sustainability

MDPI

Article

Human Posture Estimation and Sustainable Events Classification via Pseudo-2D Stick Model and K-ary Tree Hashing

Ahmad Jalal [1], Israr Akhtar [1] and Kibum Kim [2,*]

[1] Department of Computer Science, Air University, Islamabad 44000, Pakistan; ahmadjalal@mail.au.edu.pk (A.J.); israrakhter.edu@gmail.com (I.A.)

[2] Department of Human-Computer Interaction, Hanyang University, Ansan 15588, Korea

* Correspondence: kibum@hanyang.ac.kr

Received: 17 October 2020; Accepted: 20 November 2020; Published: 24 November 2020

Abstract: This paper suggests that human pose estimation (HPE) and sustainable event classification (SEC) require an advanced human skeleton and context-aware features extraction approach along with machine learning classification methods to recognize daily events precisely. Over the last few decades, researchers have found new mechanisms to make HPE and SEC applicable in daily human life-log events such as sports, surveillance systems, human monitoring systems, and in the education sector. In this research article, we propose a novel HPE and SEC system for which we designed a pseudo-2D stick model. To extract full-body human silhouette features, we proposed various features such as energy, sine, distinct body parts movements, and a 3D Cartesian view of smoothing gradients features. Features extracted to represent human key posture points include rich 2D appearance, angular point, and multi-point autocorrelation. After the extraction of key points, we applied a hierarchical classification and optimization model via ray optimization and a K-ary tree hashing algorithm over a UCF50 dataset, an hmdb51 dataset, and an Olympic sports dataset. Human body key points detection accuracy for the UCF50 dataset was 80.9%, for the hmdb51 dataset it was 82.1%, and for the Olympic sports dataset it was 81.7%. Event classification for the UCF50 dataset was 90.48%, for the hmdb51 dataset it was 89.21%, and for the Olympic sports dataset it was 90.83%. These results indicate better performance for our approach compared to other state-of-the-art methods.

Keywords: context-aware features; human pose estimation; K-ary tree hashing; pseudo 2D stick model; ray optimization; sustainable events classification

1. Introduction

Human posture estimation (HPE) and sustainable event classification (SEC) are the most interesting and challenging areas of current research. Researchers put in a tremendous amount of effort to find a way to obtain ever better performance and results from HPE systems by applying different machine learning methods. Digital globalization means an immense amount of data is uploaded on social media, safe city projects, daily activity monitoring systems, hospital data, educational intuitional data, virtual reality, and robotics. These data need to be processed, evaluated or investigated by researchers in order to find human pose estimations, human motion information, and sustainable event classification [1–4]. On social media, a huge amount of video and image data is uploaded and shared for human-human and human-machine interaction. Human posture analysis enables us to identify human motion information and to estimate human postures such as walking, running, sitting, standing, and rest positions. Sustainable event detection is used to identify and classify human event information such as playing, sitting, running, handshaking, to name a few classifications from social media data [5–9].

Safe City Projects provide one of the best examples of security surveillance systems. Using safe city data, we can detect anomalous events in daily human life-log environments. When an anomalous event occurs, the system generates an alarm and activates nearby emergency service institutions [10–13]. These projects help save lives, time, manpower, and cost, but it remains a challenging domain for researchers. Using SEC, professionals can monitor the activities of patients, doctors, and other staff members inside hospitals. It is also helpful in sports events where indoor and outdoor activities can be monitored and classified. SEC systems can identify classes of sports using video data but there remain difficulties in classifying human events due to their complex nature and sometimes camera position is a critical factor. The identification of sustainable events and human posture estimation still need to be improved in the domains of feature mining, human skeleton extraction, and classification [14–16].

In this research article, we present a novel method for sustainable event detection and human pose estimation in which we propose a pseudo-2D stick model based on a view-independent human skeleton, full-body, and key points context-aware features extraction. After features extraction, ray optimization was applied for data optimization and K-ary tree hashing was applied for sustainable event classification. In features extraction, we extracted two main types of features. At first, features were extracted from the full human body; these features included energy features, sine features, distinct motion body parts features, and a 3D Cartesian view of smoothing gradients. Features which were extracted from human body parts are rich 2D appearance features, angular point features, and multi-point autocorrelation features. Furthermore, we applied a hierarchal optimization model in which we first applied ray optimization as a data optimizer, while event classification was archived by a K-ary hashing algorithm. The key contributions in this paper are:

- Due to the complex movement of the human body and complex events, 19 human body key points were detected using frame size, human silhouette size, and a change detection approach.
- We propose a pseudo-2D stick model using the information from detected key points, 2D stick model, volumetric data, degree of freedom, and kinematics. This produced much better accuracy for sustainable event classification.
- Context-aware feature extraction based upon the full human body and human body key points was applied. Energy features, sine features, distinct motion body parts, and a 3D Cartesian view of smoothing gradients were extracted from the full human silhouette. Rich 2D appearance features, angular point features, and multi-point autocorrelation features were extracted using human body key points information. With the help of the extracted feature vector, we produced more accurate sustainable event classification results.
- The hierarchical optimization approach is adopted in this paper where ray optimization was used for data optimization and K-ary tree hashing was applied for sustainable event classification.

The organization of this paper is as follows: Section 2 describes related works. In Section 3, SEC system methodology is discussed. Section 4 presents the experimental setup and results plus a comparison with state-of-the-art methods. In Section 5, conclusions and future directions are outlined.

2. Related Work

Cameras and various types of sensors provide the key parameters for SEC research in HPE and sustainable event classification systems. In this section, we discuss previous works in SEC based on 2D/3D images and body-marker sensors.

2.1. Sustainable Event Classification via 2D/3D Images

In sustainable event classification using 2D or 3D images, cameras play a vital role in monitoring human interaction in different places such as sports grounds, indoor and outdoor activities, shopping malls, educational institutions, hospitals, and roads. In [17], Li et al. presented a novel technique for event detection using a low-rank and compact coefficient dictionary learning (LRCCDL) algorithm. The image background is removed by extracting the histogram of the projected optical flow, then a low-rank compact

dictionary coefficient is obtained using joint optimization. In the last stage, event detection is performed using norm optimization and reconstruction cost. In [18], Einfalt et al. designed an approach for event detection in the motion of athletes with a two-step method in which the extraction of chronological 2D pose structures from videos are implemented. They designed a convolutional sequence network to precisely detect the event using translation task categorization. In [19], Yu et al. proposed a heuristic approach that can identify events through a single move from videos of soccer matches. This is shared with the replay recognition system to find maximum temporal context features for fulfilling spectator needs and producing story clips. In [20], Franklin et al. designed a general deep learning method for normal and anomalous event detection. They obtained results using a graph-based thresholds technique for classification and segmentation. For time utilization, they found normal and abnormal features from videos using a deep learning-based approach. In [21], Lohithashva et al. developed a novel fusion features descriptors method for violent event detection using local binary pattern (LBP) and gray level co-occurrence matrix (GLCM). They used supervised classification algorithms with extracted features for event classification. Feng et al. [22] designed a guided attention Long Short Term Memory (LSTM) system in which they extract the Convolutional Neural Network (CNN)based features and temporal location in complex video data. To detect humans in videos they used the You only Look Once YOLO v3 model while event detection is performed by a guided Long Short Term Memory (LSTM) based approach. Researchers implement these approaches with deep learning, Convolutional Neural Network (CNN) methods, or an inadequate set of features in video and image datasets. Our designed approach is based on statistical machine learning classification methods with a pseudo-2D stick model and seven different context-aware features. For experimental analysis, we used three publicly available benchmarked datasets.

2.2. Sustainable Event Classification via Body-Marker Sensors

In body marker-based SEC approaches, several body-markers are used such as diodes, infrared-based markers, and other sensors. These sensors are connected to the human body to estimate human body motion information [23]. Human joints and bones are the key areas to attach these digital sensors. Medical, sports, activity analysis, and human tracking systems are key fields for detection systems based on these sensors. In [24], Khan et al. proposed a body-marker-based approach for patients to provide therapy at home. Body-markers with a color indication system were attached to the human body joints to record the motion information of 10 patients. For sports activities, motion monitoring body-marker-based sensors were also used in [25] where Esfahani et al. designed a lightweight trunk motion approach (TMA) via body-worn sensors (BWS). In this system, 12 attachable sensors were designed to estimate 3D trunk movements and seven actions were performed. In [26], Golestani et al. developed a novel wireless system to estimate human physical activities. They tracked human events via a magnetic induction wire; body-markers were connected to human body key points. Deep RNN (recurrent neural network) and laboratory estimation were used to improve performance.

3. Designed System Methodology

In this section, we describe our proposed methodology to estimate human posture and sustainable event classification (SEC). Initially, the pre-processing phase covered the input of videos, noise removal, and frame sizing tasks. Then, the silhouette extraction, human detection, and human body key points were detected and represented as a pseudo-2D stick model. After that, seven different types of features that belong to two major categories (full-body features and key point features) were extracted. Then, ray optimization was applied to optimize the feature vector. Finally, a K-ary tree hashing classifier was applied for sustainable event classification. Figure 1 presents the complete model of the proposed system (SEC) methodology.

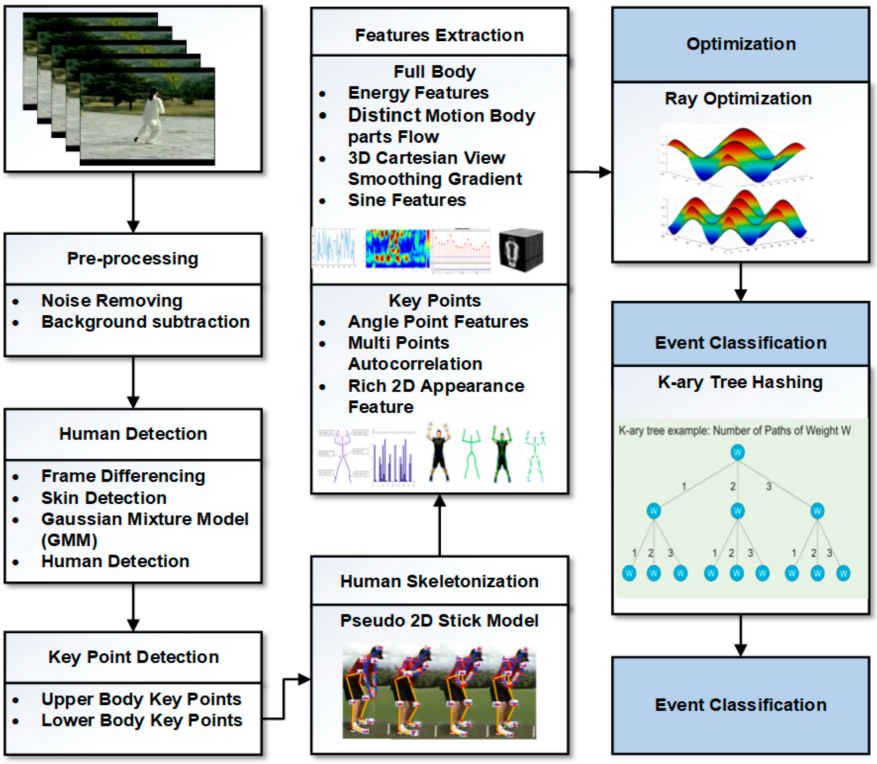


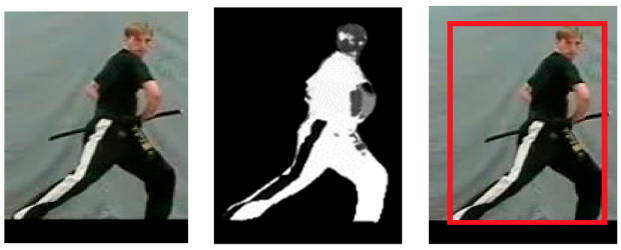
Figure 1. Design architecture of proposed sustainable event classification model.

