Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042561

ARTICLE

Dense Spatial-Temporal Graph Convolutional Network Based on Lightweight OpenPose for Detecting Falls

Xiaorui Zhang[1,2,3,*], Qijian Xie[1], Wei Sun[3,4], Yongjun Ren[1,2,3] and Mithun Mukherjee[5]

[1]School of Computer Science, Nanjing University of Information Science & Technology, Nanjing, 210044, China

[2]Wuxi Research Institute, Nanjing University of Information Science & Technology, Wuxi, 214100, China

[3]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, 210044, China

[4]School of Automation, Nanjing University of Information Science & Technology, Nanjing, 210044, China

[5]School of Artificial Intelligence, Nanjing University of Information Science & Technology, Nanjing, 210044, China

*Corresponding Author: Xiaorui Zhang. Email: zxr365@126.com

Received: 04 June 2023 Accepted: 04 September 2023 Published: 31 October 2023

ABSTRACT

Fall behavior is closely related to high mortality in the elderly, so fall detection becomes an important and urgent research area. However, the existing fall detection methods are difficult to be applied in daily life due to a large amount of calculation and poor detection accuracy. To solve the above problems, this paper proposes a dense spatial-temporal graph convolutional network based on lightweight OpenPose. Lightweight OpenPose uses MobileNet as a feature extraction network, and the prediction layer uses bottleneck-asymmetric structure, thus reducing the amount of the network. The bottleneck-asymmetrical structure compresses the number of input channels of feature maps by 1×1 convolution and replaces the 7×7 convolution structure with the asymmetric structure of 1×7 convolution, 7×1 convolution, and 7×7 convolution in parallel. The spatial-temporal graph convolutional network divides the multi-layer convolution into dense blocks, and the convolutional layers in each dense block are connected, thus improving the feature transitivity, enhancing the network's ability to extract features, thus improving the detection accuracy. Two representative datasets, Multiple Cameras Fall dataset (MCF), and Nanyang Technological University Red Green Blue + Depth Action Recognition dataset (NTU RGB + D), are selected for our experiments, among which NTU RGB + D has two evaluation benchmarks. The results show that the proposed model is superior to the current fall detection models. The accuracy of this network on the MCF dataset is 96.3%, and the accuracies on the two evaluation benchmarks of the NTU RGB + D dataset are 85.6% and 93.5%, respectively.

KEYWORDS

Fall detection; lightweight OpenPose; spatial-temporal graph convolutional network; dense blocks

1 Introduction

Falling, as the number one killer of elderly injury and death, has great harm to the health of the elderly [1]. According to statistics, the incidence rate of falls among the elderly is 31.8%, and 32.5%

of the falls require medical treatment and even endanger life [2]. Falls are most common in the home environment [3], which threatens the elderly who live alone more severely than others. Given that the elderly living alone stay on the floor longer when they fall without the help of others, it is more prone to result in health problems such as dehydration, internal bleeding, and even death. Therefore, there is an urgent need for an effective automatic fall detection system that can help the elderly timely after a fall.

Researchers have carried out a lot of research in this area and made a significant advance. Currently, many solutions for fall detection have been proposed. These solutions can be broadly classified into three categories: wearable non-visual sensors-based approaches, environmental sensors-based approaches, and computer vision-based approaches.

The method based on wearable usually collects human motion data using portable sensors such as embedded bracelets and belts, and then implements fall detection through Support Vector Machine (SVM) or set threshold [4–6]. However, the method is likely to fail to collect reliable data, e.g., people may forget to wear the device, especially for the elderly or people with dementia [7]. The method based on environmental sensors is to detect whether a human falls by placing sensors in the detection area, in which commonly used sensors such as pressure sensors, infrared sensors, and sound sensors are included. The method does not require people to wear any equipment and has good comfort. However, this method has high equipment cost, and long-term exposure to infrared rays will lead to many adverse effects such as premature skin aging, pigment disorder, and eye damage. At the same time, the frequency of infrared rays will be affected by sunlight, which will affect the accuracy of detection. The method based on computer vision is to use some cameras to collect video/image information of the human body, extract human body features by image processing technology, and then analyze the motion state of the human body [8].

Falls can generally be detected by spatial-temporal information [9], optical flow information [10], temporal features [11], and sequence of human skeletons [12]. Among them, the sequences of human skeletons-based methods usually transmit important information that can be used to detect human action [13,14]. At present, OpenPose is mostly used for the extraction of skeletal sequences. This model can realize the tracking of a human face, torso and limbs, and even fingers. It is not only suitable for single people but also for multiple people. However, OpenPose uses Visual Geometry Group (VGG-19) to extract features and uses multiple 7×7 convolutions in the prediction stage, resulting in a large number of parameters and calculations. It is difficult to meet the requirement for low delay and fast response in daily home scenes. Therefore, it is necessary to design a lightweight OpenPose. And the early methods of fall detection using skeletons are to use coordinates of skeletal joints to form feature vectors at each time step, and then analyze the temporal features. But these methods cannot explicitly exploit the spatial relationships between skeletal points. The spatial-temporal graph convolutional network (ST-GCN) [15], which is based on the sequence of skeletal graphs, can focus on the edges of skeletal joints with consistent connectivity and the connected edges of the same skeletal joints in a continuous time step, automatically capturing the spatial information and time dynamic information of the skeletal joints. Although the model has higher resolution, and contains more location information and detail information, it is prone to fall into gradient disappearance due to ignoring shallow features, which affects the detection results.

