inventions

MDPI

Article

Deep Learning-Based Action Recognition Using 3D Skeleton Joints Information

Nusrat Tasnim, Md. Mahbubul Islam and Joong-Hwan Baek *

Department of Electronics and Information Engineering, Korea Aerospace University, Goyang 10540, Korea; tasnim.nishu70@kau.kr (N.T.); mahbubcse@cu.ac.bd (M.M.I.)

* Correspondence: jhbaek@kau.ac.kr

Received: 20 July 2020; Accepted: 15 September 2020; Published: 17 September 2020

check for updates

Abstract: Human action recognition has turned into one of the most attractive and demanding fields of research in computer vision and pattern recognition for facilitating easy, smart, and comfortable ways of human-machine interaction. With the witnessing of massive improvements to research in recent years, several methods have been suggested for the discrimination of different types of human actions using color, depth, inertial, and skeleton information. Despite having several action identification methods using different modalities, classifying human actions using skeleton joints information in 3-dimensional space is still a challenging problem. In this paper, we conceive an efficacious method for action recognition using 3D skeleton data. First, large-scale 3D skeleton joints information was analyzed and accomplished some meaningful pre-processing. Then, a simple straight-forward deep convolutional neural network (DCNN) was designed for the classification of the desired actions in order to evaluate the effectiveness and embonpoint of the proposed system. We also conducted prior DCNN models such as ResNet18 and MobileNetV2, which outperform existing systems using human skeleton joints information.

Keywords: 3D skeleton data; action recognition; deep learning; pre-processing

1. Introduction

Over the past years, various machines such as computers, mobiles, and cameras have been introduced by many scholars with the massive improvement of modern science. At an early age, most of the devices were so enormous and they were very difficult to handle when fulfilling tasks. The input-output peripherals, including mouse, keyboards, and speakers, were too mountainous and uncomfortable for use. With the evolution of modern science, many popular research organizations such as Google, Amazon, Microsoft, and Naver spent a lot of time developing easy, efficient, and comfortable methods of input-output for controlling devices. Nowadays, computers, mobiles, and cameras are compacted into a small chunk to make them easily portable. Ways of interacting with these machines has also developed at an alarming rate. Recently, we are grasping the use of wireless mouses, keyboards, and headphones, as well as many other digital devices to interact with systems in luxurious and comfortable ways when performing desired tasks. The development of hardware-based input-output devices is now at its end because machine learning is progressing at a high speed. The world of interaction is moving towards vision-based technology. Several methods have been proposed for communicating with machines via vision-based methods, for example bio-metric, face recognition, voice recognition, iris recognition, and other types of pattern recognition. Computer vision-based methods, for instance hand gesture recognition, gait recognition, human action recognition, and pose estimation, are the most popular in today's research era. Hand gesture recognition has gained extensive popularity in terms of communicating with the devices, particularly those interacting with objects in augmented and virtual reality. A gesture is an alternate form of communication that can be rendered from person to person or person to machine to indicate a specified motive. In [1], Kopuklu et al. derived a real-time hand gesture recognition system using both EgoGesture and NVGestures datasets, considering ResNet as the detector and classifier. Molchanov et al. [2] introduced a dynamic hand gestures system for online detection and classification with a recurrent 3D convolution neural network. Hand gesture recognition inspired human action recognition in which most of the actions are performed by hands only.

Gait recognition is also important for person identification. It is a passive mode of collecting data about persons to unify them using their body shapes and types of movements. Zou et al. [3] proposed a method to extract discriminative features from inertial data using DCNN and long short-term memory (LSTM), respectively. Then, two fully connected layers were integrated to perform gait classification by combining the extracted features. Wu et al. [4] constructed a convolution neural network (CNN) model to generate meaningful features from gait energy images and applied different stages, which were fused to classify the gait.

With the advantage of low cost, portability, and easy to use sensors, action recognition has become a popular research field among contemporary researchers. Nowadays, using these sensors, we get some renowned and perfect datasets, which are used in experiments. In the early days, many handcrafted methods were proposed for the extraction of distinctive features from the skeleton data of the actions in order to perform the recognition. Feature extraction for depth and red, green, and blue (RGB) sequences need more computation than skeleton data in terms of time and computing resources. Most of the existing models represented skeleton joints information into spatial format and performed classification using handcrafted features. Few methods used hidden Markov models for capturing temporal information from the skeleton data in the early days. After that, deep learning networks including recurrent neural network (RNN) or CNN were used largely for skeleton-based action recognition in the last few years.

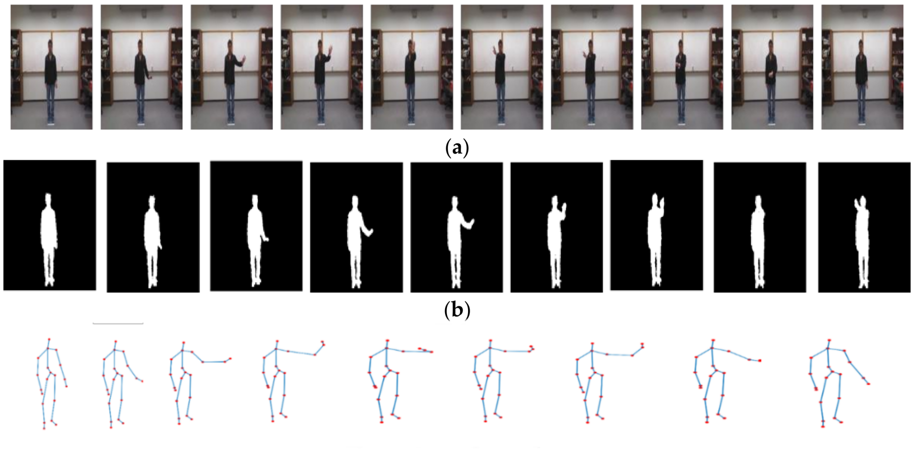
2. Literature Review

This section covers related research, including action recognition using various modalities including color, depth, inertial, and skeleton.