3.1. Pre-Processing of Data and Human Detection

The primary aim of data pre-processing is to extract useable information to avoid extra data processing cost. Initially, we transformed video to images, and then converted RGB to gray-scale images to reduce image noise and unwanted information by applying a Gaussian filter [27]. After that, the background subtraction phase was achieved in which we applied the Gaussian mixture model (GMM) and a frame differencing technique to segment the human silhouette as a foreground. Then, segmentation of the human silhouette was performed which included histogram oriented human skin pixel thresholding along with Otsu's method to extract the human silhouette [28]. Equation (1) represents Otsu's method. Figure 2 shows the RGB image, human silhouette detection, and human detection.

$$O_m = \begin{cases} R\left(\frac{\mathrm{Th}_m + \mathrm{Th}_{ax}}{4}\right), & if\ \mathrm{Th}_{ax} \leq 10 \\ R\left(\frac{\mathrm{Th}_m + \mathrm{Th}_{ax}}{2}\right), & if\ \mathrm{Th}_{ax} > 10 \end{cases} \quad (1)$$

where R is the loop variable, Th_m is the threshold range which is defined by Otsu's procedure and Th_{ax} is the highest color frequency value of the defined histogram.



(**a**) (**b**) (**c**)

Figure 2. (**a**) RGB image, (**b**) human silhouette extraction, and (**c**) human detection results.

3.2. Human Posture Estimation: Human Key Points Detection

In order to initially identify human body key points, the torso point was detected by calculating the center of the detected human silhouette that estimated the outer human shape of the human body pixels H_{sp}. For detection of the torso point, Equation (2) was formulated as

$$Z_{Tp}^{f} \leftarrow Z_{T}^{f-1} + \Delta Z_{T}^{f-1} \tag{2}$$

where Z_{sp}^{f} is the torso point location in any video frame f. It was derived by calculating the frame differences (See Figure 3). For the detection of the human knee point, we took the point halfway between the foot and the hip points. Equation (3) represents the human knee point,

$$Z_{SK}^{f} = \left(Z_{SF}^{f} - Z_{SH}^{f}\right)/2 \tag{3}$$

where Z_{SK}^{f} is the human knee point, Z_{SF}^{f} is the human foot point, and Z_{SH}^{f} denotes the human body hip point. For elbow point extraction, we took the point halfway between the hand and shoulder points which is shown in Equation (4).

$$Z_{SE}^{f} = \left(Z_{SHN}^{f} - Z_{SSD}^{f}\right)/2 \tag{4}$$

where Z_{SE}^{f} is the human elbow point, Z_{SHN}^{f} is the human hand point, and Z_{SSD}^{f} was denoted as the human body shoulder point. Algorithm 1 shows the complete description of human body key point detection.

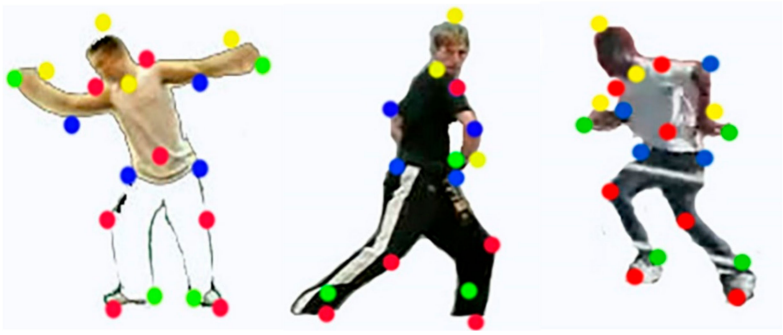


Figure 3. The 19 human body key points detection over UCF50, hmdb51, and Olympic sports datasets.

In this section, the human skeletonization [29,30] is considered to be a pseudo-2D stick model. Figure 4 presents a detailed overview of the pre-pseudo-2D stick model which comprises 19 human body key points which in turn comprise 3 main skeleton parts: upper body skeleton (*Ubs*), midpoint (*Mp*), and lower body skeleton (*Lbs*). *Ubs* is based on the linkage of the head (*HImg_H*), neck (*HImg_N*), shoulders (*HImg_S*), elbow (*HImg_E*), wrist (*HImg_W*), and hand points (*HImg_Hn*). *Lbs* is based upon the linkage of hips (*HImg_Hp*), knees (*HImg_K*), ankle (*HImg_A*), and feet *HImg_F*). Each key point takes a specific time t to perform a specific action. Equations (5)–(7) represent the relationships in the pre-pseudo-2D stick model:

$$U_{bst} = HImg\_H \bowtie HImg\_N \bowtie HImg\_S \bowtie HImg\_E \bowtie HImg\_W \bowtie HImg\_Hn \tag{5}$$

$$M_{pt} = \text{HImg\_Hp} \bowtie U_{bst} \tag{6}$$

$$L_{bs} = \text{HImg\_K} \bowtie \text{HImg\_A} \bowtie \text{HImg\_F} \bowtie M_{pt} \tag{7}$$

Algorithm 1. Human body key points detection.

Input: H_{ES}: Extracted human silhouette
Output: 19 body parts, specifically, head, neck, shoulders, elbows, wrists, hands, mid, hips, knees, ankles, and feet.
H_{ES} = human silhouette, H_S = human shape, H_I = height, W_I = width, L_I = left, R_I = right, I_H = head, I_N = neck

Repeat
For i = 1 to *N* **do**
Search (H_{ES})
I_H = *Human_head_point_Area;*
*HImg_*H_I = *UpperPoint* (I_H)
EH_I = *Human _End_Head_point* (I_H)
HImg_Feet = Bottom (H_S)
HImg_Mid = mid ($H_{I,}W_I$)/2
HImg_Foot = Bottom(H_S)*&earch*(L_I,R_I)
HImg_K = mid(HImg_Mid, HImg Foot)
HImg_H = HR &search(L_I,R_I)
HImg_S = search(I_H, I_N)*&search*(R_I,L_I)
HImg_E = mid(HImg_H, HImg_S)
HImg_W = mid (HImg_H, Img_S)/2
HImg_Hp = HImg_Mid &search(R_I,L_I)
HImg_A = mid (HImg_K, HImg Foot)/4
End
Until largest regions of extracted human silhouette are searched.
return 19 body parts: head, neck, shoulders, elbows, wrists, hands, mid, hips, knees, ankles, and feet.
H_{ES} = human silhouette, H_S = human shape, H_I = height, W_I = width, L_I = left, R_I = right, I_H = head, I_N = neck

Figure 4. A few examples of the pre-pseudo-2D stick model over UCF50, hmdb51 and Olympic sports datasets.

3.3. Pseudo 2D Stick Model

In this section, we propose a pseudo-2D stick model that enables an unbreakable human skeleton during the movement of the human body [29], and, due to its unbreakable nature, helps us achieve more accurate sustainable event classification. To achieve this, we detected 19 human body key points, then self-connection with each node was performed for the interconnection of every node. After this, we performed a fixed undirected skeleton graph which is called the 2D stick model (Section 3.2). Scaling of the stick was performed in which we included the upper and down side scaling. If the scaling limit of 20 pixels is exceeded, the stick model will not perform well. Equation (8) shows the mathematical representation of the stick model's scaling.

$$L_{bS} = Kps\left\{ \begin{array}{c} 1,\ if\ U \| L \leq 20 \\ 0,\ U \| L > 20 \end{array} \right\} \quad (8)$$

where L_{bS} is denoted as a human 2D-stick model, U is the upper limit, L is the lower limit, and Kps denotes key points scaling. We used volumetric and kinematic data to allow the skeleton to track

the human body key points. We used the size of the human silhouette to calculate the upper and lower distances. Then, frame size, consisting of the height and the width of the frame was estimated. Equation (9) was used to find the head position;

$$Z_{H\mathrm{h}}^{f} \leftarrow Z_{H\mathrm{h}}^{f-1} + \Delta Z_{H\mathrm{h}}^{f-1} \tag{9}$$

while Z_{Hh}^{f} is denoted as the head position in a given frame. To achieve pseudo-2D, we considered human motion direction, frame differencing from frame 1 to the next frame, and change detection which occurred in frame 1 to the upcoming frame. Human body edge information helped us to apply the degree of freedom for the angular rotation of the human skeleton while the local and global coordinate system and histogram of the oriented gradient (HOG) were performed. The detailed description of HOG is presented in Section 3.4.7.

We applied the Cartesian product [31] to achieve the ultimate results of the pseudo-2D stick model. Figure 5 shows the results of the pseudo-2D stick model and Algorithm 2 describes the complete procedure of the pseudo-2D stick model.

Algorithm 2 Pseudo 2D stick model.

