applied sciences

MDPI

Article

NPU RGB+D Dataset and a Feature-Enhanced LSTM-DGCN Method for Action Recognition of Basketball Players

Chunyan Ma *, Ji Fan, Jinghao Yao and Tao Zhang

School of Software, Northwestern Polytechnical University, 1 Dongxiang Road, Chang'an District, Xi'an 710129, China; 2019203480@mail.nwpu.edu.cn (J.F.); yjh_2019213633@mail.nwpu.edu.cn (J.Y.); tao_zhang@nwpu.edu.cn (T.Z.)

* Correspondence: machunyan@nwpu.edu.cn

Featured Application: Action classification definitions and this NPU RGB+D dataset for basketball players in accordance with standardized basketball rules will enable researchers to apply, develop, and adapt to various data-driven technologies for the action recognition of basketball players. The feature-enhanced LTSM-DGCN outperformed the most advanced general action recognition method across multiple evaluation criteria on our NPU RGB+D dataset. Our method helps improve the level of intelligent service for the training and competition of basketball.

Abstract: Computer vision-based action recognition of basketball players in basketball training and competition has gradually become a research hotspot. However, owing to the complex technical action, diverse background, and limb occlusion, it remains a challenging task without effective solutions or public dataset benchmarks. In this study, we defined 32 kinds of atomic actions covering most of the complex actions for basketball players and built the dataset NPU RGB+D (a large scale dataset of basketball action recognition with RGB image data and Depth data captured in Northwestern Polytechnical University) for 12 kinds of actions of 10 professional basketball players with 2169 RGB+D videos and 75 thousand frames, including RGB frame sequences, depth maps, and skeleton coordinates. Through extracting the spatial features of the distances and angles between the joint points of basketball players, we created a new feature-enhanced skeleton-based method called LSTM-DGCN for basketball player action recognition based on the deep graph convolutional network (DGCN) and long short-term memory (LSTM) methods. Many advanced action recognition methods were evaluated on our dataset and compared with our proposed method. The experimental results show that the NPU RGB+D dataset is very competitive with the current action recognition algorithms and that our LSTM-DGCN outperforms the state-of-the-art action recognition methods in various evaluation criteria on our dataset. Our action classifications and this NPU RGB+D dataset are valuable for basketball player action recognition techniques. The feature-enhanced LSTM-DGCN has a more accurate action recognition effect, which improves the motion expression ability of the skeleton data.

Keywords: basketball action recognition; RGB+D dataset; feature-enhanced LSTM-DGCN

Citation: Ma, C.; Fan, J.; Yao, J.; Zhang, T. NPU RGB+D Dataset and a Feature-Enhanced LSTM-DGCN Method for Action Recognition of Basketball Players. *Appl. Sci.* **2021**, *11*, 4426. https://doi.org/10.3390/app11104426

Academic Editors: Sławomir Nowaczyk, Mohamed-Rafik Bouguelia, Hadi Fanaee and Seokwon Yeom

Received: 18 March 2021
Accepted: 6 May 2021
Published: 13 May 2021

1. Introduction

With the development of computer vision and deep learning technology, the applications of depth cameras in human–computer interactions, automatic driving, virtual reality, and so on have increased in number. One of the most popular application scenarios is the use of 3D vision for human action recognition and behavior analysis [1]. Human action recognition datasets in RGB image data and Depth data (RGB+D) format have also emerged [2]. However, only a few large-scale open RGB+D datasets are dedicated to basketball to support research on basketball player action recognition. As a result, we cannot build a benchmark that can evaluate and compare basketball player action recognition methods. Recent data-driven technology, such as deep learning methods, has a high

demand for data. Nowadays, almost all related datasets have limitations in the following aspects.

First, a huge gap exists between general human action recognition datasets and professional basketball action recognition datasets. Although many large human action recognition datasets contain basketball action data such as dribbling and shooting, as is shown in [3], they are far from adequate to support data-driven methods. To an extent, some daily actions are similar to those of basketball players, such as walking and dribbling and carrying and holding, but the relationships are rare and far-fetched. In addition, the introduction of nonstandard actions as sample data can negatively influence the result of action recognition. Second, no absolute action standard exists in basketball, and all basketball actions that conform to the rules are allowed. For example, different basketball players can choose their own habitual shooting action according to their own situations to help improve the hit rate, so the shooting actions of different basketball players may vary greatly. In other words, the actions of basketball players often have the characteristics of their own styles [4]. Therefore, the classification of basketball players' actions is very challenging, so as much video data as possible from different players is necessary to improve the reliability and versatility of the dataset. In addition, different actions may have the same pattern, such as dunking, layups, passing, and catching, so complete sequences and accurate marking are important. Third, the deficiency of basketball player action datasets restricts the application of the most advanced data-driven learning methods to the problem of basketball action recognition. The superiority of the deep learning algorithms in the field of action recognition has been fully described in [5]. So far, despite the many attempts, the problem of over fitting remains, so we have to narrow the range of learning parameters. We need more samples to train a classification network or model.

To overcome these difficulties, we defined 32 kinds of atomic actions that can cover most of the complex basketball actions and constructed a new large-scale dataset for a 3D basketball player action analysis. This dataset is composed of RGB+D videos of 12 basketball actions performed by 10 subjects captured by the ZED Stereo Camera launched by Stereolabs. We processed the original 3D video data and obtained depth maps, left and right view frame sequences, and 3D coordinates of 25 key points of the human body. In our dataset, basketball players of different ages and heights performed dribbling (standing dribble, moving dribble, front dribble, cross-leg dribble, etc.), passing and catching (chest passing, overhead passing, chest catching, overhead catching, etc.), and shooting (one-hand shooting, chest shooting, overhead shooting, etc.). The 12 kinds of actions in our dataset cover most of the common technical actions in basketball games, which makes the dataset abundant and diverse. Owing to the uncontrollability of outdoor court background and lighting, our dataset was collected in the standard indoor basketball court. We provide environmental inconsistency through video collection at different positions and angles of the court. Changes in subjects and views make it possible to evaluate various 3D-based action analysis methods accurately across subjects and views. Our dataset can be widely used in the field of basketball players' action recognition and motion analysis. It can help researchers who study basketball action recognition based on computer vision and deep learning move forward without the limitations of time and conditions. Our dataset also makes the application of data-driven methods such as deep learning technology possible.

Another contribution was inspired by the application of graph convolution neural networks in the field of skeleton-based human action recognition [6]. We propose a feature-enhanced LSTM-DGCN model based on the deep graph convolutional network (DGCN) and long short-term memory (LSTM) methods for basketball action recognition. This model combines a feature extraction technique and the data-driven deep learning method. Compared with the current mainstream skeleton-based motion recognition methods, which use joint coordinates directly, we use the feature extraction module to filter the effective basketball motion features and calculate the corresponding joint distance, joint angle, and other eigenvalues before inputting the end-to-end model. In so doing, we achieve targeted

and efficient motion recognition. According to small numbers of samples and parameters, we can achieve an enhanced classification effect.

In summary, the main contributions of this study are as follows: (1) defining action classification and introducing the NPU RGB+D dataset with RGB image data and Depth data captured in Northwestern Polytechnical University for basketball action recognition; (2) proposing the feature-enhanced LSTM-DGCN algorithm, which is used to classify and identify basketball players' actions; and (3) conducting an experiment with several state-of-the-art action recognition methods to evaluate and set the benchmarks on the NPU RGB+D dataset.

The rest of this paper is organized as follows. Section 2 explores the current activity analysis datasets and action recognition methods. Section 3 defines action classification and introduces the proposed dataset, its structure, and evaluation criteria for basketball players. Section 4 presents the feature-enhanced LSTM-DGCN method. Section 5 shows the experimental evaluations of state-of-the-art action recognition methods alongside the proposed feature-enhanced LSTM-DGCN method on our benchmark. Section 6 discusses our method, and Section 7 concludes the study.

2. Related Work

In this section, we briefly introduce several open datasets of human activity analysis. According to the different types of research data, we introduce human action recognition methods based on RGB video data, RGB+D video and depth maps, skeleton data, and sensor data. We then analyze their application prospects in the field of basketball action recognition one by one.

2.1. Activity Analysis Datasets

2.1.1. Human Action Datasets

The UCF101 dataset [7] is a classic RGB action recognition dataset which contains 101 action categories (diving, typing, biking, basketball, etc.) and 13,320 videos collected from YouTube. Researchers have been studying this dataset for a long time, and the reported results of this dataset have already reached very high accuracy [8,9].

The MSR-Action3D dataset [10] is one of the earliest depth-based action recognition datasets. It records 20 kinds of actions (forward punch, high throw, hand clap, bend, etc.) and 10 subjects. It provides a total of 567 depth map sequences and corresponding skeleton coordinates. It is often regarded as a challenging common benchmark dataset in the field of action recognition based on skeleton depth data [11,12].

The NTU RGB+D dataset [13] is one of the most popular public human action recognition datasets. It contains 60 action classes performed by 40 distinct subjects in total, which are divided into three groups: 40 daily actions (drinking, eating, reading, etc.), 9 health-related actions (sneezing, staggering, falling down, etc.), and 11 mutual actions (punching, kicking, hugging, etc.). State-of-the-art action recognition methods [14,15] often use this dataset as a benchmark.

Basketball action recognition is a sub-task of human action recognition. All of the above-mentioned datasets are famous human action recognition datasets, which cover a variety of data modalities, and they have greatly contributed to human behavior analysis. However, they only contain small amounts of basketball action data, which is inadequate to support the related research.

2.1.2. Basketball Action Datasets

The NCAA basketball dataset [16] was collected from 257 NCAA basketball games available on YouTube. The dataset is manually divided into 11 types of activities: three-point success, three-point failure, two-point success, two-point failure, free-throw success, free-throw failure, layup success, layup failure, slam dunk success, slam dunk failure, and steals. Using the NCAA basketball dataset as a benchmark dataset for player detection or event detection is reasonable [17,18].

The APIDIS basketball dataset [19] was constructed from a basketball game. A unique server captured frames from seven color cameras. All basketball events (shooting, holding, violation, etc.) were manually annotated, and the basketball court measurements, event annotations, and camera calibration data are available in the dataset. This dataset is a complete recording of a basketball game which contains multiple players, so it is suitable as a benchmark for the research on player tracking [20,21].

The basketball datasets listed above are all from basketball game videos. Basketball game videos often contain multiple basketball players, so confusion of the background and body occlusion are inevitable. In addition, the definitions of action categories in the existing basketball action datasets are often confusing and vague, lacking the theoretical basis of basketball rules and the guidance of professional coaches. As such, these datasets do not qualify as benchmarks for research on basketball player action recognition.