Therefore, we design a dense spatial-temporal graph convolutional network (DST-GCN) based on lightweight OpenPose for fall detection. The lightweight OpenPose uses MobileNet to replace VGG-19 as the feature extraction network, and adopts a bottleneck-asymmetrical structure for the prediction layer. The bottleneck-asymmetrical design compresses the quantity of input channels of feature maps by 1 × 1 convolution, and replaces the 7 × 7 convolution structure with the asymmetric structure of 1 × 7 convolution, 7 × 1 convolution, and 7 × 7 convolution in parallel. In this way, the amount of calculation of the OpenPose is reduced. And to improve the accuracy of model detection, the dense spatial-temporal graph convolution network divides the multi-layer spatial-temporal graph convolution layer in the model into two dense blocks. In each dense block, all spatial-temporal convolutional layer is connected with the previous spatial-temporal convolution layers among the dense block. In this way, the transitivity of features is strengthened, the network's ability to extract features is strengthened, and the accuracy of human fall detection is improved. Also, a transition layer is added after each dense block, which can relieve the increased calculation problem resulting from the dense connection.

In short, our contributions are summarized as follows:

1. We design a lightweight OpenPose, which first extracts feature through the MobileNet, and then feeds the features into the prediction layer containing the bottleneck-asymmetrical structure to predict the key points of the skeleton. Our method reduces the calculation amount of OpenPose.
2. We propose a dense graph convolutional model for extracting spatial-temporal features. Spatial-temporal graph convolutional layers are densely connected in each dense block to fully extract temporal and spatial features, improving the accuracy of human fall detection.
3. Our method is verified on multiple datasets, experimental results show the proposed network has less computation on the premise of meeting the fall detection accuracy.

The rest of the article will be organized as follows. Section 2 introduces the related work of this paper. Section 3 presents the model proposed in this paper in detail. Section 4 introduces the datasets and hardware used in the experiments, and presents the experimental results and analysis. Section 5 summarizes the content of this paper and gives directions for further research.

2 Related Work

In recent years, due to the elderly population increasing and the falls frequent, research on the fall detection system for the elderly has attracted extensive attention. This paper aims at improving the real-time and accuracy of fall detection by improvement of OpenPose and ST-GCN. The following will introduce our related work through OpenPose and ST-GCN.

2.1 OpenPose

The OpenPose model [16] proposed by Cao et al. is different from the traditional detection methods. It mainly uses Part Affinity Fields (PAF) [17] to perform a bottom-up method for human pose estimation. The structure of OpenPose can be divided into two parts. The role of the first part is to extract features through VGG-19. And the second part is the prediction layer, which is used to obtain the position of skeletal points through the feature. The prediction layer uses multiple stages to extract information on the position of the skeletal points, and the internal structure of the prediction layer uses multiple 7 × 7 convolution kernels in all stages except the first stage.

The detection speed of OpenPose is significantly improved in comparison with the traditional networks, the network's detection accuracy is high. The detected number of people in an image does not significantly affect the detection accuracy and speed of the network. Many researchers have done further research on OpenPose. Reference [18] combined OpenPose with a bidirectional long short-term memory (LSTM) to effectively improve the accuracy of human pose recognition in complex environments. Reference [19] extracted and marked the key points of the human body through OpenPose, and then defected falls through MobileNetV2.

Among the above methods, most of the performance indicators of OpenPose in detection capability are good. However, OpenPose extracts feature through VGG-19, which has a large amount of calculation [20]. Therefore, OpenPose is difficult to apply in practical scenarios.

At present, many researchers have improved OpenPose. Reference [21] proposed a lightweight real-time human pose detection network that used ResNet 18 to extract features. Reference [22] used MobileNet to replace 12 convolution modules in VGG 19, and proposed a real-time detection of the human skeletal points network.

At the same time, the prediction layer uses multiple 7 × 7 convolution kernels. Although a large convolution kernel can obtain a larger receptive field, it also causes a large amount of calculation burden. Therefore, this paper designs a bottleneck-asymmetric structure for the prediction layer based on using Mobilenet to replace VGG 19 to extract features. The bottleneck-asymmetrical structure compresses the number of input channels of feature maps by 1 × 1 convolution, and replaces the 7 × 7 convolution structure with the asymmetric structure of 1 × 7 convolution, 7 × 1 convolution, and 7 × 7 convolution in a parallel manner, which reduces the amount of calculation of the OpenPose. See Section 3.2 of this article for details.

2.2 ST-GCN

Previous deep learning models mostly adopted traditional Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to learn spatial and temporal features, respectively [23,24]. However, the traditional CNN cannot operate on topological maps, and LSTM is computationally intensive, making it difficult to train. To solve the above problems, ST-GCN first proposed a general graph-based framework to model human skeletons, jointly extracting spatial-temporal features from both temporal and spatial dimensions. ST-GCN contains 9 spatial-temporal convolutional layers, and each spatial-temporal convolutional layer contains a spatial convolutional layer, a temporal convolutional layer, and a residual structure. The skeleton sequence graph first obtains the spatial features through the spatial convolution layer, and then obtains the temporal features through temporal convolution. Finally, the residual mechanism is used to fuse the original input and spatial features and temporal features to form the output features of the spatial-temporal graph convolution layer.

