Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 3955218, 11 pages
https://doi.org/10.1155/2022/3955218

Research Article

Human Behavior Recognition Method Based on Edge Intelligence

Yongxia Sun and Weijin Jiang

School of Computer Science, Hunan University of Technology and Business, Changsha 410205, China

Correspondence should be addressed to Weijin Jiang; jwj3666@163.com

Received 9 March 2022; Revised 31 May 2022; Accepted 4 July 2022; Published 21 August 2022

Academic Editor: Mouquan Shen

The increasingly intelligent video surveillance system is the certain result of the gradual maturity of information technology. Human behavior recognition is one of the important tasks in the area of intelligent security monitoring. This paper proposes a human behavior recognition mechanism that uses edge-cloud collaborative computing. Firstly, at the edge node N_0, the video is preprocessed to remove similar frames and the extracted skeleton sequence is expressed in multiple levels. Then the cloud trains the spatial-temporal graph ConvNet model and deploys it to the edge nodes $N_1 \sim N_m$. The edge uses the trained model to complete behavior recognition tasks and uploads the results to the cloud for fusion to obtain the final behavior category. The experimental results prove that the advantages of edge-cloud collaboration have made the model recognition accuracy rate steadily increase by more than 2.2%.

1. Introduction

With the support of digitization, ultraclearization, networking, and intelligence, video surveillance technology plays a key role in many information scenarios such as smart cities and smart homes. In real life, many people install surveillance cameras in their homes. When they go out, they check the video content to see if the parents need help. However, this method does not reflect the situation on time and also seriously violates the privacy of the elderly. This problem can be solved well by the research results of this paper. The behavior of the elderly is identified by surveillance video; for example, the elderly fall, get out of bed at night, and call for help. An alarm is issued when danger is identified, which makes life safer for the elderly. The interactive development of surveillance systems and big data, artificial intelligence, cloud computing, and Internet of Things technologies has promoted the explosive growth of related applications. However, the massive amount of video image data generated by the existing ultraclear surveillance equipment has caused great pressure on the video surveillance architecture with cloud computing as the core. In response to the above problems, the edge-cloud collaboration proposed a new solution for building a new type of video surveillance system driven by big data. The edge nodes and cloud servers compose an edge-cloud collaboration that uses their advantages to perform computing tasks collaboratively and achieve the overall goal together. Among them, the edge node completes some computing tasks according to the overall arrangement of the cloud server, for example, clean and reduce noise on the data that needs to be preprocessed, eliminate redundant and invalid content, and then upload key data and partial calculation results to the cloud server. The cloud manages all edge nodes as a whole, and continuously optimizes business rules and algorithm models for edge nodes through data analysis and model training to achieve the goal of enhancing edge intelligence and application deployment [1].

Behavior recognition in the surveillance system focuses on how to judge the action being performed by the human body from the video image. However, the image data in many practical application scenarios are generated from non-Euclidean space. The graph structure contains rich semantic information. Features such as disordered nodes, unequal sizes, and different neighborhood sets in the graph lead to irregular graph data. As a result, some important operations, such as convolution, are easy to calculate on the image, but it is not suitable for direct use in the image

domain. As a representative network of deep learning, CNN (Convolutional Neural Network) has advantages such as translation invariance and parameter sharing. Compared with traditional neural networks, its recognition results are more accurate. However, the scope of the application of CNN is still limited to feature extraction in image sequences based on Euclidean space. This paper introduces ST-GCN (spatial-temporal graph ConvNet) to solve the behavior detection and recognition in images generated in non-Euclidean space. Compared with the traditional CNN model, its biggest difference is that it can complete the feature extraction of human joints in space and time in the graph structure data.

The main contributions of this paper are as follows:

(1) It is proposed that the behavior recognition mechanism is based on edge-cloud collaborative computing that extends the previous centralized computing to edge and cloud collaborative processing. The cloud trains the model and deploys it to each edge node. The edge uses the trained model to complete behavior recognition tasks and uploads the results to the cloud for fusion decision-making.

(2) A set of similar frame detection mechanisms is designed to reduce the redundancy of similar images in the video, and the similarity of the two images is calculated by using cosine-perceived hash similarity. The edge device removes extremely similar video frames and retains the seed frame so that the edge device can increase the processing speed while obtaining valuable data.

(3) The ST-GCN model based on the dynamic skeleton sequence is used to complete the human behavior recognition mission, which makes up for the shortcomings of the GCN model based on the spatial domain. The model can extract the spatial features from the graph structure and combine them with the temporal features.

The structure of this paper is arranged as follows: Chapters 1 and 2 outlines the work related to edge-cloud collaborative computing and behavior recognition. Chapter 3 introduces the overall mechanism of behavior recognition under edge collaborative computing. In chapter 4, first, the removal of similar frames and the representation of the skeleton sequence are carried out, and then the ST-GCN model is formally constructed, and the cloud fusion method is given. Chapter 5 evaluates the effectiveness of the edge-cloud collaborative computer system proposed in this paper through related experiments. Finally, chapter 6 makes a summary to point out the advantages and disadvantages of the method.

The application of behavior recognition technology in the surveillance system for real-time video detection and video content analysis can achieve true intelligent security and has a wide range of application prospects in the construction of smart cities. Given the massive surveillance video data, the existing cloud computing model can no longer fully and effectively deal with its calculation and processing. The cloud computing model is expanded to edge-cloud collaborative computing to improve the efficiency of the surveillance system. In [2], a model of initial VM fault-tolerant placement for star topological data centers of cloud systems is built on the basis of multiple factors. Then, a heuristic ant colony algorithm is proposed to solve the model. Reference [3] proposes a lightweight Physical Unclonable Function (PUF)-based and cloud-assisted authentication mechanism for multihop body area networks. Such an authentication mechanism can significantly reduce the storage overhead and resource loss in the data transmission process. In [4], the definition, classification, and characteristics of big data are discussed, along with various cloud services, such as Microsoft Azure, Google Cloud, Amazon Web Services, International Business Machine cloud, Hortonworks, and MapR. In [5], this study investigates the problem of finding an optimal offloading scheme in which the objective of optimization aims to maximize the system utility for leveraging between throughput and fairness. In [6], they extend the design to the scenario where the intelligent jammer can launch a hybrid mode jamming attack, and propose a DNN Stackelberg game-based defense scheme. In [7], this paper presents a multiobjective adaptive fast evolutionary algorithm (F-SGCD) for community detection in complex networks. In [8], this paper presents a multiobjective adaptive fast evolutionary algorithm (F-SGCD) for community detection in complex networks. In [9], a spatio-temporal weight coding method based on skeleton features is designed. These methods do not fully consider the spatial relationship of human joints, and this spatial relationship is essential to recognize human behavior, so ST-GCN is introduced to complete the feature extraction of human joints in space and time in graph structure data.