As shown in Table 1, our dataset has advantages over the existing datasets in basketball action categories, views, and data modalities. The advantages are summarized as follows: (1) accurate and reasonable classification of multiple basketball actions, (2) more samples for each action class, (3) more intra-class variations (interacted objects, age of actors, etc.), and (4) depth map and 3D joint point data modalities.

Table 1. A comparison between the NPU RGB+D dataset and several publicly available datasets for action analysis.

	Samples	Basketball Action Classes	Subjects	Views	Modalities
UCF101	13,320	2	-	-	RGB
MSR-Action3D	567	0	10	1	DepthMap + Skeleton
NTU RGB+D	56,880	0	40	80	RGB + DepthMap + Skeleton
NCAA	257	11	-	1	RGB
APIDIS	1	10	-	7	RGB
NPU RGB+D	2169	12	10	5	RGB + DepthMap + Skeleton

In the process of research, we conducted a number of retrievals from UCI Machine Learning Repository, Kaggle Datasets, and Google Datasets. We studied the data modalities, action classifications, and data sizes of the relevant vision-based basketball datasets from the above public datasets, and listed them in Table 1. Other basketball datasets are not listed in Table 1 because they only contain sensor data or game scoring data.

For five kinds of datasets in Table 1:

1. UCF101 and MSR-Action3D are human action recognition datasets. UCF101 does not contain skeleton data so we could not evaluate our method on it. MSR-Action3D contains skeleton data but there are not actions similar to basketball players' actions so we could not evaluate our method on it.
2. NCAA Dataset from Kaggle Datasets and APIDIS Dataset from Google Datasets contain basketball action image data. They contain multiple basketball players and not contain the skeleton data that we need as the input of LSTM-DGCN, so we could not evaluate our method on them.
3. We selected basketball actions and daily actions similar to 12 basic atomic basketball actions from the NTU RGB+D human action recognition dataset to construct the mapping between them, such as walking and moving dribbling, throwing, and passing. Our LSTM-DGCN also achieved the same results with it as the most advanced human action recognition algorithm 2s-AGCN.

We propose NPU RGB+D as a benchmark for basketball action recognition research. Compared with other basketball datasets, it has more data modalities and clearer action classification. It can support our research of skeleton-based basketball action recognition

method but can also support the research of basketball action recognition methods based on RGB video, depth maps, and RGB+D video.

We agree that a study of this type would require a larger dataset. We choose 12 basic atomic actions for experiments to give a benchmark dataset and verify the effectiveness of our method. Compared with large public datasets listed in Table 1, we already have enough data for each kind of basketball action to verify the effectiveness of our method. UCF101 and NTU RGB+D have large amounts of data, mainly because they contain more types of actions. For example, NTU RGB+D contains 60 kinds of actions, with a total of 56,880 samples.

According to Table 1, our NPU RGB+D dataset has abundant data of basketball actions. For each action class, we have enough data to do the research on basketball action recognition. It also has the richest data modalities, including RGB sequences, depth maps, and skeleton coordinates. Our dataset contains video data of 10 professional basketball players performing 3–5 times in five positions. We think it has enough scale to study 12 basic atomic basketball actions.

2.2. Action Recognition Methods

2.2.1. RGB Video-Based Methods

RGB video-based methods have been very common in recent years owing to abundant datasets. The traditional methods mainly include the method based on the spatio-temporal interest points [22] and the method based on trajectory [23]. The former extracts the region features with violent changes in the spatio-temporal dimension, whereas the latter uses the key points of human bones or the action trajectories of joints to represent the actions. Dai et al. [24] proposed a two-stream convolution neural network for action classification. One stream obtains frame sequence as the input of 3D convolution residual network, and the other stream extracts action features on the basis of optical flow information. Each stream trains independently, and finally the prediction results of the two flows are fused together. Some attempts at basketball action recognition based on RGB video were also carried out. For example, Chen et al. [25] designed an encoder–decoder framework based on convolutional neural networks (CNN) and LSTM by extracting court boundary line and video keyframe, and they realized action recognition and prediction. Pan et al. [26] collected a dataset containing a large number of basketball video clips from YouTube. Then, the Gaussian Mixed Model (GMM) algorithm was used to extract moving features of motion blocks, and the gradient histogram was used to represent the posture descriptors of basketball actions. Finally, through the linear combination of motion and posture descriptors, the K-Nearest Neighbor (KNN) algorithm was used to realize basketball action recognition.

The results of these methods are closely related to the quality of videos and the choice of descriptors, so RGB video-based methods are generally not robust enough.

2.2.2. RGB+D Video-Based Methods

With the development of deep vision, the technology of human action recognition in RGB+D video has also made a big step forward. Differently from RGB video, RGB+D video can provide supplementary information for 3D vision data, so the accuracy and robustness of action recognition can be improved. Chalavadi et al. [27] proposed a method to classify and recognize human actions in RGB+D video by extracting the action sequence features from RGB and binary depth video streams and training the CNN model. Mukherjee et al. [28] used two parallel ResNeXt-101 to generate dynamic images for RGB video and depth video to eliminate unnecessary background information. The method based on RGB+D video is a major breakthrough in depth-based action recognition technology.

2.2.3. Skeleton-Based Methods

RGB+D data contain more information, which is conducive to more accurate action recognition. However, more background noise is introduced into the samples. To solve

this problem, Vemulapalli et al. [29] proposed for the first time to extract human skeleton coordinates from RGB+D data, model them as curves in lie group, and combine them with dynamic time warping for action recognition. Du et al. [30] attempted to apply Recurrent Neural Network (RNN) to skeleton sequences to simulate the action trajectories of bones and joints in time series, and their work inspired the research on skeleton-based deep learning methods. Since Spatial Temporal Graph Convolutional Networks (ST-GCN) was proposed in [6], it has become one of the most popular methods in the field of human action recognition to process human skeleton data by the convolution neural network of a spatiotemporal graph. With ST-GCN as a basis, a series of action recognition methods, such as Action-structure Graph Convolutional Networks (AS-GCN) [31] and Two-Stream Adaptive Graph Convolutional Networks (2s-AGCN) [32], were proposed.

As previously mentioned, ST-GCN, AS-GCN, 2s-AGCN, and many other state-of-the-art skeleton-based human action recognition methods have achieved the leading results in the field of human action recognition, and they offer great inspiration for solving the problem of basketball action recognition. Most of these methods directly take the skeleton graph constructed from joint point coordinate data as input and use a graph convolution neural network to extract action features. In this way, end-to-end action recognition can be realized without human intervention. However, totally relying on a neural network to extract motion features from the 3D coordinates of all human joint points consumes a lot of computing resources and learning time, and learning noise data is difficult to avoid.

2.2.4. Sensor-Based Methods

The sensor-based action recognition method collects the velocity and direction data of different body parts of players and calculates the features through these data. These eigenvalues are then used to train the machine learning model to achieve action classification. There have also been many meaningful attempts in basketball action recognition. Sensor-based basketball action recognition methods can be represented by Daniel et al. [33]. Through the acceleration sensors, gyroscopes, and magnetic sensors worn on athletes, the body movement data of athletes in different action states were collected as action features, including speed, acceleration, angular velocity, and other data. Support vector machine (SVM) was used as the classification algorithm to identify walking, running, jump shots, and other actions. Similarly, Holzemann [34] used an acceleration sensor, a magnetic sensor, and a wrist gyroscope to obtain motion vector data and classified three different dribble modes according to KNN and random forest. The sensor-based basketball action recognition method is a relatively mature method, and it achieves good results. However, the application prospects for the sensor are limited because it may affect the normal performances of players. Nevertheless, it can save much computing power by pre-calculating features, and the model learning is more targeted and effective compared to training directly on unprocessed data.

Combining the advantages of the above two methods, we propose a feature-enhanced model named LSTM-DGCN that contains a feature extraction technique and a deep learning method. It can automatically extract effective features from the input joint coordinates through a built-in feature extraction module instead of directly inputting the skeleton graph as the direct feature of GCN. Then, it can select the most important joint angles and joint distances in basketball action recognition as the indirect feature input GCN and use LSTM to mine the temporal relationship between adjacent frames. Compared with the direct application of joint coordinates, LSTM-DGCN is more accurate and requires fewer samples.

3. Action Classification Definition and the NPU RGB+D Dataset for Basketball

3.1. Action Classification Definitions for Basketball

Technical actions in basketball are very complex and diverse, and proposing a standard action template to match all the actions of basketball players is impossible. Nevertheless, we can analyze the rules of basketball to get the common basic actions in basketball games.

With the analysis of the changes of basketball rules of FIBA in the recent 10 years [35] as a basis, we found that shooting, free throw, dribbling, and other common actions play important roles in basketball games. Under the guidance of professional basketball coaches, we defined the actions that do not depend on the semantics of the game environment and established the judgment standard of basketball action classification according to the spatial positions of players' body parts. The basketball actions are divided into three categories: movement, ball control, and shooting, with a total of 55 complex movements. We further decomposed 55 kinds of complex actions into 32 kinds of atomic action combination, simplified the action category, shortened the longer action sequences, and reduced the difficulty of action recognition.

In this paper, J represents the joint, followed by a number to indicate the joint number. For example, J1 represents the neck (The number 1 after J comes from joint 1 in Figure 1, and the following numbers after J all refer to Figure 1), J11 represents the right ankle, and the number of other joint points is shown in Figure 1. J is followed by a number in brackets, such as J(2–3), indicating the position between joint J2 and J3, that is, the position from the right shoulder to the right elbow. "∠" refers to the angle of the joint, followed by three joint numbers to indicate the angle formed by the three joints. For example, ∠2–3–4 refers to the angle formed by the right shoulder, right elbow, and right wrist. The joint number is followed by "x", "y", "z" to represent the x-direction coordinate value, y-direction coordinate value, and z-direction coordinate value of the joint, respectively. All the classification and judgment conditions of basketball complex actions and their atomic action decomposition are shown in Figure 2.