At present, many researchers have improved ST-GCN. Reference [25] divided the skeletal sequence into multiple parts (such as head and limbs) according to the body structure, input them into the spatial-temporal graph convolutional layers respectively, and finally merged them into the overall result. Reference [26] took the joint points and skeletons in the graph of skeletal sequence as independent inputs, and constructed a dual-stream spatial-temporal graph convolution network based on different inputs. Reference [27] used a lightweight displacement map operation instead of a convolution operation to obtain spatial-temporal features through spatial displacement maps and time displacement maps. Reference [28] proposed a multimodal feature fusion learning strategy, which uses spatial-temporal graph convolutional networks and one-dimensional (1D) convolution to generate two sets of spatial-temporal kinematic gait features from skeleton sequences. Reference [29] proposed an adaptive multi-level graph convolutional network that uses spatial convolutions to extract spatial features and multi-scale temporal convolutions to capture temporal features. The above algorithms based on ST-GCN have improved the accuracy of human action recognition to a certain extent. However, in the above methods, the correlation between each spatial-temporal graph convolutional layer and the feature transitivity are poor, which limits the capabilities of the spatial-temporal graph convolutional network.

To solve the above problems, we design a dense spatial-temporal graph convolutional neural network. All spatial-temporal graph convolution layers are divided into two dense spatial-temporal graph convolution blocks, and the spatial-temporal graph convolution layer in each dense block is connected by a dense connection mechanism [30], which improves the communication between layers and strengthens the transitivity of features, making more efficient use of features. And in order to solve the problem of increased calculation caused by dense connection, a transition layer is added after each dense block. See Section 3.3 of this article for details.

3 Proposed Method

In this section, we design a Dense Spatial Temporal Graph Convolution Network (DST-GCN) based on lightweight OpenPose for fall detection. Next, we will introduce Architecture, Lightweight OpenPose, DST-GCN, and Classifier in detail.

3.1 Architecture

The network framework of our method is shown in Fig. 1. Our method mainly includes lightweight OpenPose, DST-GCN, and a classifier. In our current research, we first use Lightweight OpenPose to obtain skeletal data from ordinary video data, and use the data to construct a spatial temporal skeleton sequence map as input of the dense spatial-temporal graph convolutional network. The multi-layer spatial-temporal graph convolutional network is used for convolution, and a higher-level feature map is gradually generated on the graph. Finally, the Softmax classifier is used to detect whether there is a fall behavior in the video.

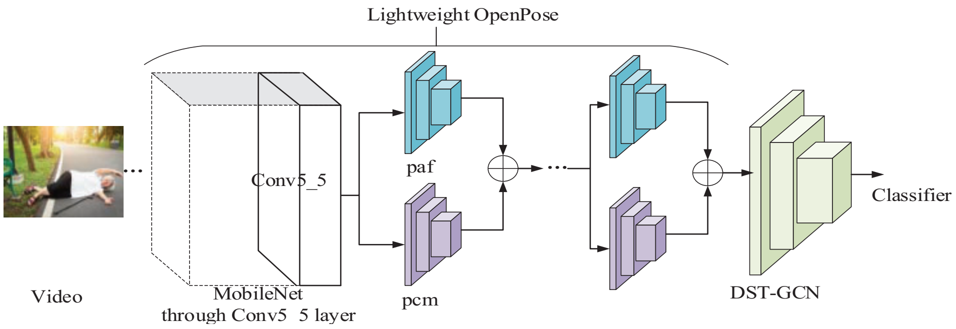


Figure 1: Model for detecting elderly falls

First, the video is sent to lightweight OpenPose, and the skeletal sequence is obtained through MobileNet and prediction layers inside lightweight OpenPose. Then input the skeletal sequence into DST-GCN to get spatial-temporal features. The classifier detects whether is a fall behavior based on the obtained features. Next, we introduce Lightweight OpenPose, DST-GCN, and classifier one by one.

3.2 Lightweight OpenPose

As shown in Fig. 2, the Lightweight OpenPose structure can be roughly divided into two parts. The first part is a feature extraction layer, and the second part is a dual-branch multi-stage prediction layer.

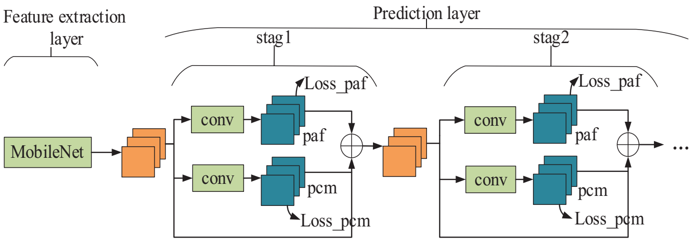


Figure 2: Lightweight OpenPose structure

We use MobileNet to extract features. To save spatial resolution and reuse backbone weights, we use dilation convolution and set the dilation parameter value to 2. And in order to improve the effect of the model, we only use a part of the layers of MobileNet, including from the first layer to the conv5_5 layer.

OpenPose uses a large number of 7 × 7 convolution kernels in the prediction layer. The 7 × 7 convolution kernel can obtain a large receptive field, but it will also cause a large amount of calculation. To alleviate the problem, this paper adopts the bottleneck-asymmetric structure for each 7 × 7 convolution of the prediction layer, as shown in Fig. 3.

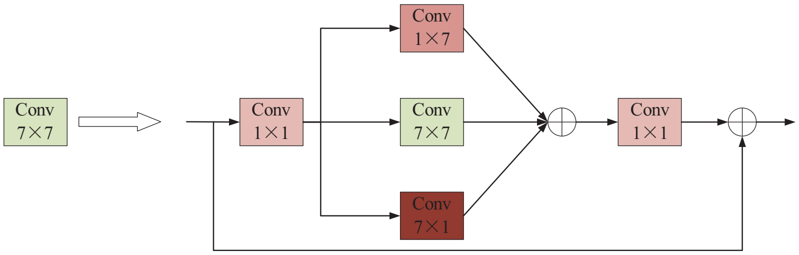


Figure 3: Bottleneck-asymmetric architecture

First, the number of channels of the input features of the asymmetric convolution layer is compressed through 1 × 1 convolution; the compressed features are input into the asymmetric convolution for calculation. After the calculation and fusion are completed, the channel number of the output feature of the convolutional layer is restored by 1 × 1 convolution.