The current hot deep learning has achieved good results in behavior recognition. Among them, the convolutional neural network mainly mines the spatial domain pattern of behavior recognition, and the recurrent neural network mines the time domain pattern of behavior recognition [10]. Convolutional neural networks can greatly improve the expression ability of behavioral features in the spatial domain. In [11], this article investigates the problem of memory event-triggered H∞ output feedback control for neural networks with mixed delays (discrete and distributed delays). In [12], a novel weighted integral event-triggered scheme (IETS) is proposed based on the past information of the system dynamics. In [13], the decomposition model of the convolutional network on the spatio-temporal sequence is studied, that is, the 3D spatio-temporal convolution is solved into a 2D spatial convolution kernel and a 1D temporal convolution layer to complete the representation and recognition of human behavior. In [14], the combination strategy of 2D spatial convolution and 1D time pooling is further studied. In [15], the 2D convolution operation is extended to 3D convolution, and the dual-stream I3D (Inflated 3D ConvNet) is realized. In [16], in order to complete the extraction of the temporal and spatial characteristics of human behavior, a dual-stream pooling network is designed to further enhance the expressive ability of features. In [17], a synchronous appearance and relationship

module SMART is proposed, and the learning of behavior spatio-temporal characteristics is completed by stacking the model. In [18], a multifiber network is designed, and each fiber uses a lightweight convolution, which greatly improves the operating speed of behavior recognition.

In the process of extracting behavioral features, the above methods or models do not consider the different characteristics of different features in the spatio-temporal sequence. This paper believes that these features can better constrain the feature extraction in the spatio-temporal sequence. Therefore, inspired by these previous studies, this paper uses the ST-GCN model based on the dynamic skeleton sequence to complete the behavior recognition task in the video image through edge-cloud collaborative computing, and the simulation experiments confirm the effectiveness of the mechanism.

2. Edge-Cloud Collaborative Computing Process

The overall architecture of the mechanism is composed of the following three parts: the monitoring device, the edge node, and the cloud server end. The specific functions implemented by each part are as follows:

(1) Monitoring equipment: It will collect the video images and upload them to the edge node N_0 connected to it.

(2) The edge: The edge node N_0 removes similar frames from the video image. In the seed frame, the open-source skeleton joint point extraction algorithm alpha pose proposed by Shanghai Jiaotong University is used to estimate the pose of the human body inside the frame. The extracted human skeleton data is transmitted to the adjacent edge nodes $N_1 \sim N_m$. Nodes $N_1 \sim N_m$ use the ST-GCN model to represent the skeleton sequence at multiple levels and construct an undirected spatio-temporal graph $G = (V, E)$.

The behavior recognition mechanism designed in this paper includes two steps of training and recognition under the side-cloud collaboration. Firstly, the edge node N_0 uploads the extracted skeleton features to the cloud server. And the cloud server uses these features to train the spatial-temporal graph ConvNet model, then edge nodes $N_1 \sim N_m$ receive the model issued by the cloud server; Secondly, in the recognition process, the edge node $N_1 \sim N_m$ will input the skeleton sequence into the trained model. When the recognition result is obtained, it will be transmitted to the cloud server for fusion. In the recognition process, the skeleton sequence of the edge node is not uploaded to the cloud server, which reduces the network transmission volume and alleviates network congestion. The computing tasks on the edge nodes are independent of each other during execution and do not interfere with each other, which increases the fault tolerance of this mechanism. Adding nodes can improve the recognition accuracy. Therefore, when computing resources are sufficient, the number of edge nodes can be appropriately increased, which can improve the accuracy while achieving dynamic scheduling of computing resources [19].

(3) The cloud server: In the training process, the cloud server receives the features uploaded by the edge node N_0 and uses these features to train the ST-GCN model, then delivers the trained model to each edge node. In the recognition process, the cloud server will merge the recognition results uploaded by the edge nodes $N_1 \sim N_m$ to make a fusion decision, and complete the determination of the behavior category in the video image [20]. The computing tasks of the recognition process are jointly completed by multiple edge nodes, which gives full play to the computing capabilities of the edge nodes while reducing the computing pressure on the cloud server.

3. Behavior Recognition Method

3.1. Removal of Similar Frames and Construction of Skeleton Sequence Diagrams. This section introduces the edge node N_0 to detect and remove the massive similar frames generated by the surveillance video and construct the skeleton sequence spatio-temporal graph. Without considering Alpha, the video image has three dimensions as follows: low frequency, intermediate frequency, and high frequency. When the video image is processed for dimensionality reduction, it can be found that most of the information contained in the image is low frequency. For example, the arms, legs, and torso of the human body have always existed, but their movements have changed [21]. High-frequency information is the relatively static part of the image, such as the background of the image, the eyes and mouth of the human body, and other details. The low-frequency domain information determines the general structure of the video image, while the high-frequency domain information completes the details in the video image. Therefore, the video image is first scaled and only the brightness information is retained, which can effectively remove the details of the video image and show the low frequency part of the image.

Calculating the similarity of pictures is achieved by comparing the frequency domain information of the pictures. In actual implementation, the first thing is to convert the acquired video pictures from RGB to YCbCr format, and only extract Y among them to participate in the calculation to achieve dimensionality reduction. Then the video image is scaled to a $32 * 32$ real number matrix. The two-dimensional discrete cosine transform is used to decompose the frequency of the image into a cluster and stepwise to make the behavioral characteristics in the image more prominent and easier to handle. The positive transformation formula of the two-dimensional discrete cosine transform is as follows:

$$F(u,v) = \frac{2}{\sqrt{MN}} \sum_{x=0}^{M-1} \cos\frac{(2x+1)u\pi}{2M} \left\{ \sum_{y=0}^{N-1} f(x,y) \cos\frac{(2x+1)v\pi}{2N} \right\}. \tag{1}$$

Among them, $f(x, y)$ is an $M \times N$ digital image matrix. $x = 0, 1, 2 \ldots, M-1$, $y = 0, 1, 2 \ldots, N-1$; $F(u, v)$ is the transformation domain matrix obtained after calculation. $u, v = 0, 1, 2 \ldots, N-1$. The result is a 32 $*$ 32 size matrix.