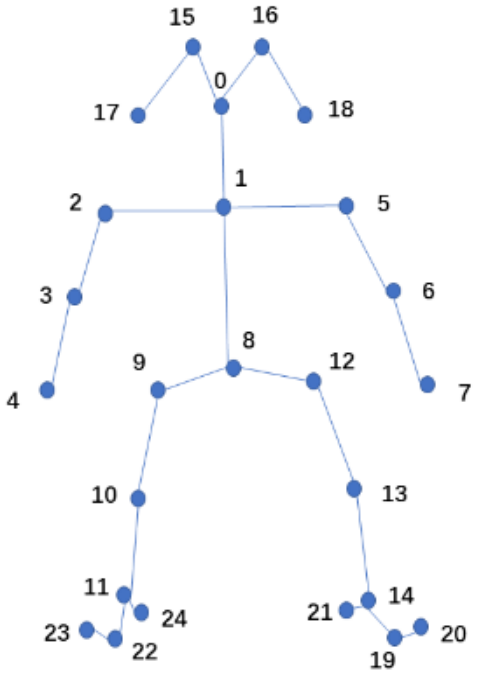
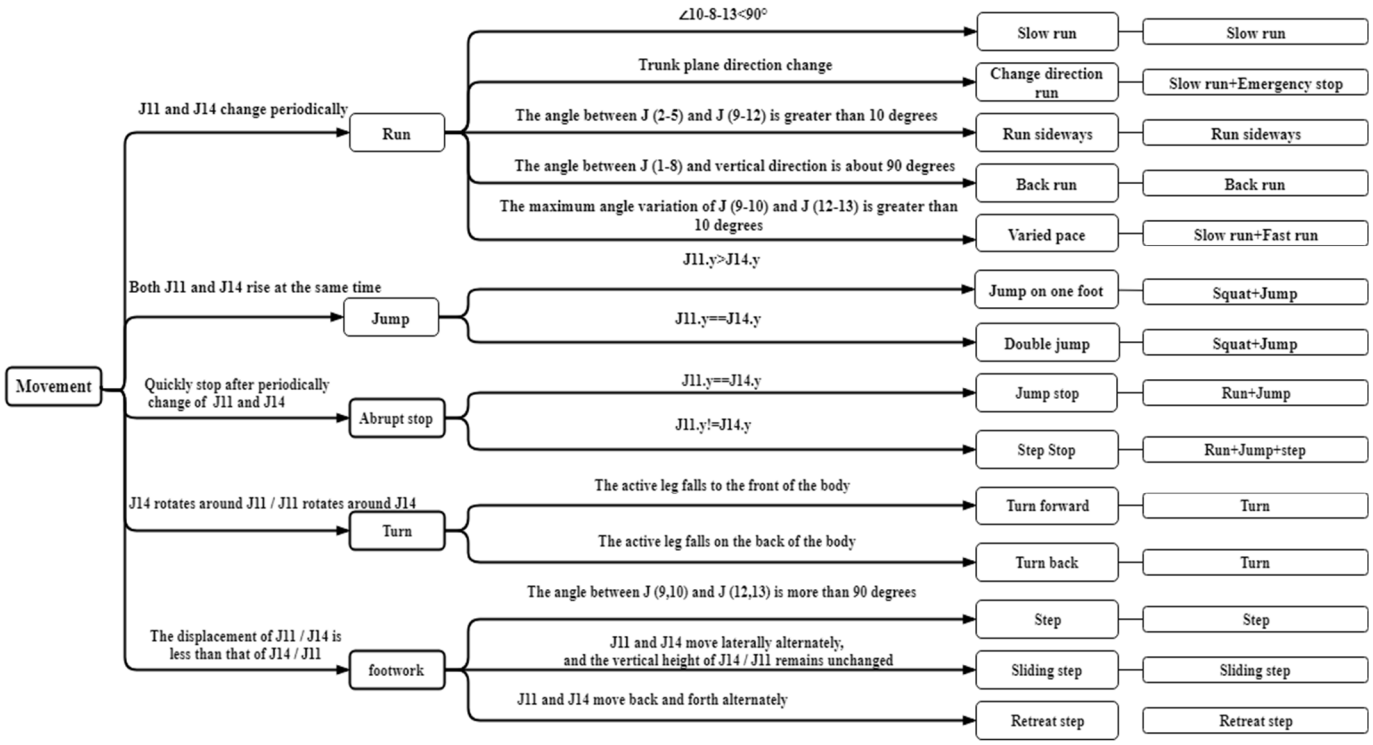


Figure 1. Configuration of 25 body joints in our dataset. The labels of the joints are: 0 = Nose, 1 = Neck, 2 = RShoulder, 3 = RElbow, 4 = RWrist, 5 = LShoulder, 6 = LElbow, 7 = LWrist, 8 = MidHip, 9 = RHip, 10 = RKnee, 11 = RAnkle, 12 = LHip, 13 = LKnee, 14 = LAnkle, 15 = REye, 16 = LEye, 17 = REar, 18 = LEar, 19 = LBigToe, 20 = LSmallToe, 21 = LHeel, 22 = RBigToe, 23 = RSmallToe, 24 = RHeel. R = right, L = left.

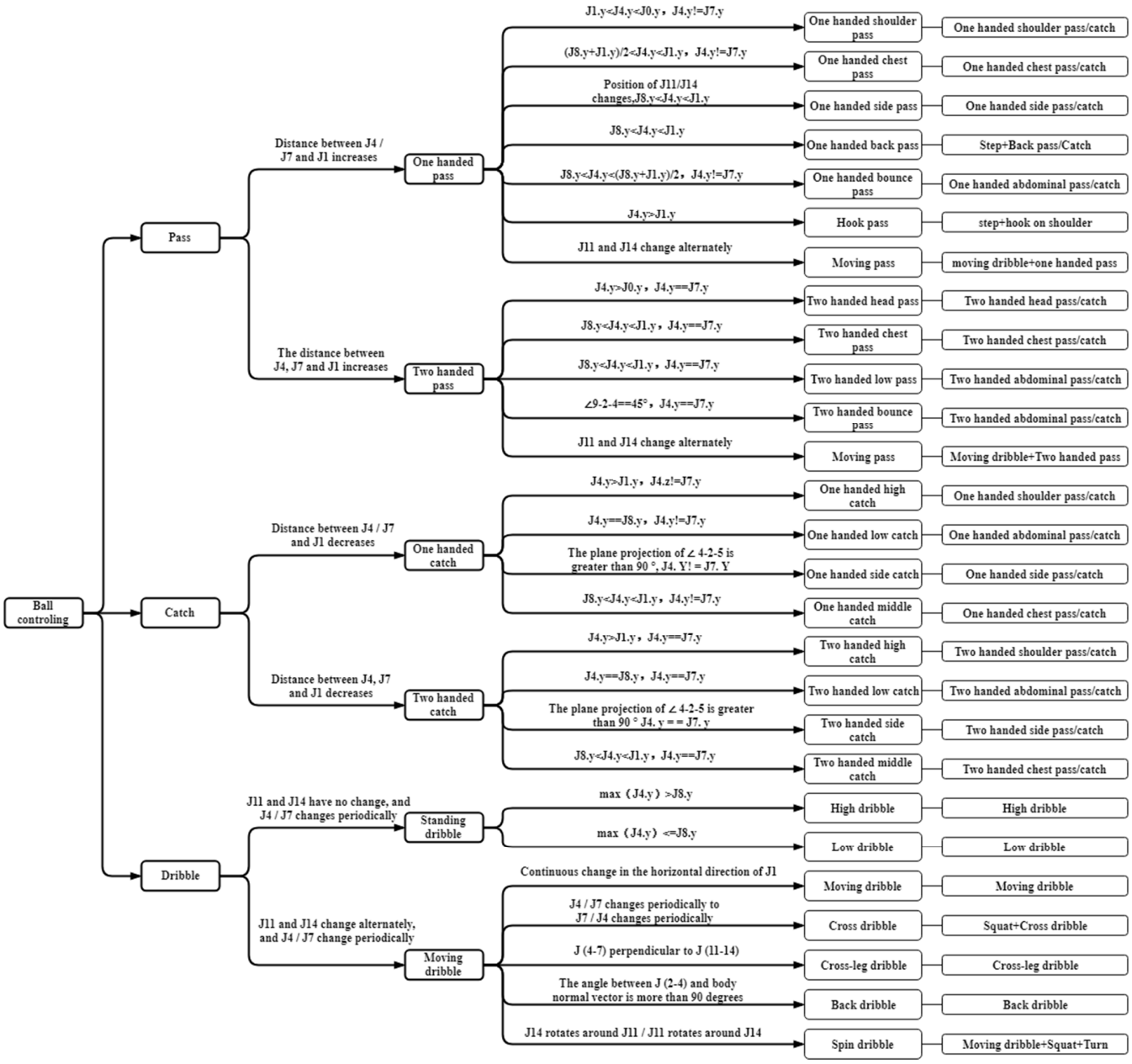
The combination of 32 kinds of atomic actions can cover most of the complex basketball actions. As an attempt, we selected 12 kinds of basic atomic actions that are the most representative to collect and make the dataset. On this basis, we carried out experiments to set the benchmark of the dataset and test the effectiveness and robustness of the basketball action recognition algorithm. The basketball players' basic atomic actions cover three major aspects, including dribbling (standing dribble, moving dribble, etc.), shooting (chest shoot, overhead shoot, etc.), and passing and catching (chest passing and catching, overhead passing and catching). This classification not only reduces the number of categories but also avoids the problem that the same features exist in different actions, which leads to difficulties in distinguishing them. We invited 10 players from the NPU basketball team to complete these 12 types of action shooting tasks so as to ensure that the action data in our dataset conform to the basketball rules. We included the action styles of different players to enhance the representations and richness of the dataset. Detailed action classes can be viewed in Table 2.

The above 12 classes of basketball players' actions cover all possible actions in basketball, and unmentioned actions can find some corresponding category. The layup can be divided into two stages: running with the ball and shooting with one hand, corresponding to action 09 and action 10, respectively. A complete set shot can also be seen as a combination of squatting (07) and overhead shooting (08). Chest shooting, chest passing, and catching are very similar, all of which can be categorized under action 11.