Meanwhile, a single 7 × 7 convolution structure is replaced with a parallel structure of a 7 × 7 convolution, a 7 × 1 convolution, and a 1 × 7 convolution in the asymmetric convolution, which improves model performance without increasing computation. The parallel structure is mainly operated by Batch Normalization (BN) operation and branch fusion. BN is performed after each branch of the parallel structure, and then the outputs of the three branches are fused to get final output.

When the neural network transmits information, there will be a problem of informational loss, we use the residual structure to add the original input feature to the output feature of the above-mentioned asymmetric-bottleneck structure. The residual structure solves the problem of informational loss by directly transmitting the input feature to the output.

The aforementioned lightweight OpenPose is used to obtain the spatial-temporal skeletal sequence from the video, and then the spatial-temporal skeletal sequence is sent to DST-GCN to extract the spatial-temporal features. Finally, the features are sent to the classifier to detect whether there is falling behavior.

3.3 DST-GCN

To strengthen the transitivity of features and the ability of the model to extract features, we adopt a dense connection mechanism in the multi-layer spatial-temporal graph convolutional layer. The nine-layer spatial-temporal graph convolutional structure includes two dense blocks. The first five layers are one dense block, and the last four layers are one dense block. In each dense block, every layer of spatial-temporal graph convolution is connected with all layers ahead of itself to strengthen the transitivity of features and improve the reliability of human fall detection.

The dense connection mechanism means that the input of the t layer is not only related to the output of the $t-1$ layer, but also related to the output of every layer in the model. The output of the t layer is denoted as:

$$X_t = H_t\left([X_0, X_1, \ldots, X_{t-1}]\right) \tag{1}$$

where $[X_0, X_1, \ldots, X_{t-1}]$ represents to stack the feature maps of $X_0, X_1, \ldots, X_{t-1}$ in the channel dimension. H represents nonlinear transformation.

A dense connection is realized by cross-layer feature channel splicing. The splicing is to expand the depth of the channel, and increase the number of channels, resulting in an increased model calculation amount. To solve the above problems, a transition layer is designed behind each the dense block to control the model calculation amount. Reduce the number of channels by 1 × 1 convolution layer, and use an average pooling layer with two strides to halve the width and height of the feature map. Fig. 4 shows the structure of dense-spatial temporal graph convolution layer, and Fig. 5 shows the internal structure of each layer of spatial-temporal graph convolution.

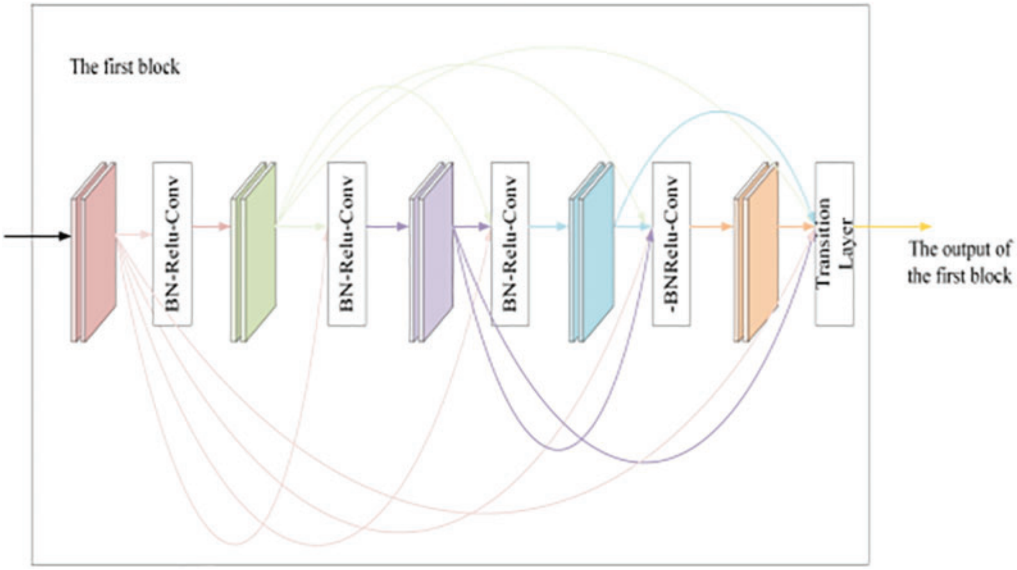


Figure 4: (Continued)

