Two-stream Graph Convolutional Networks for 2D Skeleton-based Fall Detection

Yan Liu[*1], Yuelin Deng[*2], Chen Jia[*3], Yanru Yang[*4], Ruonan Wang[*5]and Chi Li[*6]

[*1]Taikang Insurance Group Co., Ltd., Taikang Innovation Center, No.21-1 Science Park Road, Changping District, Beijing, P.R.China
E-mail: lyan2006@gmail.com
[*2]Taikang Insurance Group Co., Ltd., Taikang Innovation Center, No.21-1 Science Park Road, Changping District, Beijing, P.R.China
E-mail: dengyl29@taikanglife.com
[*3]Taikang Insurance Group Co., Ltd., Taikang Innovation Center, No.21-1 Science Park Road, Changping District, Beijing, P.R.China
E-mail: jiachen05@taikanglife.com
[*4]Taikang Insurance Group Co., Ltd., Taikang Innovation Center, No.21-1 Science Park Road, Changping District, Beijing, P.R.China
E-mail: yangyr25@taikanglife.com
[*5]Taikang Insurance Group Co., Ltd., Taikang Innovation Center, No.21-1 Science Park Road, Changping District, Beijing, P.R.China
E-mail: wangrn07@taikanglife.com
[*6]Taikang Insurance Group Co., Ltd., Taikang Innovation Center, No.21-1 Science Park Road, Changping District, Beijing, P.R.China
E-mail: lichi01@taikanglife.com

Abstract. The vision-based fall detection solutions play more and more significant role in the field of elder care. By reducing waiting time for rescue, life is saved. To improve the performance of fall detection, we propose a 2D skeleton-based fall detection method relying on the graph convolutional networks in this paper. The method is designed to a two-steam structure. Both the Cartesian coordinate and the polar coordinate are used to represent the skeleton feature of human body. The detection process to action sequence is accomplished by the fusion of two-stream of spatial temporal graph convolutional networks. To enhance the detection effect, we extend the scale of training dataset by converting the public 3D skeleton to 2D skeleton. The experimental results demonstrate that the performance of our method exceeds baseline method on both the benchmark NTU-RGB dataset and the proposed dataset.

Keywords: Fall Detection, 2D Skeleton-based, Graph Convolutional Networks, Two-stream, Polar Coordinate, Action Recognition

1. INTRODUCTION

With the rapid growth of the elderly population, there is a growing concern on elderly safety care. One major risk of elderly people is the fall accident. Every year, an estimated 30-40% of patients over the age of 65 fall at least once[1]. Most of the time, the consequences of fall in the elder people without timely medical assistance are unimaginable[2]. Automatic fall detection solution can detect the fall accident and generate alarm quickly in case of danger, which lets the fallen person get medical attention in a timely manner.

The ways of fall detection are divided into two categories by sensors, which are wearable sensors and environmental sensors. Wearable sensors such as switches, accelerometers, and gyroscopes[3], are inconvenient to wear and need to be recharged frequently. Compared with the wearable devices, the environmental sensors are free from the inconvenience of wearing the device.

Among a variety of environmental sensor-based approaches, the vision-based approach is the most common one. These optical camera devices do not cause sensory side effects on human health, and do not affect people's normal life[4]. Meanwhile, the development of intelligent monitoring and the internet of things paradigms forms an optimal context for vision-based solutions. Therefore, the reliable vision-based fall detection solutions play more and more important role in future elder care systems[4].

As a kind of special human action, falls can be recognized from multiple vision-based modalities, such as appearance, depth, optical flows, and body skeletons. Among these modalities, dynamic human skeletons usually convey significant information of human action especially in the task of fall detection. Recently, graph convolutional networks (GCNs), achieve remarkable performance for skeleton-based action recognition[5]. Thus, it is our research motivation to apply and improve GCNs in the fall detection task.

It must be mentioned that we take fall detection as a special kind of action recognition in our work, and also use the same evaluation metrics as action recognition. Here, we focus on the 2D skeleton-based approach because most cameras in the real environment can't capture the depth information. In this paper, we propose a 2D Skeleton-based fall detection method with two-stream GCNs on the basis of Spatial Temporal Graph Convolutional Networks (ST-GCN)[5], the main contributions are as follows:

(1) We propose a novel two-stream architecture, called Cartesian-polar Stream Graph Convolutional Networks (CPS-GCN). The innovative point is that it contains both Cartesian and polar representations of the 2D skeleton features. And it is proved that the two-stream architecture performs better than original one-stream architecture.

(2) We present an approach to obtain the 2D skeleton data from the NTU-RGB+D[6] which is a large 3D skeleton dataset containing fall category. Thus we can align the definitions of skeleton data and overcome the difficulty of training data shortage.

(3) We propose an Indoor Specific Action (ISA) dataset for fall detection in the daily vision-based monitoring environment, by using OpenPose[7] to extract the 2D skeleton. Meanwhile, we set up the complete pipeline for fall detection in the real environment.

The experimental results show that our method achieves higher accuracy compared with baseline method ST-GCN, on both the benchmark NTU-RGB dataset and the ISA dataset.

2. RELATED WORK

2.1. Overview of Existing Vision-based Methods

Existing vision-based fall detection methods are divided into traditional algorithms and neural networks algorithms.