2.1. Action Recognition System

An action is a group of motions or frames that exhibits what a person is doing, for example running, walking, or waving. Usually, the duration of the action lasts no more than a few seconds. The main concern of human action recognition systems is how to identify what type of action exists in a set of frames. With the massive advancement of modern science, a variety of digital devices including the RGB camera, depth camera, RGB-D camera, Microsoft Kinect, RealSense Camera, Leap Motion Controller, and Z-Depth Camera have been invented by several companies with the help of a group of scholars. These devices provide separate facilities to capture different types of action's data including color, depth, skeleton, and so on, in order to detect and classify them into different classes. The numerous actions captured using the above devices have several effects in terms of different views, time, and types. After capturing the action through different views, the action can vary based on time and the person performing the action. Actions captured using different sensors with different modalities including RGB color images, depth, and skeleton are shown in Figure 1.

RGB cameras were used for indicating actions from image sequences in the early days of action recognition. After that, depth cameras came out, providing clearer information by giving a 3D structural view. Moreover, the depth camera is beneficial in the daytime and the nighttime. Depth sensors such as Microsoft Kinect and RealSense also provide three types of data (depth images, RGB images, and skeleton joints) [5]. Thus, these devices make it possible to proceed with the recognition of human actions. Every dataset includes different actions, and those actions are performed by several subjects at multiple times. Feature extraction is one of the major tasks in action recognition using machine learning. The performance of feature extraction methods for an action recognition system is evaluated based on classification.



(**a**)

(**b**)

(**c**)

Figure 1. Different modality of an action: (**a**) RGB, (**b**) depth(silhouette), and (**c**) skeleton representations.

Chen et al. [6] introduced a new method that was computationally efficient in the recognition of human actions using depth videos. The authors of this paper tried to extract depth motion map (DMM) from three different views including front, side, and top. Then, they used the local binary pattern (LBP) algorithm for generating features from the DMMs. First, they showed the experimental results for feature-level concentration using the LBPs features. Both feature-level and decision-level concatenations outperformed over two different datasets. Trelinski et al. [7] optimized a CNN model for action recognition that used only depth maps. Consecutive depth maps and depth maps projected onto an orthogonal Cartesian plane both were used as input sequences for a multi-channel when the model was trained. Wang et al. [8] described a new method for human action recognition from depth maps using weighted hierarchical depth motion maps and three-channel deep convolutional neural networks (3ConvNets). Simonyan et al. [9] developed an action recognition model based on two-stream convolutional neural networks. They improved the model in three ways. First, they combined spatial and temporal networks to form a two-stream ConvNet architecture. Second, ConvNet was trained in multi-frame dense optical flow. Finally, they applied the proposed method to different datasets for recognizing performed actions. Dollar et al. [10] introduced an effective method for detecting spatiotem-poral interest points using a temporal Gabor filter and a spatial gaussian filter. Wu at al. [11] suggested a method for action recognition by combining both local and global representation. They used bag of correlated poses as a temporal local feature descriptor and extended-MHI, which is a holistic motion descriptor. Ahmed at al. [12] described a body silhouette feature, optical flow feature, and combined feature for action recognition and then used HMM for modeling and testing actions. Xia et al. [13] introduced a methodology to recognize human action using histograms of 3D joint locations and used HMM for classification. Luo et al. [14] used a temporal pyramid matching approach for feature representation, and the classification support vector machine (SVM) was conducted. Megavannan et al. [15] demonstrated a feature extraction method similar to Hu Moments and a silhouette bonding box for depth images, which was classified by SVM.

2.2. *Skeleton-Based Action Recognition*

Many researchers have spent their valuable time preparing some challenging datasets based on the skeleton of the human body for further experiments including NTU, MSR Action3D, Berkeley MHAD, HDM05, and UTD multimodal human action dataset (UTD-MHAD) datasets. Several ideas developed based on skeleton data [16–29] are used to separate action classes. In [16], J. Imran et al. evaluated a method based on skeleton augmented data of 3D skeleton joints information using 3D transformations and designed a RNN-based BiGRU for the purpose of classification. In [17], Li et al. introduced a method for action classification using CNN with 7-layers. Du et al. [18] supported the recognition of action by proposing an end-to-end hierarchical architecture with CNN using the quantified image of skeleton joints. Chen et al. [19] illustrated a method called temporal pyramid skeleton motion maps (TPSMMs) to encode spatio-temporal information into color images.

Li et al. [20] represented a simple efficient method by encoding the spatio-temporal information of the skeleton sequences into color texture images known as joint distance maps (JDMs). An effective method based on the skeleton MHAD-UTD dataset was proposed by Hou et al. in [21] by expressing a skeleton sequence of the action. Color variation was used to represent the discrete trajectories of joints to capture the temporal information, and adopted to encode the temporal information into hue. In [22], Wang et al. introduced a compact, effective, yet simple method to encode spatio-temporal information carried in 3D skeleton sequences into a multiple 2D images joint. Rashmi et al. [23] illustrated a new technique for skeleton data where the most informative distance and the angle between joints were taken as a feature set, and deep neural network was used for action recognition. They described the skeleton joints as a tree where the center was considered the root node of the tree. Huynh-The et al. [24] proposed a novel encoding method to transform skeleton information into image-based information called pose-transition feature and then into image representation for deep convolutional neural networks. Si et al. [25] developed a hierarchical spatial reasoning network that receives information about each part, and the joints of each part, of the body using a graph neural network. For dynamic skeleton sequences, they also proposed a temporal stack learning network. Li et al. [26] represented skeleton sequences as a subset of geometric algebra to extract temporal and spatial features such as shape and motion. They also designed a rotor based view transformation method and spatio temporal view-invariant model to eliminate the effect of viewpoint variation and to combine skeleton joints and bones to capture spatial configuration and temporal dynamics, respectively.

Yang et al. [27] introduced a discriminative framework called multi-instance multitask learning to disclose the relationship between skeleton joints and action. In [28], a method based on a spatio-temporal pyramid is used to capture spatio-temporal information to perform action classification by Yang et al. Zanfir et al. [29] discussed a pose descriptor in terms of pose information as well as differential quantities such as speed and acceleration, and after that, the k-nearest neighbors (KNN) classifier is used.

