applied sciences

MDPI

Article

A Graph-Based Approach to Recognizing Complex Human Object Interactions in Sequential Data

Yazeed Yasin Ghadi [1], Manahil Waheed [2], Munkhjargal Gochoo [3], Suliman A. Alsuhibany [4], Samia Allaoua Chelloug [5,*], Ahmad Jalal [2] and Jeongmin Park [6,*]

[1] Department of Computer Science and Software Engineering, Al Ain University, Al Ain 15551, United Arab Emirates; yazeed.ghadi@aau.ac.ae
[2] Department of Computer Science, Air University, Islamabad 44000, Pakistan; manahilwaheedar@gmail.com (M.W.); ahmadjalal@mail.au.edu.pk (A.J.)
[3] Department of Computer Science and Software Engineering, United Arab Emirates University, Al Ain 15551, United Arab Emirates; mgochoo@uaeu.ac.ae
[4] Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia; salsuhibany@qu.edu.sa
[5] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[6] Department of Computer Engineering, Tech University of Korea, 237 Sangidaehak-ro, Siheung-si 15073, Korea
* Correspondence: sachelloug@pnu.edu.sa (S.A.C.); jmpark@tukorea.ac.kr (J.P.)

Featured Application: The proposed methodology is a computer vision application for monitoring and recognizing human–object interactions and has been evaluated over three challenging benchmark datasets. Therefore, this technique can be used to develop advanced surveillance and security systems to locate human and object targets and classify their interactions.

check for updates

Citation: Ghadi, Y.Y.; Waheed, M.; Gochoo, M.; Alsuhibany, S.A.; Chelloug, S.A.; Jalal, A.; Park, J. A Graph-Based Approach to Recognizing Complex Human Object Interactions in Sequential Data. *Appl. Sci.* **2022**, *12*, 5196. https://doi.org/10.3390/app12105196

Academic Editor: Emanuele Carpanzano

Received: 18 April 2022
Accepted: 19 May 2022
Published: 20 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Abstract: The critical task of recognizing human–object interactions (HOI) finds its application in the domains of surveillance, security, healthcare, assisted living, rehabilitation, sports, and online learning. This has led to the development of various HOI recognition systems in the recent past. Thus, the purpose of this study is to develop a novel graph-based solution for this purpose. In particular, the proposed system takes sequential data as input and recognizes the HOI interaction being performed in it. That is, first of all, the system pre-processes the input data by adjusting the contrast and smoothing the incoming image frames. Then, it locates the human and object through image segmentation. Based on this, 12 key body parts are identified from the extracted human silhouette through a graph-based image skeletonization technique called image foresting transform (IFT). Then, three types of features are extracted: full-body feature, point-based features, and scene features. The next step involves optimizing the different features using isometric mapping (ISOMAP). Lastly, the optimized feature vector is fed to a graph convolution network (GCN) which performs the HOI classification. The performance of the proposed system was validated using three benchmark datasets, namely, Olympic Sports, MSR Daily Activity 3D, and D3D-HOI. The results showed that this model outperforms the existing state-of-the-art models by achieving a mean accuracy of 94.1% with the Olympic Sports, 93.2% with the MSR Daily Activity 3D, and 89.6% with the D3D-HOI datasets.

Keywords: dense trajectories; graph convolution network; human–object interaction; image foresting transform; image skeletonization

1. Introduction

Artificial intelligence (AI) has revolutionized the healthcare sector. From rehabilitation systems through assisted living programs to patient activity monitoring solutions, the advances in the field of human interaction recognition (HIR) have made numerous contributions to the field of medical engineering. By actively monitoring a person's actions, the

early symptoms of a serious disease can be detected in a timely fashion. Similarly, by recognizing human actions, their progress can be evaluated in rehabilitation programs. Such systems are usually complimented by the use of wearable and vision sensors. Therefore, the accurate recognition of human–object interactions is crucial for the success of a wide range of systems designed for smart homes [1–4], sports [5,6], healthcare [7,8], surveillance [9], counting and tracking [10–13], e-learning [14], and monitoring [15,16].

Despite the recent advances, human–object interaction recognition (HOIR) remains a challenging task because of various reasons, including illumination variation, background clutter, intra-class variation, inter-class similarities, scale variation, multiple viewpoints, and occlusion. Moreover, the quality of input data and its challenges depend on the medium of acquisition. Apart from regular static cameras, drone [17,18] and depth cameras [19,20] are also quite popular. Over the past few years, it has also been observed that researchers have employed various sensors, including body-worn [21,22] and inertial sensors [23], to record human activities and their interactions with the surrounding objects. Moreover, instead of relying only on RGB (red, green, blue) data, the use of depth [24,25] and RGB-D (red, green, blue, depth) [26] data is also common.

Therefore, this article proposes a robust HOIR system which takes sequential data as input and recognizes the HOI interaction being performed in it. In particular, firstly, the system pre-processes the input data by adjusting the contrast and smoothing the incoming image frames. Then, it locates the human and object through image segmentation. Based on this, 12 key body parts are identified from the extracted human silhouette through a graph-based image skeletonization technique called image foresting transform (IFT). After this, three types of features are extracted: full-body feature, point-based features, and scene features. Dense trajectories and local intensity order patterns (LIOP) are obtained from full-body silhouettes. Similarly, kinematic postures and local occupancy patterns (LOP) are obtained from the 12 key body points. Finally, spatial pyramid matching (SPM) and generalized search tree (GIST) descriptors are stored as scene features. The next step involves optimizing the six different features using isometric mapping (ISOMAP). Lastly, the optimized feature vector is fed to a graph convolution network (GCN), which performs the HOI classification.

The main contributions of this paper are as follows:

- We used a graph-based image skeletonization called the image foresting transform (IFT) technique to detect 12 human body parts.
- We proposed a multi-feature approach involving three different types of features: full-body features, point-based features, and scene features. We also provided types of feature descriptors for each category.
- We optimized the large feature vector obtained through isometric feature mapping (ISOMAP).
- We implemented a graph convolution network (GCN) and tuned its parameters for the final classification of human locomotion activities.

The rest of the article is arranged as follows: Section 2 investigates the related work in the field of HOI recognition. Section 3 explains the proposed method in great detail. Section 4 describes the different datasets that were used to validate the performance of the proposed method and the results of those experiments. Section 5 presents an analysis of the results achieved by the system and discusses the strengths and weaknesses of the system. Section 6 concludes the paper and describes the future plans.

2. Related Work

The past few years have witnessed an unprecedented increase in the number of researchers who have developed efficient human–object interaction recognition (HOIR) systems for improving healthcare. To establish a link between those systems and the method proposed in this paper, the recent research trends can be divided into two major categories: multi-feature HOIR and graph-based HOIR. These are explained in the following sections.

2.1. Multi-Feature HOIR

Combining multiple features extracted from the humans and objects involved in the interactions has shown high interaction recognition accuracies in the past [27–30]. Fang et al. [31] proposed a pairwise body-part attention model, which focused on crucial parts and their correlations for HOI recognition. They used visual geometry group (VGG) convolutional layers until the Conv5 layer for the extraction of full human features. Then, they used a pairwise body-part attention module to select discriminative body-part pairs and obtained their feature maps. Mallya et al. [32] used a simple network that fused features from a person bounding box and the whole image to recognize HOIs. The authors employed and compared both early and late fusion techniques. In early fusion, they concatenated the features and then applied dimensionality reduction. In late fusion, they reduced the dimensionalities of the two types of features and then concatenated them. Their results showed that the early fusion strategy gives better results.

Moreover, Yan et al. [33] proposed an HOI recognition system based on a multi-task neural network. They offered a digital glove called "WiseGlove" to detect hand motions. The system employed YOLO (you only look once) v3 to detect objects and a deep convolutional network in order to identify the interactions. For experimentation, the authors utilized both RGB and skeletal data to achieve a good recognition rate. However, the dataset only had eight action classes. Moreover, their system was only able to work with a few pre-defined objects. Gkioxari et al. [34] detected the human, verb, and object triplets by localizing humans through their appearance and objects through action-specific density. They used two RGB datasets to prove the validity of their system. Similarly, Li et al. [35] proposed a 3D pose estimation system and a new benchmark named "Ambiguous-HOI." They used 2D and 3D representation networks to mine features. Moreover, a cross-modal consistency task and joint learning structure were used to represent humans and objects. They performed extensive experiments on two datasets to prove the efficiency of their system.

2.2. Graph-Based HOIR

Many researchers have advocated the use of graph-based method for HOIR [36]. Xia et al. [37] constructed a fully connected graph with the detected humans and objects as nodes. The initial undirected graph was then pruned to obtain an HOI graph containing only those edges that connect human and object nodes. For robust feature extraction from the human and object nodes, the authors employed two different attention-based networks that modeled global and local contexts respectively. They used the V-COCO and HICO-DET datasets to prove that their system performed better than other state-of-the-art methods. These two datasets were also used by Yang et al. [38], who proposed a novel graph-based interactive reasoning model called interactive graph. Their approach exploited the interactive semantics among visual targets. Their model inferred HOIs using instance features and dynamically parsed pairwise interactive semantics among visual targets by integrating two-level in-Graphs, i.e., scene-wide and instance-wide in-Graphs. The proposed framework was end-to-end trainable and free from costly annotations, such as human pose. Although these two methods proved their effectiveness on image datasets, Sunkesula et al. [39] used video datasets too. They proposed a hybrid approach that uses GCN and hierarchical recurrent neural networks (RNNs), for recognizing human–object interactions in videos. Their approach did not rely on hand-crafted features. Instead, they used pure visual features derived from a re-trainable off-the-shelf network to represent the inputs.

Qi et al. [40] detected and recognized human–object interactions in images and videos using the graph parsing neural network (GPNN). For a given scene, their GPNN model inferred a parse graph that included the HOI graph structure represented by an adjacency matrix and the node labels. Within a message passing inference framework, the proposed GPNN iteratively computed the adjacency matrices and node labels. Liu et al. [41] used the few-shot learning (FSL) approach for HOIR, which means using only a few samples to perform the task. However, it is difficult to do so and therefore, the traditional FSL methods do not perform well in complex HOI scenes. Thus, the authors proposed the

use of dynamic graph-in-graph networks (DGIG-Net) for achieving better results. They built a knowledge reconstruction graph to learn the latent representations for different HOI categories and a dynamic relation graph which integrated both constructible visual nodes and dynamic task-oriented semantic information.

3. Materials and Methods

The proposed method takes sequential data as input. Videos containing a wide range of human–object interactions from three large datasets are fed into the system. The videos are converted into image frames which are pre-processed. After enhancing the images, the human and object pair in each one of them is localized using an image segmentation technique. Then, the key human body parts are identified using a key point detection algorithm. Next, three kinds of features are obtained: full-body features, point-based features, and scene features. The multiple features are combined to form a feature vector which is optimized and fed into a classifier. The details of each step of the process are discussed in the following subsections. The general architecture of the proposed method is shown in Figure 1.