In traditional algorithms, manually crafted features like silhouettes or bounding boxes are extracted from the frames in order to facilitate fall detection[4]. Some approaches use those features as input for a classifier to automatically fall detection, such as k-Nearest Neighbor (KNN) Classifier[8], Support Vector Machine (SVM)[9][10], Gaussian Mixture Model (GMM)[11], Hidden Markov Model (HMM)[12] and so on. However, these traditional methods can not achieve high accuracy due to the limitation of algorithms.

The neural networks, especially Convolutional Neural Networks (CNNs), perform better compared with traditional algorithms[4][13][14], because the neural networks have strong ability in feature learning. CNNs are also the basis of mainstream image identification models, which are usually made up of the convolutional layer, pooling layer and so on[15]. In recent years, more and more action recognition methods based on neural networks and deep learning achieve significant performance improvement, which are also applied in fall detection task.

2.2. Skeleton-based Methods

Among these methods based on neural networks, skeleton-based methods attract much attention due to their robustness against the dynamic circumstance and complicated background[5, 16-28]. With the development of deep learning, researchers structure the skeleton data as a pseudo-image[22, 29-31] or a sequence of coordinate vectors[3, 6, 18, 32-33], which is fed into CNNs or Recurrent Neural Networks (RNNs) to predict.

Recently, ST-GCN is proposed to model the skeleton data with GCNs, which extends graph neural networks to a spatial-temporal graph and achieves substantial improvements over mainstream methods. Based on the concept of ST-GCN, different improved methods are put forward, such as Actional-Structural Graph Convolutional Networks (AS-GCN)[34], Spatial-Temporal Graph Routing (STGR) networks[35], Two-Stream Adaptive Graph Convolutional Networks (2s-AGCN)[21] and so on. To these methods, OpenPose[7][36] and ST-GCN have great significances. OpenPose is essential in the step of skeleton feature extraction, and ST-GCN is the foundation work of this kind of methods.

2.2.1. OpenPose

OpenPose, built by Perceptual Computing Lab at CMU, detects human keypoints in 2D images for multi-person. The approach uses a non-parametric representation, which is referred to as Part Affinity Fields (PAFs), to learn to associate body parts with individuals in the image. As can be seen from **Fig. 1-(B)**, given the feature maps generated by a convolutional network (VGG), the architecture is designed to jointly learn part locations and their association via two branches with multi-stage of the same sequential prediction process. Each stage in the first branch predicts confidence maps S^t, and each stage in the second branch predicts PAFs L^t. After each stage, the predictions from the two branches, along with the image features, are concatenated for next stage. The architecture encodes global context, allowing a greedy bottom-up parsing step that maintains high accuracy while achieving real-time performance.

2.2.2. ST-GCN

As the first work applying GCNs in action recognition, ST-GCN proposes a generic graph-based formulation for modeling dynamic skeletons and several principles in designing convolution kernels to meet the specific demands in skeleton modeling[5].

Given the sequences of body joints in the form of 2D or 3D coordinates, a spatial-temporal graph is set up. There are two types of edges, namely the spatial edges that conform to the natural connectivity of joints and the temporal edges that connect the same joints across consecutive time steps. Multiple layers of spatial-temporal graph convolution operations are applied on the graph to extract the high-level features.

In the spatial dimension, the graph convolution operation on vertex v_{ti} is formulated as:

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(\mathbf{p}(v_{ti}, v_{tj})) \cdot \mathbf{w}(v_{ti}, v_{tj}) \quad (1)$$

Where $\mathbf{p}(\cdot)$, $\mathbf{w}(\cdot)$ denote sampling function and weighting function, respectively. $f_{in}(\mathbf{p}(v_{ti}, v_{tj}))$ is the feature map and $Z_{ti}(v_{tj})$ is the normalizing term. $B(v_{ti}) = \{v_{tj} | d(v_{ti}, v_{tj}) \le D\}$ denotes the sampling area of the convolution for v_{ti} and $d(v_{ti}, v_{tj})$ denotes the minimum length of any path v_{ti} from v_{tj}. In the work of ST-GCN, D = 1 is used for all cases.

Similarly, for modeling the spatial temporal dynamics, the concept of neighborhood is extended to also include temporally connected joints as:

$$B(v_{ti}) = \{v_{qj} \mid d(v_{tj}, v_{ti}) \le K, |q - t| < \lfloor \Gamma / 2 \rfloor\} \quad (2)$$

The parameter Γ controls the temporal range to be included in the neighbor graph and can thus be called the temporal kernel size. It is straightforward to perform the graph convolution similar to the classical convolution operation. In this way, the convolution operation on the constructed spatial temporal graphs is well defined.

To implement the graph-based convolution, the intra-body connections of joints within a single frame are represented by an adjacency matrix **A** and an identity matrix **I** representing self-connections. To improve the recognition performance, a learnable mask **M** is added on each layer of spatial temporal graph convolution. The whole model is trained in an end-to-end manner with back-propagation.

In the following sections, all of our work is based on the application and improvement of ST-GCN in the fall detection task.