The aforementioned methods have used skeleton joints information for human activity recognition because its application areas are spreading significantly with the massive development in modern technologies, for instances pose estimation [30], human behavior [31], and facial expression detection [32]. There is some modern research utilizing skeleton joints data to analyze movements such as dance motion [33], choreography analysis, and track and field sports.

3. Proposed Methodology

We integrate the successes of existing methods focusing on action recognition, specifically skeleton-based action discrimination, in Section 2.2. From the above interlocution, it can be summarized that the majority of papers have focused on the spatial representation of skeleton joints coordinates values and the capturing of temporal variation using different color models [19–22], but they lose information due to trajectory overlapping and view dependency. However, we consider the translated joint's coordinate values along X, Y, and Z axes, which are view independent. Several methods used a segment of video, defined as clip length (24 and 32), of RGB and depth datasets [1,2] for gestures or actions recognition. Thus, it is pressingly necessary to design a system that can distinguish actions by extracting features from a segment of video using skeleton joints information. The key ideas of the resolved research, preprocessing, and the designed network are elaborated in the rest of the subsections.

3.1. Research Proposal

For the simplification of the implementation and improvements of accuracy, we propose a new scenario for human action recognition using skeleton joints data. First, we observe and analyze the 3D skeleton joints values along the XYZ axes of each frame and then perform preprocessing. After that, the deep convolution neural network (DCNN) is used for discrimination of the preprocessed data. Details of the proposed method are illustrated below.

Let us consider $F_i\left(X_{ij},\ Y_{ij},\ Z_{ij}\right)$ as the i^{th} frame along XYZ axes where $j = 1,\ 2, \ldots, 20$ is the number of joints. These are the following steps of our proposed method as listed in Algorithm 1:

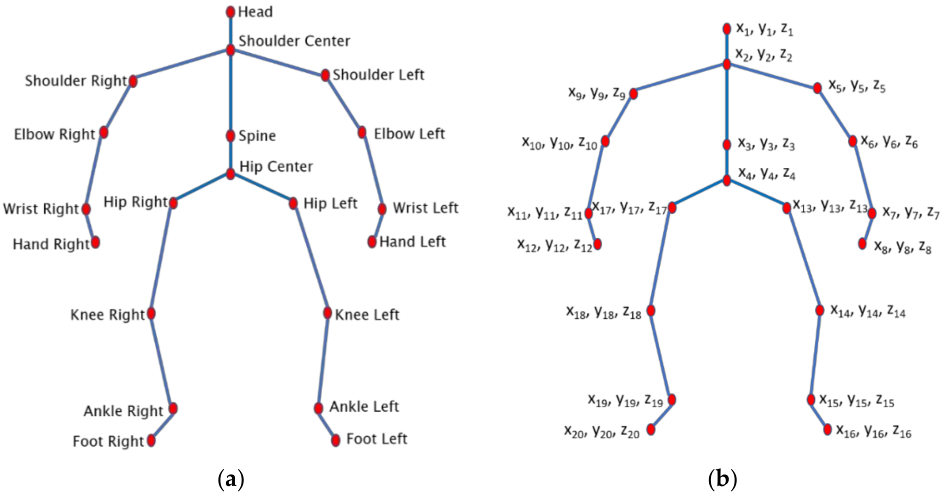
Algorithm 1. Working principle of the proposed method.

1. Input a sequence of frames ($F_i;\ i = 1,\ 2,\ \ldots,\ n$), where n is the length of the video.
2. Calculate the difference between the adjacent frames F_i and F_{i+1} separately for X, Y, and Z axes, yielding a new sequence of frames with a length of $n-1$; ($F'_i;\ i = 1,\ 2,\ \ldots,\ n-1$).
3. Make sure each training and testing dataset have $N(20,\ 22, \ldots, 32)$ frames in which half of the frames are overlapped.
4. Classify them for the desired group and repeat step 1 to 4.
5. Finish.

We try to present a detailed description of the preprocessing and architecture of the proposed system in the next sub-sections.

3.2. Skeleton Joints Analysis and Preprocessing

We unify the twenty joints by numbering in each frame, as shown in Figure 2. First, we number the twenty joints (head, houlder_center, spine, hip_center, left_center, left_elbow, left_wrist, left_hand, right_shoulder, right_elbow, right_wrist, right_hand, left_hip, left_knee, left_ankle, left_foot, right_hip, right_knee, right_ankle, right_foot) with separate variables (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}), respectively, in each frame shown in Figure 2. The same sequences for numbering are also assumed for both Y and Z axes respectively. When we analyze joints information along the X, Y, and Z axes, we observe that there is an inevitable variation among joints positions when actions are performed using different parts of our bodies.



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

Figure 2. Skeleton views: (**a**) joint names and (**b**) corresponding numbering of joints.

We calculate the subtraction of coordinate values for every joint in the i^{th} frame $F_i\left(X_{ij},\ Y_{ij},\ Z_{ij}\right)$ from the every joint in the adjacent $(i+1)^{th}$ frame $F_{i+1}\left(X_{(i+1)j},\ Y_{(i+1)j},\ Z_{(i+1)j}\right)$, where $j\ =\ 1,\ 2,\ \ldots,\ 20$. This is presented in Equations (1)–(4).

$$F_i'\left(X_{ij}',\ X_{ij}', X_{ij}'\right)\ =\ F_{i+1}\left(X_{(i+1)j},\ Y_{(i+1)j},\ Z_{(i+1)j}\right)-F_i\left(X_{ij},\ Y_{ij},\ Z_{ij}\right) \tag{1}$$

Along the X axis,

$$X_{ij}'\ =\ X_{(i+1)j}-X_{ij} \tag{2}$$

Similarly, along Y, and Z axes,

$$Y_{ij}'\ =\ Y_{(i+1)j}-Y_{ij} \tag{3}$$

and

$$Z_{ij}'\ =\ Z_{(i+1)j}-Z_{ij} \tag{4}$$

where $i\ =\ 1,\ 2,\ldots,\ n$; n is the length of the video, and $F_i'\left(X_{ij}',\ Y_{ij}',\ Z_{ij}'\right)$ is the translated joints in the i^{th} frame along XYZ axes. The resultant values form a matrix (I_M) for i^{th} frame temporal information are shown in Equation (5).

$$Matrix,\ I_M\ =\begin{bmatrix} {x'}_{i1} & {y'}_{i1} & {z'}_{i1} \\ {x'}_{i2} & {y'}_{i2} & {z'}_{i2} \\ x'\ \ _{i3} & y'\ \ _{i3} & z'\ \ _{i3} \\ , & , & , \\ . & . & . \\ . & . & . \\ . & . & . \\ x_{i20}' & y_{i20}' & z_{i20}' \end{bmatrix} \tag{5}$$

Then, we construct the training and testing dataset with clip length $N(20,\ 22,\ldots,32)$ to form a matrix, presented below in Equation (6):

$$I_{MN}'\ =\ \{I_{M1}, I_{M2}, \ldots,\ I_{MN}\} \tag{6}$$

In order to reduce the correlation between the adjacent clip, we consider a stride with half of the clip length in every case from 20 to 32.