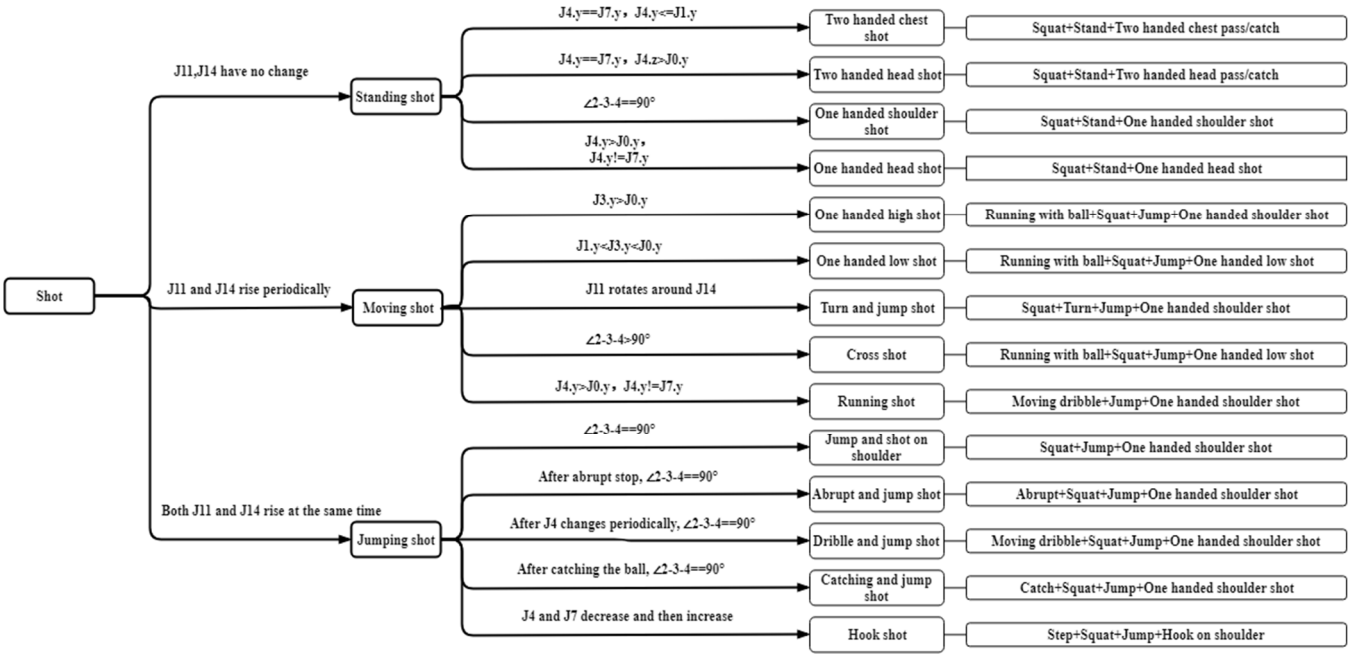
(**a**)

Figure 2. *Cont.*



(**b**)

Figure 2. *Cont.*



(**c**)

Figure 2. The definitions and a decomposition of all complex basketball movements. From top to bottom, it is the detailed action classification of (**a**) movement, (**b**) ball control, and (**c**) shots in basketball.

Table 2. The list of all 12 basketball action categories.

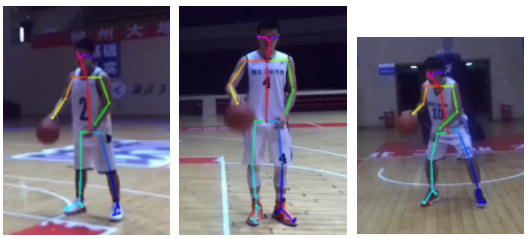
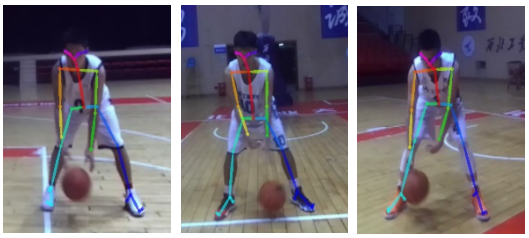
Basketball Player's Action Classes		
Number	**Action**	**Sketch Map**
01	Standing dribble	
02	Moving dribble	
03	Front dribble	

Table 2. *Cont.*

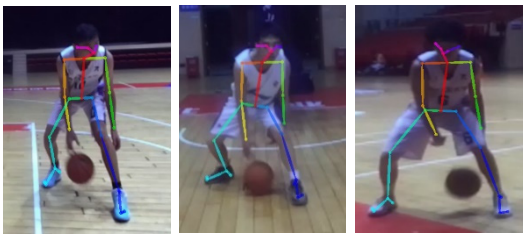
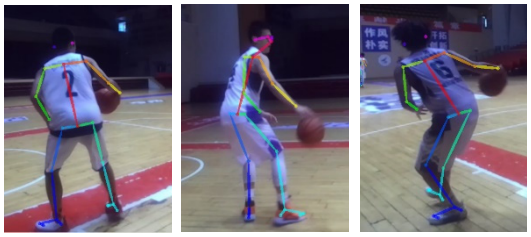
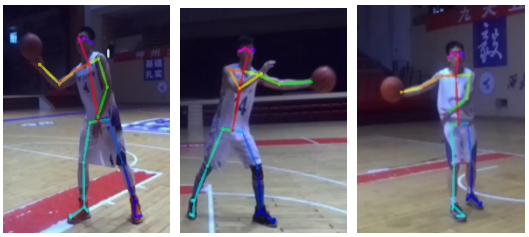
Basketball Player's Action Classes		
Number	**Action**	**Sketch Map**
04	Behind-the-back dribble	
05	Cross-leg dribble	
06	Turning around	
07	Squat	
08	Overhead pass and catch or shooting	
09	Run with ball	

Table 2. *Cont.*

Basketball Player's Action Classes		
Number	**Action**	**Sketch Map**
10	One-hand shoot	
11	Chest pass and catch or shoot	
12	Side throw	

3.2. NPU RGB+D Dataset for Basketball Players

In this section, we introduce the details and the evaluation criteria of NPU RGB+D basketball action recognition dataset (Available at https://github.com/Medjed46/NPU-RGBD-Basketball-Dataset (Accessed on 28th April 2021)). It is composed of 2169 high-quality 3D basketball sports videos labeled by action categories. Unlike with the common RGB sports video dataset collected from YouTube, our dataset was obtained by recording 12 kinds of basketball actions of 10 basketball players from five different visual angles on an indoor basketball court with a ZED camera. At present, there are three kinds of ZED cameras including ZED, ZED 2, and ZED Mini. In this study, we used the ZED camera. It has a size of 175 × 30 × 33 mm, a field of view of 90° (H) × 60° (V) × 100° (D), and a depth perception range of 0.3–25 m. Its camera calibration parameters were the default values. These default parameters can be found on the official ZED website (https://www.stereolabs.com/zed/ (Accessed on 18th April 2021)). The shooting parameters were set to 1280 × 720 resolution and 60 fps frame rate to ensure that the video can be used as good source data without missing key feature information. Our dataset is available for free download by all researchers for scientific research purposes. To protect the privacy of basketball players who took part in the video production of the dataset, we blurred the faces of all participants, which does not affect the use of the dataset.

3.2.1. Subjects

The production of this dataset is attributed to the strong support of the NPU school basketball team and the basketball coach of the NPU Sports Department. Ten NPU basketball team players participated in the video shooting of the dataset in the indoor basketball court of Soaring Stadium. The participants were all well-trained basketball players and varied in age and height. The jersey number of each player is used as a unique serial

number for distinction—2, 4, 6, 7, 10, 13, 14, 16, 18, and 21. Among them, each player completed the required 12 sets of actions in five positions on the court, and each set of actions was completed three to five times.

3.2.2. Court Views

We used the ZED camera support and SDK developed by stereo lab to cut the videos into left and right (L&R) view frame sequences. Owing to the limitation of the number of devices, we can only provide the left and right viewing angles of the ZED camera.

To further increase the camera views, on each setup, we changed positions of the camera on the basketball court and the horizontal views of the subjects. Figure 3 shows the positions of the camera setup on the basketball court. Data of all views corresponding to all position numbers A to E are provided in our NPU RGB+D dataset.

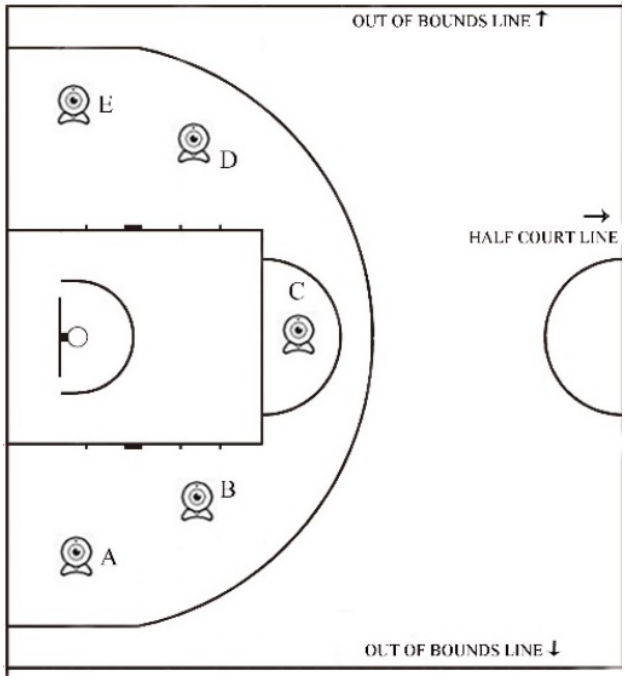


Figure 3. The positions of cameras on the basketball court.

3.2.3. Data Modalities

We obtained RGB+D videos in svo file format from a ZED camera, and then we extracted the other three data modalities: the L&R views frame sequence, depth maps, and 3D skeleton joint point coordinates as shown in Figure 1. The data acquisition process includes the acquisition process of L&R view frame sequences, depth maps, and skeleton joint point coordinates.

A L&R view frame sequence is an RGB image sequence extracted from RGB+D videos by using the export function of ZED SDK, which represents the frames taken by ZED camera from left and right views. The acquisition process of the L&R view frame sequence is as follows.

Firstly, we visited https://download.stereolabs.com/zedsdk (Accessed on 26th April 2021) to get the ZED SDK. This paper used ZED SDK 2.8 of CUDA10.0 in Ubuntu16.

Secondly, we entered the export folder in the svo recording directory and executed ZED_SVO_ Export "path/to/file.svo" "path/to/output/folder/" 2, where "path/to/file.svo" and "path/to/output/folder/" were replaced with the actual location of the svo file and the location of the target folder.

Finally, the left and right frame sequences of svo video were automatically generated for the specified directories.

A depth map is a representation of depth information extracted from a 3D video. The ZED camera captures 3D video with high resolution and frame rate and estimates the pixel depth by comparing the pixel displacement between left and right images to obtain the depth map. The depth map stores the distance value (z) of each pixel (x, y) in the image, representing the distance from the back of the camera's left eye to the object.

The acquisition process of the depth map was as follows.

