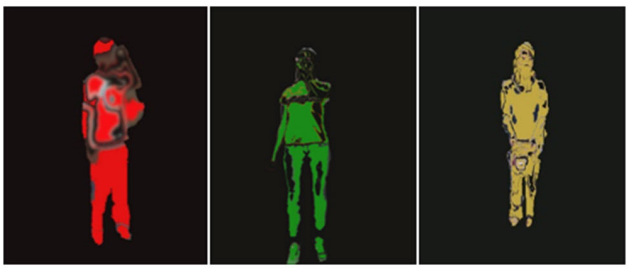

FIGURE 5. Examples of ridge extraction results over the Northwestern-UCLA dataset.

2) MARKOV RANDOM FIELD

The pixels with the same color are usually classified as a single region A_j, belongs to the same class, even if they are not connected. Therefore to maintain consistency the over-segmented regions have been merged into the meaningful region using Markov random field (MRF) [63] as shown in Fig. 6. MRF labels the connected components for establishing the probabilistic distribution of interacting features that has been depicted as;

$$R = (S, N, D) \tag{9}$$

where R depicts the relational structure of a set of nodes represented as S, neighborhood nodes are represented as N, and D depicts the degree of relationship. In this paper, D =3 has been depicted in Algorithm 2.

Algorithm 2 MRF Extraction of Human Silhouette

Input: A = 2.5D cloud points, Z = {natural numbers}
Output: MRF features detection
do

1. Unary features = single region
2. Region label = $l(A_j) \in Z$
3. Region size $\alpha\left(A_j\right) = \left|A_j\right|$, pixels in A_j
4. Color = H, S, V components of particular region
5. Centroid = M, median point of region
6. Border pixel set: $\varphi(A_j)$ {8 adjacently connected pixels correspond to contour of particular region}
7. Binary features: two adjacent regions
   - List of regions $L(A_j)$
   - A_j is adjacent to A_k
   - Border ratio of A_j adjacent to A_k
8. Tertiary features: adjacently connected three regions
   - $\tau\left(A_j\right) = \left\{ \begin{array}{cc} 1 & A_i \in L\left(L\left(A_j\right)\right), j \neq i \\ 0 & otherwise \end{array} \right\}$

end
While (MRF features extraction)
return MRF features

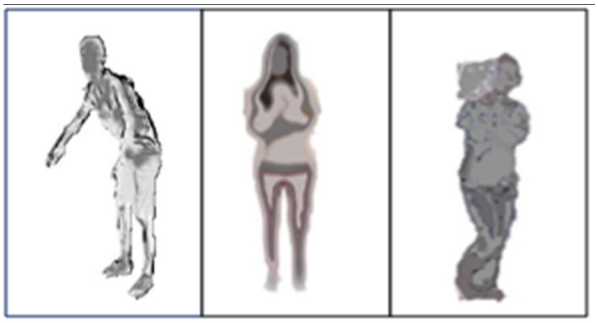

FIGURE 6. MRF results over the benchmark dataset of SYSU-ACTION dataset.

3) ORIENTED-BASED FEATURES

The movement of hands always formed a certain angle against the body [64]. The symmetry principle has been kept in mind while measuring the angle of hands against the upper, lower, and middle body points. The upper body key points include head I_hd, neck I_nk, and shoulders $[I\_shr1, I\_shr2]$. The center body points include mid-body points I_md, and hips points $[I\_hip1, I\_hip2]$.

The angle formation of hands against the lower body points includes hips [I_hip1, I_hip2], knee [I_kne1, I_kne2], and feet [I_fet1, I_fet2]. These angles have shown as fulfilling the decision criteria for the particular action at the specified time within every fifteen adjacent consecutive frames, the time interval is 0.5 seconds. The six activities have been selected that form an angle against the upper half of the body includes drinking, eating, making a phone call, taking off the jacket, putting on the jacket, and wearing contact lenses. The angle formation of hands against the center body point has been measured by using a remote, entering the room, exiting the room, getting up, sitting down, standing up, writing on the whiteboard, stirring, relaxing on the couch, and talking on the couch. The angle formation of hands against the lower body points that include hips [I_hip1, I_hip2], knee [I_kne1, I_kne2], and feet [I_fet1, I_fet2] has formed the most important element of determining the two crucial activities i.e., mop the floor and go to bed; within the selected three benchmark datasets. In Fig. 7, the angle detection procedure [65] is shown. Moreover, the formation of the angle A_1 in coordinates of hands (w_1, w_2) against the upper, lower, and middle body points (z_1, z_2) at time t has been expressed as;

$$A_1(t) = tan \left| \frac{w_1 - z_1}{w_2 - z_2} \right| \tag{10}$$

where, A_t depicts the angle formation of hands against the upper body point.

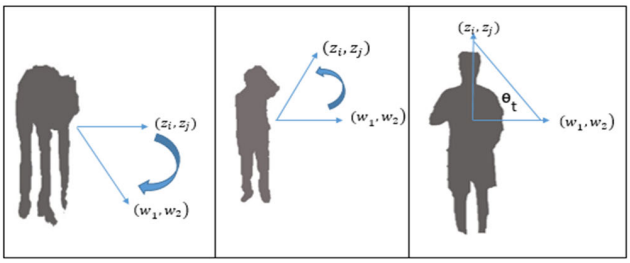

FIGURE 7. The results of angle orientation of the body over Northwestern-UCLA dataset.

4) ANGULAR-GEOMETRIC FEATURES

The angular-geometric features measure changes in angular values of key points within the consecutive frames [66]. To extract angular-geometric features in this paper, seven extreme body points have been selected that include: the head, shoulders, arms, and feet. Then, three inter-silhouette geometric shapes (pentagon, quadrilateral, and triangle) have been created by connecting these extreme points. Fig. 8 shows the formation of different geometric shapes by connecting the key points of human silhouette. The inverse cosine of each body has been measured after the development of geometric shapes, and it has been depicted as;

$$\theta_t = cos^{-1} \frac{x.y}{|x|\,|y|} \tag{11}$$

where, x and y vectors have been used to measure the shape area of inter-silhouette triangles. The area of the inter-silhouette triangle has been calculated as;

$$G_t = \sqrt{P(P-u)(P-v)(P-z)} \tag{12}$$

where, u, v, and z vectors have been used to measure the area of the inter-silhouette triangles. The area of inter-silhouette triangle has been calculated as;

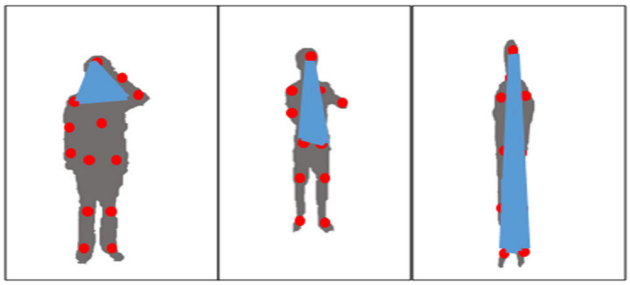

FIGURE 8. The results of triangle shapes results on the silhouette over SYSUACTION dataset.

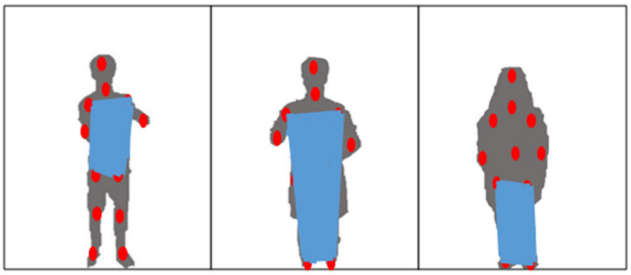

FIGURE 9. The results of quadrangular shapes of the silhouette over PRECIS HAR dataset.