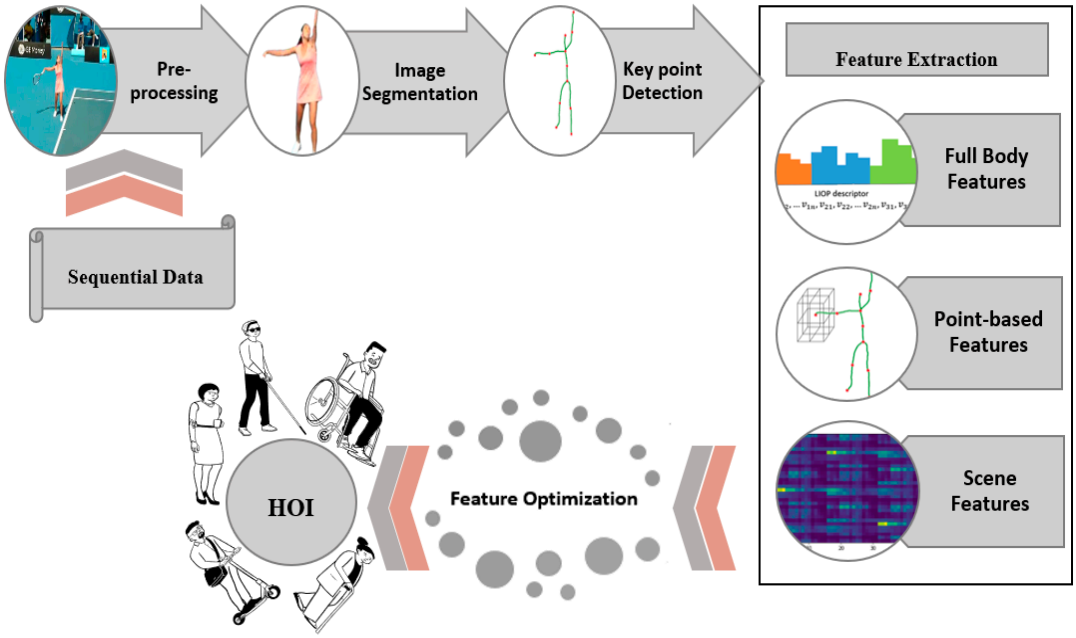


Figure 1. The architecture of the proposed HOI recognition system.

3.1. Pre-Processing

Most of the input videos used for experimentation contain fast camera motions, illumination variation, and noise. Hence, all images are pre-processed first. To enhance the intensity values of the image pixels, sigmoid stretching is used. Then, for removing the noise, Gaussian filtering is employed. Figure 2a represents an original image frame, and Figure 2b shows the image after performing sigmoid stretching on it. Since it is often difficult to analyze the differences from the naked eye, the histograms of both images are also shown in Figure 2d,e. Figure 2c shows the denoised image after applying Gaussian filtering, and Figure 2f shows that the pixel values in the resultant image are closer to the mean.

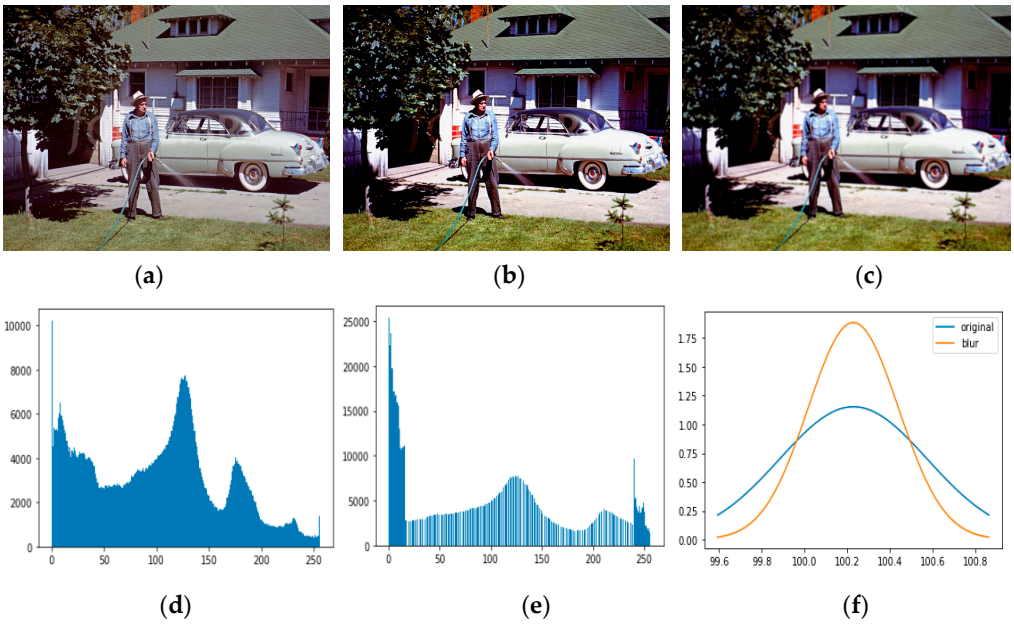


Figure 2. Results of image pre-processing (**a**) original image (**b**) sigmoid stretched image (**c**) filtered image (**d**) histogram of original image (**e**) histogram of sigmoid stretched image (**f**) mean and standard deviation plot of filtered image.

3.1.1. Sigmoid Stretching

Sigmoid stretching is a linear image transformation technique that depends upon piecewise linear functions. It is considered as an image enhancement technique that attempts to improve the contrast by stretching the intensity values of an image to fill the entire dynamic range. The transformation function used is always linear and monotonically increasing. The sigmoid contrast stretching technique highlights moderate pixel values in the images while maintaining sufficient contrast at the extremes. It places all of the pixel values along a sigmoidal function (an S-shaped curve). The result of this is less contrast in very bright and very dark areas, and more contrast in areas between these extremes. This is an ideal stretch for almost any image and performs very well when there are clouds and water in the image. Equation (1) shows the sigmoid function that allows the input pixel values x to be stretched using the sigmoid function.

$$Sigmoid\ (x) = \frac{1}{(1 + e^{-x})} \tag{1}$$

3.1.2. Gaussian Filtering

The Gaussian filter is a filter whose impulse response is a Gaussian function. The Gaussian smoothing operator is a 2D convolution operator that is used to make the images smooth by removing noise. Since some of the details are also removed in the process, the resultant image appears to be blurred. It outputs a weighted average of each pixel's neighborhood, with the average weighted more toward the value of the central pixels. In this sense, it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian ('bell-shaped') hump. Equation (2) represents the resultant smooth image $G(x, y)$ after applying a Gaussian filter on it, where x and y represent the x and y coordinate values of the 2D image and σ is the standard deviation.

$$G(x, y) = \frac{1}{2\pi\sigma^2}\left(e^{-\frac{x^2+y^2}{2\sigma^2}}\right) \tag{2}$$

3.2. Image Segmentation

After pre-processing the images, image segmentation is applied to locate the desired human–object pair. Image segmentation means dividing an image into multiple segments. These segments are also known as super-pixels. Hence, the entire image is divided into super-pixels using the quick shift algorithm [42]. It is a fast mode seeking algorithm, similar to mean shift. The algorithm segments an RGB image (or any image with more than one channel) by identifying clusters of pixels in the joint spatial and color dimensions. Segments are local and can be utilized for further processing. Given an image, the quick shift algorithm generates a forest of pixels including branches that are labeled with a distance value. The output represents a hierarchical segmentation of the image, where segments corresponding to subtrees. Useful super-pixels can be identified by cutting the branches whose distance label is above a given threshold. This is shown in Figure 3.

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

Figure 3. Results of image segmentation. (**a**) Original image, (**b**) super-pixels obtained using quick shift, (**c**) merged super-pixels, (**d**) extracted human–object pair.

As shown in Figure 3, the quick shift algorithm divides the given image into super-pixels. To locate the desired human–object pair, a super-pixel merging technique similar to the one proposed by Xu et al. [43] is used. According to this technique, similar and adjacent super-pixels are merged to form bigger super-pixels on the basis of similarity until the desired number of super-pixels is obtained. Four types of features are extracted from each super-pixel: mean, covariance, SIFT (scale-invariant feature transform), and SURF (speeded-up robust features). These four values are combined to obtain the feature vector of a super-pixel. If the similarity between the feature vectors of any two adjacent super-pixels is above a set threshold, the two super-pixels are merged into one big super-pixel. The process continues until three super-pixels are left, i.e., the background, the human, and the object. The super-pixel with the largest area is considered the background and removed to obtain the desired silhouette.

3.3. Key Point Detection

After extracting the human silhouette, 12 key human body parts are identified. For this purpose, the first step is to convert the human silhouette into a binary silhouette whose image skeleton is then obtained. In the binary image, the foreground is black and the background is white. The process of skeletonization keeps reducing the foreground until no more pixels can be removed. The skeletal remnant is then used to identify key points. For image skeletonization, a graph-based technique called the image foresting transform (IFT) [44] is used. The IFT defines a minimum-cost path forest in a graph, whose nodes are the image pixels and whose arcs are defined by an adjacency relation between pixels. The cost of a path in this graph is determined by an application-specific path-cost function, which usually depends on local image properties along the path—such as color, gradient, and pixel position. The roots of the forest are drawn from a given set of seed pixels. For suitable path-cost functions, the IFT assigns one minimum-cost path from the seed set to each pixel in such a way that the union of those paths is an oriented forest, spanning the whole image. The IFT outputs three attributes for each pixel: an optimum path from the root, the cost of that path, and the corresponding root. The path attribute can be used to find an image skeleton.

Seven key points are obtained from the nodes marking the start and end positions of various paths in the obtained skeleton. These points are identified as the head, left hand, right hand, upper torso, bottom torso, left foot and right foot. Using the obtained seven points, five additional key points are also found, namely, the neck, left elbow, right elbow, left knee, and right knee. The method of finding these additional points is simple: the mid-point of any two key points is calculated, and a point on the contour lying closest to the obtained mid-point is stored as an additional key point. The mid-point (x_m, y_m) of two existing points j and k is calculated using Equation (3). Each step of the process is shown in Figure 4.

$$(x_m, y_m) = \left(\frac{x_j + x_k}{2}, \frac{y_j + y_k}{2} \right) \tag{3}$$

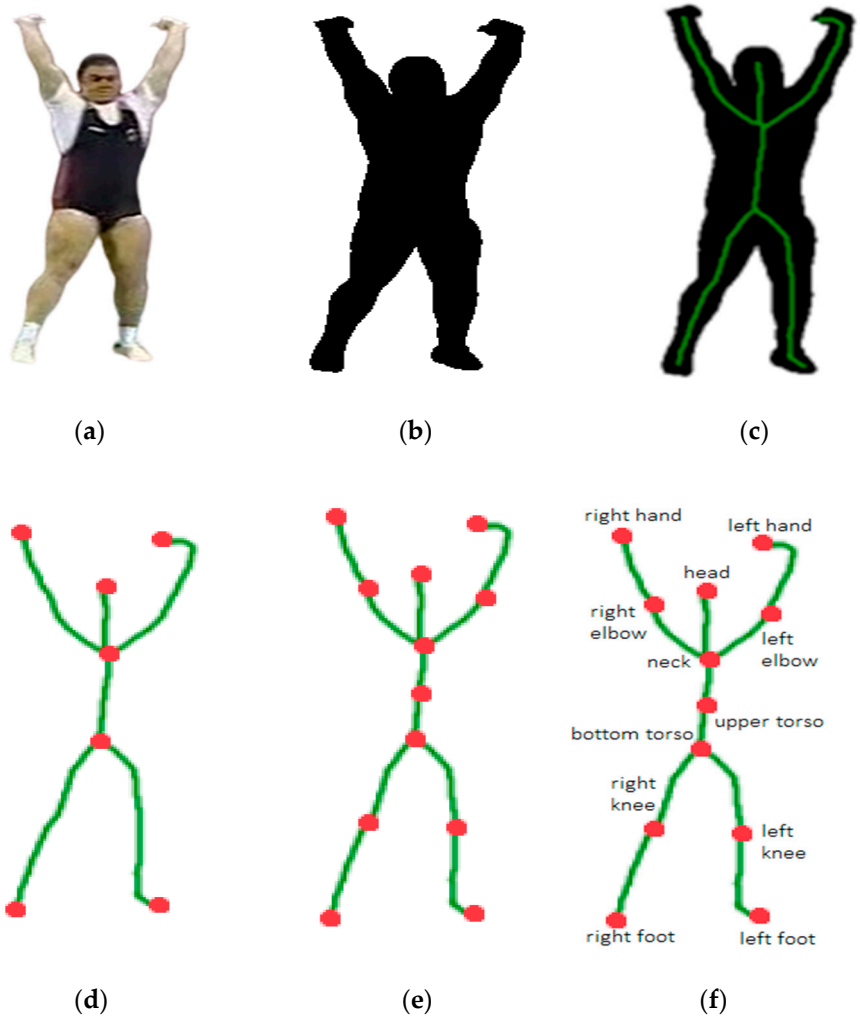


Figure 4. Results of key point detection. (**a**) Full body silhouette, (**b**) binary silhouette, (**c**) IFT graph, (**d**) graph nodes, (**e**) additional nodes, (**f**) 12 key body points.