The video image is calculated by a two-dimensional discrete cosine sine transform to obtain a 32 $*$ 32 matrix. The low-frequency information exists in the upper left part and the high-frequency information exists in the lower right part. Only the 8 $*$ 8 matrix in the upper left part is retained to present the low-frequency information in the picture and the parameters of the 8 $*$ 8 low-frequency region are extracted from the frequency domain matrix. Get D by calculating the mean value of DCT, set the 64-bit Hash value S of 0 or 1, and then compare the two, if $s > D$, it is recorded as 1, if $s < D$, it is recorded as 0.1, 0 is stored in bits. A picture fingerprint can be obtained. Calculate the Hamming distance α according to the hash value S of each image. According to a large number of previous monitoring data tests, the similarity coefficient can be set to 5. When $\alpha < 5$, it is determined that the two images are similar, $\alpha > 5$, it is determined that the two images are not similar. The comparison result of video image similarity is shown in Figure 1.

After removing similar frames, the skeleton extracted from the seed frame is represented in multiple levels. The skeleton sequence is represented by the two-dimensional or three-dimensional coordinates of the key nodes of the human skeleton. In previous studies, only a single feature vector formed by the connection of all related nodes was used to recognize human actions. On this basis, this paper uses spatio-temporal convolution graphs to represent the skeleton sequence at multiple levels and constructs an undirected spatio-temporal graph $G = (V, E)$ on the skeleton point sequence with N nodes and T frames. This sequence has in vivo connections and interframe connections. Among them, the node matrix set $V = \{V_{ti} | t = 1, \ldots T, i = 1, \ldots, N\}$ includes all the joint points in the bone sequence [22]. As the input of ST-GCN, the feature vector of node F is composed of the coordinate vector and confidence of the i node in the t frame; the edge set E contains two sets. One subset contains the connecting edges of adjacent bone points in the frame, $E_S = \{v_{ti}v_{tj} | (i, j) \in H\}$, the other subset contains the connecting edges of the same bone points between frames, $E_F = \{v_{ti}v_{(t+1)i}\}$.

Constructing a space-time diagram on the skeleton sequence is divided into two steps. First, the open-source skeleton joint point extraction algorithm alpha pose proposed by Shanghai Jiaotong University is used to estimate the pose of the human body inside the video frame. All joints are naturally connected according to the connectivity of the human body structure. Then the connection between the same joint points corresponding to the frame is used to represent the timing relationship of the joint points [23]. Filter the posture joint point data to remove the coordinate matrix with more missing values, at the same time, fill in the coordinate matrix with fewer missing values. Then normalize the position coordinates of all related nodes. The coordinate matrix V is composed of the normalized joint point position coordinates. The connection set up like this is naturally defined and there is no need to manually assign the design so that this model can handle data sets with different numbers of joints or joint connections [24]. For example, using the 2D pose estimation result from alpha pose as the input of the model on the Kinetics dataset, 18 joint points are generated, while the 3D skeleton is used as the input of the model on the NTU-RGB + D 120 dataset, and 25 joint points are generated. The ST-GCN model works well with different joint points and maintains the same superior performance.

3.2. Spatio-Temporal Graph Convolutional Neural Network Modeling. According to the definition of a two-dimensional image or feature map convolution operation, the input image will be regarded as a two-dimensional grid and the output feature map after convolution operating will be a two-dimensional grid as well. When the appropriate step size is selected, the output feature map and the size of the input image remain the same. The following discussion builds on the above foundation, suppose a $K \times K$ convolution kernel, f_{in} is the input image, and c is the number of channels. The output of a single channel at position x is

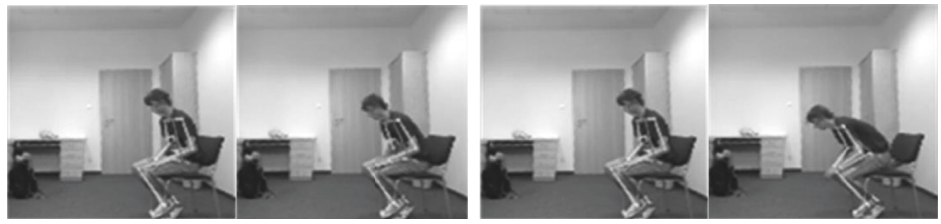
$$f_{\text{out}}(\mathbf{x}) = \sum_{h=1}^{K} \sum_{w=1}^{K} f_{\text{in}}(p(x,h,w)) \cdot w(h,w). \tag{2}$$

Among them, Sample function p: $Z^2 \times Z^2 \longrightarrow Z^2$ represents position x and its neighborhood. In image convolution, $p(x, h, w) = x + p'(h, w)$, Weight function w: $Z^2 \longrightarrow R^c$ represents the c dimension weight vector in real space. It calculates the inner product of the input feature vector in the c-dimensional sample. The value of the weight-dependent variable has nothing to do with the position of the input x, therefore, the filter weights involved in the input image can be shared. By encoding the rectangular grid in $p(x)$, the standard convolution of the image domain is realized. Apply the convolution operation of the above formula to the input features of the spatial graph V_t. The input feature map f_{in}^t: $V_t \longrightarrow R^c$ is any node in the image. The two functions are now optimized to make the model applicable to space-time graphs.

In the video image, $p(h, w)$ in the sample function is the neighborhood pixel of the center pixel x. In the space-time graph, the neighborhood set of a node can be similarly defined $B(v_{ti}) = \{v_{tj} | d(v_{tj}, v_{ti}) \le D\}$, $D = 1$, this is, take the neighborhood set with a distance of 1. Among them, $d(v_{tj}, v_{ti})$ represents the shortest path from v_{tj} to v_{ti}. Therefore, the sample function $\mathbf{p}$: $B(v_{ti}) \longrightarrow V$ can be expressed as follows:

$$\mathbf{p}(v_{ti}, v_{tj}) = v_{tj}. \tag{3}$$

In two-dimensional convolution, there is naturally a rigid grid around the center position, and the adjacent pixels have a stable spatial order, which performs the weight function according to the tensor of the spatial order index



$\alpha = 1$ $\alpha = 7$

Figure 1: Cosine-aware hash flow chart.