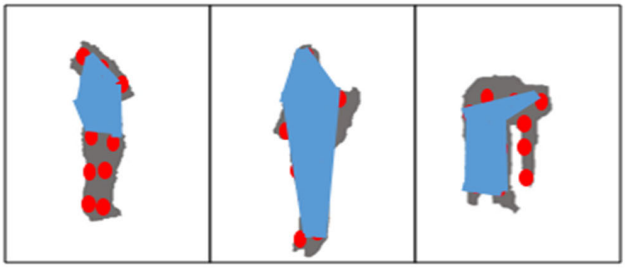

FIGURE 10. The pentagon detection results of the silhouette body over Northwestern-UCLA dataset.

5) CURVE POINTS DETECTION FEATURES

The 8 Freeman chain code algorithm has been used to measure the curve points along the silhouette and determine the change in intensity along the body's curve points [67]. The curve points along the boundary of the silhouette has been depicted as;

$$P_n = \{P_0, P_1, \ldots, P_n\} \tag{13}$$

The P_0 has been taken as the initial point of the features and move in a clockwise direction along the boundary until there is a change in direction denoted by P_1. The curve points along the boundary has been calculated as a feature f (see Fig. 11). In this way, all the features of the human silhouette has been calculated.

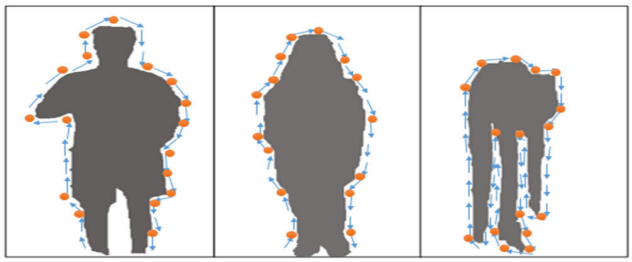

FIGURE 11. The curve point's detection over PRECIS HAR dataset. The blue arrow depicts boundary and orange dots depicts features f.

E. POPULATION-BASED INCREMENTAL LEARNING (PBIIL) OPTIMIZATION

The PBIL algorithm is a stochastic guided search method that gets its direction of the next solution from the prior best solutions. Three parameters of the PBIL that have been used include search rate (s), learning rate (l), and population size (p) [68]. The PBIL gave best performance and ends automatically as the process converges on a single solution unlike other stochastic optimization algorithms. The parameters are represented as a binary chromosome with b bits of total length. Each variable is encoded in binary form, and any earlier arguments are concatenated to create a single chromosome. A population of chromosomes' bit generation is biased using a prototype vector (P). For each bit location, there are b elements in the prototype vector. The prototype vector stores the likelihood that the associated bit is a 1 at each position [69]. In order to generate unbiased bits, each location is initially set to 0.5. The prototype vector is used to bias the production of bits and create a population of potential solutions. The bits are chosen for each chromosome in the population by producing a uniformly distributed random number for each bit in the range [0, 1]. If the random number is lower than the matching prototype vector element, the chromosomal bit is set to one; if not, it is set to zero. The best chromosomes are then determined after all of them have been evaluated by the objective function. Then, to incorporate the directionality of the best chromosome, the following equations have been applied to the prototype vector. The PBIL has been elaborated as;

$$P_{n+1} = ((1-l)\,P_n + l.C_B)\,(1-f) + \frac{f}{2}(1) \tag{14}$$

$$f = \frac{2sl}{1-2s(1-l)} \tag{15}$$

where l is the learning rate and s is the searching rate. C_B is the best chromosome and comprises a pattern of ones and zeros.

The PBIL optimization use static approach and tweak its parameters iteratively to minimize a given function to its local minimum. The probability vector is initialized to 0.5 and is updated gradually until the solution converge towards 0 as shown in Fig 12 a, b, and c.

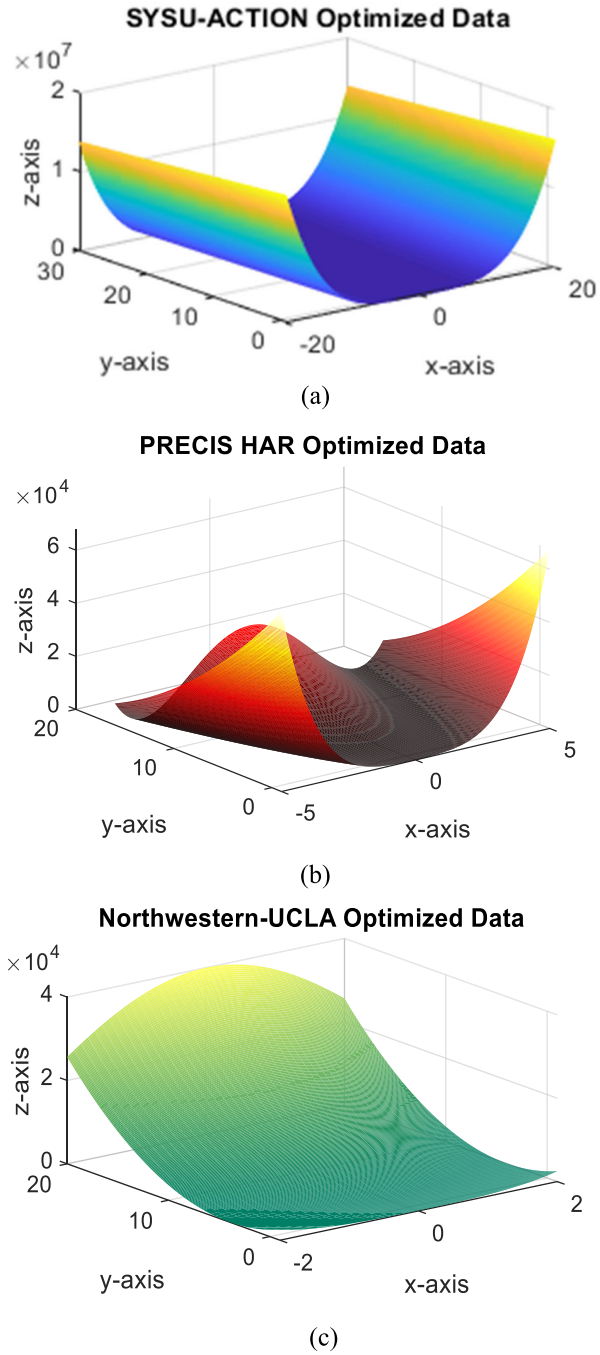

FIGURE 12. The PBIL optimized data results over (a) SYSU-ACTION (b) PRECIS HAR and (c) Northwestern-UCLA datasets.