3.3. Architecture of the Proposed System

The overall architecture of the proposed system is shown in Figure 3. There are two basic parts in the proposed method: (1) pre-processing and (2) classification. We design a simple DCNN model and also conduct ResNet18 and MobileNetV2 for further evaluations and comparisons. The interpretation of the preprocessing referred to previously in Section 3.2 comprises the processing of skeleton data and the preparation of training and testing datasets.

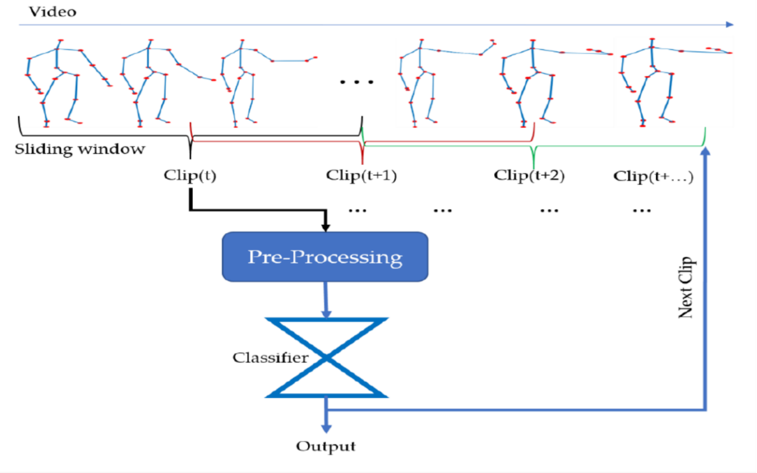


Figure 3. Block diagram of the proposed method.

In Figure 3, the clip at time t is first passed through the preprocessor, and then the preprocessed input is sent to the classifier for discriminating among the classes. This process continues until the input is finished. There are six main blocks in the classifier networks, as shown in Figure 4. The first layer is the input layer. The next four blocks include convolutions, batch normalizations, and ReLU, followed by max-pooling layers. The fifth block is integrated with a fully connected layer, batch normalization, ReLU, and dropout. The final block consists of a fully connected layer and a softmax layer that produces classification output.

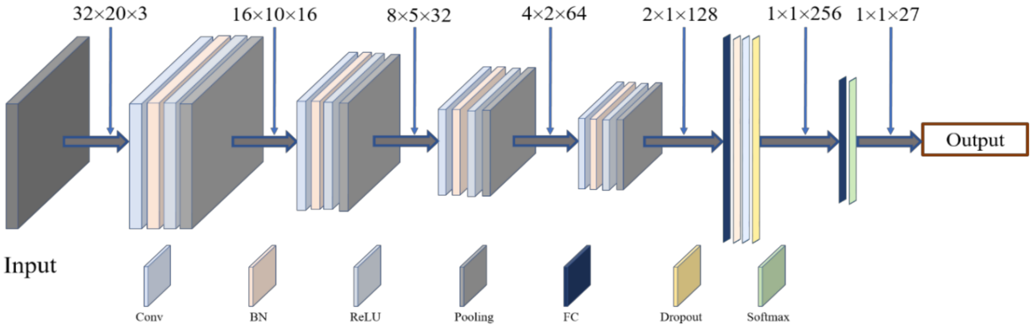


Figure 4. The designed deep convolutional neural network (DCNN) model for the purposes of classification.

The dimension of the input is L × 20 × 3 along X, Y, and Z axes, where L represents clip length, and 20 × 3 represents the number of joints along X, Y, and Z axes. Here, we briefly describe the inputs and outputs for clip length 32. The input is first passed through a convolution layer, having 3×3 kernel, 16 learnable filters, same padding, and 1 stride that produces output of size 32 × 20 × 16 by following batch normalization and ReLU. The output is passed through a max-pooling layer, having 2 × 2 kernel and 2 stride that produces output of size 16 × 10 × 16. The learnable parameters weights and bias for the first convolution layer are 3 × 3 × 3 × 16, and 1 × 1 × 16, respectively. The above operations continue four times and generate output of size 2 × 1 × 128. In every convolution's layer of four blocks, the learnable parameters (weights and bias) are 3 × 3 × input_channels × output_channels and 1 × 1 × output_channels, respectively. The batch normalization, having learnable parameters (offsets, bias), are also defined as 1 × 1 × output_channels and 1 × 1 × output_channels, respectively. Then, the generated output of size 2 × 1 × 128 passes through a fully connected layer by following batch normalization, ReLU, and dropout layers of 50% that produce output of size 1 × 1 × 256. The learnable parameters (weights, bias) of fully connected layers are 256 × 256 and 256 × 1. In our system, the total trainable parameters are 0.171 (M) and the non-trainable parameter is 0. The second fully connected layer produces the final outputs by following a softmax layer. For purpose of optimization, we consider stochastic gradient descent (SGD) techniques.

4. Experimental Results

This section includes experimental setup, performance evaluation, and state-of-the-arts comparison.

4.1. Experimental Setup

4.1.1. Dataset

In the overall experiments, we consider two popular skeleton datasets: UTD-MHAD [34] and MSR-Action3D [35].