(c, K, K). For the above-mentioned general graphs with irregular arrangements, [25] proposed the definition of order, which is determined by the process of marking the neighborhood graph around the root node [26]. The weight function is constructed from this. Any neighborhood node is not individually assigned a label, but the neighborhood set $B(V_{ti})$ of a certain articulation point V_{ti} is divided into a stable quantity of K subsets, where each subset is assigned a digital label, and the mapping relationship is: l_{ti}: $B(v_{ti}) \longrightarrow \{0, \ldots, K-1\}$. Therefore, the weight function $\mathbf{w}(v_{ti}, v_{tj})$: $B(v_{ti}) \longrightarrow R^c$ can be carried out by indexing the vector of (c, K) dimensions.

$$\mathbf{w}\left(v_{ti}, v_{tj}\right) = \mathbf{w}'\left(l_{ti}\left(v_{tj}\right)\right). \tag{4}$$

Based on the redefinition of the two functions, now formula (1) is applied to graph convolution.

$$f_{\text{out}}\left(v_{ti}\right) = \sum_{v_{tj} \in B\left(v_{ti}\right)} \frac{1}{Z_{ti}\left(v_{tj}\right)} f_{\text{in}}\left(p\left(v_{ti}, v_{tj}\right)\right) \cdot w\left(v_{ti}, v_{tj}\right). \tag{5}$$

Among them, the regularization term $Z_{ti}(v_{tj}) = |\{v_{tk} | l_{ti}(v_{tk}) = l_{ti}(v_{tj})\}|$ is the cardinality of the corresponding subset, which makes different subsets contribute equally to the output. From formulas (2)–(4), we can get

$$f_{\text{out}}\left(v_{ti}\right) = \sum_{v_{tj} \in B\left(v_{ti}\right)} \frac{1}{Z_{ti}\left(v_{tj}\right)} f_{\text{in}}\left(v_{tj}\right) \cdot w\left(l_{ti}\left(v_{tj}\right)\right). \tag{6}$$

In the case that the input image is regarded as a regular 2D grid, Equation (6) can be operated as a standard 2D convolution. For applying to the 3×3 convolution operation, in the pixel-centered 3×3 grid neighborhood set, it is divided into 9 subsets, each of which contains one pixel. After the spatial map, CNN is defined, and the skeleton sequence is dynamically modeled in space and time. Connect the same points between adjacent frames to form a spatial graph. Now the space map is extended to the space-time map and the neighborhood set is expanded to include the set of time-connected joint points.

$$B\left(v_{ti}\right) = \left\{ v_{qj} \middle| d\left(v_{tj}, v_{ti}\right) \le K, \middle| q - t \middle| \le \lfloor \frac{\Gamma}{2} \rfloor \right\}. \tag{7}$$

Among them, the parameter Γ represents the size of the time kernel, which controls the time range between the neighborhood graphs. The sample function, weight function, and label graph l_{ST} are still needed to accomplish the convolution operation on the space-time map, where the sample function is identical to the sample function in the space graph. Because the time axis is ordered, the mapping function of directly modifying the label map for the spatio-temporal neighborhood with v_{ti} as the root is

$$l_{ST}\left(v_{qj}\right) = l_{ti}\left(v_{tj}\right) + \left(q - t + \lfloor \frac{\Gamma}{2} \rfloor\right) \times K. \tag{8}$$

Among them, $l_{ti}(v_{tj})$ is the label map of v_{ti} single frame. So far, there is a clear convolution operation on the space-time map. After clarifying the representation of the spatio-temporal graph convolution operation, it is particularly momentous to devise a suitable partition strategy to carry out the label map. This paper explores three partitioning strategies because equation (8) can be used to naturally extend it to the space-time domain, so now we simply talk out the single-frame case.

(1) Unified Label. The most straightforward strategy is to concentrate the entire set of neighborhoods in one partition. Among them, the feature vector on any adjacent node has the same inner product of the weight vector. The insufficient of this strategy is that for a single frame of the image. This strategy can calculate the inner product between the weight vector and the average feature vector of all neighboring nodes. For skeleton sequence classification, this operation will lead to the loss of local differential features, so this is not the optimal strategy, expressed as $K = 1, l_{ti}(v_{tj}) = 0$, $\forall i, j \in V$.

(2) Distance Division. Another partitioning strategy is to divide the set by judging the distance $d(\cdot, v_{ti})$ between the remaining nodes and the root node v_{ti}. This set is divided into two subsets, when $d = 0$, it means the root node v_{ti}, when $d = 1$, the subset stores the nodes adjacent to the root node. Thus, different distances correspond to different weight vectors so that local differential characteristics can be modeled [27]. It is expressed as $K = 1$, $l_{ti}(v_{tj}) = d(v_{tj}, v_{ti})$.

(3) Space Configuration. Because human bones are spatially distributed in regions, this specific spatial configuration can be used in the segmentation process. The motion of body parts involves concentric movement and eccentric movement. Based on this, the neighborhood set of a node is divided into three subsets: (1) the root node itself; (2) the afferent group: Nodes closer to the barycenter of the skeleton; (3) the exocentric group. Among them, the average

coordinate of the related nodes in the single frame skeleton diagram is the center of gravity of the overall skeleton. It is expressed as

$$l_{ti}(v_t j) = \begin{cases} 0, & \text{if } r_j = r_i, \\ 1, & \text{if } r_j < r_i, \\ 2, & \text{if } r_j > r. \end{cases} \tag{9}$$

Among them, r_i represents the average distance from the barycenter of the overall framework to each joint point i in all frames of the video image.

Figure 2 above shows the three visualization partitioning strategies proposed by Niepert et al. [28], among them, (a) is the input skeleton example frame, the joint points of the human body are marked with blue dots, and the receiving field with $D = 1$ filter is framed with the red dotted line; (b) Unified labeling strategy, marking every node in the neighborhood with green dots; (c) Distance division strategy, the two subsets involve the root node itself and its adjacent nodes (marked by blue dots); (d) Spatial configuration strategy. The black cross marks the barycenter of the overall framework. The nodes are marked according to the distance from each joint point to the barycenter of the framework and the root node (green). The shorter distance from the barycenter is the centripetal joint point (blue), while the distance between the centrifugal node (yellow) and the barycenter is longer than the root node.

Using a graph convolution implementation method similar to [29], adjacency matrix A and the identity matrix I represent the in vivo connections of joint points in a single video frame. When processing a single frame of the video image, the output result of using the first partition strategy is calculated by formula (10).

$$f_{out} = \Lambda^{-1/2}(A + I)\Lambda^{-1/2} f_{in} W. \tag{10}$$

Among them, $\Lambda^{ii} = \sum_j (A^{ij} + I^{ij})$, add the weight vectors of multiple output channels to get the weight matrix W. Perform $1 \times \Gamma$ standard two-dimensional convolution, then multiply the normalized adjacency matrix $\Lambda^{-1/2}(A + I)\Lambda^{-1/2}$ with the result tensor.