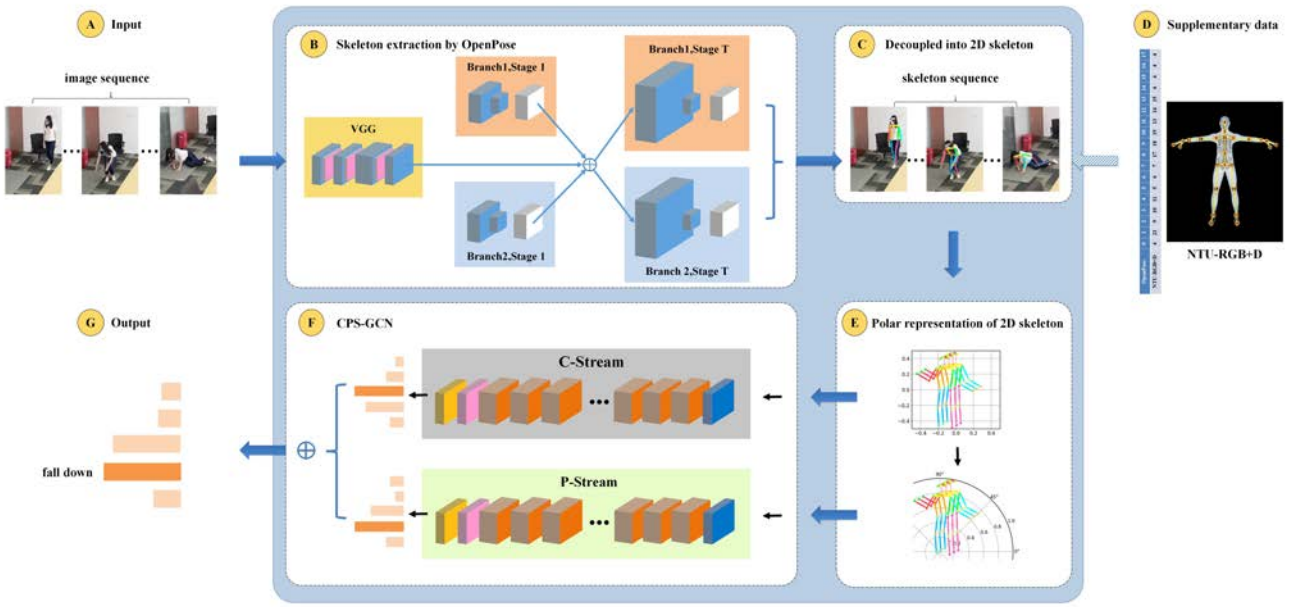


Fig. 1 The overview of our framework. (A) Input image sequence. (B) Skeleton extraction by OpenPose. (C) 2D skeleton sequence. (D) 3D skeleton in NTU-RGB+D. (E) Polar representation of 2D skeleton. (F) Architecture of CPS-GCN, containing C-stream and P-stream. (G) Result of fall detection.

3. Our Method

3.1. Framework Overview

Fig. 1 shows the overview of our framework. First of all, we extract the 2D skeleton sequence from input image sequence by OpenPose as introduced in Sec.3.2.2. Simultaneously, to expand training data, we transform the NTU-RGB+D dataset from 3D skeleton to 2D skeleton as described in Sec.3.2.1. Secondly, we calculate the polar coordinates of 2D skeleton from the Cartesian coordinates as presented in Sec.3.3. And then, we put these Cartesian coordinates and polar coordinates into the C-Stream and P-Stream graph convolutional networks in parallel, as described in Sec.3.4. Finally we get the late fusion softmax score generated by CPS-GCN to predict the action label.

3.2. 2D Skeleton Acquisition

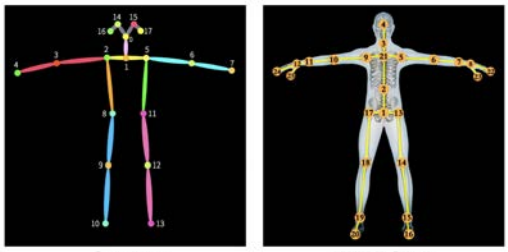
Skeleton features reveal more robust and accurate information about human actions than other features. The reason why we choose 2D but not 3D skeleton is that most vision-based systems only employ the RGB images captured by ordinary optical cameras. Therefore, in Sec.3.2, the methods for 2D skeleton acquisition are discussed.

3.2.1. NTU-RGB: 3D skeleton to 2D skeleton

To expand the training dataset in the case of 2D fall data scarcity, the NTU-RGB+D[6] dataset is chosen. Considering that the NTU-RGB+D dataset only contains 3D skeleton, and the definition of skeleton is diffrerent from OpenPose which is the toolbox to extract 2D skeleton, a 3D NTU-RGB+D to 2D OpenPose skeleton tranformation is necessary to align the definition of skeleton data.

The configuration of body joints is defined in original NTU-RGB+D dataset, which includes the base of the spine, middle of the spine, neck, head , left shoulder, 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, spine, tip of the left hand, left thumb, tip of the right hand and right thumb, respectively. These 25 features correspond to the labels ranging from 1 to 25. Specifically, there are 18 points described in OpenPose[7][36], they are nose, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, left ankle, right eye, left eye, right ear and left ear, respectively. The points correspond to the labels ranging from 0 to 17, which are similar to the format[5] of Kinetics[37] and COCO dataset[38]. With these preparation, a conversion operation from 3D joints to 2D skeletons is performed by projecting the 25 keypoints of NTU-RGB+D onto 18 locations of OpenPose. As a result , the 2D skeleton sequences of each subject in video clips are acquired naturally for a supplement of the finite training data. The dataset made up of these 2D skeleton sequences is represented as NTU-RGB to distinguish from NTU-RGB+D in this paper.