Input: Human body key point and 2D stick model
Output: Pseudo 2D stick model ($p_1, p_2, p_3, \ldots, p_n$)
HD = human key points detection, SN = self-connection with each node, SS = scaling of sticks,
FG = fix undirected skeleton graph, VD = volumetric data, HK = human body key points tracking and kinematic dependency, KE = key points and edges information, DF = degree of freedom, LG = local and global coordinate system, CP = Cartesian product of skeleton graph.
% initiating pseudo 2D %
Pseudo 2D stick model ← []
P2DSM_Size ← Get P2DSM_Size ()
% for loop on segmented silhouettes frames of all interaction classes %
For I = 1:*N*
P2DSM_interactions ← GetP2DSM(interactions)
%Extracting HD, SN, SS, FG, VD, HK, KE, DF, LG, CP%
Human key points ← HD(P2DSM_interactions)
Self-connection with each node ← SN(P2DSM_interactions)
Scaling of sticks and key points ← SS(P2DSM_interactions)
Fix undirected skeleton graph ← FG(P2DSM_interactions)
Volumetric data ← VD(P2DSM_interactions)
Key points tracking ← HK(P2DSM_interactions)
Key points and edges information ← KEP2DSM_interactions)
Degree of freedom with root position ← DF (P2DSM_interactions)
Local and global coordinate system ← LG(P2DSM_interactions)
Cartesian product of skeleton graph ← CP(P2DSM_interactions)
Pseudo 2D stick model ← Get P2DSM
Pseudo 2D stick model.append (P2DSM)
End
Pseudo 2D stick model ← Normalize (pseudo 2D stick model)
return Pseudo 2D stick model ($p_1, p_2, p_3, \ldots, p_n$)

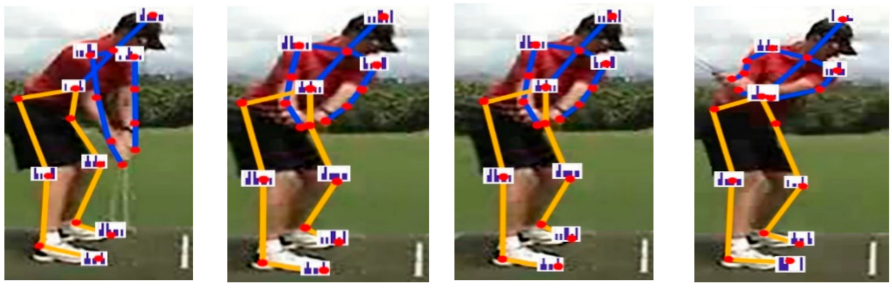


Figure 5. Pseudo-2D stick model results over the UCF50 dataset.

3.4. Context-Aware Features

In this section, we explain the complete overview of context-aware features that includes full-body features and three human body key points features for SEC. Algorithm 3 explains the procedure of context-aware feature extraction.

Algorithm 3 Context-aware feature extraction.

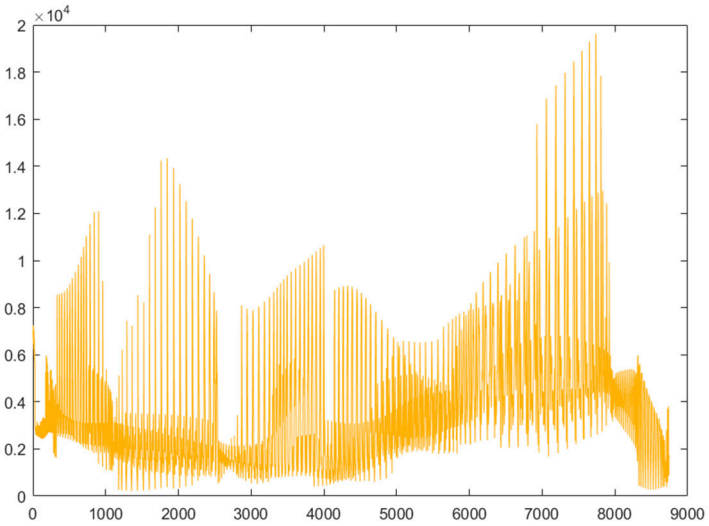
Input: N: Segmented silhouettes frames of RGB images
Output: context- awarefeature vectors(f_1,f_2,f_3, … ,f_n)
% initiating feature vector for sustainable event classification %
context-awarefeature-vectors ← []
Fearurevectorsize ← GetVectorsize ()
%for loop on segmented silhouettes frames of all interaction classes %
For i = 1:N
Featuresvectors_ interactions ← Getfeaturesvectors(interactions)
% extracting energy features, disting motion body parts flow, 3D cartesian view smoothing gradient, sine features , multi points auto correlation, rich 2D appearance feature%
Energy Features ← ExtractEnergyFeatures(Featuresvectors _interactions)
Disting Motion Body Parts Flow←ExtractdistinctMotionBodyPartsFlowFeatures (Featuresvectors _interactions)
3D Cartesian View Smoothing Gradient ←
Extract3DCartesianViewSmoothingGradientFeatures(Featuresvectors _interactions)
Sine Features ← ExtractSineFeatures(Featuresvectors _interactions)
Multi Points Auto correlation ← ExtractMultiPointsAutocorrelation(Featuresvectors _interactions)
Rich 2D Appearance Feature ← ExtractRich2DAppearanceFeatures(Featuresvectors _interactions)
Vectors Angle Point features ← ExtractVectorsAnglePointFeatures(Featuresvectors _interactions)
Feature-vectors ← GetFeaturevector
Context-aware Feature-vectors.append (Feature-vectors)
End
Context-aware Feature-vectors ← Normalize (context-aware Feature-vectors)
return context-awarefeature-vectors(f_1,f_2,f_3, … ,f_n)

3.4.1. Full Body: Energy Feature

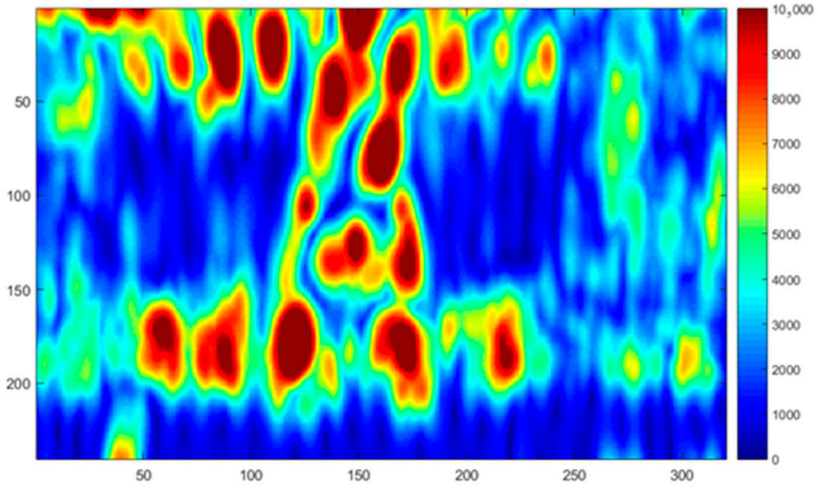
The full-body context-aware energy feature *En(t)* estimates human body key points movement in the energy matrix, which contains a set of indexes [0–10,000] over the detected human silhouette. After the distribution of energy indexes, we gathered only higher energy indexes using the thresholding method and mapped them all in a 1D vector. Energy distribution is represented in Equation (10) and the results of energy features are shown in Figure 6.

$$En(t) = \sum_{0}^{w} ImgR(i) \tag{10}$$

where $En(t)$ indicates the energy vector, i expresses index values, and $ImgR$ denotes the index value of assured RGB pixels.



(**a**)



(**b**)

Figure 6. Results of (**a**) 1D representation of energy vector, and (**b**) energy features.

3.4.2. Full Body: Distinct Motion Body Parts Flow Features

In this subset of a context-aware feature that is based upon full human body features, we estimated the motion flow of human body parts from frame 1 to frame n using motion estimation and changes detection methods. After obtaining the motion flow of human body parts we mapped the flow of all the distinct motion body parts in a 1D vector and then concatenated with energy feature. Equation (11) shows the correlation between distinct body parts flow.

$$f(Dmbf) = \sum_{n=1}^{n} (s_n M \parallel E_n M) \tag{11}$$

where $Dmbf$ is denoted as distinct motion body parts flow vector, n denotes the index integer, S is the starting index of motion flow, E is the ending index of motion, and M shows the motion flow. Figure 7 shows the distinct motion body parts features.

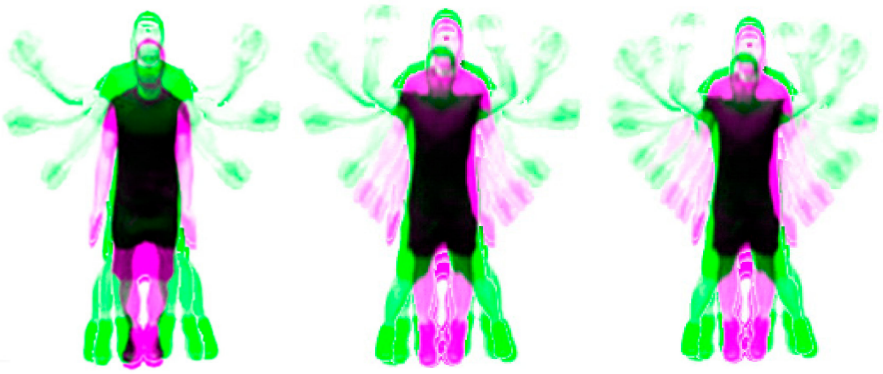


Figure 7. Distinct motion body parts features flow from starting as pink in color and ending in green.

3.4.3. Full Body: 3D Cartesian View Smoothing Gradient Features

From these full-body features, we determined the smoothing gradient of the detected human silhouette and calculated the gradient values of the refined full-body silhouette. After this, a 3D Cartesian product of the smoothing gradient values was obtained and the 3D Cartesian view was found so that we could obtain the 3D values. We then found the difference between every two sequential frames f and $f-1$ of the human silhouettes Hs. Equation (12) shows the mathematical formula for the 3D Cartesian view smoothing gradient. After extracting 3D values, we mapped them in a vector and concatenated with the main feature vector as;

$$CV_{TSF}(f) = \left| Hs^{f}_{TSF} - Hs^{f-1}_{TSF} \right| \tag{12}$$

where CV denotes the Cartesian view vector, and TSF is the top, side, and front views of the 3D Cartesian view smoothing gradient. Figure 8 shows the 2D and 3D Cartesian view smoothing gradient.

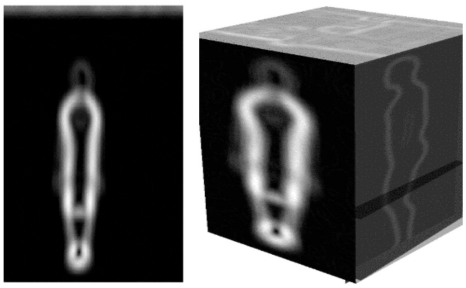


Figure 8. 2D and 3D representation of 3D Cartesian view smoothing gradient.