3.4. Feature Extraction

Robust features play a critical role in identifying an HOI interaction. Therefore, three different types of features are employed by the proposed system: full-body, point-based, and scene features. Two different feature descriptors are obtained for each type. All six features are explained in detail in the following subsections, and a pseudocode for this entire process is given in Algorithm 1.

Algorithm 1: Feature Extraction

Input: N: full images, full body silhouettes, and 12 key body points
Output: combined feature vector (*f1*, *f2*, *f3* . . . *fn*)
% initiating feature vector for HOIR %
feature_vector ← []
F_vectorsize ← GetVectorsize ()
% loop on all images%
For *i* = 1:*n*
% extracting scene features%
SPM ← *GetSPM*(*i*)
Gist ← *GetGist*(*i*)
FeatureVector.append(SPM, Gist)
% loop on extracted human silhouettes %
J ← len (silhouettes)
For *i* = 1:J
% extracting full body features%
Trajectory ← GetTrajectory(*silhouette*[*i*]))
LIOP ← GetLIOP(*silhouette*[*i*]))
FeatureVector.append(Trajectory, LIOP)
% loop on 12 key points of each silhouette%
For *i* = 1:12
% extracting key point features%
KinematicPosture ← GetKinematicPosture(*i*)
LOP ← GetLOP(*i*)
FeatureVector.append(LOP, KinematicPosture)
End
End
End
Feature-vector ← Normalize (FeatureVector)
return feature vector (*f1*, *f2*, *f3* . . . *fn*)

3.4.1. Full Body Feature: Dense Trajectory

For full-body silhouettes, dense trajectories are obtained, as they are robust to abrupt motion [45]. Every point $P_t = (x_t, y_t)$ on the silhouette is tracked from frame I_t until the next frame I_{t+1} by computing its dense optical flow field $\omega_t = (u_t, v_t)$. Points of subsequent frames are concatenated to form trajectories: $(P_t, P_{t+1}, \ldots)$. For each frame, if no tracked point is found in a $W \times W$ neighborhood, a new point is sampled and added to the tracking process so that a dense coverage of trajectories is ensured. A sample of the dense trajectories is shown in Figure 5. Given a trajectory of length L, which is set to 15 as in [45], its shape is described by a sequence S of displacement vectors ΔP_t, given in Equation (4), where each displacement vector is calculated using Equation (5). The resulting 30-dimensional vector for each point is normalized by the sum of displacement vector magnitudes as shown in Equation (6).

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

Figure 5. Dense trajectories. (**a**) Trajectory at frame 310, (**b**) trajectory at frame 430, (**c**) trajectory at frame 700.

$$S = (\Delta P_t, \ldots \Delta P_{t+L-1}) \tag{4}$$

$$\Delta Pt = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t) \tag{5}$$

$$T = \frac{(\Delta P_t \ldots \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} ||\Delta P_j||} \tag{6}$$

3.4.2. Full Body Feature: LIOP

The local intensity order pattern (LIOP) feature [46] performs better in the case of low contrast and illumination changes within an image but not very well in the case of rotation and scale variation. It is also robust to many other geometric and photometric transformations, such as view-point change, image blur, and JPEG compression. It is a novel method for feature description based on intensity order. Firstly, the overall intensity order is used to divide the local patch into sub regions called ordinal bins. Next, the LIOP descriptor of each point is defined based on the relationships among the intensities of its neighboring sample points. If $P(x)$ is a vector consisting of the intensities of the neighboring sample points of a point x in the local patch, the LIOP of the point x can be defined using Equation (7), where $\omega(x)$ is a weighting function described in Equation (8). The LIOP descriptor is constructed by concatenating the LIOPs of points in each ordinal bin, respectively. As in [46], each image is divided into six ordinal bins, and the number of neighboring sample points is set to four, resulting in a feature vector of size $4! \times 6 = 144$. Each step of this process is shown in Figure 6, and Equation (9) represents the LIOP descriptor of each bin.

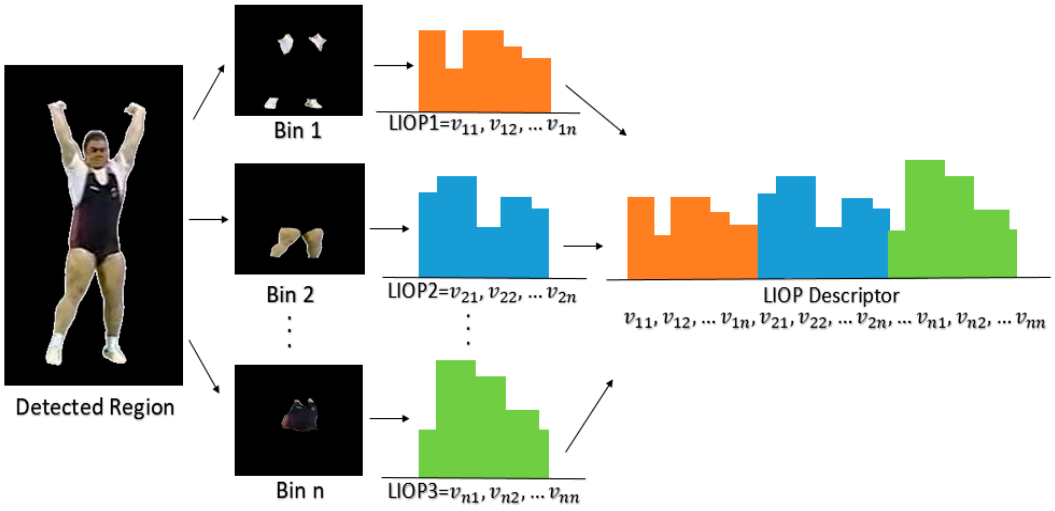


Figure 6. LIOP feature descriptor.

$$\omega(x) = \sum_{i,j} sgn\Big(\big|I(x_i) - I(x_j)\big| - T_{lp}\Big) + 1 \tag{7}$$

$$LIOP(x) = \varnothing(\gamma(P(x))) \tag{8}$$

$$des_i = \sum_{x \in bin_i} \omega(x) LIOP(x) \tag{9}$$

3.4.3. Point-Based Feature: Kinematic Posture

Kinematic posture involves the extraction of two feature sets, namely the linear joint position feature (LJPF) and angular joint position feature (AJPF) [47]. Every joint i is represented by a three-dimensional vector J_i in the coordinate space of Kinect. The distance of each joint with respect to the head joint J_{head} is obtained. This distance $d_{i(head)}$ is then

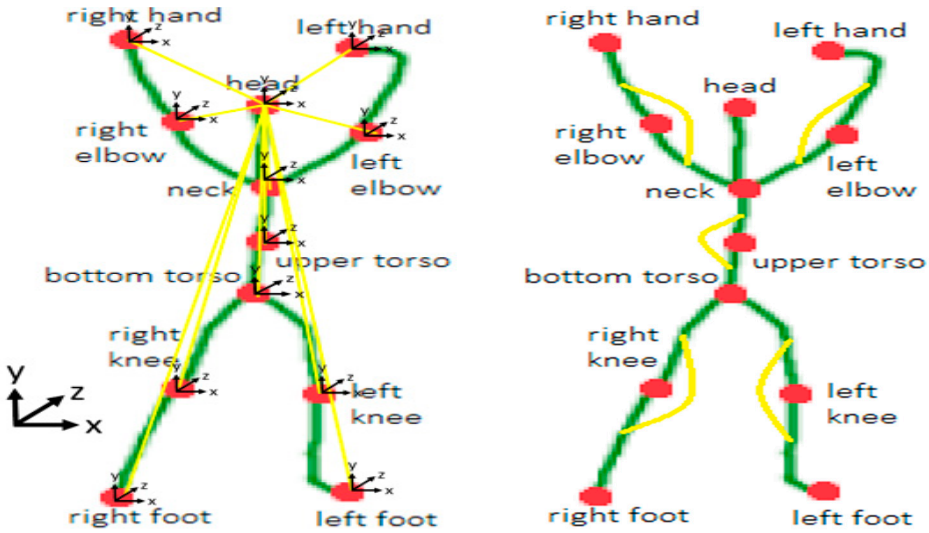
normalized with respect to the distance vector between the neck joint and torso joint. Hence, for 12 joints, the LJPF for each frame n can be represented by Equation (10).

$$LJPF_n = \left[d_{[n,1]}, d_{[n,2]}, \ldots d_{[n,12]}\right] \tag{10}$$

Next, the angles between different bone segments are calculated using three joints. The AJPF encodes the angles between different bone segments. For example, the angle between the left upper arm and forearm is calculated using the neck, left elbow, and left-hand joints. Since the angle between the neck and the head is almost constant for all actions, only five angles are computed. Hence, the AJPF for each frame n can be represented by Equation (11).

$$AJPF_n = \left[a_{[n,1]}, a_{[n,2]}, \ldots a_{[n,5]}\right] \tag{11}$$

Finally, for each video frame, these two features are combined to generate the kinematics posture feature (KPF) set. This feature encodes the change in joint position and angles across video frames. Both LJPF and AJPF features of an image skeleton are shown in Figure 7.



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

Figure 7. Kinematic posture. (**a**) Linear joint position feature (LJPF), (**b**) angular joint position feature (AJPF).