Firstly, just like getting the L&R view frame sequence, we went to the export directory of ZED SDK.

Secondly, we executed ZED_SVO_ Export "path/to/file.svo" "path/to/output/folder/" 3 command, where "path/to/file.svo" and "path/to/output/folder/" were replaced with the actual location of the svo file and the location of the target folder.

Finally, we got the depth maps in the specified directory.

3D skeleton joint information consists of the 3D locations of 25 major body joints for detected and tracked human bodies in a scene. We extracted the 3D skeleton coordinates based on OpenPose and the ZED camera. OpenPose [36] is an open-source project enabling the real-time tracking of human 2D key points. ZED camera can realize high-precision position tracking and depth perception and provide 3D point cloud information corresponding to 2D points in the picture frame by frame.

The acquisition process of extracting the 3D data using OpenPose from a ZED camera was as follows.

Firstly, we visited https://github.com/stereolabs/zed-openpose (Accessed on 26th April 2021) to obtain the zed-OpenPose provided by stereolabs, the technology development laboratory of ZED camera.

Secondly, ran main.cpp in the src directory to start the ZED camera and capture video frame by frame.

Thirdly, after receiving the RGB frame data, the program automatically woke up the OpenPose thread and obtained the 2D position of 25 human joint points in each frame recognized by OpenPose.

Fourthly, the depth information and confidence of the 2D coordinates of joint points were extracted one by one from the point cloud data automatically generated by ZED camera. If the extraction failed, the depth information of the nearest point within the range of 10 pixels was found as a substitute.

Fifthly, the 2D coordinates and depth information of each joint point were integrated to get 3D coordinates, which were added to the skeleton point coordinate array.

Finally, the skeleton coordinates of each frame of the RGB image were extracted and output to the specified location of the skeleton file.

The depth information was obtained by the built-in 3D reconstruction algorithm of the ZED camera. 3D reconstruction is mainly based on binocular vision technology. Several groups of spatial feature points are imaged in two cameras, and the corresponding coordinates of the points in the two images are obtained. Under the condition of knowing the calibration parameter matrix of two cameras, the world coordinates of the point can be solved by the least square method by establishing four linear equations with the world coordinates of the point as unknowns.

All data modalities provided by NPU RGB+D dataset can be seen in the following example (see Figure 4).

```
keypoint0:(-0.387786,-0.238274,2.41491)
keypoint1:(-0.440411,-0.130271,2.53526)
keypoint2:(-0.615366,-0.12925,2.51539)
keypoint3:(-0.786105,0.165805,2.69283)
keypoint4:(-0.706764,0.417012,2.5838)
keypoint5:(-0.315801,-0.133912,2.60612)
keypoint6:(-0.277307,0.166113,2.69784)
keypoint7:(-0.197129,0.468234,2.76611)
keypoint8:(-0.580109,0.25248,2.79081)
keypoint9:(-0.676236,0.250073,2.7642)
keypoint10:(-0.776886,0.579865,2.61421)
keypoint11:(-0.862256,1.0251,2.77866)
keypoint12:(-0.473661,0.25437,2.8117)
keypoint13:(-0.296528,0.543891,2.74455)
keypoint14:(-0.217131,0.995959,3.04678)
keypoint15:(-0.414578,-0.248014,2.38654)
keypoint16:(-0.329502,-0.250058,2.40621)
keypoint17:(-0.494082,-0.275474,2.40737)
keypoint18:(-0.306474,-0.271014,2.42405)
keypoint19:(-0.0922079,1.09043,2.89394)
keypoint20:(-0.0703188,1.08174,2.93218)
keypoint21:(-0.244237,1.03306,3.08588)
keypoint22:(-0.775914,1.12718,2.65792)
keypoint23:(-0.846308,1.13734,2.68187)
keypoint24:(-0.864569,1.04981,2.78612)
```

Figure 4. An example of NPU RGB+D dataset modalities. Starting from the upper left corner are a left view, right view, depth map, and skeleton joint point coordinate data map.

4. Methods for Basketball Action Recognition

In this section, we first briefly introduce the principles of traditional GCN and LSTM. On this basis, we introduce our basketball action recognition method LSTM-DGCN in detail.

4.1. Traditional GCN and LSTM

GCN was proposed to deal with non-Euclidean structured data. The core idea of graph convolution is to use the edge information to aggregate the node information to generate a new node representation. Graph convolution is done to spread the features of each node and its neighbors to the next layer after the weighted average.

The input of the l-layer convolution neural network is the adjacency matrix A and the characteristic matrix $H^{(l)}$. By making the inner products together with the weight matrix W, a simple neural network layer is formed by an activation function. Finally, the output characteristic $H^{(l+1)}$ is obtained. Mathematically, it is expressed as follows:

$$H^{(l+1)} = f\left(H^{(l)}, A\right) = \sigma\left(AH^{(l)}W^{(l)}\right) \tag{1}$$

where the adjacency matrix A represents the relationship between the nodes of the input graph, and the feature matrix H and weight matrix w represent the features and weights of the nodes updated by the previous learning.

On this basis, we add a self-ring to adjacency matrix A and make symmetry standardization to prevent the features of nodes from being ignored and the original feature distribution from changing. $\widetilde{A}$ and $\widetilde{D}$ represent the adjacency matrix with self ring and its degree matrix respectively. Finally, we obtain the propagation formula of the GCN layer as follows:

$$f\left(H^{(l)}, A\right) = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{2}$$

GCN has excellent performance. It can achieve good results even if it only relies on random initialization parameters and a small number of network layers. Suppose we construct a two-layer GCN, and the activation functions are relu and softmax. The overall forward propagation formula is then

$$Z = f(X, A) = \text{softmax}\left(\hat{A}\text{ReLU}\left(\hat{A}XW^{(0)}\right)W^{(1)}\right) \tag{3}$$

$$\hat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}} \tag{4}$$

where X represents the input characteristic matrix and Z represents the output characteristic matrix. As a common activation function of a neural network, relu function can perform a nonlinear operation on the output of the first layer of the GCN and avoid gradient explosion. After the operation of the second layer, the GCN is finished. The softmax function is used to map the real numbers between 0 and 1 as output.

By inputting a graph structure to the GCN model, through several layers of GCN neural network, the characteristics of each node change from X to Z. However, no matter how many hidden layers there are in the middle, the connection relationship A among nodes is shared. Every node in the graph is constantly changing its state until the final balance owing to the influence of neighbors and further points.

A recurrent neural network (RNN) is a kind of neural network used to process sequence data. Long short-term memory (LSTM) is a special RNN, which is mainly used to solve the problems of gradient disappearance and gradient explosion in the training of a long sequence. The core of LSTM is the cell state c_t, which runs through the whole cell like a conveyor belt. However, there are only a few branches, which can ensure that the information flows through the whole neuron unchanged. LSTM mainly discards or updates cell state information by forgetting gate f_t, input gate i_t and output gate o_t. Figure 5 shows an LSTM neuron, where output state h_t is determined by the input x_t and the hidden state h_{t-1} of the previous time step.

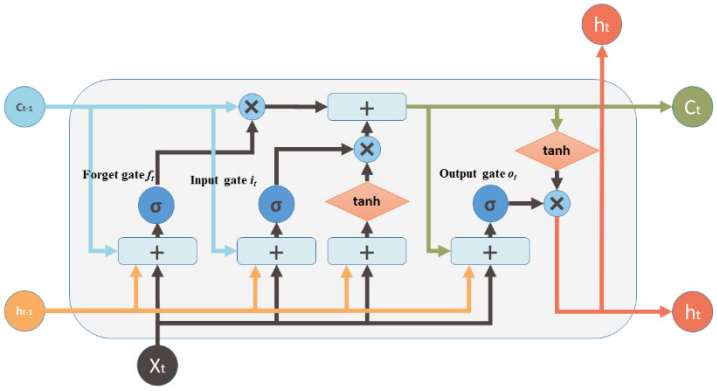


Figure 5. Long short-term memory (LSTM) structure diagram.

The core formula cluster of LSTM is shown in (5), in which input, cell output, and hidden state are represented by x_t, c_t, and h_t, respectively, and three logic gates are represented by i_t, f_t, and o_t.

$$\begin{aligned} i_t &= \sigma(W_{xi} \times X_t + W_{hi} \times H_{t-1} + b_i) \\ f_t &= \sigma\left(W_{xf} \times X_t + W_{hf} \times H_{t-1} + b_f\right) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot tanh(W_{xc} \times X_t + W_{hc} \times H_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} \times X_t + W_{ho} \times H_{t-1} + b_o) \\ H_t &= o_t \cdot tanh(c_t) \end{aligned} \tag{5}$$

4.2. Feature-Enhanced LSTM-DGCN

In this section, we introduce the model architecture of our feature enhanced LSTM-DGCN in detail and how it models the players' skeleton sequences in our dataset to achieve accurate action classification. The human skeleton can be represented by a graph structure naturally; a graph convolution neural network can be used to extract skeleton features to realize action classification. However, as mentioned in Section 2, directly using the traditional GCN may not be the best choice for processing the skeleton data of basketball players. The topological structure of the skeleton graph cannot express the

basketball action features very well. In addition, completely relying on GCN to learn the action features may result in paying attention to irrelevant information and ignoring the important feature descriptions. Therefore, we propose a new feature-enhanced model of structure representation based on a skeleton graph, which selects part of the joint angles and joint distances to construct a feature graph for basketball. It is more concise and targeted than a skeleton graph.

Complete basketball actions are often composed of series of coherent body actions. The relative positions of the limbs change greatly at different time points. To explore the characteristics of basketball players' actions, all the key points of the human skeleton must be extracted, further analyzed, and dealt with while considering the spatial relationship of body structure and the temporal relationship of action sequence so as to realize the accurate identification of basketball players' actions.

The human skeleton can be represented by graph structure naturally; a graph convolution neural network is popularly used to extract human skeleton information features directly to realize action classification. However, we think that learning action features completely via neural network may waste computing resources and process irrelevant information, while ignoring important feature descriptions. For example, when distinguishing between front dribbling and standing dribbling, according to human common sense, it can be easily completed by comparing the knee joint angle. A neural network may be persistent in analyzing the positional relationship between left and right hands though. Therefore, we need a feature pre-extraction module to efficiently extract the key spatial features. It can improve the skeleton's action expression, enhance the role of features in action description, and help effectively explore the co-occurrence relationship between spatiotemporal domains from skeleton sequences. Rather than the confusing 3D position of the joint points, we input to the graph convolution network the distance between the joints and the joint angle that is calculated by feature extraction module, which are feature descriptors with prior knowledge calculated from the data.