3.4.4. Full Body: Sine Features

To analyze the human body key points, we projected the human body key points in the x,y planes. In this way, we obtained the row and column values of the human body key points and applied trigonometric addition using sin $(\boldsymbol{\alpha} + \boldsymbol{\beta})$. Equation (13) represents the sine features:

$$\sin(\boldsymbol{\alpha} + \boldsymbol{\beta}) = \sin \boldsymbol{\alpha} \cos \boldsymbol{\beta} + \sin \boldsymbol{\beta} \cos \boldsymbol{\alpha} \tag{13}$$

After extracting the sine features, we mapped them in a vector and a concatenate sine feature vector with the main feature vector. Figure 9 shows the results of sine features over two different classes:

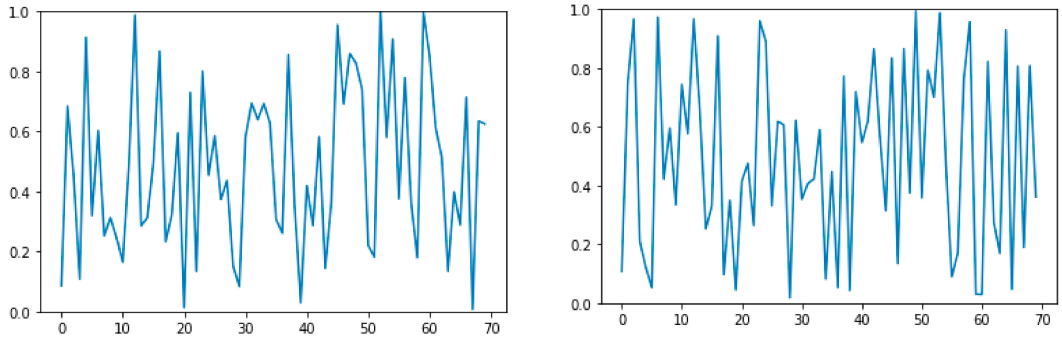


Figure 9. 1D representation of sine feature vector.

3.4.5. Key Body Points: Angle Point Feature

Angle point features were based on key body points. We divided the human silhouette into six main parts: part (A) contained the head, right shoulder, and right hand points; part (B) contained the head, left shoulder, and left hand points; part (C) contained neck, mid, and right knee points; part (D) contained neck, mid, and left knee points; part (E) contained right knee, right ankle, and right foot points; and, finally, part (F) contained left knee, left ankle and left foot points. By using these points, we got six triangles and determined the area. Equation (14) represents the area of human body parts and Figure 10 represents the angle point features

$$Area = \frac{1}{2}\{H_1(R_{H2} - R_{S2}) + R_{H1}(R_{S2} - H_2) + R_{S1}(H_2 - R_{H2})\} \quad (14)$$

where *Area* is defined as the area of a triangle, H_1, H_2 is denoted as the head point, R_{H1}, R_{H2} is denoted as the right-hand point, and R_{S1}, R_{S2} is denoted as the right shoulder point. We considered the same equation with different parameters to estimate the area of the remaining triangles.

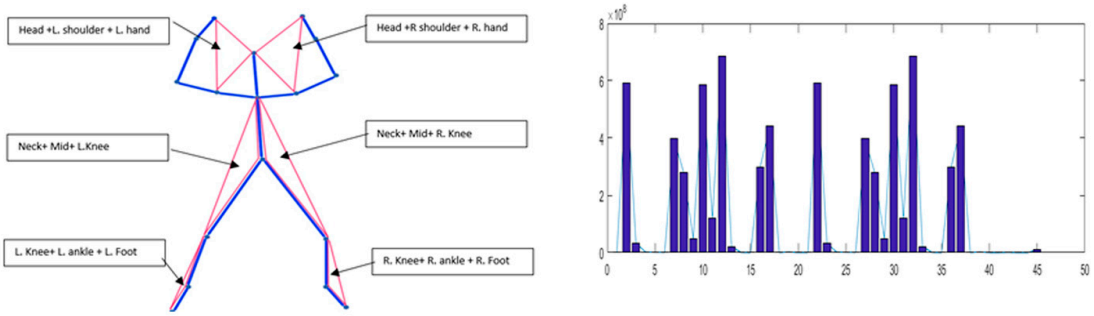


Figure 10. Angle point features (human body key points) jumping jack class result region vector [$(H+R_H+R_S)$, $(H+L_H+L_S)$, $(N+M+R_K)$, $(N+M+L_K)$, $(R_K+R_A+R_F)$, $(L_K+L_A+L_F)$].

3.4.6. Key Body Points: Multi-Points Autocorrelation Features

In the multi-points autocorrelation feature, we applied the windowing method on the 19 detected human body key points. We considered certain points as centers and took a 5×5-pixel window from the center in frame 1 to frame n. We then repeated this method for all detected human body parts. After obtaining a 5×5 window of all human body key points, we determined the autocorrelation. To find the mean of data $x_i, \ldots, x_n$ is

$$\overline{X} = \frac{1}{n\sum_{i=1}^{n} X_i} \quad (15)$$

where $\overline{X}$ is the mean, X_i is the input data, and lag p for $p \geq 0$ of the time is represented as

$$R_p = \frac{S_p}{S_0} \quad (16)$$

where S_0 is the variance of data, and R_p is the correlogram furcation. Figure 11 shows the results of the multi-points autocorrelation of various event-based classes.

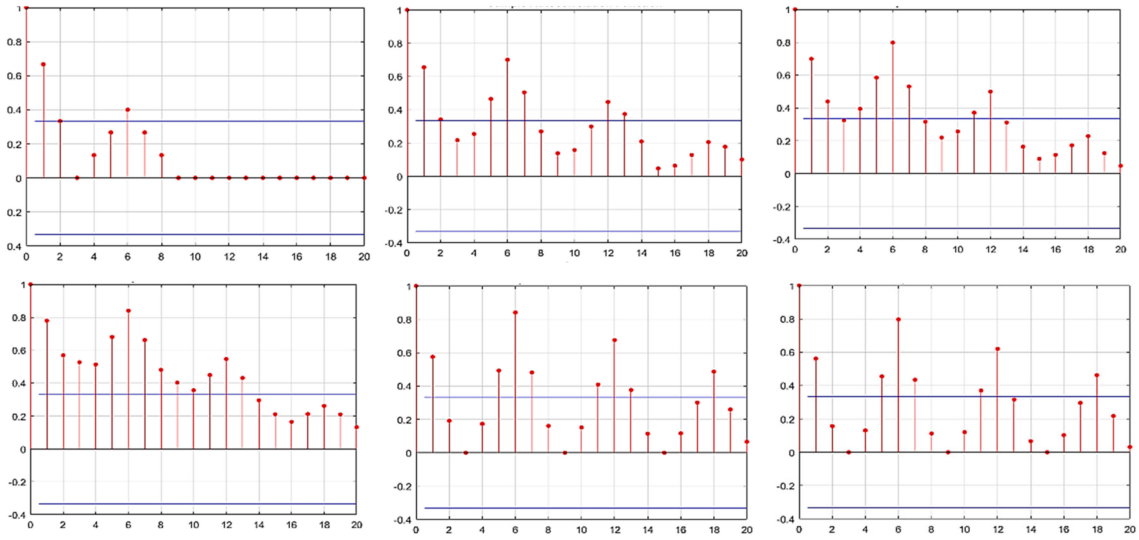


Figure 11. Results of multi-points autocorrelation on various human body key points.

3.4.7. Key Body Points: Rich 2D Appearance Feature

Context-aware rich 2D appearance features were based on key body points. Regarding these features, we applied a window of 10 × 10 pixels from the center point of the detected human body key point and repeated the same step for the other 18 human body key points. We then mapped these on a 2D stick model and found the histogram of the oriented gradient (HOG) for each 10 × 10 window from frame 1 to frame n with the spatial Gaussian window with σ = 10 pixels. Considering the horizontal gradient kernel range [−1, 0, 1] and a vertical gradient kernel range is [−1, 0, 1] we applied the following formula;

$$R_{2d} = W_{bp} \parallel L_{bs} \parallel H_{Ogb} \left\{ \begin{array}{c} Hgk\,[-1,\ 0,\ 1] \\ VGK[-1,\ 0,\ 1]^t \end{array} \right\} \tag{17}$$

where R_{2d} are rich 2D features, W_{bp} represents the windows of human body key points, L_{bs} is the human 2D stick model, H_{Ogb} is the histogram of the oriented gradient, Hgk is the horizontal gradient kernel, and VGK is the vertical gradient kernel. Figure 12 shows the complete description of the 2D rich appearance feature along with the windowing process, 2D stick, and HOG.

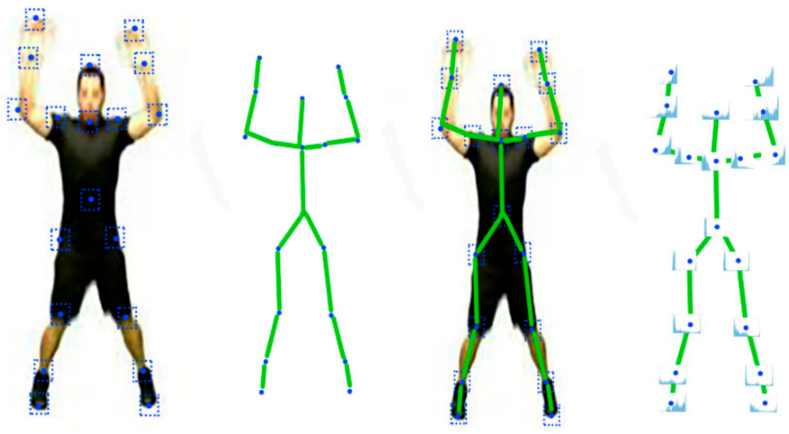


Figure 12. Functional structure of 2D rich appearance feature and 2D stick model.

3.5. Sustainable Event Optimization: Ray Optimization

The ray optimization (RO) algorithm consisted of several variables of a given problem [32]. These variable agents were represented as light rays which were based on the law of light refraction derived by Snell's law. Light refracts and changes its direction when it travels to a darker medium from a lighter medium. Every transparent material has refraction indexes; lighter material indexes are shown as *Lm*, and darker material is denoted by *Dm*. Equation (18) expresses Snell's law:

$$Lm.\sin\theta \;=\; Dm.\sin\varnothing \tag{18}$$

where θ and $\varnothing$ are present as the angles of the normal surface. *Lm* and *Dm* are refracted ray vectors. The RO has several agents representing the parameters of the layout dilemma, as any other meta-heuristic mechanism. The ray-tracing, which would be the primary basis of the RO, was discussed in two and three-dimensional structures as stated earlier, but a method for executing the algorithm's steps in high dimension spaces had to be implemented. We assumed there were four specification variables for an objective function. The target function for this objective function is calculated in the first step. Now, based on the RO algorithm, this solution vector had to be replaced by a new location in the search space. To accomplish this, the key value was split into two members of each group and then the respective participants were transferred to the new positions as shown in Equation (19)

$$A_{ij} \;=\; A_{j.min} + Rnd\left(A_{j.max} - A_{j.min}\right) \tag{19}$$

where A_{ij} is the ith agent of jth variable, $A_{j.max}$ and $A_{j.min}$ are the minimum and maximum limits, and *Rnd* denotes the random number with a limit of 0 to 1. A certain agent should now be relocated to its new location. The point that each entity transfers had to first be calculated. The core of this argument is called cockshy and it is defined by:

$$B_i^k \;=\; \frac{(iat+k).BG + (iat-K)BL_i}{2.iat} \tag{20}$$

where B_i^k denotes the *i*th agent of the origin, *iat* is the limit of total iterations while *BG* and *BL* are the local and global top agents. Figure 13 shows the RO flow chart, and Figure 14 shows the results over various event classes.