3.4.4. Point-Based Feature: LOP

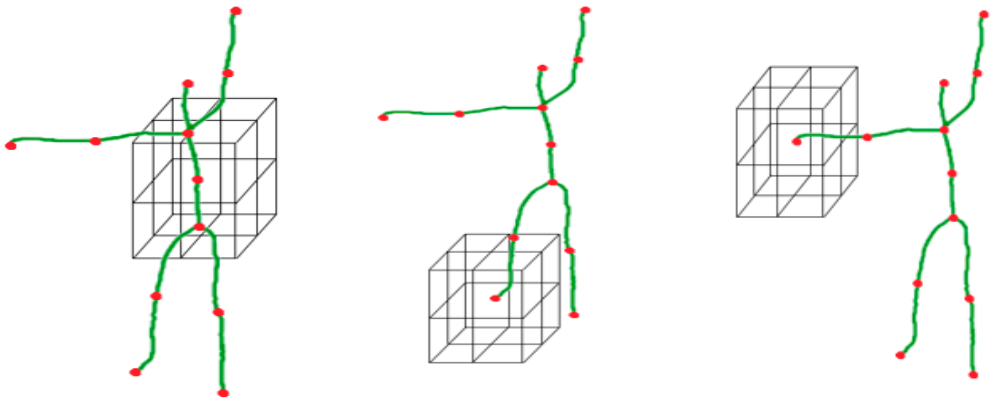
The local occupancy pattern (LOP) [48] of each key body part is obtained as a point-based feature. It measures the space occupied by an object around a certain body part. For example, when a person picks up a glass, the space around his/her hand is occupied by that glass. Then if the person drinks from that glass, the space around both the hand and the head is occupied by the object. This occupancy information can be useful for identifying the interaction of drinking from other HOIs. For each joint, the local region of size (72, 72, 320) around it is partitioned into $12 \times 12 \times 4$ bins, and the size of each bin is (6, 6, 80). During each frame, the number of points that fall into each bin bin_{xyz} is counted, and a sigmoid normalization function is applied to obtain the feature O_{xyz} for this bin, as shown in Equation (12):

$$O_{xyz} = \delta\left(\sum_{q \epsilon bin_{xyz}} I_q\right) \tag{12}$$

The LOP feature of a joint i is a vector consisting of the feature O_{xyz} of all the bins in the spatial grid around the joint. The value of I_q is 1 if there is a point at location q and

0 otherwise. $\delta(.)$ represents sigmoid normalization. For 12 joints, the size of LOP feature descriptor across each frame will thus be $12 \times 12 \times 4 \times 12 = 6912$.

Both LOP features of an image skeleton are shown in Figure 8.



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

Figure 8. LOP feature (**a**) at 'upper torso' joint, (**b**) at 'right foot' joint, (**c**) at 'right hand' joint.

3.4.5. Scene Feature: SPM

Spatial pyramid matching (SPM) [49] is a solution to the bag-of-words (BOW) model which represents an image as an order-less collection of local features but discards the spatial relationships of local descriptors, which severely limits its descriptive power. In SPM, an image is partitioned into increasingly finer regions, and features are evaluated in the local regions. This is done in three levels in which the image is divided into 1, 4, and 16 regions respectively. Then local SIFT descriptors are extracted from the small image regions, and they are coded using a dictionary (learned using features from several training images). The code vectors in each spatial region are then pooled together by building histograms. Finally, the histograms of the different spatial regions are concatenated to give the complete SPM feature vector of an image frame. Each step of this process is shown in Figure 9.

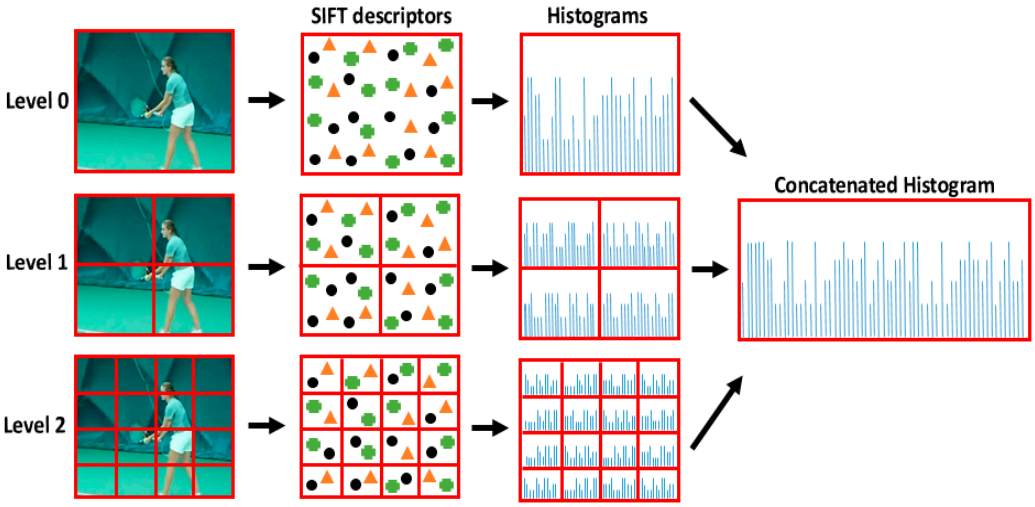


Figure 9. SPM feature descriptor.

3.4.6. Scene Feature: GIST

The GIST [50] feature descriptor is a global texture-based feature extraction technique which is used for extracting the dominant spatial structural of a scene. This low-level representation is made using a set of five perceptual dimensions, i.e., naturalness, openness, roughness, expansion, and ruggedness. Initially, the input frames are converted into gray-scale images. To obtain the GIST descriptor of an image frame, it is first convolved with 32

Gabor filters at 4 scales (σ) and 8 orientations (θ), resulting in a series of 32 feature maps of the same size as the input image frame. Each feature map is divided into 9 regions, and then the values within each region are averaged. These 9 values of the 32 feature maps are then concatenated to give the 288-dimensional GIST feature vector for each frame. The GIST descriptors of two different scenes are visualized in Figure 10.

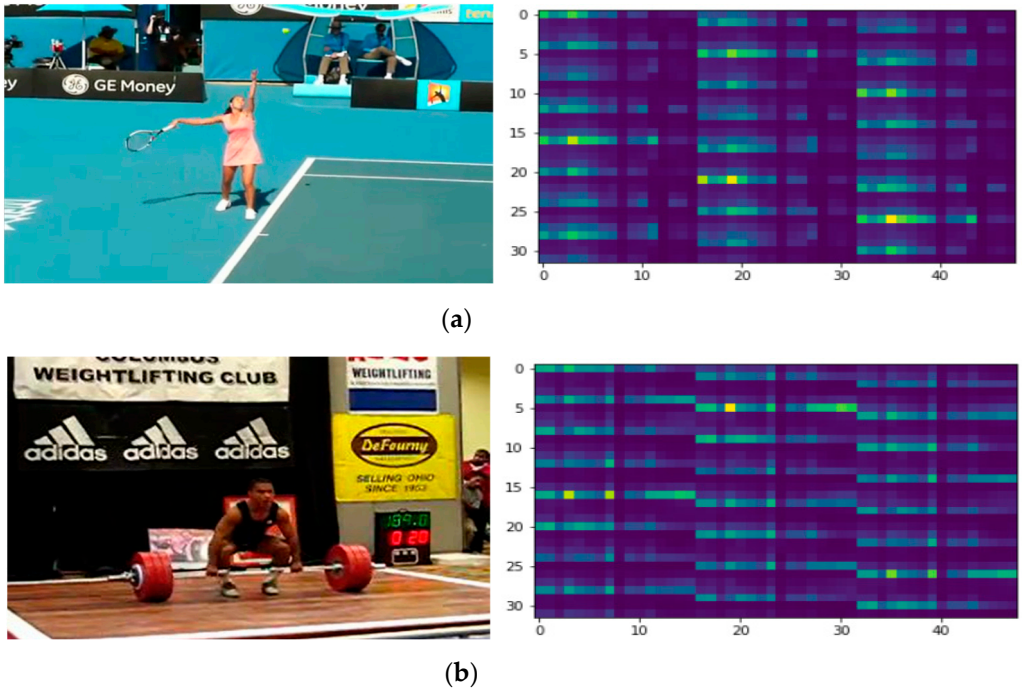
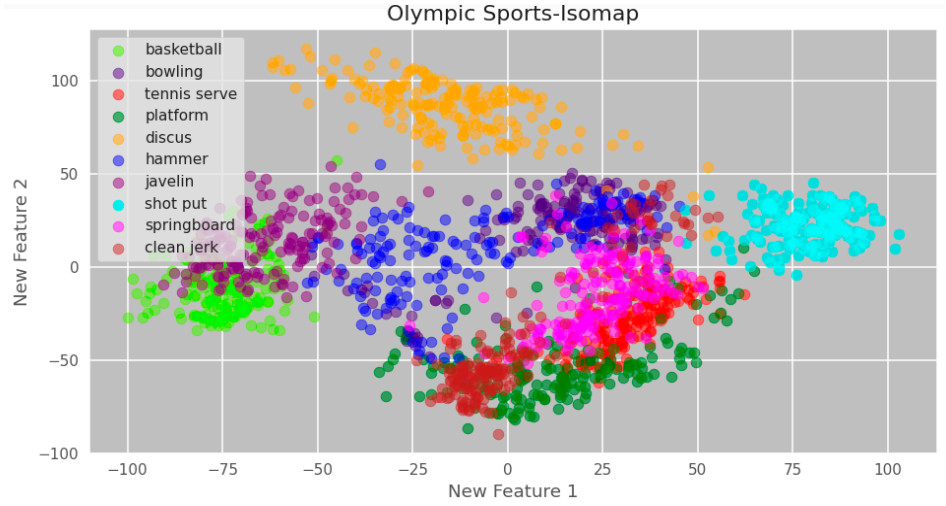


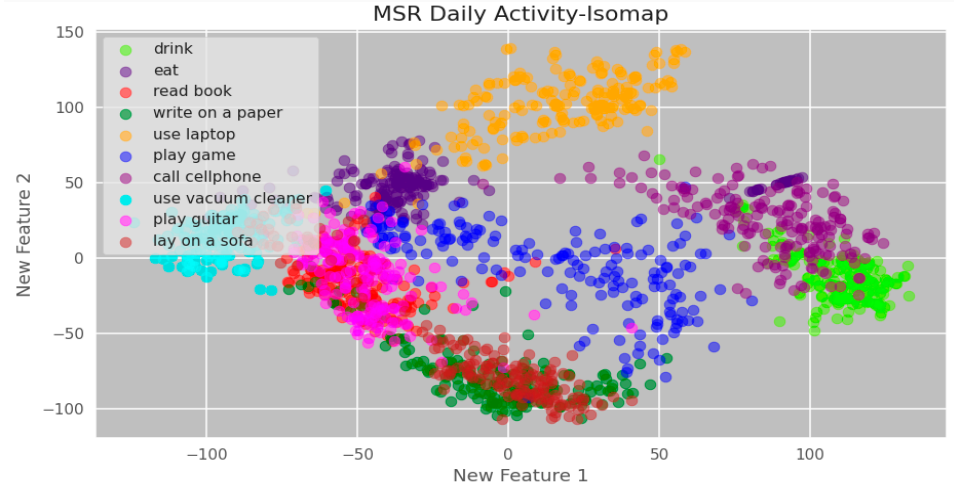
Figure 10. GIST descriptor. (**a**) An outdoor (tennis court) scene (left) and its GIST descriptor (right), (**b**) an indoor (gymnasium) scene and its GIST descriptor.