For partitioning strategies with multiple subsets, that is, distance partitioning and spatial configuration partitioning, the above implementation methods are also used. The adjacency matrix is decomposed into several matrices A_j, $A + I = \sum_j A_j$, for example, in the distance division strategy, $A_0 = I$, $A_1 = A$, formula (10) is transformed into formula (11).

$$f_{out} = \sum_j \Lambda_j^{-1/2} A_j \Lambda_j^{-1/2} f_{in} W_j. \tag{11}$$

Among them, $\Lambda_j^{ii} = \sum_k (A_j^{ik}) + \alpha$, set $\alpha = 0.001$ to avoid empty lines of A_j. For each adjacency matrix, a learnable weight matrix M is attached. Use $(A + I) \otimes M$ and $A_j \otimes M$ to replace the matrix $A + I$ in equation (9) and A_j in equation (10). Here $\otimes$ denotes the element-wise product between the two matrices. The mask M is initialized to a matrix of all 1 s [30].

3.3. Cloud Server Fusion Results. Upload the spatio-temporal skeleton features extracted from the edge to the cloud. The cloud will train the above model and deploy it to each edge node. After the edge node is identified, the result will be uploaded to the cloud. The cloud server will fuse to obtain the final behavior category. The final fusion result is

$$y_{fusion} = \underset{i \in \{1, 2, \ldots, C\}}{\arg\max} \left[\sum_{d \in D} \left[l_{ti}(v_{tj}) \right] \right]^{(i)}. \tag{12}$$

Among them, y is the behavior label, and C is the number of behavior types with recognition. The average value of all the recognition results of the edge node is used for the result fusion.

4. Experiment Analysis

In this section, the edge-cloud collaborative computing environment is built and the evaluation indicators are determined. The system utility of the behavior recognition mechanism under the edge-cloud collaborative computing is verified through simulation experiments.

4.1. Experimental Configuration

4.1.1. Experimental Environment. In this experimental system, the edge node uses 6 PCs as the computing platform, and 1 rack server as the cloud server. In order to simulate the situation where the computing power of edge nodes will be restricted in real life, the VMWare virtual machine is used to limit its computing, memory, and storage resources. The hardware parameters of edge nodes and cloud servers are shown in Table 1.

4.1.2. Datasets

(1) NTU-RGB + D 120 [31]. This dataset has the most skeleton data samples, containing 114480 samples and 120 types of behaviors. In the case of different shooting heights and distances, the shooting angle of view was increased to 155 and the measured objects to 106 people. The behavior content covers a wide range and can correctly reflect the behavior recognition methods in actual application scenarios. The data set provides two verification standards, cross-subject, and cross-setup. In the cross-subject, the training set contains behavioral samples of 53 people and the test set contains the remaining 53 people. The cross-setup uses even numbers for the training set and odd numbers for the test set [32]. Therefore, use this data set to complete the verification of the accuracy of the recognition model.

(2) Kinetics. This data set covers 400 human action categories retrieved by YouTube, and any action has at least 300 video clips of about 10 seconds. These actions cover a wide range of categories, including human-object interactions, such as playing a musical instrument, and human-human interactions, such as shaking hands. This data set provides the original video set without skeleton data. According to the recommendations of the data set author, top-1 and top-5 classification accuracy are used to evaluate the recognition

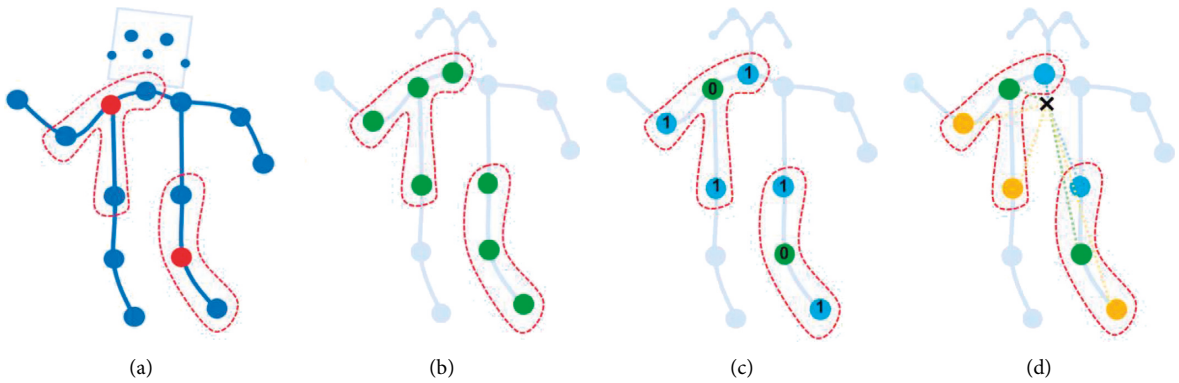


Figure 2: Three partition strategies.

Table 1: Hardware parameters of edge nodes and cloud servers.

Platform	Hardware	Configuration	RAM (GB)	Storage
Edge	PC	CPU: i5-8400 GPU: RTX2060	16	512 GB
Cloud	Rack server	CPU: Xeno-4116 GPU: Tesla 100	128	1 TB

performance. The data set provides 240,000 video training sets and 20,000 verification sets.

4.2. Edge-Cloud Collaborative Computing Advantage Verification. This section compares the utility performance of the behavior recognition method under the edge-cloud collaborative computing proposed in this paper with the following schemes. To ensure the accuracy of the experiment, each experiment was repeated three times.

(1) Behavior recognition strategy under single-edge computing: using the method in [33], the video collected in the monitoring equipment is imported to the PC, where PyCharm is used as the running platform to complete the identification task on the PC.

(2) Behavior recognition strategy under single-cloud computing: upload the video collected from the monitoring equipment to the cloud server, and complete the model training and human behavior recognition tasks on the cloud server.

The network transfer volume is the total amount of data transferred from the edge to the cloud.