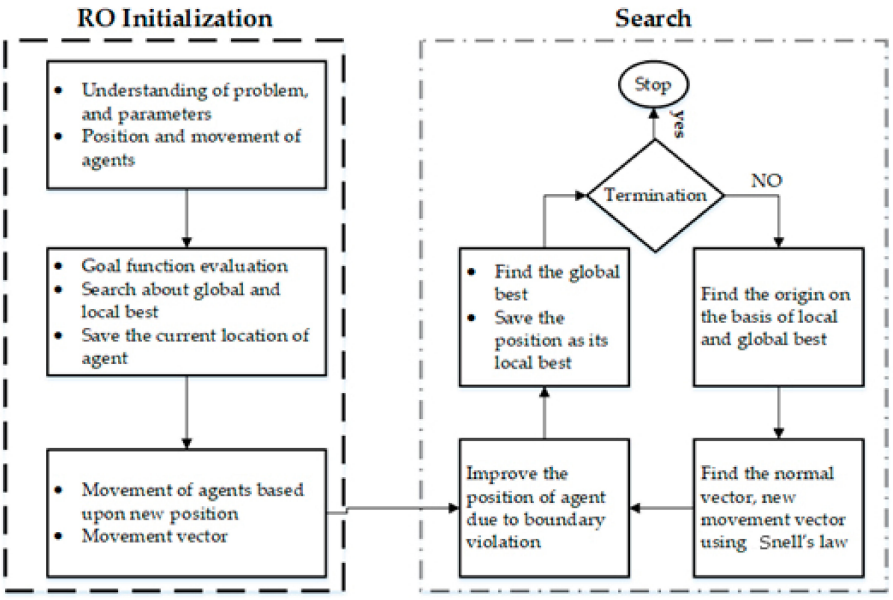
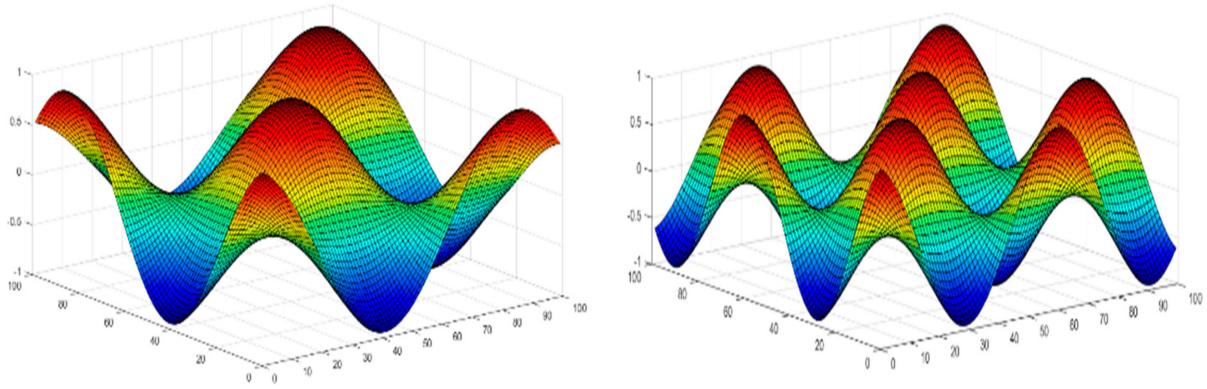


Figure 13. Ray optimization flow chart.



(**a**) (**b**)

Figure 14. Results of ray optimization of (**a**) diving and (**b**) golf swing over UCF50 dataset.

3.6. Sustainable Event Classification: K-ary Tree Hashing Algorithm

The K-ary tree hashing algorithm is based on a rooted tree in which every node has maximum k children [33]. For the classification, the min hashing approach was applied as the pre-step of K-ary tree hashing in which a similarity check between two sets A_i and A_j was performed. A set of B hash functions $\{\tilde{n}_b\}_{b=1}^{B}$ for A_* was applied, therefore the min hash function for A_* was $(\tilde{n}_b(A_*))$. To generate a permutation index we had:

$$\tilde{n}_b(j) \; = \; mod((L_d j \; + M_d), N_d) \tag{21}$$

where $L_d j$, M_d, N_d were the random permutations from the dataset, from set $|A|, 0$ j was the index. To find the best solution, the K-ary tree hashing algorithm adopts two approaches; to fix the number of neighboring nodes, a naïve approach is adopted, while for size fixing of any number, MinHashing is applied. The naïve approach is defined in Algorithm 4.

Algorithm 4 Naïve approach.

Require: L, N_i
Ensure: T(v)
% N is neighbor, L is Data, and T is size fixing approach%
1. *Temp* ← sort (L(N_i))
2. j ← min(j,|N_i|)
3. *t(i)* ← [i, index(*temp*(1 : j)]

Figure 15 shows the basic and graphical model of the K-ary hashing algorithm.

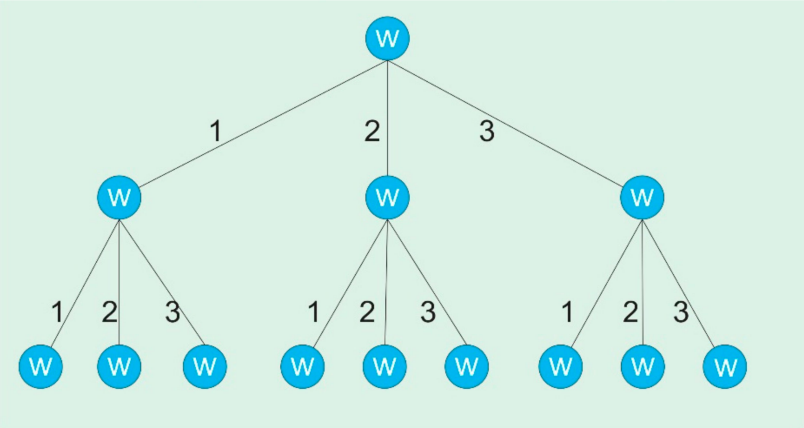


Figure 15. An example and classification model of the K-ary tree hashing algorithm.

4. Experimental Results and Analysis

In this section, we present complete details of our experimental model and data evaluation. Event classification accuracy of human body key points detection was used for the performance evaluation of the proposed system over available challenging publicly datasets. We used Matlab to carry out the experiments and a hardware system with an Intel Core i5 CPU at 1.6 GHz and 8 GB of RAM. To evaluate the performance of our proposed event classification system, we used three different publicly available datasets: UCF50 [34], hmdb51 [35], and Olympic sports [36] datasets. Other descriptions, the experimental data, and comparative analysis of the proposed datasets with the sustainable event classification method and other state-of-the-art event classification approaches are given below.

4.1. Datasets Description

4.1.1. UCF50

UCF50 is a famous publicly available action/event recognition dataset with 50 different classes, consisting of realistic videos that are taken from a famous YouTube website. UCF50 dataset's 50 classes collected from YouTube are baseball pitch, basketball shooting, bench press, biking, billiards shot, breaststroke, clean and jerk, diving, drumming, fencing, golf swing, playing guitar, high jump, horse race, horse riding, hula hoop, javelin throw, juggling balls, jump rope, jumping jack, kayaking, lunges, military parade, mixing batter, nunchucks, playing piano, pizza tossing, pole vault, pommel horse, pull-ups, punch, push-ups, rock climbing indoor, rope climbing, rowing, salsa spins, skateboarding, skiing, skijet, soccer juggling, swing, playing table, TaiChi, tennis swing, trampoline jumping, playing violin, volleyball spiking, walking with a dog, and yo-yo. Figure 16 shows examples of images from the UCF50 dataset.

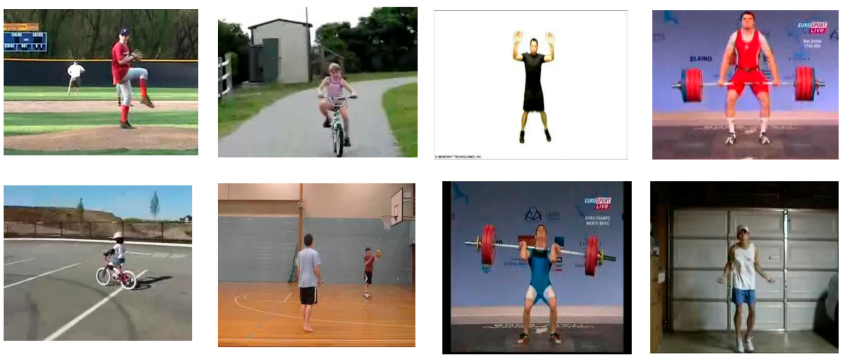


Figure 16. A few example frames from the UCF50 dataset.

4.1.2. hmdb51

The sports dataset known as hmdb51 is a publicly available dataset, collected from various sources, mostly from movies, and a small proportion from public databases such as the Prelinger archive, YouTube, and Google videos. The dataset contains 6766 clips divided into 51 classes: cartwheel, catch, clap, climb, climb_stairs, dive, draw_sword, dribble, fall_floor, fencing, flic_flac, golf, handstand, hit, hug, jump, kick, kick_ball, pick, pullup, punch, push, pushup, ride_bike, ride_horse, run, shake_hands, shoot_ball, shoot_bow, shoot_gun, situp, somersault, swing_baseball, sword, sword_exercise, *throw, turn,* and *walk.* Figure 17 shows examples of images from the hmdb51 dataset.