F. K-ARY ENTROPY CLASSIFIER

The existing tree-based classification algorithms usually rely on a thorough listing of substructure patterns, where the

number of substructures grows rapidly with respect to the size of the vector set. Recently, the Wu et al. [70] hashing tree classifier (KATH) has achieved an optimal performance in terms of efficiency, accuracy, and handling classification of large-scale graphs. This algorithm is further used for classification of human activity recognition in [15] and [28]. The KATH has achieved an optimal performance in both graph classification and classification of human activity recognition. However, the computational efficiency of KATH is not able to obtain competitive accuracy to ensure the fairness of nodes in a real-time environment. We propose a unique K-Ary entropy classifier (KEC) that classify the similar features into same subtree patterns clusters in terms of child nodes of the KEC tree. The optimized data has been given as input to the KEC. The overall functionality of dividing nodes into tree and subtree patterns is similar to be created in K-Ary tree [70]. Recall that in K-Ary tree algorithm [70], traversal table has been constructed to assign indexes to the nodes and MinHash function has been used to fingerprint subtree patterns. For practical applications, where the data is highly dimensional, this traversal table construction along with MinHash function is often a bottleneck. Hence, we have used one level entropy based hashing that enables partitioning a very large set of features into many much-smaller, uniformly distributed subtree patterns, based on the high correlation among the features to a similar hash patterns. This minimizes the classifier searches to the relevant subtree patterns to which the particular belongs to, and therefore significantly shortens the classification process. Moreover, existing K-Ary classifier [70] has used a naive approach to select the nearest nodes. The naive approach is not capable of producing the exact results and may predict the wrong classes even if the probability of belongingness of an object to a certain class is zero. However, in our approach Euclidean distance has been calculated on the integer vector array and then hamming distance has given more robust results than the naïve approach in the existing K-Ary classifier. To this end, a modified KEC has been proposed that boosts the performance of the classifier faster than existing K-Ary graph classifier.

TABLE 1. Example of data scaling using normalization.

Input	0.0463	0.1783	0.2927	0.4035	0.5023	1.0460
Scaled Data	0.0	0.1	0.3	0.4	0.7	0.8

1) DATA SCALING

The optimized data has been given as an input to KEC that further need to be scaled to an appropriate level to boost the performance of modified KEC algorithm. For this purpose, the input data vector v_i^n has been partitioned into 2^n evenly sized vector array. The higher values have been assigned to higher range of vector data and vice versa as shown in Table 1.

The equation (16) efficiently scale the data in the range of $[0, 2^n - 1]$.

$$v_i^n(x_i) = \left\lceil \frac{x_i - min(V_i)}{max(V_i) - max(V_i)} .2^n \right\rceil - 1 \tag{16}$$

where, min and max are the minimum and maximum vector and x_i depicts the current vector of the input data.

Now, the input can be efficiently divided into tree and subtree patterns scaling the data into a range of $[0, 2^n - 1]$. Next, the middle vector of the input data has been selected as parent node p and the rest of the input data has been further divided into child nodes by converting the data into integer and binary vectors. The normalized has been converted to integers using BitBooster technique represented in [70]. By using the BitBooster technique, the selected vector has been firstly converted into dimension of 0s and 1s and then converting the resultant binary bits to a single integer. The conversion of normalized data to integer representation using the BitBooster is shown in Table 2.

TABLE 2. Single integer representation using bitbooster technique.

Input	0.42	3.94	3.44	0.82	2.63	0.92	0.14	4.43
Scaled Data	0	1	1	0	1	0	0	1
x_B	$(01101001)_2$=53							

The technique has been represented as;

$$x_b = \sum_{i=1}^{|V|} 2^{|V|-i} . v_i(x_i) \tag{17}$$

where, the first-dimension value x_1 has been represented by the most significant bit of x_b and the least significant bit has been depicted in last dimension x_n.

2) SELECTION OF NEAREST NEIGHBORS

The next step is calculating Euclidean distance and hamming weight on the integer representative vector array that selects the nearest neighbors in the integer vector array by measuring the distance between the two points. The hamming weight has been used to count the total number of high bits in an integer [71], [72]. The existing K-Ary classifier has used a naive approach to select the nearest nodes. The naive approach is not capable of producing the exact results and may predict the wrong classes even if the probability of belongingness of an object to a certain class is zero. However, in our approach Euclidean distance has been calculated on the integer vector array as shown in Fig. 13, and then hamming distance has given more robust results than the naïve approach in the existing K-Ary classifier.

$$S_b = \sqrt{HW\{d(x_1, x_2, \ldots, x_n)\}} \tag{18}$$

where, d is the Euclidean distance of the vector x_1, x_2 up to x_n. The HW is the hamming weight which counts the number of high bits in the integer.

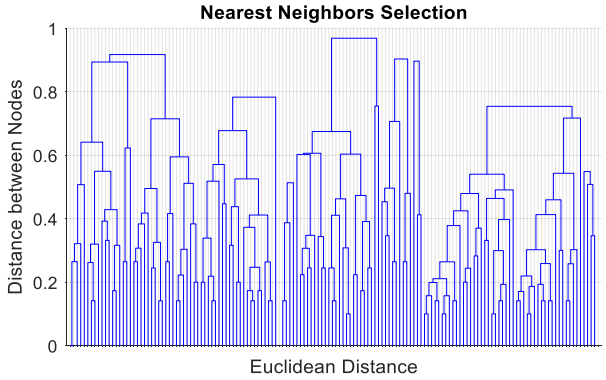

FIGURE 13. The nearest neighbor's selection of nodes using Euclidean distance.

3) SUBTREE PATTERN CLASSIFICATION

The existing K-Ary has used the MinHash algorithm to classify subtrees patterns. The numerous systems, such as relational data systems, key-value stores, compilers, and networks, depend on hashing. The computational and data-intensive nature of hashing makes it a core system bottleneck. Hash tables in the TPC-H benchmark may cost 50% of the total cost for a single database query. Similarly, Google spends at least 2% of its overall processing on C++ hash tables. Only one hashing operation alone results in a significant annual cost footprint. Moreover, MinHash is its $O(t.|A|)$ running time. For practical applications, where the data is highly dimensional, this sketch creation time is often a bottleneck [73]. Therefore, the one level entropy based hashing has been used in a novel KEC to classify the subtree patterns.

In a proposed KEC subtree pattern classification, within-cluster pattern entropy and between-cluster entropy have been used to evaluate the consistency of information within a single subtree and with other subtree patterns as shown in Fig. 14. The within-cluster entropy correlates the values of the subtree pattern with its own nodes. If the information is highly consistent with the subtree pattern nodes, it results in smaller entropy. The difference between two subtree patterns has also been assessed using the between-cluster entropy. The resultant will be high if the subtree patterns are more distinct from others.

The within-cluster entropy and between-clusters calculation has been done in parallel using parallel processing. For n subtree patterns N threads have been used to calculate the entropy of KEC classifier in no-time. The parallel processing on within-cluster and between-cluster enhance the computation efficiency of the novel KEC classifier. The following equation (19) and equation (20) have been used to calculate the entropy of within-cluster and between cluster subtree patterns as;

$$e(x_i) = -\frac{1}{1-a} log \int pdf^a(x_i)dx \tag{19}$$

$$E(X_i) = -log \frac{1}{(2\pi)^n 2k^2 h^{2n}}$$

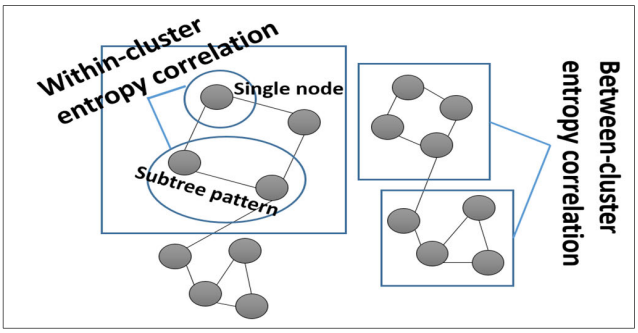

FIGURE 14. The subtree pattern classification, within-cluster pattern entropy and in between-cluster entropy.

$$\times \sum_{i=1}^{k} \sum_{j=1}^{k} M(d_i d_j) exp\left(\frac{\sum_{p=1}^{n} (w_{ip} - w_{jp})^2}{2h^2}\right) \tag{20}$$

where, a is the entropy constant that has been set as 2 where pdf represents probability density function in equation (19). Where, h has been set to 0.5. X_i is the subtree patterns in the K-Ary and k is depicts dimensional vector space. w_{ip} and w_{jp} depicts the weight of pth subtree patterns in equation (20).