Moreover, GCN can only output the classification results of a single frame, so the mainstream time series processing methods are often based on LSTM. We think of embedding the LSTM unit in the GCN model, that is, to integrate the single frame recognition results and make full use of the characteristics of action sequences. Finally, we can get the recognition results of each complete action interval.

Given the above intuition, we propose a feature-enhanced basketball action recognition model based on GCN and LSTM called LSTM-DGCN. This model combines artificial features and is data driven.

First, we get the players' skeleton data from the NPU RGB+D dataset in the unit of action sequence. When the skeleton data in different coordinate systems are mixed together, the model may learn wrong information. Therefore, we do not directly use the skeleton sequence data as the input. Rather, we use the joint distance calculation method provided in the built-in feature extraction module to calculate the distance from other joint points to the reference point as a substitute so as to unify the skeleton data in different camera coordinate systems. To avoid uneven joint distance caused by the height difference among different basketball players, we further scale the distance between all 3D joint points according to the spine distance between neck and mid-hip. To avoid the lack of direction information and the difficulty of action recognition caused by simply relying on the distance between joints, we also screen out the important joint angles as important features of basketball players' action judgment, such as shoulder joint, elbow joint, and knee joint. Finally, the 0-means method is used to normalize the joint angle and distance. The skeleton data after feature extraction are input into GCN.

To further extract the deep action features of basketball players, we need the superposition of GCN layers. For the traditional neural network, the deeper the number of layers, the higher the general accuracy. However, with the deepening of the network, down sampling can lead to the loss of more detailed information, resulting in similar results between the deep network and shallow network. To avoid the gradient disappearing, we introduce

the design of residual block with reference to the deep residual network proposed by Li et al. in [37]. This neural network structure overcomes two problems. That is, the learning rate is low, and the accuracy cannot be effectively improved owing to the deepening of the network depth. The principle of the residual block is shown in Figure 6. The original data output of the previous layers directly skips the layers and is introduced into the input part of the later data layer, which is used as the input of the later network data together with the output of the previous layer after the network processing.

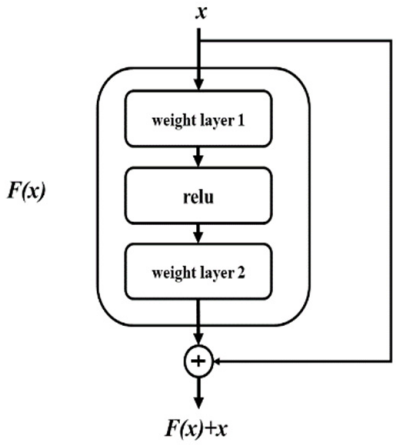


Figure 6. A schematic diagram of the residual principle.

F(x) is a two-layer network without jump connection. The residual block can then be expressed as $H(x) = F(x) + x$. The residual network is helpful to solve the problem of gradient disappearing and gradient exploding so that we can train a deeper network and guarantee good performance at the same time. The concept of residual is also introduced into the model.

If we use G to represent the set of nodes and edges, then the convolution operation of the first graph in the first layer can be expressed as follows:

$$G_{l+1} = F(G_l, W_l) \tag{6}$$

When we consider adding residual structure to the GCN model with a depth of more than three layers, we propose a graph residual learning framework, which can learn the underlying mapping H by fitting another mapping F, and then add the vertices after the transformation to get G_{l+1}. Residual mapping F learning takes a graph as input and outputs a residual graph for the next level.

$$G_{l+1} = H(G_l, W_l) = F(G_l, W_l) + G_l \tag{7}$$

Over-fitting and over-smoothing are two common problems in deep graph convolution neural networks. Over-fitting weakens the generalization ability of the model to other datasets. Over-smoothing weakens the correlation between output representation and input features with the increase of network depth, which hinders the model from upgrading through training. Therefore, we exert much effort. At the same time, we perform dropout and dropedge as proposed by Rong et al. in [38] for processing on the model. We also discard neural units and association relations randomly during training so as to achieve the effect of data enhancement to offset the influence of over-fitting and over-smoothing.

Although our deep graph convolution neural network model can obtain the single-frame action recognition results according to the input skeleton frame sequence, some complex basketball players' actions are difficult to accurately judge by a single frame. It needs to be combined with the before and after time sequence relationship. To accurately convert it into the action recognition result of the whole video interval, we choose to combine LSTM and GCN. First, GCN is used to extract the features of each frame's bone

information, and then LSTM is used to describe its temporal relationship. Finally, a judgment result of the action sequence is obtained to realize end-to-end training.

Our proposed LSTM-DGCN consists of deep GCN and LSTM network, as shown in Figure 7. The input of the model is an action sequence with equal time step, in which the joint distance and joint angle can be calculated according to the 3D coordinates of human key points in each frame. The label vector is composed of the category information of each action interval. The adjacency matrix is composed of the relationship between adjacent action frames in the action interval. After dropout and dropedge, GCN is used to do a graph convolution operation to extract the spatial features of skeleton map. LSTM is then used to learn the temporal features between action frame sequences under a certain time span. In this way, GCN and LSTM alternately process the skeleton sequences. The final output is the predicted results of the action classification labels of the action frame sequences.

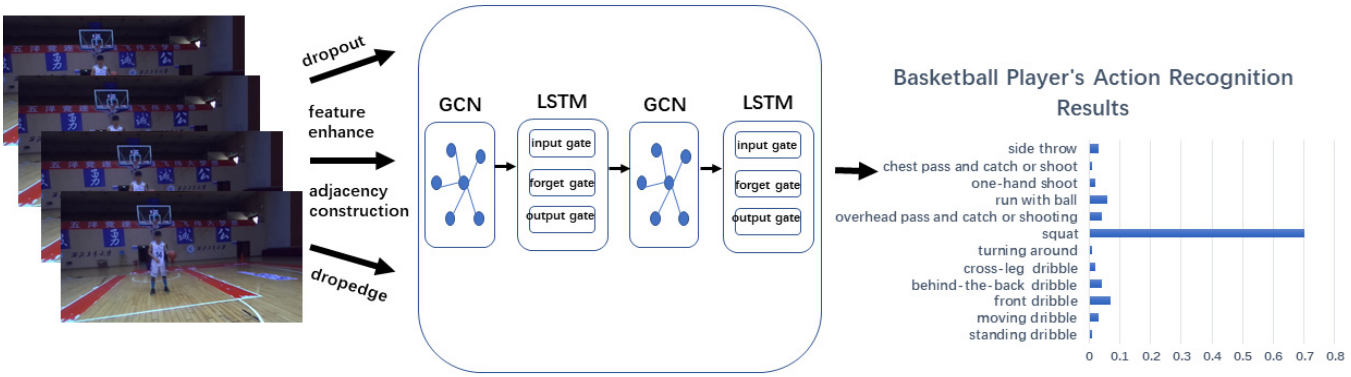


Figure 7. The pipeline of the proposed LSTM-DGCN.

5. Experiment and Evaluation

To setup a benchmark for our dataset, we implemented three of the most advanced action recognition models based on different data modalities: (1) RGB video-based methods, (2) depth map-based methods, and (3) skeleton-based methods. In this section, we describe the preprocessing method for the dataset, the construction of the experimental model architecture, and the three benchmark evaluation criteria of our dataset. Finally, we show the comparison of our proposed LSTM-DGCN and other action recognition methods on our NPU RGB+D dataset.

5.1. Experimental Setup

We used the open implementations of five different action recognition methods and applied them to our new dataset benchmark. Among them, Inflated 3D ConvNets (I3D) [39] and Temporal Segment Networks (TSN) [40] use RGB video for action recognition. HON4D [41] extracts features directly from depth maps without using the skeletal information, whereas 2s-AGCN, ST-GCN, and LSTM-DGCN are skeleton-based methods.

5.1.1. RGB Video-Based Methods

A mainstream idea of action recognition based on RGB video is the two-stream network. Two different CNN streams are used to run on RGB and optical stream respectively, and the results are combined in the end. In the NPU RGB+D dataset, the L&R view RGB video frame sequence is provided. Random video frames can be used as input of RGB stream, and inter-frame action features can be used as input of optical flow. Both the implementations of I3D and TSN are open source, which can be easily deployed according to the requirements of documents.

5.1.2. Depth Map-Based Methods

A depth map-based method is generally based on the normal vector in the depth image rather than the gradient in the color image to better describe the geometric features of the research object. In this study, we selected a representative method, Histogram of Oriented 4D Normals (HON4D), by computing the histogram of oriented 4D normals for activity recognition from depth sequences. The open-source code is the MATLAB version of the code, with the depth map of the NPU RGB+D dataset as the input. The HON4D descriptor is calculated as the feature, and SVM is used to classify the actions.

5.1.3. Skeleton-Based Methods

Skeleton-based action recognition methods focus on the action characteristics of the key points of the human skeleton. They have high efficiency and accurate recognition results.

The skeleton data provided by the NPU RGB+D dataset contain the original 3D positions of the basketball player's body key points in the corresponding camera coordinate system.

At present, the most popular skeleton-based action recognition methods are based on GCN. We chose the best baseline methods ST-GCN and 2s-AGCN as the experimental methods and built the model framework on the basis of open-source code.

To set up a unified evaluation benchmark, we uniformly reserved approximately 5% of the training data as the verification set. Considerable iterative training was carried out on the network. The network with the smallest verification error was selected from all iterations, and its performance is reported in the test data. All experiments were run on NVIDIA 2080ti GPU.

5.2. Benchmark Evaluations

To have standard evaluations for all the reported results on this benchmark, we defined precise criteria for three types of action classification evaluation, namely, cross-subject, cross-view, and cross validation, as described in this section. For each of these three, we report the classification accuracy in percentage. In the subsequent experimental analysis and results in comparison, we use these three evaluation criteria to evaluate the performances of mainstream action recognition methods on our NPU RGB+D dataset.

5.2.1. Cross-Subject Evaluation

We split the 10 subjects into training and testing groups. The training group consisted of seven randomly selected subjects; the remaining subjects were reserved for testing. For this evaluation, the training and testing sets had 1585 and 584 samples, respectively. The IDs of training subjects in this evaluation were 2, 4, 6, 10, 13, 16, and 18. The remaining subjects were reserved for testing.

5.2.2. Cross-View Evaluation

We chose one of the five perspectives as the testing set, and the other four as the training set. The effectiveness of the cross-view evaluation results is based on the premise that no significant differences are noted in the actions of the observation subjects from different perspectives.