UTD-MHAD is provided by the ESSP Laboratory at the University of Texas at Dallas. The dataset is captured using a Microsoft Kinect sensor in an indoor environment. It contains 27 different classes of skeleton data in which each frame has 20 joints along X, Y, and Z axes. The 27 classes of actions are performed by 8 different subjects including four females and four males. Each class has 32 videos, except three of them, which are corrupted, making a total of 861 videos of skeleton data. Most of the action is performed by hands, for instance swipe left, swipe right, wave, clap, throw, arm cross, basketball shoot, draw x, draw circle, draw triangle, bowling, boxing, baseball swing, tennis swing, arm curl, tennis serve, push, knock, catch, pickup, and throw. Some of the actions are also captured by legs, such as jogging, walking, and lunging. There are only two actions that are acted by the whole body. Each person repeated each action four times in every class. We consider the dataset obtained by the first five subjects for training and three subjects for testing.

MSR-Action3D was created by Wanqing Li during his time at Microsoft Research Redmond. It contains 20 different classes of actions performed by 10 different subjects, which are repeated 3 times. Some corrupted files are discarded, and a total of 567 sequences are used for the experiments. MSR-Action3D also contains 20 joints coordinate values similar to UTD-MHAD. This dataset consists of actions such as high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw cross, draw tick, draw circle, hand clap, two-hand wave, side-boxing, bend, forward kick, side-kick, jogging, tennis swing, tennis serve, golf swing, and pick up throw. The first six subjects are used for training and the remaining four subjects for testing.

4.1.2. System Specification

The hardware and software used for our experiments are listed in Table 1. For training the proposed method, we initially set the learning rate to 0.001, and the learning rate dropping factor to 0.5, which decreased after every 5 epochs. We trained the system until the completion of 20 epochs. To illustrate more about the effectiveness of the proposed system, we trained and tested the system using clip length 20 to 32. Details of the training and testing configurations are listed in Table 2.

Table 1. Hardware and software requirements for carrying out the overall experiments.

Hardware Requirements	Software Requirements
Intel Core i7	Ubuntu 16.04, MS Office
NVIDIA Graphics Card	MATLAB2019b
Microsoft Kinect	Python2.7

Table 2. Training and testing configurations for extracting and discriminating the actions.

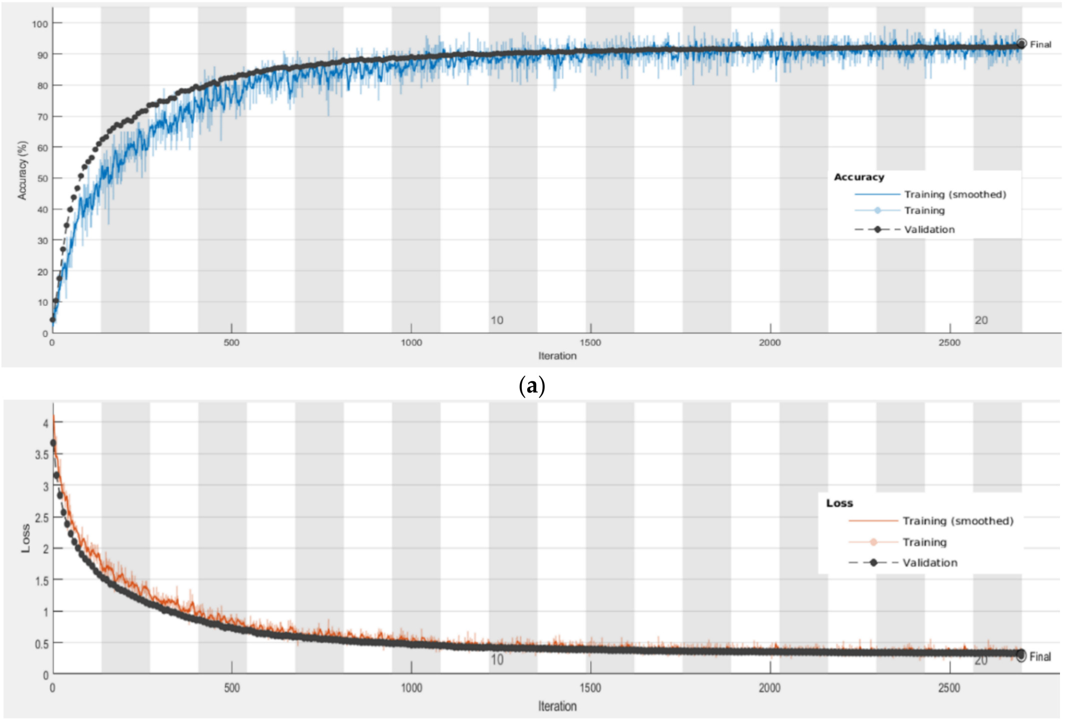
Parameters	Values
Clip length	20, 22, . . . , 32
Number of epochs	20
Initial learning rate	0.001
LearningRateDropFactor	0.5
LearningRateDropPeriod	5

4.2. Performance Evaluation

The experimental results are shown in terms of accuracy given by the following equations:

$$\text{Accuracy}(\%) = \frac{\text{Total Correctly Predicted Observations}}{\text{Total Number Observations}} \times 100 \tag{7}$$