Fig. 2 manifests the corresponding relation between 3D features and 2D skeletons, which is self-defined because the differences in skeletons make the alignment pretty formidable. As is shown in **Fig. 2**, the top left corner represents the skeletons with 18 keypoints detected by OpenPose mentioned in Sec.2.2.1, while the top right describes the 25 joints of the NTU-RGB+D dataset. Furthermore, the proposed corresponding relation of the two skeleton structures is shown in the bottom table. It is easy to find out that we extend the keypoints of the head while abandoning the joints of the hands and feet. Experimental results in Sec.4.3.2 below also prove the importance of 2D skeleton features and the validity of our proposed mapping relation for training phase.



OpenPose	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NTU-RGB+D	4	21	9	10	11	5	6	7	17	18	19	13	14	15	4	4	4	4

Fig. 2 The corresponding relation between joints locations of OpenPose and NTU-RGB+D form.

3.2.2. Action clip to 2D skeleton

The dataset acquired by ordinary optical cameras can also be represented as 2D skeleton sequences. Similarly, we transform those action clips to the skeleton sequences by OpenPose. The method for 2D skeleton is outlined as followed.

- Step1: The extracted feature maps are obtained by feeding the given frames from each action clip into the first 10 layers of VGG-19 network.
- Step2: The process of feature representations is carried out in a continuous multi-stage network. Each stage possesses two branches, one predicts the S^t denoted as part confidence maps and the other outputs the L^t described as the part affinity fields.
- Step3: The coordinate points of S^t are connected to generate the skeletons with the help of L^t.
- Step4: The nearest neighbor interpolation is introduced as a compensation of missing keypoints when dealing with the predictions in each frame.
- Step5: A conversion of data format is conducted for each action clip.

More information is described in Sec.4.1.

3.3. Polar Representation of 2D Skeleton

As a classical spatial modeling method, polar representation indeed carries distinct information and has different significance in the context of location and motion description[39]. Some actions exhibit different characteristics in the polar coordinate such as falling down, waving hand and so on. Therefore, polar representation is used as the supplement and enhancement of Cartesian representation in the task of action detection, especially fall detection.

In the work of ST-GCN, a skeleton sequence is represented by the Cartesian coordinates of each human joint in each frame, utilizing the spatial temporal graph to form hierarchical representation of the skeleton sequences. In this graph, the node set $V = \{v_{ti} \mid t = 1,\ldots,T, i = 1,\ldots,N\}$ includes all joints in a skeleton sequence. As ST-GCN's input, the feature vector in the Cartesian coordinate on a node $F_c(v_{ti})$ consists of Cartesian coordinate vectors and estimation confidence of the *i*-th joint on frame *t*[5]. The location of the *i*-th joint on frame *t* can be represented as (x_{ti}, y_{ti}) in the Cartesian coordinate.

Similarly, the feature vector in the polar coordinate on a node $F_p(v_{ti})$ consists of polar coordinate vectors and its estimation confidence. The location of the *i*-th joint on frame *t* can be represented as (r_{ti}, θ_{ti}) in the polar coordinate. Here, r_{ti} is the radial coordinate which represents the distance from the pole (the origin of the polar coordinate), and θ_{ti} is the angular coordinate which represents the angle from the horizontal direction.

With the location of the 2D skeleton in the Cartesian coordinate obtained by the method described in Sec.3.2, we can get the polar representation of the 2D skeleton in the following way. Here, we choose the joint (x_{t10}, y_{t10}) as the pole (x_{tp}, y_{tp}) in the polar coordinate, which is the 10-th joint of OpenPose form in each skeleton sequence, as shown in **Fig. 2**.

It is well established that the Cartesian coordinates (x_{ti}, y_{ti}) can be represented by polar coordinates (r_{ti}, θ_{ti}) using the trigonometric functions sine and cosine as shown in:

$$\begin{cases} x_{ti} = r_{ti}\cos\theta_{ti} + x_{tp} \\ y_{ti} = r_{ti}\sin\theta_{ti} + y_{tp} \end{cases}, \; r_{ti} \geq 0 \text{ and } \theta_{ti} \in [0, 2\pi) \quad (3)$$

Inversely, the polar coordinates (r_{ti}, θ_{ti}) can be converted from the Cartesian coordinates (x_{ti}, y_{ti}) as shown in:

$$\begin{cases} r_{ti} = \sqrt{(x_{ti} - x_{tp})^2 + (y_{ti} - y_{tp})^2} \\ \theta_{ti} = \arctan\dfrac{y_{ti} - y_{tp}}{x_{ti} - x_{tp}} \end{cases}, \; r_{ti} \geq 0 \text{ and } \theta_{ti} \in (-\frac{\pi}{2}, \frac{\pi}{2}) \quad (4)$$

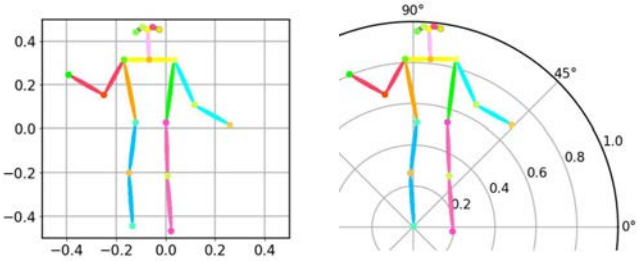


Fig. 3 The representation of 2D skeleton with different coordinates.