5.2.3. Cross Validation

We aggregated the data of each type of action and then randomly extracted 70% of the data as the training set and 30% of the data as the test set. Cross validation is equivalent to training and testing the dataset randomly across perspectives and objects at the same time so that the results can better prove the effectiveness and robustness of the algorithm.

The total dataset was all divided into the training set and testing set for each evaluation standard. The data division under each evaluation standard is listed in Table 3.

Table 3. Data division under three evaluation standards.

Evaluation Method	Train	Test
Cross-Subject Evaluation	1585 (73%)	584 (27%)
Cross-View Evaluation	1767 (81%)	402 (19%)
Cross Validation	1517 (70%)	651 (30%)

5.3. Experimental Results

Our evaluation results of the above methods are shown in Table 4. The first two rows are RGB video-based baseline methods I3D and TSN. Row 3 is depth maps-based baseline method HON4D. The following rows report the performances of ST-GCN, 2s-AGCN, and the proposed feature-enhanced LSTM-DGCN model. Our model learning model outperformed other methods in all the evaluation settings.

Table 4. The results of the three evaluation settings of our benchmark using different methods.

Method	Cross Validation Accuracy (%)	Cross-Subject Evaluation Accuracy (%)	Cross-View Evaluation Accuracy (%)
I3D [27]	12.50	16.67	25.00
TSN [28]	14.58	20.83	35.41
HON4D [29]	28.71	30.35	41.02
ST-GCN [6]	70.40	72.40	72.80
2s-AGCN [20]	71.80	74.80	78.16
Feature-enhanced LSTM-DGCN	80.90	81.60	83.70

The first two lines show the evaluated I3D and TSN, which are action recognition methods based on RGB video. Compared with other methods, the accuracies of these methods were slightly lower because the characteristics of a basketball players' actions cannot be well represented in a 2D image.

The evaluation result of HON4D for the cross-view is better than for the cross-subject view. The reason for this difference is that in the cross-view scene, the habits of the same action are almost the same, and methods based on the depth map are easier to use for learning the appearance- or subject-related action mode.

We can see that the results of our method are much better than those of other classical action recognition methods on our dataset. On the one hand, the original experimental datasets for other methods are relatively simple, not having the complex environmental conditions and action set such as our NPU basketball players' action dataset. On the other hand, the method for a video frame and depth map with serious background noise interference is difficult to use for guaranteeing robustness. In addition, compared with the traditional skeleton-based motion recognition methods, our method pays more attention to the features of critical joint angles and joint distances and takes the human skeleton structure and action timing relationship into account. It is not sensitive to the discontinuity caused by the missing frame data as well, which reduces the data requirement.

We compared the two methods of human skeleton data processing using GCN: ST-GCN and 2s-AGCN. The former is a classic baseline of such methods, which is often used for comparison, whereas the latter is the most advanced model of such methods so far and has achieved the best results. We chose to compare these two methods to prove the effectiveness and excellence of our method. To set up the running environment of ST-GCN and 2s-AGCN, we first extracted the coordinates of key points from NPU RGB+D bone data to generate a skeleton file by imitating the format of an NTU RGB+D dataset, and then analyzed them to get the files with npy as the suffix of a cross-subject, cross-view, and

cross validation input of the model. Then, we added the parameter configuration file of the NPU RGB+D dataset to set the parameters of the feeder and model. Finally, we ran the models and obtain the evaluation results under different data partitions. As shown in Table 3, the accuracies of ST-GCN were 70.40%, 72.40%, and 72.80% in cross validation, cross-object, and cross-view, respectively, whereas the results of 2s-AGCN are better, with 71.80%, 74.80%, and 78.16% respectively. By contrast, our LSTM-DGCN achieved better results under the same conditions. The accuracies of cross-subject, cross-view, and cross validation were 80.90%, 81.60%, and 83.70%, respectively.

Although our method was originally designed for basketball action recognition, the experimental results in Table 5 show that our method also achieved better results compared with other mainstream human motion recognition algorithms based on the depth data on the NTU RGB+D dataset as the benchmark. Under the evaluation criteria of cross objects and cross perspectives, our method achieved accuracy rates of 87.0% and 94.6% on the NTU RGB+D dataset, which are better than most mainstream action recognition methods and similar to the accuracy rates of 88.5% and 95.1% by the most advanced 2s-AGCN. The NTU RGB+D dataset is a general human action dataset rather than a special basketball action dataset. Therefore, the LSTM-DGCN proposed here actually maps the NPU with 12 types of actions selected from the NTU RGB+D dataset. The results of the 12 kinds of basketball players' actions in NTU RGB+D are much better than those in our dataset, which not only proves the effectiveness of our method but also shows that NTU RGB+D's action categories are easier to distinguish than our basketball players' actions. The results of LSTM-DGCN on the NPU RGB+D dataset are not as good as those on the NTU RGB+D dataset for a few reasons. On the one hand, our dataset is smaller than NTU RGB+D, both in terms of research objects and perspectives and the number of videos. On the other hand, basketball players' actions are far more complex than daily actions, so they are more difficult to classify. Nevertheless, it is enough to prove that our method is also effective in the field of general action recognition.

Table 5. Comparison of action recognition performance on NTU-RGB+D. The classification accuracies on both cross-subject and cross-view benchmarks are presented.

Method	Cross-Subject Accuracy	Cross-View Accuracy
Lie-Group [17]	50.1%	52.8%
H-RNN [42]	59.1%	64.0%
PA-LSTM [9]	62.9%	70.3%
STA-LSTM [34]	73.4%	81.2%
ST-GCN [6]	81.5%	88.3%
AS-GCN [19]	86.8%	94.2%
2s-AGCN [20]	88.5%	95.1%
LSTM-DGCN	87.0%	94.6%

In addition, we provide the recognition accuracy of each type of action in Table 6. Interval accuracy is the accuracy of action sequence recognition based on LSTM-DGCN to distinguish from single-frame accuracy. The table indicates that we achieved excellent action recognition results in almost all basketball action categories.

Table 6. Recognition accuracies of all action categories.

Number	Action	Interval Accuracy (%)
01	Standing dribble	91.9
02	Moving dribble	78.3
03	Front dribble	83.4
04	Behind-the-back dribble	87.9
05	Cross-leg dribble	77.3
06	Turning around	72.1
07	Squat	80.6
08	Overhead pass and catch or shooting	81.9
09	Running with ball	89.3
10	One-hand shot	88.4
11	Chest pass and catch or shooting	72.7
12	Side throwing	81.5

6. Discussion

The NPU RGB+D dataset proposed in this study is a single-player basketball video shot at an indoor basketball court. At present, it includes 12 kinds of basketball player actions, involving dribbling, passing, and catching, and shooting. It contains the basic atomic basketball actions selected based on the FIBA basketball rules and the technical guidance of professional basketball coaches. It has an important foundational significance for the recognition of all complex basketball actions. It not only has important value in the research on basketball action recognition but also has high application value in the intelligent judgment of basketball games and daily training of basketball players. However, the dataset does not involve the actions of multiple basketball players in a basketball game. First, multiple depth cameras are needed to cover the whole field of view of the basketball court. Second, multiple basketball players should be tracked. Third, basketball players must be matched with different camera fields of vision through triangulation positioning to determine their positions on the court and solve the problem of mutual occlusion. The information of each player on each timestamp is separated and written into the dataset as valid data.

When extracting skeleton coordinates from RGB+D video taken by ZED camera, the skeleton data may be incomplete and inaccurate owing to overly long distances and insufficient illumination. Therefore, we should not only improve the accuracy of the 3D human key point coordinate extraction algorithm based on ZED and OpenPose, but also improve the robustness of the LSTM-DGCN algorithm to deal with noise and missing data.

Our NPU RGB+D dataset, as the only existing large RGB+D dataset for basketball action recognition, has rich and diverse action data, which can provide support and a benchmark for the further research on basketball action recognition, especially for the depth-based action recognition method. The experimental results show that large-scale and effective available data are very helpful for the training of a data-driven deep learning model. A part of the shooting data can even be selected to train a scoring system, which can be very useful.

Our method can also be used for game statistics and basketball training. A basketball video is composed of various actions, and the actions are composed of continuous body posture sequences. LSTM-DGCN can judge changes of action according to the changes of the frame classification results in certain intervals, recognize the current action timely and accurately, and divide the video up automatically. Much labor and time can be saved in dividing the basketball video to count the basketball game score and do action analysis. If

the input is an action sequence that is manually divided, then a faster and more accurate result of action classification can be obtained.

The basketball action recognition algorithm LSTM-DGCN proposed here achieved good results on our NPU RGB+D dataset. However, it remains unsatisfactory for distinguishing the similar actions of two-handed chest shooting and two-handed chest passing, as well as two-handed overhead passing and two-handed overhead shooting. More training data and more effective features are needed to further improve the accuracy of recognition between similar actions.

7. Conclusions

In this paper, research on the action recognition of basketball players was presented. We provided an action classification method for basketball players and constructed a rule-based large-scale basketball dataset NPU RGB+D which contains 2169 RGB+D videos and 75,000 frames, including an RGB frame sequence, a depth map, and skeleton coordinates, covering the most basic 12 types of basketball actions. We also proposed a feature-enhanced method LSTM-DGCN of action recognition based on the spatio–temporal relationship of skeleton data. Moreover, we provided three evaluation criteria under different scenarios.

Using our NPU RGB+D dataset, we compared RGB video-based baseline methods I3D and TSN, the depth map-based baseline method HON4D, and skeleton-based methods ST-GCN and 2s-AGCN. With our method, we achieved 80.90%, 81.60%, and 83.70% accuracies in cross validation, cross-subject evaluation, and cross-view evaluation, which are higher values than those achieved by the other human action recognition methods. On the open NTU RGB+D dataset, our LSTM-DGCN method achieved 87.0% and 94.6% accuracies under cross-subject and cross-view modes, respectively—close to the 88.5% and 95.1% accuracy of the state-of-the-art human action recognition method 2s-AGCN. The experimental results show that the proposed basketball player action dataset contains appropriate action categories and rich scenes and objects, which is challenging for some classic action recognition methods. Our dataset can also be used as a benchmark dataset for basketball action recognition. Our proposed action recognition method makes full use of the temporal and spatial characteristics of player skeleton sequence and had good performance under all evaluation criteria of different benchmark datasets. Our method exhibits good robustness and adaptability.

We hope that our proposed basketball action recognition dataset can become a challenging benchmark dataset in the field of action recognition. We also hope that our research on basketball action recognition method can provide inspiration to our peers in the field of action recognition.