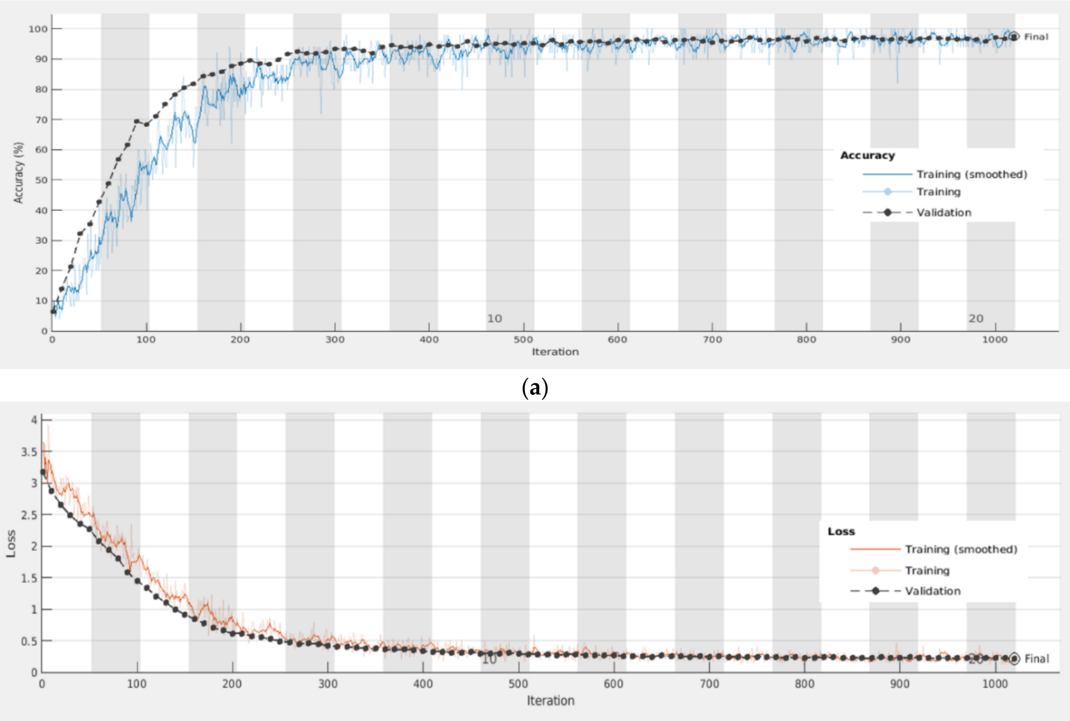
We got the highest training and testing results for clip length 32, which are used for further illustration in the next subsections. To show the training performance of the proposed system, the training accuracy and loss for clip length 32 is depicted in Figures 5 and 6. Figure 5a,b represents the accuracy and loss of the UTD-MHAD dataset, while the accuracy and loss of MSR-Action3D dataset is visualized in Figure 6a,b. From Figures 5 and 6, it is clear that training accuracies are increasing and losses are reducing at a noticeable rate in every epoch.



(**a**)

(**b**)

Figure 5. Training results of UTD-MHAD dataset: (**a**) accuracy and (**b**) loss.



(**a**)

(**b**)

Figure 6. Training results of MSR-Action3D dataset; (**a**) accuracy and (**b**) loss.

We evaluated the test results for two different datasets. In the Table 3, we display the clip length and testing results for our designed model in terms of accuracies for both UTD-MHAD and MSR-Action3D datasets. For the UTD-MHAD dataset, we achieved higher accuracies of approximately 95.01% using clip length 32, while the lowest accuracy was recorded for clip length 20 (87.42%). A similar trend can be noticed for the MSR-Action3D dataset in which we obtain about 89.62% and 95.36%, using clip length 20 and 32, respectively.

Table 3. Classification results of UTD-MHAD and MSR-Action3D skeleton datasets.

Clip Length	UTD-MHAD	MSR-Action3D
20	87.42%	89.62%
22	89.21%	91.96%
24	90.86%	93.44%
26	91.84%	93.71%
28	93.33%	94.84%
30	94.58%	95.06%
32	95.01%	95.36%

Discrimination accuracies improved with increases in clip length. This is because a larger clip length can capture more temporal information, which significantly affects the performance of the classifier.

4.3. State-of-the-Art Comparisons

In order to clarify the effectiveness and robustness of the proposed system, we compare the results with four related methods [19–22] that used UTD-MHAD, as described in Section 2.2. The results are shown in Table 4 in which the first column represents the methods' names, and the second column represents accuracies. From Table 4, we can see that the proposed method achieves higher performance compared with the reference methods, with accuracies of about 93.42%, 95.88%, and 95.01% for

MobileNetV2, ResNet18, and our model, respectively, while the method [17] had 93.26% at its peak. Thus, it can be concluded that the proposed method outperforms MHAD-UTD 3D skeleton datasets when classifying human actions in 27 actions classes.

Table 4. Comparisons of the classification accuracy of existing systems for the UTD-MHAD dataset.

Methods	Accuracy
Ours	95.01%
ResNet18	95.88%
MobileNetV2	93.42%
Ref. [19]	93.26%
Ref. [20]	88.10%
Ref. [21]	86.97%
Ref. [22]	85.81%

We also conducted a comparison of the proposed method using an MSR-Action3D dataset with the existing methods referred to in [27–29], as listed in Table 5. From Table 5, we can say that our method outperforms the prior works with accuracies of about 93.69%, 96.28%, and 95.36% using MobileNetV2, ResNet18, and our models with clip length 32.

Table 5. Comparisons of classification accuracy with the existing systems for the MSR-Action3D dataset.

Methods	Accuracy
Ours	95.36%
ResNet18	96.28%
MobileNetV2	93.69%
Ref. [27]	93.63%
Ref. [28]	93.09%
Ref. [29]	91.70%

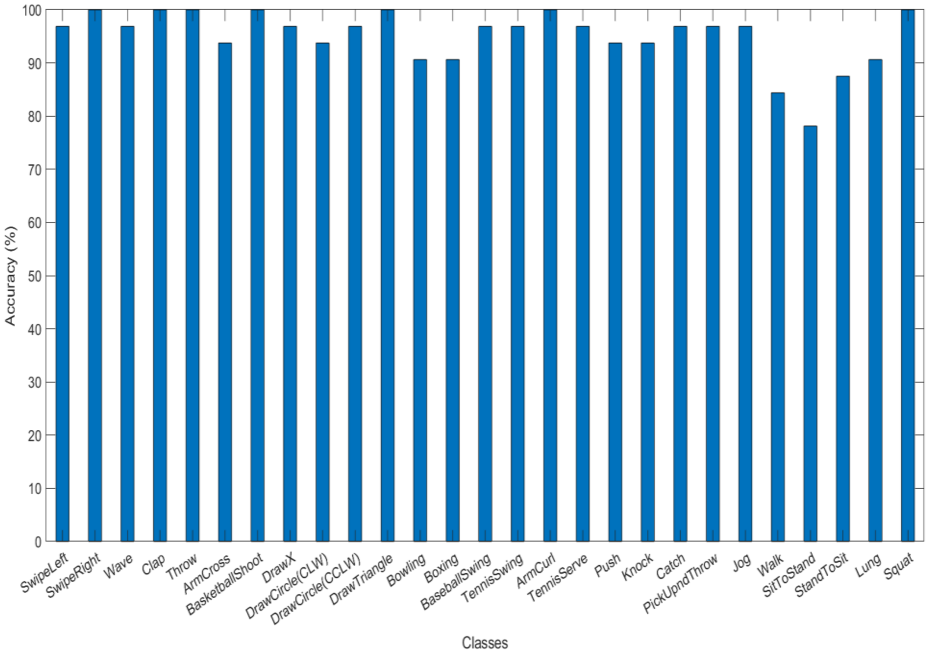
We obtained the highest accuracy of about 95.88% and 96.28% using ResNet18 with clip length 32 for UTD-MHAD and MSR-Action3D datasets, which are very close to the results of 95.01% and 95.36% achieved using our model. The number of parameters and floating point operations per seconds (FLOPs) is much higher in ResNet18 compared with MobileNetV2 and our model. According to the classification accuracies, it can be said that our model has similar classification performance to ResNet18 but is more efficient in terms of complexity.