3.5. Feature Optimization

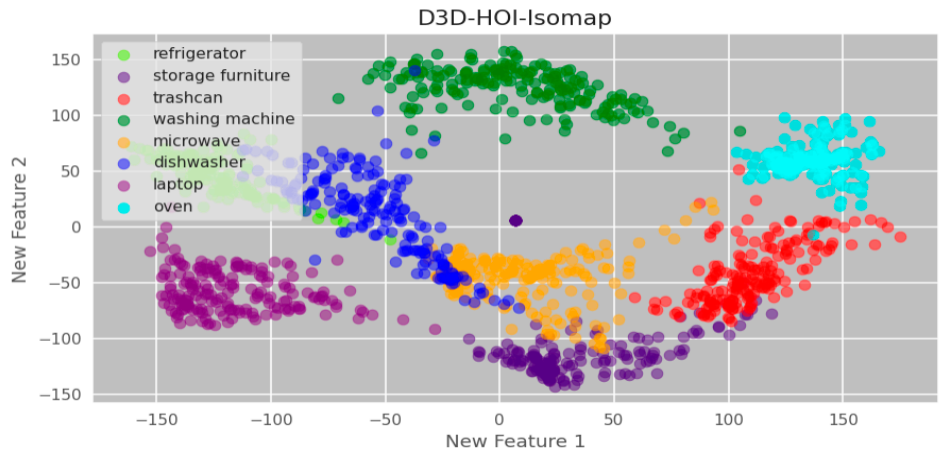
After obtaining the six different features, they are concatenated to obtain a feature vector which is very high dimensional. The size of the dense trajectory feature descriptor is 50 × 30 or 1 × 1500 and that of the LIOP feature is 1 × 144. The size of the kinematic posture feature is 1 × 17 and that of LOP feature is 1 × 6912 for the 12 joints. Lastly, the size of the SPM feature descriptor is 1 × 128, and that of the GIST descriptor is 1 × 512 for each image frame. Therefore, the combined feature vector is of the size 1 × 9213 for each input image. To reduce the dimensions, the isometric feature mapping (ISOMAP) [51] technique is used. It is a nonlinear dimensionality reduction method for estimating the intrinsic geometry of a data manifold based on a rough estimate of each data point's neighbors on the manifold. It extends metric multidimensional scaling (MDS) by incorporating the geodesic distances imposed by a weighted graph. The Isomap defines the geodesic distance to be the sum of edge weights along the shortest path between two nodes (computed using Dijkstra's algorithm, for example). The top n eigenvectors of the geodesic distance matrix, represent the coordinates in the new n-dimensional Euclidean space. The Isomaps of all three datasets are shown in Figure 11.



(**a**)



(**b**)



(**c**)

Figure 11. Isomaps of different datasets. (**a**) Olympics Sports, (**b**) MSR Daily Activity 3D, (**c**) D3D-HOI.

3.6. Human-Object Interaction Recognition (HOIR)

The final step of the proposed system is the classification of interactions that is performed by a graph convolution network (GCN) [52]. GCNs are a very powerful neural network architectures for machine learning on graphs. In fact, they are so powerful that even a randomly initiated 2-layer GCN can produce useful feature representations of nodes in networks. The relative nearness of nodes in the network is preserved in the 2-dimensional representation, even without any training. The classic method to perform image classification is using convolutional neural networks (CNN). Images are represented in the form of pixels, and the CNN runs sliding kernels (or filters) across the images; the model subsequently learns important features by looking at the adjacent pixels. GCNs, on the other hand, view images as complete graphs. Each node represents each pixel. Node feature represents the pixel value. Edge feature represents the Euclidean distance between each pixel. The closer two pixels are to each other, the larger the edge values. Figure 12 shows a general architecture of a GCN.

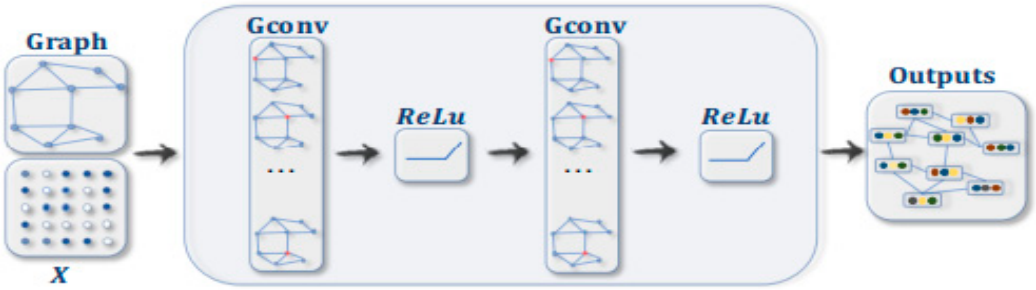


Figure 12. A general architecture of a GCN.

The feature representation at any layer in a neural network can be represented by Equation (13):

$$H_i = \sigma(W_{i-1}H_{i-1} + b_{i-1}) \tag{13}$$

where $\boldsymbol{H_i}$ is the feature representation at layer $\boldsymbol{i}$, $\boldsymbol{\sigma}$ is the activation function, $\boldsymbol{W_{i-1}}$ represents the weight matrix at layer $\boldsymbol{i-1}$, $\boldsymbol{H_{i-1}}$ is the feature representation at layer $\boldsymbol{i-1}$, and $\boldsymbol{b_{i-1}}$ is the bias at layer $\boldsymbol{i-1}$. For the forward propagation equation of a GCN, the adjacency matrix $\boldsymbol{A}$ is also taken into account, as shown in Equation (14):

$$\boldsymbol{H_i = \sigma(W_{i-1}H_{i-1}A^*)} \tag{14}$$

where $\boldsymbol{A^*}$ is the normalized adjacency matrix. The GCN architecture used in this research consists of two GCN layers with output dimensionalities of 1024 and 2048, respectively. For both layers, the Leaky ReLU activation function with the negative slope of 0.2 is used. Equation (15) represents the Leaky ReLU activation function, where $\boldsymbol{x}$ represents the input vector. It is a non-linear activation function which leads to faster convergence in experiments. Moreover, each convolutional layer is followed by a max pooling layer. The output from the final pooling layer is flattened and entered into a fully convolutional (FC) layer which is followed by a softmax layer, as represented by Equation (16), where $\boldsymbol{x_j}$ represents the input vector. Equation (17) represents the output of this model. During training, the input images are cropped and resized into 448×448. For network optimization, stochastic gradient descent (SGD) is used as the optimizer. The momentum is set to be 0.9 and weight decay is 10^{-4}. The initial learning rate is 0.01, which is decayed by a factor of 10 for every 40 epochs, and the network is trained for 100 epochs in total.

$$\boldsymbol{LeakyReLU}(x) = \begin{cases} \boldsymbol{0.01x}, & \boldsymbol{x < 0} \\ \boldsymbol{x}, & \boldsymbol{x \geq 0} \end{cases} \tag{15}$$

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{16}$$

$$Z = softmax(A^* LeakyReLU(W_{i-1}H_{i-1}A^*)W_i) \tag{17}$$

4. Experimental Results

This section describes the three publicly available datasets that are used to validate the proposed system. The description is followed by the implementation details and the results of different experiments performed on the three datasets. The GCN is used for classification, and the proposed system is evaluated using the leave one subject out (LOSO) cross-validation technique. In this technique, each subject is used once as the test set. It is a special type of k-fold cross validation, in which the number of folds is equal to the number of instances in the dataset. Then, the efficiency of the proposed key point detection algorithm is analyzed by comparing the detected body parts with ground truth values. All the processing and experiments are performed using Python on a Windows-10 operating system with 16 GB RAM, and a processor of core-i7-7500U CPU @ 2.70 GHz. Finally, the performance of the proposed system is compared with the accuracies of other state-of-the-art systems tested on these datasets.

4.1. Datasets Description

The Olympic Sports dataset [53] consists of RGB sequences only. It contains sports videos from YouTube in which athletes perform a total of 16 activities: high-jump, long-jump, triple-jump, pole-vault, basketball, bowling, tennis-serve, platform, discus, hammer, javelin, shot put, spring board, snatch, clean jerk, and vault. A total of 783 videos are available. However, only 10 of the activities are human–object interactions which involves humans performing an activity on or using a certain object. Therefore, only 480 videos are used. Moreover, this dataset poses the challenge of illumination variation and motion blur.

The MSR Daily Activity 3D dataset [54] consists of both depth and RGB sequences. It was recorded using a Kinect sensor at Microsoft Research Redmond. Ten different subjects perform a total of sixteen daily activities: drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up, sit down. Each subject performs the same action twice, once while standing and then while sitting. Hence, a total of 320 videos are available. However, only 10 of the activities are human–object interactions which involves humans performing an activity on or using a certain object. Therefore, only 200 videos are used. Moreover, this dataset poses the challenge of very high intra-class variation.

The D3D HOI dataset [55] consists of RGB sequences. The authors used human–object relations to improve the 3D reconstruction process of the human poses and objects. However, in this article, the dataset is only used to classify human–object interactions. It includes eight classes, named after the objects involved in the interactions: refrigerator, storage furniture, trashcan, washing machine, microwave, dishwasher, laptop, and oven. Each subject opens and closes all eight objects, once while standing and then while sitting. Hence, a total of 256 videos are available. Moreover, this dataset poses the challenge of viewpoint variation. The respective action classes for each dataset are provided in Table 1.

Table 1. Human–object interactions included in all three datasets.

Serial No.	Olympic Sports	MSR Daily Activity 3D	D3D-HOI
1	basketball	drink	refrigerator
2	bowling	eat	storage furniture
3	tennis-serve	read book	trashcan
4	platform	write on a paper	washing machine
5	discus	use laptop	microwave
6	hammer	play game	dishwasher
7	javelin	call cellphone	laptop
8	shot put	use vacuum cleaner	oven
9	spring board	play guitar	-
10	clean jerk	lay on a sofa	-

4.2. Experiment I: HOI Classification Accuracies

The interaction recognition accuracies achieved for various classes of the three datasets are represented in the form of confusion matrices. A confusion matrix summarizes the performance of a classifier in terms of true and false positives and negative. The number of true positives, that is, accurately identified classes, is displayed on the diagonal of the matrix. Figures 13–15 show the confusion matrices of the Olympics Sports, MSR Daily Activity 3D, and D3D HOI datasets, respectively.

The confusion matrices in Figures 13–15 show that although most of the interaction classes are predicted accurately, a few similar interaction classes are still confused with each other. More importantly, one or more interactions involving the same objects (for example, calling on a cellphone or playing game on a cellphone) or similar objects (for example, opening or closing a washing machine or a microwave) are more likely to be confused with each other.

	BB	BL	TS	PT	DS	HR	JV	SP	SB	CJ
BB	**0.97**	0.01	0	0	0.01	0	0	0.01	0	0
BL	0.02	**0.95**	0	0	0.01	0.01	0	0.01	0	0
TS	0	0	**0.96**	0	0.01	0.01	0.02	0	0	0
PT	0	0	0.01	**0.95**	0	0	0	0	0.03	0.01
DS	0.01	0	0	0	**0.94**	0.02	0.01	0.02	0	0
HR	0.01	0	0	0	0.03	**0.93**	0.02	0.01	0	0
JV	0	0	0.01	0	0.01	0.02	**0.94**	0.02	0	0
SP	0.01	0.02	0	0	0.02	0.02	0.01	**0.92**	0	0
SB	0	0.01	0.02	0.03	0	0	0	0	**0.92**	0.02
CJ	0.03	0.02	0	0.01	0	0	0	0	0.01	**0.93**

Figure 13. Confusion matrix of Olympic Sports dataset, BB = basketball, BL = bowling, TS = tennis-serve, PT = platform, DS = discus, HR = hammer, JV = javelin, SP = shot put, SB = spring board, CJ = clean jerk.