4) WORKING OF K-ARY ENTROPY HASHING

In this section, the novel KEC algorithm has been proposed to classify the optimized data features in terms of tree and subtree patterns. The input data vector has been denoted by N_v in the K-Ary entropy classifier tree pattern. This algorithm has been motivated by BitBooster which calculate the Euclidean distance on the integer and calculate the hamming distance to find the nearest nodes in the input data [74]. Moreover, hashing computation cost too high in a real-time processing. Therefore within-subtree pattern and between-subtree patterns have been calculated to efficiently classify the data.

The working of overall algorithm start by linearly scaling $v_i^n(x_i)$ the data x_i into a range of $[0, 2^n - 1]$. The normalized data v_i^n has been then converted into integers using BitBooster technique x_b. The parent node p has been selected half way between the normalized data vector array. Next, subtree pattern have been formulated by calculating Euclidean distance S_b over the resultant integer vector. Finally, within-subtree patterns and between-subtree patterns have been finally classified using entropy $E(X_i)$. The results has been depicted in Fig. 15.

The Fig. 15 illustrate the final results of KEC in which nodes connected together to form subtree pattern and all subtree pattern have been connected together to form a tree. Furthermore, the tree structure contains a parent node and rest of the nodes are child nodes that have been linked together using Euclidean distance to form the pattern and subtree patterns. The final classification of subtree patterns have been done using Euclidean distance. The complete flow of the proposed KEC has been depicted in Fig. 16. It illustrate the

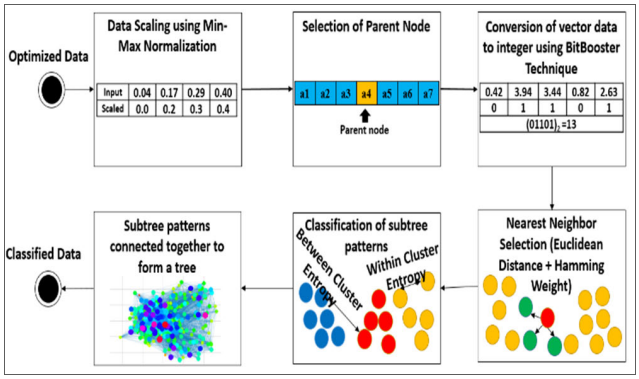

FIGURE 15. The flow diagram of proposed K-Ary entropy classifier.

final results of KEC in which nodes connected together to form subtree pattern and all subtree pattern have been connected together to form a tree. Furthermore, the tree structure contains a parent node and rest of the nodes are child nodes that have been linked together using Euclidean distance to form the pattern and subtree patterns. The final classification of subtree patterns have been done using Euclidean distance.

Our framework has not only enhanced accuracy but also enhanced computational efficiency. In order to obtain all embedded pivots in the tree, the algorithm requires a computational complexity of O(nlogn), where n is the number of nodes in the tree. In particular, we show that our classifier performs faster than the existing K-Ary tree classifier on the benchmark datasets.

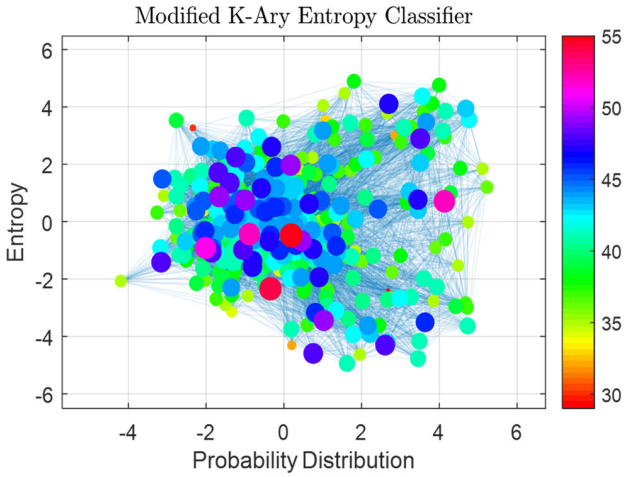

FIGURE 16. The results of the modified K-Ary entropy classifier.

IV. EXPERIMENTAL SETUP AND RESULTS

Experiments are conducted on a hardware platform with an Intel Core i7 processor clocked at 5 GHz, 16 GB RAM, Windows 11, and 64-bit operating system. All studies were carried out in Matlab by using various image-processing techniques and libraries. We conducted experiments including classification accuracy, precision, recall, and F1 score, taking into account the leave one subject out (LOSO) cross-validation scheme, to thoroughly evaluate the suggested framework. Recall is the ration of true positive instance over the sum of true positive and true negative instance respectively. Precision is the ratio of true positive instances to the total of truly predicted right instances.

The weighted average of recall and precision is the F1 score, which is as follows:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{21}$$

$$Recall = \frac{TruePositive}{TruePositive + TrueNegative} \tag{22}$$

$$F1score = \frac{2(Precision \times Recall)}{(Precision + Recall)} \tag{23}$$

We have taken into account three challenging benchmark datasets, namely ORGBD [57], RGBD-HuDaAct [ref], and CAD-60 [56], to assess the performance of the proposed architecture. We have described datasets in detail. Multiple experiments have been performed to compare the performance results of benchmark datasets against other state-of-the-art methods in the following subsections.

FIGURE 17. A sample image of SYSU-ACTION dataset on 12 activities that include pouring water, drink tea, listening to a phone call, wear a bag, scrolling mobile phone, sitting on a chair, putting books in a bag, putting wallet in the pocket, moving the chair aside, clean floor with a besom, take the card off the wallet, and mop the floor.

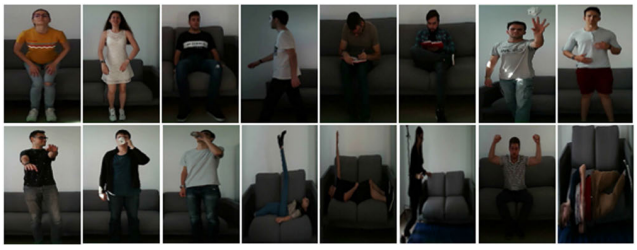

FIGURE 18. A sample image of PRECIS HAR dataset performed 16 different actions of sitting down, standing up, sitting still, walking, writing, reading, throwing paper, moving hands close to the body, move hands in front of the body, drink from mug, drink from the bottle, raising one leg up, raising one hand up, faint, cheer up, and fall from the bed.