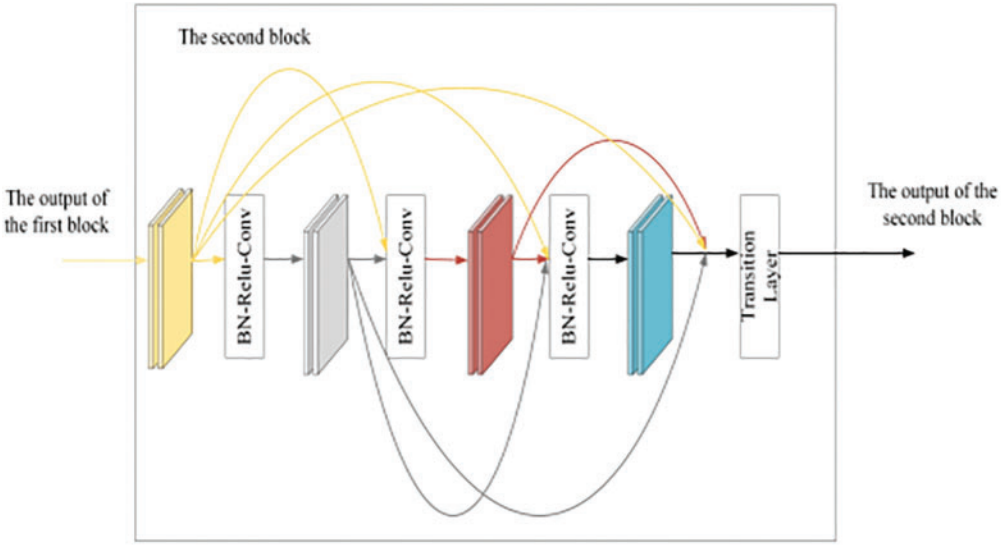


Figure 4: The structure of dense spatial-temporal graph convolution layer

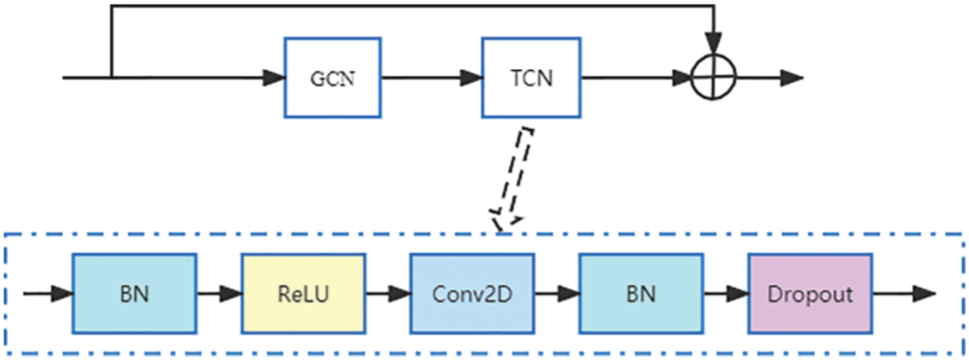


Figure 5: The internal structure of each layer

DST-GCN is used to extract the spatiotemporal features of the bone sequence, and the features are sent to the classifier for classification to detect whether there is a drop in the video.

3.4 Classifier

We use the Softmax classifier to output the result of fall detection. There are two fully connected layers in the classifier. The second fully connected layer does not use dropout and reduces the dimension to the number of categories, and finally outputs the classification result of falls.

This paper uses the binary cross-entropy loss function, and adds the L2 regularization term to further avoid over-fitting of the model. We use stochastic gradient descent (SGD) to constrain each parameter update of the loss function to an appropriate size. The objective loss function including L2 regularization is:

$$L = -\frac{1}{m}\sum\nolimits_{1}^{m} \left[\hat{y}_a log\, y_a + \left(1 - \hat{y}_a\right) log\left(1 - y_a\right)\right] + \lambda\|\theta\|^2 \tag{2}$$

where L represents the target the objective loss function, a represents the sample index, m represents the total number of samples, $\hat{y}_a$ represents the sample label, where the negative class recorded as 0, the positive class recorded as 1, y_a represents the predicted positive probability, and $\lambda\|\theta\|^2$ represents a regular term of L2.

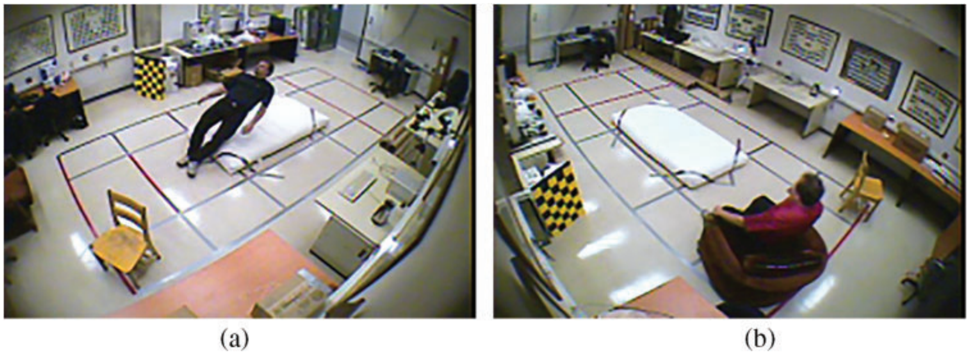
4 Experiment

Below we evaluate the method proposed in this article. First, we will verify our proposed method through ablation experiments and then compare our method with currently existing methods. We will introduce the datasets, evaluating indicators, implementation details, and experimental results from four aspects.

4.1 Datasets

We select two widely used fall datasets to complete the fall detection task, the two datasets are MCF [31] and NTU RGB + D [32].

The MCF dataset contains 24 sets of video data, each captured from 8 ordinary cameras positioned at different angles. Among them, the first 22 sets of data include one or multiple instances of falling behavior, as well as daily activities such as walking, lying down, and some interfering actions like squatting and lying on a sofa. The last two sets of data (23rd and 24th) do not contain any falling behavior but consist of common daily activities. This dataset covers a variety of falling postures, including but not limited to frontal falls, lateral falls, and consecutive falls, enabling an effective evaluation of fall detection algorithms' performance. In this paper, we conducted separate training for the data from the 8 different camera angles. The data was split into training and testing sets at an approximate ratio of 7:3, where sets 8 to 24 were used as the training set, and sets 1 to 7 were used as the testing set. Fig. 6 contains the key frames of the MCF dataset with different 'Fall' and 'No Fall' poses.