	DR	ET	RB	WP	UL	PG	CC	UV	PR	LS
DR	**0.93**	0.04	0.01	0	0	0.01	0.01	0	0	0
ET	0.04	**0.93**	0.01	0	0	0.01	0.01	0	0	0
RB	0.01	0.01	**0.92**	0.03	0.03	0	0	0	0	0
WP	0	0	0.03	**0.93**	0.02	0	0	0	0	0.02
UL	0.01	0.01	0	0.02	**0.93**	0	0.01	0.02	0	0
PG	0.01	0	0.02	0	0.01	**0.92**	0.03	0.01	0	0
CC	0	0.02	0.01	0	0.01	0.03	**0.91**	0	0	0.02
UV	0	0	0	0	0.02	0	0	**0.96**	0.02	0
PR	0	0	0	0.01	0.02	0.02	0	0.01	**0.93**	0.01
LS	0	0	0	0.02	0.02	0	0	0	0	**0.96**

Figure 14. Confusion matrix of MSR Daily Activity 3D dataset, DR = drink, ET = eat, RB = read book, WP = write on a paper, UL = use laptop, PG = play game, CC = call cellphone, UV = use vacuum cleaner, PR = play guitar, LS = lay on a sofa.

	RG	SF	TC	WM	MV	DW	LP	OV
RG	**0.92**	0.03	0	0.02	0.01	0.01	0	0.01
SF	0.04	**0.90**	0.01	0.03	0.01	0	0	0.01
TC	0.01	0.02	**0.87**	0.03	0.03	0.02	0.02	0
WM	0	0	0.03	**0.91**	0.04	0	0	0.02
MV	0	0.01	0.01	0.03	**0.92**	0	0.01	0.02
DW	0.03	0.03	0	0	0	**0.88**	0.02	0.04
LP	0	0	0.02	0.01	0.02	0.03	**0.89**	0.03
OV	0.02	0.02	0.01	0.02	0.02	0.01	0.02	**0.88**

Figure 15. Confusion matrix D3D-HOI dataset, RG = refrigerator, SF = storage furniture, TC = trashcan, WM = washing machine, MV = microwave, DW = dishwasher, LP = laptop, OV = oven.

4.3. Experiment II: Computational Complexity

While determining the efficiency of an algorithm, its computational complexity should also be considered. The most common way of doing this is in terms of the big-O notation. The big-O notation represents the upper bound of an algorithm. For neural networks, the complexity depends on their layers. Hence, the total complexity of a neural network is the sum of the complexity of each layer. There are three types of layers in the used GCN model: convolutional layers, pooling layers, and a fully connected layer. A convolution is the sum of the row-wise dot products of a filter with a region matrix. For a filter of size k, the cost of the dot product is $O(k.d^2)$, where d represents the depth dimension. Since the filter is applied over the input $n-k+1$ times, where n is the length of the input or the input nodes in this case, the final complexity of a convolutional layer is $O(k.n.d^2)$. A max pooling layer basically finds the maximum value in each region. For search in an unsorted array, where each element of the array is visited at least once, the complexity is $O(n)$. Finally, in a fully connected layer, the input row vector is multiplied with the weight matrix. Hence, its computational cost is $O(k.n.d)$. Table 2 provides an overview of the running times of each layer.

Table 2. Computational complexity of the GCN model.

Layer Type	Equation	Complexity	Dataset	Feature Map	Time (s)
Convolutional	$z^l = \sum_{k=1}^{K} h_k^{n-1} * w_{ij}^n$ (18)	$O(k.n.d^2)$	Olympic Sports	$1 \times 3{,}316{,}800$	29,851,200
			MSR Daily Activity	$1 \times 1{,}382{,}000$	12,438,000
			D3D HOI	$1 \times 1{,}768{,}960$	15,920,640
Max Pooling	$h_{xy}^l = max_{i=0,\ldots s, j=0,\ldots s} h_{(x+i)(y+j)}^{l-1}$ (19)	$O(n)$	Olympic Sports	$1 \times 3{,}316{,}800$	7,462,800
			MSR Daily Activity	$1 \times 1{,}382{,}000$	3,109,500
			D3D HOI	$1 \times 1{,}768{,}960$	3,980,160
Fully Connected	$z^l = \sum_{i=1}^{n} w_{jk} x_i + w_{j0}$ (20)	$O(k.n.d)$	Olympic Sport	3,316,800	31,095,000
			MSR Daily Activity	1,382,000	39,801,600
			D3D HOI	1,768,960	31,841,280

4.4. Experiment III: Body Part Detection Rate

The accurate detection of human body parts leads to better classification results. Hence, the class-wise accuracies of the 12 body parts detected using the proposed key-point

detection algorithm are also discussed. First, the Euclidean distance D between the ground truth value and the detected value of each key body part is computed using Equation (21):

$$D_i = \sqrt{(DV_{ix} - GT_{ix})^2 + (DV_{iy} - GT_{iy})^2} \quad (21)$$

where DV is the detected value and GT is the ground truth value of a body part i. Based on its distance from the ground truth value, the accuracy of the detected body part is computed using Equation (22):

$$Acc_i = \frac{100}{K}\left[\sum_{n=1}^{K}\begin{cases} 1 \; if \; D_i \leq Th \\ 0 \; if \; D_i > Th \end{cases}\right] \quad (22)$$

where Th is the threshold value, which is set to 15, and n represents the total sample frames of each interaction class. Tables 3–5 show the average body part detection accuracies achieved by the proposed system over the Olympic Sports, MSR daily Activity 3D, and D3D HOI datasets, respectively.

Table 3. Body part detection rate achieved over Olympic Sports dataset.

Part	BB	BL	TS	PT	DS	HR	JV	SP	SB	CJ	AVG
HD	92.23	90.34	90.03	90.12	92.24	93.4	94.32	90.45	88.02	90.12	91.13
RE	95.67	93.03	92.12	89.45	93.56	96.05	92.35	94.32	89.45	87.61	92.36
LE	87.61	95.67	91.78	94.38	87.61	89.45	93.62	87.61	93.56	89.45	91.07
RH	91.45	90.51	89.45	95.67	92.23	94.38	90.56	93.27	95.67	93.27	92.65
LH	89.45	90.12	88.02	91.45	92.35	88.02	92.03	93.56	96.05	92.23	91.33
NK	94.38	96.05	94.38	95.67	93.35	93.35	91.14	92.23	87.61	95.67	93.38
TRS	93.62	92.72	95.67	87.24	87.61	87.61	93.35	95.67	92.23	87.61	91.33
BTR	92.23	95.67	89.45	92.06	94.38	92.06	87.61	89.45	93.56	88.02	91.45
RK	93.35	91.39	94.38	92.23	88.02	92.23	93.24	93.56	94.38	92.23	92.50
LK	93.56	88.02	93.56	95.67	94.32	93.56	90.76	94.32	92.23	92.06	92.81
RF	89.45	91.45	89.45	91.45	94.38	92.23	89.45	94.38	88.02	94.32	91.46
LF	87.61	89.45	95.67	94.12	92.06	93.27	93.56	89.45	89.72	88.02	91.29
Average part detection rate = **91.89%**											

HD = head, RE = right elbow, LE = left elbow, RH = right hand, LH = left hand, NK = neck, TRS = upper torso, LTR = bottom torso, RK = right knee, LK = left knee, RF = right foot, LF = left foot, AVG = average.

Table 4. Body part detection rate achieved over MSR Daily Activity 3D dataset.

Part	DR	ET	RB	WP	UL	PG	CC	UV	PR	LS	AVG
HD	92.23	90.34	90.03	90.12	92.24	93.4	94.32	90.45	94.35	90.12	91.76
RE	95.67	93.03	92.12	90.11	93.56	96.05	92.35	94.32	93.27	96.05	93.65
LE	93.35	95.67	91.78	94.38	96.05	94.38	93.62	92.23	93.56	94.32	93.93
RH	91.45	90.51	91.63	95.67	92.23	94.38	90.56	93.27	95.67	93.27	92.86
LH	97.59	90.12	95.67	91.45	92.35	97.59	92.03	93.56	96.05	92.23	93.86
NK	94.38	96.05	94.38	95.67	93.35	93.35	91.14	92.23	97.59	95.67	94.38
TRS	93.62	92.72	95.67	87.24	94.32	95.67	93.35	95.67	92.23	94.32	93.48
BTR	92.23	95.67	97.59	93.56	94.38	94.38	90.42	92.23	93.56	96.05	94.01
RK	93.35	91.39	94.38	92.23	97.59	92.23	93.24	93.56	94.38	92.23	93.46
LK	93.56	97.59	93.56	95.67	94.32	93.56	90.76	94.32	92.23	94.38	94.00
RF	93.27	91.45	92.23	91.45	94.38	92.23	91.09	94.38	97.59	94.32	93.24
LF	95.67	92.35	95.67	94.38	93.27	93.27	93.56	92.35	94.38	93.27	93.82
Average part detection rate = **93.53%**											

HD = head, RE = right elbow, LE = left elbow, RH = right hand, LH = left hand, NK = neck, TRS = upper torso, LTR = bottom torso, RK = right knee, LK = left knee, RF = right foot, LF = left foot, AVG = average.

Table 5. Body part detection rate achieved over D3D-HOI dataset.

Part	BB	BL	TS	PT	JV	SP	SB	CJ	AVG
HD	93.21	90.34	90.03	90.12	92.24	93.4	94.32	90.45	91.764
RE	92.23	93.03	92.12	90.11	92.35	96.05	92.35	94.32	92.820
LE	86.29	95.67	91.78	94.38	96.05	84.12	93.62	92.23	91.768
RH	91.45	90.51	91.63	95.67	86.45	94.38	90.56	93.27	91.740
LH	97.59	90.12	95.67	91.45	92.35	97.59	92.03	82.32	92.390
NK	94.38	96.05	94.38	95.67	85.23	82.75	91.14	92.23	91.479
TRS	93.62	92.72	95.67	87.24	94.32	95.67	93.35	95.67	93.533
BTR	92.23	95.67	97.59	92.23	94.38	83.03	90.42	92.23	92.223
RK	90.09	91.39	94.32	92.23	97.59	92.23	93.24	95.67	93.345
LK	88.43	97.59	92.23	95.67	94.32	82.45	90.76	94.32	91.971
RF	93.27	91.45	92.23	91.45	84.77	92.23	91.09	94.38	91.359
LF	94.38	92.35	95.67	94.38	93.27	93.27	93.03	92.35	93.588
Average part detection rate = **92.33%**									

HD = head, RE = right elbow, LE = left elbow, RH = right hand, LH = left hand, NK = neck, TRS = upper torso, LTR = bottom torso, RK = right knee, LK = left knee, RF = right foot, LF = left foot, AVG = average.

4.5. Experiment IV: Comparison with Other Well-Known Classifiers

In this experiment, the performance of the proposed system is analyzed by comparing the results obtained with GCN to the results of two other well-known classifiers, namely, the CNN [56] and fully convolutional network (FCN), on the same feature vector. The comparison of the performance is conducted in terms of precision, recall, and F1-score. This process is repeated for all three datasets that were mentioned before. Table 6 represents the comparison results over the Olympic Sports dataset while Table 7 shows the comparison using MSR Daily Activity 3D, and finally, Table 8 summarizes the comparison results achieved over the D3D HOI dataset.