According to the quadrant of $(x_{ti} - x_{tp}, y_{ti} - y_{tp})$, the angle θ_{ti} is converted to the interval $[0, 2\pi)$. And then, using the relation between radians and degrees, θ_{ti} is converted to the range $[0^\circ, 360^\circ)$. After that, these polar coordinates are used as another input of ST-GCN. The representations of 2D skeleton with different coordinates are shown in **Fig. 3**.

3.4. CPS-GCN

The idea of two-stream is often applied to the task of action recognition in recent years[40][21]. In this paper, we propose explicitly spatial modeling in the polar coordinate combined with that in the Cartesian coordinate as a two-stream architecture to enhance the recognition.

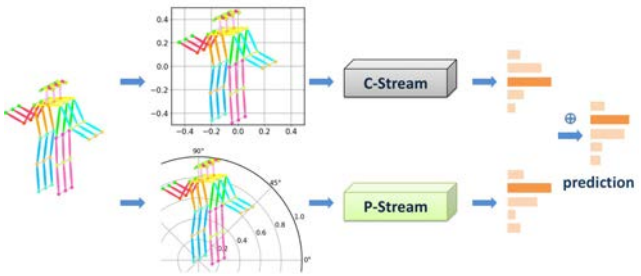


Fig. 4 The two-stream architecture of CPS-GCN.

The Cartesian and polar coordinate networks are represented by C-Stream and P-Stream, respectively. The two-stream architecture of CPS-GCN is shown in **Fig. 4**. The model in each stream is formulated on top of a sequence of skeleton graphs represented in the Cartesian coordinate and in the polar coordinate separately, and multiple layers of spatial temporal graph convolution are constructed thereon. The network of each stream is the same as ST-GCN, which is consists of a batch normalization (BN) layer, 9 ST-GCN units, a global average pooling (GAP) layer and a fully connected (FC) layer. The ST-GCN unit is the spatial temporal graph convolution operator as defined in Sec.2.2.2, and the Resnet mechanism is applied on each unit. The information of location and motion is integrated with both the spatial and the temporal dimension in each stream, and the information in different streams is ultimately fused by weighted averaging, to be the complement and enhancement for each other.

To be more specific, for a input sample, the polar coordinates of 2D skeleton are calculated based on the Cartesian coordinates as described in Sec.3.3 firstly. And then the skeleton features represented in the Cartesian coordinate and the polar coordinate are fed into the C-Stream and the P-Stream in parallel. Finally, the output of the two streams are added to obtain the softmax score and the predicted action label. The final score vector $\mathbf{z}_{cp}$ is represented as:

$$\mathbf{z}_{cp} = softmax(w_c \mathbf{z}_c + w_p \mathbf{z}_p) \quad (5)$$

Where $\mathbf{z}_c$ and $\mathbf{z}_p$ denote the output vectors of C-Stream and P-Stream, respectively. w_c and w_p are weights to adjust their respective contribution.

The graph topology in each model is learned in an end-to-end way based on the input data. This data-driven method and two-stream architecture not only lead to greater expressive power and thus higher performance as shown in our experiments in Sec.4.3.1, but also improve the flexibility, generality and scalability of model construction graph to adapt to different data representations.

4. EXPERIMENTS

Since fall is treated as a special action, it is rational to follow the ways of action recognition experiments in fall detection experiments.

In this section, experiments on two action recognition datasets are conducted to verify the efficiency of our method. One is the NTU-RGB dataset, and the other is the ISA dataset.

4.1. Datasets and Evaluation Metrics

NTU-RGB: NTU-RGB+D dataset is currently a large-scale in-house captured dataset with 3D joints annotations for human action recognition task. There are more than 56 thousand video samples and 4 million frames, which collected from 40 distinct subjects with three camera views (-45°, 0°, 45°) recorded simultaneously[5][6]. It is well known that each video contains RGB, depth, skeleton, and infrared information provided by Microsoft Kinect v2 sensors. Skeleton information consists of 3D locations of 25 major body joints for detected and tracked human bodies. The corresponding pixels on RGB frames and depth maps are also provided for each joint on every frame. Here, we only select the locations on RGB frames to get the 2D skeleton. Considering the final input of ST-GCN is a uniform format that represented as the tensors of (3, *T*, 18, 2) dimensions, we guarantee each clip having at most 2 subjects as described in [5].

In a word, the content of each sample mainly includes video name, resolution, number of frame, category ID, frame ID, person ID and the 2D positions of keypoints, and the dataset is represented as NTU-RGB. The sample number statistics of the NTU-RGB dataset are shown in **Table. 1**. There are two different division modes named cross-subject and cross-view, and we follow the definition in [5] to denote them as X-sub and X-view. It needs to be emphasized that the skeleton information is converted into 2D form, while the corresponding 25 joints are changed into 18 keypoints for each subject. More details of the conversion from 3D features to 2D skeleton representations are discussed in Sec.3.2.1.

Table. 1 The sample number statistics of the NTU-RGB dataset.