A. DATASETS DESCRIPTION

We have taken into account three RGBD datasets (i.e., SYSU-ACTION dataset [75], PRECIS HAR dataset [76], Northwestern-UCLA (N-UCLA)) [77]. The following information about these datasets is provided:

TABLE 3. Confusion matrix of gait recognition accuracies over SYSU-ACTION dataset.

activity	PW	DT	LP	WB	SM	SC	PB	WP	MC	FB	CW	MF
PW	**96.00**	1.00	0.50	0.00	0.50	0.00	0.50	0.00	0.50	0.00	1.00	0.00
DT	2.00	**93.50**	1.00	0.50	0.00	0.00	0.50	1.00	0.00	0.50	0.00	1.00
LP	0.00	0.78	**95.22**	0.50	1.00	0.50	0.00	2.00	0.00	0.00	0.00	0.00
WB	0.50	0.00	0.50	**94.50**	0.00	0.00	0.50	0.00	2.00	0.50	0.00	1.50
SM	1.00	0.00	0.50	0.00	**96.00**	0.50	0.00	0.50	0.00	1.00	0.50	0.00
SC	0.50	0.50	2.00	0.50	0.00	**95.00**	0.00	0.50	1.00	0.00	0.00	0.00
PB	0.00	0.00	0.00	0.00	0.00	0.00	**98.00**	2.00	0.00	0.00	0.00	0.00
WP	0.55	2.00	0.00	0.50	2.50	2.50	0.00	**90.45**	0.00	0.00	1.50	0.00
MC	0.00	0.50	2.50	0.00	1.50	0.00	0.50	0.00	**94.50**	0.00	0.50	0.00
FB	1.50	0.00	0.00	0.50	0.00	0.00	0.50	0.00	0.00	**97.00**	0.00	0.50
CW	1.00	0.50	0.00	0.50	0.00	2.20	0.00	0.50	0.50	0.00	**94.80**	1.00
MF	0.00	2.27	0.00	0.00	1.00	0.00	0.50	0.50	0.00	0.00	0.00	**95.73**
Mean gait recognition accuracy = 95.05%												

*PW =pour water; DT =drink tea; LP =listen to phone call; WB = wear a bag; SM = scroll mobile phone; SC = sit on a chair; PB = put books in a bag; WP = put wallet in a pocket; MC = move the chair aside; FB = clean floor with besom; CW = take card off the wallet; MF = mop the floor.

TABLE 4. Confusion matrix of individual activity recognition accuracies over precis HAR dataset.

Objects	SD	SU	SS	WK	WR	RD	TP	HC	HF	DM	DB	RL	RH	FT	CU	FB
SD	**95.50**	2.50	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SU	1.72	**96.28**	0.50	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SS	0.00	0.00	**99.00**	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WK	0.00	0.00	0.00	**100.0**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WR	1.00	1.50	0.50	0.00	**90.45**	0.55	0.50	0.50	0.00	0.50	0.50	1.00	2.50	0.50	0.00	0.00
RD	0.50	0.50	1.50	0.00	0.50	**93.45**	0.00	0.55	0.00	0.50	0.00	0.50	0.00	1.50	0.50	0.00
TP	0.50	0.00	0.00	0.00	2.50	0.00	**92.50**	0.00	0.00	0.00	2.50	0.00	0.50	0.00	1.50	0.00
HC	0.00	0.00	0.00	1.50	0.00	0.00	0.00	**96.50**	0.50	0.50	0.00	0.00	0.50	0.50	0.00	0.00
HF	1.00	0.50	0.50	0.00	0.00	0.55	0.00	0.00	**95.45**	0.00	0.00	0.50	0.50	0.00	0.00	1.00
DM	0.00	0.50	0.50	0.50	0.00	0.50	0.00	0.00	0.00	**97.50**	0.00	0.00	0.50	0.00	0.00	0.00
DB	0.50	0.00	0.50	0.00	0.55	0.00	0.00	0.50	0.00	0.00	**97.45**	0.00	0.50	0.00	0.00	0.00
RL	0.00	2.00	0.00	0.45	0.00	0.55	0.00	0.00	2.00	0.00	0.00	**93.00**	0.00	2.00	0.00	0.00
RH	2.50	1.00	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.00	0.50	0.50	**90.50**	1.50	0.50	0.00
FT	0.00	0.50	0.00	0.00	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	**98.50**	0.00	0.00
CU	0.50	0.50	0.55	0.50	0.50	0.45	0.5	0.00	1.50	0.00	0.00	0.00	0.00	0.00	**95.00**	0.00
FB	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	**98.00**
Mean gait recognition accuracy = 95.56%																

*SD = sit down, SU = stand up, SS = sit still, WK = walk, WR = write, RD = read, TP = throw paper, HC = move hands close to the body, HF = move hands in front of the body, DM = drink from mug, DB = drink from bottle, RL = raise one leg up, RH = raise one hand up, FT = faint, CU = cheer up, FB = fall from bed.

The RGBD dataset consists of an RGBD dataset performed by 40 participants based on human-object interactions. The participants manipulated 5 different objects including a bag, phone, chair, wallet, besom, and mop to perform 12 different actions that include pouring water, drinking tea, listening to phone calls, wearing a bag, scrolling mobile phone, sitting on a chair, put books in a bag, put the wallet in the pocket, move the chair aside, clean floor with a besom, take the card off the wallet, and mop the floor. A Microsoft Kinect was used to record 480 video clips of RGB, depth sequence, and skeleton data frames within the range of 1.9s to 21s. In this paper, only RGB and depth sequence frames

TABLE 5. Confusion matrix of gait recognition accuracies over Northwestern-UCLA dataset.

Scenes	PO	DT	WA	CR	SU	PT	TH	DF	SD	DN
PO	**95.45**	0.00	0.00	1.00	1.00	2.55	0.00	0.00	0.00	0.00
DT	0.50	**95.25**	0.50	1.75	0.50	0.00	0.50	0.00	0.50	0.50
WA	0.50	0.50	**94.65**	1.50	0.35	0.50	0.50	1.00	0.00	0.50
CR	2.00	0.00	0.00	**96.35**	0.00	1.65	0.00	0.00	0.00	0.00
SU	2.50	0.00	0.00	0.00	**94.60**	2.50	0.00	0.40	0.00	0.00
PT	0.50	0.50	0.45	0.50	0.50	**95.55**	0.50	0.50	0.00	1.00
TH	0.25	1.00	0.50	2.50	0.00	0.00	**93.25**	0.00	2.50	0.00
DF	1.00	0.65	0.50	0.50	0.50	0.50	0.50	**94.35**	0.50	1.00
SD	0.00	0.65	0.50	0.50	1.00	0.50	0.00	0.00	**96.35**	0.50
DN	0.00	0.50	0.50	0.00	0.50	0.50	0.00	2.95	0.00	**95.05**
Mean gait recognition accuracy = 95.08%										

* PO = pick up with one hand; DT = drop trash; WA = walk around; CR = carry; SU = stand up; PT = pick up with two hands; TH = throw; DF = doffing; SD = sit down; DN = donning.

TABLE 6. Measurements of precision, recall and f1 score of the proposed method over SYSU-ACTION dataset.

Class	Precision	Recall	F1 score	Class	Precision	Recall	F1 score
PW	0.960	0.952	0.957	**PB**	0.980	0.959	0.938
DT	0.935	0.939	0.931	**WP**	0.904	0.912	0.907
LP	0.952	0.957	0.957	**MC**	0.945	0.950	0.927
WB	0.945	0.935	0.948	**FB**	0.970	0.970	0.972
SM	0.960	0.949	0.967	**CW**	0.948	0.957	0.947
SC	0.950	0.938	0.947	**MF**	0.957	0.917	0.936
Mean Precision = 0.950			**Mean Recall = 0.944**				**Mean F1 score = 0.944**

TABLE 7. Measurements of precision, recall and f1 score of proposed method over precis HAR dataset.

Class	Precision	Recall	F1 score	Class	Precision	Recall	F1 score
SD	0.955	0.928	0.954	**HF**	0.954	0.909	0.931
SU	0.962	0.952	0.963	**DM**	0.975	0.922	0.948
SS	0.990	0.979	0.983	**DB**	0.974	0.961	0.961
WK	0.999	0.998	0.972	**RL**	0.930	0.927	0.937
WR	0.904	0.915	0.907	**RH**	0.905	0.907	0.919
RD	0.934	0.925	0.937	**FT**	0.985	0.967	0.973
TP	0.925	0.907	0.927	**CU**	0.950	0.948	0.952
HC	0.965	0.967	0.976	**FB**	0.980	0.978	0.976
Mean Precision = 95.56			**Mean Recall = 0.943**				**Mean F1 score = 0.951**

have been used for the recognition of smart surveillance system.