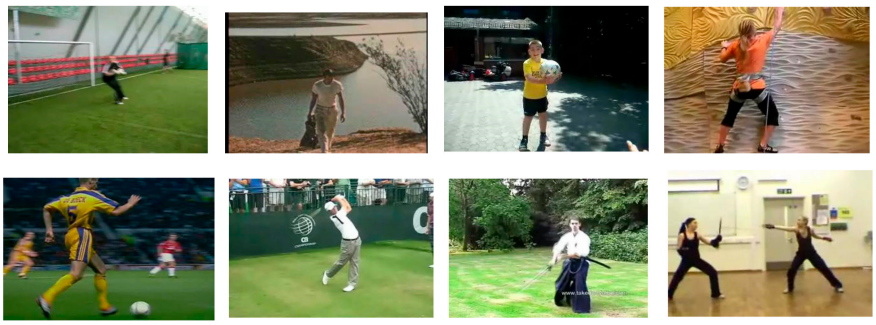


Figure 17. A few example frames from the hmdb51 dataset.

4.1.3. Olympic Sports

The Olympic sports dataset contains videos of athletes practicing in different sports. We obtained all video sequences from YouTube. They represent 16 sports classes: high jump, basketball_lay_up, long jump, bowling, triple jump, tennis serve, pole vault, platform diving, discus throw, springboard driving, hammer throw, snatch, javelin throw, clean and jerk, shot put and gymnastic vault. Figure 18 shows examples of images from the Olympic sports dataset.

Figure 18. A few example frames from the Olympic sports dataset.

4.2. Experimental Analysis

Three experimental measures were used to evaluate the performance of the system: accuracy of human body key points detection, the mean accuracy of sustainable event classification, and comparisons with other existing systems. The proposed SEC system achieved better performance compared to existing state-of-the-art methods.

4.2.1. Experiment 1: Human Body Key Points Detection

To test the efficiency of the proposed human body key points system, we estimated the Euclidean distance between the ground truth and labeled human body parts via the Euclidean range [37]. The threshold range of the ground truth where the error margin is set as 14 was used to find mean accuracy thus:

$$EC_D = \sqrt{\sum_{N=1}^{N} \left(\frac{X_{GN}}{S_{GN}} - \frac{Y_{DN}}{S_{DN}} \right)^2} \quad (22)$$

where X_{GN} is the defined ground truth, Y_{GN} is denoted as a detected point of the proposed method and EC_D denotes Euclidean distance. For estimation of human body parts accuracy, we used Equation (23):

$$A_{hp} = \frac{100}{N}\left[\sum_{N=1}^{N}\begin{cases} 1, \; if \; EC_D \leq 14 \\ 0, \; EC_D > 14 \end{cases}\right] \tag{23}$$

where A_{hp} is the estimated accuracy of the N human body part. If the estimated distance of a detected human key point was higher than 14, that detected body point was ignored. Otherwise the detected human body key point was included in the evaluation method. We repeated this procedure for all detected human key points from 1 to N in the UCF50 dataset as 80.9%, in the hmdb51 dataset as 82.1%, and the Olympic sports dataset as 81.7%. Table 1 shows that the proposed SEC model has the best human key points detection mean accuracies compared to other state-of-the-art methods.

Table 1. Human body key points detection accuracy.

Body Key Points	Distance	UCF50	Distance	Hmdb51	Distance	Olympic Sports
HP	9.3	91	11.3	85	9.9	90
NP	9.5	84	9.7	87	10.8	83
RSP	9.7	81	9.3	83	10.7	82
REP	11.0	76	10.2	78	12.1	80
RWP	9.4	72	10.7	75	9.7	75
RHP	12.3	83	11.3	84	11.4	80
LSP	11.2	82	12.7	84	13.2	81
LEP	10.3	77	12.9	79	11.1	80
LWP	11.5	71	10.8	77	12.5	74
LHP	9.3	82	10.3	86	8.8	85
MP	10.3	92	9.3	92	11.0	91
RHP	11.7	74	10.6	80	10.9	79
LHP	13.0	74	11.5	76	13.4	80
LKP	12.1	85	13.2	83	11.3	80
RKP	11.9	87	9.8	86	12.9	82
RAP	10.2	78	11.5	78	9.7	79
LAP	10.5	74	13.5	76	12.7	70
LFP	9.9	85	10.8	91	11.3	90
RFP	8.2	90	9.3	80	10.2	92
Mean Accuracy Rate		80.9%		82.10%		81.7%

HP = head point, NP = neck point, RSP = right shoulder point, REP = right elbow point, RWP = right wrist point, RHP = right hand point, LSP = lift shoulder point, LEP = left elbow point, LWP = left wrist point, LHP = left hand point, MP = mid-point, RHP = right hip point, LHP = left hip point, LKP = left knee point, RKP = right knee point, RAP = right ankle point, LAP = left ankle point, LFP = left foot point, RFP = right foot point.

4.2.2. Experiment 2: Event Classification over the UCF50 Dataset

In the first step of event classification, we applied the K-ary tree hashing algorithm over-optimized feature vectors of the UCF 50 datasets and we got a 90.48% mean accuracy rate. Table 2 shows the accuracy table of K-ary tree hashing over the UCF50 dataset and Table 3 shows the precision, recall, and F-1 score over the UCF50 dataset.

Table 2. Mean accuracy result of K-ary hashing over the UCF50 dataset.

Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc
BP	0.9	BB	1.0	BE	0.9	BK	0.8	CJ	0.7	DI	0.8	FE	1.0
GS	1.0	HJ	0.9	HR	0.9	HO	0.8	HH	0.9	JT	0.9	JB	0.9
JJ	1.0	JR	0.9	KK	0.9	LU	0.8	NC	0.9	PT	0.9	PV	0.9
PH	0.9	PU	1.0	PU	0.9	PU	0.9	RD	1.0	RC	1.0	RO	0.9
SS	0.9	SB	0.9	SK	0.9	SK	0.9	SJ	1.0	SW	0.9	TA	1.0
TS	0.8	TD	0.9	TJ	0.9	VS	1.0	WD	0.9	YY	0.8		
Mean event classification accuracy = 90.48%													

BP = baseball pitch, BB = basketball, BE = bench press, BK = biking, CJ = clean and jerk, DI = diving, FE = fencing, GS = golf swing, HJ = high jump, HR = horserace, HO = horse riding, HH = hula hoop, JT = javelin throw, JB = juggling balls, JJ = jumping jack, JR = jump rope, KK = kayaking, LU = lunges, NC = nunchucks, PT = pizza tossing, PV = pole vault, PH = pommel horse, PU = pull ups, PU = punch, PU = pushups, RD = rock climbing indoor, RC = rope climbing, RO = rowing, SS = salsa spin, SB = skateboarding, SK = skiing, SK = skijet, SJ = soccer juggling, SW = swing, TA = TaiChi, TS = tennis swing, TD = throw discus, TJ = trampoline jumping, VS = volleyball spiking, WD = walking with dog, YY = yo-yo.

Table 3. Precision, recall and F-Score over UCF 50 dataset.

Events	Precision	Recall	F1-Score	Events	Precision	Recall	F1-Score
Baseball pitch	0.429	0.900	0.581	Pull ups	1.000	0.769	0.857
Basketball	0.714	0.714	0.714	Punch	0.900	0.750	0.870
Bench press	1.000	0.818	0.900	Push-ups	0.900	0.900	0.818
Biking	0.727	0.800	0.762	Rock climbing indoor	1.000	0.833	0.900
Clean and jerk	0.778	0.636	0.700	Rope climbing	0.769	0.769	0.909
Diving	0.320	0.800	0.457	Rowing	0.900	0.900	0.769
Fencing	1.000	0.833	0.909	Salsa spin	0.818	0.692	0.900
Golf swing	1.000	0.769	0.870	Skateboarding	0.750	0.900	0.750
High jump	0.692	0.750	0.720	Skiing	1.000	0.900	0.818
Horse race	0.750	0.750	0.750	Skijet	0.750	1.000	0.947
Horse riding	0.400	0.800	0.533	Soccer juggling	1.000	0.909	0.857
Hula hoop	1.000	0.818	0.900	Swing	0.900	0.818	0.952
Javelin throw	0.563	0.750	0.643	TaiChi	1.000	0.833	0.857
Juggling balls	1.000	0.900	0.643	Tennis swing	0.800	0.818	0.909
Jumping jack	0.909	0.833	0.947	Throw discus	0.750	0.818	0.809
Jump rope	0.900	10.000	0.870	Trampoline jumping	0.692	0.750	0.783
Kayaking	0.900	10.000	1.651	Volleyball spiking	0.909	0.769	0.720
Lunges	0.889	0.727	1.651	Walking with dog	1.000	0.818	0.833
Nunchucks	0.818	0.750	0.800	Yo-yo	1.000	0.800	0.900
Pizza tossing	1.000	0.900	0.783	Pommel horse	1.000	0.750	0.857
Pole vault	1.000	0.750	0.947				

4.2.3. Experiment 3: Event Classification over the Hmdb51 Dataset

After the first step of event classification, we applied the K-ary tree hashing algorithm over the optimized feature vector of the hmdb51 datasets and we got a mean accuracy rate of 89.21%. Table 4 shows the accuracy table for K-ary tree hashing over the hmdb51 dataset and Table 5 shows the precision, recall, and F-1 score over the hmdb51 dataset.

Table 4. Mean accuracy result of K-ary hashing over hmdb51 dataset.

Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc
CW	1.0	CA	0.9	DA	1.0	DI	0.9	DS	0.8	DI	1.0	DS	0.8
DB	0.9	FF	0.9	FC	0.8	FF	0.9	GO	1.0	HS	0.8	HI	0.9
HU	0.9	JU	1.0	KI	0.8	KB	0.9	PI	0.8	PU	1.0	PU	0.9
PU	0.8	PU	1.0	RB	0.8	RH	0.9	RU	1.0	SH	0.8	SB	0.9
SB	0.8	SG	0.9	SU	1.0	SS	0.8	SB	0.8	SW	1.0	SE	0.9
TH	0.9	TU	0.8	WA	0.9								
Mean event classification accuracy = 89.21%													

CW = cartwheel, CA = catch, DA = dap, DI = dimb, DS = dimb_stairs, DI = dive, DS = draw_sword, DB = dribble, FF = fall_floor, FC = fencing, FF = flic_flac, GO = golf, HS = handstand, HI = hit, HU = hug, JU = jump, KI = kick, KB = kick_ball, PI = pick, PU = pullup, PU = punch, PU = push, PU = pushup, RB = ride_bike, RH = ride_horse, RU = run, SH = shake_hands, SB = shoot_ball, SB = shoot_bow, SG = shoot_gun, SU = situp, SS = somersault, SB = swing_baseball, SW = sword, SE = sword_exercise, TH = throw, TU = turn, WA = walk.

Table 5. Precision, recall and F-Score over hmdb51 dataset.