Division of train set and validation set	**Division of cross-subject (X-sub) and cross-view (X-view)**	
	# X-sub	***# X-view***
Training set	39772	37338
Validation set	16333	18767

ISA: The Indoor Specific Action (ISA) is a small action recognition dataset, containing 5 action categories of 17 distinct subjects. There are totally 832 video clips with 9 camera views. As shown in **Table. 2**, training set and validation set are divided randomly. The ISA dataset is represented as two different manners which are called cross-subject and cross-view. Just like the NTU-RGB dataset, we abbreviate them as X-sub and X-view separately. In addition, **Fig. 5** illustrates the 9 camera views in 9 different scenes among all 832 action clips. As can be seen from it, some views are relatively monotonous while others are slightly more complex.

Table. 2 The sample number statistics of the IAS dataset.

Division of train set and validation set	**Division of cross-subject (X-sub) and cross-view (X-view)**	
	# X-sub	***# X-view***
Training set	575	613
Validation set	257	219

Fig. 5 The 9 views of the ISA dataset.

Fig. 6 displays 4 primary actions in the ISA dataset, including "fall down", "stand up", "sit down" and "squat down" with their skeletons of OpenPose results. The rest actions do not belong to the classes mentioned above, such as "press up", "push a chair" and so on. Here we classify them as "others". Follow the steps in Sec.3, we briefly obtain the skeleton sequences for each sample and save them as the specific format at the same time. The corresponding operation details can be summarized:

- Clip the original video according to the predefined five action categories. This operation generates 832 action clips in total.
- Use OpenPose to predict 18 joints for each frame in all action clips.
- Bring to nearest neighbor interpolation method for data completion when occurring the missing values of predicted keypoints.
- Save the basic information (video name, resolution, number of frame, category ID, frame ID, person ID and the positions of joints) as the specific format.
- Divide the training set and validation set in two different ways (cross-subject and cross-view).
- Convert the Cartesian coordinates of 2D skeleton to the corresponding polar coordinates.
- Employ the proposed CPS-GCN to compute the fusion score for action prediction.

Fig. 6 The 4 actions with the skeletons of OpenPose keypoints detection result.

Metrics: As is known to all, action recognition is usually regarded as a classification task in video analysis filed. Therefore a generic evaluation criteria *Accuracy* is employed for evaluating the overall performance of multi-classification task. Here C denotes the total number of samples.

$$Accuracy = \frac{1}{C}\sum_{i=1}^{C} f(i) \tag{6}$$

Suppose y_i and y_i' describe the predicted label and ground-truth label, respectively. The $f(\cdot)$ here can be represented as:

$$f(i) = \begin{cases} 0 & \text{if } y_i' \neq y_i \\ 1 & \text{if } y_i' = y_i \end{cases}, \quad i \in [1, C] \tag{7}$$

Taking our target for fall detection task and practical application of health care into consideration, we use F_1 score to serve for the category "fall down" as well. The F_1 measure is defined in:

$$F_1 = \frac{2P(TP, FP)R(TP, FN)}{P(TP, FP) + R(TP, FN)} \tag{8}$$

The $P(\cdot)$ and $R(\cdot)$ are separately denoted as:

$$P(TP, FP) = \frac{TP}{TP + FP} \tag{9}$$

$$R(TP, FN) = \frac{TP}{TP + FN} \tag{10}$$

Where TP, FP and FN denote the number of true positive samples, false positive samples and false negative samples, respectively.

4.2. Training Details

All experiments are conducted in mmskeleton[41] which is a PyTorch deep learning framework. We train the models as the multi-classification task.

For the NTU-RGB dataset, we fine-tune the Kinetics-Skeleton[5][21] pre-trained model. Stochastic Gradient Descent (SGD) with Nesterov momentum (0.9) is applied as the optimization strategy. The batch size is 64. Cross-entropy is selected as the loss function to back-propagate gradients. The weight decay is set to 0.0001. There are at most two people in each sample of the dataset. If the number of bodies in the sample is less than 2, we pad the second body with 0. The max number of frames in each sample is 150. For the sample with less than 150 frames, we repeat the sample until it reaches 150 frames. The learning rate is set as 0.01 and is divided by at the 3th epoch and 6th epoch. The training process is ended at the 9th epoch.

For the ISA dataset, the parameters of training are the same as NTU-RGB. We fine-tune two different pre-trained models respectively, which are the Kinetics-Skeleton pre-trained model and the NTU-RGB pre-trained model. The different experimental results are shown in Sec.4.3.2.

4.3. Results and Comparisons

4.3.1. Comparisons between one-stream and two-stream

An important improvement of our method is the utilization of the features represented in the polar coordinate. Here, we compare the performance of using each type of input data alone and the performance when combining them as described in Sec.3.4. As shown in **Table. 3**, the two-stream method CPS-GCN outperforms ST-GCN that is the original one-stream method in the Cartesian coordinate and the validity of polar representation is proved.

Table. 3 Comparisons of the accuracy and F_1 with methods using different streams on different datasets.

Datasets	Methods	X-Sub		X-View	
		Accuracy	F_1	*Accuracy*	F_1
NTU-RGB	C-Stream(ST-GCN)	79.53%	97.11%	85.82%	99.21%
	P-Stream	76.37%	93.45%	81.71%	93.95%
	2-Stream(CPS-GCN)	80.78%	97.45%	86.81%	99.52%
ISA	C-Stream(ST-GCN)	94.55%	95.08%	89.95%	95.50%
	P-Stream	94.55%	95.24%	91.32%	99.05%
	2-Stream(CPS-GCN)	96.11%	95.16%	90.41%	97.25%

4.3.2. Comparisons between different pre-trained models

In the experiments on our dataset, to verify the influences of different pre-trained models, we fine-tune two different models respectively, which are trained on Kinetics-Skeleton and NTU-RGB. Here, we compare the performance of the two different pre-trained models. As shown in **Table. 4**, the large 2D dataset NTU-RGB which is converted from 3D skeleton, can provide a better pre-trained model. Thus it can be seen that the transformation work described in Sec.3.2.1 is very important.