To illustrate the complexity of the models used for the classification of actions, we calculated the number of parameters and floating point operations per seconds (FLOPs) in millions (M) as listed in Table 6. The number of parameters and FLOPs is much lower in our model compared with the ResNet18 and MobileNetV2.

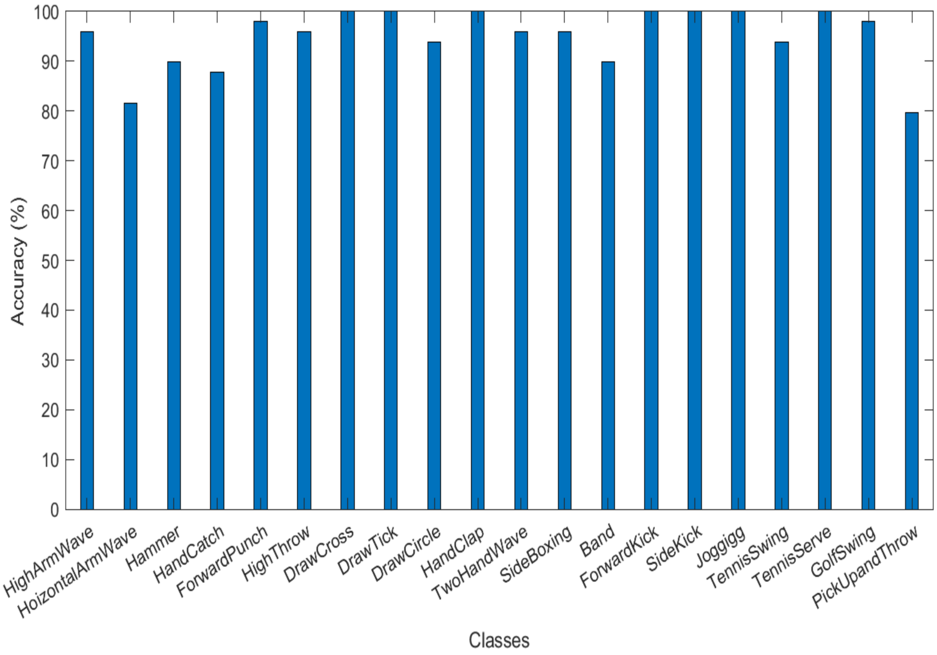
Table 6. Model complexity.

Model	Parameters (M)	FLOPs (M)
ResNet18	11.69	31.6
MobileNetV2	3.505	8.3
Ours	0.171	2.6

To examine the classification accuracy of each class, we used a bar chart to visualize the classification accuracy versus class names. The individual class results are shown in Figure 7a,b for UTD-MHAD and MSR-Action3D datasets, respectively.



(**a**)



(**b**)

Figure 7. Individual class classification results: (**a**) UTD-MHAD and (**b**) MSR-Action3D datasets.

5. Conclusions and Future Work

We propose a new method for human action recognition using well-known 3D skeleton datasets (UTD-MHAD and MSR-Action3D) based on deep learning. The 3D skeleton data are first analyzed along with the X, Y, and Z axes, and then subtraction is conducted between the current and next frames for capturing temporal changes. The results of the subtraction are considered as input to the network for classification. Instead of using whole sequences as input to the DCNN, we consider a portion of frames ranging from 20 to 32 and perform the classification. All the experiments are done with separate training and testing datasets. The results described in the experimental section show better performance in the discrimination of human actions using a skeleton dataset compared with existing systems. Thus, we can conclude that our proposed system outperforms state-of-the-art methods. We also conduct prior, well known DCNN models such as ResNet18 and MobileNetV2 to show the effectiveness of the proposed system. Since the UTD-MHAD and MSR-Action3D skeleton datasets contain only the actions data, we were not able to perform detection operations with the classification. In further implementation, we will try to consider a dataset that contains both actions and no actions data so that we can conduct both detection and classification.

Author Contributions: Conceptualization, analysis, methodology, manuscript preparation, and experiments, N.T.; data curation, writing—review and editing, N.T., M.M.I. and J.-H.B.; supervision, J.-H.B.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the GRRC program of Gyeonggi province [GRRC Aviation 2017-B04, Development of Intelligent Interactive Media and Space Convergence Application System].

Acknowledgments: We would like to acknowledge Korea Aerospace University with much appreciation for its ongoing support to our research.