Events	Precision	Recall	F1-Score	Events	Precision	Recall	F1-Score
Cartwheel	0.909	0.833	0.870	Punch	0.900	0.900	0.900
Catch	0.900	0.750	0.818	Push	0.889	0.889	0.889
Dap	0.909	0.833	0.870	Pushup	0.909	0.909	0.909
Dimb	0.750	0.900	0.818	ride_bike	0.889	0.889	0.889
dimb_stairs	0.800	0.889	0.842	ride_horse	1.000	0.750	0.857
Dive	0.909	0.909	0.909	Run	0.909	0.909	0.909
draw_sword	0.727	0.889	0.800	shake_hands	0.727	0.889	0.800
Dribble	1.000	0.818	0.900	shoot_ball	0.900	0.900	0.900
fall_floor	0.900	0.900	0.900	shoot_bow	1.000	0.889	0.941
Fencing	0.889	0.889	0.889	shoot_gun	0.818	0.818	0.818
flic_flac	0.750	0.900	0.818	Situp	1.000	0.833	0.909
Golf	0.769	0.833	0.800	Somersault	0.889	0.889	0.889
handstand	0.800	0.889	0.842	swing_baseball	0.889	0.800	0.842
Hit	0.900	0.818	0.857	Sword	0.909	0.909	0.909
Hug	1.000	0.900	0.947	sword_exercise	1.000	0.900	0.947
jump	0.909	1.000	0.952	Throw	0.818	0.818	0.818
Kick	0.889	0.889	0.889	Turn	0.800	0.889	0.842
kick_ball	0.900	0.900	0.900	Walk	1.000	0.900	0.947
Pick	0.727	0.800	0.762	Pullup	0.667	0.833	0.741

4.2.4. Experiment 4: Event Classification over the Olympic Sports Dataset

In the last step of event classification, we applied a K-ary tree hashing algorithm over the optimized feature vectors of the Olympic sports datasets and we got the mean accuracy rate of 90.83%. Table 6 shows the accuracy table of K-ary tree hashing over the Olympic sports dataset and Table 7 shows the precision, recall, and F-1 score over classes of the Olympic sports dataset.

Table 6. Mean accuracy result of K-ary hashing over the Olympic sports dataset.

Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc	Event	Acc
BO	1.0	DT	0.9	DP	0.9	HT	0.8	JT	0.9	LJ	0.9	PV	1.0
SP	0.9	SN	0.8	TM	0.9	TJ	1.0	VA	0.9				
Mean Event Classification Accuracy = 90.83%													

BO = bowling, DT = discus_throw, DP = diving_platform_10m, HT = hammer_throw, JT = javelin_throw, LJ = long_jump, PV = pole_vault, S P = shot_put, SN = snatch, TM = toolbox-master, TJ = triple_jump, VA = vault.

Table 7. Precision, recall and F-Score over the Olympic sports dataset.

Events	Precision	Recall	F1-Score	Events	Precision	Recall	F1-Score
Bowling	0.769	0.833	0.800	pole_vault	1.000	0.909	0.952
discus_throw	0.692	0.900	0.783	shot_put	1.000	0.818	0.900
diving_platform_10m	1.000	0.750	0.857	snatch	1.000	0.800	0.889
hammer_throw	0.889	0.727	0.800	toolbox-master	0.750	0.818	0.783
javelin_throw	0.900	0.900	0.900	triple_jump	0.769	0.909	0.833
long_jump	0.643	0.900	0.750	vault	1.000	0.900	0.947

4.3. Comparison Analysis

4.3.1. Experiment 5: Comparison Using Various Classifiers

In this section, we applied various data classification algorithms such as an artificial neural network (ANN), genetic algorithm (GA), and AdaBoost over feature vectors. Table 8 shows better performance via the proposed K-ary tree hashing algorithm over other well-known statistical classifiers.

Table 8. Classifiers comparison table for datasets UCF50, hmdb51, Olympic sports.

Classifiers	Dataset	Accuracy	Dataset	Accuracy	Dataset	Accuracy
ANN	UCF50	84.63	hmdb51	83.94	Olympic sports	85.8
G.A	UCF50	86.34	hmdb51	86.31	Olympic sports	83.3
Adaboost	UCF50	85.36	hmdb51	88.42	Olympic sports	86.6
K-ary Tree	UCF50	**90.48**	hmdb51	**89.21**	Olympic sports	**90.83**

4.3.2. Experiment 6: Comparison of Various Features Combinations

We tested K-ary tree hashing over three features such as energy features, distinct motion body flow, and sine features. Then, we applied K-ary tree hashing over five features: 3D cartesian view smoothing gradient, angle point features, multi-points autocorrelation, rich 2D appearance features, and sine features. Finally, we applied K-ary tree hashing over seven features: 3D cartesian view smoothing gradient, angle point features, multi-points autocorrelation, rich 2D appearance features, sine features, energy feature, and distinct motion body flow, and obtained better mean classification accuracy. The pseudo-2D stick model helped us find sufficient accurate key points and the context-aware features helped us to estimate human posture, and it also helped us with sustainable event classification. Table 9 shows the comparison results for the three, five, and seven features sets over the UCF50, hmdb51, Olympic sports datasets using K-ary Tree hashing.

Table 9. Features comparison table for the datasets UCF50, hmdb51, Olympic sports using K-ary tree classifiers.

Features Name	UCF50/Accuracy	hmdb51/Accuracy	Olympic Sports/Accuracy
EF,DMBF, SF	76.09	74.21	75.83
3D-CVM, APF, MPA, R-2DA, SF	82.68	80.78	81.66
EF,DMBF, 3D-CVM, APF, MPA, R-2DA, SF	**90.48**	**89.21**	**90.83**

3D-CVM = 3D Cartesian view smoothing gradient, APF = angle point features, MPA = multi-points autocorrelation, R-2DA = rich 2D appearance features, SF = sine features, EF = energy feature, DMBF = distinct motion body flow.

4.3.3. Experiment 7: Event Classification Comparison with State-of-the-Art Methods

In [38], M. Jain et al. proposed a novel dense trajectories method to achieve enhanced performance and accuracy but their weak features extraction method is one of the main drawbacks of the system. Shi et al. [39] designed an Histogram of gradient (HOG), Histogram of Flow (HOF), HOG3D and Motion Boundary Histograms MBH descriptors-based approach which works on a random sampling technique, however, existing features are one of the main reasons for low accuracy. In [40], J. Uijlings et al. designed

a robust approach to optimize the HOG, HOF, MBH-based features for classification; existing features affect the overall performance of the system. H. Wang et al. [41] proposed a motion estimation model with an explicit camera to enhance the dense trajectory features. They used the Speed-up Robust Features (SURF) descriptor and optical flow for the extraction of feature vectors but the drawback of the system is its small number of features. K. Hara et al. [42] developed a Convolutional Neural Network CNNs-based approach to recognize human events in video and image data; although the overall system is robust, the issue is the weak model of the system. Y. Li et al. [43] proposed a novel spatio-temporal based deep residual neural network via categorized attentions (STDRN-HA) for human video event detection and classification. In [44], Q. Meng et al. proposed the Support Vector Machine SVM classification based novel features extraction method for event classification and detection, however, single SVM is the reason for the system's low accuracy. In [45], S. Sun et al. developed a guided optical flow feature extraction approach via Convolutional Neural Network (CNN) for human event detection. Here, limited and full-body features are the cause of low mean accuracy for event detection. E. Park et al. [46] proposed the feature amplification method using the Convolutional Neural Network (CNN) map for a handcrafted features and spatially fluctuating multiplicative fusion approach with Convolutional Neural Network (CNN)s for event detection and classification. D. Torpey et al. [47] designed a simple approach using local appearance and gesture features along with Convolutional Neural Network (CNN); while classification is accomplished using SVM, local and existing features are the main weak points of the system. Y. Zhu et al. [48] described a detailed system architecture for identifying events in human-based videos. By using deep learning and fusing trajectory analysis, the system comprises its computing cost and Graphical Processing Unit (GPU). L. Zhang [49] presented a novel two-level neural network-based learning approach for human video event classification. For the first level, Convolutional Neural Network CNNs are trained to provide information using video to an event, which understands the important content of the video. At the second level, they use a Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU)based technique to handle both temporal and spatial information. In [50], A. Nadeem proposed a robust structure approach that discovers multidimensional features along with body part representations; for human pose estimation they used quadratic discriminant analysis with extracted features and for classification they used a maximum entropy Markov technique. Table 10 shows the comparison of the proposed system with state-of-the-art methods. Overall results show that the proposed SEC approach produces the best classification accuracy rates.

Table 10. Event classification comparison with state-of-the-art methods.

Methods	UCF50	Methods	hmdb51	Methods	Olympic Sports
J. Uijlings [40]	81.8%	M. Jain et al [38]	52.10%	L. Zhang [49]	59.1%
F. Shi [39]	83.3%	H. Wang [41]	60.10%	S. Sun [45]	74.2%
Y. Zhu [48]	83.1%	D. Torpey [47]	62.80%	M. Jain et al. [38]	83.2%
D. Torpey [47]	86.4%	Y. Li [43]	70.69%	E. Park [46]	89.1%
L. Zhang [49]	88.0%	K. Hara [42]	70.20%	H. Wang [41]	89.6%
H. Wang [41]	89.1%	Y. Zhu [48]	76.30%	A. Nadeem [50]	88.26%
Q. Meng [44]	89.3%	A. Nadeem [50]	89.09%	—	—
Ours	90.48		89.21		90.83

5. Conclusions

Event classification and detection are two of the challenging tasks of the current era. In this research article, we proposed a novel technique for the classification of sustainable events. We proposed a pseudo-2D stick model along with full body and key points context-aware features. For classification, we used ray optimization for pre-classification and K-ary tree hashing for sustainable event classification achieving the mean accuracy rate of 90.48% for the UCF50 dataset, 89.21% for the hmdb51 dataset, and 90.83% for the Olympic sports dataset. In the future, we will develop the distributed classification of gait event detection and scene-aware intensity features.

Author Contributions: Conceptualization, I.A.; methodology, I.A. and A.J.; software, I.A.; validation, A.J.; formal analysis, K.K.; resources, A.J. and K.K.; writing—review and editing, A.J. and K.K.; funding acquisition, A.J. and K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. 2018R1D1A1A02085645).