Table. 4 Comparisons of the accuracy and F_1 with different methods using different pre-trained models on the ISA dataset.

Methods	Pre-trained models	X-Sub		X-View	
		Accuracy	F_1	*Accuracy*	F_1
ST-GCN	Kinetics-Skeleton	94.55%	95.08%	89.95%	95.50%
	NTU-RGB	95.72%	95.16%	89.95%	97.25%
CPS-GCN	Kinetics-Skeleton	96.11%	95.16%	90.41%	97.25%
	NTU-RGB	97.28%	97.60%	91.78%	98.15%

4.3.3. Final comparisons

After the comparisons of ablation experiments, we compare the final performances on both the NTU-RGB dataset and the ISA dataset, as shown in **Table. 5**.

Table. 5 Comparisons of the final accuracy and F_1 with different methods on different datasets.

Datasets	Methods	X-Sub		X-View	
		Accuracy	F_1	*Accuracy*	F_1
NTU-RGB	ST-GCN	79.53%	97.11%	85.82%	99.21%
	CPS-GCN	80.78%	97.45%	86.81%	99.52%
ISA	ST-GCN	94.55%	95.08%	89.95%	95.50%
	CPS-GCN + NTU-RGB pre-trained	97.28%	97.60%	91.78%	98.15%

Through the comparisons of the final accuracy and F_1 on both of the NTU-RGB dataset and the ISA dataset, our method achieves better performance than the baseline method ST-GCN which is also the most typical action recognition model based skeleton. Thus, the results presented well prove the effectiveness of our method.

5. CONCLUSIONS

In this paper, we present a two-stream graph convolutional networks structure called CPS-GCN, which combining the skeleton information of Cartesian coordinates and polar coordinates together to tackle the task of fall detection. For the subsequent spatial temporal graph convolutions on skeleton sequences, polar representation is embraced to be another input stream cooperated with the Cartesian representation. To address the lack of training data, a module is designed to convert 3D skeleton features to 2D skeleton features through a predefined mapping relation. Moreover, a simple light-weight action recognition dataset is provided in this paper. Experimental results on two action recognition datasets indicate that the proposed method possesses the ability of capturing motion information in dynamic 2D skeleton sequences and clearly validate our advantages over the baseline method ST-GCN. It also demonstrates the significance and necessity of the transition from 3D to 2D. In the application of real environment, the pipeline is set up by the means of decoding and clipping the monitor video streams, extracting 2D skeleton by OpenPose, calculating the polar coordinates and using CPS-GCN to detect fall accident in a real time. We are looking forward more further research based on ours, to assist in the intelligent home control and health assistance system in the future.

REFERENCES:

[1] A. F. Ambrose, G. Paul, and J. M. Hausdorff, "Risk factors for falls among older adults: A review of the literature," Maturitas, vol. 75, no. 1, pp. 51-61, 2013.

[2] T. H. Tsai, and C.W. Hsu, "Implementation of fall detection system based on 3D skeleton for deep learning technique," IEEE Access, pp. 7:1-1, 2019.

[3] U. Asif, B. Mashford, S. von Cavallar, S. Yohanandan, S. Roy, J. Tang, and S. Harrer, "Privacy preserving human fall detection using video data," Machine Learning for Health (ML4H) at NeurIPS 2019, Proceedings of Machine Learning Research, 2019, pp. 1-12.

[4] A. N. Marcos, G. Azkune, and I. A. Carrera, "Vision-based fall detection with convolutional neural networks," Wireless Communications & Mobile Computing, vol. 2017, no. 1, pp. 1-16, 2017.

[5] S. J. Yan, Y. J. Xiong, and D. H. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," AAAI Conference on Artificial Intelligence, 2018, pp. 7444-7452.

[6] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: a large scale dataset for 3D human activity analysis," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1010-1019.

[7] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1302-1310.

[8] C. Liu, C. Lee, and P. Lin, "A fall detection system using k-nearest neighbor classifier," Expert Systems with Applications, vol. 37, no. 10, pp. 7174-7181, 2010.

[9] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Definition and performance evaluation of a robust SVM based fall detection solution," in Proceedings of the 8th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2012, pp. 218-224, Italy, November 2012.

[10] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "Automatic monocular system for human fall detection based on variations in silhouette area," IEEE Transactions on Biomedical Engineering, vol. 60, no. 2, pp. 427-436, 2013.

[11] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 5, pp. 611-622, 2011.

[12] N. Zerrouki and A. Houacine, "Combined curvelets and hidden Markov models for human fall detection," Multimedia Tools and Applications, pp. 1-20, 2017.

[13] L. Alhimale, H. Zedan, and A. Al-Bayatti, “The implementation of an intelligent and video-based fall detection system using a neural network,” Appl. Soft Comput., vol. 18, pp. 59-69, May 2014.