(a) (b)

Figure 6: Different frames of multi camera fall dataset (a) fall (b) no fall

The NTU RGB + D is currently the largest indoor action recognition dataset with 60 action categories. In our research, we utilized a subset of samples from the NTU RGB + D dataset, specifically including A8 sitting down, A43 falling down, A80 squatting, A108 flipping down, and A111 walking categories. These behavior categories have ambiguous boundaries, and they exhibit some similarities in motion patterns and postures, making them prone to classification errors and therefore more demanding models. Among these, we selected all samples from the A43 falling down category and also sampled a portion of data from the remaining four categories. By combining these selected samples with the A43 falling down samples, we obtained a comprehensive dataset consisting of 948 falling samples and 1052 non-falling samples. The authors of this dataset proposed two evaluation benchmarks: CS (cross subject) benchmark and CV (cross view) benchmark.

4.2 Evaluating Indicators

We use floating point operations (FLOPs) and accuracy (Acc) to comprehensively evaluate our method. FLOPs are floating-point numbers, which are usually understood as the amount of calculation and can be used to measure the complexity of models. Acc represents the probability of correctly detecting a fall. The specific calculation formula is as follows:

$$FLOPs = 2HW\left(C_{in}K^2 + 1\right)C_{out} \tag{3}$$

$$Acc = \frac{TP + TN}{P + N} \tag{4}$$

where H is the height of the input feature map, W is the width of the input feature map, C_{in} is the number of input channels, K is the size of the convolution kernel, and C_{out} is the number of output channels. The experiment selected the fall action as a positive sample, and the rest of the action as a negative sample. TP (True Positive) in the formula represents a fall sample predicted by the model as fall; TN (True Negative) represents a non-falling sample predicted by the model as non-falling; P and N represent positive and negative samples, respectively. In the application of fall detection, the model Acc should be as high as possible, because there may be serious consequences if it is not detected when the fall occurs.

4.3 Implementation Details

The equipment we used included a computer with an Intel Core i7-10870 CPU and 16 GB RAM, and a remote server consisting of two NVIDIA GeForce RTX 3090 GPUs, and we use Pytorch 1.11 to build our network model.

In this paper, the batch size is set to 32 and trained by SD optimizer. At the same time, we set the momentum size to 0.9, set the weight decay size and the initial learning rate to 0.001, and after every 30 epochs of training, the learning rate becomes nine-tenths of the previous rate. The study was conducted for a total of 90 epochs in training.

4.4 Experimental Results

We divide the experiment into two parts: the ablation experiment and the comparative experiment. The results of these two experiments will be used to illustrate the effectiveness of our method respectively.

4.4.1 Ablation Experiment

In order to analyze the importance of each component to the entire model, we designed ablation experiments. Table 1 shows the comparison between the lightweight OpenPose model in this paper and the original OpenPose model in terms of the calculation amount, and Table 2 shows the comparison of the DST-GCN in this paper and the ST-GCN model. The detection accuracy and the amount of calculation have changed to different degrees, which proves the effectiveness of our proposed method.

Table 1: Ablation experiment results of lightweight models

Methods	FLOPs (million)
Our method	20457
No lightweight of OpenPose	29134

Table 2: Ablation experiment results of dense connection mechanism

Methods	ACC (%)	Data
Our method	96.3	MCF
No dense	93.4	MCF
Our method	85.6	NTU RGB + D (CS)
No dense	82.1	NTU RGB + D (CS)
Our method	93.5	NTU RGB + D (CD)
No dense	89.3	NTU RGB + D (CD)

The calculation amount of the lightweight OpenPose in this paper is 20457 million, and the accuracy of this network on the MCF dataset is 96.3%, and the accuracy on the two evaluation benchmarks of the NTU RGB + D dataset are 85.6% and 93.5%, respectively. First of all, we remove the lightweight processing of lightweight OpenPose, and use the original OpenPose without processing the feature extraction layer and prediction layer to process the output data. The calculation amount of the OpenPose model is 29124 million, which greatly increases the amount of calculation compared with the lightweight structure.

Secondly, we remove the dense connection mechanism of spatial-temporal graph convolution layers, and use the original ST-GCN to detect whether there is falling behavior. The accuracy rates of this network on the MCF dataset and the NTU RGB + D dataset are 93.4%, 82.1%, and 89.3%, all of which have different degrees of decline. In summary, the ablation experiment results show that our method can reduce the computational complexity of the model and improve the detection accuracy.

4.4.2 Comparative Experiments

Our method is characterized by less computation and higher accuracy for fall detection. Table 3 shows the amount of calculation of our method and the other two methods.

Table 3: The comparison between our method and other methods

Methods	FLOPs (million)
Our method	20457
OpenPose [16]	29124
Mask R-CNN [33]	25181

The experimental results show that for the amount of calculation, our method outperforms the other two methods, and the amount of calculation of our method is 20457 million. This is because we use the MobileNet to replace VGG-19 as the feature extraction network, the MobileNet is a lightweight network with less amount of calculation. At the same time, we design a bottleneck-asymmetrical structure for the prediction layer, we use two 1 × 1 convolutions in the bottleneck-asymmetric structure to reduce the amount of calculation by reducing the number of channels in the feature map.

We adopt ST-GCN [15], Deep progressive reinforcement learning for skeleton-based action recognition (DPRL + GCNN) [34], spatial reasoning and temporal stack learning (SRN-TSL) [35] as benchmarks to compare our method, and the results are shown in Table 4. To see the advantages of our method more clearly and intuitively, Fig. 7 is made based on the data in Table 4. It can be found from the experimental results that the detection accuracy of our model is higher than that of the comparative methods. This is because we design a multi-layer spatial-temporal graph convolution structure that includes two dense blocks, every spatial-temporal graph convolution layer in each dense block is connected with all the spatial-temporal graph convolutions ahead of itself. It strengthens the transitivity of features and the model's ability to extract features, and improves the accuracy of human fall detection.