At present, the 12 types of actions performed by 10 players with 2169 RGB+D 21videos and 75,000 frames in our open NPU RGB+D dataset are the results of four attempts at shooting. The dataset includes RGB frame sequences, depth maps, and skeleton coordinates. At the same time, we put forward a feature-enhanced LSTM-DGCN method for action recognition of basketball players. Due to the influences of the site, professionals, and environment, we only selected high-quality RGB+D videos from a large number of datasets for preprocessing. What is more, this involved the acquisition process of L&R view frame sequences, depth maps, and skeleton joint point coordinates; video segmentation; interval division; category marking; and coordinate extraction, which took our team 6 months. Hence, there are difficulties in data collection and processing. We chose 12 basic atomic actions for experiments to give a benchmark dataset and verify the effectiveness of our method. Our experimental data collection and processing process is worth learning, and it is sufficient to verify our method. By repeating our current technical path, future work can identify more basketball players' actions (i.e., other kinds of atomic actions in Section 3.1), to apply our method more effectively for game statistics and basketball training in practice. In future work, we will further expand the dataset of basketball player's actions to verify the effectiveness of our method and publish it to the public. Colleagues in related research fields are welcome to supplement our work and improve on it.

In the future, we will conduct real-time tracking and action recognition research on multiple basketball players on the court. Song et al. proposed a unified RGBD tracking evaluation benchmark Princeton tracking benchmark in [43] to quantitatively compare tracking algorithms based on RGB and RGBD. The weighted convolution operators tracker [44] proposed by Liu et al. was evaluated on the benchmark and achieved top performance compared with the existing RGBD tracking algorithm. Inspired by the above work, our future research work will be carried out in the following steps:

Firstly, we will set up multiple zed cameras at different positions by the court to shoot synchronously to obtain datasets of different perspectives. For example, we know that a research team is using eight ZED cameras to shoot basketball games, and got video shooting data from eight perspectives.

Secondly, the weighted convolution operators tracker mentioned in the above paper will be used to track the positions of players captured from each perspective.

Thirdly, we will match the tracking results of multiple cameras and calculate the position of each player in the field coordinate system.

Fourthly, after solving the problem of mutual occlusion between players through an algorithm, we will use ZED and OpenPose to extract the skeleton coordinates of players from each perspective and integrate them.

Finally, for each player's skeleton sequence, LSTM-DGCN is used for action recognition.

In future works, we will also contemplate a distribution of 70% for training, 20% for validation, and 10% for testing and show the results obtained on the data reserved for testing.

Author Contributions: Conceptualization, C.M., J.F. and T.Z.; Data curation, T.Z.; Investigation, C.M. and T.Z.; Methodology, C.M., J.F. and J.Y.; Resources, T.Z.; Software, J.F. and J.Y.; Supervision, C.M.; Validation, J.F., J.Y. and T.Z.; Visualization, J.F. and J.Y.; Writing—original draft, C.M. and J.F.; Writing—review & editing, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: NPU RGB+D basketball players action recognition dataset: https://github.com/Medjed46/NPU-RGBD-Basketball-Dataset (accessed on 28 April 2021).

Acknowledgments: This project was carried out by Chunyan Ma's Laboratory of Software School of Northwestern Polytechnic University. It was guided and supported by the research direction and technical support of Min Xu andYu Peng's team from Sydney University of science and technology. We sincerely thank them for their help in revising the paper.

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

References

1. Aggarwal, J.; Xia, L. Human activity recognition from 3D data: A review. *Pattern Recognit. Lett.* **2014**, *48*, 70–80. [CrossRef]
2. Zhang, J.; Li, W.; Ogunbona, P.O.; Wang, P.; Tang, C. RGB-D-based action recognition datasets: A survey. *Pattern Recognit.* **2016**, *60*, 86–105. [CrossRef]
3. Chaquet, J.M.; Carmona, E.J.; Fernández-Caballero, A. A survey of video datasets for human action and activity recognition. *Comput. Vis. Image Underst.* **2013**, *117*, 633–659. [CrossRef]
4. Wang, H.; Wang, L. Learning content and style: Joint action recognition and person identification from human skeletons. *Pattern Recognit.* **2018**, *81*, 23–35. [CrossRef]
5. Wang, P.; Li, W.; Ogunbona, P.; Wan, J.; Escalera, S. Escalera, RGB-D-based Human Motion Recognition with Deep Learning: A Survey. *IEEE Int. Conf. Comput. Vision* **2017**, *171*, 118–139.
6. Yan, S.; Xiong, Y.; Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
7. Soomro, K.; Zamir, A.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv* **2012**, arXiv:1212.0402.

8. Qiu, Z.; Yao, T.; Ngo, C.-W.; Tian, X.; Mei, T. Learning Spatio-Temporal Representation with Local and Global Diffusion. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 July 2019; pp. 12048–12057.
9. Shu, Y.; Shi, Y.; Wang, Y.; Huang, T.; Tian, Y. P-ODN: Prototype-based Open Deep Network for Open Set Recognition. *Sci. Rep.* **2020**, *10*. [CrossRef] [PubMed]
10. Li, W.; Zhang, Z.; Liu, Z. Action recognition based on a bag of 3D points. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition—Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 9–14.
11. Pham, H.; Khoudour, L.; Crouzil, A.; Zegers, P.; Velastin, S.A. Exploiting deep residual networks for human action recogni-tion from skeletal data. *Comput. Vision Image Underst.* **2018**, *170*, 51–66. [CrossRef]
12. Ashwini, K.; Amutha, R. Compressive sensing based recognition of human upper limb motions with kinect skeletal data. *Multimed. Tools Appl.* **2021**, *80*, 10839–10857. [CrossRef]
13. Shahroudy, A.; Liu, J.; Ng, T.-T.; Wang, G. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1010–1019.
14. Song, S.; Lan, C.; Xing, J.; Zeng, W.; Liu, J. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
15. Liu, M.; Liu, H.; Chen, C. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognit.* **2017**, *68*, 346–362. [CrossRef]
16. Ramanathan, V.; Huang, J.; Abu-El-Haija, S.; Gorban, A.; Murphy, K.; Li, F.-F. Detecting Events and Key Actors in Mul-ti-person Videos. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3043–3053.
17. Acuna, D. Towards real-time detection and tracking of basketball players using deep neural networks. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
18. Li, W.-H.; Hong, F.-T.; Zheng, W.-S. Learning to Learn Relation for Important People Detection in Still Images. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4998–5006.
19. de Vleeschouwer, C.; Chen, F.; Delannay, D.; Parisot, C.; Chaudy, C.; Martrou, E.; Cavallaro, A. Distributed Video Acquisition and Annotation for Sport-Event Summarization, NEM Summit, 8 2008. Available online: https://www.researchgate.net/publication/229017805_Distributed_video_acquisition_and_annotation_for_sport-event_summarization (accessed on 26 April 2021).
20. Liang, Q.; Wu, W.; Yang, Y.; Zhang, R.; Peng, Y.; Xu, M. Multi-Player Tracking for Multi-View Sports Videos with Improved K-Shortest Path Algorithm. *Appl. Sci.* **2020**, *10*, 864. [CrossRef]
21. Thomas, G.; Gade, R.; Moeslund, T.B.; Carr, P.; Hilton, A. Computer vision for sports: Current applications and research topics. *Comput. Vis. Image Underst.* **2017**, *159*, 3–18. [CrossRef]
22. Li, Y.; Xia, R.; Huang, Q.; Xie, W.; Li, X. Survey of Spatio-Temporal Interest Point Detection Algorithms in Video. *IEEE Access* **2017**, *5*, 10323–10331. [CrossRef]
23. Wang, P.; Li, W.; Li, C.; Hou, Y. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowl.-Based Syst.* **2018**, *158*, 43–53. [CrossRef]
24. Dai, W.; Chen, Y.; Huang, C.; Gao, M.-K.; Zhang, X. Two-Stream Convolution Neural Network with Video-stream for Action Recognition. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
25. Chen, L.; Wang, W. Analysis of technical features in basketball video based on deep learning algorithm. *Signal Process. Image Commun.* **2020**, *83*, 115786. [CrossRef]
26. Pan, Z.; Li, C. Robust basketball sports recognition by leveraging motion block estimation. *Signal Process. Image Commun.* **2020**, *83*, 115784. [CrossRef]
27. Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. *IEEE Access* **2018**, *6*, 1155–1166. [CrossRef]
28. Mukherjee, S.; Anvitha, L.; Lahari, T.M. Human activity recognition in RGB-D videos by dynamic images. *Multimed. Tools Appl.* **2020**, *79*, 19787–19801. [CrossRef]
29. Vemulapalli, R.; Arrate, F.; Chellappa, R. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 588–595.
30. Du, Y.; Fu, Y.; Wang, L. Skeleton based action recognition with convolutional neural network. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 579–583.
31. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3590–3598.
32. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.

33. Nguyen, L.N.N.; Martín, D.M.R.; Català, A.; Pérez-López, C.; Samà, A.; Cavallaro, A. Basketball Activity Recognition using Wearable Inertial Measurement Units. In Proceedings of the XVI International Conference on Human Computer Interaction, Vilanova i la Geltru, Spain, 7–9 September 2015; Volume 60, p. 60.
34. Hölzemann, A.; Van Laerhoven, K. Using Wrist-Worn Activity Recognition for Basketball Game Analysis. In Proceedings of the Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction, Berlin, Germany, 20–21 September 2018; pp. 1–6.
35. Shi, W. The influence of the evolution of basketball rules on the development of basketball technique and tactics. *Agro. Food Ind. Hi-Tech.* **2017**, *28*, 556–559.
36. Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei, S.-E.; Sheikh, Y.A. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef]
37. Li, G.; Muller, M.; Thabet, A.; Ghanem, B. DeepGCNs: Can GCNs Go as Deep as CNNs? In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
38. Rong, Y.; Huang, W.; Xu, T.; Huang, J. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. *arXiv* **2018**, arXiv:1907.10903.
39. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4724–4733.
40. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. *Temporal Segment Networks: Towards Good Practices for Deep Action Recognition*; Springer: Cham, Germany, 2016.
41. Oreifej, O.; Liu, Z. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 July 2013; pp. 716–723.
42. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1110–1118.
43. Song, S.; Xiao, J. Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 233–240.
44. Liu, W.; Tang, X.; Zhang, C. Robust RGBD Tracking via Weighted Convolution Operators. *IEEE Sens. J.* **2020**, *20*, 4496–4503. [CrossRef]