Table 6. Comparison of GCN with CNN and FCN classifiers on Olympic Sports dataset in terms of precision, recall and F1-score.

HOI Class	CNN			FCN			GCN		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
BB	0.87	0.88	0.87	0.93	0.93	0.93	0.95	0.97	0.96
BL	0.91	0.91	0.91	0.91	0.91	0.91	0.94	0.95	0.94
TS	0.92	0.92	0.92	0.93	0.94	0.93	0.95	0.96	0.95
PT	0.92	0.91	0.91	0.92	0.93	0.92	0.95	0.95	0.95
DS	0.89	0.90	0.89	0.89	0.90	0.89	0.93	0.94	0.93
HR	0.88	0.89	0.88	0.88	0.89	0.88	0.93	0.93	0.93
JV	0.89	0.89	0.89	0.91	0.91	0.91	0.92	0.94	0.93
SP	0.87	0.88	0.87	0.87	0.88	0.87	0.92	0.92	0.92
SB	0.87	0.87	0.87	0.87	0.87	0.87	0.91	0.92	0.91
CJ	0.88	0.88	0.88	0.89	0.89	0.89	0.92	0.93	0.92
Mean	0.89	0.89	0.89	0.90	0.91	0.90	0.93	0.94	0.94

BB = basketball, BL = bowling, TS = tennis-serve, PT = platform, DS = discus, HR = hammer, JV = javelin, SP = shot put, SB = spring board, CJ = clean jerk.

Table 7. Comparison of GCN with CNN and FCN classifiers on MSR Daily Activity dataset in terms of precision, recall and F1-score.

HOI Class	CNN			FCN			GCN		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
DR	0.88	0.87	0.87	0.91	0.91	0.91	0.92	0.93	0.92
ET	0.88	0.88	0.88	0.91	0.91	0.91	0.92	0.93	0.92
RB	0.85	0.86	0.85	0.88	0.89	0.88	0.91	0.92	0.91
WP	0.87	0.88	0.87	0.89	0.91	0.90	0.93	0.93	0.93
UL	0.89	0.89	0.89	0.89	0.91	0.90	0.93	0.93	0.93
PG	0.85	0.86	0.85	0.87	0.88	0.87	0.92	0.92	0.92
CC	0.84	0.85	0.84	0.89	0.89	0.89	0.90	0.91	0.90
UV	0.91	0.91	0.91	0.91	0.93	0.92	0.94	0.96	0.95
PR	0.85	0.85	0.85	0.89	0.91	0.90	0.92	0.93	0.92
LS	0.91	0.91	0.91	0.92	0.93	0.92	0.95	0.96	0.95
Mean	0.87	0.88	0.87	0.90	0.91	0.90	0.92	0.93	0.93

DR = drink, ET = eat, RB = read book, WP = write on a paper, UL = use laptop, PG = play game, CC = call cellphone, UV = use vacuum cleaner, PR = play guitar, LS = lay on a sofa.

Table 8. Comparison of GCN with CNN and FCN classifiers on D3D HOI dataset in terms of precision, recall and F1-score.

HOI Class	CNN			FCN			GCN		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
RG	0.87	0.89	0.88	0.91	0.91	0.91	0.91	0.92	0.91
SF	0.87	0.88	0.87	0.87	0.88	0.87	0.88	0.90	0.89
TC	0.83	0.83	0.83	0.85	0.85	0.85	0.87	0.87	0.87
WM	0.82	0.83	0.82	0.85	0.86	0.85	0.91	0.91	0.91
MW	0.86	0.85	0.85	0.88	0.89	0.88	0.91	0.92	0.91
DW	0.87	0.86	0.86	0.86	0.86	0.86	0.88	0.88	0.88
LP	0.82	0.82	0.82	0.85	0.87	0.86	0.89	0.89	0.89
OV	0.84	0.84	0.84	0.88	0.89	0.88	0.88	0.88	0.88
Mean	0.85	0.85	0.85	0.87	0.88	0.87	**0.89**	**0.90**	**0.89**

RG = refrigerator, SF = storage furniture, TC = trashcan, WM = washing machine, MV = microwave, DW = dishwasher, LP = laptop, OV = oven.

As shown in the tables, GCN acts as the best classifier for all three datasets while FCN comes second in terms of precision, recall, and F1-scores.

4.6. Experiment V: Comparison with Other State-of-the-Art Methods

For all three datasets that are used for testing, the interaction recognition accuracies achieved by the proposed system are compared with other state-of-the-art methods that were evaluated on the same datasets. The mean interaction recognition accuracy is computed by dividing the number of correct predictions by the total number of predictions that were made by the classifier as shown in Equation (23).

$$C_{Accuracy} = \frac{Correct\ Predictions}{Total\ Predictions} \times 100\% \tag{23}$$

Table 9 shows the results of the proposed system with some other state-of-the-art HOIR methods that were evaluated on one or more of the three datasets used in this research. The accuracy scores highlight that the proposed system outperforms all of them by a good margin.

Table 9. A comparison of proposed HOIR system with other state-of-the-art methods.

Method	Mean Accuracy %		
	Olympic Sports	**MSR Daily Activity 3D**	**D3D-HOI**
Metric learning autoencoder [57]	-	67.1	-
Deep moving poselets [58]	-	84.4	-
CNN [56]	-	-	85.1
DCS motion descriptors [59]	85.2	-	-
Actionlet ensemble [48]	-	86.0	-
CNN-LSTM [60]	-	-	87.2
Improved trajectories [45]	90.2	-	-
Combined deep architectures [61]	-	91.3	-
Spatial feature fusion [62]	-	92.9	-
Proposed	**94.1**	**93.2**	**89.6**

5. Discussion

In the proposed HOIR framework, all input images were pre-processed using sigmoid stretching and Gaussian filtering techniques. Raw input images produced relatively poor results; hence, their quality was improved using these pre-processing techniques. Although context and value of each key body part is computed scene information can provide additional cues, it is also important to observe the humans and objects in isolation. Therefore, the human–object pairs in the pre-processed images were localized, and the backgrounds were removed. Using the detected human silhouettes, 12 key body parts were identified. Then, dense trajectories and LIOP features were extracted from the full-body silhouettes. Similarly, kinematic posture and LOP features were mined from the key body parts. Lastly, SPM and GIST feature descriptors were obtained from the entire images. Experiments showed that each one of the six of feature descriptors contributed toward the system's overall efficiency. Removing any of these features in the final feature vector showed a negative effect on the system's performance. The three different types of features were merged together to form a large feature vector, which was then reduced using ISOMAP. The optimized feature vector was finally sent to a fine-tuned GCN architecture for the classification of interactions.

Although the proposed system classifies complex human–object interactions with high accuracy rates, it is not without limitations. It struggles to identify the correct interaction when two interactions are of a similar nature: for instance, opening and closing the same object in the same scene. One example of this would be the opening and closing of the microwave in the kitchen in the D3D HOI dataset. Although the system uses a time sequence to distinguish between the two interactions, it often misclassifies them. Similarly, when the size of the object is too small, the accuracy drops. This was evident during the testing of some examples in the MSR Daily Activity 3D dataset; for example, playing a game on a phone and reading a book were misclassified because the subject has both objects in their hands in a similar fashion and the sizes of the objects are too small to form a clear distinction.

The results and analysis of the proposed system are as follows. The mean interaction recognition accuracy of the proposed system over Olympics Sports dataset is 94.1%. For the MSR Daily Activity 3D dataset, the accuracy is 93.2% and 89.6% accurate results are achieved with the D3D HOI dataset. These results prove that the proposed HOIR system outperforms the available state-of-the-art techniques.

6. Conclusions and Future Works

The proposed HOIR system is capable of classifying complex human–object interactions in sequential data for healthcare. In particular, the proposed model involves pre-processing the image frames through contrast adjustment and noise removal. Then, it locates the human–object pair in them and removes the background from the images. Using the detected human silhouettes, its 12 key body parts are identified. Then dense

trajectories and LIOP features are extracted from the full-body silhouettes. The results of evaluating the proposed HOIR system showed higher accuracies as compared to many state-of-the-art systems. Some new feature descriptors can be added to further enhance this system, and different sensors can be used to obtain the input in future.

Author Contributions: Conceptualization, M.W. and Y.Y.G.; methodology, M.W. and A.J.; software, M.W., S.A.A. and M.G.; validation, M.W., Y.Y.G., S.A.C. and J.P.; formal analysis, M.G., S.A.A., S.A.C. and J.P.; resources, Y.Y.G., M.G. and J.P.; writing—review and editing, M.W., M.G., S.A.C. and J.P.; funding acquisition, Y.Y.G., S.A.C., S.A.A. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), 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). It was also supported by Emirates Center for Mobility Research (ECMR) Grant #12R012.

Acknowledgments: In addition; the authors would like to thank the support of the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University. This work has also been supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R239), Princess Nourah bint Abdulrahman.University, Riyadh, Saudi Arabia.

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

References

1. Jalal, A.; Sarif, N.; Kim, J.T.; Kim, T.-S. Human Activity Recognition via Recognized Body Parts of Human Depth Silhouettes for Residents Monitoring Services at Smart Home. *Indoor Built Environ.* **2012**, *22*, 271–279. [CrossRef]
2. Jalal, A.; Uddin, Z.; Kim, T.-S. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Trans. Consum. Electron.* **2012**, *58*, 863–871. [CrossRef]
3. Jalal, A.; Lee, S.; Kim, J.T.; Kim, T.-S. Human Activity Recognition via the Features of Labeled Depth Body Parts. In Proceedings of the Smart Homes Health Telematics, Artimono, Italy, 12–15 June 2012; pp. 246–249. [CrossRef]
4. Jalal, A.; Kim, J.T.; Kim, T.-S. Development of a life logging system via depth imaging-based human activity recognition for smart homes. In Proceedings of the International Symposium on Sustainable Healthy Buildings, Brisbane, Australia, 8–12 July 2012; pp. 91–95.
5. Tahir, S.B.U.D.; Jalal, A.; Batool, M. Wearable Sensors for Activity Analysis using SMO-based Random Forest over Smart home and Sports Datasets. In Proceedings of the 2020 3rd International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 17–19 February 2020; pp. 1–6.
6. Jalal, A.; Nadeem, A.; Bobasu, S. Human Body Parts Estimation and Detection for Physical Sports Movements. In Proceedings of the IEEE International Conference on Communication, Computing and Digital Systems, Islamabad, Pakistan, 6–7 March 2019.
7. Javeed, M.; Gochoo, M.; Jalal, A.; Kim, K. HF-SPHR: Hybrid Features for Sustainable Physical Healthcare Pattern Recognition Using Deep Belief Networks. *Sustainability* **2021**, *13*, 1699. [CrossRef]
8. Ansar, H.; Jalal, A.; Gochoo, M.; Kim, K. Hand Gesture Recognition Based on Auto-Landmark Localization and Reweighted Genetic Algorithm for Healthcare Muscle Activities. *Sustainability* **2021**, *13*, 2961. [CrossRef]
9. Khalid, N.; Gochoo, M.; Jalal, A.; Kim, K. Modeling Two-Person Segmentation and Locomotion for Stereoscopic Action Identification: A Sustainable Video Surveillance System. *Sustainability* **2021**, *13*, 970. [CrossRef]
10. Mahmood, M.; Jalal, A.; Kim, K. WHITE STAG model: Wise human interaction tracking and estimation (WHITE) using spatio-temporal and angular-geometric (STAG) descriptors. *Multimed. Tools Appl.* **2020**, *79*, 6919–6950. [CrossRef]
11. Kamal, S.; Jalal, A.; Kim, D. Depth Images-based Human Detection, Tracking and Activity Recognition Using Spatiotemporal Features and Modified HMM. *J. Electr. Eng. Technol.* **2016**, *11*, 1857–1862. [CrossRef]
12. Jalal, A.; Mahmood, M.; Hasan, A.S. Multi-features descriptors for Human Activity Tracking and Recognition in Indoor-Outdoor Environments. In Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019; pp. 371–376. [CrossRef]
13. 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 2020), Lahore, Pakistan, 17–19 February 2020; pp. 1–6.
14. Jalal, A.; Mahmood, M. Students' behavior mining in e-learning environment using cognitive processes with information technologies. *Educ. Inf. Technol.* **2019**, *24*, 2797–2821. [CrossRef]
15. Gochoo, M.; Tahir, S.B.U.D.; Jalal, A.; Kim, K. Monitoring Real-Time Personal Locomotion Behaviors over Smart Indoor-Outdoor Environments via Body-Worn Sensors. *IEEE Access* **2021**, *9*, 70556–70570. [CrossRef]
16. Jalal, A.; Kamal, S.; Kim, D. A Depth Video-based Human Detection and Activity Recognition using Multi-features and Embedded Hidden Markov Models for Health Care Monitoring Systems. *Int. J. Interact. Multimed. Artif. Intell.* **2017**, *4*, 54–62. [CrossRef]