Table 4: The comparison between our method and other methods

Data	Acc (%)	Methods
MCF	96.3	Our method
	93.4	ST-GCN
	90.7	DPRL-GCNN
	91.2	SRN-TSL
NTU RGB + D (CS)	85.6	Our method
	81.5	ST-GCN
	83.5	DPRL-GCNN
	84.8	SRN-TSL
NTU RGB + D (CD)	93.5	Our method
	88.3	ST-GCN
	89.8	DPRL-GCNN
	92.4	SRN-TSL

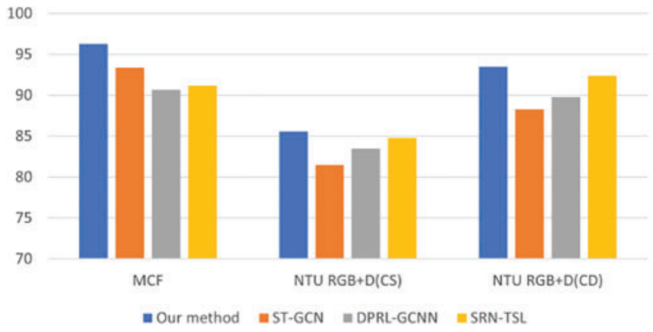


Figure 7: The comparison between our method and other methods

5 Conclusion

In this paper, we propose a dense spatial-temporal graph convolutional network based on lightweight OpenPose. This paper improves the OpenPose model, replaces the feature extraction network, and improves the structure of the prediction layer to reduce the calculation of the model. And this paper adopts a dense connection mechanism for a spatial-temporal graph convolutional network, which strengthens the transitivity of features and the network's ability to extract features, and the accuracy of human fall detection is improved.

In the future, we will further study from the following directions: We will include the model's robustness, fault tolerance, and flexibility in our considerations. Robustness refers to the model's resistance to perturbations and disturbances in the input data. A highly robust model can maintain good performance even when facing noise, outliers, or incomplete data, and it is less affected by such disturbances. A model with high robustness demonstrates better generalization ability in the real world, allowing it to handle diverse data situations. Fault tolerance refers to the model's tolerance towards errors. A model with high fault tolerance can continue to function properly and provide reasonable output results when facing errors, defects, or poor-quality data. Fault tolerance indicates the model's ability to adapt to exceptional situations and ensures system stability and reliability. Flexibility refers to the model's adaptability to different types of data and various problems. A highly flexible model can accommodate diverse data distributions and problems, allowing it to fit different types of samples flexibly. In the design and selection of the model, it is essential to consider these properties. An ideal model should possess robustness and fault tolerance, maintaining stable performance when facing disturbances, errors, and variations. Simultaneously, it should demonstrate a certain level of flexibility, enabling it to adapt flexibly to diverse data and problem scenarios. Such a model can better handle uncertainties and changes in practical applications, exhibiting superior adaptability and reliability.

Acknowledgement: We are grateful to Nanjing University of Information Science and Technology for providing a research environment and computing equipment.

Funding Statement: This study was supported, in part, by the National Nature Science Foundation of China under Grant Numbers 62272236, 62376128; in part, by the Natural Science Foundation of Jiangsu Province under Grant Numbers BK20201136, BK20191401.

Author Contributions: Study conception and design: X. Z., Q. X.; data collection: Q. X.; analysis and interpretation of results: X. Z., Q. X.; draft manuscript preparation: X. Z., Q. X., W. S. and Y. R. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All datasets and materials are publicly available.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Dai, M. Zhang, H. Li and L. Zhao, "Research progress of fall risk assessment among the eldly," *Chinese General Practice*, vol. 22, no. 27, pp. 3347–3352, 2019.

[2] H. Jiang and C. Chen, "Correlation between excretion mode and living ability and the risk of falls in the elderly," *Chinese Journal of Gerontology*, vol. 16, no. 13, pp. 3292, 2019.

[3] J. Pynoos, B. A. Steinman and A. Q. D. Nguyen, “Environmental assessment and modification as fall-prevention strategies for older adults,” *Clinics in Geriatric Medicine*, vol. 26, no. 4, pp. 633–644, 2010.

[4] J. R. Villar, C. Chira, V. M. Gonzalez, S. B. Khojasteh and J. Sedano, “Autonomous on-wrist acceleration-based fall detection systems: Unsolved challenges,” *Neurocomputing*, vol. 452, pp. 404–413, 2021.

[5] Z. Song, J. L. Ou, L. Shu, G. H. Hu, X. M. Xu *et al.,* “Fall risk assessment for the elderly based on weak foot features of wearable plantar pressure,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 1060–1070, 2022.

[6] Z. Q. Qian, Y. C. Lin, W. J. Jin, Z. K. Ma, H. Liu *et al.,* “Development of a real-time wearable fall detection system in the context of Internet of Things,” *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21999–22007, 2022.

[7] Z. Wang, V. Ramamoorthy, U. Gal and A. Guez, “Possible life saver: A review on human fall detection technology,” *Robotics*, vol. 9, no. 3, pp. 55, 2020.

[8] X. R. Zhang, X. L. Zeng, W. Sun, Y. J. Ren and T. Xu, “Multimodal spatio-temporal feature map for dynamic gesture recognition,” *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 671–686, 2023.