[14] K. Adhikari, H. Bouchachia, and H. Naitcharif, “Activity recognition for indoor fall detection using convolutional neural network,” International Conference on Machine Vision, 2017, pp. 81-84.

[15] K. Jaganathan, S. K. Panagiotopoulou, J. F. McRae, S. F. Darbandi, D. Knowles, Y. I. Li, J. A. Kosmicki, J. Arbelaez, W. Cui, G. B. Schwartz, E. D. Chow, E. Kanterakis, H. Gao, A. Kia, S. Batzoglou, S. J. Sanders, and K.-K.-H. Farh, “Predicting splicing from primary sequence with deep learning,” Cell, vol. 176, no. 3, pp. 535-548, Jan. 2019.

[16] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1110-1118.

[17] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Actional-structural graph convolutional networks for skeleton-based action recognition,” in 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3595-3603.

[18] H. Liu, J. Tu, and M. Liu, “Two-stream 3D convolutional neural network for skeleton-based action recognition,” arXiv: Computer Vision and Pattern Recognition, arXiv:1705.08106, 2017.

[19] J. Liu, A. Shahroudy, D. Xu, and G. Wang, “Spatio-temporal lstm with trust gates for 3D human action recognition,” in European Conference on Computer Vision, Springer, 2016, pp. 816-833.

[20] L. Shi, Y. F Zhang, J. Cheng, and H. Q. Lu, “Skeleton-based action recognition with directed graph neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7912-7921.

[21] L. Shi, Y. F Zhang, J. Cheng, and H. Q. Lu, “Two-stream adaptive graph convolutional networks for skeleton-based action recognition,” in 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12026-12035.

[22] C. Y. Si, Y. Jing, W. Wang, L. Wang, and T. N. Tan, “Skeleton-based action recognition with spatial reasoning and temporal stack learning,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 106-121.

[23] S. J. Song, C. L. Lan, J. L. Xing, W. J. Zeng, and J. Y. Liu, “An end-to-end spatio-temporal attention model for human action recognition from skeleton data,” arXiv: Computer Vision and Pattern Recognition, 2016.

[24] Y. S. Tang, Y. Tian, J. W. Lu, P. Y. Li, and J. Zhou, “Deep progressive reinforcement learning for skeleton-based action recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5323-5332.

[25] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3D skeletons as points in a lie group,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 588-595.

[26] W. Zheng, L. Li, Z. X. Zhang, Y. Huang, and L. Wang, “Skeleton-based relational modeling for action recognition,” arXiv: Computer Vision and Pattern Recognition, arXiv: 1805.02556, 2018.

[27] P. F. Zhang, C. L. Lan, J. L. Xing, W. J. Zeng, J. R. Xue, and N. N. Zheng, “View adaptive recurrent neural networks for high performance human action recognition from skeleton data,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2136-2145.

[28] K. Cheng, Y. F. Zhang, X. Y. He, W. H. Chen, J. Cheng, and H. Q. Lu, “Skeleton-based action recognition with shift graph convolutional network,” in 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[29] Q. H. Ke, M. Bennamoun, S. J. An, F. Sohel, and F. Boussaid, “A new representation of skeleton sequences for 3D action recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4570-4579.

[30] T. S. Kim and A. Reiter, “Interpretable 3D human action analysis with temporal convolutional networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) workshops (CVPRW), 2017, pp. 1623-1631.

[31] C. Li, Q. Y. Zhong, D. Xie, and S. L. Pu, “Skeleton-based action recognition with convolutional neural networks,” in 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2017, pp. 597-600.

[32] K. Wang, G. Cao, D. Meng, W. Chen, and W. Cao, “Automatic fall detection of human in video using combination of features,″ 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2016, pp. 1228-1233.

[33] Z. Y. Huang, Y. Liu, Y. J. Fang, and B. K. P. Horn, “Video-based fall detection for seniors with human pose estimation,” 2018 4th International Conference on Universal Village (UV), 2018.

[34] M. Li, S. H. Chen, X. Chen, Y. Zhang, Y. F. Wang, and Q. Tian, “Actional-structural graph convolutional networks for skeleton-based action recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3595-3603.

[35] B. Li, X. Li, Z. F. Zhang, and F. Wu, “Spatio-temporal graph routing for skeleton-based action recognition,” in Thirty-Third AAAI Conference on Artificial Intelligence, 2019.

[36] G. Hidalgo, Z. Cao, T. Simon, S. E. Wei, H. Joo, and Y. Sheikh, “OpenPose”, CMU Perceptual Computing Lab. [Online]. Available: https://github.com/CMU-Perceptual-Computing-Lab/openpose [accessed June, 2020].

[37] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserma, “The Kinetics human action video dataset,” Computer Vision and Pattern Recognition, ArXiv:1705.06950, 2017.

[38] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: common objects in context,” European Conference on Computer Vision (ECCV), 2014, pp. 740-755.

[39] Y. Adato, T. Zickler, and O. Ben-Shahar, “A polar representation of motion and implications for optical flow,” in 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1145-1152.

[40] K. Simonyan, and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” Advances in neural information processing systems, 2014, pp. 568-576 .

[41] S. J. Yan, Y. J. Xiong, J. B. Wang and D. H. Lin, “MMSkeleton”, CUHK Multimedia Lab (MMLab). [Online]. Available: https://github.com/open-mmlab/mmskeleton [accessed June, 2020].