The comparative experimental results of the network transmission volume under the three schemes are shown in Figure 3. Under single-edge computing, all tasks are calculated at the edge, so there is no network transmission volume; under single-cloud computing, assuming that the uploaded video has the same size, the network transmission volume shows a linear upward trend with the increase of the number of tasks. The more tasks, the greater the network transmission volume; Under edge-cloud collaborative computing, the edge end needs to upload the extracted skeleton features to the cloud before performing the

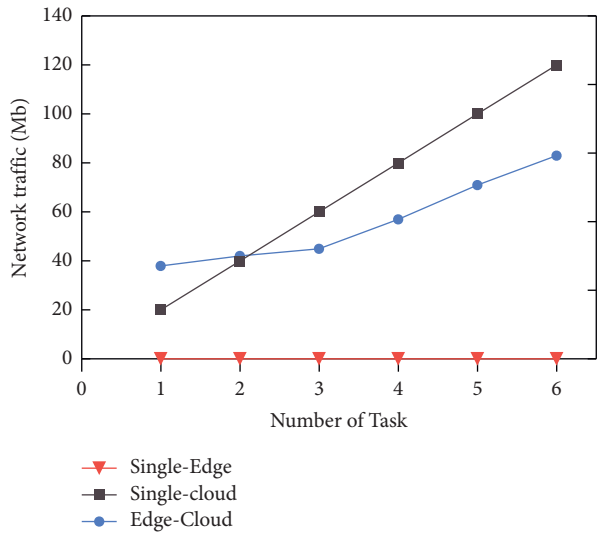


Figure 3: Comparison of the network transmission volume of the three schemes under the same number of tasks.

recognition task, and the cloud uses these features to train the model and then send it to the edge node. Therefore, the network transmission volume to complete the first task is relatively large. When performing subsequent recognition tasks, only the recognition results of the edge nodes need to be uploaded. The increase in network transmission volume slows down, and the more tasks there are, the more obvious the advantages of edge-cloud collaborative computing.

Energy consumption is the average of the proportions of CPU, memory, and hard disk used by each device when performing computing tasks. The comparative experimental results of the energy consumption of the equipment under the three schemes are shown in Figure 4. The energy consumption of edge-cloud collaboration is between single edge and single cloud. It makes full use of the computing resources of the cloud and the storage capacity of the edge while ensuring the completion of the identification task.

The total task time is the total time to complete the data transmission and identification tasks. The comparative experimental results of the total time-consuming tasks under the three schemes are shown in Figure 5. Under single-edge computing, limited by node memory and CPU performance,

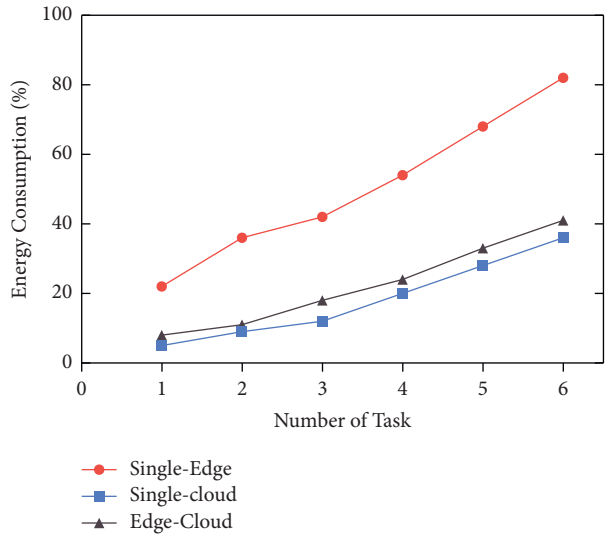


Figure 4: Comparison of energy consumption of the three schemes under the same number of tasks.

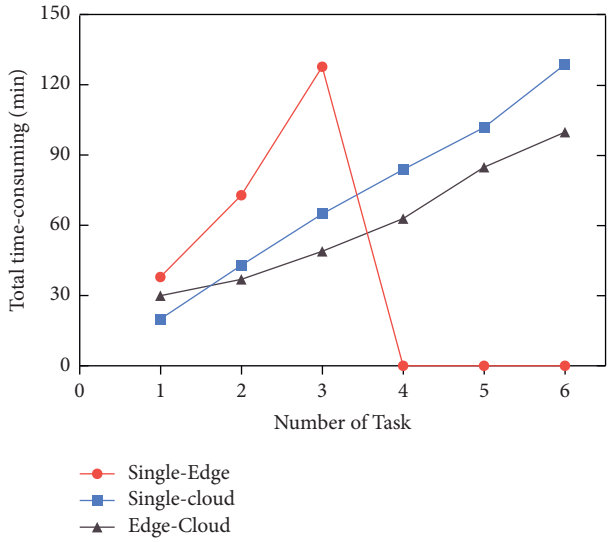


Figure 5: Comparison of the total time-consuming of the three schemes under the same number of tasks.

it takes a long time to complete a single recognition task. As the number of tasks increases, it takes longer and longer to complete tasks. When the upper limit of computing power is reached, the edge recognition task execution is interrupted, the task cannot be completed, and the time-consuming is 0; Under single-cloud computing, the total task time will increase as the number of tasks increases; Under the edge-cloud collaborative computing, the training and deployment of the cloud model before the first task takes more time, so the total time to complete the first task is more than that of a single cloud. However, as the number of tasks increases, the total time to perform the identification task is shorter than in the other two schemes. The superior performance of edge-cloud synergy has been fully verified.

Take a video with a resolution of 1280×720, a time length of 2 s, and a frame rate of 20 fps as an example, the typical bit rate is 1.5 Mbits. When single-edge computing is used, the network transmission volume is 0, but the total time and energy consumption of completing tasks increases significantly with the increase in the number of tasks. And when the number of tasks increased to 4, a situation where the CPU load was too heavy to complete the identification task. When using single-cloud computing, the video needs to be uploaded, and the data volume is about $2\,\text{mbits} \times 2\,\text{s} = 4\,\text{Mb}$. When the number of tasks increases, that is, when the video upload volume is large, network transmission is prone to congestion. When using the edge-cloud collaborative computing in this paper, there is no need to upload the video to the cloud, just upload the recognition result of each edge node, and the amount of data is about $10 \times 50 \times 16\,\text{b} = 8\,\text{kb}$ (A total of 10 models are run on 10 nodes to identify 50 types of behaviors, and the edge node recognition results are represented by 16 bit single-precision floating-point numbers). The amount of network transmission for uploading video image files has been significantly reduced [34]. Comprehensive factors, edge-cloud collaborative computing has shown superior performance in completing human behavior recognition tasks.

4.3. Comparison of ST-GCN Model Recognition Accuracy under Three Partition Strategies. This section verifies the recognition accuracy of the ST-GCN model under the three partitioning strategies based on edge-cloud collaborative computing. Since the edge nodes $N_1 \sim N_m$ are independent of each other when performing recognition tasks, the number of edge nodes can be adjusted in the experiment to observe the changes in the recognition accuracy. The accuracy of model recognition with different numbers of nodes and different partitioning strategies is shown in Figure 6.