The PRECIS HAR is RGBD dataset that contains 16 different actions which includes sit down, standing up, sit still, walking, writing, reading, throwing paper, moving hands close to the body, moving hands in front of the body, drinking from the mug, drinking from the bottle, raising one leg up, raising one hand up, faint, cheer up, and fall from the bed.

TABLE 8. Measurements of precision, recall and f1 score of proposed method over Northwestern-UCLA dataset.

Class	Precision	Recall	F1 score	Class	Precision	Recall	F1 score
PO	0.954	0.953	0.970	PT	0.955	0.951	0.915
DT	0.952	0.941	0.945	TH	0.932	0.920	0.935
WA	0.946	0.936	0.953	DF	0.943	0.951	0.908
CR	0.963	0.968	0.945	SD	0.963	0.974	0.967
SU	0.946	0.940	0.947	DN	0.950	0.957	0.954
Mean Precision = 95.08			Mean Recall = 0.949				Mean F1 score = 0.943

TABLE 9. Comparison of recognition accuracy of proposed method with other state-of-the-art methods over MSRC, CALTECH 101 and PASCAL-VOC12 datasets.

Methods	SYSU-ACTION	PRECIS HAR	Northwestern-UCLA
Temporal fusion mechanism + CNN [29]	-	94.38%	-
Temporal + spatial + multitask convolutional neural network [30]	-	-	86.82%
BiLSTM [31]	75.35%	-	-
CNN, + RNN + LSTM [32]	86.6%	-	85.1%
K-Ary Tree Hashing classifier [15]	93.5%	-	-
Proposed Method	**95.05%**	**95.56%**	**95.08%**

A 3D camera Orbbec Astra Pro was used to record 800 videos of RGB and depth sequence frames performed by 50 subjects.

The Northwestern-UCLA (N-UCLA) is a benchmark dataset collected in a multi-view environment that contains depth and human skeleton data captured sequentially by three Kinect cameras. In this paper, only RGB and depth images have been used for the proposed method. The dataset contains total 1494 video clips. The 10 subjects have performed 10 activities including walk around, drop trash, stand up, carry, pick up with two hands, throw, doffing, sit down, pick up with one hand and donning.

FIGURE 19. Sample images of the Northwestern-UCLA dataset perform ten different activities that include picking up with one hand, dropping trash, walking around, carrying, stand up, picking up with two hands, throwing, doffing, sitting down, and donning.

B. PARAMETER SETTINGS AND EVALUATION

The experimental results show that when combined with our unique K-Ary entropy accumulation classifier, our proposed full body and point-based body features, can fairly distinguish between the various actions classes of the SYSU-ACTION dataset. With a mean accuracy of 95.05%, Table 3 presents the action recognition findings for each distinct activity as a confusion matrix. Some findings have confused some actions, such as the action of pouring water with the actions of drinking water, listening to a phone call, scrolling a mobile phone, and so forth. But overall outcomes have been fairly impressive.

The suggested technique was tested on 16 different actions using the PRECIS HAR dataset, and the results have been displayed in Table 4 with the best classification accuracy of 95.56%. However, it has been shown that a few actions, such as sitting still, walking, fainting, and falling from bed, have acquired the highest accuracies due to considerable feature behaviors, which have been favorably reflected in their recognition performance.

In this experiment Table 5 shows the performance of action recognition over ten different activities using the Northwestern-UCLA dataset with a mean accuracy of

95.08% when utilizing a novel K-Ary entropy accumulation classifier. Here, a few activities improve overall performance.

The precision, recall, and F1 score for each action are depicted in Tables 6, 7, and 8 of three benchmark datasets: SYSU-ACTION, PRECIS HAR, and Northwestern-UCL. At the same time, Table 9 depicts the comparison findings between the proposed and existing models.

V. CONCLUSION

In this paper, the proposed model has been based on five modules: pre-processing, feature extraction, optimization, and classifier. The concept of ground–plane and voxel density models has been incorporated into a 2.5D model which was later used in the features extraction module. Moreover, a novel entropy-based K-Ary classifier (KEC) has been implemented, which is originally based on the K-Ary hashing classifier KATH [70]. The KEC linearly scale the data using min-max normalization process. The normalized data is further converted to integer using BitBooster technique. Later on, parent node has been selected half way between normalized data vector. Next, subtree patterns have been formulated by calculating Euclidean distance and Hamming weight. Finally, entropy calculation lead the nodes into subtree patterns. The algorithm has attained a computational complexity of O(nlogn), where n is the number of nodes in the tree. The results have achieved an accuracy of 95.05%, 95.56%, and 95.08% over SYSU-ACTION, PRECIS HAR, and Northwestern-UCLA datasets.

Future research will focus on entropy-based features, depth features, and energy characteristics of numerous activities to enhance the results of activity recognition accuracy. The deep learning methods with our novel K-ary entropy accumulation classifier will greatly enhance the accuracy of the surveillance system. Moreover, we have also planned to implement our datasets based on RGB-D datasets for activity recognition.

REFERENCES

[1] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 596–603.

[2] Y. Abdulazeem, H. M. Balaha, W. M. Bahgat, and M. Badawy, "Human action recognition based on transfer learning approach," *IEEE Access*, vol. 9, pp. 82058–82069, 2021, doi: 10.1109/ACCESS.2021.3086668.

[3] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3200–3225, Mar. 2023, doi: 10.1109/TPAMI.2022.3183112.

[4] M. Batool, A. Jalal, and K. Kim, "Telemonitoring of daily activity using accelerometer and gyroscope in smart home environments," *J. Electr. Eng. Technol.*, vol. 15, no. 6, pp. 2801–2809, Nov. 2020.

[5] R. U. Shekokar and S. N. Kale, "Deep learning for human action recognition," in *Proc. 6th Int. Conf. Converg. Technol. (I2CT)*, Apr. 2021, pp. 1–5, doi: 10.1109/I2CT51068.2021.9418080.

[6] J. Hsieh, M. Chiang, C. Fang, and S. Chen, "Online human action recognition using deep learning for indoor smart mobile robots," in *Proc. Int. Conf. Comput., Commun., Intell. Syst. (ICCCIS)*, Feb. 2021, pp. 425–433, doi: 10.1109/ICCCIS51004.2021.9397242.

[7] A. Jalal, A. Ahmed, A. A. Rafique, and K. Kim, "Scene semantic recognition based on modified fuzzy C-mean and maximum entropy using object-to-object relations," *IEEE Access*, vol. 9, pp. 27758–27772, 2021.

[8] J. Tang, J. Luo, T. Tjahjadi, and Y. Gao, "2.5 D multi-view gait recognition based on point cloud registration," *Sensors*, vol. 14, no. 4, pp. 6124–6143, 2014.

[9] X. Suau, J. R. Casas, and J. Ruiz-Hidalgo, "Real-time head and hand tracking based on 2.5D data," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2011, pp. 1–6.

[10] M. H. Khan, M. S. Farid, and M. Grzegorzek, "Spatiotemporal features of human motion for gait recognition," *Signal, Image Video Process.*, vol. 13, pp. 369–377, Mar. 2019.

[11] M. Gochoo, S. B. U. D. Tahir, A. Jalal, and K. Kim, "Monitoring real-time personal locomotion behaviors over smart indoor-outdoor environments via body-Worn sensors," *IEEE Access*, vol. 9, pp. 70556–70570, 2021.

[12] Y. Shavit and I. Klein, "Boosting inertial-based human activity recognition with transformers," *IEEE Access*, vol. 9, pp. 53540–53547, 2021, doi: 10.1109/ACCESS.2021.3070646.