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

References

1. Tzelepis, C.; Ma, Z.; Mezaris, V.; Ionescu, B.; Kompatsiaris, I.; Boato, G.; Yan, S. Event-based media processing and analysis: A survey of the literature. *I&VC* **2016**, *53*, 3–19.
2. Susan, S.; Agrawal, P.; Mittal, M.; Bansal, S. New shape descriptor in the context of edge continuity. *CAAI Trans.* Intell. Technol. **2019**, *4*, 101–109. [CrossRef]
3. Tingting, Y.; Junqian, W.; Lintai, W.; Yong, X. Three-stage network for age estimation. *CAAI Trans. Intell. Technol.* **2019**, *4*, 122–126. [CrossRef]
4. Zhu, C.; Miao, D. Influence of kernel clustering on an RBFN. *CAAI Trans. Intell. Technol.* **2019**, *4*, 255–260. [CrossRef]
5. Jalal, A.; Sharif, N.; Kim, J.T.; Kim, T.-S. Human activity recognition via recognized body parts of human depth silhouettes for residents monitoring services at smart homes. *Indoor Built Environment.* **2013**, *22*, 271–279. [CrossRef]
6. Jalal, A.; Kamal, S.; Kim, D. Individual Detection-Tracking-Recognition using depth activity images. In Proceedings of the 12th IEEE International Conference on Ubiquitous Robots and Ambient Intelligence, KINTEX, Goyang City, Korea, 28–30 October 2015; pp. 450–455.
7. Jalal, A.; Majid, A.; Quaid, K.; Hasan, A.S. Wearable Sensor-Based Human Behavior Understanding and Recognition in Daily Life for Smart Environments. In Proceedings of the IEEE Conference on International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2018; pp. 105–110.
8. Jalal, A.; Mahmood, M.; Sidduqi, M.A. Robust spatio-temporal features for human interaction recognition via artificial neural network. In Proceedings of the IEEE International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2018; pp. 218–223. [CrossRef]
9. Jalal, A.; Mahmood, M.; Hasan, A.S. Multi-features descriptors for human activity tracking and recognition in Indoor-outdoor environments. In Proceedings of the 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019; pp. 371–376. [CrossRef]
10. Jalal, A.; Nadeem, A.; Bobasu, S. Human body parts estimation and detection for physical sports movements. In Proceedings of the 2nd International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, Pakistan, 6–7 March 2019; pp. 104–109. [CrossRef]
11. Ahmed, A.; Jalal, A.; Rafique, A.A. Salient Segmentation based Object Detection and Recognition using Hybrid Genetic Transform. In Proceedings of the 2019 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 27–29 August 2019; pp. 203–208. [CrossRef]
12. Nadeem, A.; Jalal, A.; Kim, K. Human actions tracking and recognition based on body parts detection via Artificial neural network. In Proceedings of the 3rd International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 17–19 February 2020; pp. 1–6. [CrossRef]
13. Badar, S.; Jalal, A.; Batool, M. Wearable Sensors for Activity Analysis using SMO-based Random Forest over Smart home and Sports Datasets. In Proceedings of the 3rd International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 17–19 February 2020; pp. 1–6. [CrossRef]
14. Badar, S.; Jalal, A.; Kim, K. Wearable Inertial Sensors for Daily Activity Analysis Based on Adam Optimization and the Maximum Entropy Markov Model. *Entropy* **2020**, *22*, 1–19.
15. Rehman, M.A.; Raza, H.; Akhter, I. Security enhancement of hill cipher by using non-square matrix approach. In Proceedings of the 4th International Conference on Knowledge and Innovation in Engineering Science and Technology, Berlin, Germany, 21–23 December 2018. [CrossRef]
16. Wiens, T. Engine speed reduction for hydraulic machinery using predictive algorithms. *Int. J. Hydromech.* **2019**, *2*, 16–31. [CrossRef]
17. Li, Z.; Miao, Y.; Cen, X.-P.; Zhang, L.; Chen, S. Abnormal event detection in surveillance videos based on low-rank and compact coefficient dictionary learning. *Pattern Recognit.* **2020**, *108*, 107355. [CrossRef]

18. Einfalt, M.; Dampeyrou, C.; Zecha, D.; Lienhart, R. Frame-level event detection in athletics videos with pose-based convolutional sequence networks. In Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports, New York, NY, USA, 6–8 October 2019; pp. 42–50. [CrossRef]
19. Yu, J.; Lei, A.; Hu, Y. Soccer video event detection based on deep learning. In Proceedings of the International Conference on Multimedia Modeling, Thessaloniki, Greece, 8–11 January 2019; pp. 377–389. [CrossRef]
20. Franklin, R.J.; Dabbagol, V. Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks. In Proceedings of the 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 8–10 January 2020; pp. 632–637. [CrossRef]
21. Lohithashva, B.H.; Aradhya, V.N.M.; Guru, D.S. Violent video event detection based on integrated LBP and GLCM texture features. *Revue Intell. Artif.* **2020**, *34*, 179–187. [CrossRef]
22. Feng, Q.; Gao, C.; Wang, L.; Zhao, Y.; Song, T.; Li, Q. Spatio-temporal fall event detection in complex scenes using attention guided LSTM. *Pattern Recognit. Lett.* **2020**, *130*, 242–249. [CrossRef]
23. Rado, D.; Sankaran, A.; Plasek, J.; Nuckley, D.; Keefe, D.F. A Real-Time Physical Therapy Visualization Strategy to Improve Unsupervised Patient Rehabilitation. In Proceedings of the IEEE Transactions on Visualization and Computer Graphics, Atlantic City, NJ, USA, 11–16 October 2009; Volume 15, pp. 1–2.
24. Khan, M.H.; Zöller, M.; Farid, M.S.; Grzegorzek, M. Marker-Based Movement Analysis of Human Body Parts in Therapeutic Procedure. *Sensors* **2020**, *20*, 3312. [CrossRef] [PubMed]
25. Mokhlespour Esfahani, M.I.; Zobeiri, O.; Moshiri, B.; Narimani, R.; Mehravar, M.; Rashedi, E.; Parnianpour, M. Trunk Motion System (TMS) Using Printed Body Worn Sensor (BWS) via Data Fusion Approach. *Sensors* **2017**, *17*, 112. [CrossRef] [PubMed]
26. Golestani, N.; Moghaddam, M. Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks. *Nat. Commun.* **2020**, *11*, 1551. [CrossRef] [PubMed]
27. Jalal, A.; Kamal, S.; Kim, D. Depth Silhouettes Context: A new robust feature for human tracking and activity recognition based on embedded HMMs. In Proceedings of the 12th International Conference on Ubiquitous Robots and Ambient Intelligence, KINTEX, Goyang City, Korea, 28–30 October 2015; pp. 294–299.
28. Zhang, J.; Hu, J. Image segmentation based on 2D Otsu method with histogram analysis. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; pp. 105–108. [CrossRef]
29. Moschini, D.; Fusiello, A. Tracking human motion with multiple cameras using an articulated model. In Proceedings of the International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications, Rocquencourt, France, 4–6 May 2009; pp. 1–12. [CrossRef]
30. Li, H.; Lu, H.; Lin, Z.; Shen, X.; Price, B. Inner and inter label propagation: Salient object detection in the wild. *IEEE Trans. Image Process.* **2015**, *24*, 3176–3186. [CrossRef] [PubMed]
31. Jalal, A.; Kim, Y.H.; Kim, Y.J.; Kamal, S.; Kim, D. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recognit.* **2017**, *61*, 295–308. [CrossRef]
32. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray optimization. *Comput. Struct.* **2012**, *112*, 283–294. [CrossRef]
33. Wu, W.; Li, B.; Chen, L.; Zhu, X.; Zhang, C. K -Ary Tree Hashing for Fast Graph Classification. *IEEE Trans. Knowl. Data Eng.* **2017**, *30*, 936–949. [CrossRef]
34. Reddy, K.K.; Shah, M. Recognizing 50 human action categories of web videos. *Mach. Vis. Appl.* **2013**, *24*, 971–981. [CrossRef]
35. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2556–2563. [CrossRef]
36. Niebles, J.C.; Chen, C.W.; Fei-Fei, L. Modeling temporal structure of decomposable motion segments for activity classification. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 392–405.
37. Wang, L.; Zhang, Y.; Feng, J. On the Euclidean distance of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1334–1339. [CrossRef]
38. Jain, M.; Jegou, H.; Bouthemy, P. Better exploiting motion for better action recognition. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2555–2562. [CrossRef]

39. Shi, F.; Petriu, E.; Laganiere, R. Sampling strategies for real-time action recognition. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2595–2602. [CrossRef]
40. Uijlings, J.; Duta, I.C.; Sangineto, E.; Sebe, N. Video classification with densely extracted hog/hof/mbh features: An evaluation of the accuracy/computational efficiency trade-off. *Int. J. Multimed. Inf. Retr.* **2015**, *4*, 33–44. [CrossRef]
41. Wang, H.; Oneata, D.; Verbeek, J.; Schmid, C. A robust and efficient video representation for action recognition. *Int. J. Comput. Vis.* **2016**, *119*, 219–238. [CrossRef]
42. Hara, K.; Kataoka, H.; Satoh, Y. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6546–6555. [CrossRef]
43. Li, Y.; Liu, C.; Ji, Y.; Gong, S.; Xu, H. Spatio-Temporal Deep Residual Network with Hierarchical Attentions for Video Event Recognition. *ACM Trans. MCCA* **2020**, *16*, 1–21. [CrossRef]
44. Meng, Q.; Zhu, H.; Zhang, W.; Piao, X.; Zhang, A. Action Recognition Using Form and Motion Modalities. *ACM Trans. MCCA* **2020**, *16*, 1–16. [CrossRef]
45. Sun, S.; Kuang, Z.; Sheng, L.; Ouyang, W.; Zhang, W. Optical flow guided feature: A fast and robust motion representation for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1390–1399. [CrossRef]
46. Park, E.; Han, X.; Berg, T.L.; Berg, A.C. Combining multiple sources of knowledge in deep cnns for action recognition. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), New York, NY, USA, 7–9 March 2016; pp. 1–8. [CrossRef]
47. Torpey, D.; Celik, T. Human Action Recognition using Local Two-Stream Convolution Neural Network Features and Support Vector Machines. *arXiv* **2020**, arXiv:2002.09423. Available online: https://arxiv.org/abs/2002.09423 (accessed on 19 February 2020).
48. Zhu, Y.; Zhou, K.; Wang, M.; Zhao, Y.; Zhao, Z. A comprehensive solution for detecting events in complex surveillance videos. *Multimed. Tools. Appl.* **2019**, *78*, 817–838. [CrossRef]
49. Zhang, L.; Xiang, X. Video event classification based on two-stage neural network. *Multimed. Tools. Appl.* **2020**, 1–16. [CrossRef]
50. Nadeem, A.; Jalal, A.; Kim, K. Accurate Physical Activity Recognition using Multidimensional Features and Markov Model for Smart Health Fitness. *Symmetry* **2020**, *12*, 1766. [CrossRef]