Under the three partitioning strategies, when the number of edge nodes is sequentially increased until all the edge nodes are used, the recognition accuracy will increase steadily. Among them, the partitioning strategy of unified labeling may lose local differential characteristics during the calculation process for the classification of skeleton sequences, which shows that the recognition accuracy is low. Compared with the distance division, the model recognition accuracy under the spatial configuration partition strategy is higher. Therefore, in order to improve the application value, the ST-GCN model under the spatial configuration strategy should be selected to recognize human behavior in surveillance videos, and edge resources should be fully utilized for computing tasks in actual scenarios.

4.4. Cloud Integration Effect Verification. This section uses two data sets to verify the effectiveness of cloud server fusion of multiple edge node recognition results. First, complete similar frame removal and pose estimation at the edge node N_0, then complete behavior recognition under nodes $N_1 \sim N_m$, and upload the recognition results to the cloud for fusion. Tables 2 and 3 are the experimental results of the behavior recognition accuracy of NTU-RGB + D 120, Kinetics dataset under the single edge, single cloud, and cloud integration.

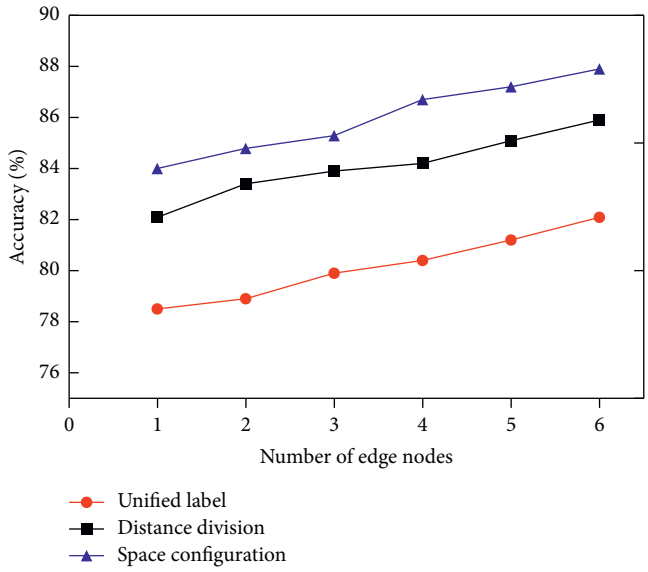


Figure 6: The influence of the number of edge nodes on the recognition accuracy.

Table 2: Edge node and cloud fusion recognition results (accuracy rate).

	Cross-subject	Cross-view
N_1	81.2	87.1
N_2	82.6	88.0
N_3	80.7	86.9
N_4	81.9	87.4
N_5	82.5	87.7
Single cloud	82.1	87.9
Fusion	**83.9**	**89.7**

In the two verification standards, the recognition accuracy rate after fusion is increased, which proves that the advantages of edge-cloud collaboration have made the model recognition accuracy rate steadily increase by more than 2.2%.

Table 3: Edge node and cloud fusion recognition results (accuracy rate).

	Top-1	Top-5
N_1	83.4	85.2
N_2	75.4	86.3
N_3	82.1	84.7
N_4	80.9	86.8
N_5	81.7	84.9
Single cloud	82.6	85.5
Fusion	84.5	88.2

In the Cross-Subject verification standard, the single-cloud recognition accuracy rate is 82.1%, the single-edge recognition accuracy rate is 80.7% to 82.6%, and the average value is 81.7%. The result after fusion on the cloud server is 83.9%. Compared with the former two, the recognition accuracy rate increased by about 2.2%. In the Cross-View verification standard, the single-cloud recognition accuracy rate is 87.9%, the single-edge recognition accuracy rate is 86.9%–88.0%, and the average value is 87.4%. The result after fusion on the cloud server is 89.7%. Compared with the former two, the recognition accuracy rate is increased by about 2.3%. According to the above results, the accuracy of behavior recognition after cloud integration has steadily increased by more than 2%.

In the top-1 verification standard, the single-cloud recognition accuracy rate is 82.6%. The N_2 node has CPU overheating and abnormal operation during the calculation process, which leads to the low recognition accuracy of the node. The recognition accuracy rate of the remaining single-edge nodes is 80.9% to 83.4%, with an average of 82.0%. The result after fusion on the cloud server is 84.5%. Compared with the former two, the recognition accuracy rate is increased by about 2.5%. In the top-5 verification standard, the single-cloud recognition accuracy rate is 85.9%, the single-edge recognition accuracy rate is 84.7%–86.8%, with an average value of 85.5%, and the result after fusion on the cloud server is 88.5%. Compared with the former two, the recognition accuracy rate increased by about 2.7%. According to the above results, when an abnormal situation occurs at an edge node, cloud fusion can still exert good performance, reduce the impact of abnormal results on the final result, and improve the fault tolerance rate for edge-cloud collaborative computing.

Based on the above experiments, in single-cloud computing, the time for the cloud server to complete the identification task is shortened and the accuracy rate is improved. However, the centralized upload of video images is likely to cause network congestion, and the storage of uploaded video images also puts a certain pressure on the cloud server. And due to the lack of frames and missing frames during the transmission of video frame files, it has an impact on the accuracy of the recognition task to a certain extent [35]. The edge-cloud collaborative computing method proposed in this paper not only reduces the network transmission volume and eases the storage pressure of the cloud server, but also improves the recognition accuracy and makes full use of the computing power of edge nodes.

5. Conclusion

This paper studies the human behavior recognition mechanism under side-cloud collaborative computing for massive surveillance video frames. Firstly, design the edge-cloud collaborative computing framework, reasonably allocate and make full use of computing resources at the edge and the cloud, remove similar frames from the video at the edge, and then use the ST-GCN model in the seed frame to recognize human behavior. Finally, the cloud integrates the recognition results of all edge nodes. Through experimental evaluation and analysis, the proposed behavior recognition mechanism under edge-cloud collaborative computing has improved recognition accuracy, and its network transmission volume, equipment energy consumption, and total task time are better than the single edge and single edge, which shows superior mechanism utility and performance. The spatio-temporal graph convolutional neural network model used in this paper can capture the motion information in the

dynamic skeleton [36]. It is a supplement to the previously used RGB model. Its flexibility also opens up many possible directions for future work.

Data Availability

Some or all data, models, or code generated or used during the study are proprietary or confidential in nature and may only be provided with restrictions.

Conflicts of Interest

The authors declared that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (No. 61772196), the Natural Science Foundation of Hunan Province (No. 2020JJ4249), and the Scientific Research Program of the Education Department of Hunan Province (No. 21A0374).