[9] L. Sun, K. Jia, D. Y. Yeung and B. E. Shi, “Human action recognition using factorized spatio-temporal convolutional networks,” in *Proc. of ICCV*, Santiago, Chile, pp. 4597–4605, 2015.

[10] L. M. Wang, Y. J. Xiong, Z. Wang, Y. Qiao, D. H. Lin *et al.,* “Temporal segment networks: Towards good practices for deep action recognition,” in *Proc. of ECCV*, Amsterdam, North Holland, NL, pp. 20–36, 2016.

[11] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang *et al.,* “Temporal action detection with structured segment networks,” in *Proc. of ICCV*, Venice, ITA, pp. 2914–2923, 2017.

[12] Y. Z. Li, J. Z. Yuan and H. Z. Liu, “Human skeleton-based action recognition algorithm based on spatio-temporal attention graph convolutional network model,” *Journal of Computer Applications*, vol. 41, no. 7, pp. 1915–1921, 2021.

[13] J. Shi, Y. Zhang, W. Wang, B. Xing, D., Hu *et al.,* “A novel two-stream transformer-based framework for multi-modality human action recognition,” *Applied Sciences*, vol. 13, no. 4, pp. 2058, 2023.

[14] Y. Qin, L. Mo and C. Li, “Skeleton-based action recognition by part-aware graph convolutional networks,” *The Visual Computer*, vol. 36, pp. 621–631, 2020.

[15] S. J. Yan, Y. J. Xiong and D. H. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recogntion,” in *Proc. of AAAI*, New Orleans, Louisiana, USA, 2018.

[16] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei and Y. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.

[17] S. Wei, V. Ramakrishna, T. Kanade and Y. Sheikh, “Convolutional pose machines,” in *Proc. of CVPR*, Las Vegas, NV, USA, pp. 4724–4732, 2016.

[18] Y. Q. Zhou and Y. L. Xu, “Real-time human posture recognition in complex environment based on bidirectional LSTM,” *Chinese Journal of Scientific Instrument*, vol. 41, no. 3, pp. 192–201, 2020.

[19] M. Gao, J. Li, D. Zhou, Y. Zhi, M. Zhang *et al.,* “Fall detection based on OpenPose and MobileNetV2 network,” *IET Image Processing*, vol. 17, no. 3, pp. 722–732, 2023.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.,* “Going deeper with convolutions,” in *Proc. of ICCV*, Boston, MA, USA, pp. 1–9, 2015.

[21] H. K. Zhu, J. W. Yin and W. Y. Feng, “Research and application of a lightweight real-time human posture detection model,” *Journal of System Simulation*, vol. 32, no. 11, pp. 2155–2165, 2020.

[22] L. Schirmer, D. Lucio, A. Raposo, L. Velho and H. Lopes, “A lightweight 2D posemachine with attention enhancement,” in *Proc. of SIBGRAP*, Porto de Galinhas, Brazil, pp. 324–331, 2020.

[23] X. R. Zhang, C. L. Yuan, W. Sun and S. K. Jha, “Image emotion classification network based on multilayer attentional interaction, adaptive feature aggregation,” *Computers, Materials & Continua*, vol. 75, no. 2, pp. 4273–4291, 2023.

[24] D. H. Zhang, N. A. Vien and M. Van, “Non-local graph convolutional network for joint activity recognition and motion prediction,” in *IROS*, Prague, Caech Republic, pp. 2970–2977, 2021.

[25] K. Thakkaer and P. J. Narayanan, “Part-based graph convolutional network for action recognition,” arXiv preprint arXiv:1809.04983, 2018.

[26] X. K. Zhang, C. Chang, X. M. Tian and D. C. Tao, “Graph edge convolutional neural networks for skeleton based action recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3047–3060, 2019.

[27] K. Cheng, Y. F. Zhang, X. Y. He, W. H. Chen, J. Cheng *et al.,* “Skeleton-based action recognition with shift graph convolutional network,” in *Proc. of ICVPR*, Seattle, WA, USA, pp. 183–192, 2020.

[28] M. Amsaprabhaa, Y. N. Jane and H. K. Nehemiah, “Multimodal spatiotemporal skeletal kinematic gait feature fusion for vision-based fall detection,” *Expert Systems with Applications*, vol. 212, pp. 118681, 2023.

[29] P. Geng, H. Li, F. Wang and L. Lyu, “Adaptive multi-level graph convolution with contrastive learning for skeleton-based action recognition,” *Signal Processing*, vol. 201, pp. 108714, 2022.

[30] G. Huang, Z. Liu and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. of CVPR*, Honolulu, HI, USA, pp. 4700–4708, 2017.

[31] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau, “Multiple cameras fall dataset,” DIRO—Université de Montréal, vol. 1350, pp. 1–24, 2010.

[32] A. Shahroudy, J. Liu, T. T. Ng and G. Wang, “NTU RGB+ D: A large scale dataset for 3D human activity analysis,” in *Proc. CVPR*, Las Vegas, NV, USA, pp. 1010–1019, 2016.

[33] K. He, G. Gkioxari, P. Dollar and R. Girshick, “Mask R-CNN,” in *Proc. of ICCV*, Venice, ITA, pp. 2980–2988, 2017.

[34] Y. Tang, Y. Tian, J. Lu, P. Li and J. Zhou, “Deep progressive reinforcement learning for skeleton-based action recognition,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 5323–5332, 2018.

[35] S. Y. Chen, J. Ya, W. Wang, L. Wang and T. Tan, “Skeleton-based action recognition with spatial reasoning and temporal stack learning,” in *Proc. of ECCV*, Munich, Germany, pp. 103–118, 2018.