[13] T. Liu, J. Kong, and M. Jiang, "RGB-D action recognition using multimodal correlative representation learning model," *IEEE Sensors J.*, vol. 19, no. 5, pp. 1862–1872, Mar. 2019.

[14] H. Hwang, C. Jang, G. Park, J. Cho, and I. Kim, "ElderSim: A synthetic data generation platform for human action recognition in eldercare applications," *IEEE Access*, vol. 11, pp. 9279–9294, 2023, doi: 10.1109/ACCESS.2021.3051842.

[15] N. Khalid, Y. Y. Ghadi, M. Gochoo, A. Jalal, and K. Kim, "Semantic recognition of human-object interactions via Gaussian-based elliptical modeling and pixel-level labeling," *IEEE Access*, vol. 9, pp. 111249–111266, 2021.

[16] Y. Liu, K. Wang, H. Lan, and L. Lin, "Temporal contrastive graph learning for video action recognition and retrieval," in *Computer Vision*. Springer, 2021.

[17] K. Chou, M. Prasad, D. Wu, N. Sharma, D. Li, Y. Lin, M. Blumenstein, W. Lin, and C. Lin, "Robust feature-based automated multi-view human action recognition system," *IEEE Access*, vol. 6, pp. 15283–15296, 2018, doi: 10.1109/ACCESS.2018.2809552.

[18] W. Hu, J. Zhang, B. Huang, W. Zhan, and X. Yang, "Design of remote monitoring system for limb rehabilitation training based on action recognition," *J. Phys., Conf. Ser.*, vol. 1550, no. 3, May 2020, Art. no. 032067.

[19] X. Weiyao, W. Muqing, Z. Min, L. Yifeng, L. Bo, and X. Ting, "Human action recognition using multilevel depth motion maps," *IEEE Access*, vol. 7, pp. 41811–41822, 2019, doi: 10.1109/ACCESS.2019.2907720.

[20] M. Waheed, A. Jalal, M. Alarfaj, Y. Y. Ghadi, T. A. Shloul, S. Kamal, and D. Kim, "An LSTM-based approach for understanding human interactions using hybrid feature descriptors over depth sensors," *IEEE Access*, vol. 9, pp. 167434–167446, 2021.

[21] Y. Y. Ghadi, M. Javeed, M. Alarfaj, T. A. Shloul, S. A. Alsuhibany, A. Jalal, S. Kamal, and D. Kim, "MS-DLD: Multi-sensors based daily locomotion detection via kinematic-static energy and body-specific HMMs," *IEEE Access*, vol. 10, pp. 23964–23979, 2022.

[22] Z. Yu and W. Q. Yan, "Human action recognition using deep learning methods," in *Proc. 35th Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Nov. 2020, pp. 1–6, doi: 10.1109/IVCNZ51579.2020.9290594.

[23] X. Weiyao, W. Muqing, Z. Min, and X. Ting, "Fusion of skeleton and RGB features for RGB-D human action recognition," *IEEE Sensors J.*, vol. 21, no. 17, pp. 19157–19164, Sep. 2021, doi: 10.1109/JSEN.2021.3089705.

[24] N. Archana and K. Hareesh, "Real-time human activity recognition using ResNet and 3D convolutional neural networks," in *Proc. 2nd Int. Conf. Adv. Comput., Commun., Embedded Secure Syst. (ACCESS)*, Sep. 2021, pp. 173–177.

[25] J. Lee and B. Ahn, "Real-time human action recognition with a low-cost RGB camera and mobile robot platform," *Sensors*, vol. 20, no. 10, p. 2886, May 2020.

[26] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "MARS: Motion-augmented RGB stream for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7874–7883.

[27] I. M. Nasir, M. Raza, J. H. Shah, M. A. Khan, and A. Rehman, "Human action recognition using machine learning in uncontrolled environment," in *Proc. 1st Int. Conf. Artif. Intell. Data Analytics (CAIDA)*, Apr. 2021, pp. 182–187.

[28] A. Jalal, I. Akhtar, and K. Kim, "Human posture estimation and sustainable events classification via pseudo-2D stick model and K-ary tree hashing," *Sustainability*, vol. 12, no. 23, p. 9814, Nov. 2020.

[29] A. Popescu, I. Mocanu, and B. Cramariuc, "Fusion mechanisms for human activity recognition using automated machine learning," *IEEE Access*, vol. 8, pp. 143996–144014, 2020.