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

References

1. Köpüklü, O.; Gunduz, A.; Kose, N.; Rigoll, G. Real-time hand gesture detection and classification using convolutional neural networks. In Proceedings of the 14th IEEE International Conference on Automatic Face & Gesture Recognition, Lille, France, 14–18 May 2019; pp. 1–8.
2. Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; Kautz, J. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4207–4215.
3. Zou, Q.; Wang, Y.; Wang, Q.; Zhao, Y.; Li, Q. Deep Learning-Based Gait Recognition Using Smartphones in the Wild. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3197–3212. [CrossRef]
4. Wu, Z.; Huang, Y.; Wang, L.; Wang, X.; Tan, T. A comprehensive study on cross-view gait based human identification with deep cnns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 209–226. [CrossRef] [PubMed]
5. Farooq, A.; Won, C.S. A survey of human action recognition approaches that use an RGB-D sensor. *IEIE Trans. Smart Process. Comput.* **2015**, *4*, 281–290. [CrossRef]
6. Chen, C.; Jafari, R.; Kehtarnavaz, N. Action recognition from depth sequences using depth motion maps-based local binary patterns. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5–9 January 2015; pp. 1092–1099.
7. Trelinski, J.; Kwolek, B. Convolutional Neural Network-Based Action Recognition on Depth Maps. In Proceedings of the International Conference on Computer Vision and Graphics, Warsaw, Poland, 17–19 September 2018; pp. 209–221.
8. Wang, P.; Li, W.; Gao, Z.; Zhang, J.; Tang, C.; Ogunbona, P.O. Action recognition from depth maps using deep convolutional neural networks. *IEEE Trans. Hum. Mach. Syst.* **2015**, *46*, 498–509. [CrossRef]
9. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in video. *Adv. Neural Inf. Process. Syst.* **2014**, *1*, 568–576.
10. Dollar, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. In Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, China, 15–16 October 2005; pp. 65–72.
11. Wu, D.; Shao, L. Silhouette analysis-based action recognition via exploiting human poses. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 236–243. [CrossRef]
12. Ahmad, M.; Lee, S.W. HMM-based human action recognition using multiview image sequences. In Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, China, 20–24 August 2006; pp. 263–266.
13. Xia, L.; Chen, C.C.; Aggarwal, J.K. View invariant human action recognition using histograms of 3d joints. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 20–27.

14. Luo, J.; Wang, W.; Qi, H. Spatio-temporal feature extraction and representation for RGB-D human action recognition. *Pattern Recognit. Lett.* **2014**, *50*, 139–148. [CrossRef]
15. Megavannan, V.; Agarwal, B.; Babu, R.V. Human action recognition using depth maps. In Proceedings of the IEEE International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 22–25 July 2012; pp. 1–5.
16. Imran, J.; Raman, B. Evaluating fusion of RGB-D and inertial sensors for multimodal human action recognition. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 189–208. [CrossRef]
17. Li, C.; Zhong, Q.; Xie, D.; Pu, S. Skeleton-based action recognition with convolutional neural networks. In Proceedings of the IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 597–600.
18. Du, Y.; Fu, Y.; Wang, L. Skeleton based action recognition with convolutional neural network. In Proceedings of the IEEE 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 579–583.
19. Chen, Y.; Wang, L.; Li, C.; Hou, Y.; Li, W. ConvNets-based action recognition from skeleton motion maps. *Multimed. Tools Appl.* **2020**, *79*, 1707–1725. [CrossRef]
20. Li, C.; Hou, Y.; Wang, P.; Li, W. Joint distance maps-based action recognition with convolutional neural networks. *IEEE Signal Process. Lett.* **2017**, *24*, 624–628. [CrossRef]
21. Hou, Y.; Li, Z.; Wang, P.; Li, W. Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 807–811. [CrossRef]
22. Wang, P.; Li, P.; Hou, Y.; Li, W. Action recognition based on joint trajectory maps using convolutional neural networks. In Proceedings of the 24th ACM international conference on ACM Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 102–106.
23. Rashmi, M.; Guddeti, R.M.R. Skeleton based Human Action Recognition for Smart City Application using Deep Learning. In Proceedings of the International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, 7–11 January 2020; pp. 756–761.
24. Huynh-The, T.; Hua, C.H.; Ngo, T.T.; Kim, D.S. Image representation of pose-transition feature for 3D skeleton-based action recognition. *Inf. Sci.* **2020**, *513*, 112–126. [CrossRef]
25. Si, C.; Jing, Y.; Wang, W.; Wang, L.; Tan, T. Skeleton-Based Action Recognition with Hierarchical Spatial Reasoning and Temporal Stack Learning Network. *Pattern Recognit.* **2020**, *107*, 107511. [CrossRef]
26. Li, Y.; Xia, R.; Liu, X. Learning shape and motion representations for view invariant skeleton-based action recognition. *Pattern Recognit.* **2020**, *103*, 107293. [CrossRef]
27. Yang, Y.; Deng, C.; Gao, S.; Liu, W.; Tao, D.; Gao, X. Discriminative multi-instance multitask learning for 3D action recognition. *IEEE Trans. Multimed.* **2017**, *19*, 519–529. [CrossRef]
28. Yang, X.; Tian, Y. Super normal vector for activity recognition using depth sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014.
29. Zanfir, M.; Leordeanu, M.; Sminchisescu, C. The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection. In Proceedings of the International Conference on Computer Vision, Sydney, Australia, 3–6 December 2013.
30. Straka, M.; Hauswiesner, S.; Rüther, M.; Bischof, H. Skeletal Graph Based Human Pose Estimation in Real-Time. In *BMVC*; Graz University of Technology: Graz, Austria, 2011; pp. 1–12.
31. Sapiński, T.; Kamińska, D.; Pelikant, A.; Anbarjafari, G. Emotion recognition from skeletal movements. *Entropy* **2019**, *21*, 646. [CrossRef]
32. Filntisis, P.P.; Efthymiou, N.; Koutras, P.; Potamianos, G.; Maragos, P. Fusing Body Posture With Facial Expressions for Joint Recognition of Affect in Child–Robot Interaction. *IEEE Robot. Autom. Lett.* **2019**, *4*, 4011–4018. [CrossRef]
33. Raptis, M.; Kirovski, D.; Hoppe, H. Real-time classification of dance gestures from skeleton animation. In Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Vancouver, BC, Canada, 5 August 2011; pp. 147–156.

34. Chen, C.; Jafari, R.; Kehtarnavaz, N. UTD-MHAD: A Multimodal Dataset for Human Action Recognition Utilizing a Depth Camera and a Wearable Inertial Sensor. In Proceedings of the IEEE International Conference on Image Processing, Quebec City, QC, Canada, 27 September 2015; pp. 168–172.
35. Li, W.; Zhang, Z.; Liu, Z. Action recognition based on a bag of 3D points. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 9–14.