References

[1] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, New Orleans Louisiana USA, February 2018.

[2] W. Zhang, X. Chen, and J. Jiang, "A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2021.

[3] X. Tan, J. Zhang, Y. Zhang, Z. Qin, Y. Ding, and X. Wang, "A PUF-based and cloud-assisted lightweight authentication for multi-hop body area network," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 36–47, 2021.

[4] A. K. Sandhu, "Big data with cloud computing: discussions and challenges," *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32–40, 2022.

[5] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2021.

[6] J. Liu, X. Wang, S. Shen, Z. Fang, and S. Yu, "Intelligent jamming defense using DNN Stackelberg game in sensor edge cloud," *IEEE Internet of Things Journal*, vol. 9, 2021.

[7] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[8] Q. Li, Z. Cao, W. Ding, and Q. Li, "A multi-objective adaptive evolutionary algorithm to extract communities in networks," *Swarm and Evolutionary Computation*, vol. 52, Article ID 100629, 2020.

[9] M. Su, J. Guo, and R. Li, "Resource deployment with prediction and task scheduling optimization in edge cloud collaborative computing," *Journal of Computer Research and Development*, vol. 4, 2021.

[10] J. Chen, Y. Jiang, Y. Xu, Y. Wang, L. Tan, and G. Liang, "A new time-aware collaborative filtering intelligent recommendation system," *Computers, Materials & Continua*, vol. 61, no. 2, pp. 849–859, 2019.

[11] S. Yan, Z. Gu, and S. K. Nguang, "Memory-event-triggered H∞ output control of neural networks with mixed delays," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[12] S. Yan, S. K. Nguang, and Z. Gu, "$H_{\infty }$ weighted integral event-triggered synchronization of neural networks with mixed delays," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2365–2375, 2021.

[13] H. Zhu, C. Zhu, and Z. Xu, "Research progress of human behavior recognition data set," *Acta Automatica Sinica*, vol. 44, no. 6, pp. 978–1004, 2018.

[14] C. Charles, A. VázquezDiosdado Jorge, and K. Jasmeet, "Machine learning algorithms to classify and quantify multiple behaviors in dairy calves using a sensor: moving beyond classification in precision livestock," *Sensors*, vol. 21, no. 1, 2020.

[15] B. Zhang, P. Li, and Q. Sun, "Spatiotemporal feature aggregation convolutional network model based on locally constrained affine subspace coding," *Chinese Journal of Computers*, vol. 43, no. 9, pp. 1589–1603, 2020.

[16] U. Amin, M. Khan, H. Tanveer, and W. Sung, "Conflux LSTMs network: a novel approach for multi-view action recognition," *Neurocomputing*, vol. 435, 2020.

[17] B. Liang and W. Ji, "Multiuser computation offloading for edge-cloud collaboration using submodular optimization," *Journal on Communications*, vol. 41, no. 10, pp. 25–36, 2020.

[18] C. Chen, T. Peng, and Z. Gan, "Aurora image classification and retrieval method based on deep hash algorithm," *Journal of Electronics and Information Technology*, vol. 42, no. 12, pp. 3029–3036, 2020.

[19] Z. Sun, Q. Wang, B. Gao, and L. Zhongjun, "Data field classification algorithm for edge intelligent computing," *Journal of Computer Applications*, 2022.

[20] N. Feng and S. Guo, "Multi-component spatial-temporal graph convolution networks for traffic flow forecasting," *Journal of Software*, vol. 30, no. 3, pp. 759–769, 2019.

[21] T. Ma, "Research and implementation of pedestrian recognition technology based on adaptive feature matching," *Electronic and communication engineering*, 2020.

[22] Y. Zhang, L. Tan, and L. Chen, "Cross-modal pedestrian re-recognition based on image and feature joint constraints," *Acta Automatica Sinica*, 2021.

[23] G. Zou, G. Fu, and M. Gao, "Research progress of metric learning methods in pedestrian re-identification," *Control and Decision*, 2021.

[24] J. Lu, H. Wang, and X. Chen, *Pedestrian Re-identification Based on Multi-Scale Feature Representation Control and Decision*, 2021.

[25] H. Huang, H. Su, Z. Chang, M. Yu, J. Gao, and S. Zheng, "Convolutional neural network with adaptive inferential framework for skeleton-based action recognition," *Journal of Visual Communication and Image Representation*, vol. 73, 2020.

[26] S. Q. Deng and X. G. Ye, "Multi-objective task offloading algorithm based on deep Q-network," *Journal of Computer Applications*, 2022.

[27] W. Du, "Multilayer recurrent neural network for action recognition," *Computer Science and Application*, vol. 10, no. 6, 2020.

[28] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of the*

33rd International Conference on International Conference on Machine Learning JMLR.org, New York NY USA, June 2016.

[29] E. Spyrou, E. Mathe, G. Pikramenos, K. Kechagias, and P. Mylonas, "Data augmentation vs. Domain adaptation-a case study in human activity recognition," *Technologies*, vol. 8, no. 4, p. 55, 2020.

[30] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," *International Journal of Computer Vision*, vol. 128, no. 1, pp. 74–95, 2020.

[31] T. Gan and H. Libo, "Review of winograd fast convolution technique research," *Journal of Frontiers of Computer Science & Technology*, 2022.

[32] Z. Du, D. Yue, Y. Chengqi, H. Boran, and L. Lin, "TID-MOP: the comprehensive framework of security management and control in the scenario of data exchange," *Data Analysis and Knowledge Discovery*, vol. 6, no. 1, pp. 13–21, 2022.

[33] J. Liu, A. Shahroudy, G. Wang, L. Y. Duan, and A. C. Kot, "Skeleton-based online action prediction using scale selection network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1453–1467, 2020.

[34] T. Wang, C. Liu, L. Wang, B. Ma, and X. Gu, "Evolution modeling with multi-scale smoothing for action recognition," *Journal of Visual Communication and Image Representation*, vol. 55, 2018.

[35] M. A. Jingqi, L. E. I. Huan, and C. Minyi, "Fall behavior detection algorithm for the elderly based on AlphaPose optimization model," *Journal of Computer Applications*, vol. 42, no. 1, p. 294, 2022.

[36] Z. Hu, P. Xi, R. Zhang, S. F. Li, and M. Y. Zhao, "Research on 3D multi-branch aggregation lightweight network video behavior recognition algorithm," *Acta Electronica Sinica*, vol. 48, no. 7, pp. 1261–1268, 2020.