[30] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3D action recognition," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2842–2855, Jun. 2018.
[31] X. Wang, J. Hu, J. Lai, J. Zhang, and W. Zheng, "Progressive teacher-student learning for early action prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3551–3560.
[32] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1963–1978, Aug. 2019.
[33] L. Wang, Y. Xu, J. Cheng, H. Xia, J. Yin, and J. Wu, "Human action recognition by learning spatio-temporal features with deep neural networks," *IEEE Access*, vol. 6, pp. 17913–17922, 2018, doi: 10.1109/ACCESS.2018.2817253.
[34] Q. Li, W. Yang, X. Chen, T. Yuan, and Y. Wang, "Temporal segment connection network for action recognition," *IEEE Access*, vol. 8, pp. 179118–179127, 2020, doi: 10.1109/ACCESS.2020.3027386.
[35] N. Ikizler-Cinbis and S. Sclaroff, "Object, scene and actions: Combining multiple features for human action recognition," in *Proc. ECCV*, Sep. 2010, pp. 494–507.
[36] H. Basly, W. Ouarda, F. E. Sayadi, B. Ouni, and A. M. Alimi, "CNN-SVM learning approach based human activity recognition," in *Proc. ICISP*, 2020, pp. 271–281.
[37] X. Ou, P. Yan, Y. Zhang, B. Tu, G. Zhang, J. Wu, and W. Li, "Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes," *IEEE Access*, vol. 7, pp. 108152–108160, 2019, doi: 10.1109/ACCESS.2019.2931922.
[38] Z. Luo, J.-T Hsieh, N. Balachandar, S. Yeung, S, G. Pusiol, J. Luxenberg, and G. Li, "Computer vision-based descriptive analytics of seniors' daily activities for long-term health monitoring," in *Proc. Mach. Learn. Healthcare, Stanford*, Aug. 2018, pp. 17–18.
[39] S. Chung, J. Lim, K. J. Noh, G. Kim, and H. Jeong, "Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning," *Sensors*, vol. 19, no. 7, p. 1716, Apr. 2019.
[40] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with deeply transferred motion vector CNNs," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2326–2339, May 2018.
[41] R. Hadidi, J. Cao, Y. Xie, B. Asgari, T. Krishna, and H. Kim, "Characterizing the deployment of deep neural networks on commercial edge devices," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Nov. 2019, pp. 35–48.
[42] R. Adnan, Y. Ghadi, A. Suliman, C. Samia, A. Jalal, and J. Park, "CNN based multi-object segmentation and feature fusion for scene recognition," in *Proc. CMC*, 2022, pp. 1–19.
[43] M. Habermann, W. Xu, M. Zollhöefer, G. Pons-Moll, and C. Theobalt, "Livecap: Real-time human performance capture from monocular video," *ACM Trans. Graph.*, vol. 38, no. 2, p. 14, 2019.
[44] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "XNect: Real-time multi-person 3D human pose estimation with a single RGB camera," *ACM Trans. Graph.*, vol. 39, no. 4, 2020.
[45] T. Sapiński, D. Kamińska, A. Pelikant, and G. Anbarjafari, "Emotion recognition from skeletal movements," *Entropy*, vol. 21, no. 7, p. 646, Jun. 2019.
[46] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[47] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6016–6025.
[48] N. Garcia, P. Morerio, and V. Murino, "Modality distillation with multiple stream networks for action recognition," in *Proc. ECCV*, Sep. 2018, pp. 103–118.
[49] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet," in *Proc. CVPR*, Jun. 2018, pp. 6546–6555.
[50] M. Lee, S. Lee, S. Son, G. Park, and N. Kwak, "Motion feature network: Fixed motion filter for action recognition," in *Proc. ECCV*, Sep. 2018, pp. 387–403.
[51] Y. Li, Y. Li, and N. Vasconcelos, "RESOUND: Towards action recognition without representation bias," in *Proc. ECCV*, Sep. 2018, pp. 513–528.
[52] Z. Luo, J.-T. Hsieh, L. Jiang, J. C. Niebles, and L. Fei-Fei, "Graph distillation for action detection with privileged modalities," in *Proc. ECCV*, Sep. 2018, pp. 166–183.
[53] J. Y. Ng, J. Choi, J. Neumann, and L. S. Davis, "ActionFlowNet: Learning motion representation for action recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1616–1624.
[54] L. Sevilla-Lara, Y. Liao, F. Guney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition," in *Proc. GCPR*, 2018, pp. 281–297.
[55] D. Sun, X. Yang, M. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.
[56] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1390–1399.
[57] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.
[58] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. ECCV*, Sep. 2018, pp. 305–321.
[59] Y. Zhu, L. Zhenzhong, S. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," in *Proc. ACCV*, 2018, pp. 363–378.
[60] G. Kapidis, R. Poppe, E. Van Dam, L. Noldus, and R. Veltkamp, "Egocentric hand track and object-based human action recognition," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2019, pp. 922–929.
[61] M. V. da Silva and A. N. Marana, "Human action recognition in videos based on spatiotemporal features and bag-of-poses," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106513.
[62] M. F. Aslan, A. Durdu, and K. Sabanci, "Human action recognition with bag of visual words using different machine learning methods and hyperparameter optimization," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 8585–8597, Jun. 2020.
[63] I. M. Nasir, M. A. Khan, A. Armghan, and M. Y. Javed, "SCNN: A secure convolutional neural network using blockchain," in *Proc. 2nd Int. Conf. Comput. Inf. Sci. (ICCIS)*, Oct. 2020, pp. 1–5.
[64] I. M. Nasir, M. A. Khan, M. Yasmin, J. H. Shah, M. Gabryel, R. Scherer, and R. Damaševičius, "Pearson correlation-based feature selection for document classification using balanced training," *Sensors*, vol. 20, no. 23, p. 6793, Nov. 2020.
[65] I. M. Nasir, M. Rashid, J. H. Shah, M. Sharif, M. Y. H. Awan, and M. H. J. C. M. I. Alkinani, "An optimized approach for breast cancer classification for histopathological images based on hybrid feature set," *Current Med. Imag.*, vol. 71, pp. 136–145, 2021.
[66] I. M. Nasir, A. Bibi, J. H. Shah, M. A. Khan, M. Sharif, K. Iqbal, Y. Nam, and S. Kadry, "Deep learning-based classification of fruit diseases: An application for precision agriculture," *Comput., Mater. Continua*, vol. 66, no. 2, pp. 1949–1962, 2021.
[67] M. A. Khan, I. M. Nasir, M. Sharif, M. Alhaisoni, S. Kadry, S. A. C. Bukhari, and Y. Nam, "A blockchain based framework for stomach abnormalities recognition," *Comput., Mater. Continua*, vol. 67, no. 1, pp. 141–158, 2021.
[68] N. Yudistira and T. Kurita, "Correlation net: Spatiotemporal multimodal deep learning for action recognition," *Signal Process., Image Commun.*, vol. 82, Mar. 2020, Art. no. 115731.
[69] K. A. Folly, "A short survey on population-based incremental learning algorithm," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 339–344, doi: 10.1109/SSCI44817.2019.9002858.
[70] W. Wu, B. Li, L. Chen, X. Zhu, and C. Zhang, "K-Ary tree hashing for fast graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 5, pp. 936–949, May 2018, doi: 10.1109/TKDE.2017.2782278.
[71] M. M. Islam and T. Iqbal, "Multi-GAT: A graphical attention-based hierarchical multimodal representation learning approach for human activity recognition," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1729–1736, Apr. 2021.

[72] D. Surís, R. Liu, and C. Vondrick, "Learning the predictability of the future," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12602–12612.
[73] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 180–189.
[74] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6417–6427.
[75] J. F. Hu, W. S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2186–2200, Nov. 2017.
[76] A.-C. Popescu, I. Mocanu, and B. Cramariuc, "PRECIS HAR," *IEEE Dataport*, vol. 1, p. 1, Sep. 2021, doi: 10.21227/mene-ck48.
[77] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning, and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2649–2656.

MOUAZMA BATOOL is currently pursuing the Ph.D. degree with Air University, Pakistan. Her research interests include wearable and optical sensors, signal acquisition, the IoT, and life-log generation.

SAUD S. ALOTAIBI received the bachelor's degree in computer science from King Abdul Aziz University, in 2000, the master's degree in computer science from King Fahd University, Dhahran, in May 2008, and the Ph.D. degree in computer science from Colorado State University, Fort Collins, USA, in August 2015, under the supervision of Dr. Charles Anderson. He is currently an Assistant Professor in computer science with Umm Al-Qura University, Makkah, Saudi Arabia, where he started his career as an Assistant Lecturer, in July 2001. After that, he was a Deputy of the IT-Center for E-Government and Application Services, Umm Al-Qura University, in January 2009. From 2015 to 2018, he was with the Deanship of Information Technology to improve the IT services that are provided to Umm Al-Qura University. He is currently with the Computer and Information College as the Vice Dean for Academic Affairs. His current research interests include AI, machine learning, natural language processing, neural computing IoT, knowledge representation, smart cities, wireless, and sensors.

MOHAMMED HAMAD ALATIYYAH is currently an Assistant Professor in computer science with the Computer Science Department, Prince Sattam Bin Abdulaziz University, Saudi Arabia. His research interests include recommender systems and computer vision, such as group recommender systems and travel recommender systems; drone vision.

KHALED ALNOWAISER received the Ph.D. degree in computer science from Glasgow University, Scotland. He is currently an Assistant Professor with the Computer Engineering Department, Prince Sattam Bin Abdulaziz University, Saudi Arabia. His research interests include computer vision, optimization techniques, and performance enhancement.

HANAN ALJUAID received the B.S. degree from KAU University, and the M.S. and Ph.D. degrees in computer science from UTM University, in 2014. She is currently with the Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah Bint Abdul Rahman University (PNU), Saudi Arabia. She has published numerous articles in pattern recognition, the IoT, and data science. Her research interests include computer vision and NLP. Much of her work has been on improving the understanding, design, and performance of pattern recognition, mainly through the application of data mining and machine learning. She has given numerous invited talks and tutorials.

AHMAD JALAL received the Ph.D. degree from the Department of Biomedical Engineering, Kyung Hee University, Republic of Korea. He is currently an Associate Professor with the Department of Computer Science and Engineering, Air University, Pakistan. He is also a Postdoctoral Research Fellow with POSTECH. His research interests include multimedia contents and artificial intelligence.

JEONGMIN PARK received the Ph.D. degree from the College of Information and Communication Engineering, Sungkyunkwan University, South Korea, in 2009. He is currently an Associate Professor with the Department of Computer Engineering, Tech University of Korea, South Korea. Before joining Tech University of Korea, in 2014, he was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI) and a Research Professor with Sungkyunkwan University. His research interests include high-reliable autonomic computing mechanism and human-oriented interaction systems.