17. Nadeem, A.; Jalal, A.; Kim, K. Automatic human posture estimation for sport activity recognition with robust body parts detection and entropy markov model. *Multimed. Tools Appl.* **2021**, *80*, 21465–21498. [CrossRef]
18. Gochoo, M.; Akhter, I.; Jalal, A.; Kim, K. Stochastic Remote Sensing Event Classification over Adaptive Posture Estimation via Multifused Data and Deep Belief Network. *Remote Sens.* **2021**, *13*, 912. [CrossRef]
19. Jalal, A.; Khalid, N.; Kim, K. Automatic Recognition of Human Interaction via Hybrid Descriptors and Maximum Entropy Markov Model Using Depth Sensors. *Entropy* **2020**, *22*, 817. [CrossRef] [PubMed]
20. Kamal, S.; Jalal, A. A Hybrid Feature Extraction Approach for Human Detection, Tracking and Activity Recognition Using Depth Sensors. *Arab. J. Sci. Eng.* **2015**, *41*, 1043–1051. [CrossRef]
21. Jalal, A.; Quaid, M.A.K.; Hasan, A.S. Wearable Sensor-Based Human Behavior Understanding and Recognition in Daily Life for Smart Environments. In Proceedings of the International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2018; pp. 105–110. [CrossRef]
22. Quaid, M.A.K.; Jalal, A. Wearable sensors based human behavioral pattern recognition using statistical features and reweighted genetic algorithm. *Multimed. Tools Appl.* **2020**, *79*, 6061–6083. [CrossRef]
23. Azmat, U.; Jalal, A. Smartphone Inertial Sensors for Human Locomotion Activity Recognition based on Template Matching and Codebook Generation. In Proceedings of the IEEE International Conference on Communication Technologies, Rawalpindi, Pakistan, 21–22 September 2021; pp. 1–6. [CrossRef]
24. Jalal, A.; Kim, Y. Dense Depth Maps-based Human Pose Tracking and Recognition in Dynamic Scenes Using Ridge Data. In Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance, Seoul, Korea, 26–29 August 2014; pp. 119–124.
25. Jalal, A.; Kamal, S. Real-Time Life Logging via a Depth Silhouette-based Human Activity Recognition System for Smart Home Services. In Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance, Seoul, Korea, 26–29 August 2014; pp. 74–80.
26. Jalal, A.; Kim, Y.; Kim, D. Ridge body parts features for human pose estimation and recognition from RGB-D video data. In Proceedings of the Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Hefei, China, 11–14 July 2014; pp. 1–6.
27. Jalal, A.; Kim, Y.; Kamal, S.; Farooq, A.; Kim, D. Human daily activity recognition with joints plus body features representation using Kinect sensor. In Proceedings of the IEEE International Conference on Informatics, Electronics and Vision, Fukuoka, Japan, 15–18 June 2015; pp. 1–6. [CrossRef]
28. Jalal, A.; Kim, J.T.; Kim, T.S. Human activity recognition using the labeled depth body parts information of depth silhouettes. In Proceedings of the 6th International Symposium on Sustainable Healthy Buildings, Seoul, Korea, 27 February 2012; pp. 1–8.
29. Jalal, A.; Kamal, S.; Farooq, A.; Kim, D. A spatiotemporal motion variation features extraction approach for human tracking and pose-based action recognition. In Proceedings of the IEEE International Conference on Informatics, Electronics and Vision, Fukuoka, Japan, 15–18 June 2015; pp. 1–6. [CrossRef]
30. Pervaiz, M.; Jalal, A.; Kim, K. Hybrid Algorithm for Multi People Counting and Tracking for Smart Surveillance. In Proceedings of the 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 12–16 January 2021; pp. 530–535. [CrossRef]
31. Fang, H.-S.; Cao, J.; Tai, Y.-W.; Lu, C. Pairwise Body-Part Attention for Recognizing Human-Object Interactions. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018; pp. 51–67. [CrossRef]
32. Mallya, A.; Lazebnik, S. Learning Models for Actions and Person-Object Interactions with Transfer to Question Answering. In Proceedings of the CVPR, New Orleans, LA, USA, 14–19 June 2020; pp. 414–428.
33. Yan, W.; Gao, Y.; Liu, Q. Human-object Interaction Recognition Using Multitask Neural Network. In Proceedings of the 2019 3rd International Symposium on Autonomous Systems (ISAS), Shanghai, China, 29–31 May 2019; pp. 323–328. [CrossRef]
34. Gkioxari, G.; Girshick, R.; Dollár, P.; He, K. Detecting and recognizing human-object interactions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8359–8367.
35. Li, J.; Xiong, C.; Hoi, S.C. Comatch: Semi-supervised learning with contrastive graph regularization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 9475–9484.
36. Li, Y.-L.; Liu, X.; Lu, H.; Wang, S.; Liu, J.; Li, J.; Lu, C. Detailed 2D-3D Joint Representation for Human-Object Interaction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10163–10172. [CrossRef]
37. Xia, L.-M.; Wu, W. Graph-based method for human-object interactions detection. *J. Cent. South Univ.* **2021**, *28*, 205–218. [CrossRef]
38. Yang, D.; Zou, Y. A graph-based interactive reasoning for human-object interaction detection. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20), Yokohama, Japan, 11–17 July 2020; pp. 1111–1117.
39. Sunkesula, S.P.R.; Dabral, R.; Ramakrishnan, G. Lighten: Learning interactions with graph and hierarchical temporal networks for hoi in videos. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 691–699.
40. Qi, S.; Wang, W.; Jia, B.; Shen, J.; Zhu, S.-C. Learning Human-Object Interactions by Graph Parsing Neural Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 401–417. [CrossRef]

41. Liu, X.; Ji, Z.; Pang, Y.; Han, J.; Li, X. DGIG-Net: Dynamic Graph-in-Graph Networks for Few-Shot Human-Object Interaction. *IEEE Trans. Cybern.* **2021**, 1–13. [CrossRef]
42. Vedaldi, A.; Soatto, S. Quick Shift and Kernel Methods for Mode Seeking. In Proceedings of the ECCV, Marseille, France, 12–18 October 2008; pp. 705–718. [CrossRef]
43. Xu, X.; Li, G.; Xie, G.; Ren, J.; Xie, X. Weakly supervised deep semantic segmentation using CNN and ELM with semantic candidate regions. *Complexity* **2019**, *2019*, 9180391. [CrossRef]
44. Falcao, A.X.; Stolfi, J.; de Alencar Lotufo, R. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 19–29. [CrossRef]
45. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 2–8 December 2013; pp. 3551–3558.
46. Wang, Z.; Fan, B.; Wu, F. Local intensity order pattern for feature description. In Proceedings of the International Conference on Computer Vision, Barcela, Spain, 6–13 November 2011; pp. 603–610.
47. Ahad, A.R.; Ahmed, M.; Das Antar, A.; Makihara, Y.; Yagi, Y. Action recognition using kinematics posture feature on 3D skeleton joint locations. *Pattern Recognit. Lett.* **2021**, *145*, 216–224. [CrossRef]
48. Wang, J.; Liu, Z.; Wu, Y.; Yuan, J. Learning actionlet ensemble for 3D human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 914–927. [CrossRef] [PubMed]
49. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the IEEE Computer Society on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 2169–2178.
50. Oliva, A.; Torralba, A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
51. Tenenbaum, J.B.; de Silva, V.; Langford, J.C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **2000**, *290*, 2319–2323. [CrossRef] [PubMed]
52. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the ICLR 2017, Toulon, France, 24–26 April 2017; pp. 1–17.
53. Niebles, J.C.; Chen, C.-W.; Fei-Fei, L. Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In Proceedings of the 11th European Conference on Computer Vision (ECCV), Heraklion Crete, Greece, 5–11 September 2010; pp. 392–405. [CrossRef]
54. Wang, J.; Liu, Z.; Wu, Y.; Yuan, J. Mining actionlet ensemble for action recognition with depth cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1290–1297.
55. Xu, X.; Joo, H.; Mori, G.; Savva, M. D3D-HOI: Dynamic 3D Human-Object Interactions from Videos. *arXiv* **2021**, arXiv:2108.08420.
56. Waheed, M.; Javeed, M.; Jalal, A. A Novel Deep Learning Model for Understanding Two-Person Intractions Using Depth Sensors. In Proceedings of the ICIC, Lahore, Pakistan, 9–10 November 2021; pp. 1–8.
57. Andresini, G.; Appice, A.; Malerba, D. Autoencoder-based deep metric learning for network intrusion detection. *Inf. Sci.* **2021**, *569*, 706–727. [CrossRef]
58. Mavroudi, E.; Tao, L.; Vidal, R. Deep moving poselets for video based action recognition. In Proceedings of the Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 111–120.
59. Jain, M.; Jégou, H.; Bouthemy, P. Improved Motion Description for Action Classification. *Front. ICT* **2016**, *2*, 28. [CrossRef]
60. Waheed, M.; Jalal, A.; Alarfaj, M.; Ghadi, Y.Y.; Al Shloul, T.; Kamal, S.; Kim, D.-S. An LSTM-Based Approach for Understanding Human Interactions Using Hybrid Feature Descriptors over Depth Sensors. *IEEE Access* **2021**, *9*, 167434–167446. [CrossRef]
61. Tomas, A.; Biswas, K.K. Human activity recognition using combined deep architectures. In Proceedings of the IEEE International Conference on Signal and Image Processing (ICSIP), Singapore, 4–6 August 2017; pp. 41–45.
62. Liang, M.; Jiao, L.; Yang, S.; Liu, F.; Hou, B.; Chen, H. Deep Multiscale Spectral-Spatial Feature Fusion for Hyperspectral Images Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 2911–2924. [CrossRef]